Kapitel 3: Anwendungsschicht

Namen und Adressen im Internet

- **Host-Name** (e.g., www.cs.princeton.edu)
  - mnemonisch (für Menschen lesbar und sinnbehaftet), variable Länge, von Menschen verstanden
  - hierarchisch strukturiert, nach Organisation vergeben
- **IP Adresse** (e.g., 128.112.7.156)
  - numerische 32-Bit Adresse, von Routern verstanden
  - hierarchisch strukturiert, nach Organisation und Topologie vergeben
- **MAC Adresse, physikalische Adresse** (e.g., 00-15-C5-49-04-A9)
  - numerische 48-Bit Adresse, von Netzwerkadaptern verstanden
  - unstrukturiert, nicht-hierarchisch, kein Bezug zur Netz-Topologie

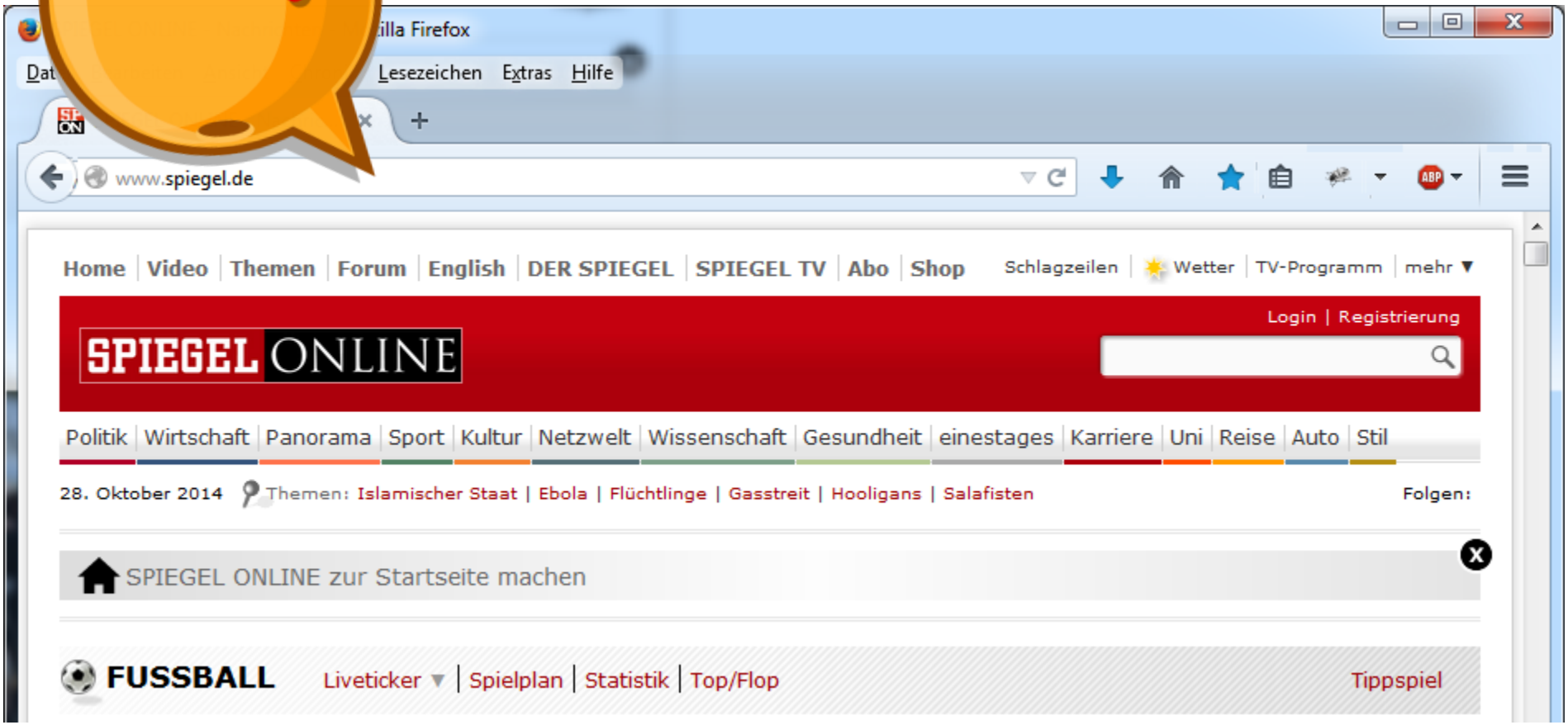
Zuweisung von Namen und Adressen

- **Host-Name:** www.cs.princeton.edu
  - Domain: ICANN (Internet Corporation for Assigned Names and Numbers) vergibt Top-Level Domains zur Verwaltung an NICs (Network Information Center), die wiederum Sub-Domains an Registrare zur Vergabe der Domains an Endkunden übergeben
    - .de: DENIC (www.denic.de)
  - Host-Name: lokaler Administrator weißt den Host-Namen innerhalb der Domain zu
- **IP Adresse:** 128.112.7.156
  - Prefixes: ICANN (IANA, Internet Assigned Numbers Authority), regionale Stellen, und ISPs
  - Hosts: statisch oder mit DHCP konfiguriert (Dynamic Host Configuration Protocol, siehe Kapitel 4)
- **MAC Adresse:** 00-15-C5-49-04-A9
  - Blöcke: von der IEEE an Hersteller vergeben
  - Netzwerkadapter/Netzwerkkarte: vom Hersteller zugewiesen
  - oft frei konfigurierbar

Kapitel 3: Anwendungsschicht

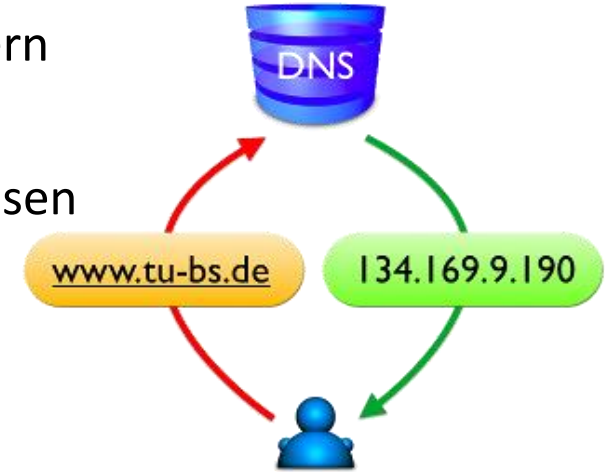
Host-Name und IP-Adresse

Wo schicke ich den GET Request hin????

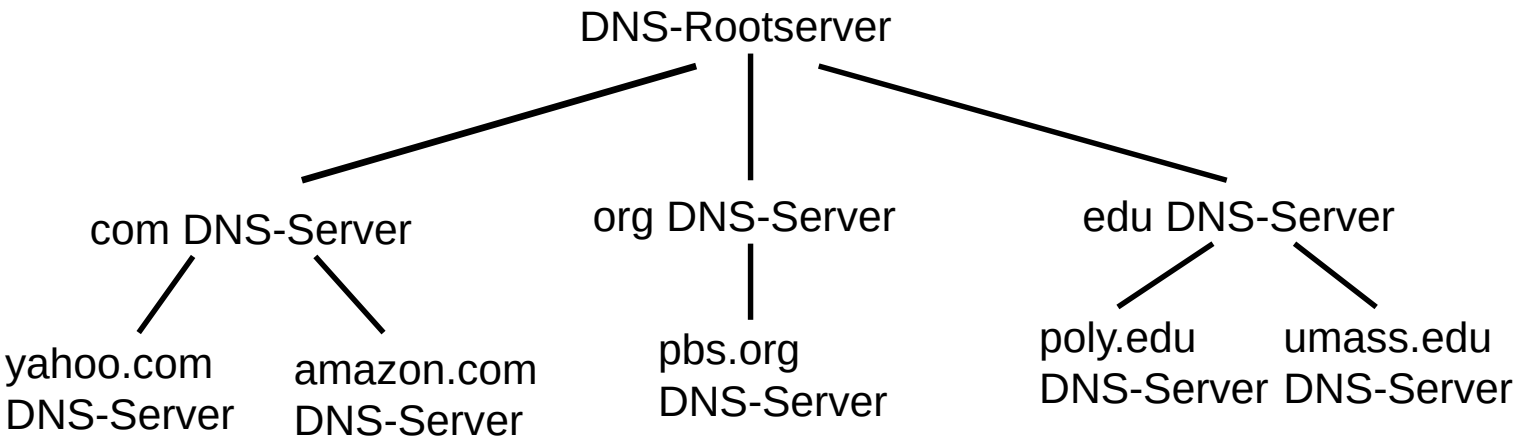


Domain Name System (DNS)

- Domain Name System:
  - Verteilte Datenbank
    - implementiert eine Hierarchie von Nameservern
  - Protokoll der Anwendungsschicht
    - wird von Hosts verwendet, um Namen aufzulösen (Abbildung zwischen Adresse und Name)
      - www.htwg-konstanz.de -> 141.37.20.17
    - bietet auch Auflösung nach Aliasen (anderen Namen)
    - bietet auch Auflösung nach Diensten, z.B. Mail-Exchange-Server
- Expliziter Aufruf
  - nslookup <hostname>



DNS: verteilte hierarchische Datenbank

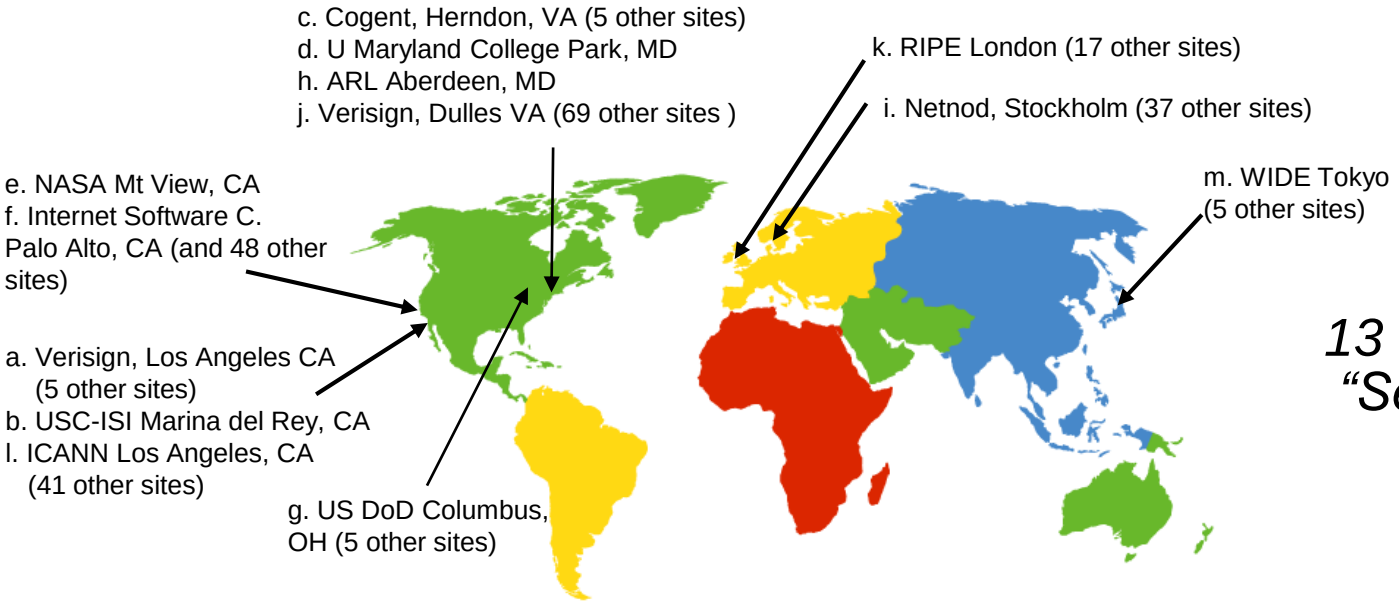


- Eigenschaften
  - hierarchische Namensstruktur unterteilt in Zonen
  - verteilt über mehrere DNS Servers
- Hierarchie der DNS Server
  - Root Server
  - Top-level Domain (TLD) Server
  - Authoritative DNS Server

- Wer gibt die Antwort?
  - Lokale DNS Server und “Client Resolver”

DNS Root Name Servers

- Es gibt 13 (A-M) Root-Server Domains (Namen)
  - jede Domain wird von einem Unternehmen betrieben
  - der Dienst einer Root-Server-Domain wird von zahlreichen Servern umgesetzt
    - Karte: root-server.org



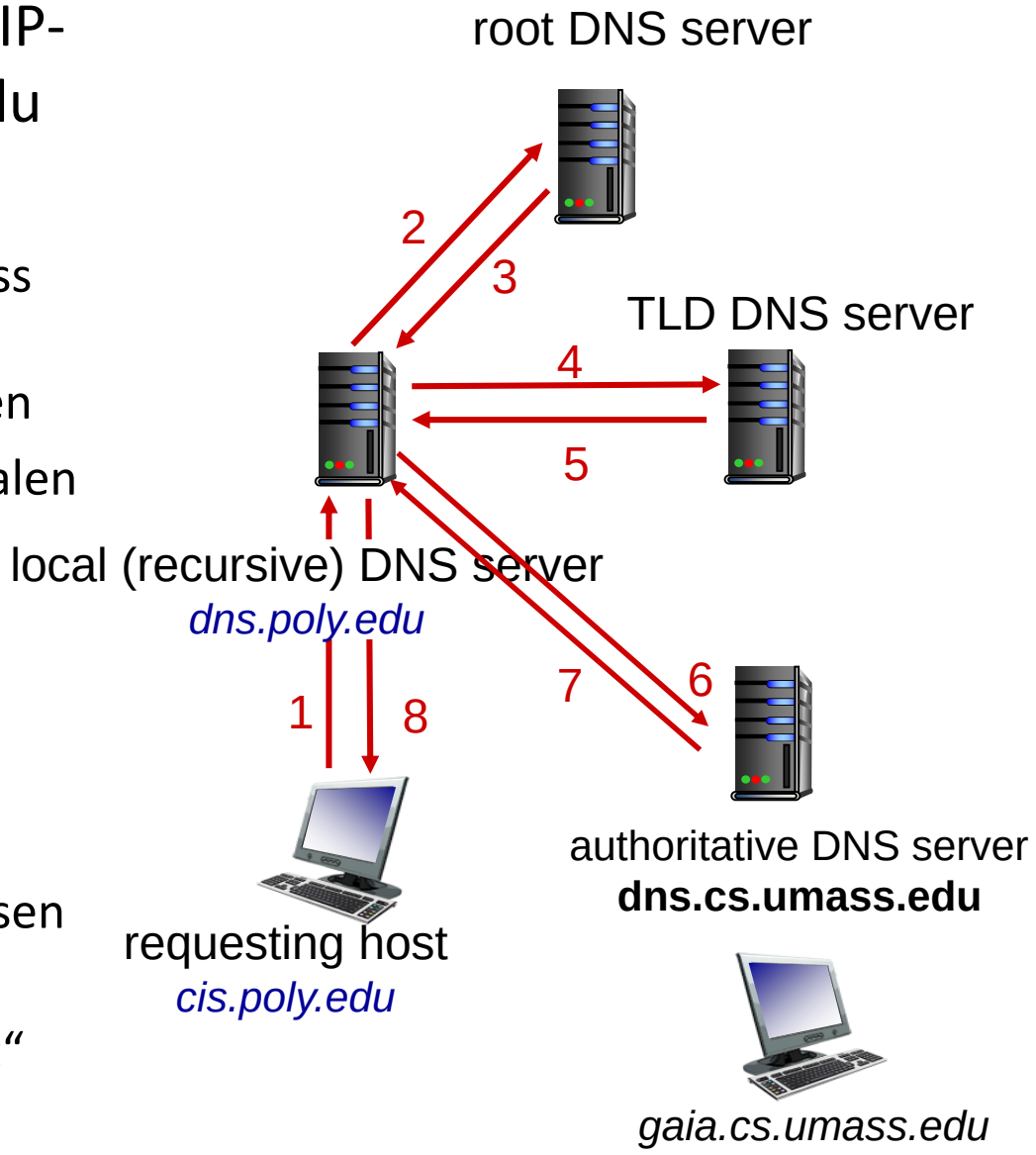
13 Root Name "Server" weltweit

TLD und Autoritativer Server

- Top-Level-Domain (TLD)-Server:
  - verantwortlich für com, org, net, edu etc. sowie für alle Länder-Domains, z.B. de, uk, fr, ca, jp
  - „Network Solutions“ ist verantwortlich für den „com“ TLD-Server
  - „Denic“ ist verantwortlich für den „de“ TLD-Server
- Autoritativer DNS-Server:
  - DNS-Server einer Organisation, der eine autorisierte Abbildung der Namen dieser Organisation auf IP-Adressen anbietet
    - gespeichert in einem Zone-File als Resource Records
  - Verwaltet von der entsprechenden Organisation oder einem Service Provider
- Alternative DNS Server:
  - Google bietet mit Public Google DNS (8.8.8.8 und 8.8.4.4) eine eigene öffentliche DNS Infrastruktur an
  - es gibt auch weitere DNS Infrastrukturen, beispielsweise ist mit Quad9 (9.9.9.9) eine "datenschutzfreundliche" Alternative zu Public Google DNS gestartet worden
  - alternative DNS Server werden von ISPs genutzt, die keinen eigenen lokalen DNS Server anbieten oder können von Privatpersonen direkt konfiguriert werden

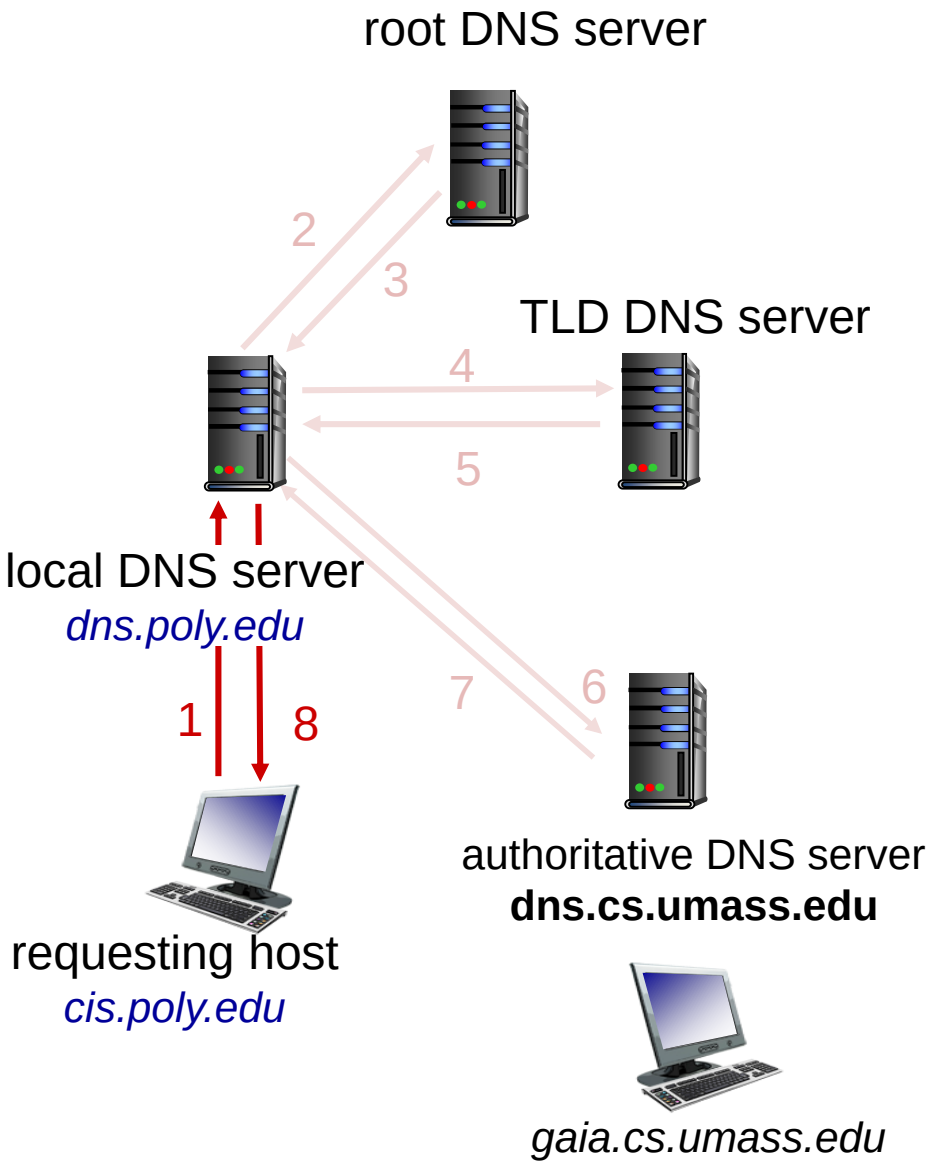
DNS: Iterativer Ablauf einer Anfrage (Query)

- Host an cis.poly.edu erfragt IP-Adresse von gaia.cs.mass.edu
- Rekursive Anfragen:
  - angefragter Name-Server muss Anfrage auflösen und mit IP Adresse oder Fehler antworten
  - Anfrage des Hosts an den lokalen DNS Server
- Iterative Anfragen:
  - angefragter Name-Server antwortet mit nächstem zuständigem Name-Server
  - „Keine Ahnung, frag doch diesen Server“
  - lokaler Name-Server „hangelt“ sich durch die Hierarchie



DNS Caching

- Latenz der DNS Anfragen
  - z.B. 1s Latenz vor dem Starten eines Download
  - Cachen verringert Overhead und Verzögerung
  - kleine Anzahl von Top-level Servern, die sich selten ändern
  - Populäre Seiten werden oft besucht
- Wo DNS Einträge cachen?
  - Lokaler DNS Server
  - Betriebssystem
    - ipconfig /displaydns
    - ipconfig /flushdns
  - Browser
- Konsistenz der gecachten Einträge
  - Time-to-live (TTL)
  - problematisch bei Änderung einer IP-Adresse während TTL



Namen Auflösen mit DNS

- Lokaler (rekursiver) DNS Server (“Default Name Server”)
  - gehört nicht in die DNS-Server-Hierarchie
  - typischerweise in der Nähe des Hosts, der eine Anfrage stellt
  - entweder auf dem Host konfiguriert oder über DHCP (Kapitel 4)
- Anwendung auf dem Client
  - extrahiert den Server-Namen z.B. von der URL
  - nutzt Befehle *gethostbyname()* oder *getaddrinfo()* um die Adresse zu erhalten
- Anwendung auf dem Server
  - extrahiert die IP Adresse des Clients vom Socket
  - kann *gethostbyaddr()* nutzen, um den Host-Namen des Clients festzustellen
- Proxy:
  - extrahiert Server-Namen aus GET-Request und löst diesen auf
- NSLOOKUP
  - Windows: nslookup in cmd
  - im Internet: network-tools.com/nslook/
- Alternative:
  - Google und andere Firmen stellen öffentliche DNS Server zur Verfügung
  - Ziel: aktuellere Einträge, kürzere Delays

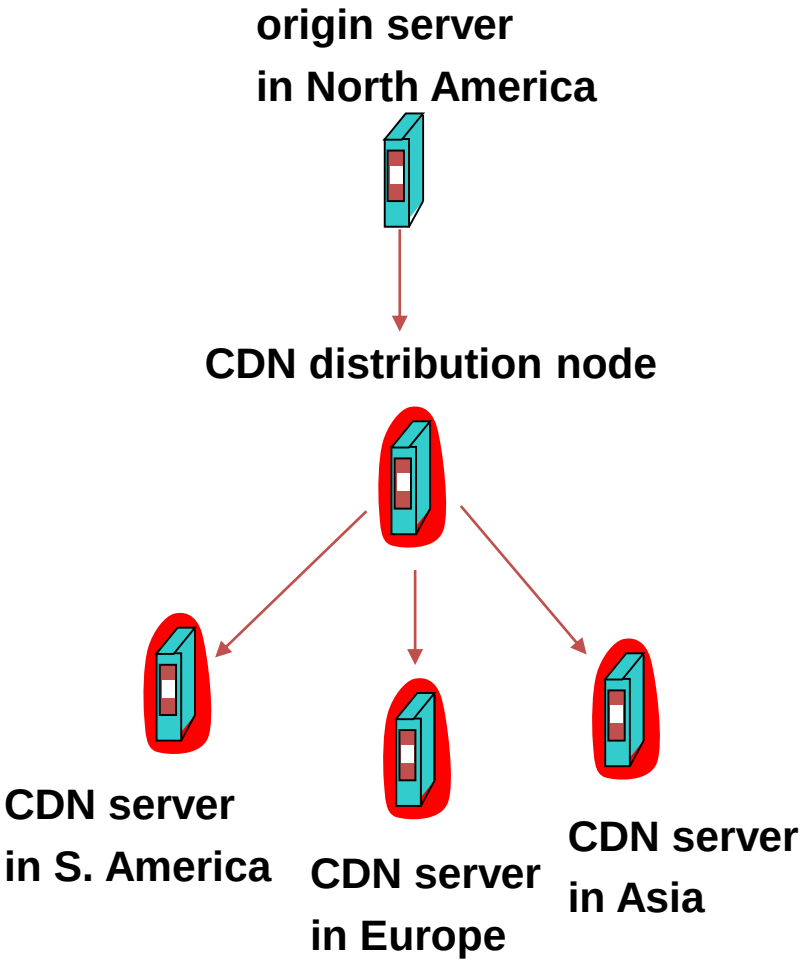
DNS Security und Privacy

- DNS ist einer der wichtigsten Angriffsvektoren, über den zahlreiche Angriffe gefahren werden
  - DNS Spoofing oder HiJacking: Übernahme oder Manipulation eines DNS Resolvers (lokaler DNS Server), um Hostnamen auf IP-Adressessen des Angreifers abzubilden, z.B. einer Phishing-Seite
    - https mit Zertifikat hilft
    - ABER: Fallback auf http wird nicht erkannt, Zertifikatfehler wird ignoriert
  - Denial-of-Service (DOS) Angriffe auf DNS-Server
  - DNS Tunneling: Nutzen des DNS-Protokolls, um Daten durch eine Firewall zu schleußen. DNS-Verkehr ist kritisch und wird oft nicht (ausreichend) überwacht.
- DNS Privacy
  - der Betreiber des lokalen DNS Servers weiß, welche Web-Seiten Sie laden
  - wenn DNS nicht verschlüsselt ist, kann das auch jeder herausfinden, der im Netz mithört oder in der Lage ist einen Fake-DNS-Server zu betreiben
- DNS Maßnahmen
  - DNSSEC: überträgt signierte DNS Records
  - DNS über HTTPS: Verschlüsselung von DNS Anfragen, normales DNS ist unverschlüsselt und erlaub Man-in-the-Middle Attacken
  - Redundante Infrastruktur oder DNS Firewall gegen Denial-of-Service Attacken
  - Konfigurieren eines vertrauenswürdigen DNS Servers
    - Vielleicht 9.9.9.9? Aber wie sichergehen?

Kapitel 3: Anwendungsschicht

Content Distribution Network (CDN)

- Proaktive Content Replikation
  - Content-Provider (e.g., CNN) hat einen Vertrag mit einem CDN
- CDN repliziert Content
  - auf vielen Serverfarmen in den verschiedenen Regionen des Internets
- Updaten der Replikas
  - Updates werden gepusht, wenn sich der Inhalt ändert (proaktiv)
- Unterschied Cache / CDN
  - Cache: reaktiv
  - CDN: proaktiv



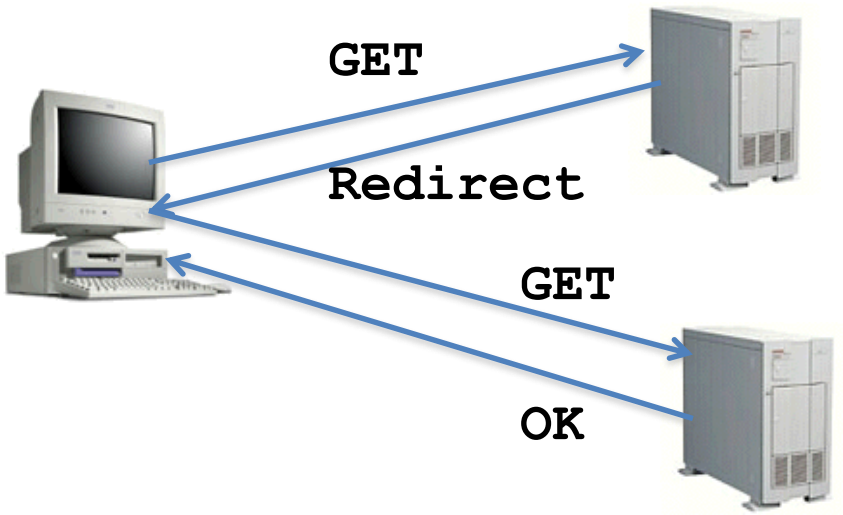
Strategien und Verfahren zur Auswahl des Servers

- Ziele:
  - hohe Verfügbarkeit, Load-Balancing, Performance, Kostenreduktion
- Strategien:
  - Live-Server, niedrigste Last, nächster Server, günstigste Serverfarm
- Mechanismen:
  - HTTP redirect
  - DNS basierte Serverauswahl

- DNS wird von CDNs genutzt,
  - um Lokalität zu erzielen, indem weltweiter Host-Name wird auf IP-Adresse des besten lokalen Servers abgebildet
    - Beobachtung aus dem Labor: die Ping-Zeiten vieler Web-Seiten liegen bei 10-20ms. Der Hostame wird von DNS auf die IP-Adresse eines lokalen CDN-Datacenters abgebildet
  - um Load Balancing (Lastausgleich) zwischen Servern zu erzielen

HTTP Redirection

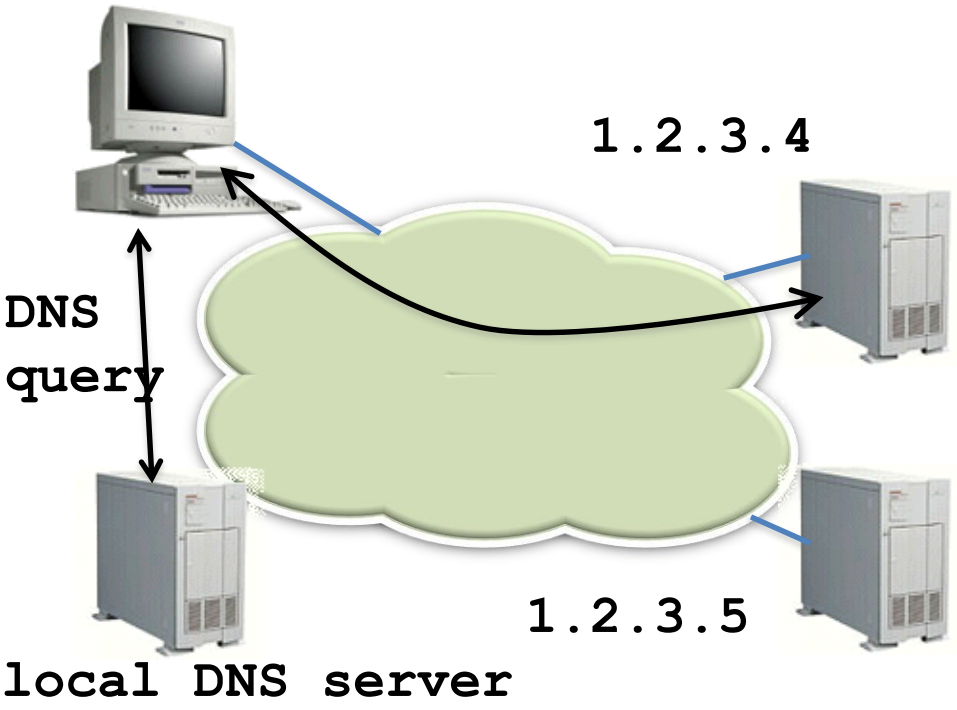
- Prinzip
  - Server antwortet mit HTTP Redirect Statuscode (3xx) und teilt die Adresse mit, unter der die Ressource zu finden ist



- Vorteil
  - präzise Kontrolle
  - Serverwahl nach Client-IP
- Nachteil
  - zusätzliche RTTs für TCP Verbindungsaufbau
  - Overhead für den Server

DNS-basierte Serverauswahl

- Prinzip:
  - DNS liefert den vom CDN bevorzugten Server

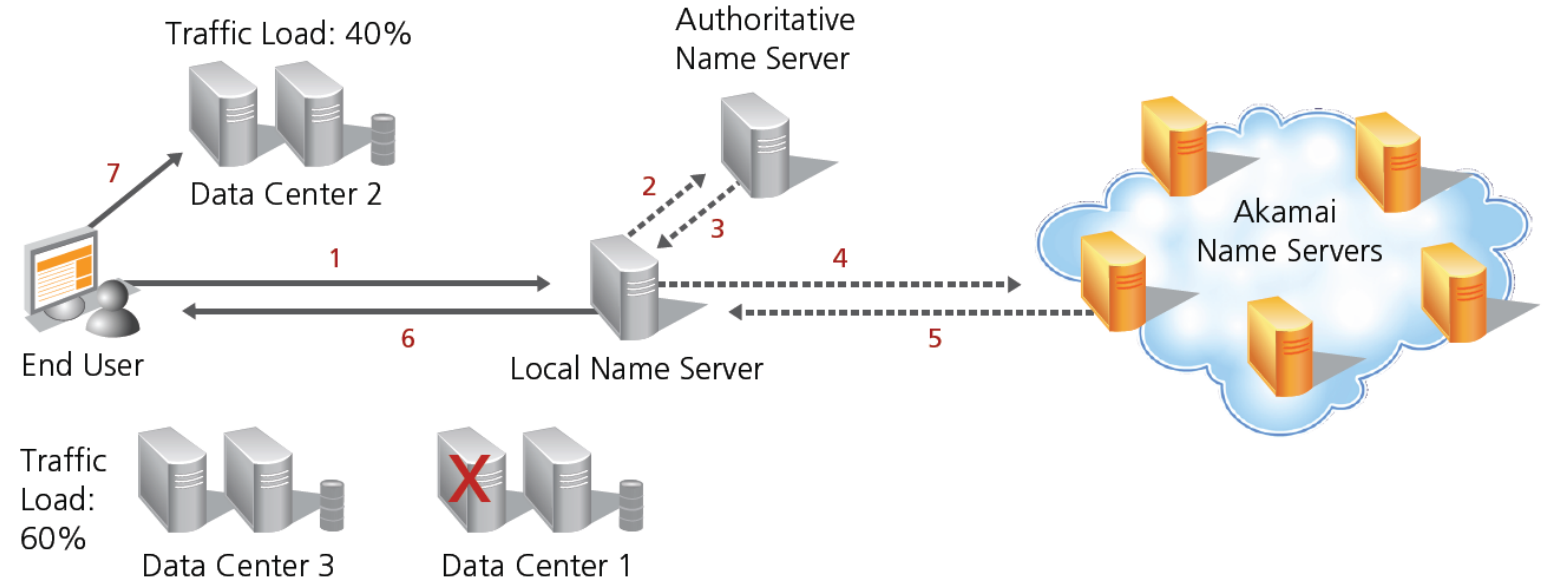


- Vorteil
  - keine zusätzlichen TCP-Verbindungen
  - DNS Caching reduziert Overhead
- Nachteil
  - Auswahl aufgrund der IP des lokalen DNS Servers
  - kleine TTL notwendig

Akamai

- Servers
  - Servers: ~100,000
  - Networks: ~1,000
  - Countries: ~70
- Kunden
  - Apple, BBC, FOX, GM IBM, MTV, NASA, NBC, NFL, NPR, Puma, Red Bull, Rutgers, SAP, …
- Anfragen
  - Milliarden pro Tag
  - 50% in den Top 45 Netzen
  - 15-20% des weltweiten Netzverkehrs

Akamai



- DNS Eintrag im autoritativen Name Server enthält nur ein Alias, keine IP-Adresse
- Lokaler Name Server kontaktiert Akamai Name Server, um die IP-Adresse zu erhalten
- Akamai entscheidet aufgrund interner Strategie, welcher Web-Server der geschickteste ist
  - Akamai betreibt auch eine Hierarchie von DNS Servers

Zusammenfassung DNS

- mit DNS werden Hostnamen auf IP-Adressen abgebildet
- der lokale (rekursive) DNS Server stellt die Anfrage, die der autoritative DNS Server für seine Domäne beantwortet
  - der lokale DNS Server findet den richtigen autoritativen DNS Server über die DNS Hierarchie
- DNS Anfragen werden von Browser, Betriebssystem und lokalem DNS Server gecacht
- DNS wird von CDNs genutzt, um Anfragen zu lenken
  - zu lokalen, wenig ausgelasteten oder aus sonstigen Gründen favorisierten Datenzentren/Servern
- DNS ist kritisch hinsichtlich Privacy und Sicherheit
  - Augen auf bei der Auswahl des DNS Servers
  - https://avoidthehack.com/best-dns-privacy (nicht geprüft)