

GRNVS Tutorium 11 - SS21

Fabian Sauter

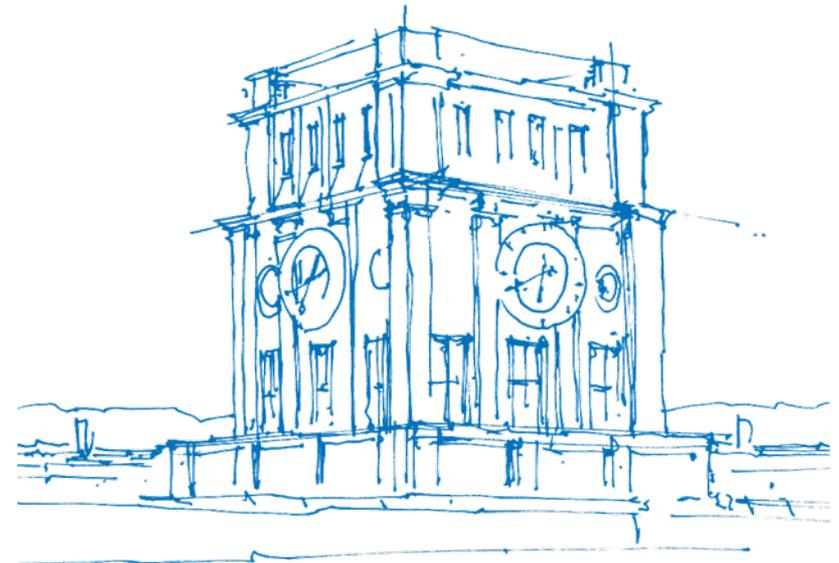
Technische Universität München

Grundlagen: Rechnernetze und Verteilte Systeme (IN0010)

Lehrstuhl für Netzarchitekturen und Netzdienste

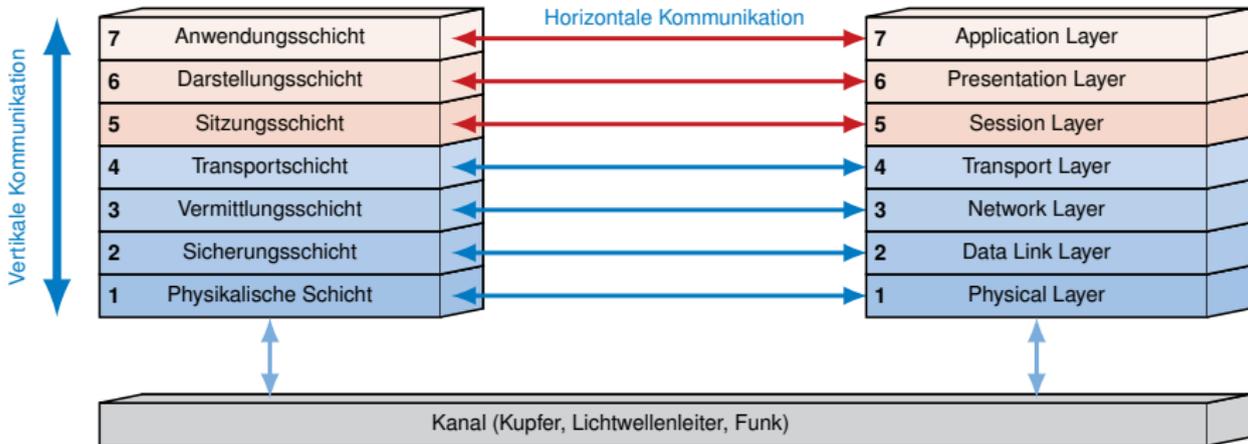
Garching, 05.07.2021

Slides & Notes: <http://grnvs.uwpx.org>



TUM Uhrenturm

Wiederholung



- Dienste der Sitzungs- und der Darstellungsschicht sind in einzelnen Fällen in Form standardisierter Protokolle implementiert.
- In vielen Fällen sind die Funktionen der Sitzungs- bzw. der Darstellungsschicht in die jeweilige Anwendung integriert.

⇒ Eine Differenzierung ist oft schwierig

Gemäß Standard X.200 sind Dienste der Sitzungsschicht (engl. [Session Layer](#)) entweder verbindungsorientiert oder verbindungslos:

verbindungsorientiert (engl. „connection-oriented“)

- Es wird eine Verbindung zwischen den Kommunikationspartnern aufgebaut.
- Die Verbindung bleibt dabei über die Dauer einzelner Transfers (oder Verbindungen) der Transportschicht hinweg bestehen.
- Wie bei TCP-Verbindungen kann auch hier zwischen den Phasen [Verbindungsaufbau](#), [Datentransfer](#) und [Verbindungsabbau](#) unterschieden werden.

verbindungslos (engl. „connection-less“)

- Daten werden im Wesentlichen nur an die Transportschicht durchgereicht.
- Es wird keine Verbindung aufgebaut und kein Zustand zwischen den Kommunikationspartnern gehalten.

Hinweis

Eine Verbindung der Sitzungsschicht ([Session](#)) ist nicht gleichbedeutend mit einer Verbindung der Transportschicht. Eine Session kann beispielsweise nacheinander mehrere TCP-Verbindungen beinhalten oder Funktionen wie Bestätigungen unserem UDP-Chat hinzufügen, welche im Fall des TCP-Chats automatisch durch die Transportschicht erbracht werden.

Beispiel 1: HTTP (Hyper Text Transfer Protocol)¹ → Details später

- HTTP in seiner ursprünglichen Form ist zunächst zustandslos.
- Zwischen unterschiedlichen Anfragen und Antworten besteht daher zunächst kein Zusammenhang.
- **Cookies** ermöglichen es, dass eine Sitzung über mehrere Anfragen und Antworten, Interaktionen und TCP-Verbindungen hinweg bestehen bleibt.
- Cookies sind kleine Datenfragmente, welche von einem Webserver (bzw. einem damit verbundenen Anwendungsprozess) an einen Webclient übertragen und dort gespeichert werden können.
- Dadurch wird es möglich, zeitlich getrennte oder von unterschiedlichen Adressen stammende Anfragen einem bestimmten Client und damit einem bestimmten Nutzer zuordnen zu können.
- HTTP wird üblicherweise der Anwendungsschicht (Schicht 7) zugeordnet, beinhaltet aber auch Funktionen der Darstellungs- und Sitzungsschicht (Schichten 6/5).

¹ Bei HTTP handelt es sich um ein Protokoll der Anwendungsschicht.

Beispiel 2: TLS (Transport Layer Security)¹

- TLS ist ein Protokoll zur Authentifizierung und verschlüsselten Übertragung von Daten über einen verbindungsorientierten Transportdienst.
- Es ist die Grundlage beispielsweise für HTTPS.
- Es bietet unter anderem
 - Authentifizierung² („ist mein Gegenüber der, für den ich ihn halte“),
 - Integritätsschutz (Schutz vor Manipulation von Daten) und
 - Verschlüsselung (Vertraulichkeit, d. h. Schutz vor unberechtigtem Mitlesen).
- Beim Verbindungsaufbau werden zunächst die Kommunikationsparameter der Sitzung (insbesondere Verfahren für Authentifizierung und Verschlüsselung, sowie Zertifikate) ausgetauscht.
- Sitzungen können mit Hilfe von Session-IDs über mehrere TCP-Verbindungen hinweg erhalten bleiben.
- Während die TLS-Funktionen zur Verwaltung und Wiederaufnahme von Sessions der Sitzungsschicht zuzuordnen sind, können die Verschlüsselungsfunktionen der Darstellungsschicht zugeordnet werden.

¹ Dient hier lediglich als Beispiel, wird im Rahmen der Vorlesung aber nicht eingehender behandelt. Man sollte allerdings wissen, wozu es dient.

² Nicht zu verwechseln mit [Authentifizierung](#), d. h. Überprüfung von Zugriffsberechtigungen.

Die Aufgabe der **Darstellungsschicht** (engl. **Presentation Layer**) ist es, den Kommunikationspartnern eine einheitliche Interpretation der Daten zu ermöglichen, d. h. Daten in einem einheitlichen Format zu übertragen.

Der Darstellungsschicht sind grundsätzlich folgende Aufgaben zugeordnet:

- die Darstellung der Daten (Syntax),
- die Datenstrukturen zur Übertragung der Daten
- die Darstellung der Aktionen an diesen Datenstrukturen, sowie
- Datentransformationen.

Hinweis

Die Darstellung auf Schicht 6 muss nicht der Darstellung auf Schicht 7 (Anwendungsschicht) entsprechen. Die Darstellungsschicht ist für die **Syntax** der Nutzdaten verantwortlich, die **Semantik** verbleibt bei den Anwendungen.¹

- Anwendungen sollen syntaxunabhängig miteinander kommunizieren können.
- Anwendungsspezifische Syntax kann von der Darstellungsschicht in eine einheitliche Form umgewandelt und dann übertragen werden.

¹ Unter **Syntax** versteht man die Darstellung von Daten nach bestimmten Regeln (**Grammatik**). Wird Daten eine Bedeutung zugeordnet, spricht man von Information (Aufgabe der **Semantik**).

Daten liegen in einer von zwei Formen vor:

(1) **Textzeichen** bzw. **Symbole** in lesbarer Form („human readable“), z. B. Buchstaben, Textrepräsentation von Zahlen, Sonderzeichen. . .

- Ein **Zeichensatz** ist eine Menge textuell darstellbarer Zeichen sowie deren Zuordnung zu einem **Codepoint**¹.
- Wie die Codepoints eines bestimmten Zeichensatzes in binärer Form (also mittels einer Sequenz von Bits) dargestellt werden wird durch **Kodierungsvorschriften** festgelegt.
- Für einen Zeichensatz können ggf. mehrere mögliche Kodierungen existieren.

(2) **Binäre Daten** (also eine Sequenz von Bits), z. B. binäre Darstellungen von Buchstaben, Zahlen und Symbolen aber auch Bilder, Musik, Filme, etc. in digitaler Form. . .

- Ein **Datum** ist eine für Computer verarbeitbare „Einheit“, d. h. eine kurze Sequenz von Bits, deren Länge meist ein Vielfaches von 8 bit ist.²
- Was ein Datum repräsentiert – eine Zahl, ein Zeichen, einen Teil davon oder doch ein Bild – ist kontextabhängig (vgl. Kapitel 1 „Information und deren Bedeutung“).
- Ist bekannt, dass es sich bei den vorliegenden Daten um Text handelt, welche Kodierung verwendet wurde und um welchen Zeichensatz es sich handelt, lassen sich die binären Daten leicht wieder in Textzeichen übersetzen.

Hinweis: Binäre Daten (z. B. ein Bild) können nicht ohne Weiteres mit einem Zeichensatz dargestellt werden. Hierzu gibt es eigene Kodierungsvorschriften, die es ermöglichen, Binärdaten zu kodieren, so dass sie mit einem Zeichensatz dargestellt werden können.

¹ Ein Codepoint ist eine eindeutige „Kennzahl“ für höchstens ein Textzeichen (nicht alle Codepoints müssen vergeben sein).

² Die kleinste im Speicher adressierbare Einheit ist für heutige Computer ein Oktett, also ein Block von 8 bit, welches im Sprachgebrauch als Byte bezeichnet wird. Üblich sind daneben noch die Größen 16 bit, 32 bit und 64 bit, welche häufig als Word, Double Word bzw. Quad Word bezeichnet werden. Speicher wird byteweise adressiert, wobei es prozessorabhängig ist, in welcher Reihenfolge die einzelnen Oktette geladen werden. Folglich ist die **Byte Order** von essentieller Bedeutung.

JSON definiert die folgenden Datentypen: number, string, boolean, array, object, null

- Zwischen den Elementen kann (beliebiger) Whitespace eingefügt werden.
- JSON wird im allgemeinen als UTF-8 kodiert.
- JSON-Dokumente besitzen, anders als XML-Dokumente, kein explizites Schema.

Beispiel:

```
{
  "Image": {
    "Width": 800,
    "Height": 600,
    "Title": "View from 15th Floor",
    "Thumbnail": {
      "Url": "http://www.example.com/image/481989943",
      "Height": 125,
      "Width": 100
    },
    "Animated" : false,
    "IDs": [116, 943, 234, 38793]
  }
}
```

JSON (eat-api)

<https://github.com/TUM-Dev/eat-api>

Bei Kompressionsverfahren muss unterscheiden werden:

1. Verlustfreie Komprimierung (engl. lossless compression)

- Komprimierte Daten können verlustfrei, d. h. exakt und ohne Informationsverlust, wiederhergestellt werden.
- Verlustfrei komprimierte Dateiformate sind beispielsweise ZIP, PNG¹ (Bilder), FLAC² (Musik), ...

2. Verlustbehaftete Komprimierung (engl. lossy compression):

- Komprimierte Daten können im Allgemeinen nicht wieder exakt rekonstruiert werden.
- Es tritt also ein Verlust von Information bei der Komprimierung auf.
- Dafür ermöglichen diese Verfahren meist höhere und in Abhängigkeit des Verlustfaktors variable Kompressionsraten.
- Verlustbehaftet komprimierte Dateiformate sind beispielsweise MP3, MPEG, JPEG, ...

¹ Portable Network Graphics

² Free Lossless Audio Codec

Konstruktion

- Gegeben Sei das Alphabet $\mathcal{A} = \{A,B,C,D,E,F,G,H\}$ sowie Auftrittswahrscheinlichkeiten $\Pr[X = z]$ für alle Zeichen $z \in \mathcal{A}$.
- Es sei außerdem vorausgesetzt, dass die einzelnen Zeichen unabhängig voneinander auftreten.¹

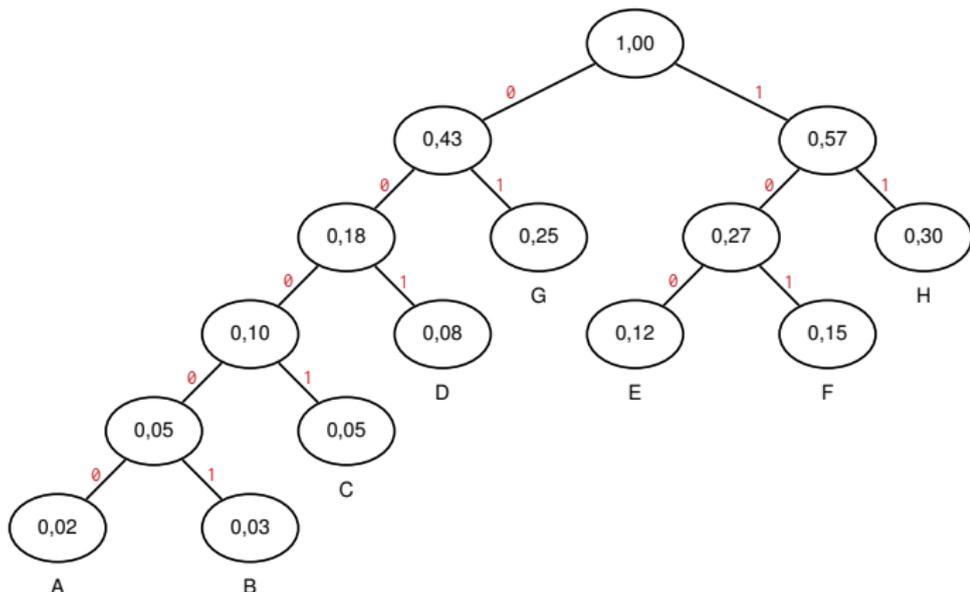
z	$\Pr[X = z]$
A	0,02
B	0,03
C	0,05
D	0,08
E	0,12
F	0,15
G	0,25
H	0,30

¹ Andernfalls würden die Aussagen zur Optimalität des Huffman-Codes im Allgemeinen nicht mehr zutreffen.

Konstruktion

- Gegeben Sei das Alphabet $\mathcal{A} = \{A,B,C,D,E,F,G,H\}$ sowie Auftrittswahrscheinlichkeiten $\Pr[X = z]$ für alle Zeichen $z \in \mathcal{A}$.
- Es sei außerdem vorausgesetzt, dass die einzelnen Zeichen unabhängig voneinander auftreten.

z	$\Pr[X = z]$
A	0,02
B	0,03
C	0,05
D	0,08
E	0,12
F	0,15
G	0,25
H	0,30



Durchschnittliche Codewortlänge

z	$\Pr[X = z]$	Huffman-Code	Länge $l_H(z)$	Uniformer Code
A	0,02	00000	5	000
B	0,03	00001	5	001
C	0,05	0001	4	010
D	0,08	001	3	011
E	0,12	100	3	100
F	0,15	101	3	101
G	0,25	01	2	110
H	0,30	11	2	111

- Uniformer Code:

$E[l_U(z)] = 3,0$, da alle Codewörter gleich lang sind

- Huffman-Code:

$$E[l_H(z)] = \sum_{z \in \mathcal{A}} \Pr[X = z] l_H(z) = 2,6$$

$$\Rightarrow \text{Die Einsparung beträgt } 1 - \frac{E[l_H(z)]}{E[l_U(z)]} \approx 13\%$$

Einordnung im ISO/OSI-Modell

Sitzungsschicht

Darstellungsschicht

Anwendungsschicht

- Domain Name System (DNS)

- Uniform Resource Locator (URL)

- HyperText Transfer Protocol (HTTP)

- Simple Mail Transfer Protocol (SMTP)

- File Transfer Protocol (FTP)

Literaturangaben

Motivation:

- Möchte ein Nutzer (Mensch) einen Computer adressieren, z. B. beim Aufruf einer Webseite, will er sich gewöhnlich nicht dessen IP-Adresse merken müssen.
- Stattdessen adressiert man das Ziel üblicherweise mittels eines hierarchisch aufgebauten Namens, z. B. www.google.com.

Das **Domain Name System (DNS)** besteht aus drei wesentlichen Komponenten:

1. Der Domain Namespace

- ist ein hierarchisch aufgebauter Namensraum, und
- hat eine baumartige Struktur.

2. Nameserver

- speichern Informationen über den Namensraum,
- jeder Server kennt nur kleine Ausschnitte des Namensraums.

3. Resolver sind Programme,

- die durch Anfragen an Nameserver Informationen aus dem Namespace extrahieren, und
- anfragenden Clients bzw. Anwendungen zur Verfügung stellen.

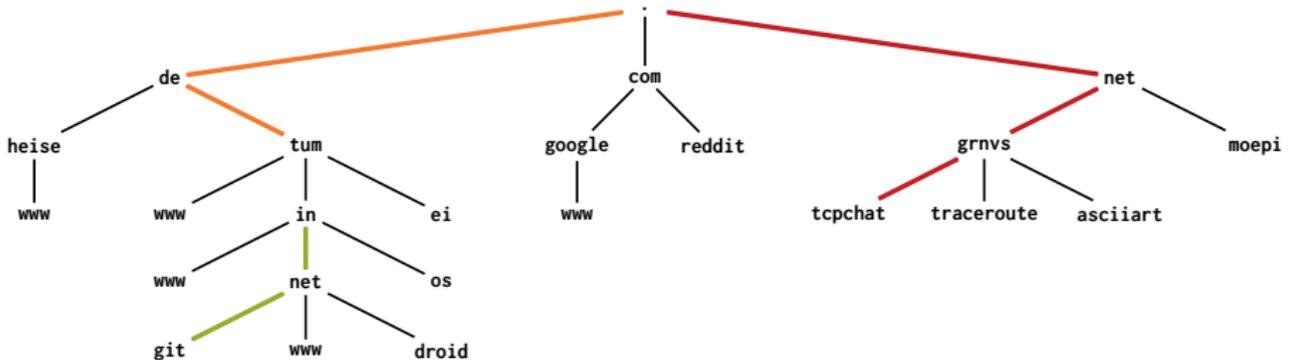
Details: siehe entsprechende RFCs [6, 7, 2, 4]

Gleichzeitig abstrahiert DNS von IP-Adressen, d. h. anstelle die IP-Adresse eines Servers z. B. im Emailprogramm konfigurieren zu müssen, kann sein Name angegeben werden. Die IP-Adresse kann sich damit sogar ändern, ohne dass die Konfiguration des Mailprogramms geändert werden muss.

Domain Name System (DNS)

Domain Namespace

Ein kleiner Auszug aus dem Namespace:



- Ein **Label** ist ein beliebiger Knoten im Namespace.
- Ein **Domain Name** ist eine Sequenz von Labels:
 - Ein **Fully Qualified Domain Name (FQDN)** besteht aus der vollständigen Sequenz von Labels ausgehend von einem Knoten bis zur Wurzel und endet mit einem Punkt, z. B. **tum.de.** oder **tcpchat.grnvs.net.**
 - Endet er nicht mit einem Punkt, handelt es sich zwar ebenfalls um einen Domain Name, allerdings ist dessen Angabe relativ ausgehend von einem anderen Knoten als der Wurzel, z. B. **git.net.in.**
 - Ein FQDN kann als **Suffix** für einen nicht-qualifizierten Namen verwendet werden, z. B. ergibt **git.net.in** zusammen mit dem FQDN **tum.de.** einen neuen FQDN **git.net.in.tum.de.**
 - Ob ein FQDN existiert (z. B. in eine Adresse aufgelöst werden kann), bleibt zunächst offen.

Die Informationen, die in einer Zone gespeichert sind, bezeichnet man als **Resource Records**:

- **SOA Record (Start of Authority)** ist ein spezieller Record, der die Wurzel der Zone angibt, für die ein Nameserver autoritativ ist.
- **NS Records** geben den FQDN eines Nameservers an. Dieser kann auch auf FQDNs in anderen Zonen verweisen.
- **A Records** assoziieren einen FQDN mit einer IPv4-Adresse.
- **AAAA Records** assoziieren einen FQDN mit einer IPv6-Adresse.
- **CNAME Records** sind Aliase, d. h. ein FQDN verweist auf einen "Canonical Name", der selbst wiederum ein FQDN ist.
- **MX Records** geben den FQDN eines Mailservers für eine bestimmte Domain an, welcher sich nicht notwendigerweise in derselben Zone befinden muss.
- **TXT Records** assoziieren einen FQDN mit einem String (Text). Kann für unterschiedliche Zwecke verwendet werden.
- **PTR Records** assoziieren eine IPv4- oder IPv6-Adresse mit einem FQDN (Gegenstück zu A bzw. AAAA Records).

Hinweise:

- Mehrere A oder AAAA Records (auch unterschiedlicher Zonen) können mit derselben IP-Adresse assoziiert sein.
- Für einen FQDN kann es maximal einen CNAME geben. Wenn ein CNAME existiert, handelt es sich um einen Alias, weswegen es keine weiteren Resource Records für den betreffenden FQDN mehr geben darf.
- Für eine Zone bzw. Domain gibt es üblicherweise mehrere NS bzw. MX Records.

Beispiel DNS

Ablauf:

- Aufgabe 1
- Aufgabe 2

Studenten zählen

(Nur als Erinnerung für mich.)