

Systeme II

6. Die Anwendungsschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

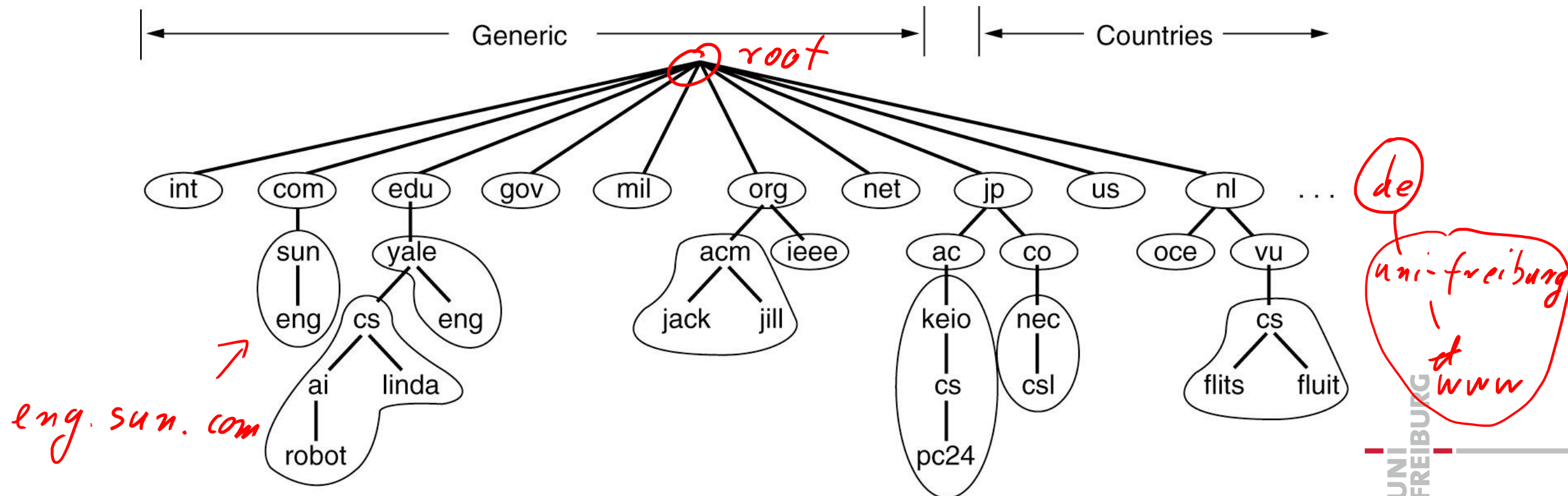
Albert-Ludwigs-Universität Freiburg

Version 06.07.2017

- Menschen kommen mit den 4-Byte IPv4-Adressen nicht zurecht:
 - 209.85.148.102 für Google
 - 132.230.2.100 für Uni Freiburg
 - Was bedeuten?
 - 77.87.229.75
 - 132.230.150.170
- Besser: Natürliche Wörter für IP-Adressen
 - Z.B. www.get-free-beer.de
 - oder www.uni-freiburg.de
- Das Domain Name System (DNS) übersetzt solche Adressen in IP-Adressen

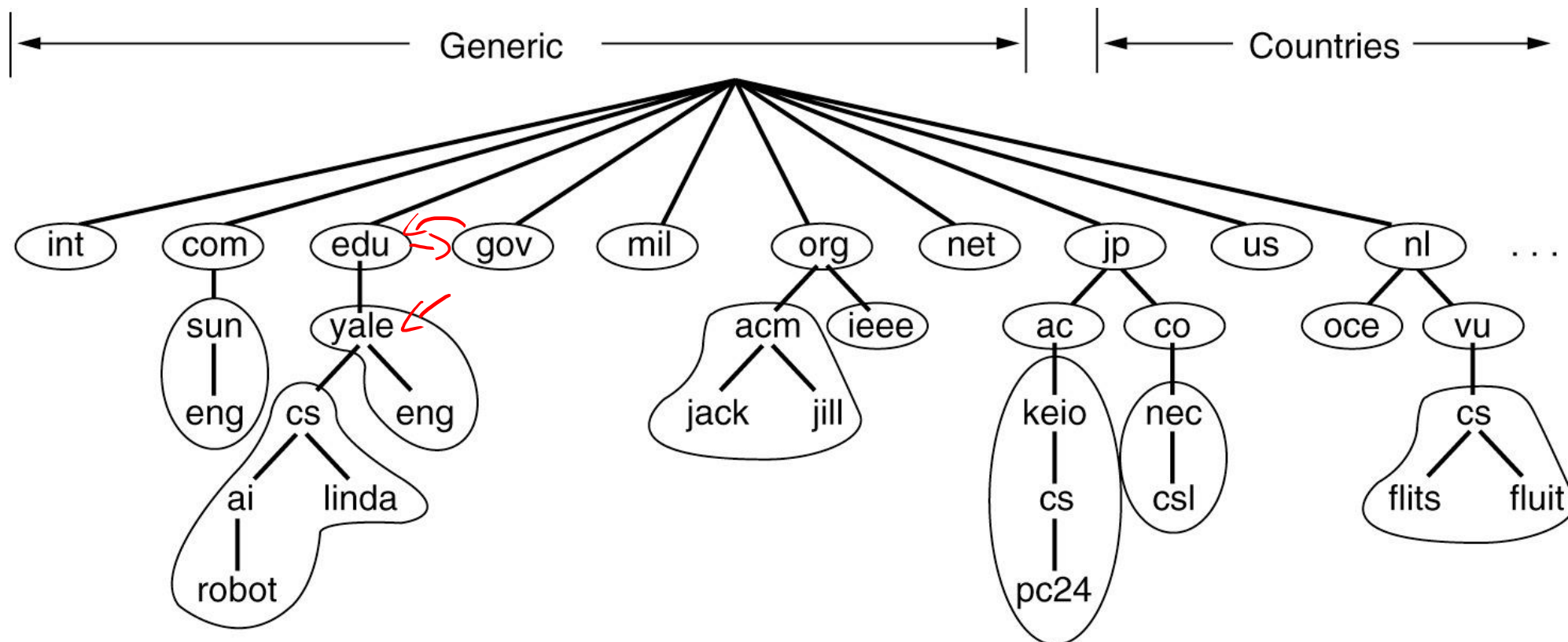
DNS – Architektur

- DNS bildet Namen auf Adressen ab
 - Eigentlich: Namen auf Ressourcen-Einträge
- Namen sind hierarchisch strukturiert in einen Namensraum
 - Max. 63 Zeichen pro Komponente, insgesamt 255 Zeichen
 - In jeder Domain kontrolliert der Domain-Besitzer den Namensraum darunter
- Die Abbildung geschieht durch Name-Server

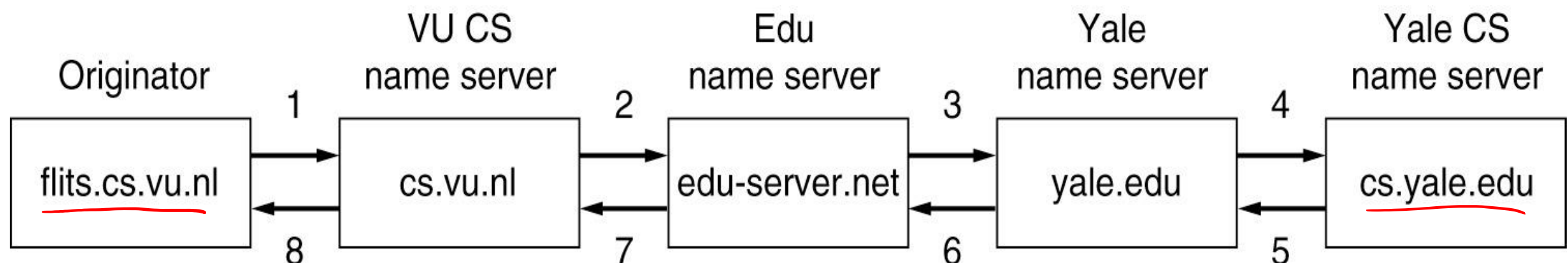


DNS Name Server

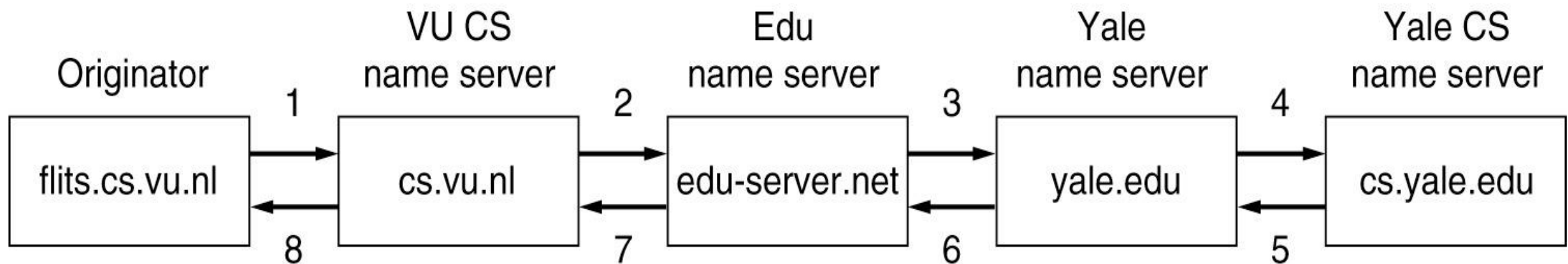
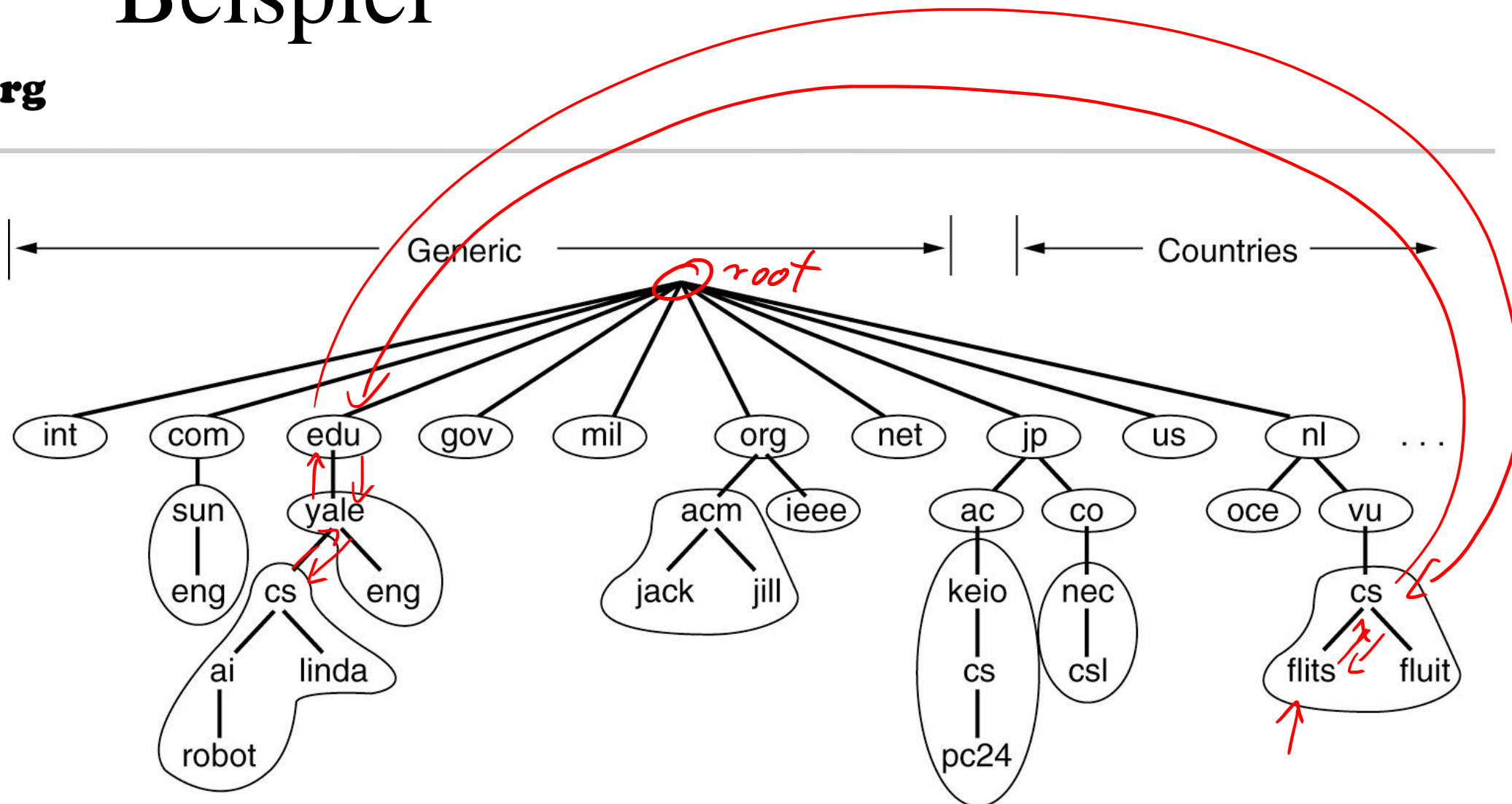
- Der Namensraum ist in Zonen aufgeteilt
- Jede Zone hat einen *Primary Name Server* mit maßgeblicher Information
 - Zusätzlich *Secondary Name Server* für Zuverlässigkeit
- Jeder Name Server kennt
 - seine eigene Zone
 - Name-Server der darunterliegenden Bereiche
 - Bruder-Name-Server oder zumindestens einen Server, der diese kennt



- Anfragen von einem End-System werden zu den vorkonfigurierten Name-Server geschickt
 - Soweit möglich, antwortet dieser Name-Server
 - Falls nicht, wird die Anfrage zu dem bestgeeigneten Name-Server weitergereicht
 - Die Antworten werden durch die Zwischen-Server zurückgeschickt
- Server darf Antworten speichern (cachen)
 - Aber nur für eine bestimmte Zeit

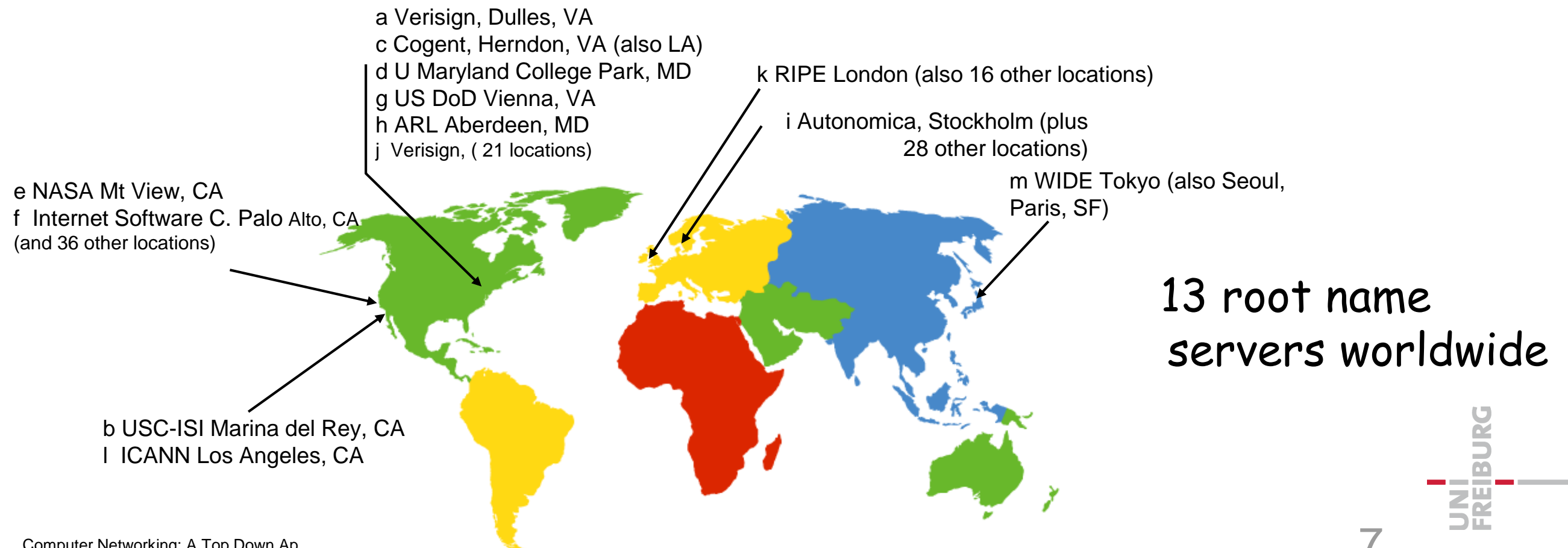


Beispiel



DNS Root Name Servers

- wird von lokalen Name-Server kontaktiert, wenn der Name nicht aufgelöst werden kann
- Root Name Server:
 - wird kontaktiert vom Name-Server falls die Zuordnung der Namen nicht bekannt ist.
 - erhält die Zuordnung
 - gibt die Zuordnung an den lokalen Name-Server weiter



TLD und autorisierte Server

■ Top-Level Domain (TLD) Server

- verantwortlich für com, org, net, edu, etc, und alle Top-Level-Country-Domains uk, fr, ca, jp. *de*

- Network Solutions unterhält Server für *com* TLD

- Educause für *edu* TLD

DeNIC verwaltet .de

■ Autorisierte DNS Servers:

- DNS-Server von Organisationen

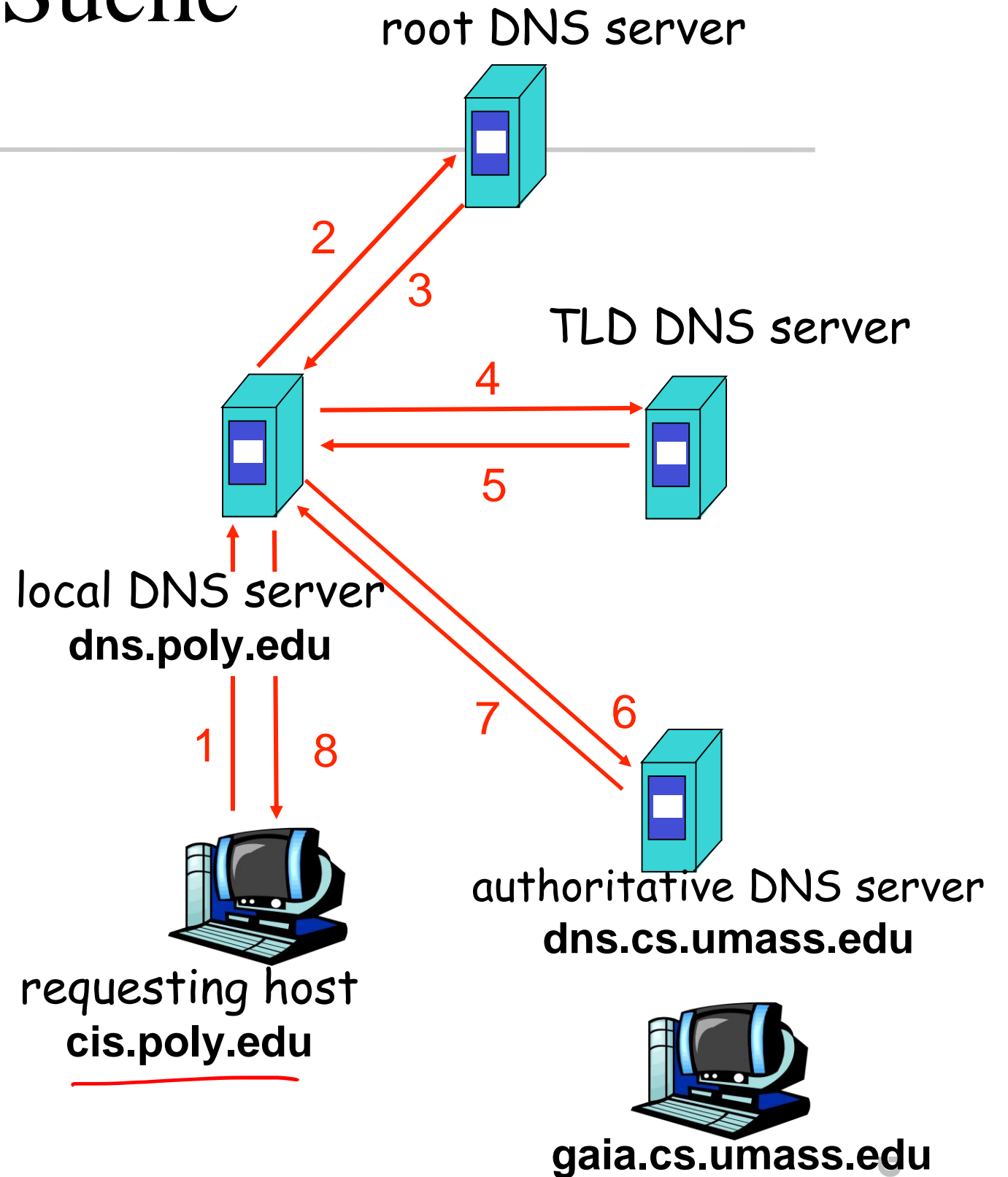
- welche verantwortlich für die Zuordnung von IP-Adresse zu Hostnamen sind

- können von den Organisationen oder Service-Provider unterhalten werden

- Jeder ISP hat einen lokalen Name-Server
 - Default Name Server
- Jede DNS-Anfrage wird zum lokalen Name-Server geschickt
 - fungiert als Proxy und leitet Anfragen in die Hierarchie weiter

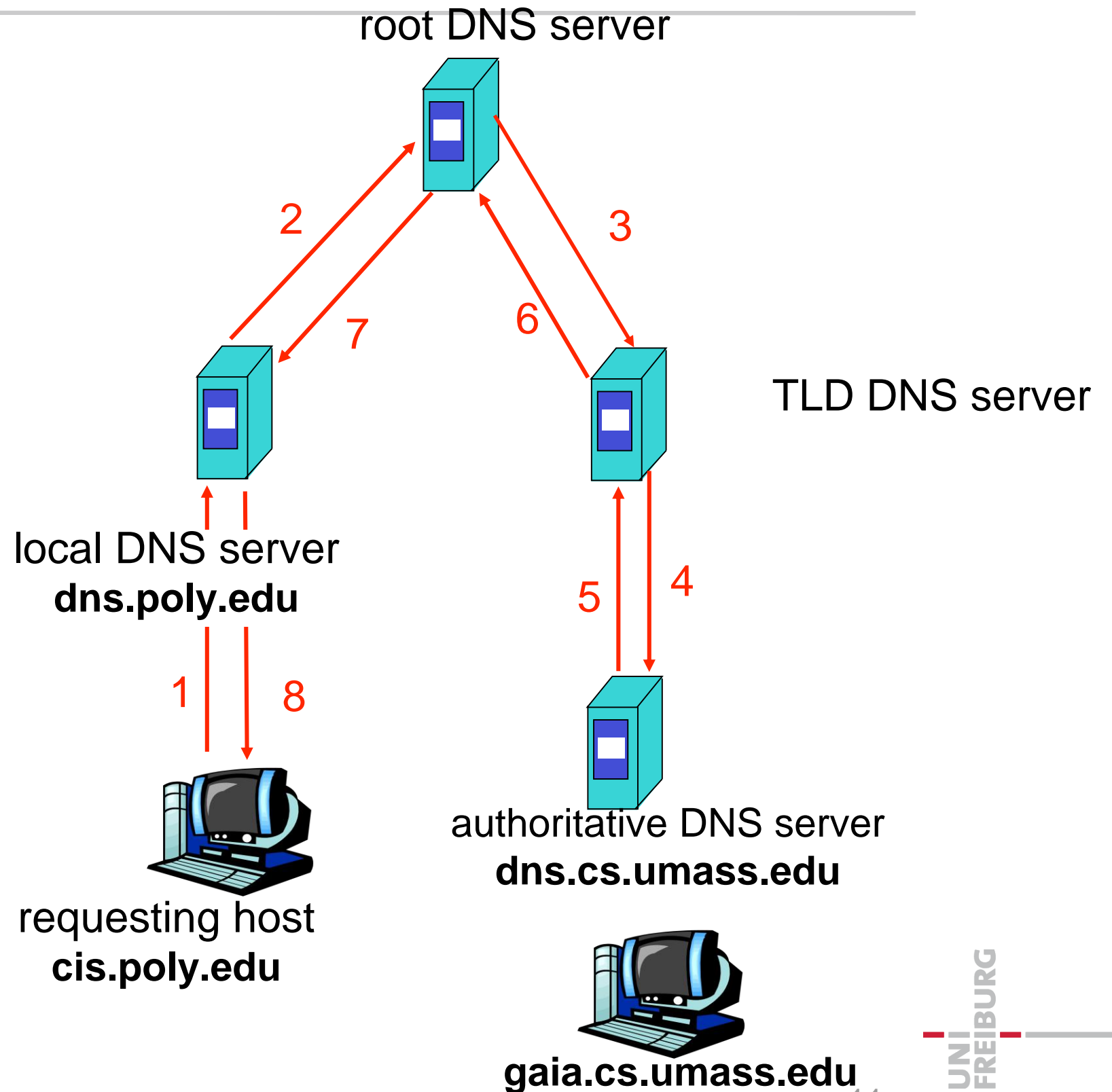
DNS Iterative Suche

- Rechner bei cis.poly.edu fragt nach IP address für gaia.cs.umass.edu
- Iterative Anfrage
 - Angefragte Server antworten
 - mit IP-Adresse
 - oder mit dem Namen des nächsten Servers
 - Lokaler DNS-Server ist selbst für Suche verantwortlich



DNS Rekursive Suche

- Jeder angefragte Server ist für die Namensauflösung zuständig
- Anfrage wird rekursiv weitergeleitet und dann zurück gegeben



- Sobald ein Name-Server einen Namen kennen lernt, speichert er die Zuordnung
 - Cache-Einträge haben einen Time-Out und werden nach einer gewissen Zeit gelöscht
 - TLD-Servers werden in lokalen Name-Servern gespeichert
 - Daher werden Root-Name-Server nicht oft besucht
- Update und Benachrichtungsmechanismus von IETF festgelegt
 - RFC 2136
 - <http://www.ietf.org/html.charters/dnsind-charter.html>

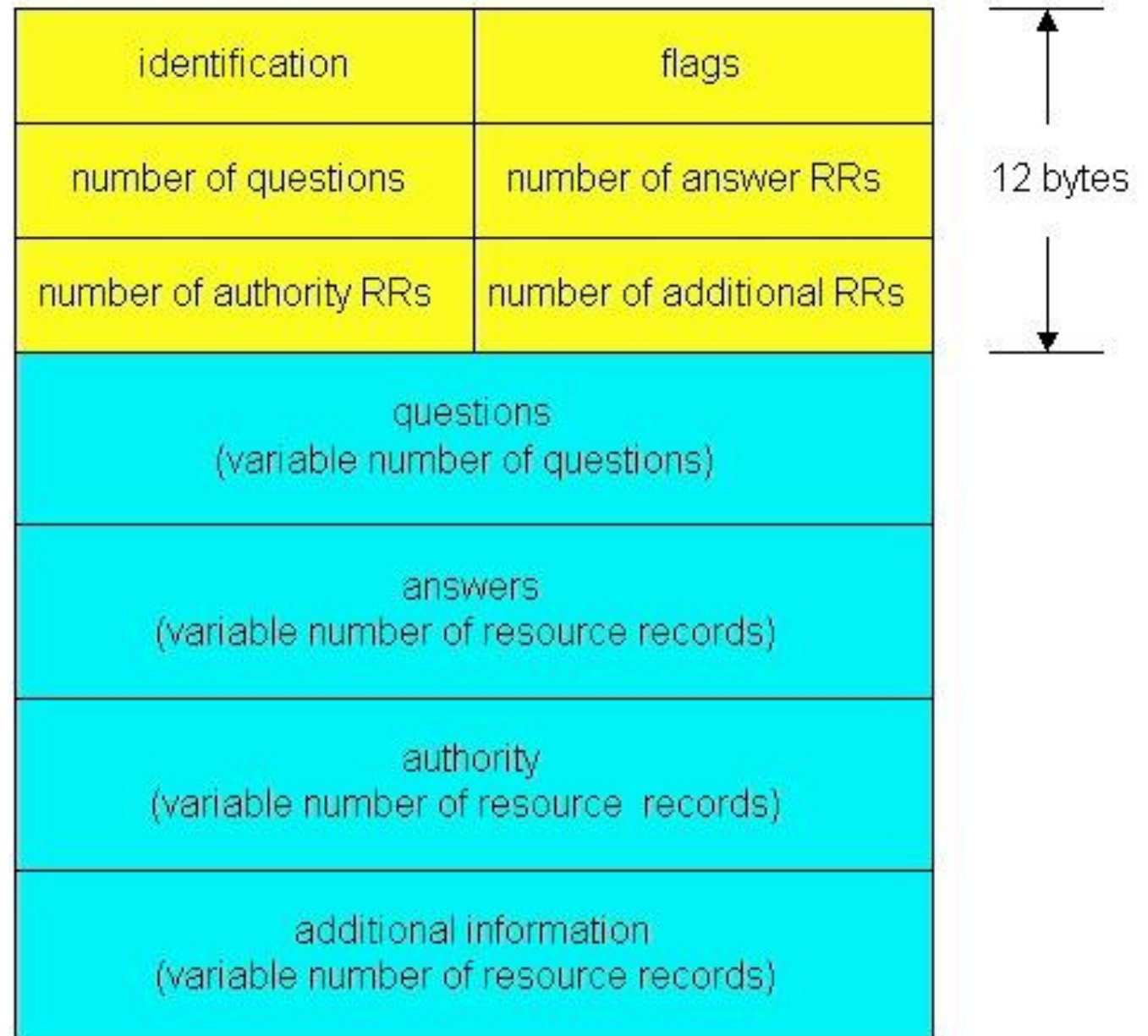
- DNS: verteilte Datenbank speichert Resource Records (RR)
- RR Format: (Name, Wert, Typ, TTL)
- Typ = A
 - Name = hostname
 - Wert = IP-Adresse
- Typ = NS
 - Name = domain (z.B. uni-freiburg.de)
 - Wert = hostname eines autorisierten Name-Servers für diese Domain
- Typ = CNAME
 - Name = Alias für einen „kanonischen“ (wirklichen) Namen
 - z.B. www.ibm.com ist in Wirklichkeit servereast.backup2.ibm.com
 - Wert ist kanonischer Name
- Typ = MX
 - Wert ist der Name des Mailservers

DNS Resource Record

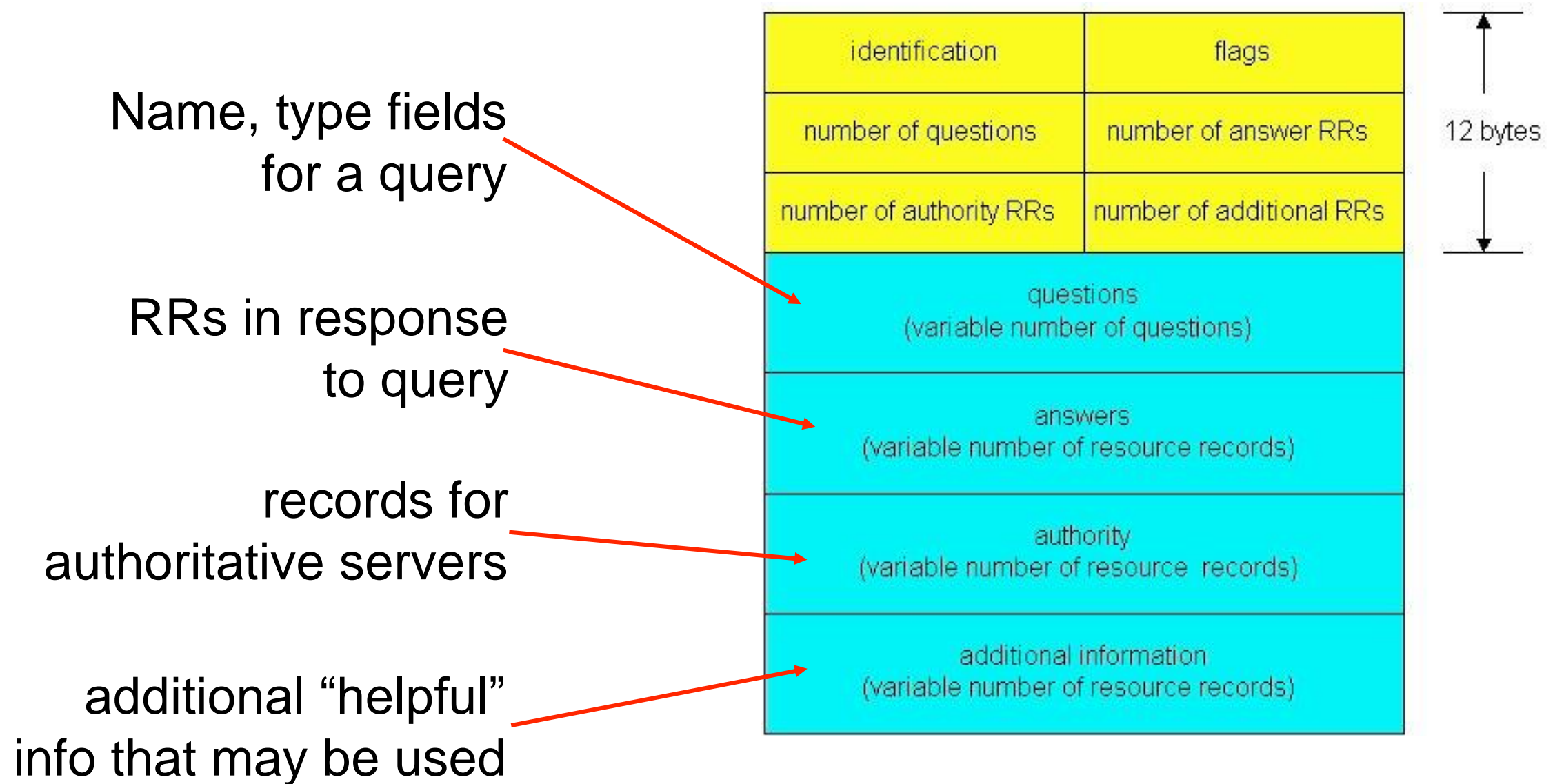
- Ressourcen-Einträge: Informationen über Domains, einzelne Hosts,...
- Inhalt:
 - Domain_name: Domain(s) des Eintrags
 - Time_to_live: Gültigkeit (in Sekunden)
 - Class: Im Internet immer "IN"
 - Type: Siehe Tabelle
 - Value: z.B. IP-Adresse

Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text

- DNS-Protokoll
 - Anfrage und Antwort im selben Format
- Nachrichten-Header
 - ID, 16 Bit für Anzahl der Anfragen, Anzahl der Antworten, ...
- Flags:
 - Query oder Reply
 - Rekursion gewünscht
 - Rekursion verfügbar
 - Antwort ist autorisiert



DNS-Protokoll und Nachrichten



- Problem
 - Zeitlich zugewiesene IP-Adressen
 - z.B. durch DHCP
- Dynamisches DNS
 - Sobald ein Knoten eine neue IP-Adresse erhält, registriert dieser diese beim DNS-Server, der für diesen Namen zuständig ist
 - Kurze time-to-live-Einträge sorgen für eine zeitnahe Anpassung
 - da sonst bei Abwesenheit die Anfragen an falsche Rechner weitergeleitet werden
- Anwendung
 - Registrierung einer Domain für den Otto Normalverbraucher
 - Siehe www.dyndns.com

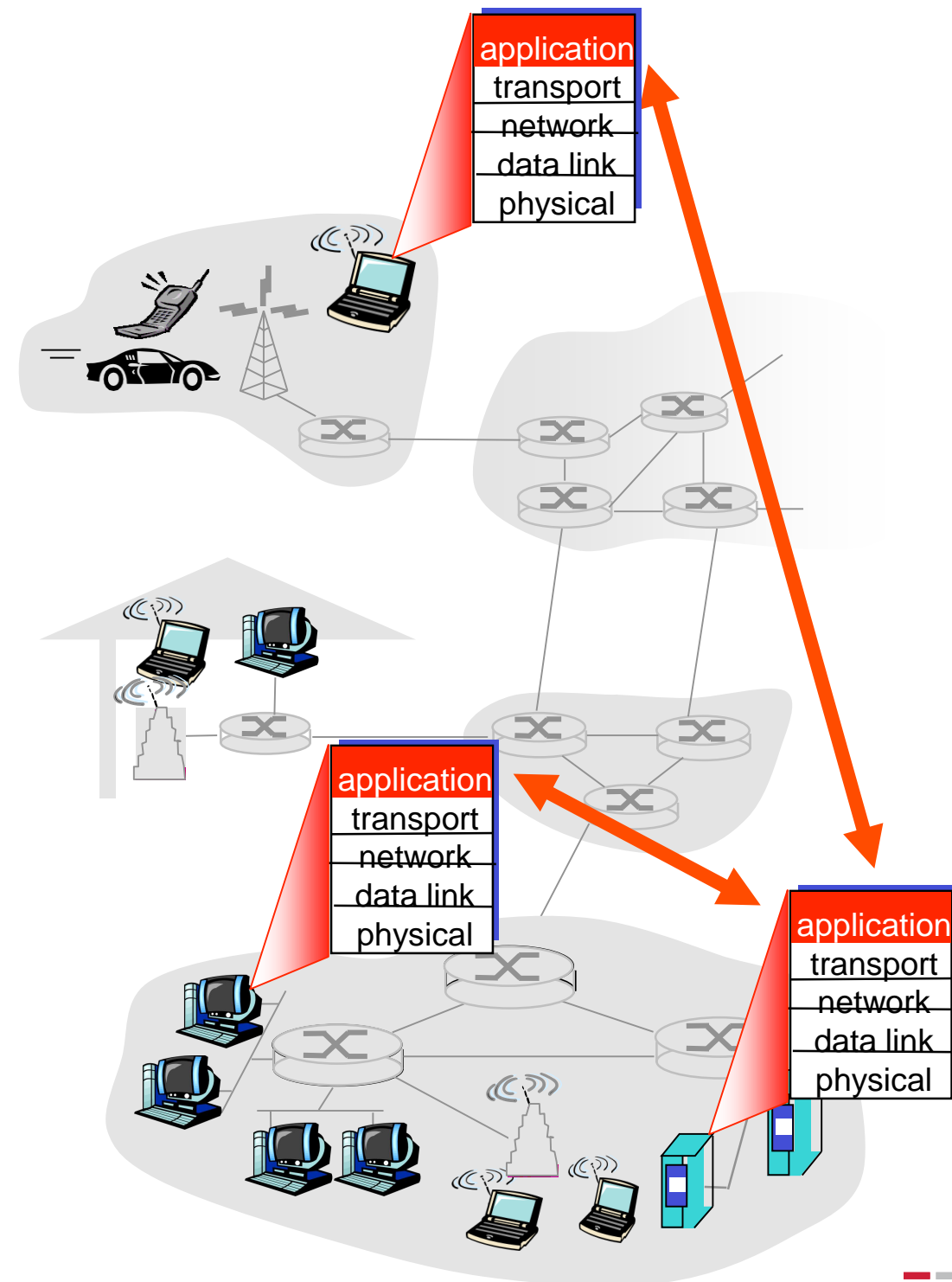
- Cache Poisoning
 - Falsche Einträge werden in DNS-Server eingebracht
 - weitergeleitete Einträge werden gecacht und können zu falschen Auskünften führen
- DNSSEC
 - implementiert seit 2010
 - Zuständiger Master-Server unterschreibt seine Einträge digital (mit Hilfe eines Public-Key-Kryptosystems)
 - Ursprüngliche Information bleibt unverschlüsselt
- Schlüsselmanagement
 - Gegenseitiges unterschreiben der Public-Keys
 - Aufwand wird gemildert durch „Chain of Trust“
 - Hierarchische Kette von unterschriebenen Schlüsseln
- Diskussion
 - aufwändigere DNS-Antworten
 - Sicherheitslücken bleiben bestehen

- Aspekte der Programmierung im Internet aus der Sicht der Anwendung
- Anforderungen an die Transportschicht
- Client-Server-Prinzip
- Peer-to-Peer-Prinzip
- Beispiel-Protokolle:
 - HTTP
 - SMTP / POP3 / IMAP
 - DNS
- Programmierung von Netzwerk-Anwendungen

- E-Mail
- Web
- Instant messaging
- Remote Login
- P2P File Sharing
- Multi-User Network Games
- Video Streaming
- Social Networks
- Voice over IP
- Real-time Video Konferenz
- Grid Computing

Erstellen einer Netzwerk-Anwendung

- Programme laufen auf den End-Punkten
 - kommunizieren über das Netzwerk
 - z.B. Web-Client kommuniziert durch Browser-Software
- Netzwerk-Router
 - werden nicht programmiert!
 - nicht für den Benutzer verfügbar
- Dadurch schnelle Programm-Entwicklung möglich
 - gleiche Umgebung
 - schnelle Verbreitung



- Client-server
 - beinhaltet auch Data Centers & Cloud Computing
- Peer-to-peer (P2P)
- Hybride Verbindung von Client-Server und P2P

Client-Server-Architektur

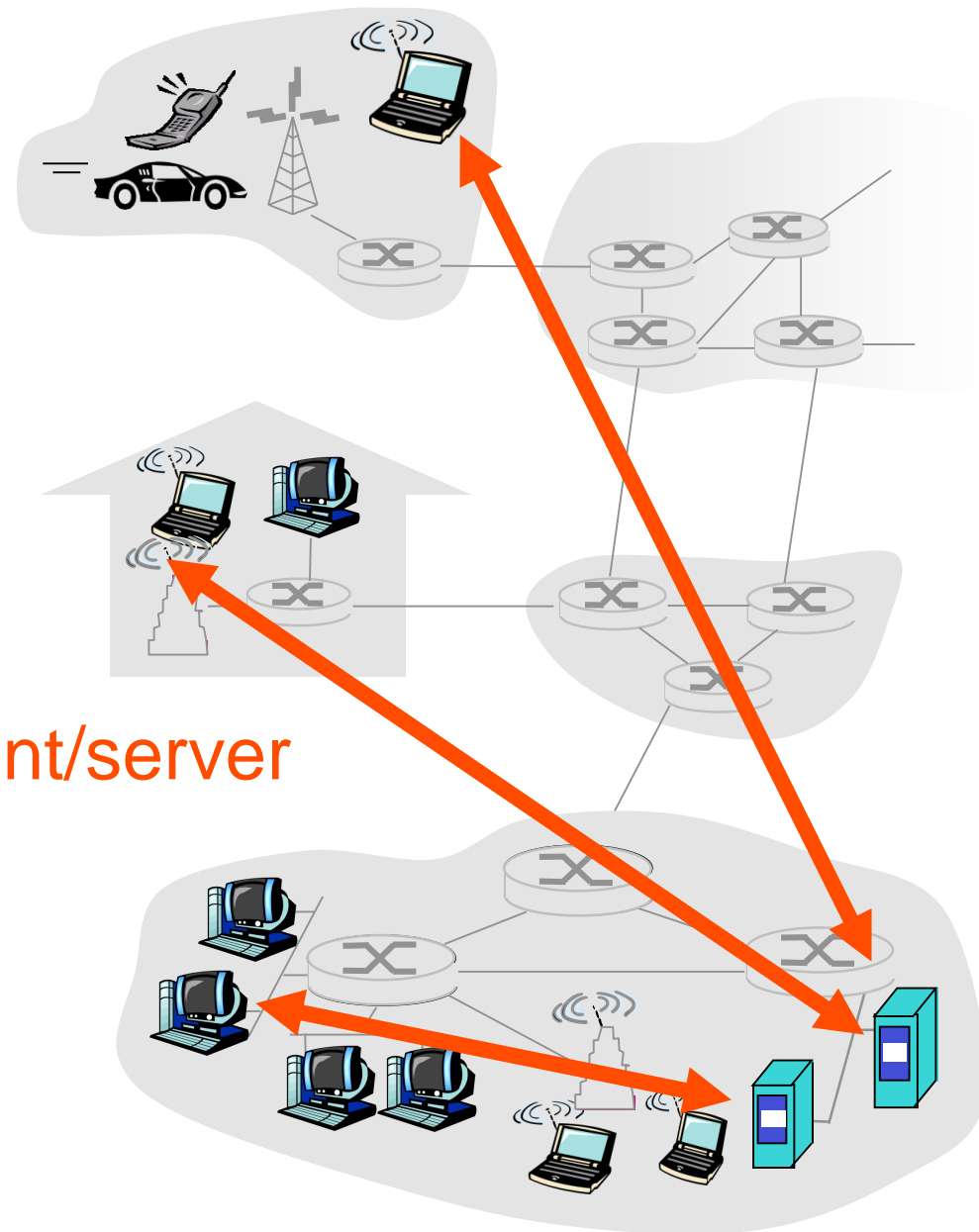
■ Server

- allzeit verfügbarer Host
- permanente IP-Address
 - oder per DNS ansprechbar
- Server-Farms wegen Skalierung

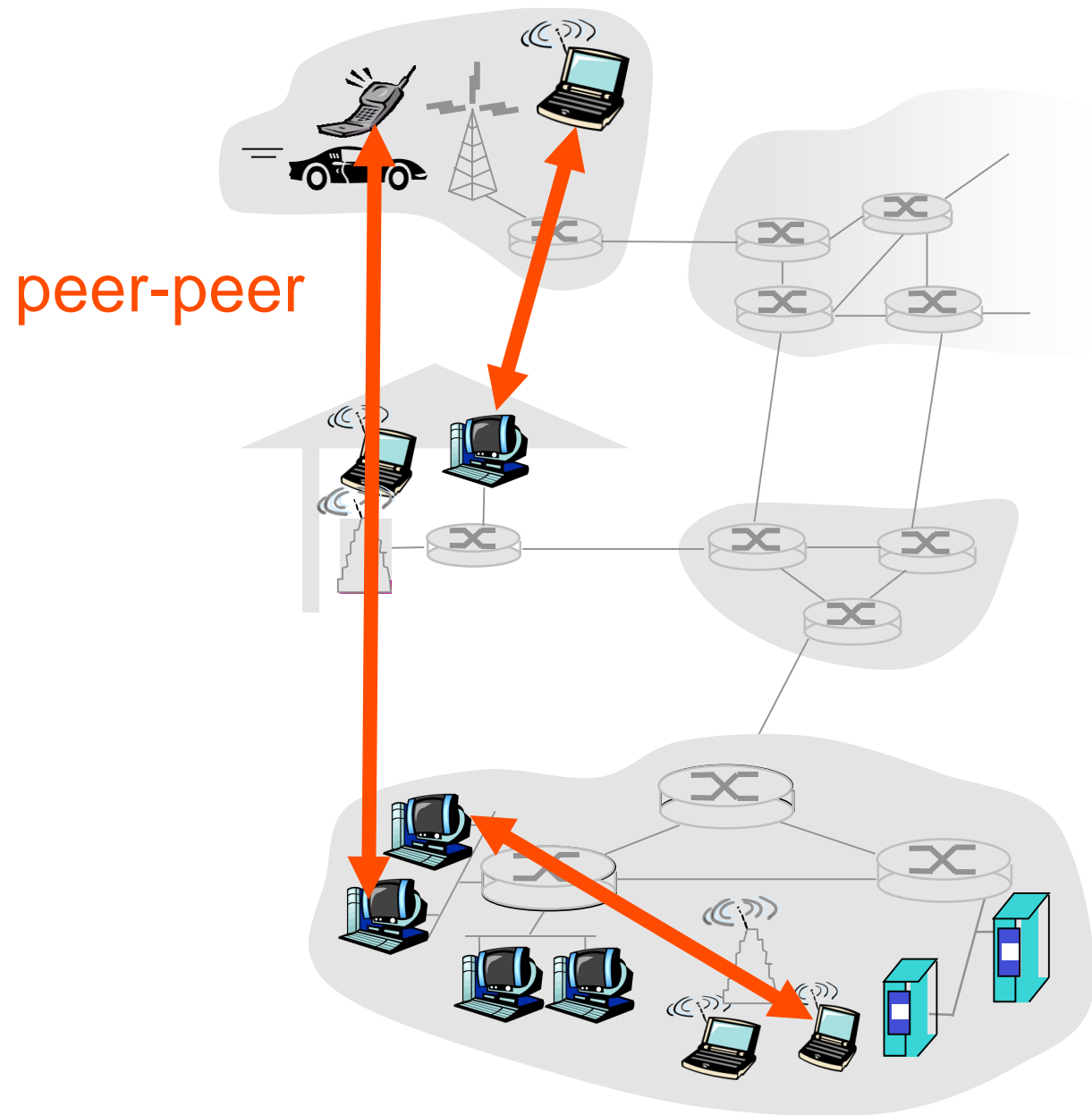
■ Client

- kommuniziert mit dem Server
- möglicherweise nicht durchgängig verbunden
- evtl. dynamische IP-Adresse
- Clients kommunizieren **nicht** miteinander

client/server



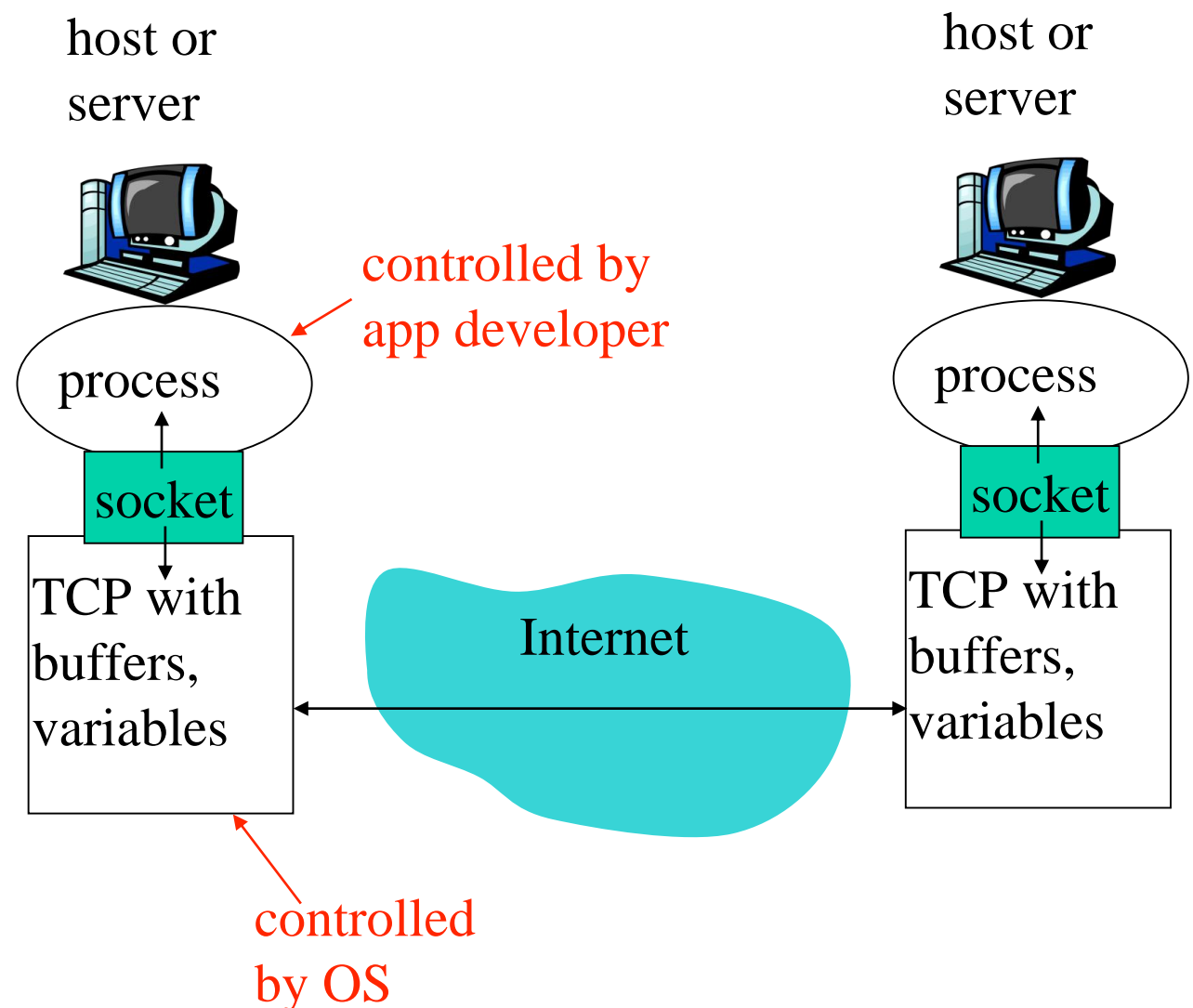
- Ohne Server
- End-Systeme kommunizieren direkt
- Peers
 - sind nur zeitlich begrenzt online
 - verändern von Zeit zu Zeit ihre IP-Adresse
- Hochskalierbar, aber schwer zu handhaben



- z.B. Skype
 - Voice-over-IP P2P
 - Server für Anmeldung und Verzeichnis
 - Telefonie und Video-Verbindung Direktverbindung
- Instant Messaging
 - Chat zwischen zwei Benutzern ist P2P
 - Zentraler Service:
 - Client-Anwesenheit
 - Suche und Zuordnung der IP-Adresse
 - Benutzer registrieren die IP-Adresse, sobald online
 - Benutzer fragen beim Server nach IP-Adresse der Partner

- Prozess: Programm auf einem Rechner (Host)
 - innerhalb des selben Rechners kommunizieren Prozesse durch Inter-Prozess-Kommunikation
 - über OS
- Prozesse in verschiedenen Rechnern
 - kommunizieren durch Nachrichten
- Client-Prozess
 - Initiiert die Kommunikation
- Server-Prozess
 - wartet auf Client-Kontakt
- P2P
 - haben Client und Server-Prozesse

- Prozesse senden und empfangen Nachrichten über Sockets (Steckdosen)
- Sockets mit Türen vergleichbar
- Sender-Prozess
 - schiebt die Nachricht zur Tür hinaus
 - vertraut auf die Transport-Infrastruktur, dass die eine Seite der Tür mit der anderen verbindet
- API
 - Wahl des Transport-Protokolls
 - kann bestimmte Parameter wählen



- Nachrichtentyp
 - z.B. Request, Response
- Nachrichten-Syntax
 - Nachrichtenfelder und Zuordnung
- Nachrichten-Semantik
 - Bedeutung der Felder
- Regeln für das Senden und Empfangen von Nachrichten
- Public-domain Protokolle
 - definiert in RFC
 - für Kompatibilität
 - z.B. HTTP, SMTP, BitTorrent
- Proprietäre Protokolle
 - z.B. Skype, ppstream

Welchen Transport-Service braucht eine Anwendung?

- **Datenverlust**
 - einige Anwendungen (z.B. Audio) tolerieren gewissen Verlust
 - andere (z.B. Dateitransfer, Telnet) benötigen 100% verlässlichen Datentransport
- **Timing**
 - einige Anwendungen (z.B. Internet Telefonie, Spiele) brauchen geringen Delay
- **Durchsatz (throughput)**
 - einige Anwendungen (z.B. Multimedia) brauchen Mindestdurchsatz
 - andere (“elastische Anwendungen”) passen sich dem Durchsatz an
- **Sicherheit**
- **Verschlüsselung, Datenintegrität**

- Web-Seiten (web page) besteht aus Objekten
- Objekte sind HTML-Datei, JPEG-Bild, Java-Applet, Audio-Datei,...
- Web-Seite besteht aus Base HTML-Datei mit einigen referenzierten Objekten
- Jedes Objekt wird durch eine URL adressiert
 - Beispiel URL:

*http://*www.someschool.edu/someDept/pic.gif

↑
protocol

host name path name