

Referenzarchitekturen und Kernkonzepte



<https://iuk.one/1010-1009>

Clemens H. Cap

ORCID: 0000-0003-3958-6136

Department of Computer Science
University of Rostock
Rostock, Germany
clemens.cap@uni-rostock.de

28. 11. 2020 Vers. 2



1. Vereinbarungen

Ziel: Warum sind Vereinbarungen gerade in der Kommunikation so wichtig – und wie sehen sie typischerweise aus?

1. Vereinbarungen

2. Schichtenmodelle

3. ISO-OSI, 1. Teil

4. Kernkonzepte

5. ISO-OSI, 2. Teil

6. Internet Architektur

Beobachtung

Erfolgreiche Kommunikation zwischen Teilnehmern erfordert eine Vielzahl bindender Vereinbarungen in Organisation und Technik.

Beispiele:

- Benutzte Symbole und deren Codierung.
- Physikalische Eckwerte.
- Bedeutung der Nachrichten.
- Verhalten zur Aufnahme und Beendigung einer Übertragung.
- Vorgehen im Fehlerfall.
- Ansprechpartner bei Mißbrauch.

1. Vereinbarungen

Standards

Definition: Standard

Ein Standard ist eine schriftliche Festlegung von Eigenschaften oder Vorgehensweisen.

De Jure Standard: Basis ist offizielle Vereinbarung, Gesetz oder Normung.

Bsp: V.24 Standard der seriellen Schnittstelle als Standard der CCITT.

Bsp: IEEE 802.11 WLAN Standard als Standard der IEEE.

De Facto Standard: Ohne offizielle Vereinbarung, aber (meist) eingehalten.

Bsp: Linux als ein de facto Betriebssystem-Standard.

Bsp: BIND als der de facto Standard bei DNS Implementierungen.

Offener Standard: Dokumente öffentlich, Rechtslage ermöglicht Umsetzung.

Bsp: Linux Betriebssystem Schnittstelle, HTTP Protokoll.

Proprietärer Standard: Dokumente geheim, Rechtslage behindert Umsetzung.

Bsp: Microsoft Windows oder Apple iOS Betriebssystem Schnittstelle (teils geheim).

Bsp: MP3 Audio Format (patentiert).

Definition: Norm

Eine Norm ist eine nützliche Vereinbarung über Eigenschaften von Produkten.

Definition: Normung nach DIN 820

Normung ist die planmäßige, durch die interessierten Kreise gemeinschaftlich durchgeführte Vereinheitlichung von materiellen und immateriellen Gegenständen zum Nutzen der Allgemeinheit.

- Normen sind sprachlich und konzeptuell oft abstrakt und schwerfällig.
- Normen haben keine eigene Gesetzeskraft, liegen aber im öffentlichen Interesse.
- Gesetze können sich auf Normen beziehen.
- Entstehung: Normen werden von sog. Normierungsgremien erstellt.

Definition: Protokoll

Ein **Protokoll** ist eine Vorschrift (Verabredung) zu Form, Ablauf und Steuerung eines technischen Prozesses.

Beispiele für den **Regelungsumfang**:

- Formate von Frames (Nutzlast + Steuerdaten in Header & Trailer, auch: Paket).
- Übertragungsgeschwindigkeiten und Zeitverhalten.
- Verhalten im Fehlerfall.

Beispiele für **Protokolle**:

- IP Internet Protokoll
- SMTP Simple Mail Transfer Protokoll
- HTTP Hypertext Transfer Protokoll
- FTP File Transfer Protokoll

1. Vereinbarungen

Referenzarchitekturen im Bauwesen



Abb. 1: Pfählbau Unteruhldingen am Bodensee. ©
Rechtsnachweis siehe Anhang



Abb. 2: Zeltanhänger mit aufgebautem Zelt. © Rechts-
nachweis siehe Anhang

Die **Architektur von Systemen** orientiert sich an den **Zielen** und **Einsatzbedingungen**, weist aber gleichwohl gewisse **Gemeinsamkeiten** auf.

- Türen zum Betreten.
- Dächer zum Schutz nach oben.
- Öffnungen zum Einlaß von Licht.

Definition: Referenzarchitektur

Eine Referenzarchitektur ist ein idealtypisches Modell für ein zu konstruierendes System.

Aus der **Architektur** übernommenes Konzept.

Geht auf den römischen Architekten **Vitruv** (100 n. Chr) zurück.

Anforderungen von Vitruv auch heute noch auf Informatik übertragbar:

- **Nützlichkeit** Lateinisch: Utilitas
- **Robustheit** Lateinisch: Firmitas
- **Anmut** Lateinisch: Venustas

Beispiele für Referenzarchitekturen:

- ISO-OSI-Modell und Internet-Architektur im Bereich Kommunikation.
- Workflow Management Coalition WfMC im Bereich Workflow Systeme.
- Lambda-Architektur im Bereich Big Data.
- Supply-Chain-Operations-Reference-Modell SCOR im Bereich Logistik.

Definition: Entwurfsmuster

Ein Entwurfsmuster (design pattern) ist eine bewährte Lösungsschablone für wiederkehrende Entwurfsprobleme.

In der Architektur von Christoph Alexander eingeführt.

Von Erich Gamma in die Informatik übertragen (**Gang-of-Four Buch**).

Zentrales Konzept im Software-Engineering.

Beispiel in Kommunikation: Proxy: Stellvertreter für einen Dienst.

Beispiel hier: Web-Proxy: Verhält sich wie ein Web-Server (ist aber keiner).

- Greift selber auf Web-Server zu, holt von dort Daten, reicht weiter.
- Verhält sich dem Client gegenüber so, daß dieser glaubt, auf dem Server zu surfen.
- Nutzbar für: Caching, Filtern (Ads, Viren, unerwünschte Inhalte, Jugendschutz), Zeit-Restriktionen, geographische Restriktionen, Data mashups uvm.

2. Schichtenmodelle

2.1. Beispiel

2.2. Abstraktes Schichtenmodell

Ziel: Schichtenmodelle sind das wichtigste Handwerkszeug in der Modellierung großer Systeme, speziell in der Kommunikation.

1. Vereinbarungen
2. **Schichtenmodelle**
3. ISO-OSI, 1. Teil
4. Kernkonzepte
5. ISO-OSI, 2. Teil
6. Internet Architektur

Wie unterhält sich ein
Chinesischer Philosoph in Peking,
der nur Chinesisch spricht,
mit einem spanischen Philosophen in Madrid,
der nur spanisch spricht,
über den Philosophen Kant?



Abb. 3: Beispiel-Architektur zum Schichtenmodell

2.1 Beispiel

Beispiel zum Schichtenmodell



Abb. 4: Implementierung des Dialogs über Kant aus Sicht der Philosophen. Die strichlierte Kommunikation besteht nur *virtuell*.

2.1 Beispiel

Beispiel zum Schichtenmodell

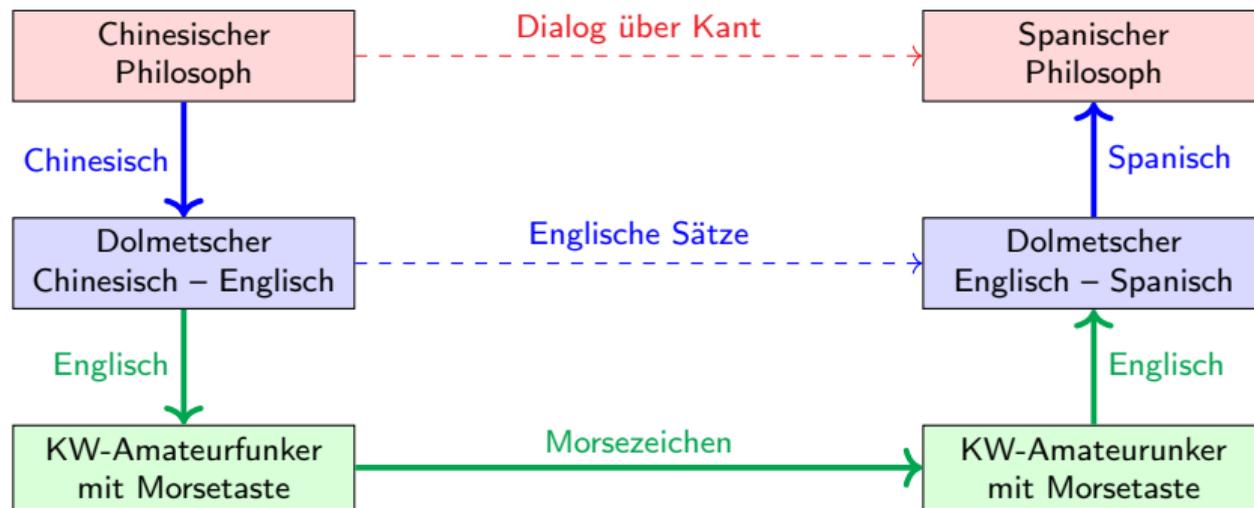


Abb. 5: Implementierung des Dialogs über Kant aus globaler Sicht. Die strichlierte Kommunikation besteht nur virtuell.

2.2 Abstraktes Schichtenmodell

Dienst-Zugang und Methoden

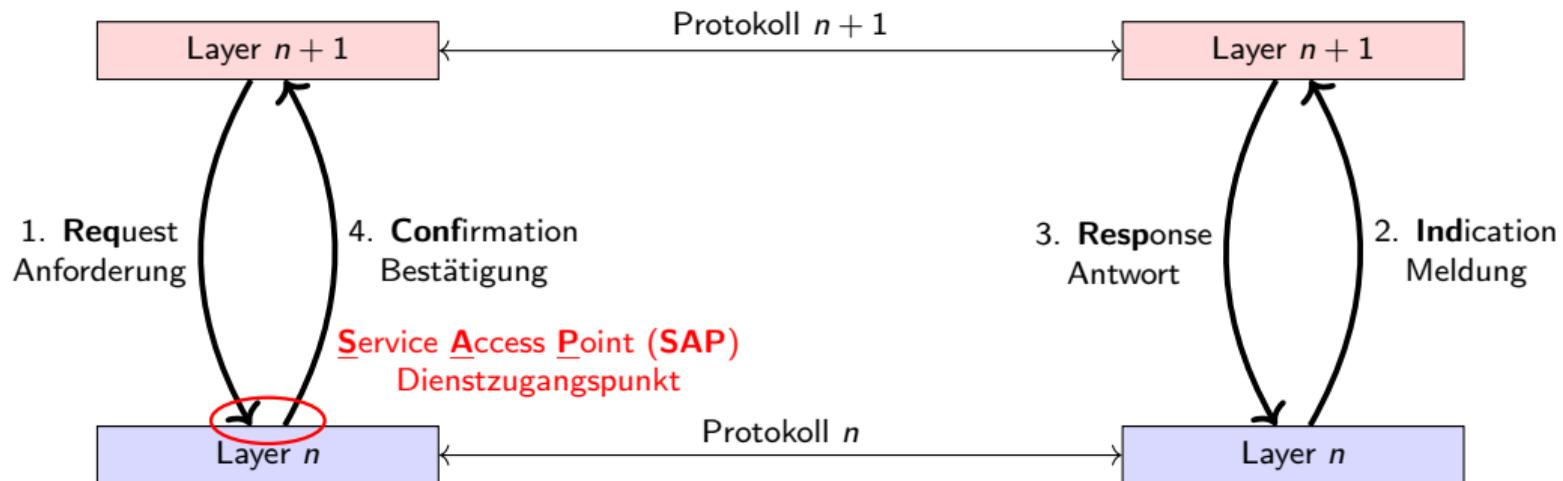


Abb. 6: Dienst-Zugang und Methoden

Prinzipien des Schichtenmodells

Spezifische Funktionalität:

Jede Schicht hat ihre eigene, wohldefinierte Funktion.

Kapselung: (Information Hiding)

Die Implementierungsdetails jeder Schicht sind nach außen verborgen.

Zugangspunkte:

Jede Schicht interagiert nur über wohldefinierte Dienst-Zugangspunkte mit der Außenwelt.

Unterschiedliche Koppelungs-Formen:

- **Strenge Schichtung:** Jede Schicht interagiert nur mit der nach oben oder unten direkt angrenzenden Schicht.
- **Offene Schichtung:** Eine Schicht darf jede unter ihr gelegene Schicht aufrufen.

3. ISO-OSI, 1. Teil

3.1. Überblick

3.2. Die unteren 4 Layer

Ziel: Wir lernen das Grundkonzept im Überblick und die unteren 4 Layer kennen. Danach lernen wir einige weitere Konzepte der Kommunikation kennen und kehren dann wieder zum ISO-OSI Modell zurück.

1. Vereinbarungen

2. Schichtenmodelle

3. ISO-OSI, 1. Teil

4. Kernkonzepte

5. ISO-OSI, 2. Teil

6. Internet Architektur

Open System Interconnection Modell der International Standardization Organization.

- Erster umfassender internationaler Standard der Kommunikation.
- Referenzmodell für erforderliche Funktionalitäten eines Kommunikations-Systems.
- Schichtenmodell mit strenger Schichtung.

Regelt unter anderem:

- Welche Konzepte und Komponenten werden benötigt?
- Welche Funktionen sind erforderlich?
- Wie sind die Schichten aufeinander aufgebaut?

Historie und Bewertung

- 1981 Draft Standard
- 1984 als International Standard
- Hohe Bedeutung als konzeptuelles, mentales Modell
- Geringe Bedeutung in seiner Implementierung

3.1 Überblick

Die 7 Schichten des ISO-OSI Modells

1: Physical Layer	Ethernet	Unsecured Point-to-Point Communication
2: Data Link Layer	Ethernet, SLIP, PPP, PPPoE, ARP, RARP	Secured Point-to-Point Communication
3: Network Layer	IP, ICMP, IGMP	Communication between Networks
4: Transport Layer	TCP, UDP, SCTP	Communication between Service Endpoints
5: Session Layer	Cookies, Session Ids	
6: Presentation Layer	ASN.1, CORBA, Protobuf	
7: Application Layer	HTTP, POP, IMAP, NTP, SMTP, FTP	Application Services with enhanced functionality

Abb. 7: Die 7 ISO-OSI Schichten; Protokolle & Konzepte der Internet-Welt, die auf dieser Schichte "leben" oder ihr zugeordnet werden könnten, wobei zu beachten ist, daß das Internet dem ISO-OSI Modell nur in groben Zügen folgt; abstrakte Funktionalitäten im Überblick. Oft wird der 7. Layer auch oben gezeichnet.

3.1 Überblick

Die 7 Schichten des ISO-OSI Modells in Klassen gruppiert

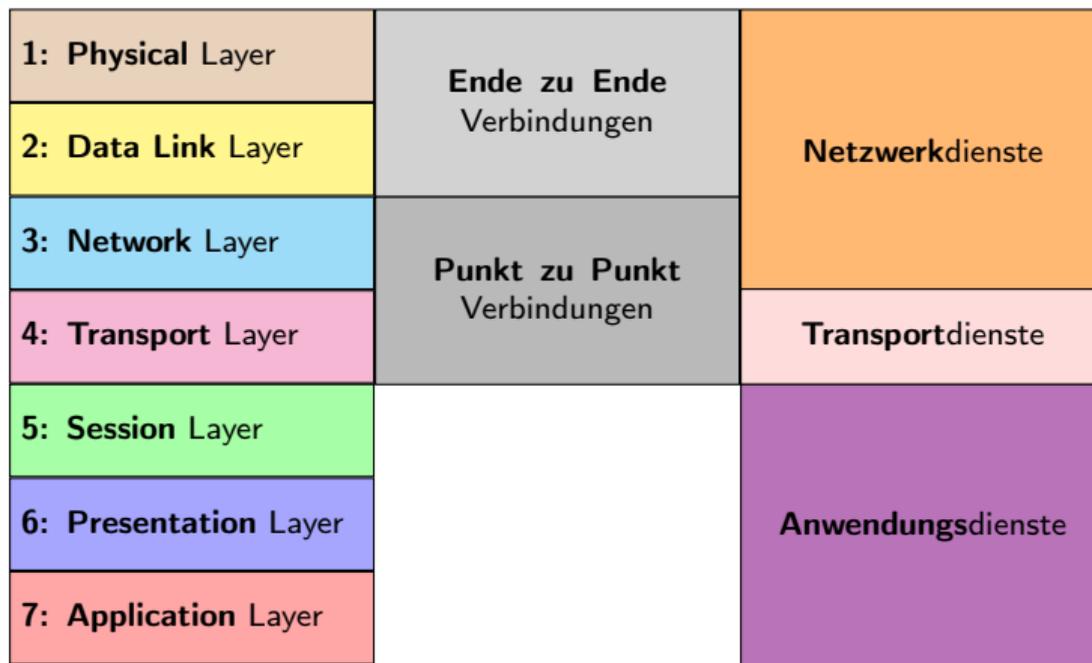


Abb. 8: Die 7 ISO-OSI Schichten in Klassen gruppiert.

3.1 Überblick

Schichtenmodell in Gesamtsicht

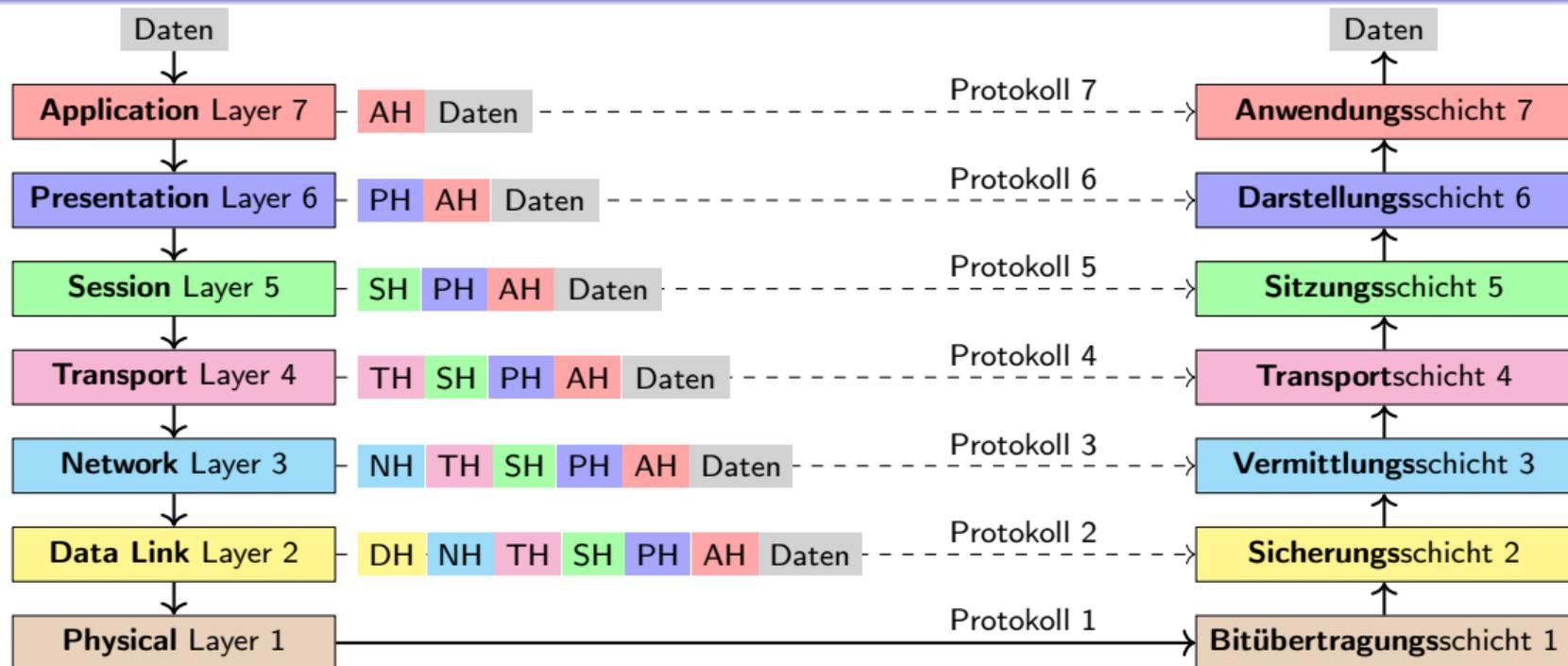


Abb. 9: Jede Schichte fügt (im Sender) auf dem Weg von Layer 7 zu Layer 1 ihren Header hinzu, auf dem Weg von Layer 1 zu Layer 7 (im Empfänger) werden die Header wieder entfernt; gestrichelte Protokolle sind virtuell.

3.1 Überblick

Eselbrücke

<u>P</u> lease	<u>P</u> hysical Layer
<u>D</u> o	<u>D</u> ata Link Layer
<u>N</u> ot	<u>N</u> etwork Layer
<u>T</u> hrow	<u>T</u> ransport Layer
<u>S</u> ausage	<u>S</u> ession Layer
<u>P</u> izza	<u>P</u> resentation Layer
<u>A</u> way	<u>A</u> pplication Layer

Physical Layer 1

Beschreibung

Bereitstellung einer physikalischen Konzeption für **Codierung**, **Transport** und **Decodierung** logischer Werte.

3 Kernaufgaben

- ① **Codierung:** Abbildung logischer Werte auf physikalische Größen.
Bsp: NRZ, Manchester.
- ② **Transport:** Übertragung physikalischer Größen von einem Ort zum anderen.
Medien-gebunden. Bsp: Telefonkabel.
Medien-frei. Bsp: Lichtsignal, Funk.
- ③ **Decodierung:** Rückinterpretation physikalischer Größen als logische Werte.
Bsp: Abtastung, Detektionsbereiche.

Data Link Layer 2

Beschreibung

Bereitstellung einer **fehlergesicherten** Übertragung über einen möglicherweise nicht fehlergesicherten Physical Layer.

5 Kernaufgaben

- 1 **Fehlerkontrolle:** Erkennen, Beheben oder Signalisieren von Übertragungsfehlern.
- 2 **Flußkontrolle:** Strategien, wenn Sender für Empfänger zu schnell sendet.
Bsp: Zwischenpuffern, Herunterregeln, warten Lassen.
- 3 **Segmentierung:** Zusammenfassen einzelner Bits zu größeren Einheiten.
Universal-typisches Bsp: 8-bit **Bytes**.
- 4 **Identifikation:** Auswahl bestimmter Sender oder Empfänger (sog. **Adresse**).
- 5 **Medienzugriff:** Aktivieren des Physical Layer.
4 typische Aspekte: Senden, Warten, Beheben und Reservieren.

Network Layer 3

Beschreibung

Bereitstellung einer Übertragung zwischen Geräten, die sich in **unterschiedlichen Netzwerken** (möglicherweise) verschiedener Layer 2 Technologien befinden.

3 Kernaufgaben

- ① **Adressierung:** Netzwerk-übergreifende Adressen.
- ② **Routing:** Auswahl der vermittelnden Zwischenknoten zur Kommunikation nicht direkt verbundenen Knoten.
- ③ **Flußsteuerung:** Strategien, wenn zu schnell gesendet wird.
 - End-to-End:** Vom Sender zum weit entfernten Empfänger.
 - Node-to-Node:** Von jedem Knoten zum Nächsten.

Transport Layer 4

Beschreibung

Bereitstellung einer Übertragung zwischen beliebigen **Teilnehmern** eines Netzes.

3 Kernaufgaben

- 1 **Adressierung:** Teilnehmer-Adressen.
- 2 **Flußsteuerung:** Nutzen der Teilnehmer-Charakteristik.
- 3 **Dienstqualität:** Sicherung der Quality of Service (QoS)
Erbringung eines Kommunikationsdienstes mit Garantien für Bandbreite, Latenz, Jitter, Fehlerrate usw.
Zugeschnitten auf den jeweiligen Teilnehmer.

4. Kernkonzepte

4.1. Adressen

4.2. Teilnehmer

4.3. Verbindung und Vermittlung

4.4. Sitzung und Zustand

4.5. Darstellung und Leserichtung

Ziel: Einige zentrale Konzepte der Kommunikationswelt kennenlernen, die uns helfen, die Bedeutung bestimmter Architekturelemente besser einzuordnen.

1. Vereinbarungen

2. Schichtenmodelle

3. ISO-OSI, 1. Teil

4. Kernkonzepte

5. ISO-OSI, 2. Teil

6. Internet Architektur

Konzept der Adresse (1)

Definition einer Adresse

Adressen **identifizieren** an einer Kommunikation teilnehmende **Komponenten**.

Physikalische "Adresse" (OSI-1)

Identifiziert eine Komponente anhand des physikalischen Übertragungswegs.

Bsp: Nummer des Kabels in der Wand.

Bsp: Frequenz des genutzten Kanals.

Link-Layer (MAC) Adresse (OSI-2)

Identifiziert ein Endgerät innerhalb einer Technologie (eines "Netzes").

Typischerweise im MAC (Media Access-Sublayer) von OSI-2: MAC Adresse

Bsp: Ethernet-Adresse.

Bsp: Bluetooth-Adresse.

Konzept der Adresse (2)

Internet Adresse (OSI-3)

Identifiziert ein Endgerät netzübergreifend.

Muß zusätzlich zu den MAC-Adressen vergeben werden.

Bsp: Internet-Adresse (IP-Adresse).

Teilnehmer Adresse (OSI-4)

Identifiziert einen Teilnehmer (statt nur ein Endgerät)

Bsp: Port-Nummer (bzw: IP –Adresse plus Port-Nummer)

Sitzungs-Adresse (OSI-5)

Identifiziert eine Sitzung eines Teilnehmers.

Bsp: Session-Id in einem Cookie oder im URL-query Teil.

Anwendungs-Adresse (OSI-7)

Identifiziert ein Element des Anwendungs-Protokolls.

Bsp: eMail Adresse (bei SMTP, POP-3, IMAP-4 Protokoll).

Bsp: URL (bei HTTP, FTP Protokoll).

Inhaltsdeskriptoren:

Jeder Teilnehmer sendet Inhalte an alle anderen Teilnehmer.

Die Inhalte haben Deskriptoren der Nutzlast.

Teilnehmer betrachten nur jene Informationen an denen sie aufgrund der Deskriptoren interessiert sind.

Bsp: CAN-Bus:

Je nach Version 2^{11} oder 2^{29} verschiedene Deskriptoren möglich.

Gruppen-Adressen

Eine Adresse kann auch **mehrere Kommunikations-Partner** identifizieren.

Bsp: MAC-Broadcast

Adressiert **alle** Endgeräte eines (OSI-2) Netz(bereichs).

Bsp: IP-Multicast

Adressiert alle Endgeräte einer Gruppe.

Anmeldung als Mitglied an einer Gruppen-Adresse erforderlich.

Bsp: Anycast

Adressiert einen Partner aus einer Gruppe, denen alle dieselbe Adresse zugeordnet ist.

Typischerweise erfolgt Kommunikation mit dem nächstgelegenen Partner.

Bsp: Unicast

Bezeichnet keine Gruppen-Adresse sondern die klassische Situation, in der eine Adresse *genau einen* Kommunikations-Partner identifiziert.

Konzept des Teilnehmers (1)

Kommunikationspartner auf Layer 3: Endgerät.

- Jedes Gerät hat (nur) eine Adresse.
- Rechner, Router, Drucker, Lichtschalter, IoT-Gerät.
- Ausnahme 1: Geräte mit mehr als einer Netzwerk-Karte.
- Ausnahme 2: Netzwerk-Karte mit mehr als einer Adresse.

Kommunikationspartner auf Layer 4: Teilnehmer.

- Historisch diverse Entwicklung in Telekommunikations- und Server-Welt.
- Bedürfnis nach Untergliederung des Adreßraums im Endgerät.

Telekommunikations-Welt: Aufteilung eines Endgeräts in mehrere Funktionen:

Neben "Telefon" treten "Fax, Anrufbeantworter, Kurznachricht".

Zunächst sind es alles einzelne Geräte.

Zur Hauptnummer gibt es unterschiedliche Durchwahlnummern.

Schließlich sind es Dienste auf einem Gerät.

Konzept des Teilnehmers (2)

Server-Welt: Mehrere Services auf einem Server.

Notwendigkeit, die einzelnen Services anzusprechen, obwohl Server nur eine Server-Adresse hat.

Neues Konzept: Port-Nummer oder auch Port-Adresse.

Server an IP-Adresse [192.168.0.1](#) bietet Dienste unter **Ports** an:

- [192.168.0.1:80](#) Web-Service,
- [192.168.0.1:25](#) Outgoing Email-Service
- [192.168.0.1:2049](#) File Service (NFS v1)
- [192.168.0.1:111](#) Portmapper: Konvertiert Dienstname in Port Nummer

4.3 Verbindung und Vermittlung

Paketvermittlung

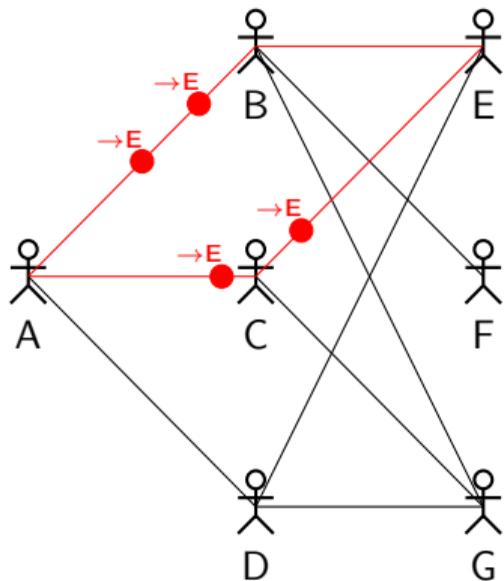


Abb. 10: Paketvermitteltes Netz.

Paketvermittlung:

- Pakete suchen sich auf **vielen Wegen** durch das Netz ihr Ziel.
- Pakete werden durch **Zwischenknoten** weitervermittelt.
- Pakete können sich dabei **gegenseitig überholen**.
- Bei hohen Verlustraten können ggf. auch mehrere Versendungen auf mehreren Wegen durchgeführt werden.
- Diese werden je nach Protokoll aber nur einmal ausgeliefert.

4.3 Verbindung und Vermittlung

Leitungsvermittlung

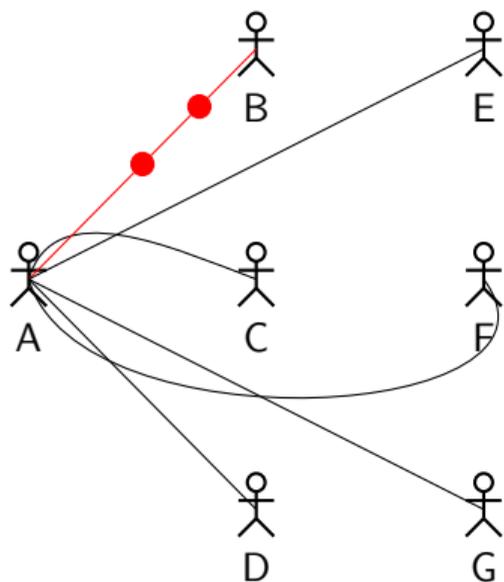


Abb. 11: Leitungsvermittelltes Netz

Leitungsvermittlung:

- Sender hat zu jedem Empfänger seine eigene Verbindung.
- Sehr hoher Leitungsaufwand!
- Senden bedeutet Auswahl der richtigen direkten Leitung.
- Pakete können sich nicht gegenseitig überholen.

4.3 Verbindung und Vermittlung

Simulierte Leitungsvermittlung

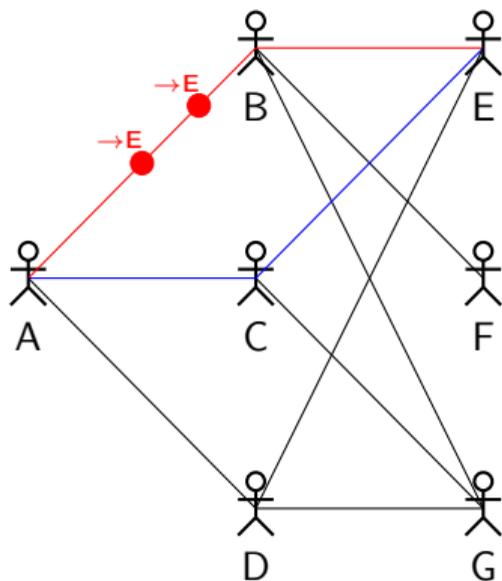


Abb. 12: Simulierte Leitungsvermittlung

Simulierte Leitungsvermittlung:

- Vor dem Senden wird auf einem paketvermittelten Netz eine **virtuelle Verbindung** (sog. virtual circuit) bestimmt und logisch "durchgeschaltet".
- Diese virtuelle Verbindung wird wie bei Leitungsvermittlung genutzt.
- Bricht diese Verbindung zusammen, kann auf eine **andere** gewechselt werden, die bereits vorgehalten wird (sog. alternative routing) oder dynamisch neu bestimmt wird (sog. adaptive routing).
- Versuch, die Vorteile beider Welten zu nutzen.

4.3 Verbindung und Vermittlung

Netzdienste

Unabhängig davon, wie das Netz vermittelt, können verschiedene Netzdienste zur Verfügung gestellt werden.

Verbindungslose Netzdienste (Connectionless Network Services)

Vorstellung: Einzelne Datenpakete auf ihrem Weg durch das Netz.

Individueller Versand einzelner Pakete an ihre Empfänger.

Typisch für paketvermittelte Kommunikation.

Verbindungsorientierte Netzdienste (Connection Oriented Network Services)

Vorstellung: Eine Leitung verbindet zwei Kommunikationspartner.

Erfordert in paketvermittelten Netzen den Aufbau virtueller Verbindungen.

Frage: Welche Eigenschaften muß man kennen, damit man die

"Semantik" eines Netzdienstes beschreiben kann?

5 Eigenschaften von Netzdiensten (1)

Integrität:

- Wenn Daten ankommen, dann kommen sie ohne Fehler an.
- **Mögliches Problem:** Störungen in Bits

Reihenfolge erhalten:

- Die Reihenfolge der Daten bleibt erhalten.
- **Mögliches Problem:** Daten nehmen verschiedene Wege durch das Netz. Früher abgesandte Daten können später ankommen als später abgesandte.

Verlust-frei:

- Alle Daten kommen an.
- **Mögliches Problem:** Daten gehen wegen technischer Fehler verloren oder werden wegen Überlastung bewußt fallengelassen.

5 Eigenschaften von Netzdiensten (2)

Verdoppelungs-frei:

- Einmal abgesandte Daten kommen auch nur einmal an.
- **Mögliches Problem:** Aus Gründen der Redundanz, Fehlerbehebung oder Effizienz werden Daten manchmal mehrfach versendet und können dann mehrfach ankommen.

Kommata:

- Daten kommen mit jenen Grenzsetzungen (Pausen oder "Kommata") an, mit denen sie abgeschickt wurden.
- **Mögliches Problem:** Beschränkte Puffergrößen lassen es nicht zu, eine Datenmenge von 10 [TByte] als eine einzige Einheit zu übertragen. Die Datenmenge wird aufgeteilt. Zwischen den Teilen treten Wartezeiten auf, wodurch zusätzliche Grenzen entstehen.

Kerneigenschaften typischer Transportdienste

Eigenschaft	TCP	UDP
Integrität erhalten	Ja	Ja
Reihenfolge erhalten	Ja	Nein
Verlust-frei	Ja	Nein
Verdoppelungs-frei	Ja	Nein
Kommata erhalten	Nein	Nein

Tab. 1: Kerneigenschaften typischer Transportdienste

TCP: Transmission Control Protocol. Internet Layer 4 Protokoll.
Steht hier für den **typischen verbindungsorientierten** Transportdienst.

UDP: User Datagram Protocol. Internet Layer 4 Protokoll.
Steht hier für den **typischen verbindungslosen** Transportdienst.

Integritätsprüfung: Kann bei UDP kann auf Treiberebene abgeschaltet werden.

Kommata: Können wegen endlicher Puffer in gar keinem Protokoll erhalten werden.

4.4 Sitzung und Zustand

Sitzung als Konzept

Versuch einer Definition:

Teilnehmer kommunizieren mehrfach oder länger miteinander.

Manche mehrfache Kommunikationsakte gehören "zusammen".

Diese Zusammengehörigkeit definiert das (abstrakte) Konzept der Sitzung.

Bsp: Bestellung über Web-Browser

Mehrere Phasen: LogOn, Auswahl, Versand, Zahlung, LogOff.

Jede Phase kann auf einzelne Request-Response Transaktionen abgebildet werden.

Wie wird die Zusammengehörigkeit sichergestellt?

Alle Phasen als eine zusammengehörige Sitzung verstehen.

Bsp: Übertragung einer großen Datei

Kann als eine Sitzung verstanden werden.

Werden Synchronisationspunkte gesetzt (Bsp: 1. Hälfte übertragen),

dann kann die Verarbeitung im Fehlerfall dort wieder aufgenommen werden.

Problem der fehlenden Sitzung

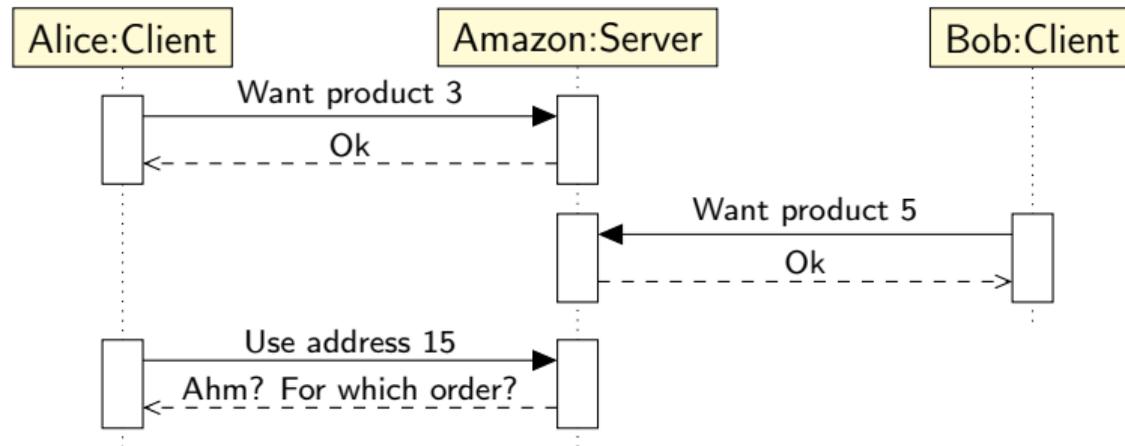


Abb. 13: Sequenz-Diagramm eines Kaufs. Mangels Sitzungs-Konzepts kann der Server verschiedene Teile der Transaktionen nicht korrekt zuordnen.

Sitzung mit Zusatz-Information

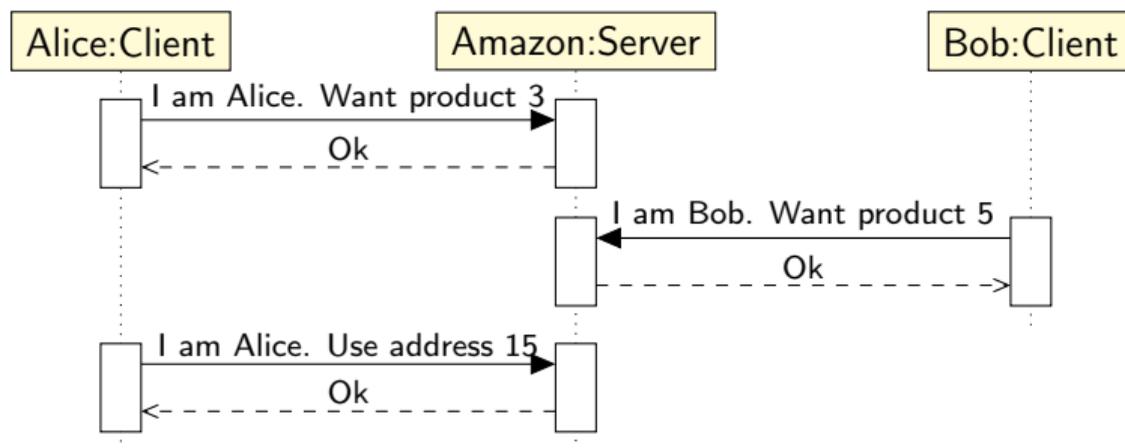


Abb. 14: Der Server kann die Transaktionen zuordnen, wenn die **Clients zusätzliche identifizierende Informationen** übermitteln. Das kann beim Browser technisch auf viele Arten geschehen: (1) in einem **Cookie**, (2) als **Teil der URL**, (3) als **Query-Parameter** am Ende der URL oder (4) in einem **HTTP-Headers**. Die Informationen sollen Session-Ids sein. IP-Adressen wären schlecht, da dann Transaktionen mit derselben IP-Adresse nicht unterschieden werden können.

Sitzung mit gefälschter Zusatz-Information

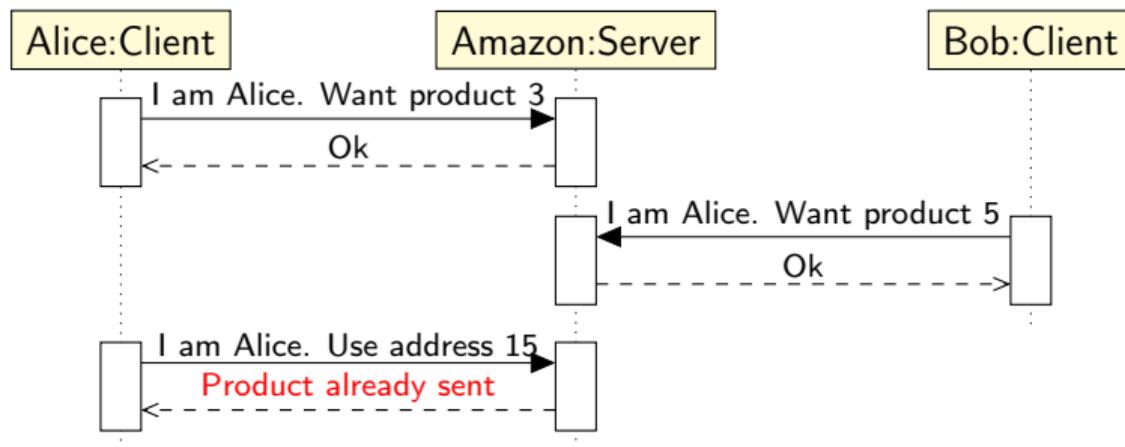


Abb. 15: Wahl des Benutzer-Namens als Sitzungs-Schlüssel ist schlecht: Er kann von anderen Clients **geraten** oder **zufällig durchprobiert** werden und so zu einer falschen Zuordnung führen.

Sitzung mit sicherer Session-Id

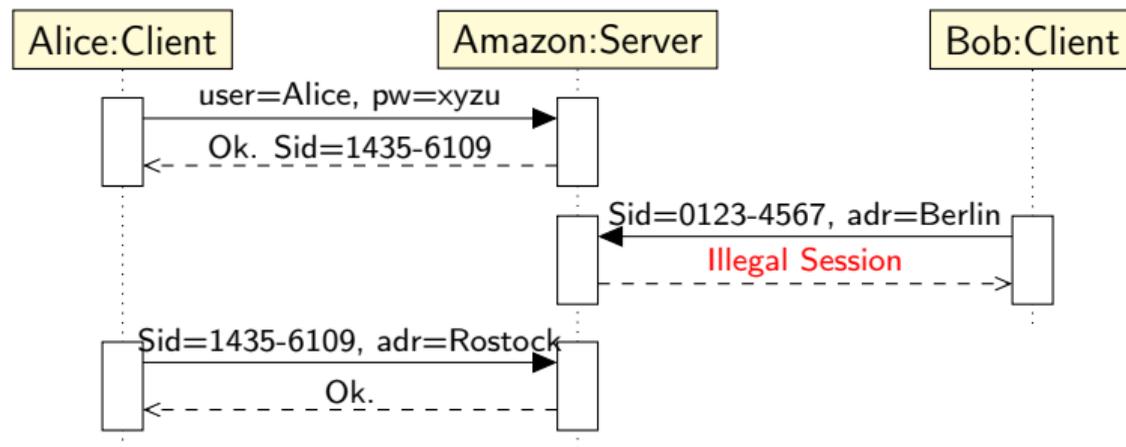


Abb. 16: Gute Session-Ids sind (1) vom Server vergebene, (2) lange, (3) zufällige Identifikatoren, die (4) verschlüsselt übertragen werden. Nur der berechtigte Client kann sie kennen. Nur dadurch ist eine sichere Konstruktion einer Session möglich. Gefälschte Sessions werden zurückgewiesen.

Sequenz-Diagramm mit langer Sitzung

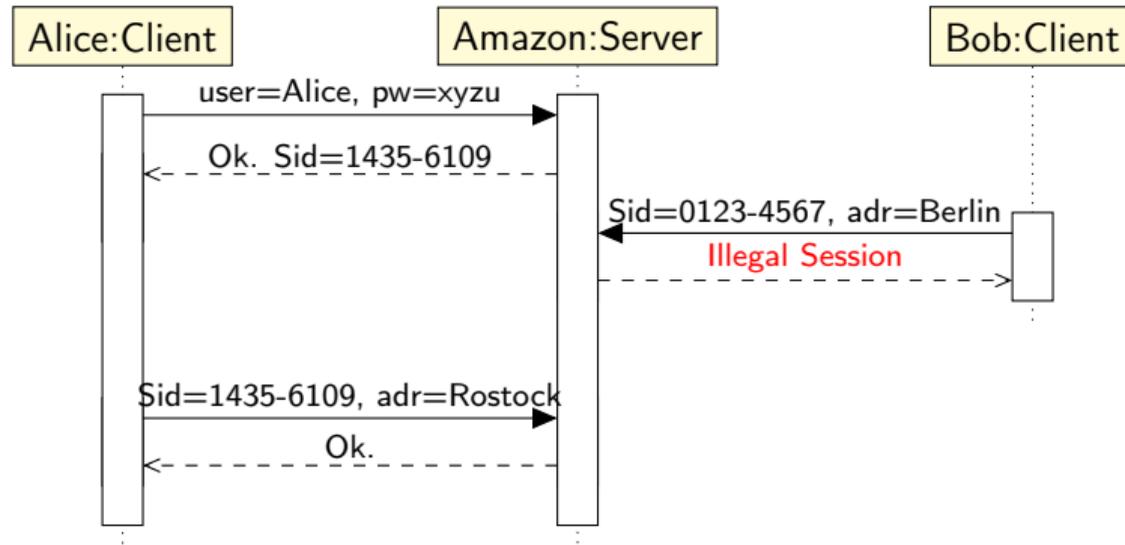


Abb. 17: Werden in einer länger andauernden Transaktion Sitzungen etabliert und aufrecht gehalten, so kann das in einem Sequenz-Diagramm auch durch eine **durchlaufend gezeichnete Aktivierung** angedeutet werden.

4.4 Sitzung und Zustand

Zustand als Konzept

Zustand (state) ist ein verwandter Begriff bei Protokollen.

Ein Protokoll heißt **zustandsfrei (state-less)**, wenn kein Teilnehmer Sitzungsinformation oder Zustandsinformation über den Protokollablauf speichert.

Unterscheide dabei:

- Zustand des Protokolls selber.
- Zustand eines Datenspeichers.

(ist hier gemeint)
(ist hier nicht gemeint)

Zustandsbehafteter Fileserver

Typische Schnittstelle:

- 1 Öffne Datei für einen Benutzer (Zustand: Geöffnet für Alice)
- 2 Stelle Dateizeiger auf bestimmte Position (Zustand: Offen, Dateizeiger)
- 3 Lesen oder schreiben (Zustand: Dateizeiger)

Bewertung: Viele Fehlerquellen, aber gut optimierbar.

- Datei leicht gegen gleichzeitige Zugriffe von anderen zu sichern.
- Muß Datei am Ende wieder schließen.
- Kann Schreibzugriffe bis zum Schließen hinauszögern.
- Was passiert bei Absturz vor dem Schließen der Datei?
- Kann mehrere Zugriffe zusammenfassen.
- Was macht Client-Programm, wenn Server "plötzlich weg ist"?
- In welchem Zustand ist dann die Datei?
- Was passiert mit dem Zustand, wenn Komponente crasht, die Zustand speichert?
- Aufteilung der Last auf mehrere Server sehr schwierig.

Typische Schnittstelle:

- 1 Schreibe Zeichen X oder lese an Position Y .
- 2 Sehr schlanke Schnittstelle.

Bewertung: Einfacher und einfacherer Wiederanlauf nach Fehlern.

- Kein Aufwand für Reservation, Prüfung, Management, Freigabe von Zuständen.
- Kein Aufwand für die Speicherung des Zustands.
- Aufgaben sind "immer gleich", da sie nicht von einem Zustand abhängen.
- Einfaches Protokoll.

4.4 Sitzung und Zustand

Idempotente Operation

Eine Operation heißt **idempotent** (lat. gleich-stark), wenn mehrfache Ausführung denselben Effekt hat wie einmalige Ausführung.

Beispiel: Setze Wert auf 0

Gegenbeispiel: Erhöhe Wert um 3.

Beispiel: Wenn Wert jetzt 7 ist, dann setze ihn auf 10.

Besser: Wenn Wert Versionsnummer 318 hat, dann setze Wert auf 10 und Versionsnummer auf 319.

Fazit: So kann man eine Operation idempotent machen.

Vorteil: Ist man sich nicht sicher, ob eine idempotente Operation korrekt ausgeführt wurde, so kann man sie problemlos ein weiteres Mal ausführen.

4.5 Darstellung und Leserichtung

Endian-ness

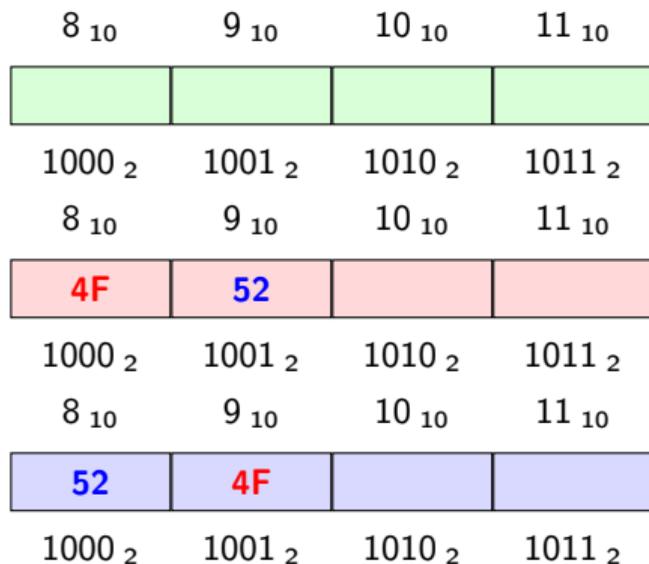
Bei der Ablage linearer Daten-Strukturen im Speicher, die mehr als ein Byte (oder auch: Wort) benötigen, sind 2 Vorgehensweisen möglich.

Vergleiche Mensch: links-nach-rechts oder sknil-hcan-sthcer.

- **Big end** jene Stelle einer Zahl, die bei Änderungen den **größten** Einfluß hat
sog. **most significant digit**.
- **Little end** jene Stelle einer Zahl, die bei Änderungen den **kleinsten** Einfluß hat
sog. **least significant digit**.

4.5 Darstellung und Leserichtung

Beispiel



Konvention: Wir lesen Speicher von links nach rechts mit aufsteigenden Adressen.

Zahlenbeispiel: $20306 = 0x4F52$

- big end ist 4F
- little end ist 52

Big Endian schreibt das big end zuerst.

Little Endian schreibt das little end zuerst.

```

#include <stdio.h>           // mache printf dem Compiler bekannt

int main() {                // definiere Haupteinstiegspunkt ins Programm
    short a=20306;          // short's sind 2 Byte; 20306 ist 0x4F52
    char*p = (char*)&a;    // ermittle Adresse auf a, in Byte-Auflösung
    printf ("%d\n", a);     // ergibt stets 20306
    printf ("%x\n", (unsigned int) &a);
                            // ergibt verschiedene Werte; ecad795e, e112295e usw.
                            // Wert wird immer gerade sein: 2-Byte Alignment
                            // Wert wird immer anders sein: Speicher Randomisierung

    printf("%x\n",*p);      // Ausgabe 52; Wert von a, als Byte uminterpretiert
    p++;                   // gehe zur nächsten Byte-Adresse
    printf("%x\n",*p);      // Ausgabe 4f; Wert von a, im nächsten Byte
}

```

Src. 1: Experiment zur Ausgabe des Zahlenwerts 20306 auf einem Little Endian Prozessor (hier: Intel Xeon E-5). Bei einem Big Endian Prozessor *sollte* die Reihenfolge der Ausgabe (52 und 4f) vertauscht sein. Aber Vorsicht: Benutzte Compiler und Sprachsysteme *können*, zumal bei virtualisierten Architekturen, auch noch eine Rolle spielen.

4.5 Darstellung und Leserichtung

Probleme der Endian-Ness

Hardware-Abhängigkeit: Programme einer Programmiersprache, welche einzelne Bytes separat adressieren können (Bsp: C, C++; Gegenbsp: Javascript) sind nicht ohne weitere Maßnahmen portabel.

Binäre Datenformate: Binäre Datenformate, die Zahlenwerte nicht als String ausgeben, erfordern ggf. Korrektur der Endian-Ness.

Netzwerk: Wenn Speicherinhalte direkt auf das Netz geschrieben werden und Knoten unterschiedliche Endian-Ness haben, sind Konvertierungsmechanismen erforderlich.

3 mögliche Strategien bei Netzwerken:

- **Sender makes it right:** Sender muß konvertieren.
- **Receiver makes it right:** Empfänger muß konvertieren
- **Network Data Format:** Standard für das Netz. Internet ist big-endian

Folge: Wenn zwei Intels (little-endian) kommunizieren, wird immer hin-und-her konvertiert.

5. ISO-OSI, 2. Teil

5.1. Die 3 oberen Layer

5.2. Abschließende Bewertung

Ziel: Die (aus Internet-Perspektive in gewisser Weise atypischen) 3 oberen Layer kennenlernen und das ISO-OSI Modell abschließend bewerten.

1. Vereinbarungen
2. Schichtenmodelle
3. ISO-OSI, 1. Teil
4. Kernkonzepte
5. **ISO-OSI, 2. Teil**
6. Internet Architektur

Session Layer 5

Beschreibung

Bereitstellung der Funktionen für das **Sitzungs(Session)-Konzept** einer Kommunikationsbeziehung.

3 Kernaufgaben

- ① **Sitzungen:** Aufbauen, verwalten, zurückweisen und beenden.
- ② **Synchronisation:** Setzen von Zwischenpunkten.
- ③ **Bündelung** Gruppierung zusammengehöriger Transaktionen.

Beschreibung

Kodierung (Darstellung) der Anwendungsdaten als Bitstrom.

4 Kernaufgaben

- 1 **Endian-Ness:** Konvertieren der *Datendarstellung*.
Bsp: little & big endian, weitere Serialisierungs-Formate.
- 2 **Zeichen:** Konvertierung der *Zeichendarstellung*
Bsp: ASCII, Unicode, EBCDIC, nationale Sonderzeichen.
- 3 **Kompression:** Komprimieren und Dekomprimieren.
- 4 **Verschlüsselung:** Verschlüsseln und Entschlüsseln.

Beschreibung

Festlegung, welche Elemente bei einer spezifischen **Anwendung** auftreten.

3 Kernaufgaben

- 1 **Grundkonzepte:** Welche Begriffe treten in der Anwendung auf.
- 2 **Protokollelemente:** Welche Anforderungen können gestellt werden?
Welche Daten werden dabei erwartet?
- 3 **Verhalten:** Wie reagiert die Anwendung auf die
Anforderungen durch diese Protokollelemente?

Stark abhängig von der jeweiligen Anwendung!
Daher wenig Systematisierung möglich.

Beispiele auf Anwendungs-Ebene

Bsp: Email

- Was sind: Email-Adresse, Email-Header, Attachment, Envelope usw.
- Was sind: Erforderliche Infrastruktur, postmaster@domain

Bsp: Virtuelle Terminal-Schnittstelle

- Für viele "klassische" Unix-Dienste verfügbar.
- Generische Dienststeuerung über Klartextbefehle.
- Was bedeuten: Befehlszeile, Statusmeldung, Header-Formate.

Bsp: Web / HTTP

- Wie sehen Anforderungen (requests) aus?
- Wie sehen Antworten (responses) aus?
- Welche Steuerinformationen (header) gibt es?
- Was ist noch wichtig?

Sehr viel...da mittlerweile umfangreiches, mehrfach überarbeitetes Protokoll.

Kritik des ISO-OSI Modells

Strikte Layer-Trennung manchmal nicht sinnvoll.

Bsp: Wenn Layer 1 zuverlässig ist, könnte Layer 2 Fehlerbehebung abschalten.

Bsp: Wenn Layer 7 Anwendung Realzeit-Verhalten erfordert, sollte Layer 3 anders routen.

Höherer Layer aber

- **kann** das mangels dafür vorgesehener Schnittstelle nicht wissen.
- **darf** es wegen Kapselung nicht außerhalb von Schnittstellen erfragen.
- **darf** in strikter Architektur nicht Zwischenschichten überspringen.

Abhilfe: Idee der sogenannten Cross-Layer Information.

Abseits praktischer Entwicklung:

IP wurde implementiert und lief (Praxis).

ISO-OSI parallel dazu als Standard eingeführt (Theorie).

IP Architektur: Auf ISO-OSI-Modell nur teilweise abbildbar.

Alleinherrschaftsanspruch: ISO-OSI wollte alle Probleme der Kommunikation lösen.

Es gibt mittlerweile aber bessere Ansätze und veränderte Anforderungen.

Referenzarchitekturen der Kommunikation

- ISO-OSI:** Theoretisches, konzeptuell gut durchdachtes Modell.
- Internet:** Praktisch funktionierendes System.
- Signaling System 7:** Referenzarchitektur der Telekommunikation.
- SNA:** Systems Network Architecture (IBM proprietär).
- Heute:** Zunehmend Konvergenz auf IP, speziell IP v6.
- Grund:** Einheitlicher Standard wichtiger als spezialisierte Nischenlösung.
- Weltweite **Interoperabilität** aller Geräte.
 - Aufwand der **Einführung** / **Änderung** bei 50 Milliarden vernetzten Geräten.
 - **Komplexität** eines Netzwerk-Stacks sehr hoch (speziell: Fehlersuche).
 - **Ausnahmen:** Spezialanwendungen bei Weltraumfahrt, NASA.

6. Internet Architektur

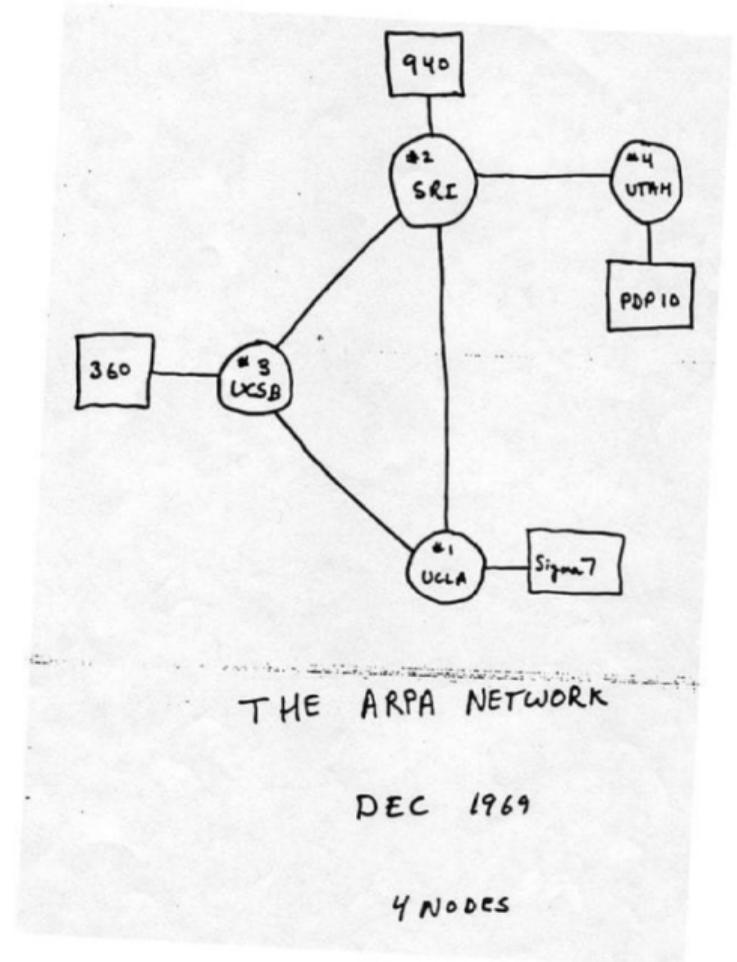
Ziel: Wie sortiert sich in diese schöne und abstrakte Theorie die Lebenspraxis der Internet-Kommunikation ein?

1. Vereinbarungen
2. Schichtenmodelle
3. ISO-OSI, 1. Teil
4. Kernkonzepte
5. ISO-OSI, 2. Teil
- 6. Internet Architektur**

Historie des Internet **gänzlich anders** als bei ISO-OSI.

- Viel früher (Geburtstag des Internet: 29. 10. 1969)
- Militärisch-akademischer Ausgangspunkt.
- Netz mit einer Architektur, das einen nuklearen Angriff auf die USA überdauert.

Abb. 18: Die ersten 4 Knoten am Internet, das im Dezember 1969 noch ARPA Net hieß nach der **A**dvanced **R**esearch **P**roject **A**gency des amerikanischen Verteidigungsministeriums. © Rechtsnachweis siehe Anhang



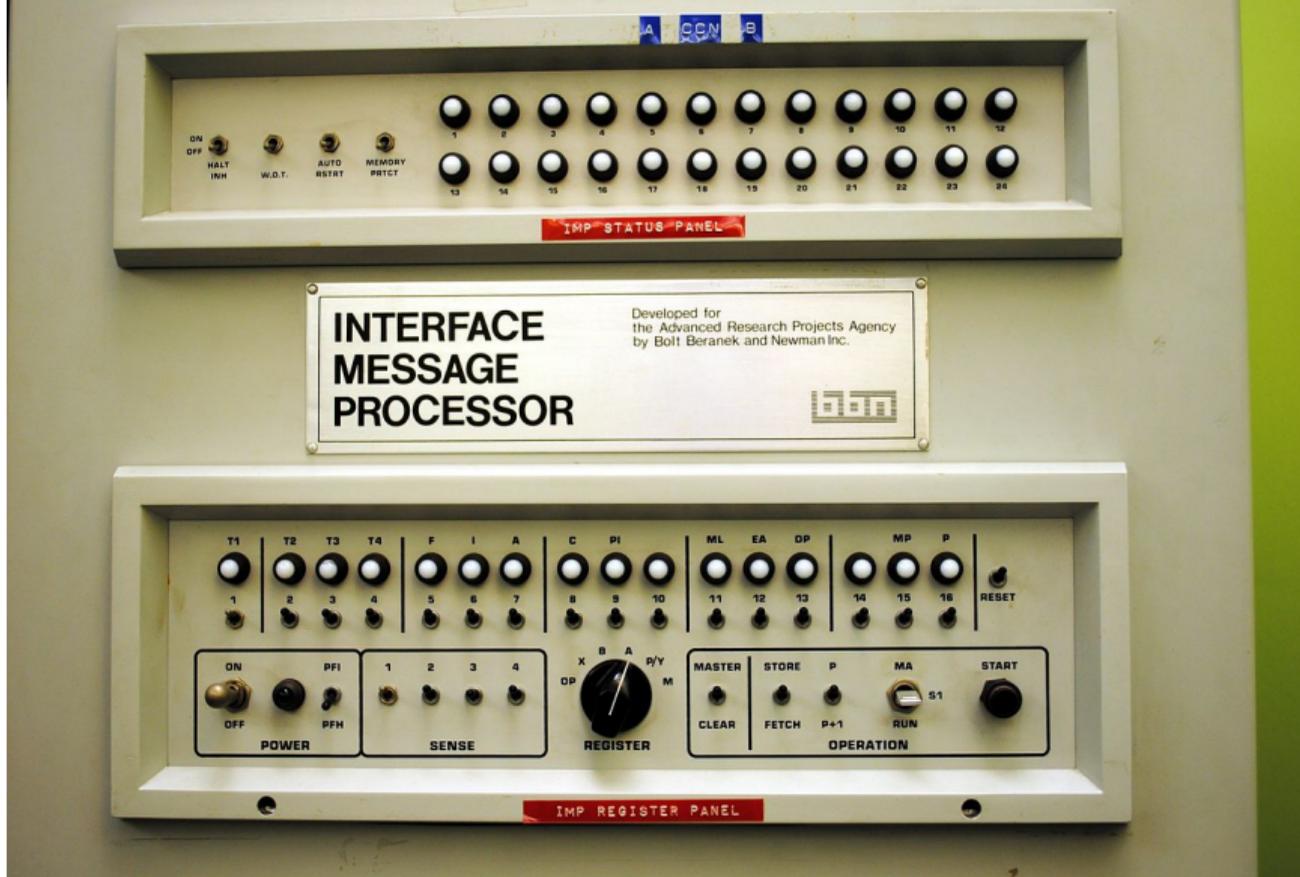


Abb. 19: Der Internet Message Processor war der erste Router des Internet © Rechtsnachweis siehe Anhang

Erstes Hochfahren des Internet

Am 29. Oktober 1969 wurde das Internet das erste Mal hochgefahren.

"We set up a telephone connection between us and the guys at SRI ...", Kleinrock said in an interview. "We typed the L and we asked on the phone..."

"Do you see the L?"

"Yes, we see the L," came the response.

We typed the O, and we asked, "Do you see the O."

"Yes, we see the O."

Then we typed the G, and the system crashed ...

Yet a revolution had begun ...

Tab. 2: Dialog beim ersten Hochfahren des Internet. Quelle: <https://top-ten.readthedocs.io/en/latest/event/10.html>

Internet Markt-Durchdringung

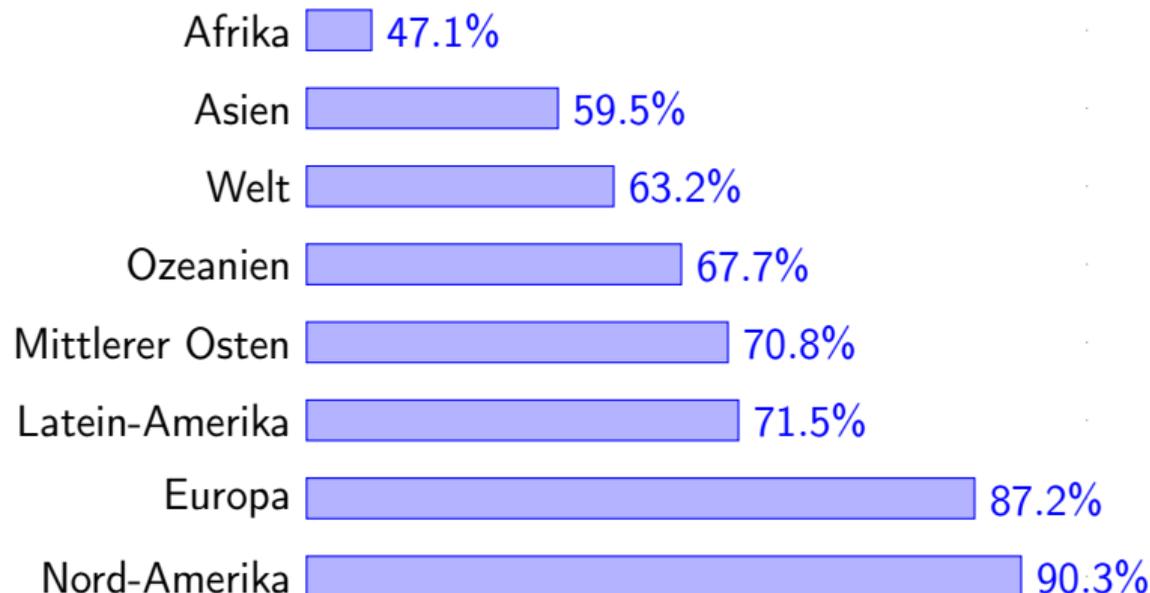


Abb. 20: Internet Markt-Durchdringung in Prozent der Bevölkerung, nach geographischen Regionen, 3. Quartal 2020. die prozentuelle Durchdringung reflektiert den technologischen Entwicklungsstand. Quelle: Internetworldstats.com

Internet-Bevölkerung

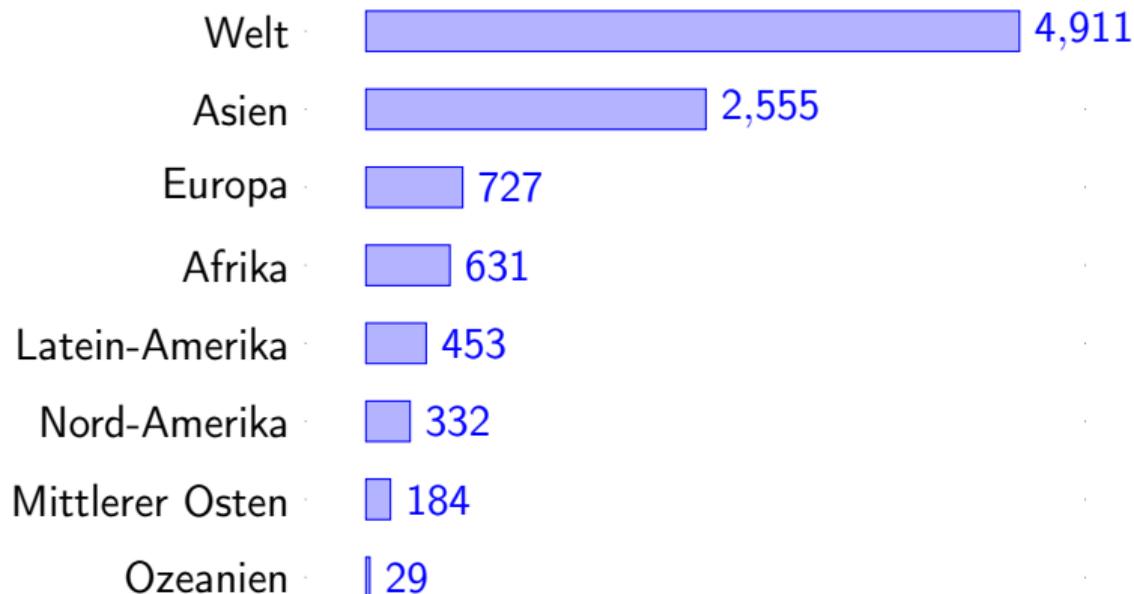


Abb. 21: Internet Bevölkerung in absoluten Zahlen (Millionen), nach geographischen Regionen, 3. Quartal 2020. Die absoluten Zahlen, deutlich anders als die prozentuellen Zahlen, weisen auf die Bedeutung als Markt hin. Quelle: Internetworldstats.com

6. Internet Architektur

Architektur-Modell

Viel weniger formalisiert als ISO-OSI.

In Literatur und Normung oft eine unterschiedliche Anzahl von Schichten.

- Kein eigener Physical und Data Link Layer.
- Setzt auf bestehende Technologien lokaler Netze auf.
- Unterster Layer ist ein **Koppelungs-Layer**, der Netzzugang herstellt.
- Weitere Layer nur **Inter**-Networking.
- Bsp: Einfach: Ethernet, serielle Schnittstelle.
- Bsp: Komplex: Bsp: 5G & Bluetooth.



Abb. 22: Die gebräuchlichste Form des Architekturmodells ist diese 4-schichtige Internet-Architektur. Sie kann nur teilweise mit dem ISO-OSI Ansatz in Verbindung gebracht werden. Auch die Zuordnung von Numerierungen wird damit gelegentlich schwierig.

Anhang

Übersicht

Programmquellenverzeichnis

Prog

Rechtsnachweise

©

Rechtliche Hinweise

§

Zitierweise dieses Dokuments

→

Index

Index

Verzeichnis aller Folien



- 1 Experiment zur Ausgabe des Zahlenwerts 20306 auf einem Little Endian Prozessor (hier: Intel Xeon E-5). Bei einem Big Endian Prozessor *sollte* die Reihenfolge der Ausgabe (52 und 4f) vertauscht sein. Aber Vorsicht: Benutzte Compiler und Sprachsysteme *können*, zumal bei virtualisierten Architekturen, auch noch eine Rolle spielen.....53

Verzeichnis aller Abbildungen

1 Pfahlbau Unteruhldingen am Bodensee.	7
2 Zeltanhänger mit aufgebautem Zelt.	7
3 Beispiel-Architektur zum Schichtenmodell	12
4 Schichtenmodell und Implementierung.....	13
5 Schichtenmodell und Implementierung.....	14
6 Dienst-Zugang und Methoden	15
7 7 ISO-OSI Schichten	19
8 7 ISO-OSI Schichten, Gruppierung	20
9 Schaubild zu Kommunikation	21

10	Paketvermitteltes Netz.....	34
11	Leitungsvermitteltes Netz.....	35
12	Simulierte Leitungsvermittlung.....	36
13	Sequenz-Diagramm ohne Sitzung.....	42
14	Sequenz-Diagramm mit Sitzung.....	43
15	Sequenz-Diagramm mit gefälschter Sitzung.....	44
16	Sequenz-Diagramm mit sicherer Session-Id.....	45
17	Sequenz-Diagramm mit langer Sitzung.....	46
18	ARPA Net im Dezember 1969.....	63
19	Der Internet Message Processor war der erste Router des Internet.....	64

20	Internet Markt-Durchdringung in Prozent der Bevölkerung, nach geographischen Regionen, 3. Quartal 2020. die prozentuelle Durchdringung reflektiert den technologischen Entwicklungsstand. Quelle: Internetworldstats.com	66
21	Internet Bevölkerung in absoluten Zahlen (Millionen), nach geographischen Regionen, 3. Quartal 2020. Die absoluten Zahlen, deutlich anders als die prozentuellen Zahlen, weisen auf die Bedeutung als Markt hin. Quelle: Internetworldstats.com	67
22	Internet-Architektur	68

1 Kerneigenschaften typischer Transportdienste	40
2 Erstes Hochfahren des Internet.....	65

Abb. 1 Von H. Zell - Eigenes Werk, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=50404836>

Abb. 2 Von Julian Nocke. JNocke in der Wikipedia auf Deutsch - Selbst fotografiert, CC BY-SA 2.0 de,
<https://commons.wikimedia.org/w/index.php?curid=30195880>

Abb. 18 Quelle:

<https://www.defensemedianetwork.com/stories/a-darpa-perspective-on-the-development-of-the-internet/>
Gemeinfreies Werk nach der US Government Works Copyright Exception.

Abb. 19 By FastLizard4 - Own work, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=17177042>

Die hier angebotenen Inhalte unterliegen deutschem Urheberrecht. Inhalte Dritter werden unter Nennung der Rechtsgrundlage ihrer Nutzung hier angeführt. Auf das Literaturverzeichnis wird verwiesen. Das **Zitat**recht in dem für wissenschaftliche Werke üblichen Ausmaß wird beansprucht. Wenn Sie eine Urheberrechtsverletzung erkennen, so bitten wir um Hinweis an den auf der Titelseite genannten Autor und werden entsprechende Inhalte sofort entfernen oder fehlende Rechtsnennungen nachholen. Bei Produkt- und Firmennamen können Markenrechte Dritter bestehen. Verweise und Verlinkungen wurden zum Zeitpunkt des Setzens der Verweise überprüft; sie dienen der Information des Lesers. Der Autor macht sich die Inhalte, auch in der Form, wie sie zum Zeitpunkt des Setzens des Verweises vorlagen, nicht zu eigen und kann diese nicht laufend auf Veränderungen überprüfen.

Alle sonstigen Inhalte unterliegen dem Copyright des Autors, Prof. Dr. Clemens Cap, ©2020.

Wenn Sie diese Inhalte nützlich finden, können Sie darauf verlinken oder sie zitieren. Jede weitere Verbreitung, Speicherung, Vervielfältigung oder sonstige Verwertung außerhalb der Grenzen des Urheberrechts bedarf der schriftlichen Zustimmung des Rechteinhabers. Dieses dient der Sicherung der Aktualität der Inhalte und soll dem Autor auch die Einhaltung urheberrechtlicher Einschränkungen wie beispielsweise **Par 60a UrhG** ermöglichen.

Die Bereitstellung der Inhalte erfolgt hier zur persönlichen Information des Lesers. Eine Haftung für mittelbare oder unmittelbare Schäden wird im maximal rechtlich zulässigen Ausmaß ausgeschlossen, mit Ausnahme von Vorsatz und grober Fahrlässigkeit. Eine Garantie für den Fortbestand dieses Informationsangebots wird nicht gegeben.

Die Anfertigung einer persönlichen Sicherungskopie für die private, nicht gewerbliche und nicht öffentliche Nutzung ist zulässig, sofern sie nicht von einer offensichtlich rechtswidrig hergestellten oder zugänglich gemachten Vorlage stammt.

Wenn Sie Inhalte aus diesem Werk nutzen oder darauf verweisen wollen, zitieren Sie es bitte wie folgt:

Clemens H. Cap: Referenzarchitekturen und Kernkonzepte. Electronic document.
<https://iuk.one/1010-1009> 28. 11. 2020.

Bibtex Information: <https://iuk.one/1010-1009.bib>

```
@misc{doc:1010-1009,  
  author      = {Clemens H. Cap},  
  title       = {Referenzarchitekturen und Kernkonzepte},  
  year        = {2020},  
  month       = {11},  
  howpublished = {Electronic document},  
  url         = {https://iuk.one/1010-1009}  
}
```

Typographic Information:

Typeset on November 28, 2020

This is pdfTeX, Version 3.14159265-2.6-1.40.21 (TeX Live 2020) kpathsea version 6.3.2

This is pgf in version 3.1.5b

This is preamble-slides.tex myFormat©C.H.Cap

1 Titelseite

1. Vereinbarungen

3 Notwendigkeit

4 Standards

5 Normen

6 Protokoll

7 Referenzarchitekturen im Bauwesen

8 Referenzarchitektur

9 Entwurfsmuster

2. Schichtenmodelle

2.1. Beispiel

11 Anforderung

12 Modell-Architektur

13 Beispiel zum Schichtenmodell

14 Beispiel zum Schichtenmodell

2.2. Abstraktes Schichtenmodell

15 Dienst-Zugang und Methoden

16 Prinzipien des Schichtenmodells

3. ISO-OSI, 1. Teil

3.1. Überblick

18 Positionierung

19 Die 7 Schichten des ISO-OSI Modells

20 Die 7 Schichten des ISO-OSI Modells in Klassen gruppiert

21 Schichtenmodell in Gesamtsicht

22 Eselsbrücke

3.2. Die unteren 4 Layer

23 Physical Layer 1

24 Data Link Layer 2

25 Network Layer 3

26 Transport Layer 4

4. Kernkonzepte

4.1. Adressen

28 Konzept der Adresse (1)

29 Konzept der Adresse (2)

30 Adress-freie Kommunikation

31 Gruppen-Adressen

4.2. Teilnehmer

32 Konzept des Teilnehmers (1)

33 Konzept des Teilnehmers (2)

4.3. Verbindung und Vermittlung

- 34 Paketvermittlung
- 35 Leitungsvermittlung
- 36 Simulierte Leitungsvermittlung
- 37 Netzdienste
- 38 5 Eigenschaften von Netzdiensten (1)
- 39 5 Eigenschaften von Netzdiensten (2)
- 40 Kerneigenschaften typischer Transportdienste

4.4. Sitzung und Zustand

- 41 Sitzung als Konzept
- 42 Problem der fehlenden Sitzung
- 43 Sitzung mit Zusatz-Information
- 44 Sitzung mit gefälschter Zusatz-Information
- 45 Sitzung mit sicherer Session-Id
- 46 Sequenz-Diagramm mit langer Sitzung
- 47 Zustand als Konzept
- 48 Zustandsbehafteter Fileserver
- 49 Zustandsfreier Fileserver
- 50 Idempotente Operation

4.5. Darstellung und Leserichtung

- 51 Endian-ness
- 52 Beispiel
- 53 ○ asd
- 54 Probleme der Endian-Ness

5. ISO-OSI, 2. Teil

5.1. Die 3 oberen Layer

- 56 Session Layer 5
- 57 Presentation Layer 6
- 58 Application Layer 7
- 59 Beispiele auf Anwendungs-Ebene

5.2. Abschließende Bewertung

- 60 Kritik des ISO-OSI Modells
- 61 Referenzarchitekturen der Kommunikation

6. Internet Architektur

- 63 ARPA Net 1969
- 64 Interface Message Processor
- 65 Erstes Hochfahren des Internet
- 66 Internet Markt-Durchdringung
- 67 Internet-Bevölkerung
- 68 Architektur-Modell

Legende:

-  Fortsetzungsseite
-  Seite ohne Überschrift
-  Bildseite