Web Engineering II

serverseitige Webentwicklung

HERZLICH WILLKOMMEN

Dennis Holtz
Enrico Keil

- Gegründet 2003
- Betreuung von kleinen und mittelständigen Unternehmen
- Konzeption und Entwicklung von Softwarelösungen für den Einzelhandel
- Bundesweite Tätigkeit im Consulting und Netzwerk-Monitoring
- Bereitstellung spezieller Hostinglösungen für andere Dienstleister
- Entwicklung und Vertrieb eigene Softwarelösung mit knapp 1.000 Kunden weltweit

THEMEN

- Allgemeines zur Vorlesung
- Internet und Protokolle
- Client-Server Architektur
- Cloud vs. Server
- Websprachen und Webserver
- Entwicklung auf Basis von NodeJS
- Continuous Integration
- Unit Testing
- Hosting as a Service
- Praxistipps

Allgemeines zur Vorlesung

- Vorlesung: jeweils Dienstags 08:15 - 11:30 Uhr
- Zeitraum: 07.03.2018 - 16.05.2018 (11 Wochen)
- Prüfungsleistung: Klausur oder praktische Übung
- Quellcode online: github.com/dpetrasch/DHBW
- Fragen: jederzeit gerne

Präsentation Projekt - Bewertungskriterien

- Vorstellung Eurer Idee (dabei keine Bewertung der Sinnhaftigkeit)
- (kurz) wer hat was im Team gemacht
- Welchen Zeitumfang hat das Projekt eingenommen
- Welche Technologien (Basis Webserver / Datenbank / Framework / ext. Komponenten) wurden verwendet
- Was war das größte Problem, war dies vorher auch absehbar oder überraschend
- Wo konnte man sich Arbeit sparen, indem man auf fertige (Teil-)lösungen zurückgreifen konnte
- Wenn Ihr nochmal von vorn anfangen müsstet, was würdet Ihr anders machen - was würdet Ihr Euren Kollegen empfehlen, wenn sie eine ähnliche Problemstellung bewältigen müssen.
- Live-Demo auf extern erreichbarem Server, Darstellbarkeit der Lösung auch auf mobilen Geräten

Modulinhalte

Lerneinheiten und Inhalte		
Lehr- und Lerneinheiten	**Präsenz**	**Selbststudium**
Labor Compilerbau	**36,0**	**39,0**
- Generatoren zur Strukturanalyse: LEX, Spezifikation regulärer Sprachen, YACC, Spezifikation kontextfreier Sprachen, Praktische Anwendungen - Implementierung der Semantischen Analyse - Bytecodegenerierung		
Web-Engineering 2	**36,0**	**39,0**
- Vertiefung oder Erlernen einer serverseitigen Programmiersprache bzw. Vertiefung oder Erlernen von JavaScript als sinnvolle Ergänzung und Fortführung von Modul Web-Engineering I - Spezielle Verwendungskontexte von JavaScript wie z.B. (SVG, Ajax, usf.) und/oder die Betrachtung Zusammenhang üblicher APIs wie z.B. jQuery - Handhabung medialer Objekte - Spezielle Laufzeitumgebungen für Web-Anwendungen wie (z.B. Tomcat oder .NET usf.) und deren aktuell übliche APIs		
Compilerbau	**36,0**	**39,0**
- Lexikalische Analyse - Syntaktische Analyse - Syntaxgesteuerte Übersetzung - Semantische Analyse - Laufzeit-Organisation - Zwischencode-Erzeugung - Code-Optimierung - Code-Erzeugung		

tr.im/modbes

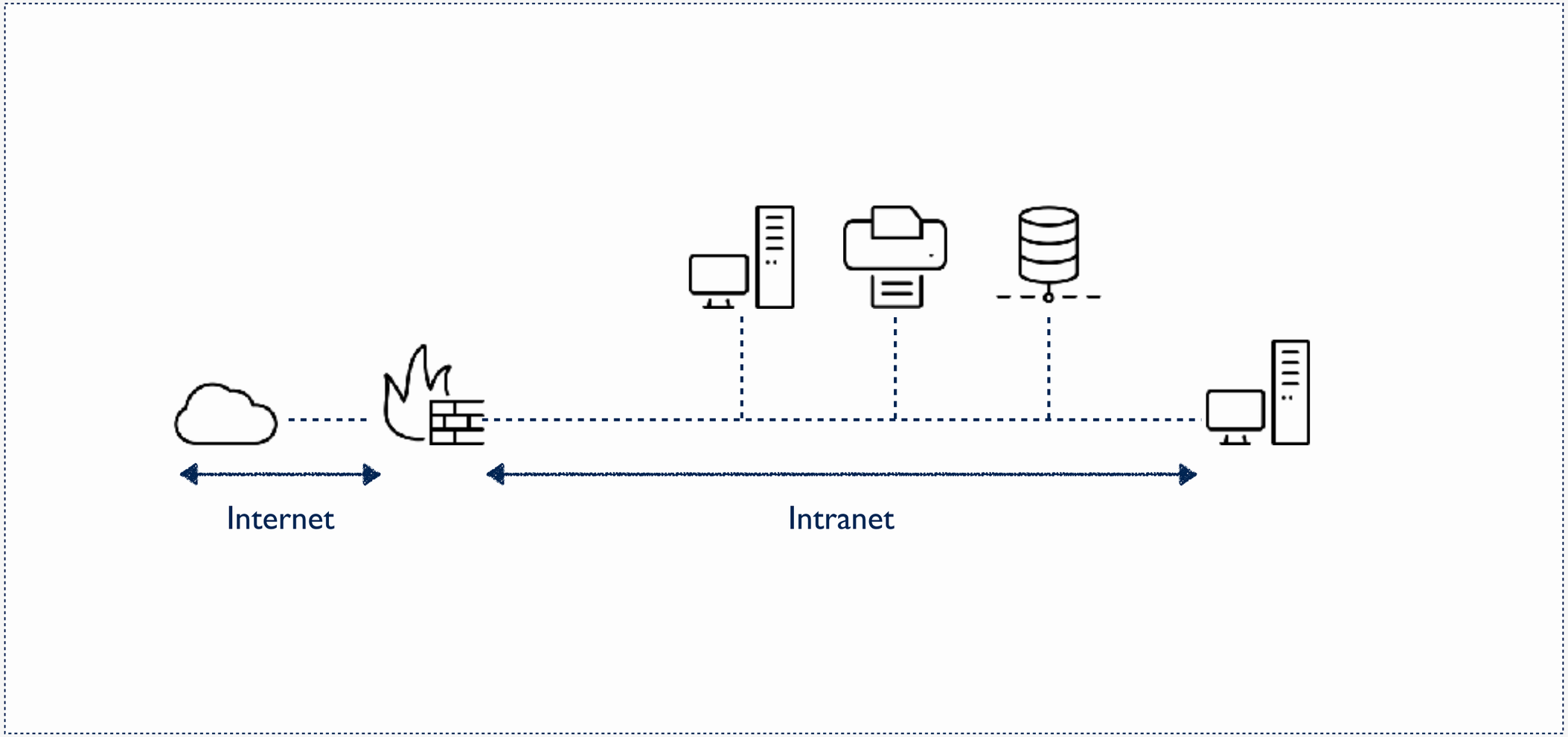
Internet und Intranet

IPv4 und IPv6

Internet

- Webtechnologien bauen auf dem Internet auf
- Das WWW ist nur ein Teil des Internets
- Technologische Basis des Internets ist das Internet Protocol
- Jeder (direkt im Internet erreichbare) Rechner ist eindeutig durch eine IP-Adresse identifiziert

Internet und Intranet



IPv4 und IPv6

- 32 Bit Adresslänge
- 4 Gruppen zu je 8 Bit
- Darstellung Dezimal
- $4{,}3 \cdot 10^9$ Adressen

00101110.00000100.00011010.01010111

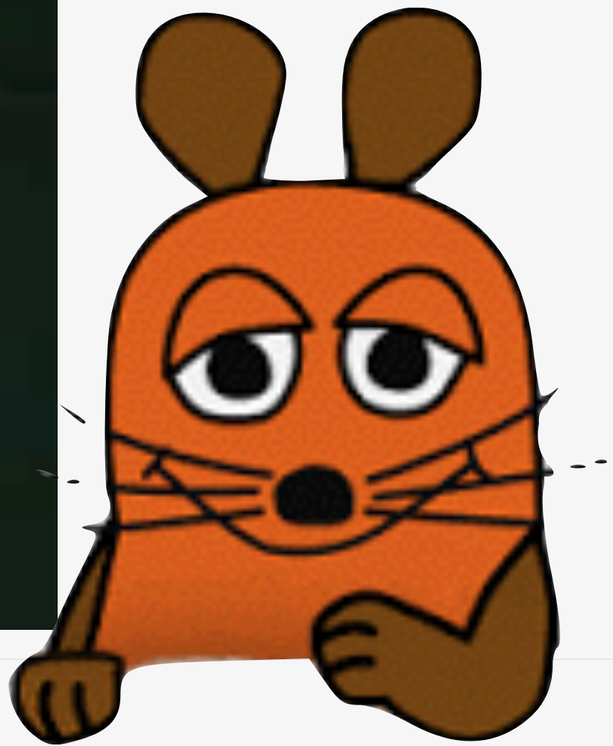
46.4.26.87

- 128 Bit Adresslänge
- 8 Gruppen zu 16 Bit
- Darstellung **Hexa**dezimal
- $3{,}4 \times 10^{38}$ Adressen

2001:4860:A002:0000:0000:0000:0000:0068

2001:4860:A002::68

Wie funktioniert das Internet?



Dual-Stack

IPv4 und IPv6 Anschlüsse

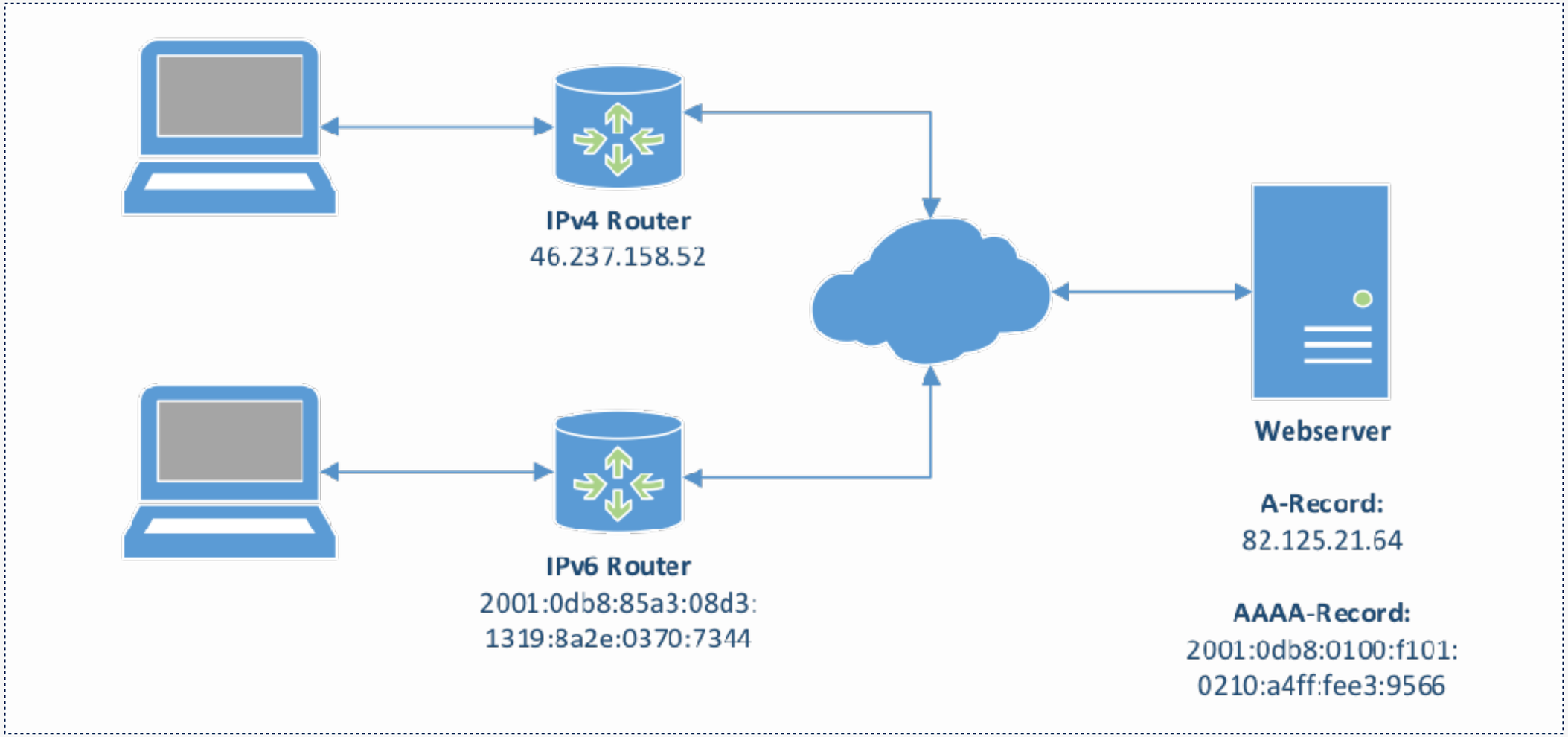
Dual Stack

- Dual-Stack beschreibt den Parallelbetrieb von IPv4 und IPv6
- Es wird eine IPv4 mit vollwertigem NAT verwendet
- Sowie ein IPv6 Netz bereitgestellt

Dual Stack

- Entworfen für den Übergang bis nur noch IPv6 genutzt wird
- Unterstützung aller Geräte notwendig
  - Betriebssysteme
  - Router / Switche
  - Server

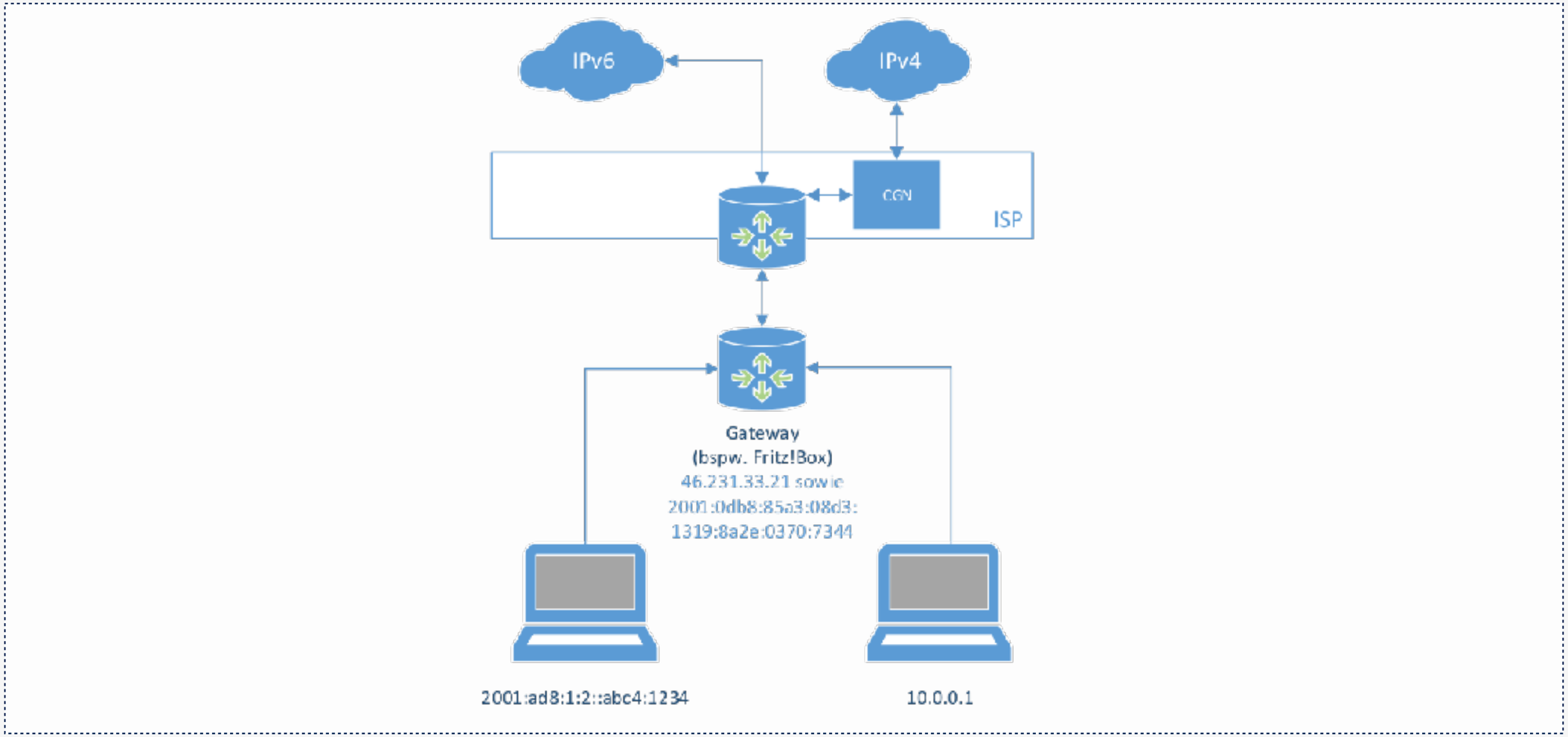
Dual Stack


IPv4 Router
46.237.158.52
IPv6 Router
2001:0db8:85a3:08d3:
1319:8a2e:0370:7344
Webserver
A-Record:
82.125.21.64
AAAA-Record:
2001:0db8:0100:f101:
0210:a4ff:fee3:9566

Dual Stack Lite

- Einschränkung: Carrier Grade NAT
  - Mehrere Nutzer teilen sich eine IPv4-Adresse
  - Kein NAT möglich
  - bspw. Unitymedia-Anschlüsse

Dual Stack Lite



Vorteile

- Nutzern werden auf IPv6 umgestellt
- kompatible IPv6-Dienste können genutzt werden
- zusätzlich Fallback auf IPv4
- Jedes Gerät erhält eine eigene IP-Adresse

Nachteile

- Dual Stack Lite
  - durch Carrier Grade NAT kein eigenes natten möglich
- Webhoster und Cloud-Plattformen unterstützen noch kein IPv6 (AWS, Azure, o.ä.)

Tunnelprovider

- Tunnelprovider ermöglichen IPv4 -> IPv6 Tunnel (Portmapper)
- Somit direkter Zugriff auf IPv6 Geräte über CGN möglich

Tunnelprovider

tr.im/festeip

Client / Server

Dienstbereitstellung

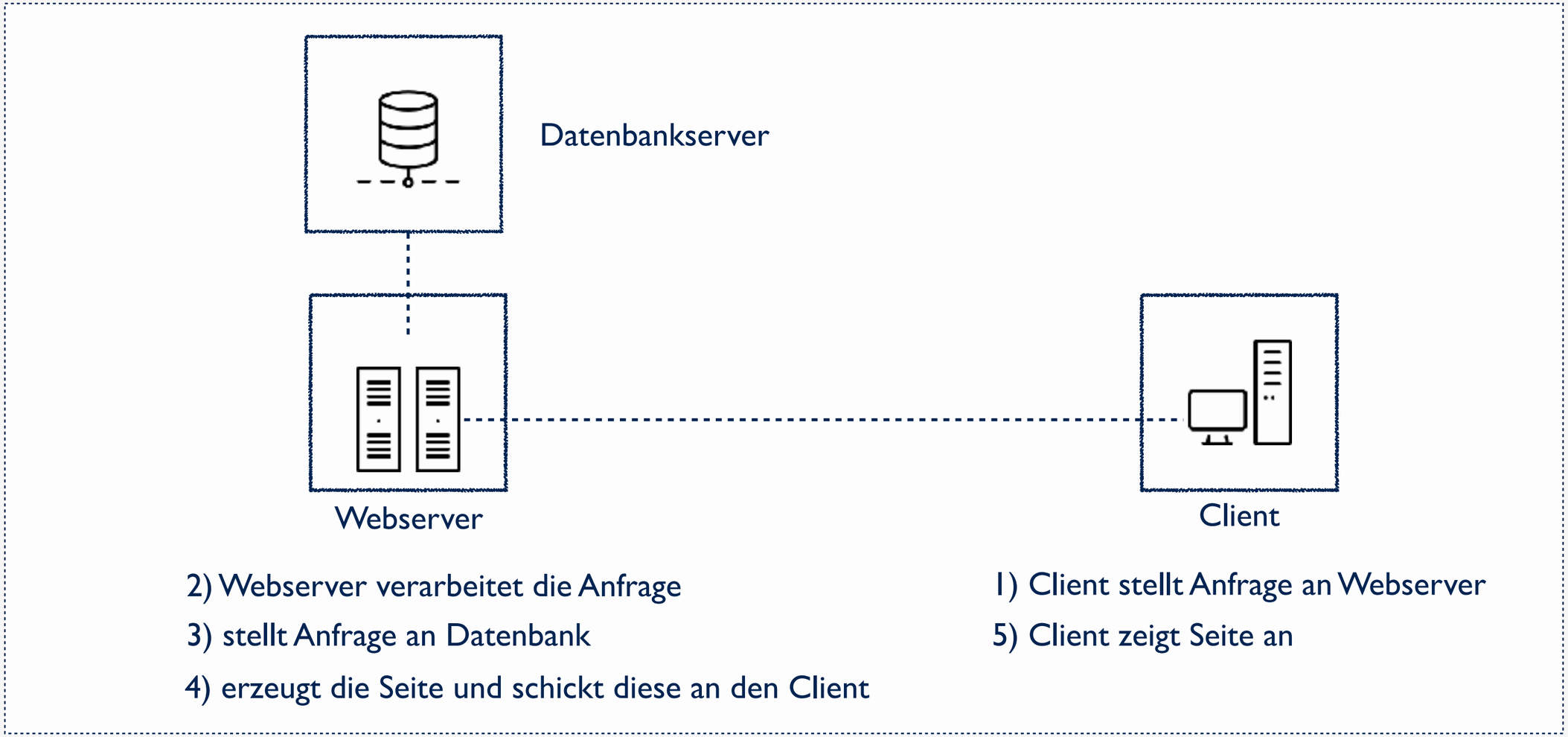
Client / Server

- Anfragen werden von einem Dienstnutzer (Client) an einen Anbieter (Server) gestellt
- Beim Zugriff aufs „Internet“ ist der Client in der Regel ein Endgerät wie z. B. Computer, Smartphones, Tablets, Infotainmentsysteme in Haushalt und Kfz
- Der Anbieter stellt unterschiedliche Dienste für einen oder mehrere Clients bereit
- Backendserver (bspw. Datenbankserver) kommunizieren nicht direkt mit dem Client

Typische Dienste

- File-Server
  - Bereitstellung von Daten über verschiedene Protokolle, in der Regel im LAN
- Datenbankserver
  - Stellt Informationen einer oder mehrerer Datenbanken bereit
  - Wird in der Regel über Anwendungen oder Webdienste angesprochen
- Groupwareserver
  - Stell Zugriff auf Mailpostfächer, Kalender, Kontakte, Workflows etc. bereit
- Mailserver
  - Nimmt E-Mails entgegen und sendet sie weiter, verarbeitet Mails an Hand von Regeln
- Webserver
  - Stellt Informationen über HTTP zur Verfügung, greift dabei auf Daten anderer Server zu

Kommunikation im Web



NAT

Network Address Translation

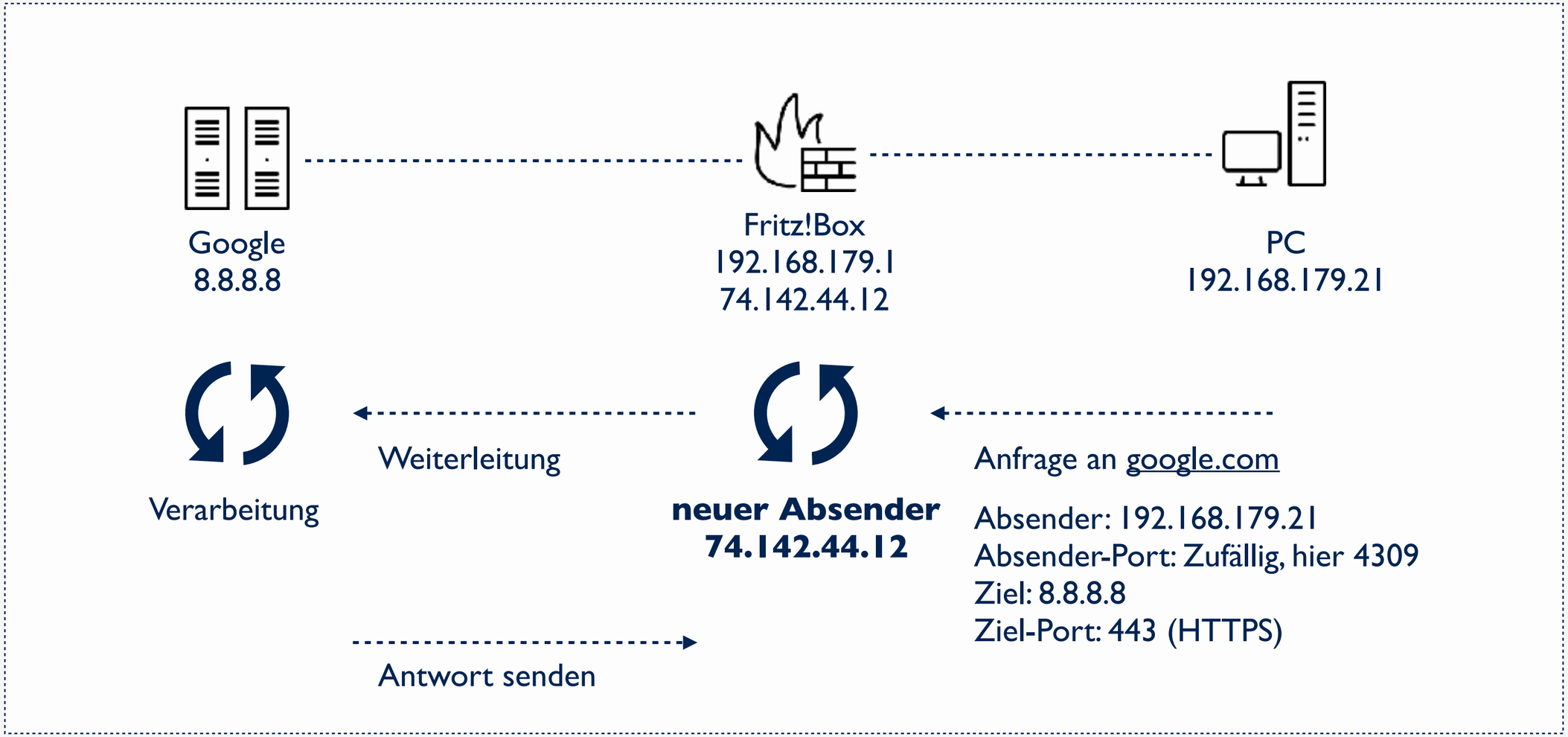
Network Address Translation

- Network Address Translation dient zum Verbinden von unterschiedlichen Netzen
- wird in Source-NAT und Destination-NAT unterteilt

Source-NAT (Masquerading)

- Mehrere Netzwerkgeräte kommunizieren über eine gemeinsame IP-Adresse nach außen. Hierbei wird die Quelladresse geändert.
- wird beispielsweise bei dem heimischen Router verwendet, um mit mehreren Geräte über eine IP-Adresse zu kommunizieren
- Der Router „übersetzt“ und ersetzt die interne Adresse (bspw. 192.168.179.12) mit der gemeinsamen Adresse (74.142.44.12) und ändert die in den Datenpaketen stehenden Quell- und Zieladressen, damit das Paket das Ziel erreicht

Source-NAT

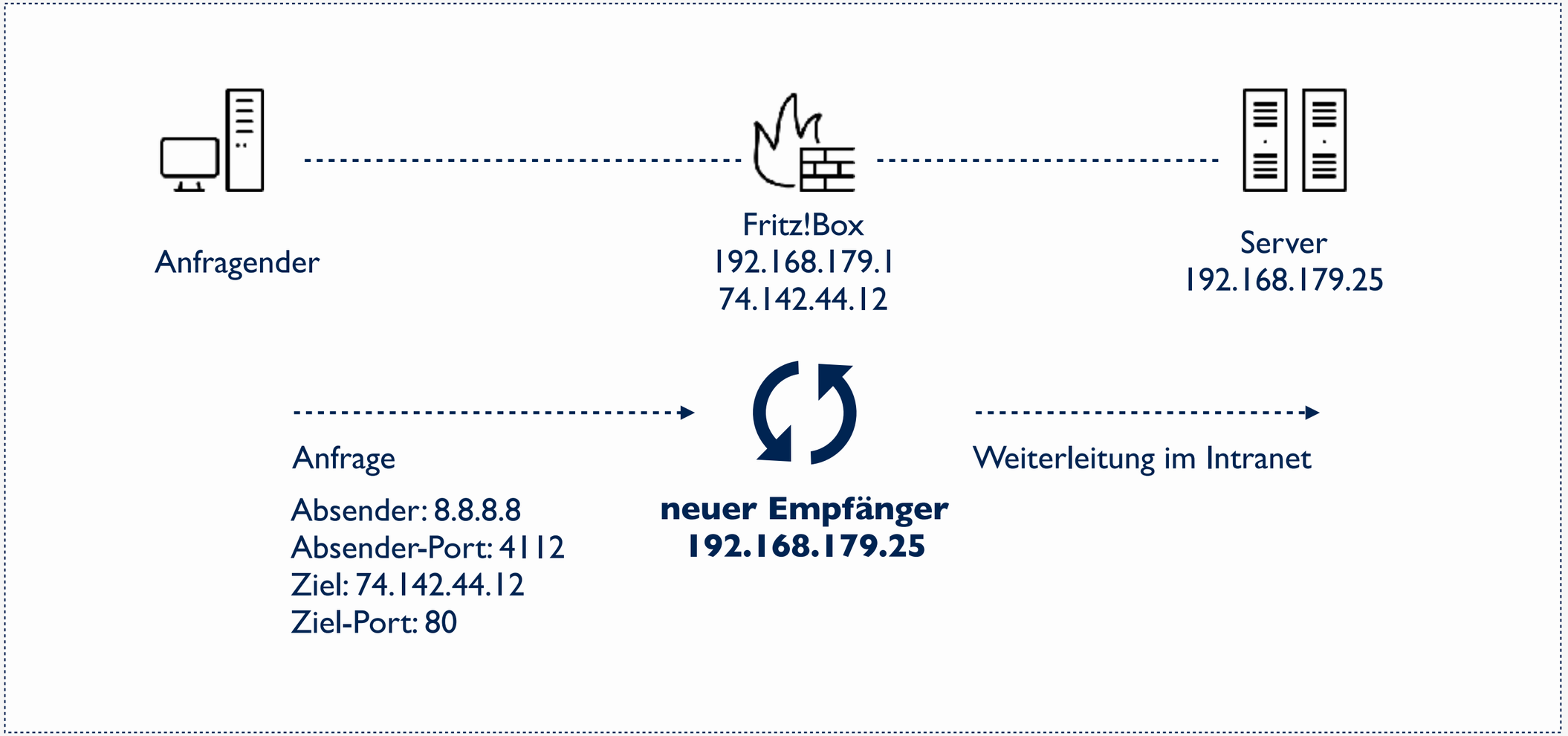

Google
8.8.8.8
Fritz!Box
192.168.179.1
74.142.44.12
PC
192.168.179.21
Verarbeitung
Weiterleitung
neuer Absender
74.142.44.12
Anfrage an google.com
Absender: 192.168.179.21
Absender-Port: Zufällig, hier 4309
Ziel: 8.8.8.8
Ziel-Port: 443 (HTTPS)
Antwort senden

Destination-NAT

- Wird genutzt, um die Zieladresse in eingehenden Paketen zu ändern.

  Pakete an 74.142.44.12:80 werden so beispielsweise zu 192.168.179.25:80 übersetzt
- nötig, um aus dem Internet auf Dienste zuzugreifen, die auf Maschinen mit privaten Adressen arbeiten („Portfreigabe“)

Destination-NAT



Webserver

Übersicht und Marktanteile

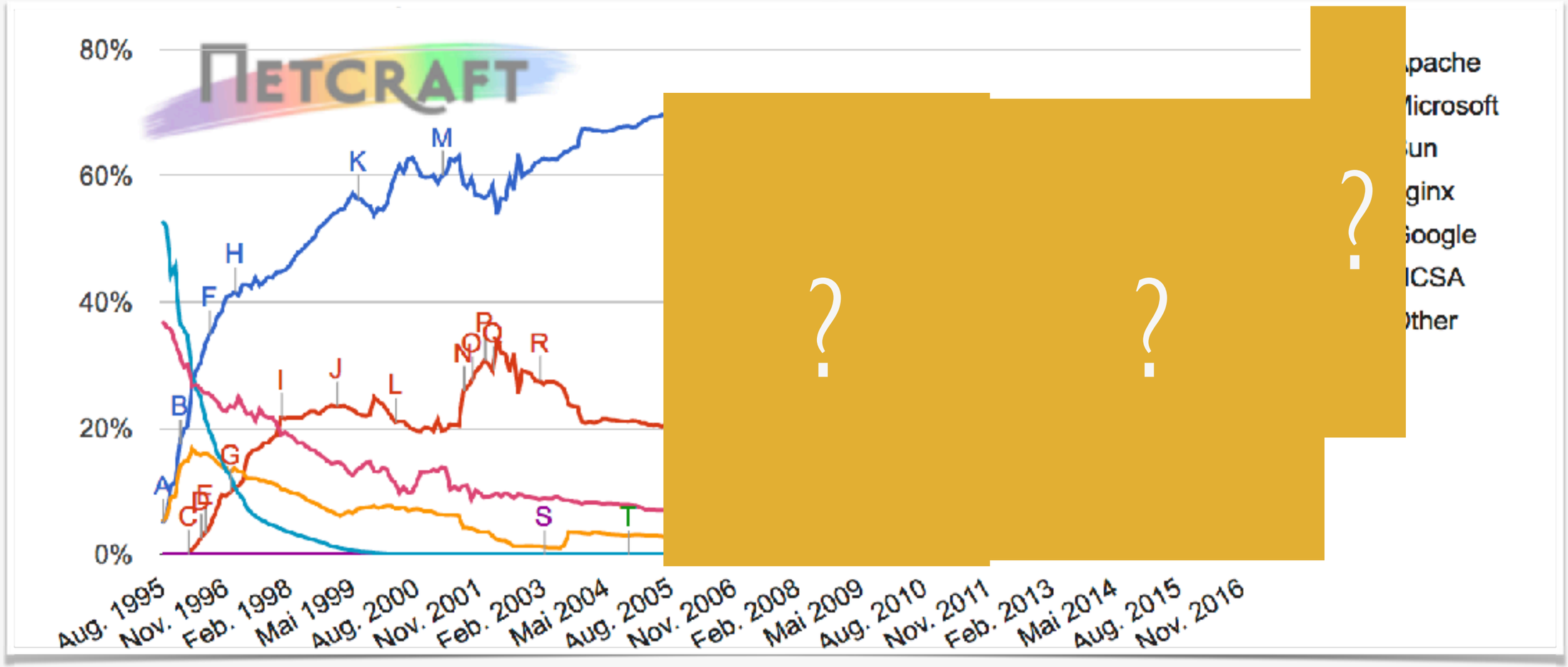
Webserver

- Ein Serverdienst, der Informationen über das HTTP-Protokoll zur Verfügung stellt
- Stellt Informationen über das HTTP-Protokoll zur Verfügung
- TCP Port 80 für HTTP
- TCP Port 443 für HTTPS (SSL/TLS)

Bekannte Webserver

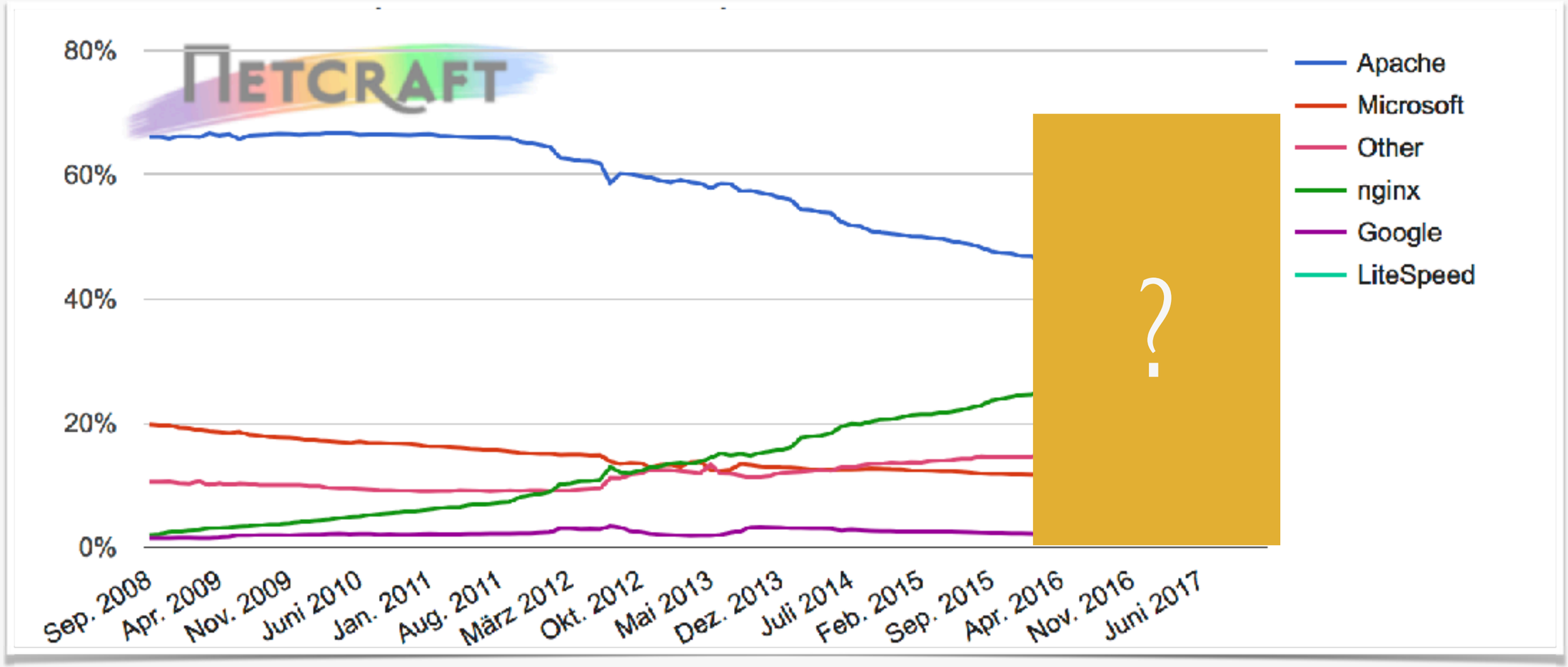
- Apache HTTP Server
- Microsoft IIS Server
- LiteSpeed Web Server
- Nginx
- Google Web Server

Webserver Marktanteile: Alle Seiten



tr.im/netcraft

Webserver Marktanteile: Top Million



trim/**netcraft**

HTTP

Abfragen gegen den Webserver

HTTP Protokoll

- HTTP Anfragen sind zustandslos, nach der Auslieferung der Antwort wird die Verbindung gelöst
- Der Client fragt einen URL (Uniform Resource Locator) an und der Webserver antwortet mit einem HTML-Dokument.
- Verweist der URL auf ein serverseitiges Skript, so wird dies zunächst ausgeführt und das generierte HTML-Dokument wird ausgeliefert.

HTTP Methoden

- GET
  - Informationen abfragen
- POST
  - Daten an den Server senden
- PUT
  - Element aktualisieren
- DELETE
  - Element löschen
- HEAD
  - Lesen der Dateiinformationen eines Dokuments
- OPTIONS
  - Angabe der erlaubten Methoden für diese URL

URL-Anfrage

steht für eine Anfrage nach dem Hypertext Transfer Protocol

steht für den Uniform Resource Locator (URL)

steht als Trennzeichen zwischen URL und Parametern

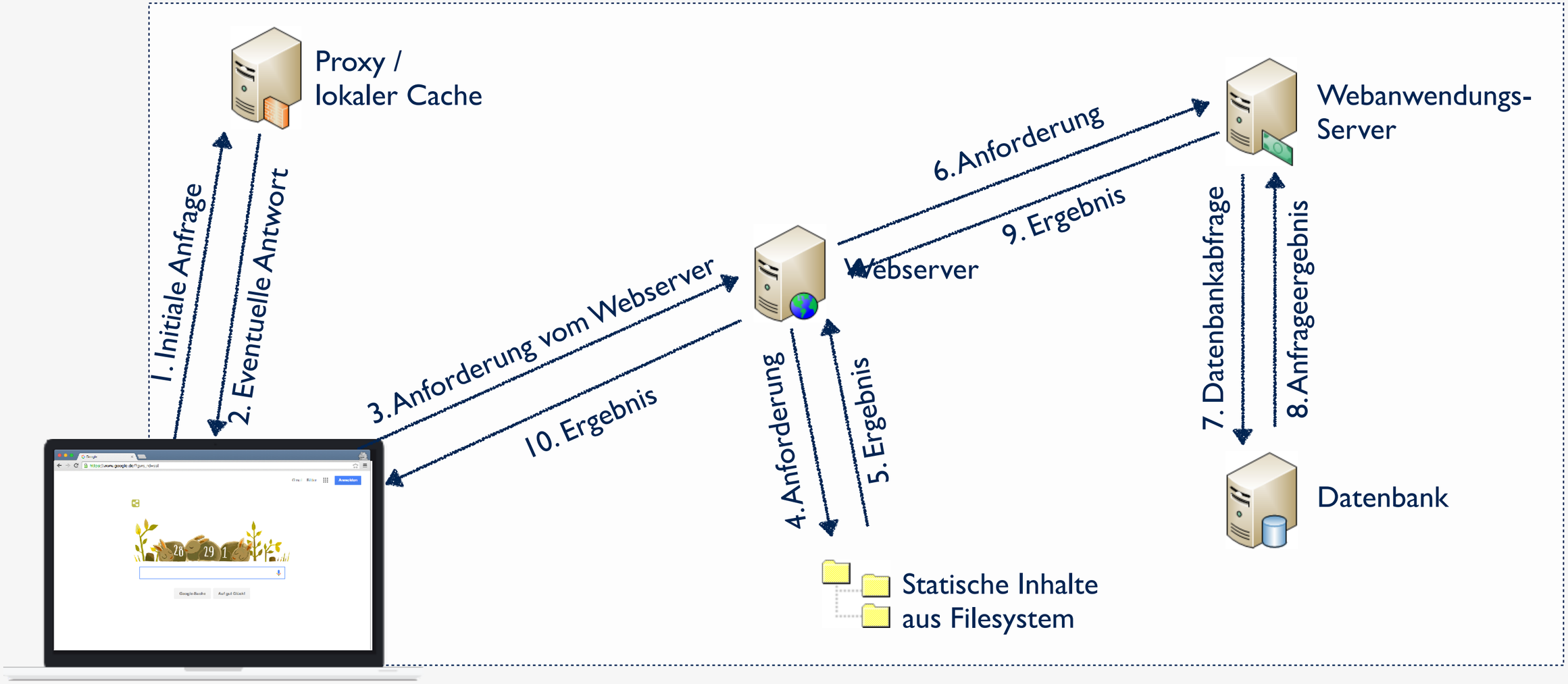
erster Parameter und Wert der Anfrage im Key-Value-Format

steht als Trennzeichen zwischen verschiedenen Parametern

zweites Key-Value Paar

Quelle: wikipedia.de

Schematischer Ablauf einer (dynamischen) Webanfrage



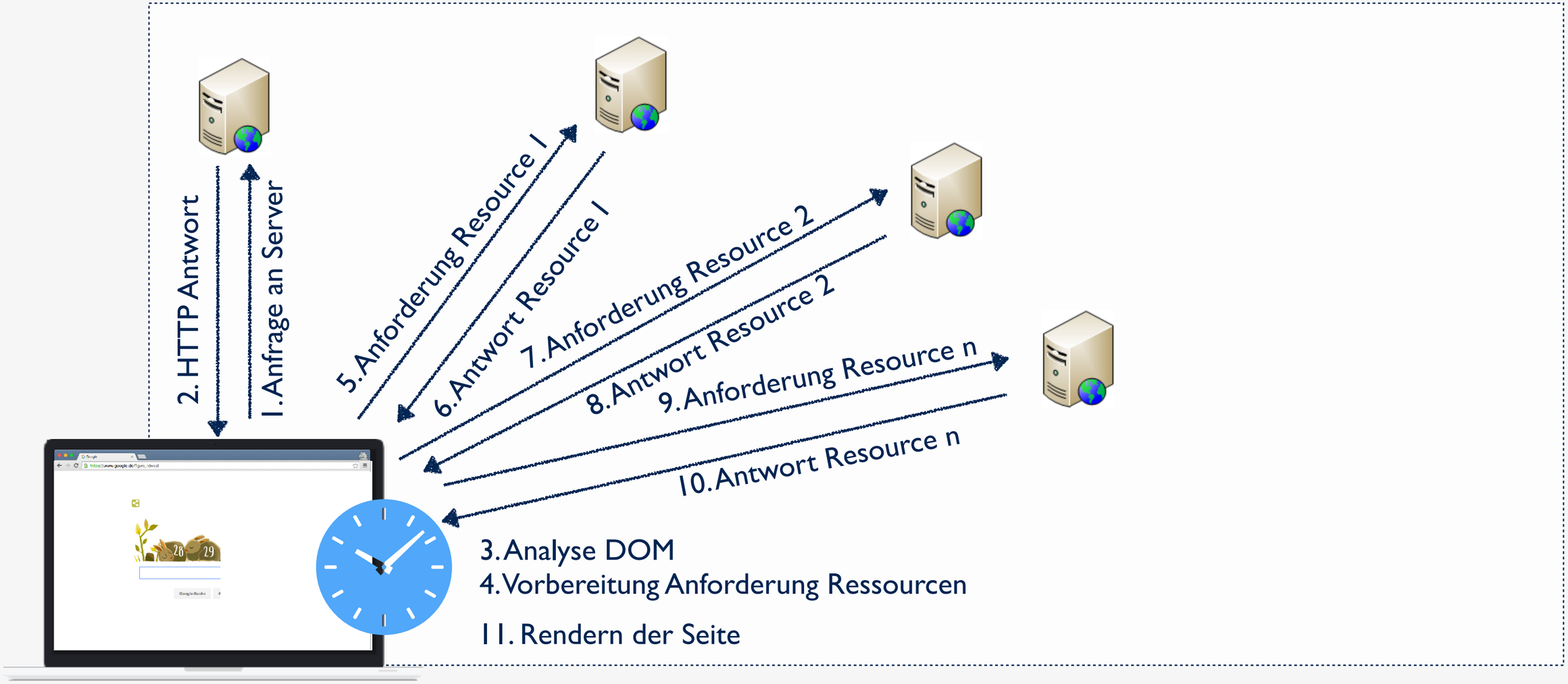
Sessionverwaltung

- HTTP ist ein zustandsloses Protokoll. Nach Bedienung der Anfrage durch den Server besteht keine Verbindung mehr zum Client.
- Gleichzeitig besteht aber die Notwendigkeit, einen Benutzer über mehrere Seiten hinweg zu verfolgen
  - Mehrstufige Formulare
  - Warenkorb beim Online-Shopping
  - Benutzer-Tracking für Werbetreibende und Social Networks
  - Webmailer/CMS/etc. mit sich öffnenden Fenstern zum Verfassen von Artikeln
- Varianten zur Realisierung des Session-Management
  - Hidden Fields in Formularen
  - URL Rewriting
  - SSL-Key der HTTPS-Verbindung
  - Cookies

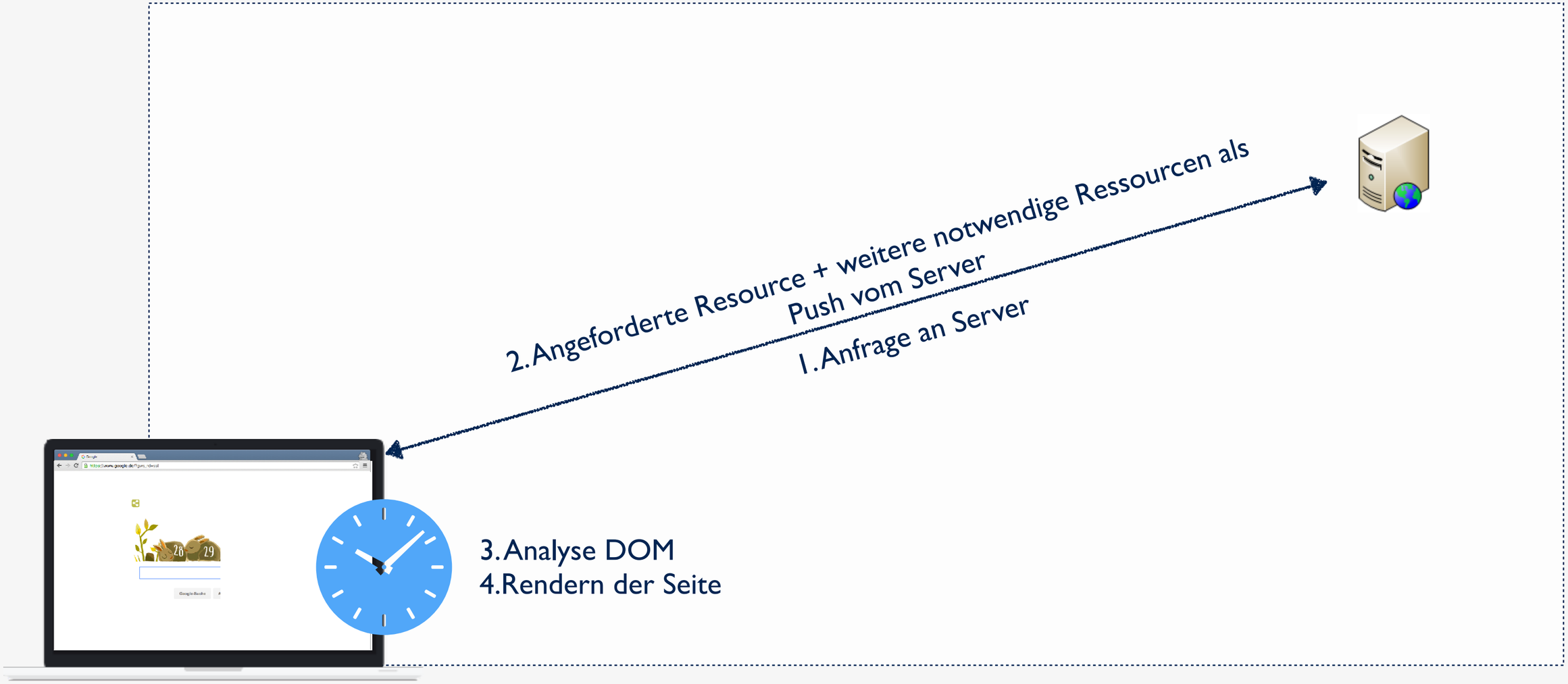
HTTP/2

- Nachfolger der Version 1.1 von 1996
- Aus dem Protokoll „SPDY“ von Google entstanden
- Bereits in allen verbreiteten Browsern und Webservern implementiert
- Zwischen Client und Server wird eine dauerhafte Verbindung aufgebaut und Daten werden über diese Verbindung ausgetauscht (Multiplexing)
- Entwicklung für aktuelle Anforderungen um folgende Probleme zu lösen
  - Performanceprobleme durch viele Dateien (CSS/JS/Images) beim Aufruf von Seiten
  - Große Header mit Session-Informationen
  - Nachladen der notwendigen Ressourcen nach Initialisierung des DOM
  - Fehlende durchgängige Kompression

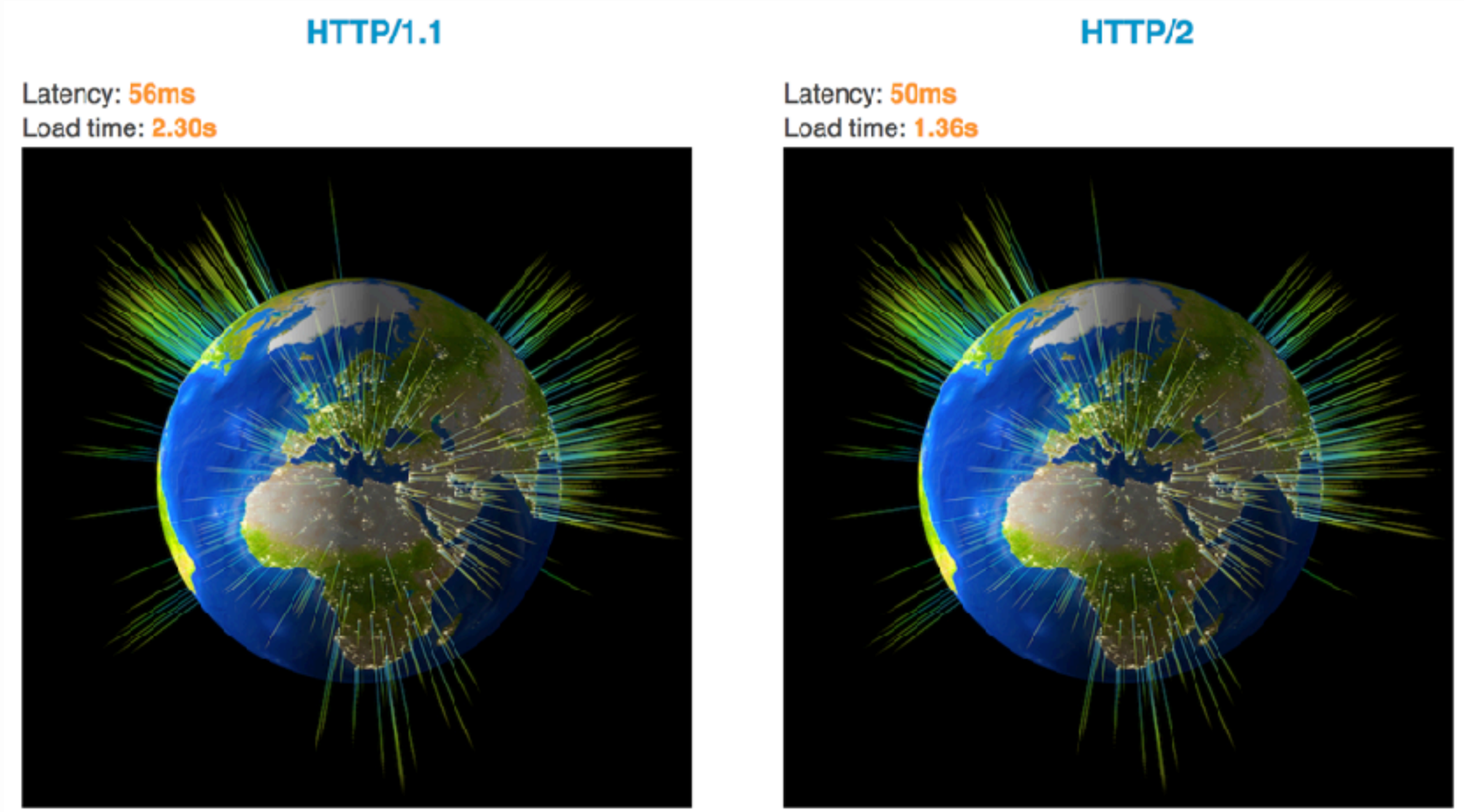
Anfrage HTTP/1.1



Anfrage HTTP/2



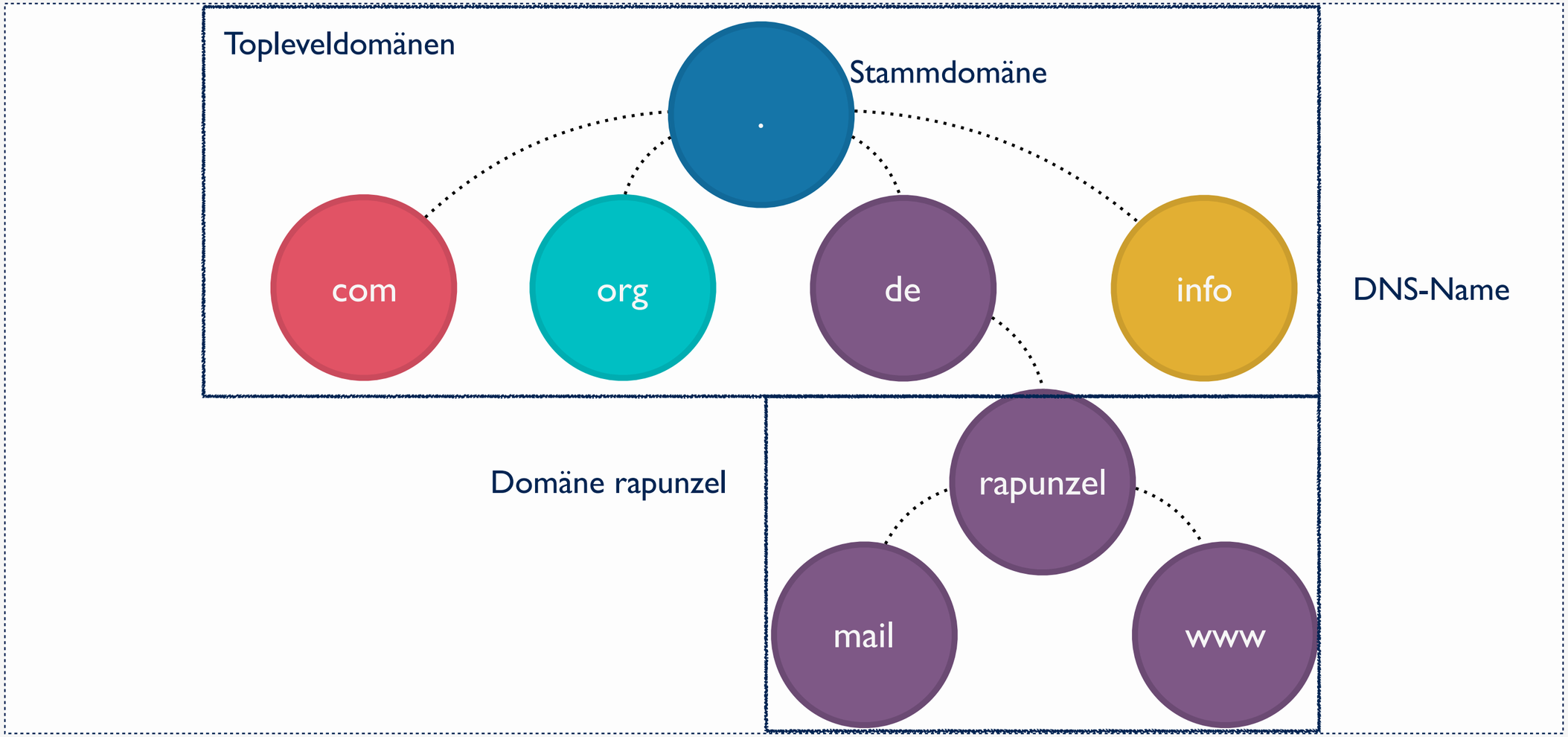
Aufbau HTTP/1.1 vs HTTP/2



DNS

Domain Name System

DNS Namespace



Quelle: wikipedia.de

Wichtige Begriffe

DNS Server	jeder Computer, der den DNS-Dienst ausführt, eine DNS-Datenbank hält und auf Clientanfragen antwortet
DNS resolvers	Programme, die mit Hilfe von DNS-Abfragen Informationen auf den Servern abfragen; Resolver können auf Clients oder Servern ausgeführt werden
Resource records	Informationen in der DNS-Datenbank, die für die Verarbeitung von Clientanfragen verwendet werden. DNS-Server halten Ressourcen-Einträge für den Anteil des Namespace, für den sie autorisiert sind.
Zones	fortlaufende Anteile des DNS-Namespace, für die der DNS-Server autorisiert ist. ein DNS-Server kann für eine oder mehrere Zonen autorisiert sein

Zonen

- Zonen
  - Zone = fortlaufender Anteil am DNS-Namespace
  - jede Zone ist an einem Domänenknoten verankert
  - Zone ≠ DNS-Domäne!
  - DNS-Domäne = Teilstruktur des DNS-Namespace
  - Zone = Teil des DNS-Namespace; kann mehrere DNS-Domänen enthalten
  - können von mehreren Servern verwaltet werden
  - Unterscheidung: primäre Zone / sekundäre Zone (Kopie)
  - Speicherung in Textfiles, AD-Integration möglich

DNS Server

- speichert Informationen über Zonen
- bedient Anfragen von Clients
  - nutzt eigene Zonen
  - nutzt Cache (auch negativen Cache)
  - fragt andere Server
  - leitet weiter
- kann gleichzeitig primärer Server und sekundärer Server einer anderen Zone sein
- nimmt Änderungen nur für primäre Zonen entgegen

Vorteile von Sekundarservern

- Fehlertoleranz
- Reduktion der Netzwerklast
- Reduktion der Datenlast auf Primärserver
- Caching-only Server zur weiteren Performance-Steigerung

Forwarders und Slaves

- bei nichterfolgreicher Abfrage der eigenen Zone ist Kontaktierung anderer DNS-Server möglich
- dadurch unter Umständen hohe Netzlast
- Vermeidung durch Definition von Weiterleitungsservern, die weitere Abfragen durchführen
- dadurch Zentralisierung der Abfragen für fremde Zonen => bessere Cacheausnutzung
- nicht exklusiver Modus = Forwarder
- exklusiver Modus = Slave

  dieser versucht auch nach fehlgeschlagener Weiterleitung keine eigenen zusätzlichen DNS-Abfrage

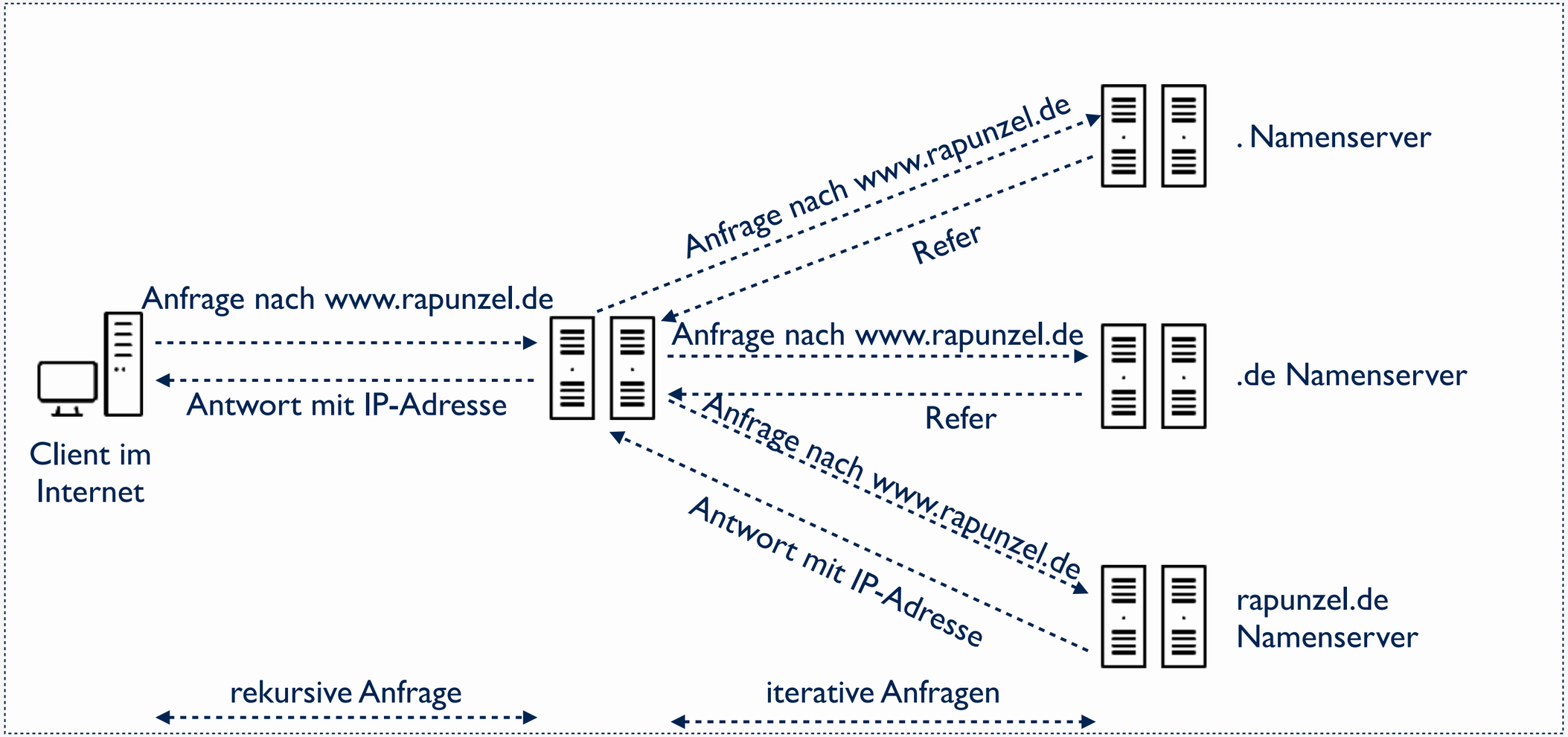
Lastaufteilung

- DNS-Server können einfache Lastaufteilung mittels Round-Robin-Verfahren durchführen
- einem Namenseintrag sind mehrere IP-Adressen zugeordnet
- bei Anfragen werden die Adressen in rotierender Folge ausgegeben
- sinnvoll z.B. bei WWW – Servern
- keine wirkliche Lastaufteilung, aber eine einfache Methode um Netzwerkressourcen besser auszunutzen

Rekursive und iterative Anfragen

- Rekursive Anfragen
  - grundsätzlicher Anfragetyp von DNS-Clients
  - Client erwartet bei rekursiver Anfrage eine endgültige Antwort des DNS-Servers
  - hat dieser keine Informationen muss er zur Beantwortung einer rekursiven Clientanfrage selbst weitere Anfragen stellen
- Iterative Anfragen
  - Client erlaubt dem Server statt einer endgültigen Antwort auch eine bestmögliche Antwort zu geben, z.B. den wahrscheinlich zuständigen DNS-Server
  - DNS-Server fragt dann bei diesem wieder iterativ nach
  - durchläuft auf diese Weise den DNS-Baum - „walking the tree“
  - üblicher Anfragetyp zwischen DNS-Servern

Einführung in DNS: Beispiel



Resource Records

Owner (Besitzer)	Name des Hosts oder der DNS-Domäne, zu dem/der der Eintrag gehört
TTL	32-Bit Integer, der die Gültigkeitsdauer angibt (optional)
Class	Protokollfamilie, immer „IN“
Type	Typ des Ressourceneintrags
RDATA	Ressourceneintragsdaten, variabler Typ, je nach Type des Eintrags

SOA Resource Record (ursprung einer Zone)

- RDATA enthält:
  - authorized server = primärer Server der Zone
  - responsible person (Achtung: kein „@“, sondern „.“ nutzen)
  - serial number (Anzahl der Aktualisierungen der Zone)
  - refresh (wie oft ist die Zone zu replizieren)
  - retry (Timeout für Zonentransfer)
  - expire (Verfall der Zone, wenn Zonentransfer nicht erfolgreich)
  - minimum TTL (Default-TTL-Wert für alle Einträge)

```
keil-it.de.   IN SOA (
              dc1.rapunzel.de        ; autorisierter Server
              dnsadmin.rapunzel.de   ; Zonenadministrator
              1441                   ; Seriennummer
              3600                   ; Refresh (1h)
              600                    ; Retry (10min)
              86400                  ; Expire (1d)
              60     )               ; minimum TTL (1min)
```

/dpetrasch/DHBW

NS Resource Record

- geben die für die Zone autorisierten Namensserver an
- jede Zone muss mindestens einen NS-Eintrag enthalten

```
demo.rapunzel.de.    IN    NS    dc1.demo.rapunzel.de
```

/dpetrasch/DHBW

A Resource Record

- Address Resource Record ordnet einem FQDN eine IP zu

```
demo.rapunzel.de.    IN    A    127.22.23.4
```

/dpetrasch/DHBW

PTR Resource Record

- Pointer Resource Record ordnet einer IP den FQDN zu
- rückwärts geschriebener IP & „in-addr.arpa.“

```
4.23.22.172.in-addr.arpa.    IN    PTR    demo.rapunzel.de
```

/dpetrasch/DHBW

CNAME Resource Record

- Canonial Name resource record erstellt einen Aliasnamen
- ein FQDN kann durch mehrere CNAMEs adressiert werden
- ermöglicht Zugriff über „freundliche Namen“

```
www.rapunzel.de.    IN    CNAME    demo.rapunzel.de
```

/dpetrasch/DHBW

MX Resource Record

- Mail Exchange record legt Mailaustauschserver fest
- mehrere Einträge können mit versch. Priorität angelegt werde
- niedrigster Wert bedeutet größte Präferenz

```
*.rapunzel.de   IN   MX    10   mail1.rapunzel.de
*.rapunzel.de   IN   MX    20   mail2.rapunzel.de
```

/dpetrasch/DHBW

SRV Resource Record

- Service Resource record ermöglichen Standortbestimmung bestimmter Dienste
- für die AD-Struktur von extrem hoher Bedeutung

```
_http._tcp.rapunzel.de  IN  SRC  0  0  80  demo1.rapunzel.de

_Service._Proto.Name TTL Class SRV Priority Weight Port Target
```

/dpetrasch/DHBW

Zonen

- Forward Lookup Zonen
  - enthalten Informationen, wie Namensanfragen in IP-Adressen aufgelöst werden
  - es können alle Resource records außer PTR auftreten
  - müssen immer einen SOA und mindestens einen NS Eintrag enthalten
- Reverse Lookup Zonen
  - enthalten Informationen, wie IPs zu Namen aufgelöst werden
  - stellen innerhalb des DNS-Namespaces die Sonderdomäne in-addr.arpa dar
  - können nur SOA, NS, PTR und CNAME enthalten
  - im Regelfall zu jeder FLZ auch eine RLZ

DynDNS

- Dynamisches DNS sind DNS-Einträge mit wechselnden IP-Adressen (bspw. im heimischen Netzwerk)
- Da sich die IP-Adresse in der Regel alle 24 Stunden ändert, muss auch das DNS aktualisiert werden
- Die DNS Einträge dürfen nicht lange zwischengespeichert werden, damit eine schnelle Änderung möglich ist. Hier wird das TTL meist auf 1 Minute gesetzt
- Eine Software oder ein Netzwerkgerät informiert bei einer Änderung der IP-Adresse den entsprechenden Nameserver eines DDNS-Anbieters, welcher den DNS-Eintrag aktualisiert
- Anbieter sind unter anderem:
  - DynDNS.org
  - NO-IP.com
  - Hersteller-eigene Dienste z.B. von NAS-Systemen

Public-Key-Infrastruktur / SSL/TLS

Absichern der Kommunikation

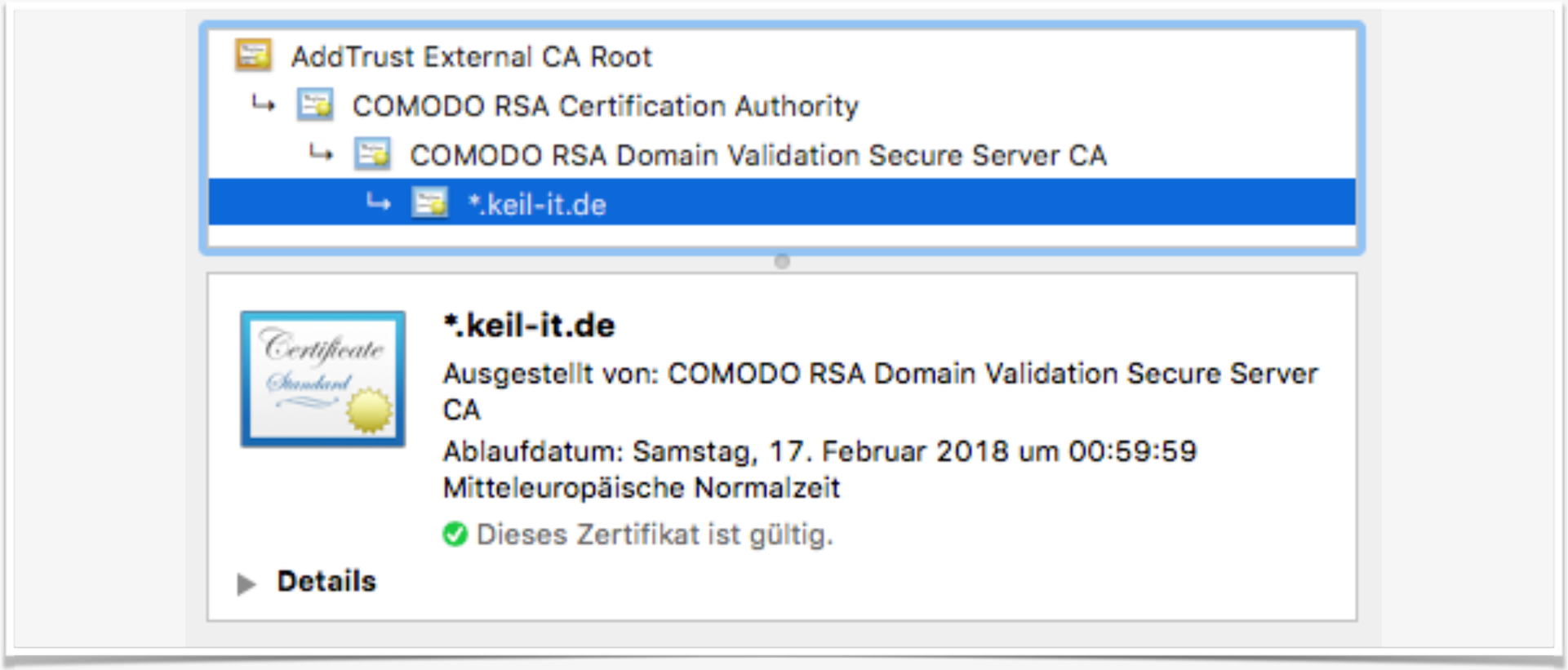
PKI Allgemein

- Public-Key-Infrastruktur bezeichnet das System des Ausstellen, Verteilen und Überprüfen von digitalen Zertifikaten
  - Besteht aus privatem und öffentlichen Schlüssel
- Aussteller ist eine Certification Authority (CA)
  - Prüfung und Ausgabeprozess
  - Verschiedene Sicherheitsklassen

Zertifikatskette

- Hauptzertifikat
  - Zwischenzertifikat
    - Ggf. weiteres Zwischenzertifikat
      - Endzertifikat

Zertifikatskette in Chrome



tr.im/modbes

Vertrauen und Widerruf

- Zertifikat signalisiert Vertrauen, da nur bestimmter Nutzerkreis signieren kann
- Wenn Zertifikat falsch ausgestellt wurde oder privater Schlüssel nicht mehr privat ist, kann das Zertifikat für ungültig erklärt werden
- Jedes Zwischenzertifikat kann nachfolgende Zertifikate widerrufen
  - Revocation-List