Anwendungsprotokolle

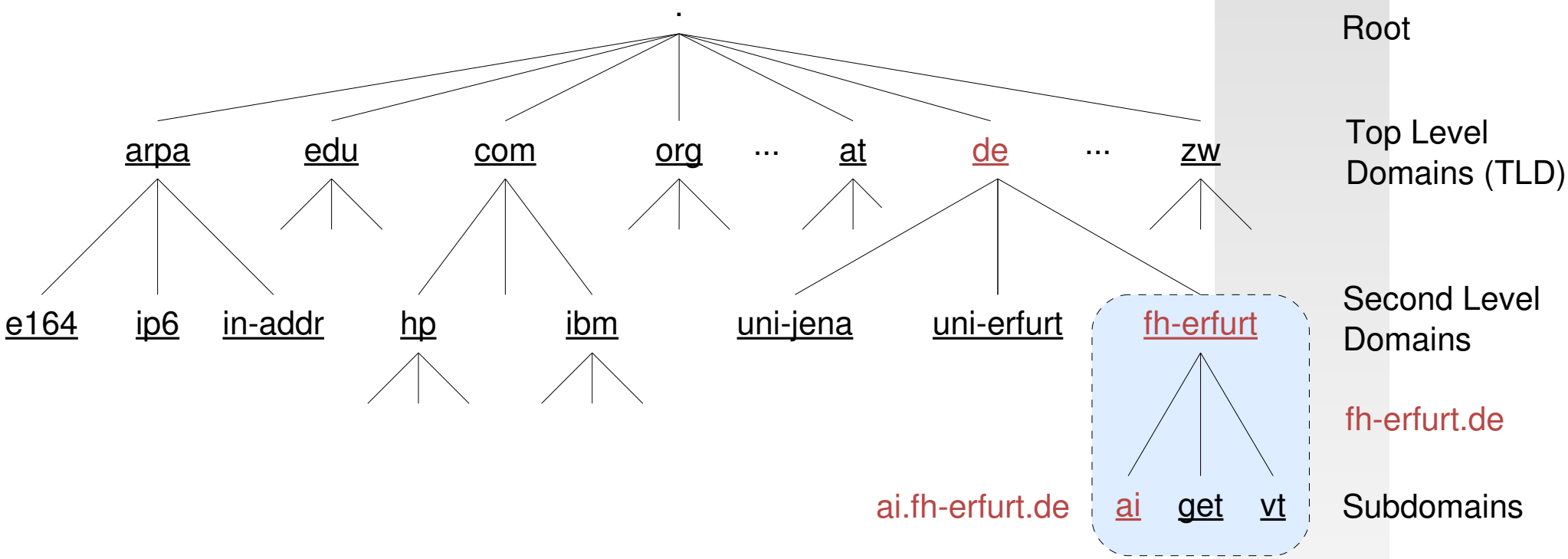
- Domain Name System (DNS)
- Dynamic Host Configuration Protocol (DHCP)
- Telecommunications Network (TELNET)
- File Transfer Protocol (FTP)
- Simple Mail Transfer Protocol (SMTP)
- HyperText Transfer Protocol (HTTP)

Domain Name System (DNS)

- Domain Name System
  - Infrastrukturdienst auf Anwendungsebene
  - Spezifikation: RFC 1034-1035
  - Erweiterungen: DYNDNS, DNSSEC
- Ziele
  - Nutzung symbolischer Namen anstelle von IP-Adressen
    Beispiel: `www.ai.fh-erfurt.de` ⇒ `194.94.204.23`
  - Nutzung von Aliases
    Beispiel: `www.ai.fh-erfurt.de` ⇒ `webserver.fh-erfurt.de`
- Aufgaben
  - verteilte Speicherung symbolischer Namen (verteilte Datenbank)
  - Auflösung symbolischer Namen und Aliases zu IP-Adressen (rekursiv oder iterativ)
  - Inverse Auflösung IP-Adressen zu symbolischen Namen (Reverse Lookup)

DNS: Struktur

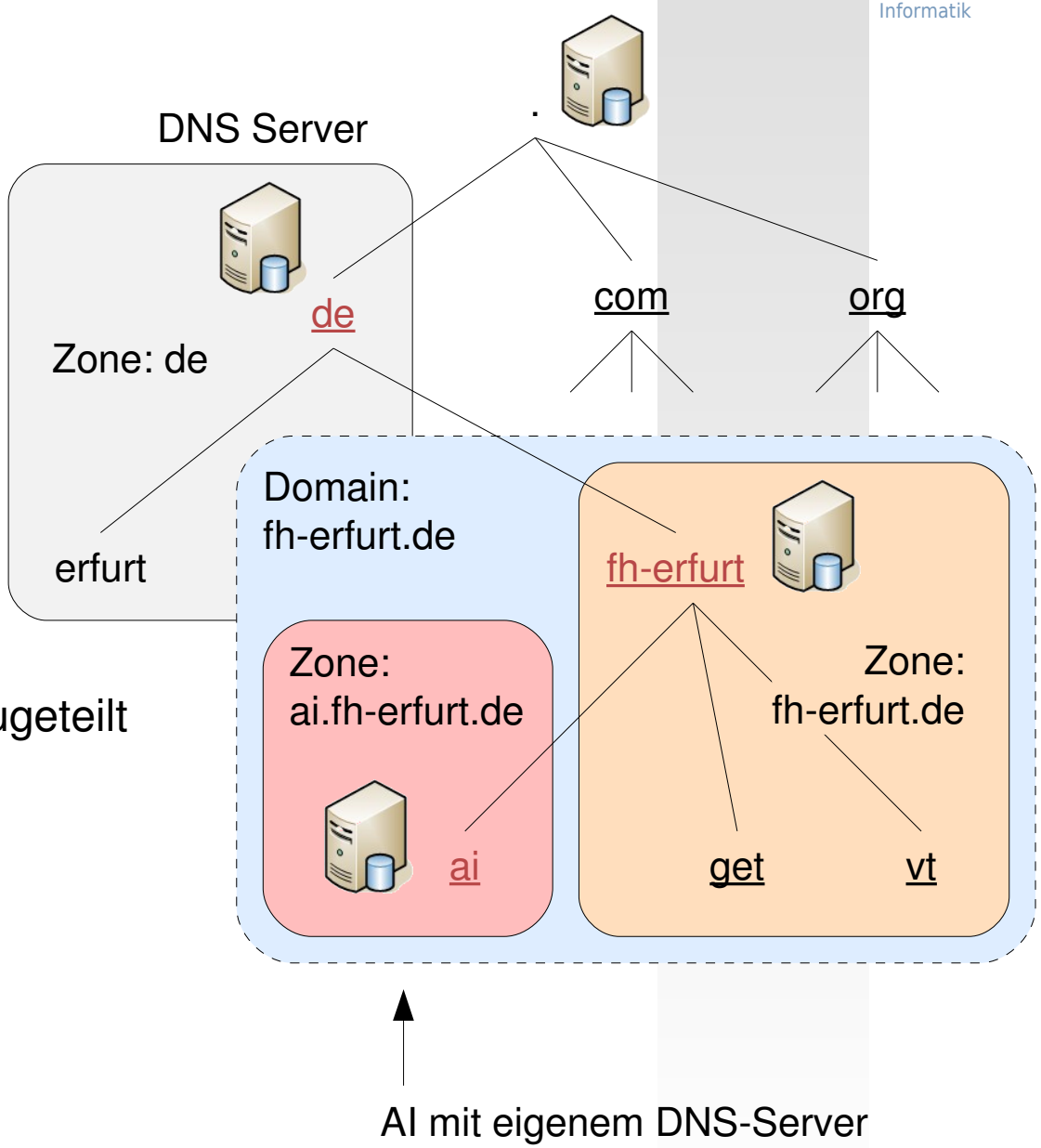
- hierarchische Baumstruktur



- Full Qualified Domain Name (FQDN)
  - vollständiger Name eines Host im DNS bestehend aus
    - Hostname
    - vollständigem Domainnamen (Folge der Domainnamen der Knoten im Baum)
  - relativ zur Wurzel (d.h. ohne abschließenden Punkt), z.B.: `www.ai.fh-erfurt.de`

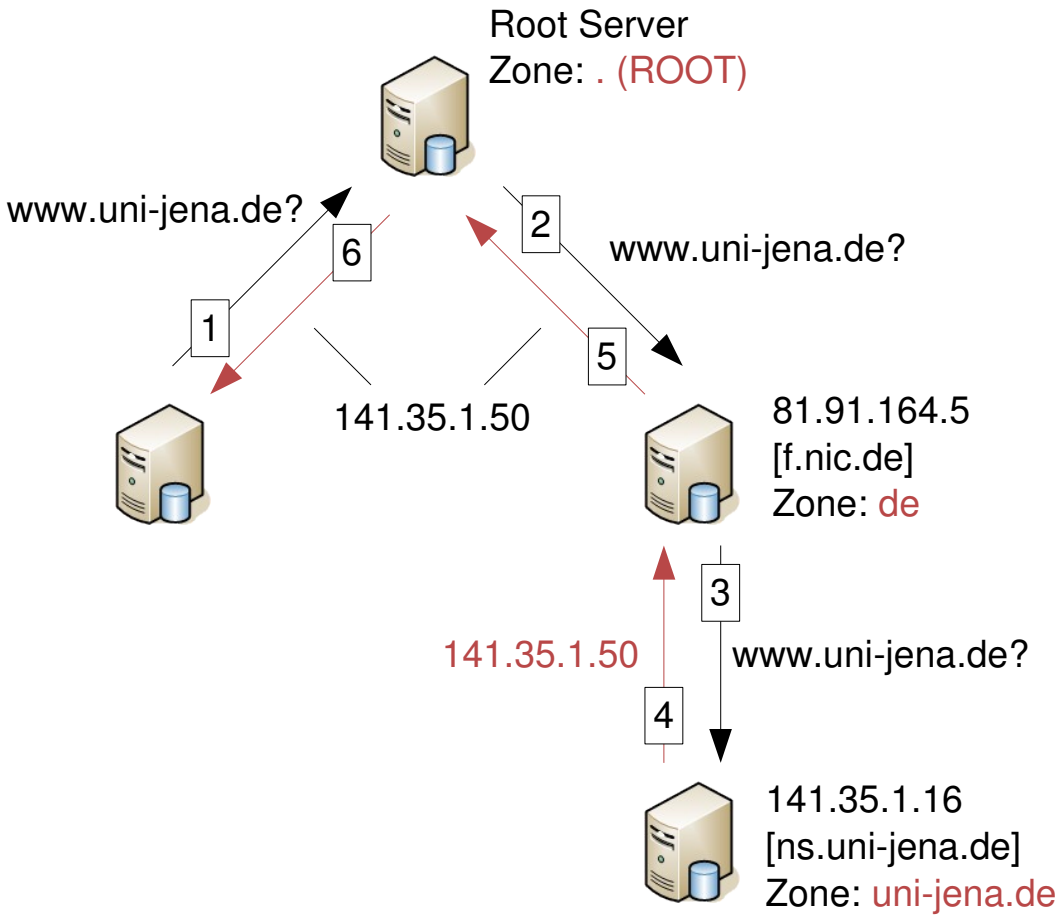
DNS: Domainen und Zonen

- Domain
  - vollständiger Teil des Baumes
  - Subdomainen möglich
- Zone
  - Namensraum mit eigener Datenbank
  - besitzt eigenen DNS-Server (authoritative nameserver)
  - ist dem DNS-Server authoritativ zugeteilt
- DNS-Server
  - verwaltet Resource Records (RR)
  - zugehörige Hosts
  - untergeordnete (delegierte) Zonen

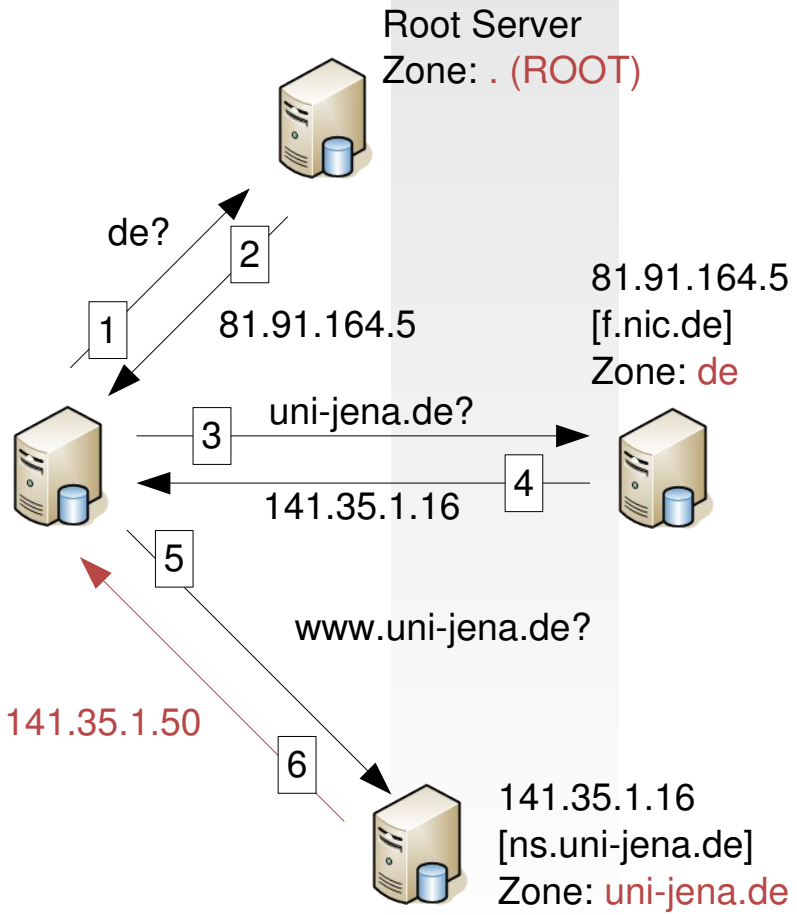


DNS: Namensauflösung

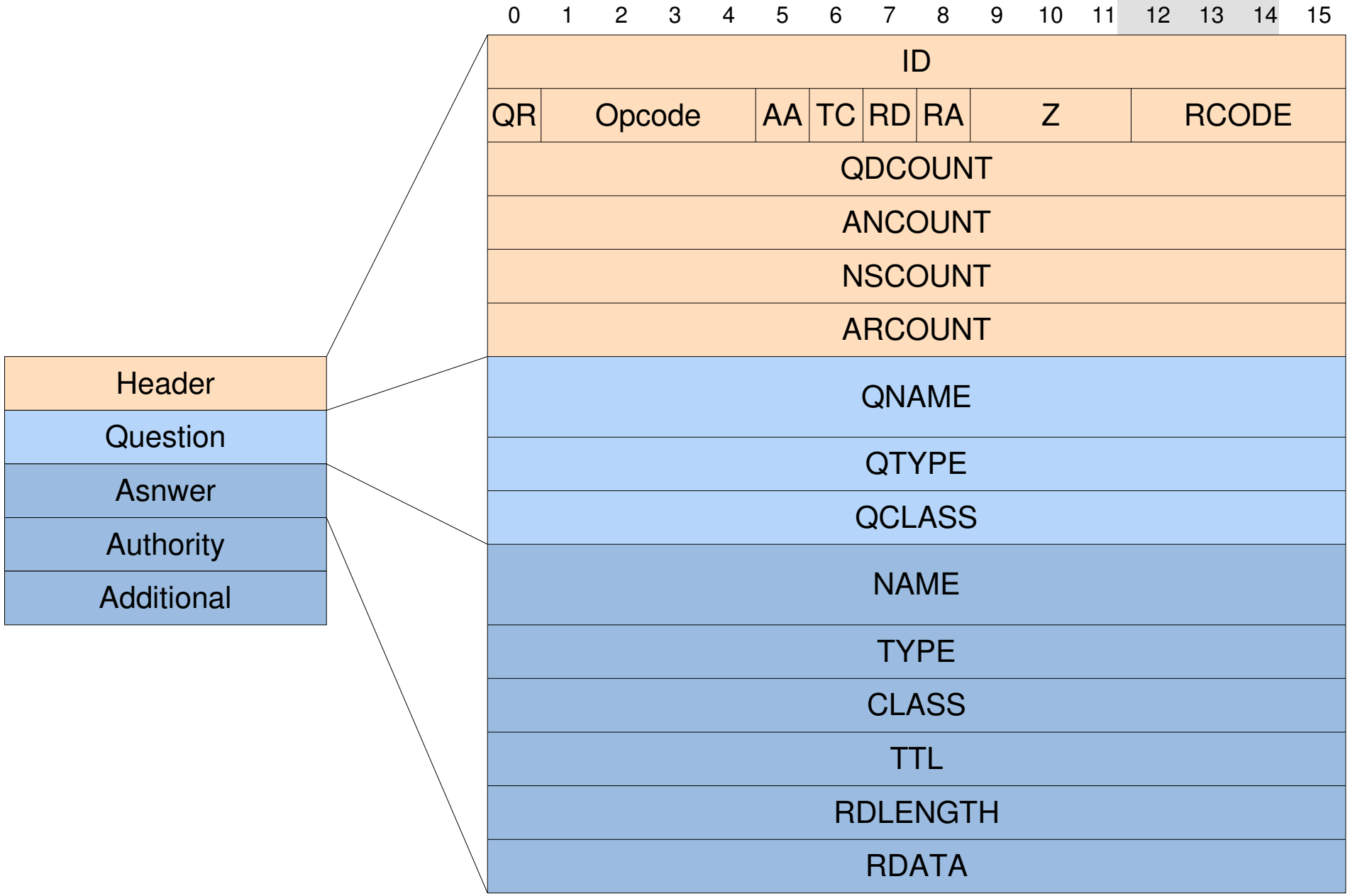
- rekursiv
  - rekursive Abfrage der DNS-Server
  - liefert IP-Adresse des Hosts



- iterativ
  - löst Zonen iterativ auf
  - liefert jeweils DNS-Server der Zonen

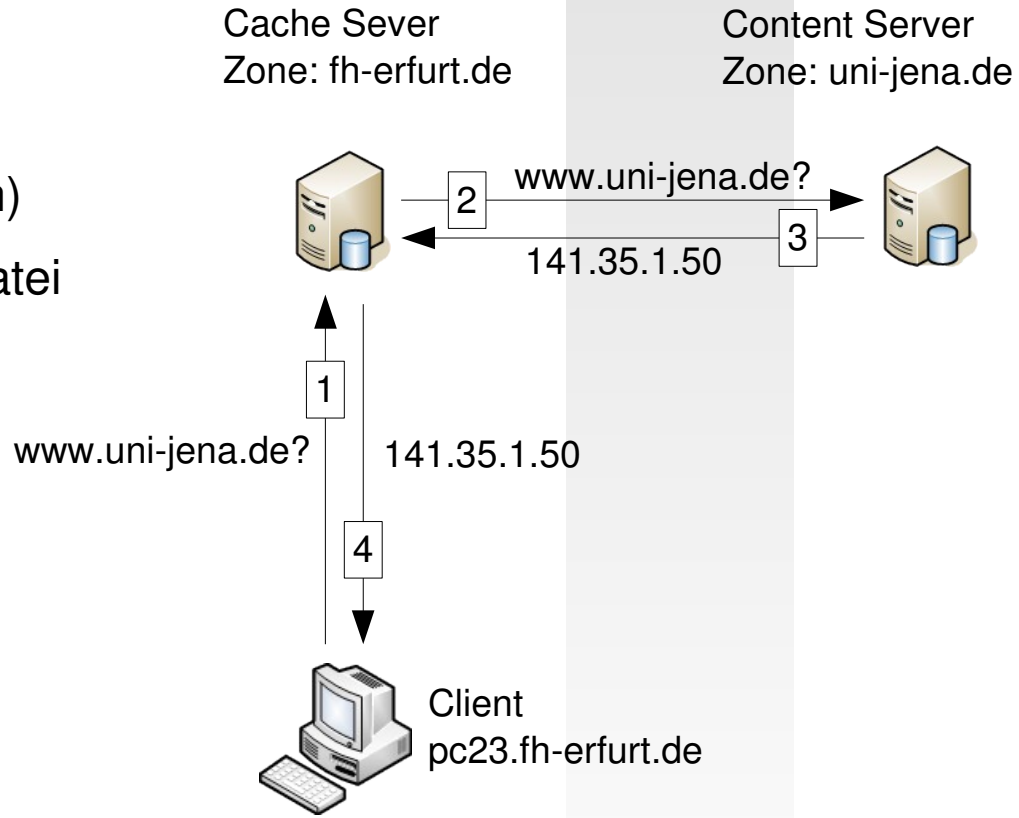


DNS: Format


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
ID
QR
Opcode
AA
TC
RD
RA
Z
RCODE
QDCOUNT
ANCOUNT
NSCOUNT
ARCOUNT
Header
Question
Asnwer
Authority
Additional
QNAME
QTYPE
QCLASS
NAME
TYPE
CLASS
TTL
RDLENGTH
RDATA

DNS: Namensauflösung - Komponenten

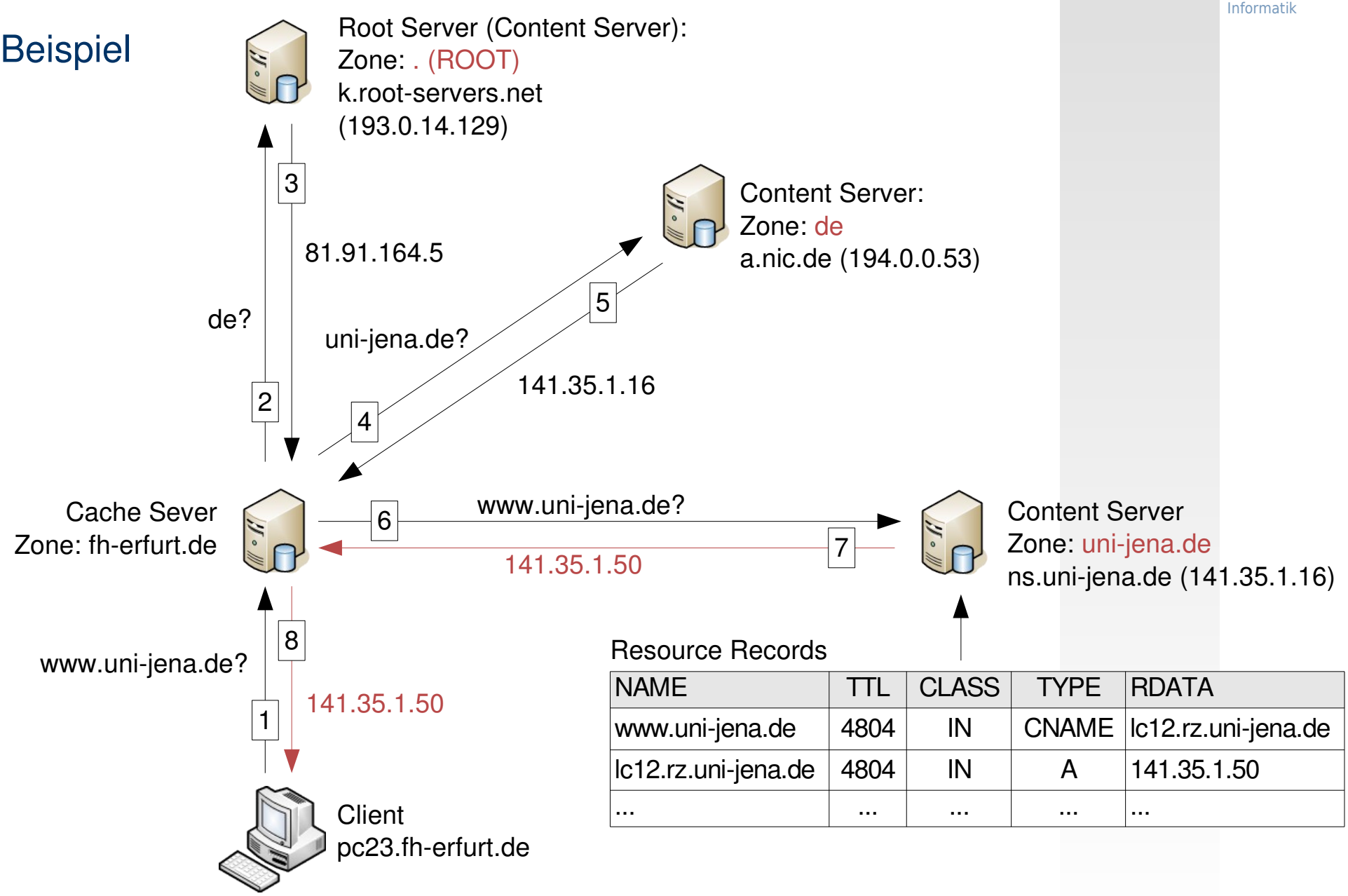
- Client (Stub Resolver)
  - Bestandteil Betriebssystem/Anwendung: dient nur der DNS-Abfrage
  - konfiguriert mit zuständigem DNS-Server
- Content Server
  - Verwaltung einer Zone (Hosts, Subzonen)
  - liefert Informationen aus lokalen Zonendatei (authoritative=amtlich)
- Cache Server (Full Resolver)
  - speichert Namensauflösungen zwischen
  - falls
    - vorhanden, Auslieferung aus dem Cache (non authoritative)
    - nicht vorhanden, iterative Auflösung über zuständigen Content Server und Auslieferung (authoritative = amtlich)



häufig: DNS-Server = Content + Cache Server

DNS: Namensauflösung - Ablauf

- Beispiel



Resource Records

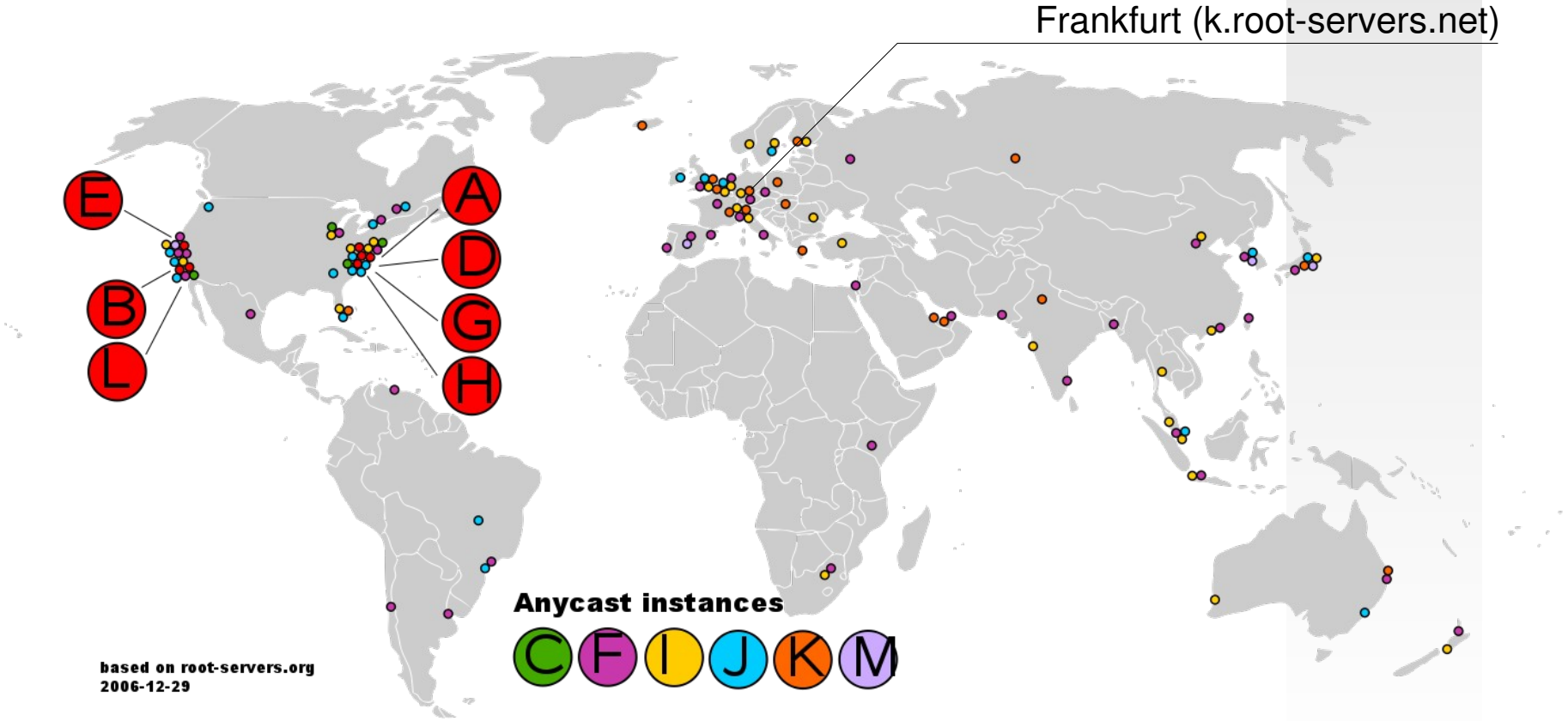
NAME	TTL	CLASS	TYPE	RDATA
www.uni-jena.de	4804	IN	CNAME	lc12.rz.uni-jena.de
lc12.rz.uni-jena.de	4804	IN	A	141.35.1.50
...	...	...	...	...

DNS: Root-Server

- Root-Server

[http://root-servers.org]

  - Anker für Namensauflösungen
  - verwalten Top Level Domainen und Referenzen auf zuständigen Name Server
  - 13 Cluster per Anycast (DNS-Namen: [a-m].root-servers.net)



DNS: Resource Records (Zonendatei)

- Format: <NAME> [<TTL>] <CLASS = IN> <TYPE> <RDATA>

Domain/Host Name
Objektname

Time To Live [s]
Gültigkeitsdauer

Resource Data
- IP Address
- Full Qualified Domain Name

TYPE	Wert	Bedeutung
A	1	IPv4 Address
AAAA	28	IPv6 Address
CNAME	5	Canonical Name (Alias)
MX	15	Mail eXchange
NS	2	Authoritative Name Server
SOA	6	Start Of zone Authority

- Beispiele

```
www.fh-erfurt.de.     3600   IN   CNAME    thalia.fh-erfurt.de.
thalia.fh-erfurt.de.         IN   A        193.174.232.73
```

DNS: Zonendatei - Beispiel fh-erfurt.de

Default TTL für Einträge ohne Anhabe TTL

Primary DNS-Server für Zone

Email-Adresse Administrator

```
$TTL 172800 ; Default TTL (2 days)
@                                   IN    SOA     mars.fh-erfurt.de. root.mars.fh-erfurt.de. (
                                                  2006120555     ; Serial yyyymmddnn
                                                  14400          ; Refresh in seconds (4 h)
                                                  7200           ; Retry in seconds (2 h)
                                                  604800         ; Expire in seconds (1 week)
                                                  172800 )       ; Negative Cache TTL (2 days)

                                    IN    NS      amor.fh-erfurt.de.
                                    IN    NS      zeus.fh-erfurt.de.

fh-erfurt.de.                       IN    MX      10 mars.fh-erfurt.de.
fh-erfurt.de.                       IN    MX      20 zeus.fh-erfurt.de.

thalia.fh-erfurt.de.                IN    A       193.174.232.73
clio.fh-erfurt.de.                  IN    A       193.174.232.65
mars.fh-erfurt.de.                  IN    A       193.174.232.66
zeus.fh-erfurt.de.                  IN    A       193.174.233.21
amor.fh-erfurt.de.                  IN    A       193.175.1.225
...
www.fh-erfurt.de.                   IN    CNAME   thalia.fh-erfurt.de.
mail.fh-erfurt.de.                  IN    CNAME   clio.fh-erfurt.de.
...
ai.fh-erfurt.de.          3600      IN    NS      dns1.ai.fh-erfurt.de.
ai.fh-erfurt.de.          3600      IN    NS      dns2.ai.fh-erfurt.de.
dns1.ai.fh-erfurt.de.     3600      IN    A       194.94.204.26
dns2.ai.fh-erfurt.de.     3600      IN    A       194.94.204.27
```

Secondary DNS-Server

Mail Server

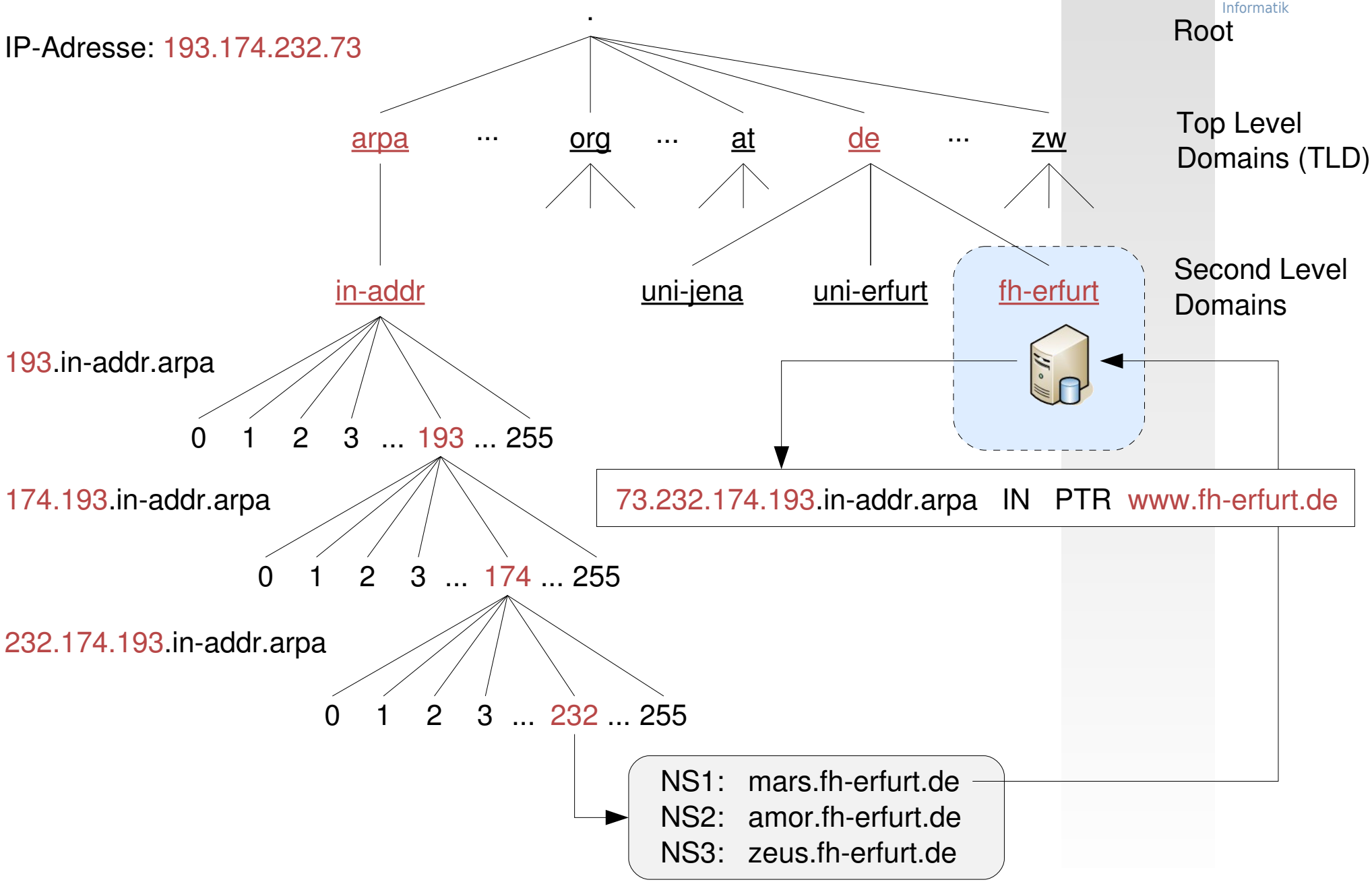
Host Adresses

Host Aliases

Einbindung Sub-Zone

DNS: Reverse Lookup (Inverse Auflösung)

FACHHOCHSCHULE ERFURT UNIVERSITY OF APPLIED SCIENCES
Angewandte Informatik

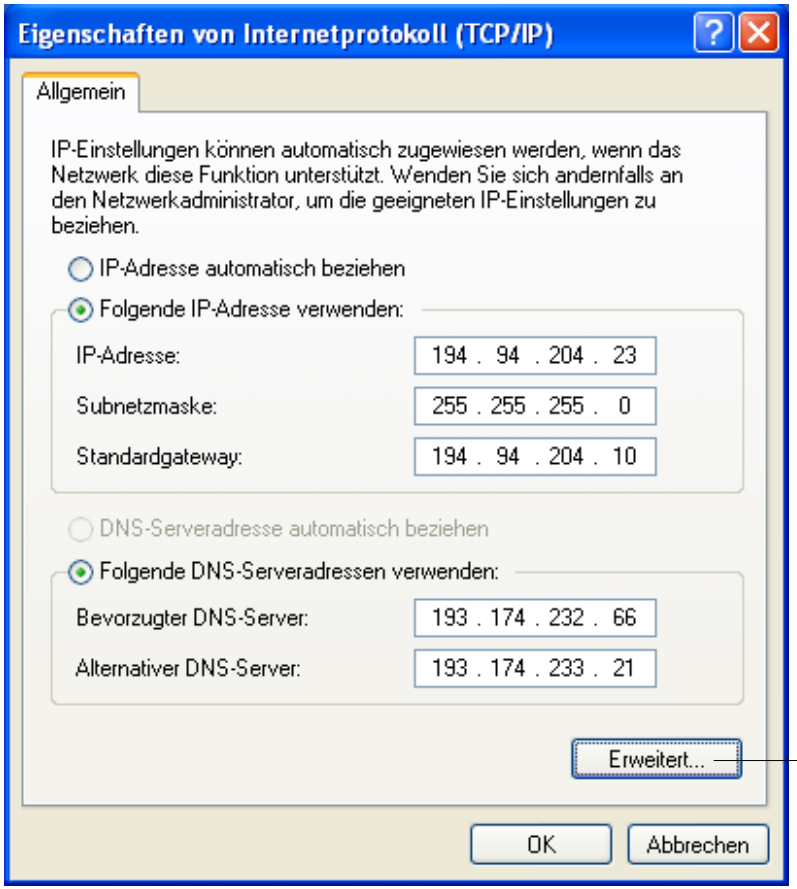


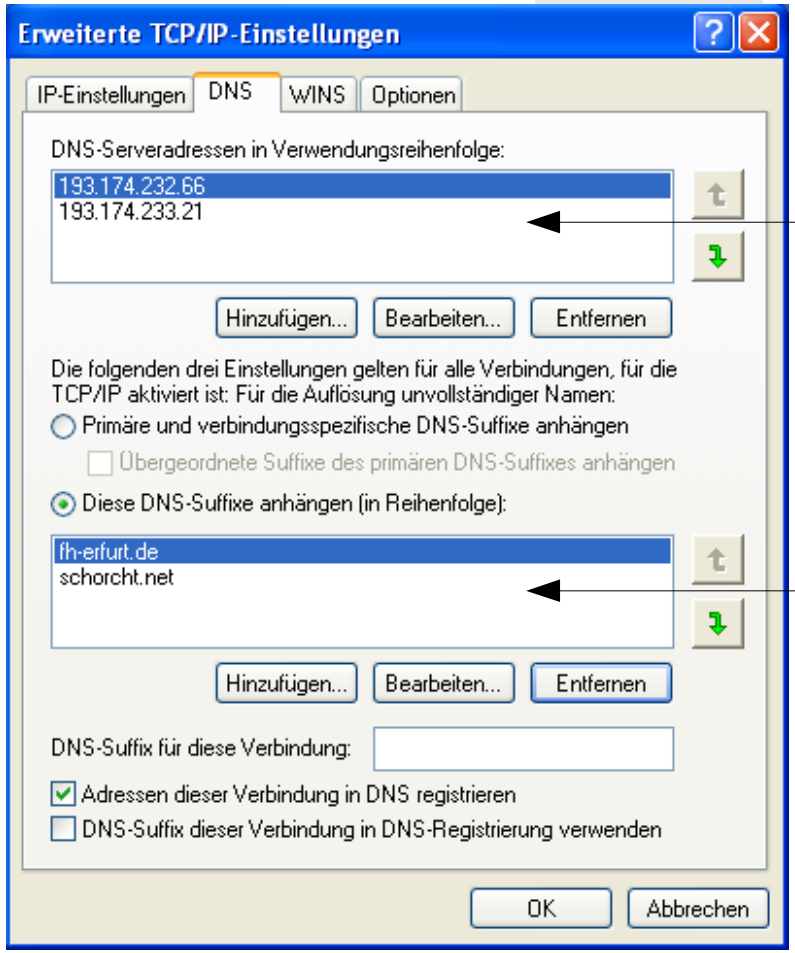
DNS: Konfiguration

- Linux `/etc/resolv.conf`

```
domain fh-erfurt.de
searchlist fh-erfurt.de schorcht.net
nameserver 193.174.232.66
nameserver 193.174.233.21
```

- Windows





Dynamic Host Configuration Protocol (DHCP)

- Aufgaben
  - Zuweisung privater IP-Adressen in LANs
  - Zuweisung öffentlicher IP-Adressen für WAN-Zugangsknoten (Routern)

- Inhalte einer Zuweisung
  - IP-Adresse des Knoten
  - Lease-Dauer

  } Pflichtangaben RFC 2131

  - IP-Parameter: Subnetmaske, Default Gateway, IP-Forwarding, IP-Source-Routing, MTU per Interface, TTL, statische Routen, ...
  - TCP-Parameter: Keep Alive , TTL, ...
  - DNS-Parameter: Host Name, Domain Name, DNS Server, ...

  } optionale Angaben RFC 2132

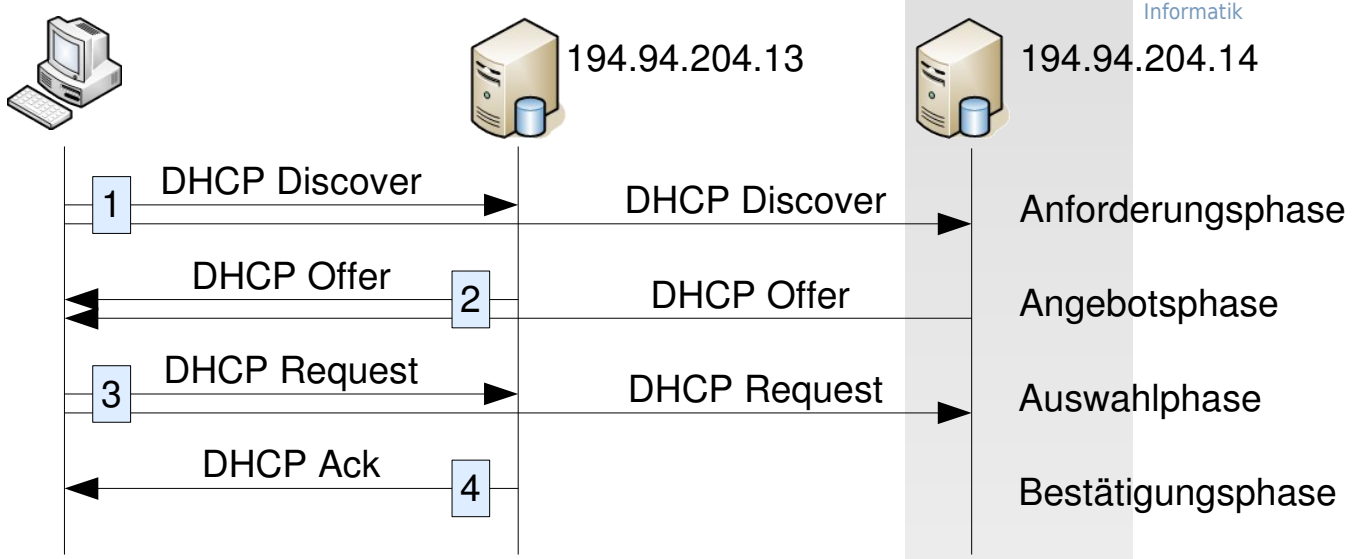
- Zuweisungsvarianten
  - automatisch: Zuweisung über gesamte Laufzeit eines Knoten fest zugewiesen
  - dynamisch: Zuweisung für definierte Lease-Time, danach erneute Zuweisung nötig
  - manuell: feste Zuweisung anhand der MAC-Adresse des Knoten

DHCP: Message Format (RFC 2131)

Source/Destination Port
67 DHCP Server
68 DHCP Client

- Messages (Option Message Type)
  - DHCP Discover: Anforderung
  - DHCP Offer: Angebot
  - DHCP Request: Auswahl
  - DHCP Ack: Bestätigung

DHCP: Ablauf (RFC 2131)



1

DHCP Discover
Src IP = 0.0.0.0
Dst IP = 255.255.255.255
chaddr = 00:0B:5D:9F:2E:94
Message Type = DHCP Discover

2

DHCP Offer
Src IP = 194.94.204.13
Dst IP = 255.255.255.255
yiaddr = 194.94.204.57
siaddr = 194.94.204.13
chaddr = 00:0B:5D:9F:2E:94
sname = ai-dhcp
Message Type = DHCP Offer
Server Identifier = 194.94.204.13
Subnet Mask = 255.255.255.0
Router = 194.94.204.10
Domain Name = ai.fh-erfurt.de
Domain Name Server = ...
IP Address Lease Time = 7200

Options

3

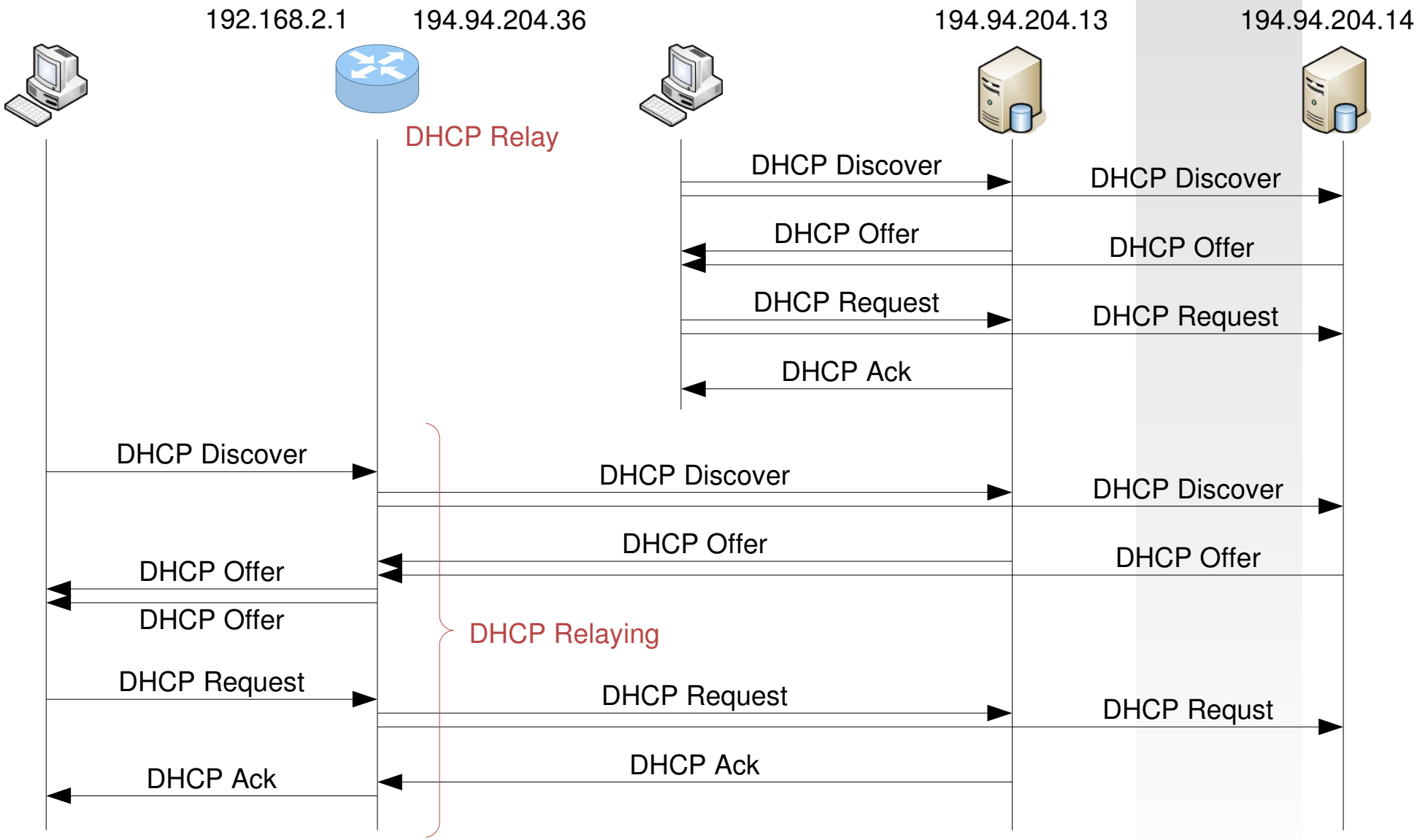
DHCP Request
Src IP = 0.0.0.0
Dst IP = 255.255.255.255
chaddr = 00:0B:5D:9F:2E:94
Message Type = DHCP Request
Requested IP = 194.94.204.57
Server Identifier = 194.94.204.13

4

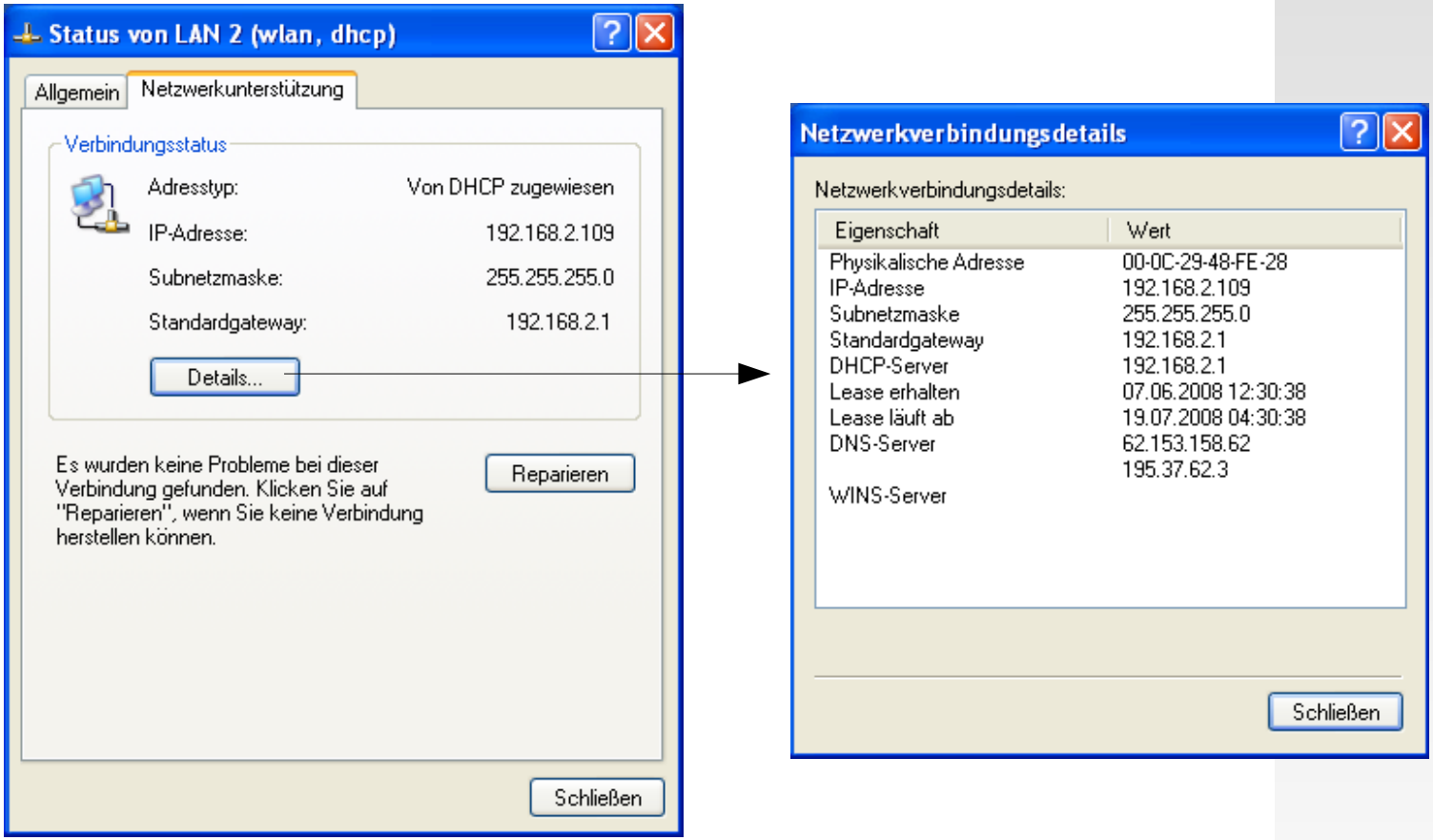
DHCP Ack
Src IP = 194.94.204.13
Dst IP = 255.255.255.255
yiaddr = 194.94.204.57
siaddr = 194.94.204.13
chaddr = 00:0B:5D:9F:2E:94
sname = ai-dhcp
Message Type = DHCP Ack
Server Identifier = 194.94.204.13
Subnet Mask = 255.255.255.0
Router = 194.94.204.10
Domain Name = ai.fh-erfurt.de
Domain Name Server = ...
IP Address Lease Time = 7200

DHCP-Relaying

FACHHOCHSCHULE ERFURT UNIVERSITY OF APPLIED SCIENCES
Angewandte Informatik



DHCP: Beispiel



DHCP: Server-Konfiguration

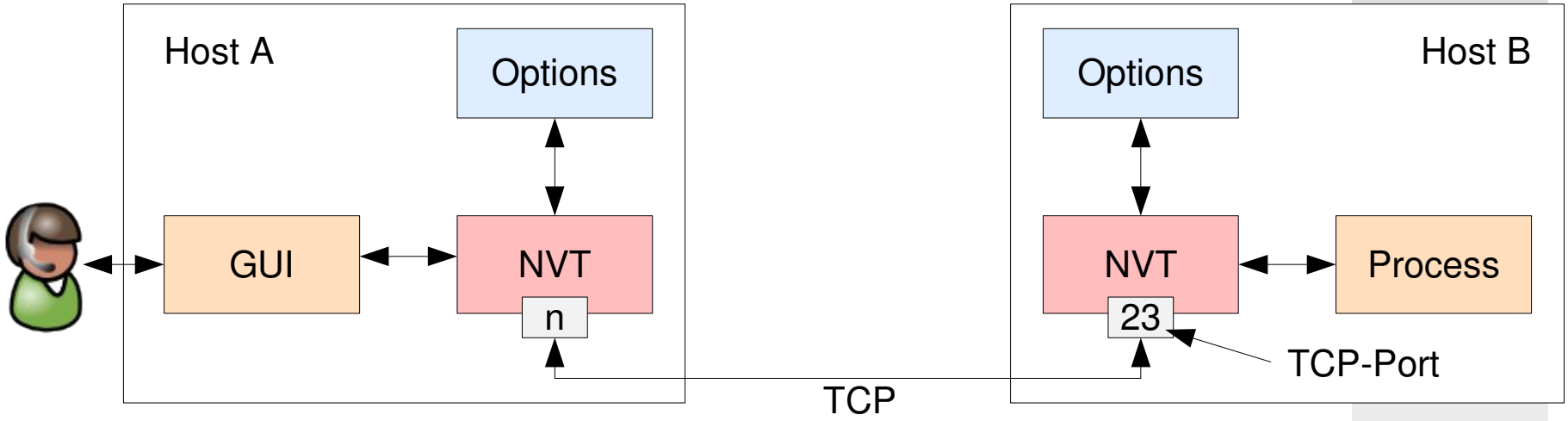
- /etc/dhcpd.conf

```
option domain-name "ai.fh-erfurt.de";
option domain-name-servers 193.174.232.66, 193.174.233.21;
option routers 194.94.204.10;
option ntp-servers ntp1.ptb.de;
option lpr-servers 194.94.204.46;
ddns-update-style none;
default-lease-time 864000;
always-broadcast true;

subnet 194.194.204.0 netmask 255.255.255.0 {
  authoritative ;
  range 194.194.204.40 194.194.204.199;
  default-lease-time 864000;
  max-lease-time 1728000;
}

host tanami {
  hardware ethernet 00:03:0d:4c:47:5a;
  fixed-address 194.94.204.45;
}
```

Telecommunications Network Protocol (TELNET)

- Ziel
  - bidirektionaler byte-orientierter Kommunikationskanal zu einem entfernten Host
  - historisch: Kommunikation mit einem entfernten Terminal
- Spezifikation: RFC 845
- Eigenschaften
  - setzt zuverlässiges Transportprotokoll voraus: TCP
  - Signalisierung erfolgt innerhalb des Zeichenstromes (Inband Signaling)
  - Datenübertragung unverschlüsselt einschließlich Nutzername und Password
  - kurze Reaktionszeiten durch Übertragung jedes Zeichens in einer PDU
- Standardanwendung
  - Remote Login bzw. Access
  - verbindet Terminals (Client) mit entferntem Prozess (Server), z.B. Command Shell
  - Terminal zumeist als Terminalemulation auf einem Arbeitsplatzrechner

TELNET: Prinzip



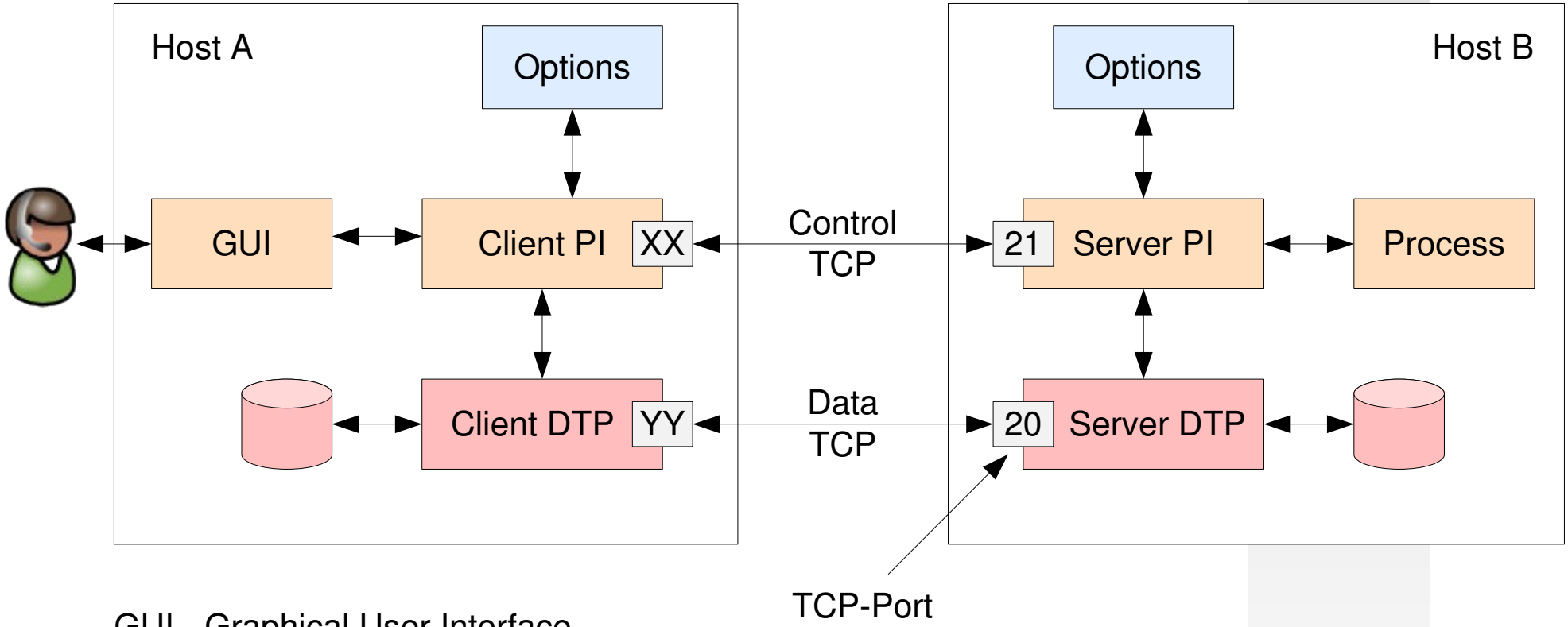
GUI Graphical User Interface
NVT Network Virtual Terminal

- Network Virtual Terminal
  - verbirgt Heterogenität
  - definiert Standardschnittstelle zwischen den Systemen
  - definiert Kodierungsregeln der jeweiligen Platform in plattformunabhängiges Format
- Probleme
  - verschiedene Terminalformate: Zeichenkodierung, Farbe, Auflösung
  - verschiedene Steuersequenzen: VT100, ..., VT240, ..., VT320, ...
  - Kommunikationspartner müssen sich auf gemeinsame Parameter einigen

File Transfer Protocol (FTP)

- Ziel
  - Dateien auf entfernten Host ablegen
  - Dateien von entfernten Host abrufen
- Spezifikation: RFC 959
- Eigenschaften
  - textbasiertes Protokoll (Meldungen im Klartext)
  - setzt zuverlässiges Transportprotokoll voraus: TCP
  - verwendet 2 Verbindungen (Outband Signaling)
    - Port 20: Transfer der Nutzdaten
    - Port 21: Kontrollinformationen bzw. Steuerung
  - FTP-Server ist statusbehaftet
  - Datenübertragung unverschlüsselt einschließlich Nutzername und Password
  - bei Bedarf Konvertierung der Zeichen währen der Übertragung

FTP: Prinzip



GUI Graphical User Interface
PI Protocol Interface
DTP Data Transfer Process

FTP: Kommandos

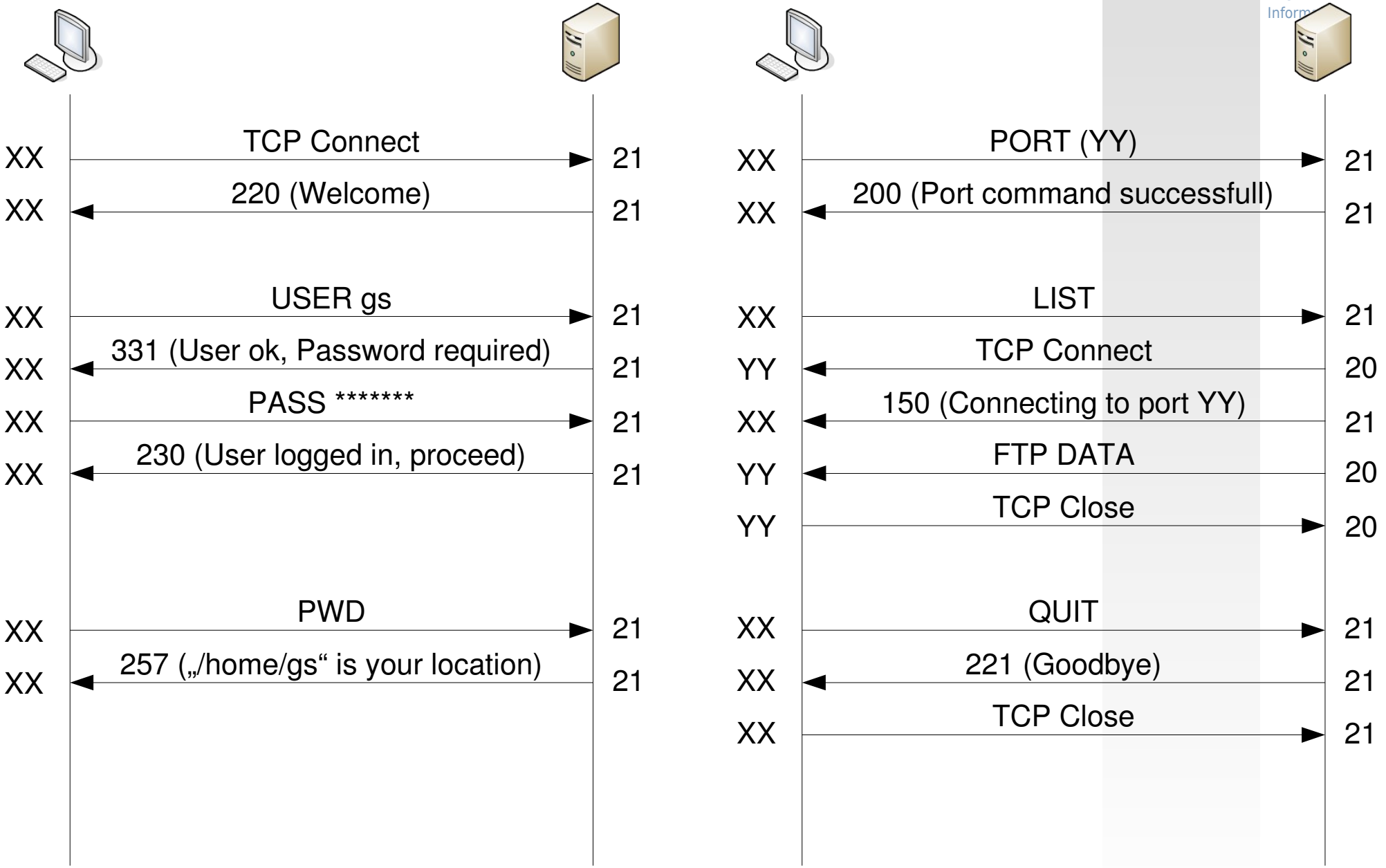
Kommandos

Command	Bemerkung
`CONNECT / CLOSE`	Verbindung mit FTP-Server auf- bzw. abbauen
`USER / PASS`	Eingabe von Benutername und Passwort
`GET / MGET / RETR`	Datei(en) von FTP-Server abrufen
`PUT / MPUT / STORE`	Datei(en) von FTP-Server ablegen
`DEL`	Datei auf FTP-Server löschen
`DIR / LIST / LS`	Zeige Inhalt des aktuellen Verzeichnisses auf dem FTP-Server
`ASCII / BINARY`	Setze bzw. verhindere Zeichenkonvertierug
`PORT`	Teilt Server den Port des Client für Datenübertragung mit

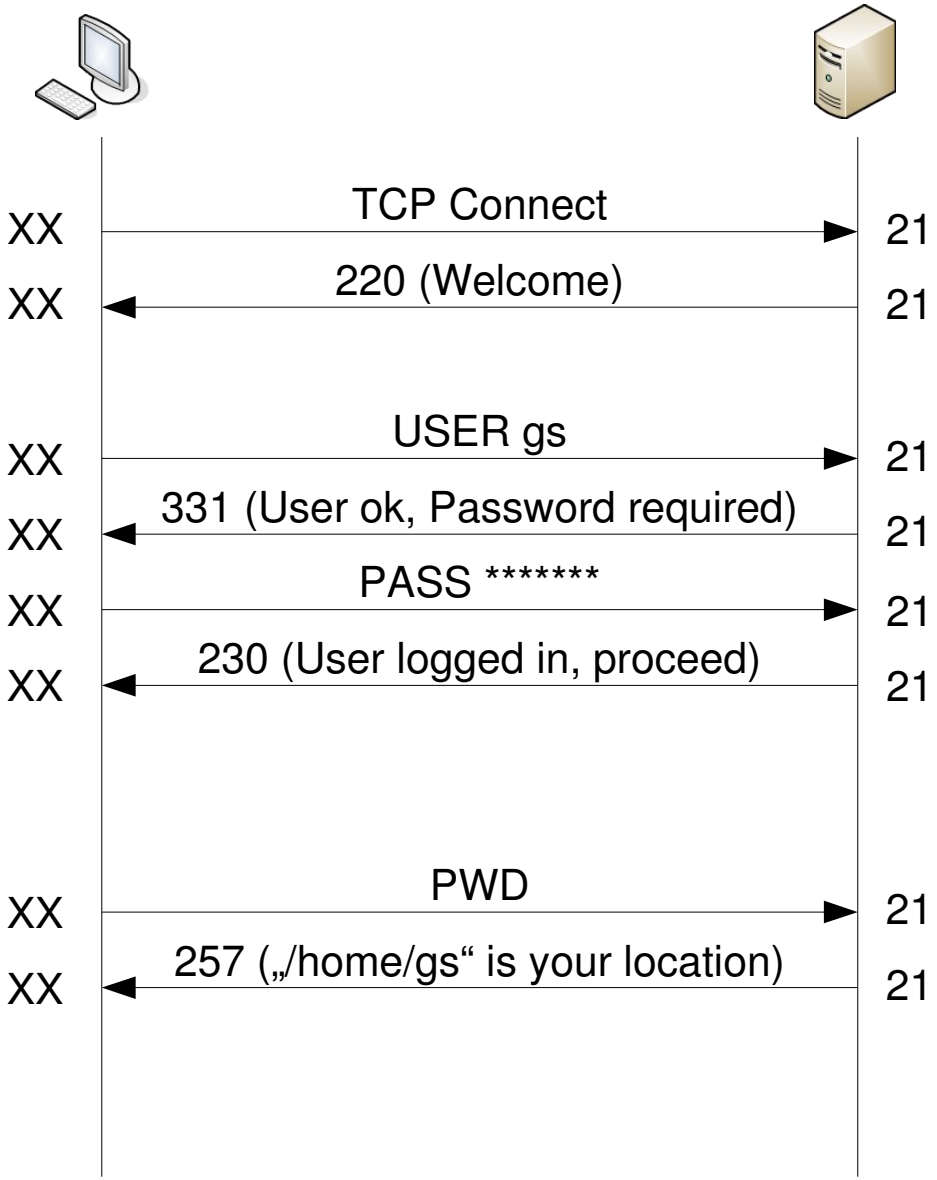
Antwort des Servers mit Status Codes

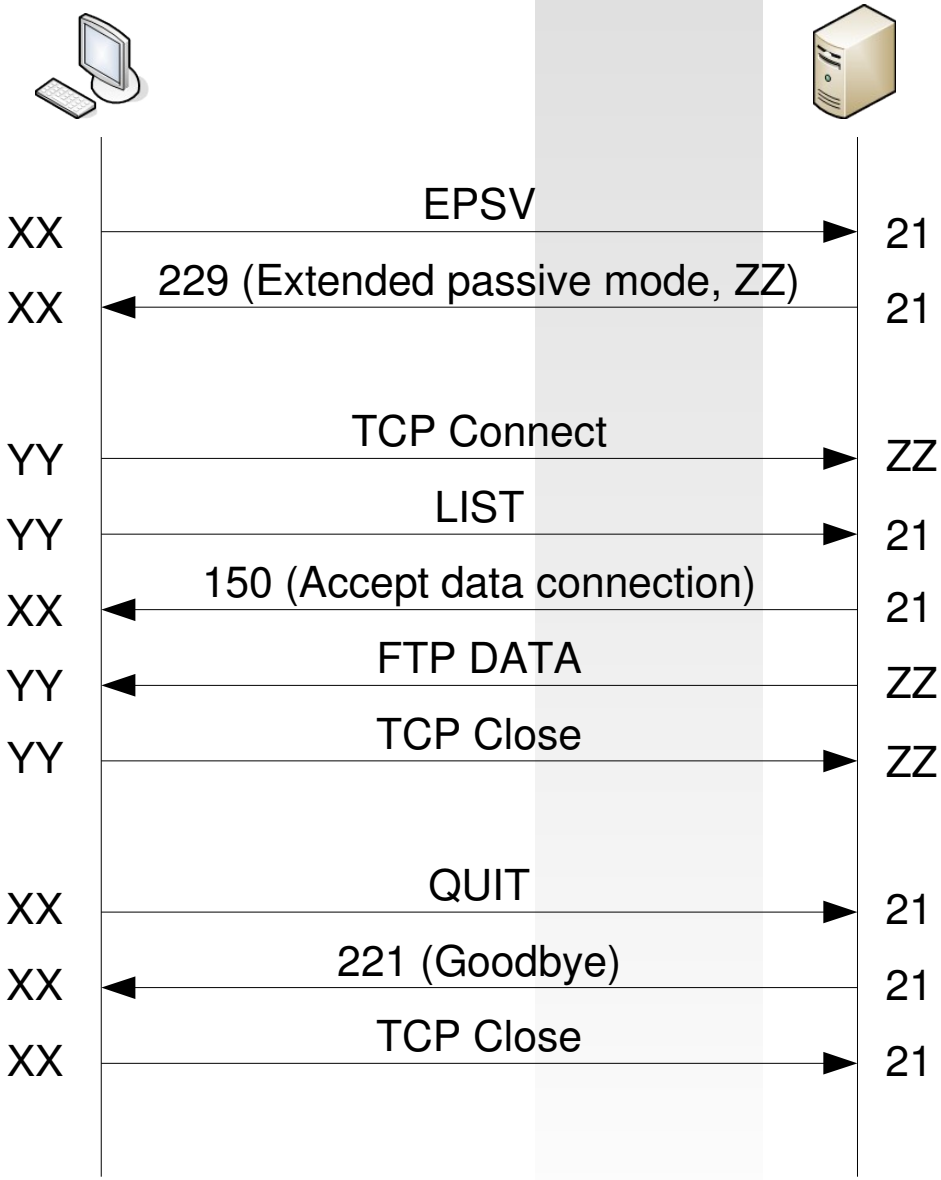
- 200 Service ready for new user
- 221 Service closing control connection. Logged out if appropriate.
- 226 Closing data connnection. Requested file action successful
- 230 User logged in, proceed
- ...

FTP: Ablauf (Active Mode)



FTP: Ablauf (Passive Mode)

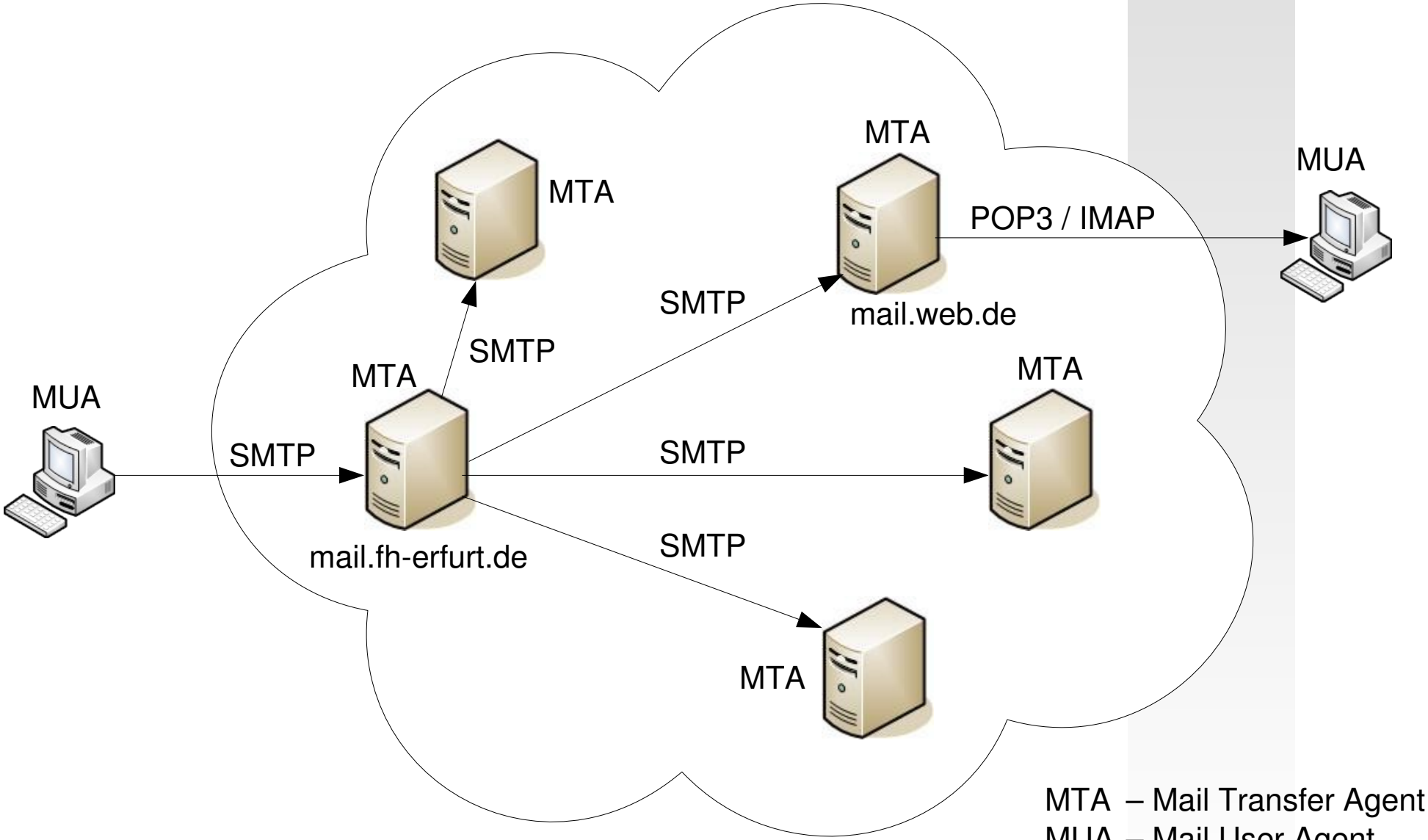




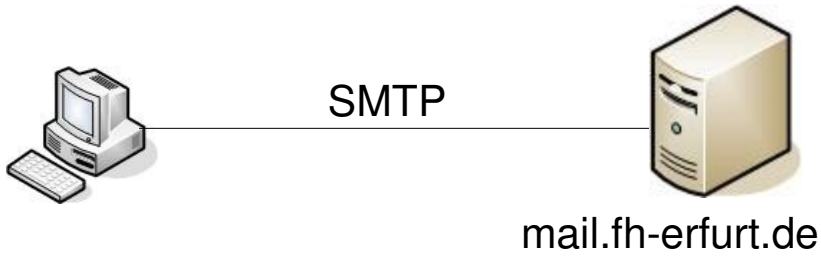
Simple Mail Transfer Protocol (SMTP)

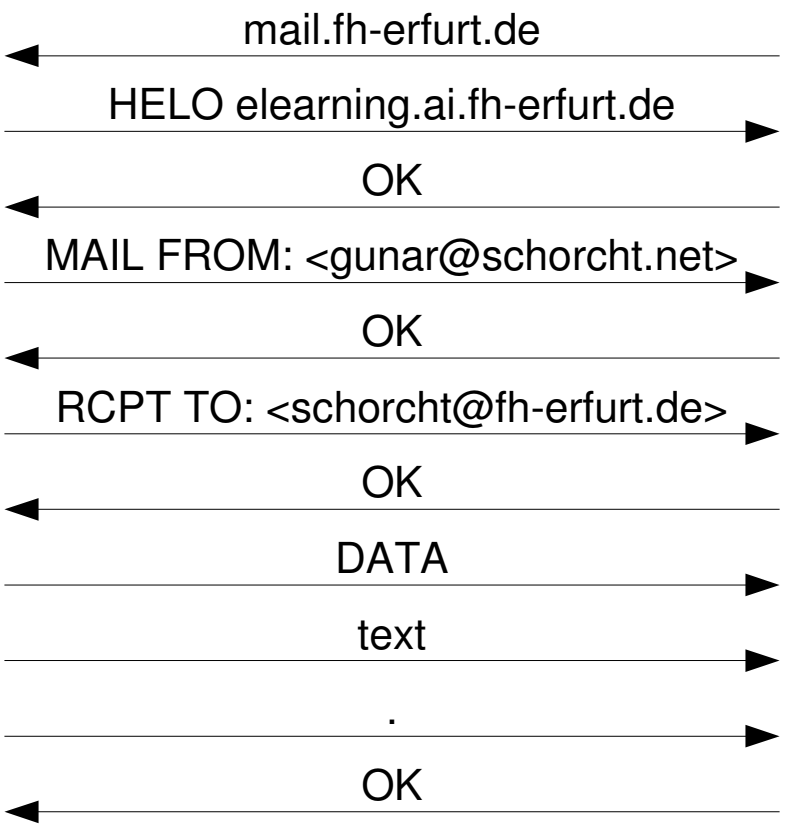
- Ziel
  - Transfer von E-Mails zwischen Mail Servern
  - Transfer von E-Mails von Mail Reader (Mail Client) an Mail Server
- Spezifikation: RFC 2821
- Eigenschaften
  - textbasiertes Protokoll (Meldungen im Klartext)
  - setzt zuverlässiges Transportprotokoll (TCP) voraus
  - Kontroll- und Nutzinformationen über einen Kanal (Port 25)
  - Mail Server tauschen gegenseitig E-Mails aus - jeder Mailserver ist Client und Server
  - sämtliche E-Mails in 7-Bit US-ASCII codiert - Problem Binaries
  - Ablauf: Verbindungsauf- und -abbau (Handshake, Greeting), Datenübertragung
  - Syntax der Kommandos ähnlich zu FTP
    - definierte Kommandos mit Parametern vom SMTP-Client
    - SMTP-Server antwortet durch definierte Status Codes

SMTP: Struktur



SMTP: Ablauf






```
telnet mail.fh-erfurt.de 25
Trying 193.174.232.65...
Connected to mail.fh-erfurt.de.
Escape character is '^]'.

220 mail.fh-erfurt.de -- Server ESMTP
HELO elearning.ai.fh-erfurt.de
250 mail.fh-erfurt.de OK, [194.94.204.25]
MAIL FROM: <gunar@schorcht.net>
250 2.5.0 Address Ok.
RCPT TO: <schorcht@fh-erfurt.de>
250 2.1.5 gunar@schorcht.net OK.
DATA
354 Enter mail, end with a single ".".
Subject: Beispiel SMTP eMail
Hallo,
das ist eine eMail.
Ciao
.
250 2.5.0 Ok.
```

SMTP: Beispiel empfangene Email


```
Return-path: <gunar@schorcht.net>
Received: from elearning.ai.fh-erfurt.de ([80.88.20.33])
   by clio.fh-erfurt.de (Sun Java(tm) System Messaging
   Server 6.3-5.02 (built Oct 12 2007; 32bit)) with SMTP id
   <0KJ500KEM40IJMC0@clio.fh-erfurt.de> for
   schorcht@fh-erfurt.de; Tue, 05 May 2009 00:05:41 +0200 (CEST)
Original-recipient: rfc822;schorcht@fh-erfurt.de
From: gunar@schorcht.net
Date-warning: Date header was inserted by clio.fh-erfurt.de
Date: Tue, 05 May 2009 00:05:40 +0200 (CEST)
Message-id: <0KJ500KER40VJMC0@clio.fh-erfurt.de>
Subject: Beispiel SMTP eMail
X-Length: 611
X-UID: 7503

Hallo,
das ist eine eMail.
Ciao
```

SMTP: Multipurpose Internet Mail Exchange

- Header `Content-Type`:
  - Aufbau: `Content-Type: type/subtype; parameters`
  - anhand `subtype` wird häufig externe Applikation (Viewer) gestartet
  - die Angabe von Parametern ist optional
  - Beispiele für `type/subtype`

`text`	`plain, html`, Parameter enthält z. B. Zeichensatz
`image`	`jpeg, jpg, gif`
`audio`	`basic` (8-bit µ-law encoded), `32kadpcm` (32 kbps coding)
`video`	`mpeg, quicktime`
`application`	`msword, octet-stream` (für beliebige unbekannte binäre Daten)
`multipart`	`mixed`, Parameter gibt Trenner

- Header `Content-Transfer-Encoding`:
  - Aufbau: `Content-Transfer-Encoding: mechanism`
  - Mail Reader kann bei unbekanntem `mechanism` Inhalt der E-Mail nicht darstellen
  - typische Verfahren: `base64, 7bit, 8bit, binary, quoted-printable`

SMTP: MIME Beispiel


```
From: gunar@schorcht.net
To: schorcht@fh-erfurt.de
Date: Tue, 28 Oct 2003 23:09:21 +0100 (CET)
Subject: Multipart Test
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----------060003040705070005060705"

------------060003040705070005060705
Test Teil 1 – nur Text.
------------060003040705070005060705
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
base64 encoded data .....
.........................
......base64 encoded data
------------060003040705070005060705
Test Teil 3 – wieder nur Text
```

- Anmerkungen
  - die `boundary` darf nicht im Body der E-Mail vorkommen
  - deshalb wird in der Regel eine lange Folge zufälliger Zeichen gewählt
  - anhand der `boundary` sind Rückschlüsse auf den Mailreader möglich

SMTP: Abruf Emails

- bisher betrachet: SMTP
  - Transfer von E-Mails zwischen Mail Servern
  - Transfer von E-Mails zwischen Mail Reader und Mail Rerver (nur diese Richtung!)
- Frage: Wie kann Mail Reader auf eingehende E-Mails zugreifen?
- Möglichkeiten
  - Post Office Protocol - Version 3 (POP3, RFC 1939, Mai 1996)
    - E-Mails vom Mail Server auf lokalen Host geladen und ggf. auf Mailserver gelöscht
    - E-Mails verteilt aufbewahrt, bei Wechsel der Clients wird Mailbox inkonsistent
  - Internet Message Access Protocol - Version 4 (IMAP4, RFC 1730, Dezember 1994)
    - E-Mails bleiben auf Mailserver - in der Regel keine Kopien auf lokalem Host
    - Zugriff auf eindeutigen Server, d. h. Mailbox bleibt auch bei Wechsel der Clients konsistent
  - Zugriff über HTTP/HTTPS mit WWW-basierten E-Mail-Diensten

HyperText Tranfser Protocol (HTTP)

- Hypertext Transfer Protocol (RFC 2616)
  - setzt zuverlässige TCP-Verbindung voraus
  - ist statuslos (Folge unabhängiger Requests und Responses) ohne Cookies
  - textbasiert (Meldungen im Klartext)
- HTTP-Nachrichten bestehen aus Header und Body
  - Header enthält Kontrollinformationen – Inline Signaling
  - Kontrollinformation werden im Klartext (ASCII) übertragen
  - Body enthält das übertragene Objekt
  - Request kann auch Nutzdaten enthalten (z.B. Password)

HTTP: Message Format

FACHHOCHSCHULE ERFURT UNIVERSITY OF APPLIED SCIENCES
Angewandte Informatik

HTTP Request

Request Line

```
GET /index.html HTTP/1.1
```

General Header

```
Date: Sun, 01 Jun 2008 21:45:41 GMT
```

Request Header

```
Host: www.ai.fh-erfurt.de
User-Agent: Mozilla/5.0 [de]
...
```

Entity Header

```
Accept: text/html, image/jpeg, image/png
Accept-Charset: utf-8, utf-
8;q=0.5Accept-Language: de, en\r\n
...
```

----- Carriage Return – Line Feed -----

Message Body (optional)

```
...
```

HTTP Response

Status Line

```
HTTP/1.1 200 OK
```

General Header

```
Date: Sun, 01 Jun 2008 21:45:43 GMT
```

Response Header

```
Host: www.ai.fh-erfurt.de
Server: Apache/2.0.54 (Linux/SUSE)
...
```

Entity Header

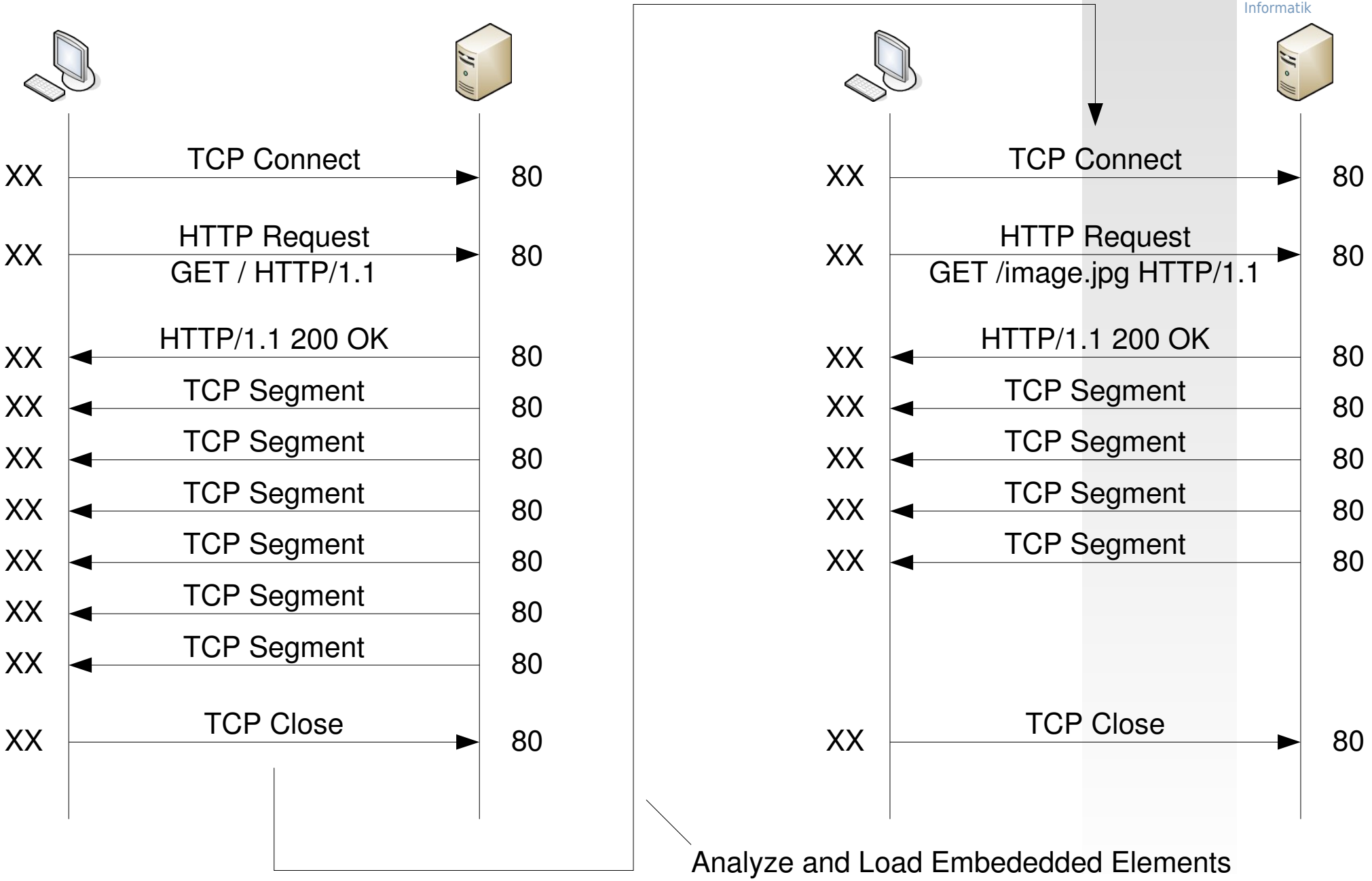
```
Last Modified: Tue, 4 Sep 2007 12:01:45
Content-Type: text/html;charset=utf-8
...
```

----- Carriage Return – Line Feed -----

Message Body

```
<html>
<body>
Hello world.
</body>
</html>
```

HTTP: Ablauf


XX
TCP Connect
80
XX
HTTP Request
GET / HTTP/1.1
80
XX
HTTP/1.1 200 OK
80
XX
TCP Segment
80
XX
TCP Segment
80
XX
TCP Segment
80
XX
TCP Segment
80
XX
TCP Segment
80
XX
TCP Close
80
XX
TCP Connect
80
XX
HTTP Request
GET /image.jpg HTTP/1.1
80
XX
HTTP/1.1 200 OK
80
XX
TCP Segment
80
XX
TCP Segment
80
XX
TCP Segment
80
XX
TCP Segment
80
XX
TCP Close
80
Analyze and Load Embededded Elements