

# Einführung Internet-Technologien

## Sommersemester 2020

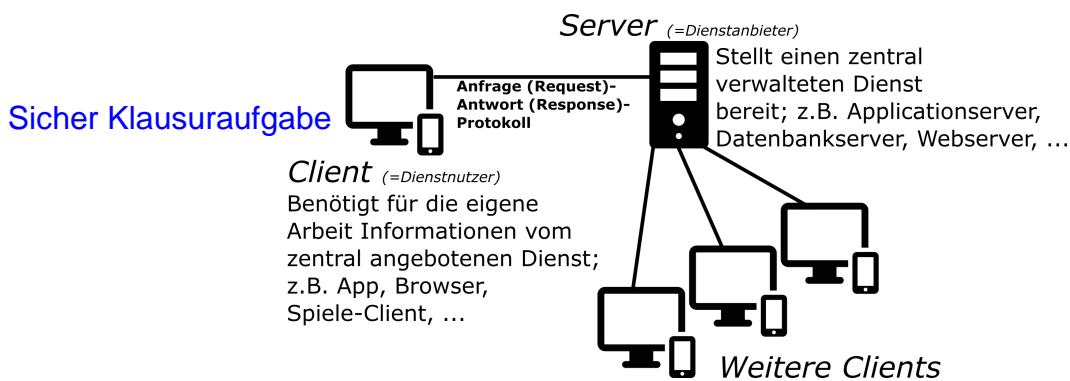
### Übungsblatt 0

## Theorie

Die Lösungen zu den Theorieaufgaben werden in den Tutorien besprochen und im Moodle veröffentlicht. Sie brauchen *keine* Lösungen zu den Theorieaufgaben abgeben!

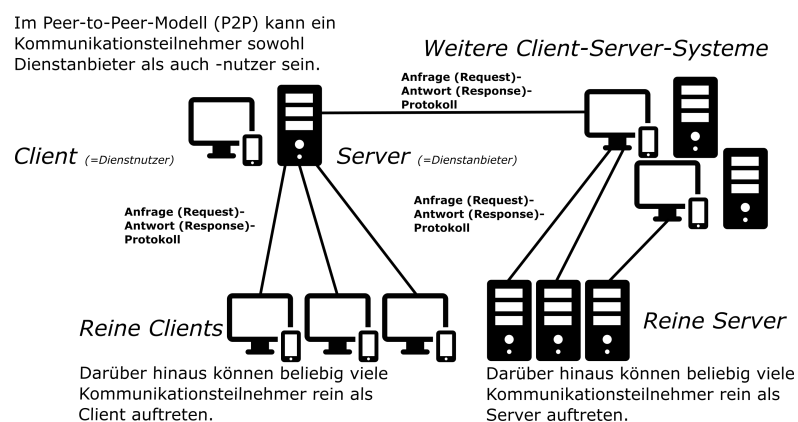
1. Skizzieren Sie das *Client-Server-Modell* und erklären Sie die einzelnen Komponenten.

Lösung:



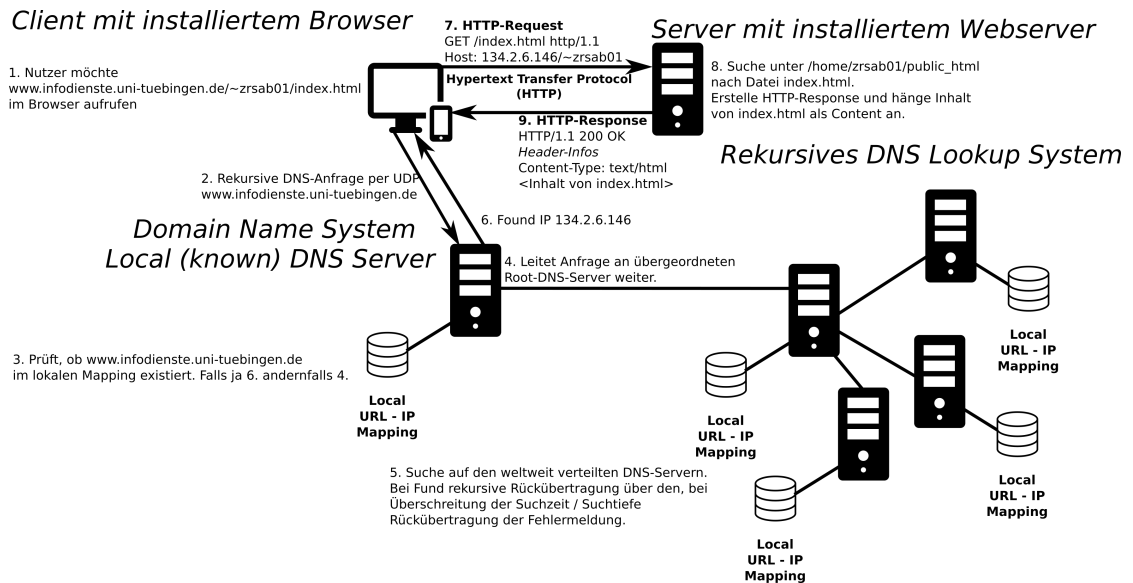
2. Skizzieren Sie das *Peer-to-Peer-Modell* und erklären Sie die einzelnen Komponenten.

Lösung:



3. Skizzieren Sie den Aufbau des World Wide Web. Integrieren Sie die Begriffe *Client*, *DNS*, *HTML*, *HTTP*, *IP*, *Port*, *Server*, *TCP*, *UDP*, *Webclient* sowie *Webserver*.

### Lösung:



4. Erklären Sie Funktion und Aufbau von IP-Adressen. Gehen Sie dabei auf IPv4 sowie IPv6 ein. Welchen IP-Adressraum besitzt die Uni Tübingen?

**Lösung:** Die IP-Adresse ist die eindeutige Netzweridentifikation von teilnehmenden Computern im Web. Über die IP-Adresse wird sichergestellt, dass sich zwei Rechner, die miteinander Kommunizieren möchten, im Rechnernetz finden und Kommunikationspakete über das Netzwerk an die korrekten Rechner gesendet werden. Die genaue Spezifikation der IP-Adresse ist im IP-Standard (RFC 791 für IPv4 und RFC 2460) beschrieben.

IPv4-Adressen sind wie folgt aufgebaut:  $a.b.c.d$ ; mit  $a, b, c, d \in \{0, 1, 2, \dots, 255\}$ ; z.B. 134.2.2.38 IPv6-Adressen sind wie folgt aufgebaut: Acht vierstellige hexadezimale Blöcke, die mit „:“ getrennt sind. Dabei kann einmalig eine Anzahl an aufeinanderfolgenden Blöcken mit Wert 0000 durch zwei aufeinander folgende „:“ abgekürzt werden; z.B. 2002:8620:2026::/48

Einige spezielle IP-Adressen werden nicht für die Teilnahme an Rechnernetzen verwendet; beispielsweise 127.0.0.1 ( $\equiv$  localhost  $\equiv$  dem eigenen Rechner) oder 0.0.0.0. ( $\equiv$  per Definition nicht erlaubt).

Vernachlässigt man die nicht verwendbaren IP-Adressen so erhält man folgende Obergrenzen für IP-Adressen:

- IPv4 verfügt über  $256^4 = 4.294.267.296$  IP-Adressen.
- IPv6 verfügt über  $(16^4)^8 = 16^{32} \approx 3,4 \cdot 10^{38}$  IP-Adressen.

Die Uni Tübingen hat ein so genanntes *Class-B*-Netz. Dabei sind die ersten beiden Blöcke der IP-Adressen fest vorgegeben: 134.2 Die hinteren beiden Blöcke der IP-Adressen sind dabei dynamisch vergebbar: 134.2.x.y;  $x, y \in \{0, 1, 2, \dots, 255\}$ . Das bedeutet, der Uni stehen  $\rightarrow 256^2 = 65536$  Mögliche Adressen zur Verfügung. Auch hier können wieder durch geschickte interne Netze mehr Netzwerkteilnehmer als vorhandene IP-Adressen integriert werden.

5. Erklären Sie Funktion und Aufbau des Hypertext Transfer Protocols (HTTP). Geben Sie ein Beispiele für eine HTTP1.1-GET-Anfrage an.

**Lösung:** Das Hypertext Transfer Protocol (HTTP) ist ein textbasiertes Kommunikationsprotokoll, das die Kommunikation zwischen Webclient und Webserver regelt. Es wird im ISO/OSI-Referenzmodell der Anwendungsschicht zugeordnet. Bisher wurden die Versionen 0.9, 1.0, 1.1 sowie seit neuestem 2.0 veröffentlicht. HTTP unterscheidet zwei Nachrichtentypen: Die *Anfrage* (den Request) sowie die *Antwort* (die Response). Ausgetauschte Nachrichten besitzen immer einen Header-Teil mit Meta-Informationen sowie den Nachrichteninhalte. Typische Header-Informationen sind die im Inhalt verwendete Zeichenkodierung, der HTTP Statuscode, Cookie-Informationen und Serverinformationen.

#### HTTP-GET am Beispiel des Aufrufs von <https://uni-tuebingen.de/> (mit Firefox)

##### Request:

```
GET / HTTP/1.1
Host: uni-tuebingen.de
User-Agent: Mozilla/5.0 (X11; Ubuntu;) Firefox/67.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: de,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Cookie: ROUTEID=.t3live01
Upgrade-Insecure-Requests: 1
```

##### Response:

```
HTTP/1.1 200 OK
Date: Mon, 15 Apr 2019 12:02:46 GMT
Server: Apache/2.4.25 (Debian)
Content-Language: de
Cache-Control: private, no-store, max-age=0
Content-Length: 195443
Expires: Mon, 15 Apr 2019 12:02:46 GMT
Content-Type: text/html; charset=utf-8
X-UA-Compatible: IE=edge
Vary: Origin
```

Statuscodes ungefähr wissen (Kategorien)

```
Set-Cookie: ROUTEID=.t3live01; path=/  
  
<!DOCTYPE html>  
<!-- Nachfolgende HTML-Ausgabe gekürzt! -->
```

6. Erklären Sie Aufgabe und Funktionsweise eines Webserver. Geben Sie Beispiele sowie Verbreitungsgrad für Webserver an.

**Lösung:** Der Webserver ist eine Anwendung, die auf einem physikalischen Server installiert ist. Diese lauscht auf den im Betriebssystem verfügbaren Ports auf eingehende Verbindungen. Eingehende Verbindungen werden anhand des HTTP-Protokolls interpretiert.

Ist eine eingehende Verbindung korrekt und wird eine Datei angefordert, so prüft der Server, ob die im HTTP-Request enthaltene angeforderte Datei in einem definierten Verzeichnis existiert (auf dem Übungsserver ist dies das public\_html-Verzeichnis). Ist dem so, erstellt der Webserver eine HTTP-Response mit dem HTTP-Statuscode 200, hängt die angeforderte Datei an und sendet das Ergebnis als HTTP-Response an den aufrufenden Client zurück. Dabei können beliebige Dateitypen ausgeliefert werden; beispielsweise HTML, mp3, png, ... . Typischerweise jedoch HTML-Dateien, welche vom Browser gerendert werden.

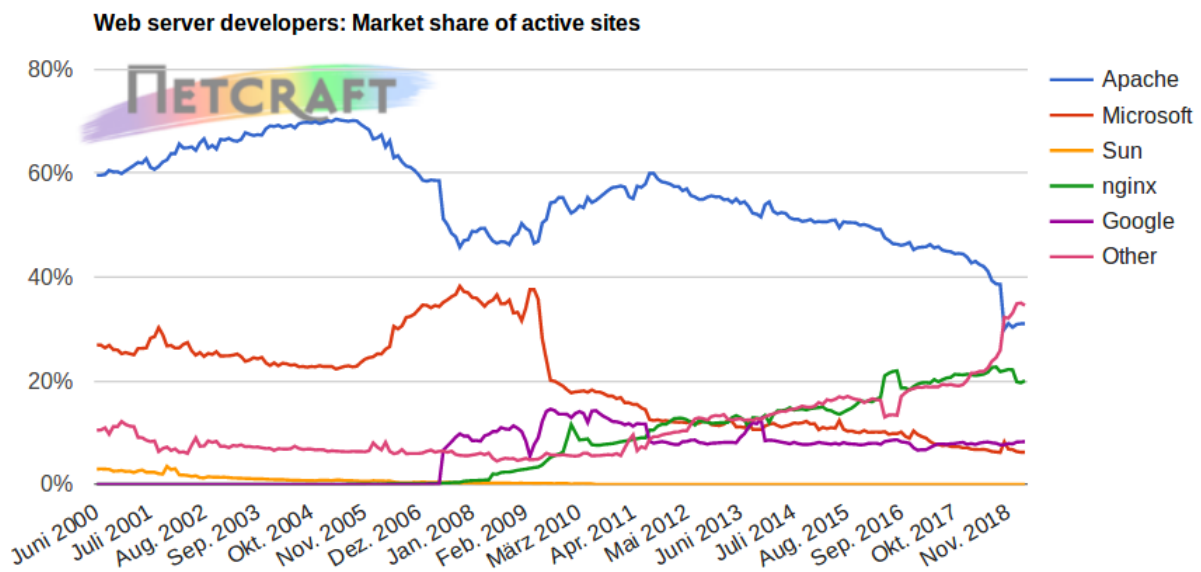


Abbildung 1: Marktanteile der Webserver, übernommen von <https://news.netcraft.com/archives/2019/03/28/march-2019-web-server-survey.html>

7. Erklären Sie Aufgabe und Funktionsweise eines Webclients. Geben Sie Beispiele sowie Verbreitungsgrad für Webclients an.

**Lösung:** Der Webclient ist eine Anwendung, die auf einem physikalischen Client

(z.B. einem Laptop) installiert ist. Dieser bietet dem Nutzer die Möglichkeit die Adresse (im Browser als *Uniform Resource Locator*, kurz URL) des zu verwendenden Dienstes anzugeben.

Wird die URL direkt IP-Adressen-basiert eingegeben, kann der Browser direkt eine Anfrage an den Dienst-bereitstellenden Server senden. Andernfalls fragt er die IP-Adresse der textuellen URL bei einem ihm bekannten DNS-Server nach und kann mit dieser eine Anfrage an den Dienst-bereitstellenden Server senden.

Um den Dienst-bereitstellenden Server anzufragen, erzeugt der Browser unter Verwendung der IP-Adresse sowie weiterer Angaben in der URL bzw. des Nutzers (welche der Browser beispielsweise aus HTML-Formularen ausliest) einen HTTP-Request, welches er an den Server sendet. Anschließend wartet der Client auf die Antwort des Servers. Aus der Serverantwort – der HTTP-Response – liest der Client zunächst alle Informationen aus dem HTTP-Header aus; beispielsweise den Statuscode, Cookie-Werte, Serverdaten, ... Sobald diese verarbeitet sind (und die Anfrage erfolgreich verarbeitet werden konnte) nimmt der Browser das Anfrage-Ergebnis aus der Response und zeigt dem Nutzer dieses an.

Insbesondere wenn das Ergebnis den Content-Type *text/html* ist, verwendet der Browser eine lokale Render-Engine, welche aus den enthaltenen HTML- sowie CSS-Befehlen die Darstellung berechnet und darstellt. Weiterhin können eingebettete Client-seitige Skriptsprachen direkt vom Browser ausgeführt werden; beispielsweise JavaScript.

8. Erklären Sie Aufgabe und Funktionsweise des Domain Name Systems (DNS).

**Lösung:** Da Menschen sich Sprach-Elementen besser merken können als Zahlen, ist die ausschließliche Verwendung von IP-Adressen zur eindeutigen Identifizierung von Server ungeeignet. Ebenfalls problematisch ist dies, wenn die IP-Adresse des Servers geändert wird. Darum wurde das Domain Name System (DNS) eingeführt, welches wie ein Telefonbuch funktioniert: Jeder Server erhält einen Namen und die Zuordnung zwischen Namen und aktueller IP-Adresse erfolgt im DNS. Das DNS agiert dabei dezentralisiert: Jeder Namensraum wird von eigenen DNS-Servern verwaltet, z.B. die .de-Domäne von anderen als die .com oder .net-Domänen.

Möchte der Client einen Request an einen Server senden, so prüft er zunächst, ob er dessen IP-Adresse vermerkt hat. Ist dies nicht der Fall, so muss er einem DNS-Server anfragen. Dessen IP-Adresse ist ihm entweder a priori bekannt, oder er kann diese per Dynamic Host Configuration Protocol (DHCP) anfordern. Er übermittelt dem DNS-Server dann die Domain, worauf dieser in seiner Zuordnungs-Tabelle nachprüft, ob er die Domain kennt. Ist dies nicht der Fall, so leitet der DNS-Server die Anfrage an andere ihm a priori bekannte DNS-Server weiter, basierend auf der Domäne. Als Antwort erhält er entweder die gesuchte IP-Adresse oder einen Verweis auf einen anderen DNS-Server, bei dem er erneut nachfragen kann. Dies wird so lange fortgeführt, bis ein Ergebnis gefunden wurde oder ein DNS-Server

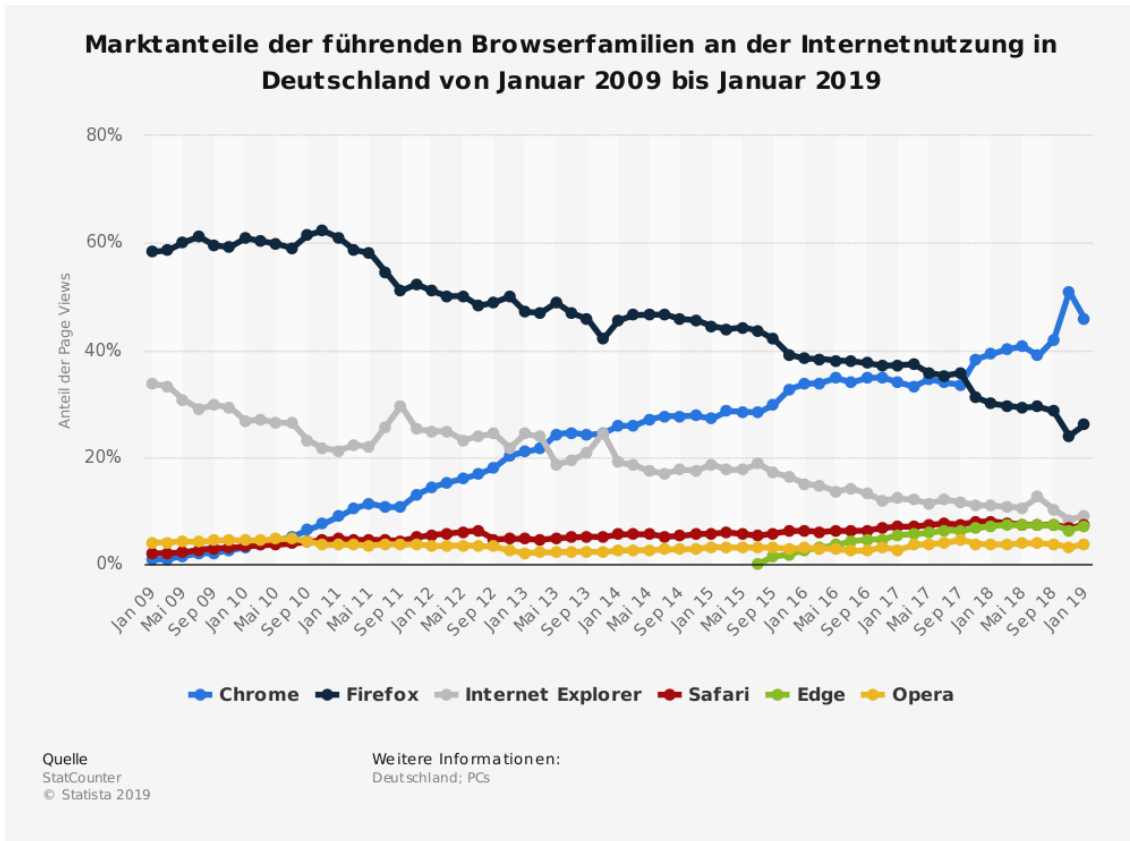
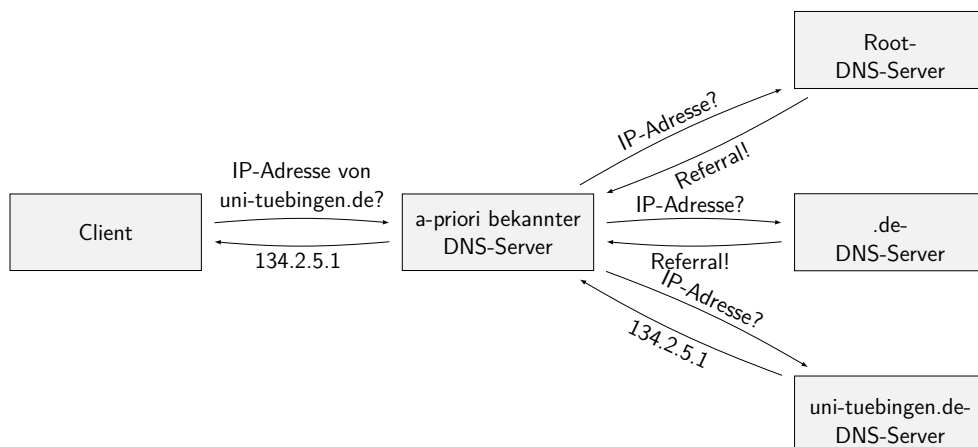


Abbildung 2: Marktanteile der Browser

final entscheidet, dass die Domain nicht existiert. Abschließend erhält der Client die IP-Adresse vom ursprünglich angefragten DNS-Server übermittelt.

Beispiel: Bestimme die IP-Adresse von uni-tuebingen.de:



9. Weshalb ist es wichtig immer die aktuellsten Versionen von Webclients und -servern zu kennen / installieren?

**Lösung:** Da über das WorldWideWeb die Kommunikation zwischen einem Endrechner sowie mehreren – zum Teil unbekannten – Servern ermöglicht wird, ist dies eine potentielle Angriffsstelle für kriminelle Handlungen. Häufig werden bei der Entwicklung von Clients- und Servern ungewollt Fehler in die Software eingebaut, welche von Hackern ausgenutzt werden. Gerade im Web ist dies problematisch, da über automatisierte Skripte in kürzester Zeit viele Systeme angegriffen werden können. Um zu verhindern, dass man selbst von solch einem automatisierten Angriff betroffen wird, ist es deshalb unerlässlich sich mit den aktuellen Versionen, veröffentlichten Schwachstellen sowie Workarounds für diese zu befassen. Optimaler Weise hat man einen Patch-Plan, über den sichergestellt ist, dass zwischen Veröffentlichung einer neuen Version und Installation dieser auf allen relevanten Systemen maximal 1 Monat Zeit vergehen lässt. Bei sicherheitskritischen Updates (z.B. Zero-Day exploits) muss dieser Patch-Plan ad hoc anpassbar sein, um die Systeme des Systems innerhalb von 48h durch die angebotenen Workarounds zu schützen.

10. Füllen Sie die folgende Tabelle mit wichtigen Linux-Kommandozeilenbefehlen aus:

**Lösung:**

Befehl	Bedeutung	Beispiel
cd	In ein Verzeichnis wechseln	cd ~/public_html
chmod	Rechte einer Datei ändern	chmod 755 file, chmod o=rwx file, chmod g+w file
cp	Datei kopieren, bei Verzeichnis mit Option -r	cp file fileCopy, cp -r dir/ newdir/
less	Textdatei scrollbar anzeigen	less file.txt
mkdir	Verzeichnis erstellen	mkdir folder
mv	Dateien umbenennen/verschieben	mv src dst
nano	einfacher Text-Editor	nano file.txt
pwd	aktuellen Verzeichnispfad anzeigen	
rm	Datei/Verzeichnis löschen (mit Option -r)	rm file, rm -r dir

Hilfe zu Befehlen und Optionen erhält man über die „Manpages“. Hierfür gibt man `man name-des-Befehls` ein, beispielsweise `man chmod`

Bei sehr vielen Befehlen erhält man zudem durch die Angabe der Option `--help` Hilfe, beispielsweise `chmod --help`