DHBW Duale Hochschule Baden-Württemberg Karlsruhe

Netztechnik I

T2INF4201.1

Applikationsschicht
(Application Layer)

Markus Götzl
Dipl.-Inform. (FH)
mail@markusgoetzl.de

http://www.dhbw-karlsruhe.de

Applikationsschicht (Applikation Layer)

nur DNS und Teil von SMTP => gesamte Verbindung zwischen DNS und EMail

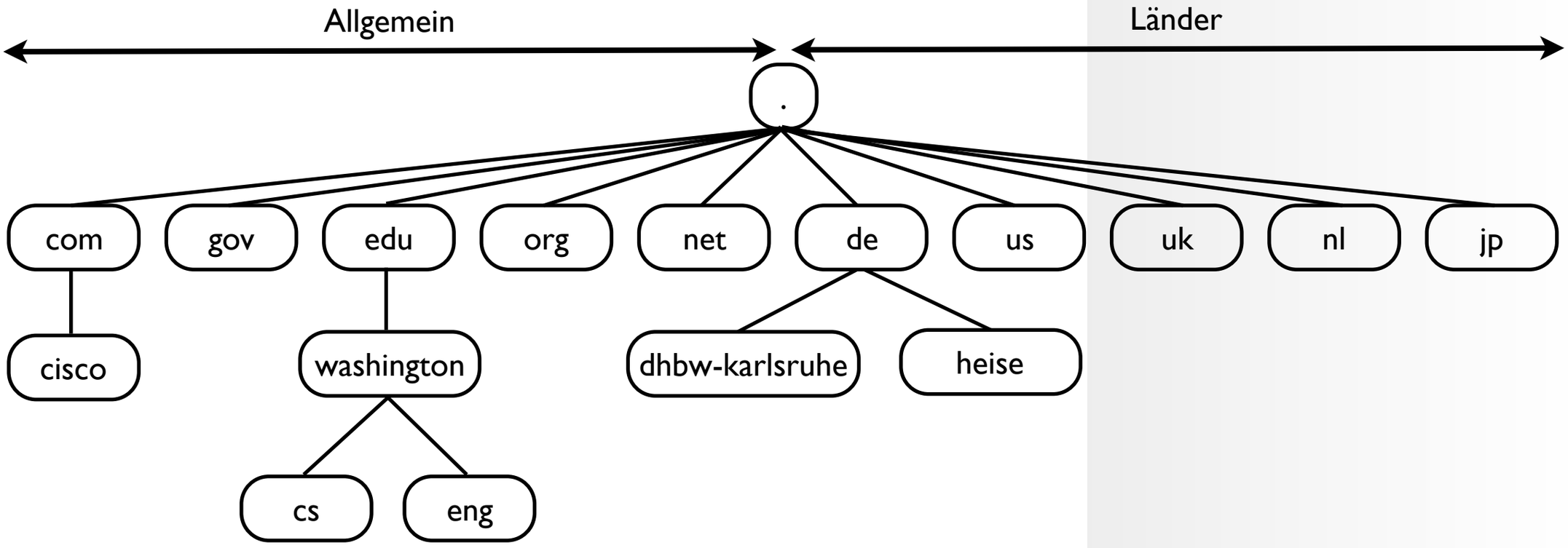
- Domain Name System (DNS)
- Simple Mail Transfer Protocol (SMTP)
- Hypertext Transfer Protocol (HTTP)

Domain Name System (DNS)

- Das Domain Name System ist zuständig für die Auflösung von Domain Namen zu IP Adressen.
  - Domain Namen sind für Menschen einfacher zu merken als IP Adressen
  - Beispiel: `81.26.175.65 -> dhbw-karlsruhe.de`
- Das Domain Name System entkoppelt IP Adressen von den Domain Namen
  - Falls sich die IP Adresse eines Rechners im Internet ändert, kann dieser immer noch unter seines Domain Namen gefunden werden.

Domain Name System (DNS)

DNS Hierarchie



- Die Domain Adressierung ist hierarchisch aufgebaut (ähnlich der Postadressen: Land - Stadt - Nachname - Vorname
  - Die Top-Level Domains werden von der ICANN (Internet Corporation for Assigned Names and Numbers)

DNS Hierarchie

- Das Internet ist in mehr als 250 Top-Level Domains aufgeteilt.
  - Diese Domainverteilung kann als Baum dargestellt werden, wobei jeder Domain aus der ersten Ebene (Top-Level) in jeder weiteren Ebene in neue Domain aufgeteilt wird.
  - Die Blätter des Baumes repräsentieren dann entweder einen Rechner im Internet oder eine ganze Organisation mit u.U. mehren tausend Rechnern.
- Für das Root-Element (.) existieren z.Z. 13 Name Server, diese werden als Root Nameserver bezeichnet und verwalten die Top-Level Domains.
- Die ICANN vergibt die Top-Level Domains an Registrare (z.B. “.net”), welche dann Second-Level Domains an ihre Kunden verlaufen (z.B. “my.net”)

Domain Name System (DNS)

Domain Namen

- Für jedes Land wird eine Top-Level Domain zur Verfügung gestellt (ISO 3166)
- Seit 2010 können Domain Namen in verschieden Alphabeten vergeben werden z.B. in arabisch, kyrillisch oder chinesisch
- Die einzelnen Ebenen werden durch ein “.” getrennt (gesprochen “dot”).
  - Beispiel: host.my.net
- Domain Namen sind “case-insensitive” (Gleichheit der Groß-/ Kleinschreibung)
- Eine Teil-Domain kann bis zu 63 Zeichen umfassen und eine Domain darf 255 Zeichen nicht überschreiten.

Domain Resource Records

- Zu jeder vergebenen Domain wird ein oder mehrere Domain Resource Record(s) angelegt.
- Ein Domain Resource Record ist ein 5-Tupel:
  - Domain Resource Record: `Domain_name, Time_to_live, Class, Type, Value`

Domain Resource Records

- `Domain_name:` Entspricht der Domain und stellt den primär Schlüssel für Suchanfragen dar.
- `Time_to_live:` Dient als Indikator für die Stabilität eines Eintrags und gibt an wie lange ein Eintrag in einem Cache vorgehalten werden kann.
- `Class:` Legt die Klasse des Eintrages fest. Andere Einträge als “IN” - sind sehr selten (CH - Chaosnet, HS- Hesiod)

Domain Resource Records

- Type: Legt den Type eines DNS Eintrags fest. Es gibt eine ganze Reihe unterschiedlicher Typen: SOA, A, AAAA, MX, NS, CNAME, PTR, SPF, SRV, TXT
- `Value:` Enthält den Wert des DNS Eintrags, seine Semantik hängt von dem Typ des DNS Eintrags ab.

Domain Resource Records - Typen

- SOA: Start of authority - dieser Typ wird für Informationen zur Name Server Zone (welche Domains werden von diesem Server verwaltet) genutzt. Zusätlich enthält dieser Typ weitere Informationen zur Zone(Administrator EMail, Seriennummer, Time outs, etc.)
- A: Dieser Typ enthält die 32 Bit IPv4 Adresse des angefragten Rechners.
- AAAA: Analog zum Typ “A” enthält dieser Typ die 128 Bit IPv6 Adresse des angefragten Rechners.
- MX: Dieser Typ enthält den Namen des Rechners der Domain, welcher EMails akzeptiert (Mail Server).

Domain Resource Records - Typen

- NS: Dieser Typ spezifiziert den Name Server für die angefragte Domain oder Subdomain.
- CName: Mit diesem Typ lassen sich Zweitnamen (alias) für spezifische für Rechner mit spezifischen Aufgaben vergeben.
- PTR: Analog zu “CName” lassen sich mit diesem Typ Zweitnamen vergeben. Dieser Typ enthält in der Regel die IP Adresse eines korrespondierenden Rechners. Diese Information kann dann für sog. **Reverse Lookups** verwendet werden (Welchen Namen hat ein Rechner mit der spezifizierten IP Adresse)
- SPF: Sender Policy Framework - hier lassen sich Rechner der Domain definieren welche EMails versenden. Damit kann ein empfangender Rechner überprüfen inwieweit eine EMail valide ist (Spam, EMail Spoofing)

Domain Resource Records - Typen

- SRV: Mit diesem Typ lassen sich Rechner definieren die für spezifische Aufgaben in einer Domain zuständig sind. Beim SRV Typ handelt es sich um eine Generalisierung des “MX” Typs welche nur für den Service EMail gedacht ist.
- TXT: Mit diesem Typ lassen sich zusätzliche frei wählbare Informationen zu einer Domain definieren.

Domain Resource Records - Beispiele

```
dig SOA example.com

;; ANSWER SECTION:
   example.com.   3509 IN SOA dns1.icann.org.
   hostmaster.icann.org. 2012081704 7200 3600 1209600
   3600
```

```
dig MX markusgoetzl.de

; ANSWER SECTION:
markusgoetzl.de. 86400    IN   MX   100
mail.markusgoetzl.de.
```

- SOA:
  - Der zuständige Master Name Server ist dns1.icann.org.
  - Die EMail Adresse des Administrators ist “hostmaster@icann.org” (erster Punkt muss durch ein @ ersetzt werden).
  - Seriennummer ist “2012081704”.
  - Die Zonen Transfer Synchronisation (Master - Salve) findet alle 7200 Sek. (2 Stunden) statt.
  - Ist der Master nicht erreichbar wird alle 3600 Sek. (1Sunde) eine neuer Versuch gestartet.
  - Kann ein Master nach 1209600 Sek. (14 Tage) nicht erreicht werden wird die Zone als inaktiv gesetzt.
  - Die TTL ist 3600 Sek. (1 Stunde) für cached DNS.
- MX:
  - Der Mail Server für die Domain ist “mail.markusgoetzl.de”
  - Priorität/Reihenfolge ist 100.

Domain Resource Records - Beispiele

```
dig NS heise.de

;; ANSWER SECTION:
heise.de.          1679 IN   NS   ns2.pop-hannover.net.
heise.de.          1679 IN   NS   ns.plusline.de.
heise.de.          1679 IN   NS   ns.heise.de.
heise.de.          1679 IN   NS   ns.pop-hannover.de.
heise.de.          1679 IN   NS   ns.s.plusline.de.
```

```
dig A ns.heise.de

; ANSWER SECTION:
ns.heise.de.       3541 IN   A    193.99.145.37
```

```
dig AAAA google.de

;; ANSWER SECTION:
google.de.         54   IN   AAAA 2a00:1450:4016:801::101f
```

- NS:
  - Zuständige Nameserver sind:
    ns2.pop-hannover.net
    ns.plusline.de
    ns.heise.de
    ns.pop-hannover.de
    ns.s.plusline.de
  - Die IP Adressen müssen über erneute abfragen (nach “A” Records) vorgenommen werden.
- A:
  - die IPv4 Adresse ist 193.99.145.37
- AAAA:
  - die IPv6 Adresse ist 2a00:1450:4016:801::101f

Domain Server

- Die DNS Infrastruktur ist hierarchisch aufgebaut.
- Die erste Ebene (Root DNS Server) wird von der ICANN betrieben.
- Alternative: http://www.orsn.org
- Übersicht: http://www.root-servers.org

Domain Server Klassen

- Root Server
  - z.Z. 13 Root Server (die meisten in den USA). Physikalische Server sind es allerdings mehr , da es sich hierbei um ein Netzwerk von replizierten Servern handelt. Die Root Server sind zuständig für die Auflösung der Top Level Domain Name Server (TLD) und werden von der ICANN (Internet Corporation for Assigned Names and Numbers) betrieben.
- Top Level Domain Name Server (TLD)
  - Diese Server sind für die Top Level Domains (TLDs) zuständig (org, com, net, …). Die IANA (Internet Assigned Numbers Authority), als Unterorganisation der ICANN vergibt diese in drei Hauptgruppen: (a) allgemeine TLDs (gTLD - uTLD,sTLD) (b) länderspezifische TLDs (ccTLD) und (c) Infrastruktur TLDs (iTLD)
- Autoritative Nameserver
  - Bei autoritativen (authoritative) Nameserver handelt es sich Nameserver, die offiziell für eine Zone zuständig sind. (Eintrag in NS Typ)

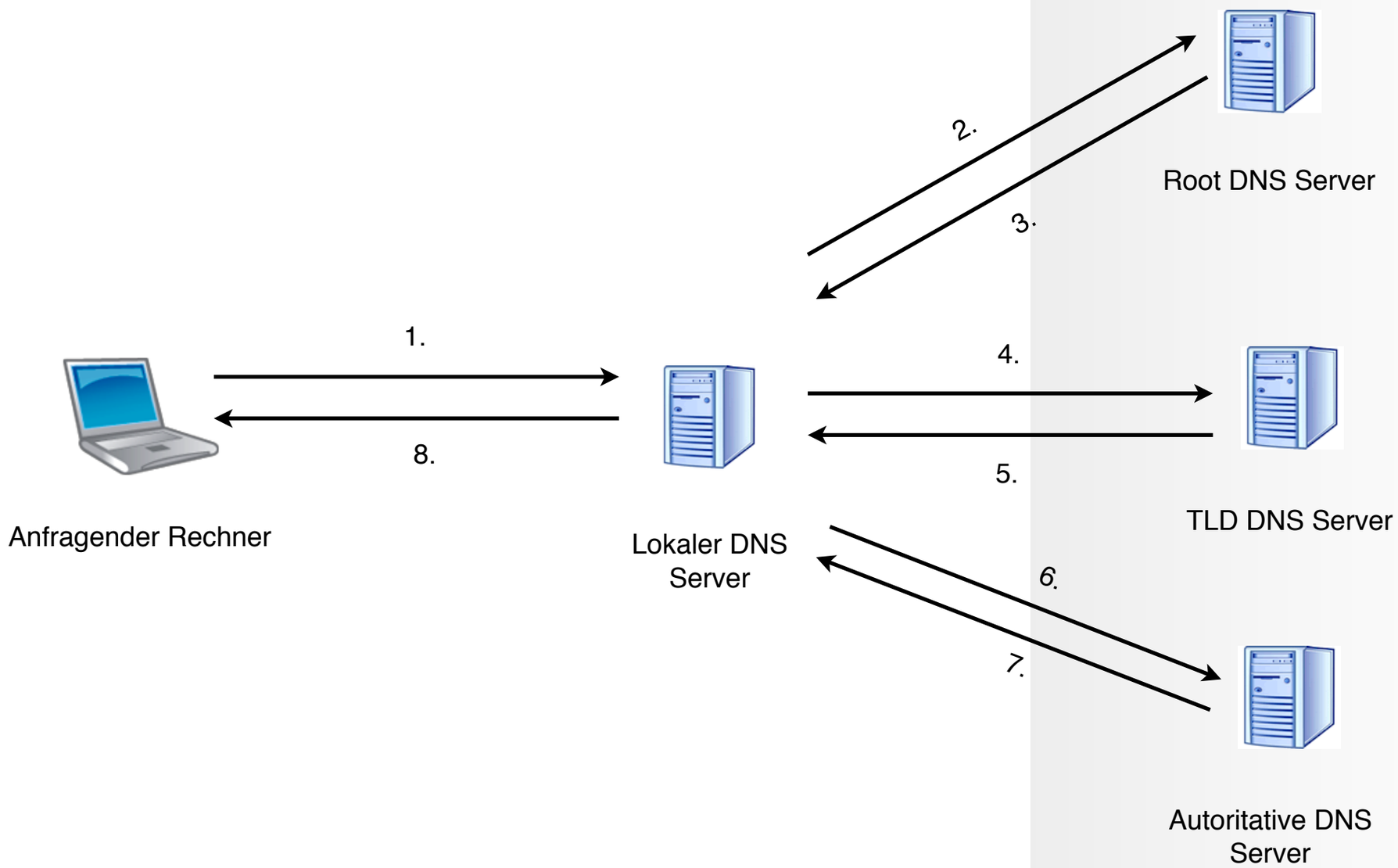
Domain Server Klassen

- Nicht autoritative Nameserver
  - Um die autoritativen Nameserver zu entlasten (und auch um Traffic nach außen zu reduzieren) betreiben Internetprovider eigene Nameserver, die allerdings nicht autoritativ sind (non-authoritative) und DNS Anfragen stellvertretend durchführen. Die so ermittelten Informationen werden dann zwischengespeichert und bei einer erneuten Anfrage direkt zurückgegeben -> Caching.
- Lokaler Nameserver
  - Kein Teil der DNS Hierarchie, allerdings findet sich ein Lokaler DNS in der meistens Konfigurationen. So verwendet fast jeder ISP eine eigenen Lokalen DNS (“verteilt” via DHCP) welche als Proxy fungiert.

DNS Zone - Domain

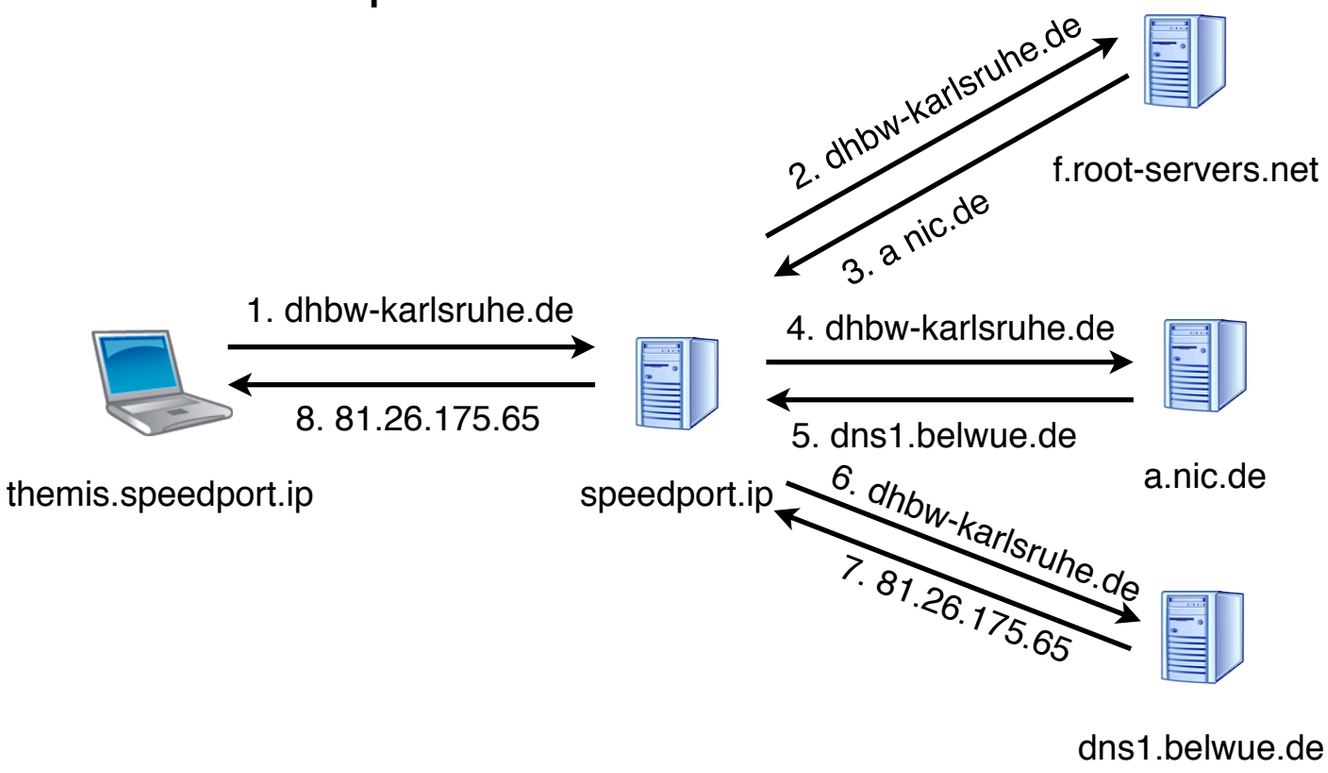
- Abgrenzung DNS Zone zu einer Domain
  - Eine Domain umfasst den gesamten untergeordneten DNS-Namensraum. Der Begriff Domain wird auch verwendet, wenn man sich auf den Inhalt oder die Eigentumsrechte bezieht.
  - Eine Domäne kann in mehrere Zonen aufgeteilt werden, indem man die Zuständigkeit für Subdomains delegiert. Von einer Zone spricht man auch, wenn man die physische Realisierung meint – also auf welchem Server und in welcher Zonendatei die DNS-Einträge liegen.

Domain Server - Abfrage Hierarchie



Domain Server

- Zur Auflösung einer Domain aus einer anderen Domain sind mehrere DNS Abfragen nötig.
- Beispiel:



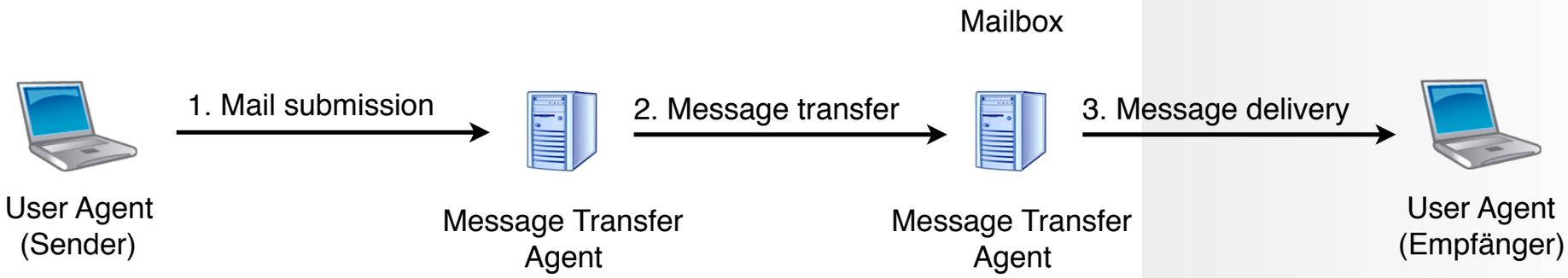

```
Themis:~ geye$ dig +trace dhbw-karlsruhe.de

; <<>> DiG 9.4.3-P3 <<>> +trace dhbw-karlsruhe.de
;; global options:  printcmd
.                       85978   IN      NS      i.root-servers.net.
.                       85978   IN      NS      l.root-servers.net.
.                       85978   IN      NS      d.root-servers.net.
.                       85978   IN      NS      m.root-servers.net.
.                       85978   IN      NS      g.root-servers.net.
.                       85978   IN      NS      b.root-servers.net.
.                       85978   IN      NS      a.root-servers.net.
.                       85978   IN      NS      k.root-servers.net.
.                       85978   IN      NS      e.root-servers.net.
.                       85978   IN      NS      f.root-servers.net.
.                       85978   IN      NS      j.root-servers.net.
.                       85978   IN      NS      c.root-servers.net.
.                       85978   IN      NS      h.root-servers.net.
;; Received 449 bytes from 192.168.2.1#53(192.168.2.1) in 23 ms

de.                     172800  IN      NS      n.de.net.
de.                     172800  IN      NS      z.nic.de.
de.                     172800  IN      NS      s.de.net.
de.                     172800  IN      NS      l.de.net.
de.                     172800  IN      NS      a.nic.de.
de.                     172800  IN      NS      f.nic.de.
;; Received 349 bytes from 192.5.5.241#53(f.root-servers.net) in 184 ms

dhbw-karlsruhe.de.      86400   IN      NS      dns3.belwue.de.
dhbw-karlsruhe.de.      86400   IN      NS      dns1.belwue.de.
;; Received 140 bytes from 194.0.0.53#53(a.nic.de) in 31 ms

dhbw-karlsruhe.de.      3600    IN      A       81.26.175.65
dhbw-karlsruhe.de.      3600    IN      NS      dns3.belwue.de.
dhbw-karlsruhe.de.      3600    IN      NS      dns1.belwue.de.
;; Received 156 bytes from 129.143.2.10#53(dns1.belwue.de) in 57 ms
```

Simple Mail Transfer Protocol (SMTP)

- SMTP ermöglicht den Austausch von Nachrichten über dezentrale Mailserver.
- SMTP wurde erstmals mit RFC 821 spezifiziert. Eine aktuelle überarbeite Fassung wurde mit RFC 5321 verabschiedet.
- Wie der Transport von Nachrichten wurde auch das Format standardisiert.
  - Der Standard für das EMail Format ist in RFC 822 spezifiziert. Dieser RFC wurde mit RCF 5322 erweitert und unterstützt damit zusätzlich multimediale Inhalte (MIME).

Simple Mail Transfer Protocol (SMTP)

SMTP - Architektur



- Das EMail System besteht aus zwei Komponenten:
  - **User Agents**: Client Applikationen die auf den Rechner des Anwenders ausgeführt werden und dem Benutzer das Lesen und das Senden von EMail erlauben.
  - **Message Transfer Agents**: Server Applikationen die im Hintergrund den Transport der EMails übernehmen.
- Der EMail Transport erfolgt in drei Stufen.
  1. Mail submission: Einstellen der EMail an den zuständigen Message Transfer Agent durch den Sender.
  2. Message Transfer: Transport der EMail an den Message Transfer Agent des Empfängers.
  3. Message delivery: Zustellen der EMail an den User Agent des Empfängers.

SMTP - Komponenten

- User Agent:
  - Bieten ein grafische oder Text basierende Bedienungsoberflächen und ermöglichen es einem Benutzer mit dem EMail-System zu interagieren.
  - Folgende Services werden von den User Agents implementiert:
    - Erstellen und Beantworten von EMails
    - Anzeige und Organisation von empfangenen EMails (Ablage, Suche, Löschung)
    - Einstellen neuer EMails in das EMail-System

SMTP - Komponenten

- Message Transfer Agent:
  - Sind werden auf Servern (Mail Servern) ausgeführt und stellen das eigentliche EMail System dar.
  - Message Transfer Agents implementieren folgende Funktionen:
    - Mailing Listen: An jeder Adresse auf der Liste werden Kopien der original EMail versendet.
    - Carbon Copies (CC - Kopie), Blind Carbon Copies (BC - Blindkopie)
    - Priorisierte EMails
    - Alternative Empfänger
    - Senden und Lesen von EMails im Auftrag

SMTP - Transport und Adressierung

- Der Message Transfer Agent des Empfängers wird über DNS, mit Hilfe des MX Records, ermittelt.
  - Beispiel: bob@uwa.edu.au, damit wird der MX Record der Domain uwa.edu.au ermittelt und die EMail an die eingetragene IP Adresse gesendet.

Simple Mail Transfer Protocol (SMTP)

SMTP - Protokoll (Auszug)

- HELO (EHLO) <my.domain>
  - Anmelden am Mail Server
- auth login
  - Authentifizierter Anmeldevorgang initiieren
- mail from: <username>
  - Absender definieren
- RCPT TO: <username>
  - Empfänger definieren

SMTP - Protokoll (Auszug)

- DATA
  - Signalisiert den Begin der Daten
- .
  - Signalisiert das Ende der Daten
- QUIT
  - Beenden der Verbindung

SMTP - Beispiel

```
dig MX markusgoetzl.de

; <<>> DiG 9.4.3-P3 <<>> MX markusgoetzl.de
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54347
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;markusgoetzl.de.         IN     MX

;; ANSWER SECTION:
markusgoetzl.de.    62202  IN     MX     100 mail.markusgoetzl.de.

;; ADDITIONAL SECTION:
mail.markusgoetzl.de.     62202  IN     A      62.75.187.244

;; Query time: 11 msec
;; SERVER: 192.168.2.1#53(192.168.2.1)
;; WHEN: Fri Dec 14 00:12:15 2012
;; MSG SIZE  rcvd: 70
```

SMTP - Beispiel

```
telnet mail.markusgoetzl.de 25
Trying 62.75.187.244...
Connected to mail.markusgoetzl.de.
Escape character is '^]'.
220 euve22849.vserver.de ESMTP Postfix (Debian/GNU)
ehlo mail.markusgoetzl.de
250-euve22849.vserver.de
250-PIPELINING
250-SIZE 10240000
250-ETRN
250-STARTTLS
250-AUTH CRAM-MD5 DIGEST-MD5 LOGIN PLAIN
250-AUTH=CRAM-MD5 DIGEST-MD5 LOGIN PLAIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
auth login
334 VXNlcm5hbWU6
YmVudXR6ZXJuYW1l
334 UGFzc3dvcmQ6
cGFzc3dvcnQ=
235 Authentication successful
```

SMTP - Beispiel

```
mail from: benutzername
250 Ok
rcpt to: empfaenger@externe.domain
250 Ok
data
354 End data with .
mein mailtext zeile 1
mein mailtext zeile 2
.
250 Ok: queued as AE600400140C
quit
```

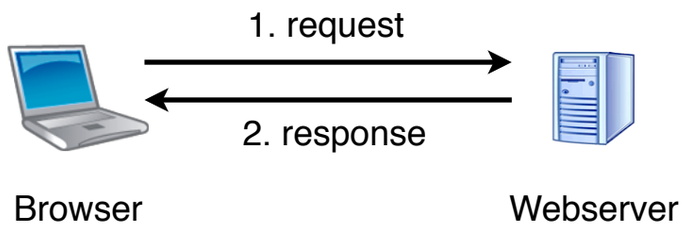
Hypertext Transfer Protocol (HTTP)

- Ab 1989 entwickelten Roy Fielding, Tim Berners-Lee und andere am CERN das Hypertext Transfer Protocol, zusammen mit den Konzepten URL und HTML, womit die Grundlagen des World Wide Web geschaffen wurden.

- HTTP ist ein Protokoll zur Übertragung von Daten über ein Netzwerk. Es wird hauptsächlich eingesetzt, um Webseiten aus dem World Wide Web (WWW) in einen Webbrowser zu laden.

- HTTP ist ein zustandsloses Protokoll. Ein zuverlässiges Mitführen von Sitzungsdaten kann z.B. über eine Session-ID implementiert werden.

HTTP Aufbau



- Jede Nachricht besteht aus zwei Teilen, dem **Message Header** (Nachrichtenkopf) oder HTTP-Header und dem **Message Body** (Nachrichtenkörper).
- Der Message Header enthält Informationen über den Nachrichtenkörper wie etwa verwendete Kodierungen oder den Inhaltstyp, damit dieser vom Empfänger korrekt interpretiert werden kann.
- Der Message Body enthält schließlich die Nutzdaten.

HTTP Header - Request

```
GET /info.html HTTP/1.1
User-Agent: curl/7.16.4 (i386-apple-darwin9.0) libcurl/
7.16.4 OpenSSL/0.9.7l zlib/1.2.3

Host: www.google.de
```

HTTP Header - Response

```
HTTP/1.1 200 OK

Server: Apache/1.3.29 (Unix) PHP/4.3.4

Content-Length: (Größe von info.html in Byte)

Connection: close

Content-Type: text/html


(Inhalt von info.html)
```

URI/URL

- URI (Uniform Resource Identifier)
  - Ein Uniform Resource Identifier ist ein Identifikator und besteht aus einer Zeichenfolge, die zur Identifizierung einer abstrakten oder physischen Ressource dient.
  - URIs werden zur Bezeichnung von Ressourcen (wie Webseiten, sonstigen Dateien, Aufruf von Webservices, aber auch z. B. E-Mail-Empfängern) im Internet und dort vor allem im WWW eingesetzt.
  - Aufbau: `URI = scheme ":" hier-part [ "?" query ] [ "#" fragment ]`

HTTP Request

- GET
  - Mit GET wird eine Ressource (z. B. eine Datei) unter Angabe eines URI vom Server angefordert.
  - Als Argumente in dem URI können also auch Inhalte zum Server übertragen werden. Aufbau: URI?Key1=Value1&Key2=Value2....
  - Die Länge des URIs ist je nach eingesetztem Server begrenzt und sollte aus Gründen der Abwärtskompatibilität nicht länger als 255 Bytes sein.
- POST
  - schickt unbegrenzte, je nach physikalischer Ausstattung des eingesetzten Servers, Mengen an Daten zur weiteren Verarbeitung zum Client.

HTTP Request

- HEAD
  - weist den Server an, die gleichen HTTP-Header wie bei GET, nicht jedoch den eigentlichen Dokumentinhalt (Body) zu senden. So kann zum Beispiel schnell die Gültigkeit einer Datei im Browsercache geprüft werden.
- PUT
  - dient dazu eine Ressource (z. B. eine Datei) unter Angabe des Ziel-URIs auf einen Webserver hochzuladen.

HTTP Request

- DELETE
  - löscht die angegebene Ressource auf dem Server. Heute ist das, ebenso wie PUT, kaum implementiert bzw. in der Standardkonfiguration von Webservern abgeschaltet, beides erlangt jedoch mit RESTful Web Services und der HTTP-Erweiterung WebDAV neue Bedeutung.
- TRACE
  - liefert die Anfrage so zurück, wie der Server sie empfangen hat. So kann überprüft werden, ob und wie die Anfrage auf dem Weg zum Server verändert worden ist – sinnvoll für das Debugging von Verbindungen.

HTTP Request

- OPTIONS
  - liefert eine Liste der vom Server unterstützen Methoden und Features.
- CONNECT
  - wird von Proxyservern implementiert, die in der Lage sind, SSL-Tunnel zur Verfügung zu stellen.

HTTP Status Codes

- 1xx - Informationen
  - Die Bearbeitung der Anfrage dauert trotz der Rückmeldung noch an. Eine solche Zwischenantwort ist manchmal notwendig, da viele Clients nach einer bestimmten Zeitspanne (Timeout) annehmen, dass ein Fehler bei der Übertragung oder Verarbeitung der Anfrage aufgetreten ist.
- 2xx - Erfolgreiche Operation
  - Die Anfrage wurde bearbeitet und die Antwort wird an den Anfragesteller zurückgesendet.
- 3xx - Umleitung
  - Um eine erfolgreiche Bearbeitung der Anfrage sicherzustellen, sind weitere Schritte seitens des Clients erforderlich. Das ist zum Beispiel der Fall, wenn eine Webseite vom Betreiber umgestaltet wurde, so dass sich eine gewünschte Datei nun an einem anderen Platz befindet. Mit der Antwort des Servers erfährt der Client im Location-Header, wo sich die Datei jetzt befindet.

HTTP Status Codes

- 4xx - Client-Fehler
  - Bei der Bearbeitung der Anfrage ist ein Fehler aufgetreten, der im Verantwortungsbereich des Clients liegt. Ein 404 tritt beispielsweise ein, wenn ein Dokument angefragt wurde, das auf dem Server nicht existiert. Ein 403 weist den Client darauf hin, dass es ihm nicht erlaubt ist, das jeweilige Dokument abzurufen. Es kann sich zum Beispiel um ein vertrauliches oder nur per HTTPS zugängliches Dokument handeln.
- 5xx - Server-Fehler
  - Es ist ein Fehler aufgetreten, dessen Ursache beim Server liegt. Zum Beispiel bedeutet 501, dass der Server nicht über die erforderlichen Funktionen (d. h. zum Beispiel Programme oder andere Dateien) verfügt, um die Anfrage zu bearbeiten.