Architekturen (3)

Strukturen, Komponenten, Netze, Clouds

https://iuk.one/RNDS-ARCH-3.pdf

Clemens H. **Cap**
ORCID: 0000-0003-3958-6136

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

2026-11-22 18:58



Übersicht

1. Komponenten eines Netzwerks
2. Cloud
3. Home Automation

1. Komponenten eines Netzwerks

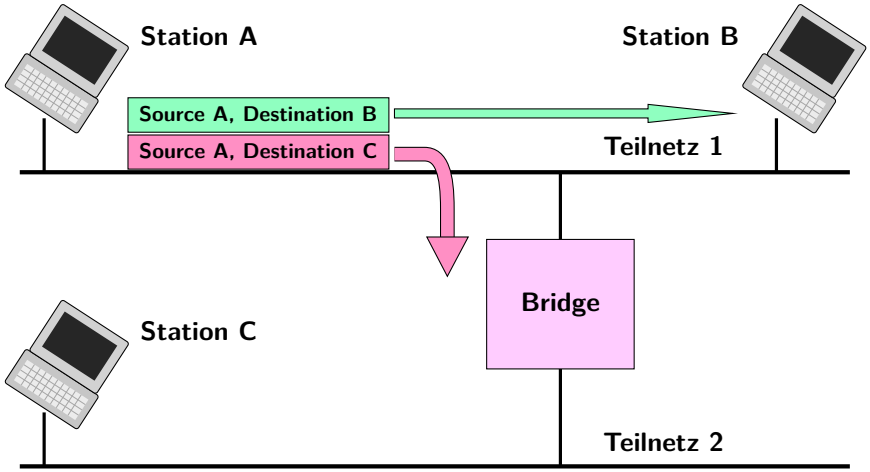
2. Cloud

3. Home Automation

1. Komponenten eines Netzwerks

Welche funktionalen Blöcke gibt es in größeren Netzen?

Bridge: Aufgabe



Bridge: Definition

Aufgabe einer Bridge

Selektives Weiterleiten von Paketen zwischen zwei Netzwerken derselben Layer 2 Technologie.

OSI: Link Layer 2.
Bridge ist für Layer 3 transparent (nicht erkenntlich).
Kann aber zur Konfiguration eine Layer 3 Adresse haben.

CSMA/CD: Teilnetze haben eigene Kollisions-Domänen.

Adressen: Müssen über alle Bridge-gekoppelten Teilnetze hinweg eindeutig sein.
Bridge lernt, welche Adressen auf beiden Seiten vorhanden sind.
Bridge kann Pakete nach Layer 2 Adressen filtern.

Bridge: Mechanismen

Lern-Mechanismus:

- Jedes Interface der Bridge hat eine Adreßtabelle.
- Adreßtabelle speichert alle an diesem Interface gesehenen Adressen.

Auslieferungs-Mechanismus:

- Adresse nicht gefunden: An alle Interfaces außer dem Ursprung senden.
- Adresse in Tabelle des Ursprungs: Paket verwerfen, wurde lokal verliefert.
- Adresse in Tabelle eines anderen Interface: Paket an dieses Interface senden.

Spanning Tree Mechanismus:

- **Problem:** Lern-Mechanismus kann bei mehreren Bridges zu Paketschleifen führen.
- Spanning Tree Algorithmus bewirkt Tabellenmodifikation, die das verhindert.

Transceiver, Bridge, Switch, Hub & Co (1)

Auf Layer-1 und Layer-2 finden sich diverse Geräte verschiedener Bezeichnungen.

Präzise Abgrenzung schwierig, Bezeichnungen Technologie- und Hersteller-spezifisch.

Transceiver: Rein physikalische Signalerzeugung.

Repeater: Weiterleitung auf Layer-1 durch Signal-Verstärkung.
Typischerweise Verbindung zweier Netzteile.

Hub: Sternverteiler auf Layer-1.
Empfangenes Paket an alle anderen verbundenen Netzteile weitergeleitet (1:n).

Konzentrator: Sternbündelung auf Layer-1 oder 2.
Bündelt viele Endgeräte an eine Netzanbindung. (n:1).

Transceiver, Bridge, Switch, Hub & Co (2)

Bridge: Verbindung zweier Netzteile.
Mehr oder weniger intelligente Auswahlmechanismen.

Switch: Verbindung zu mehreren Netzteilen, typisch sind 8 – 64
Mehr oder weniger intelligente Auswahlmechanismen.
Sehr ähnlich zur Multiport-Bridge, Unterschiede vor allem in Implementierung.
Switch ist typischerweise das "intelligentere" Gerät.

Weiterleitung bei Fehlern und Kollisionen

Cut Through: Weiterleitung, sobald Zieladresse gelesen.

- Geringe Verzögerungszeit (einige [μs]).
- Kollisionen und Fehler nicht erkennbar.
- Auch kollidierte und fehlerhafte Frames werden weitergeleitet.

Store-and-Forward: Weiterleitung erst nach Empfang des ganzen Pakets.

- Höhere Verzögerung, je nach Frame-Länge bis zu einigen [ms].
- Kollisionen und Fehler sind erkennbar.
- Nur fehlerfreie Frames weitergeleitet.

Adaptive: Anpassung an die bessere Strategie.

- Bei geringer Last: Cut-Through (nur wenige Fehler & Kollisionen zu erwarten).
- Bei hoher Last: Store-and-Forward (mehr Fehler & Kollisionen; diese verhindern).

Generelle Strategie bei Problemen (1)

Optimistische Strategie:

1. Normalsituation ist, daß kaum Probleme auftreten.
2. Erkenne Probleme und behandle sie, wenn sie aufgetreten sind.
3. Zusatzaufwand entsteht nur im Problemfall.

Pessimistische Strategie:

1. Normalsituation ist, daß öfter Probleme auftreten.
2. Verhindere im Vorfeld, daß Probleme entstehen.
3. Zusatzaufwand entsteht, auch wenn kein Problem aufgetreten ist.

Generelle Strategie bei Problemen (2)

Adaptive Strategie:

1. Messe Häufigkeit der Probleme.
2. Wechsle zwischen optimistischer und pessimistischer Strategie.

Beispiele:

1. Kollisionen: CSMA/CD oder Token-Ring
2. Cut Through oder Store-and-Forward
3. Datenbanken Edit-Konflikte

Router

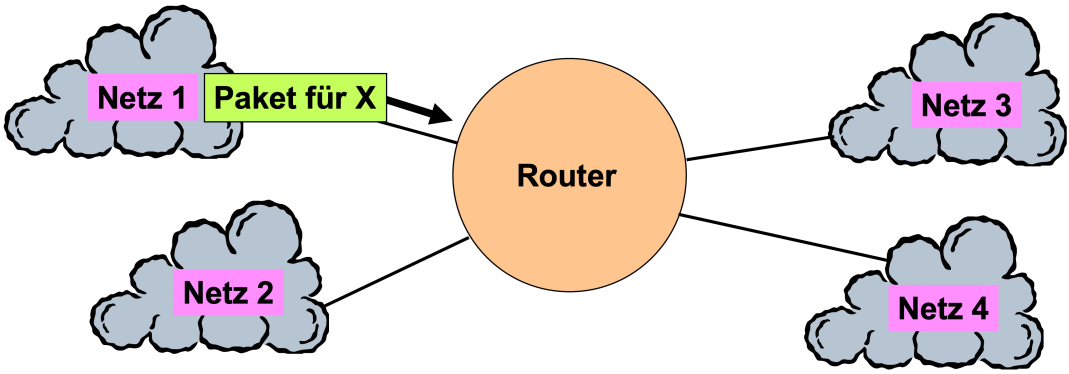


Abb. 1: Router erlauben Netzwerk-übergreifende Kommunikation zwischen Knoten, die in unterschiedlichen Netzwerken liegen. Diese können (müssen aber nicht) unterschiedliche Technologien der Kommunikation nutzen. Rechte s. Anhang.

Gateways

Aufgabe eines Gateways

Verbindung unterschiedlicher Protokoll-Welten.
Und dabei: Anpassung unterschiedlicher Sichtweisen aneinander.

OSI: Meist oberhalb Layer 4

Beispiele:

- **H.323 Gateway:** Verbindet VoIP (**v**oice **o**ver **IP**) / H.323 und PSTN (**p**ublic **s**witched **t**elephone **n**etwork)
- **Samba:** Setzt SMB (Windows) und NFS (Linux) Dateiaufrufe um.
- **OSI / SNA Gateway:** Setzt zwei komplette Stacks ineinander um.

NAT (Network Address Translation) Gateway (1)

Situation: Firma oder WG hat einen Internet-Anschluß mit einer IP-Adresse.
Dahinter befindet sich ein ganzes Netz mit vielen Endgeräten.

Problem 1: Outgoing NAT.
Wenn ein internes Endgerät einen Server außen kontaktiert
dann sieht dieser nur die eine IP-Adresse.
Wie findet die Antwort das anfragende Endgerät im internen Netz?

Problem 2: Incoming NAT.
Firma will dem Internet 3 Dienste auf je 1 Server anbieten.
Wie finden Anfragen aus dem Internet an die eine IP-Adresse
den jeweils dafür zuständigen Server?

Lösung: NAT Gateway beobachtet Pakete und übersetzt die Adressen und Ports.

NAT (Network Address Translation) Gateway (2)

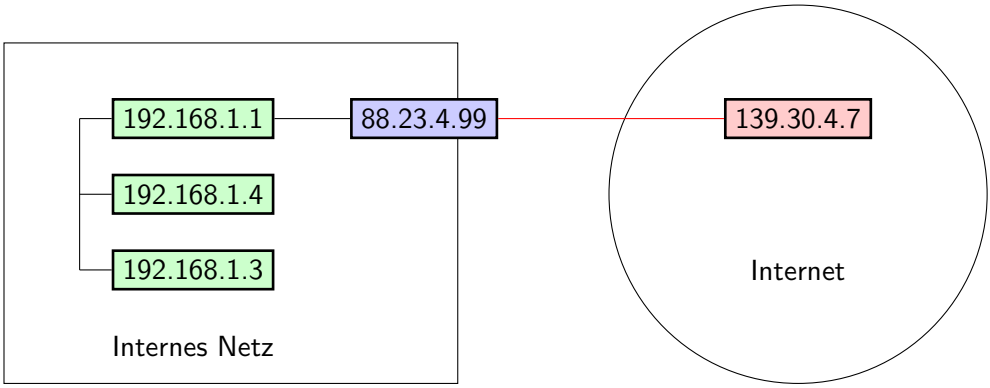


Abb. 2: Die **grünen Knoten** des internen Netzes können auf den **139-er Server** im Internet über das **88-er NAT Gateway** zugreifen. Der **139-er Server** sieht nur die Adresse des **88-er NAT Gateways**. Anfragen aus dem Internet werden an das **88-er NAT Gateway** gerichtet, das an die jeweils Port-weise konfigurierten grünen Knoten weiterleitet.

Rechte s. Anhang.

Firewall

OSI: Layer 2 und höher

Aufgaben: Unterbinden unerwünschter Kommunikation.

- Herausfiltern von Datenpaketen.
- Unterbinden verbindungsorientierter Dienste.
- Anpassen von Dienstqualität-Anforderungen an Regelvorgaben (policies).
- Meist auf Basis von Filterregeln.

Beispiele:

- Layer 2: Nach SRC oder DST Ethernet-Adresse filtern.
- Layer 3: Nach SRC oder DST IP-Adresse filtern.
- Layer 4: Nach SRC oder DST Portnummer filtern.
- Layer 7: Nach Inhalten filtern (zB: Java Applets, ActiveX, Inhalt mit unerwünschtem Hash).
- Layer 7: Nach Protokollheader filtern (zB: PICS Header für "adult" content).
- Layer 7: Nach Applikationsdaten filtern (Bsp: Subject der E-Mail).
- Layer 7: Nach Uhrzeit filtern (zB: Keine E-Mail zwischen 24 und 06 Uhr).

Proxy

OSI: Meistens Layer 4 und / oder 7

Aufgaben: Stellvertreter für einen Netzdienst.

Beispiele:

- **Web Proxy:** Web Seite vom lokalen Proxy geladen statt vom Web-Server.
  Bsp: Cache, Ad Blocking, Content Filter uvm.
- **TCP Wrapper:** Proxy auf Layer 4, protokolliert und filtert.
  Bsp: Zusätzliches Paßwort, Zugangslimitierung uvm.

Transparenz von Komponenten

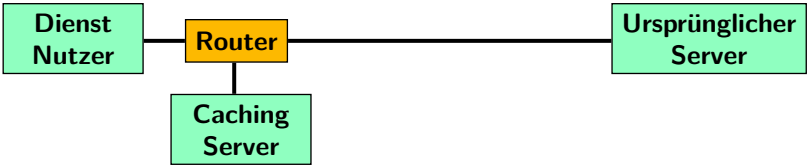


Abb. 3: Caching Server als transparentes Proxy geschaltet. Rechte s. Anhang.

Transparentes Proxy:

- Anfrage des Dienst-Nutzers wird vom Router an Caching Server weitergeleitet.
- Caching Server kann autonom bei Original Server anfragen.
- Benutzer merkt nicht, daß Antwort von anderem Gerät kommt als vom Server.

Intransparenz von Komponenten



Abb. 4: Intransparentes Proxy. Rechte s. Anhang.

Intransparentes Proxy:

- Client muß Proxy-Architektur unterstützen.
- Dienst-Nutzer aktiviert im Client einen Proxy-Server.
- Client kontaktiert Proxy-Server.
- Benutzer erkennt, daß Antwort von anderem Gerät kommt als vom Server.

2. Cloud

Wenn “Dein” (?) Computer
nur über das Netz erreichbar ist.

1. Komponenten eines Netzwerks

2. Cloud

3. Home Automation

3 Arten von Cloud Architekturen (1)

Infrastruktur as a Service (IaaS)

Bietet grundlegende IT-Ressourcen wie virtuelle Maschinen, Speicher oder Netzwerkkapazitäten.

Beispiele: Amazon EC2, MS Azure VMs, Google Computer Engine, DigitalOcean, Hetzner

Plattform as a Service (PaaS)

Bietet eine Entwicklungsplattform, auf der Anwendungen entwickelt werden können und laufen.

Beispiel: Google App Engine, MS Azure, Heroku

Software as a Service (SaaS)

Software über das Netz bereitgestellt, meist Browser / App, Anbieter macht Wartung & Update.

Beispiel: Google Workspace, Microsoft 365, OnlyOffice.

Achtung: Grenzen beginnen zu verschwimmen: (“Featureitis”).

3 Arten von Cloud Architekturen (2)

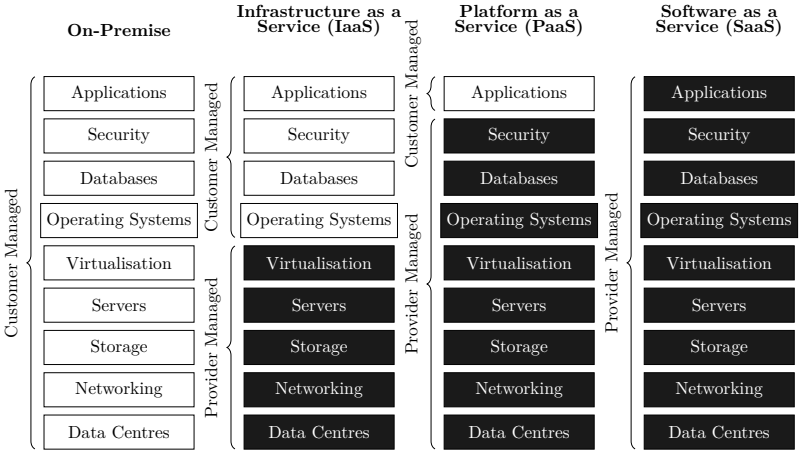


Abb. 5: 3 Arten von Cloud Architekturen und diverse Schichten.

Verwandter Begriff: Zero Install

Verwandter Begriff: Zero Install

Wesentliche SW-Komponenten werden über das Netz geladen.

Dadurch stets immer neueste Version vorgehalten.

Minimaler Aufwand für Konfiguration und Wartung.

Vorteile

Skalierbarkeit: Aufstocken der Server-Zahl per Mausklick oder Load Balancer.

Effizienz: Systeme sind fast immer ausgelastet. Gut für Kosten und Energie.

Wartungsfreiheit: Muß mich nicht um die ausgelagerten Themen selber kümmern. Bug-fixing, Updates, Konfiguration, Backup, Datensicherheit

Verfügbarkeit: Cloud selber meist redundant, daher meist besser verfügbar. Aber auch: Erhöhte Abhängigkeit.

Qualität: Fachunternehmen kann oft bessere Qualität als hauseigene IT.

Innovation: Zugang zu modernster Technologie, ohne diese selber zu kaufen.

Nachteile (1)

Vendor Lock-In:

- Schnittstellen sind oft stark Hersteller-abhängig.
- Programme, Datenformate und Abläufe werden an einen Cloud-Anbieter angepaßt.
- Wechsel zu anderen Anbieter danach schwierig.

Folge-Probleme des Vendor Lock-In:

- **Preisgestaltung**: Anbieter hebt Preis (zu) stark an
  Marketing Modell Drogenhändler: Anfixen – abzocken
- **End of Life**: Anbieter beendet die Dienstleistung
- **Continuity**: Anbieter geht in Konkurs oder hat technisches Problem
- **Netz**: Wenn instabil oder defekt, dann keine Dienste verfügbar

Nachteile (2)

Technik:

- **Latenz**: Für Realzeit-Anwendungen ev. zu hoch
  Bsp: Augmented Reality, Spiele
- **Datenrate**: Für Hochperformanz-Anwendungen ev. zu gering
  Bsp: Videos Rendern

Datenschutz:

- Angriff auf Cloud-Anbieter schlägt durch.
- Cloud-Anbieter hat selber direkten Zugriff auf die verarbeiteten Daten.

Lösungsansätze:

- Durch Rechtskonstruktion — Cloud-Anbieter darf nicht
- Durch Verschlüsselung — Cloud-Anbieter kann nicht

  Erfordert: Encryption (1) at rest, encryption (2) in transit **und** (3) Key-Management durch HSM Hardware Security Module.

Nachteile (3)

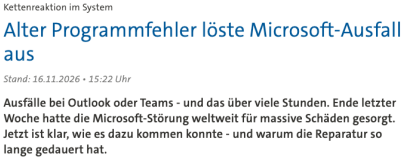
Abhängigkeit:

- Grundsätzliches Thema der technologischen Abhängigkeit.
- Verlust ansonsten vorgehaltener eigener Kompetenz.
- Entstehen monopolistischer Strukturen.
- Verlust von Daten- und Prozeß-Souveränität.
- Bsp: Abhängigkeit Europas von US / China hinsichtlich KI & Search
- Bsp: Abhängigkeit der Ukraine von Starlink
- Bsp: Abhängigkeit der USA von Taiwan hinsichtlich Chip-Fertigung

Abomodell statt Investition:

- Miete wo Eigentum möglich wäre.

Beispiele für Abhängigkeit (1)



Kettenreaktion im System

Alter Programmfehler löste Microsoft-Ausfall aus

Stand: 16.11.2026 • 15:22 Uhr

Ausfälle bei Outlook oder Teams - und das über viele Stunden. Ende letzter Woche hatte die Microsoft-Störung weltweit für massive Schäden gesorgt. Jetzt ist klar, wie es dazu kommen konnte - und warum die Reparatur so lange gedauert hat.

Abb. 6: Weltweiter Ausfall bei Microsoft 365 legt weltweit Email-Dienste lahm. Rechte s. Anhang.



Secretary Sean Duffy
@SecDuffy · **Follow**

The @FAAnews issued a nationwide ground stop for all U.S. airports following a communication issue impacting all Air Route Traffic Control Centers. FAA technicians are investigating the issue. We will provide immediate updates as we have them.

10:16 PM · Nov 13, 2026

33.7K Reply Copy link to post

Read 4.5K replies

Abb. 7: Ausfall der Flugsicherungs-Cloud groundet alle Flüge in den USA. Rechte s. Anhang.

Umfrage

Mein primärer Mail-Dienst ist:

1. Cloud-basierter Großanbieter (Gmail, MS, Yahoo, usw,)
2. Eigene Domäne, Server von Cloud-Anbieter (Mailbox, Proton usw.)
3. Eigene Domäne, eigener Server bei Cloud-Anbieter ()
4. Eigene Domäne, eigener Server als eigene Hardware in Colocation
5. Eigene Domäne, eigener Server, eigene Hardware, daheim.

https://twbk.de/2vxg

Beispiele für Abhängigkeit (2)



GESTRANDET

Fisker-Bankrott macht 3300 E-Autos möglicherweise zu Elektroschrott

Der insolvente US-Autohersteller kann die Serverdaten der Fahrzeuge nicht auf die Rechner eines möglichen Käufers transferieren

13. Oktober 2024, 15:56

Abb. 8: Cloud-Abhängigkeit im Internet of Things und Internet of Cars Rechte s. Anhang.

Beispiele für Abhängigkeit (3)

Was kann zum Problem bei softwarebasierten Fahrzeugen werden?

Fisker setzte auf ein softwarezentriertes Fahrzeugdesign, das eine ständige Verbindung zur Hersteller-Cloud erforderte, um grundlegende Funktionen wie das Batteriemanagement und sogar das Öffnen und Schließen der Türen zu steuern. Diese Entscheidung führte dazu, dass selbst kleinere Probleme nicht ohne die Serververbindung behoben werden können. Nach der Abschaltung dieser Server im Zuge der Insolvenz bleiben viele Fahrzeuge schlicht „gebrickt“, also unbrauchbar. Besitzer berichten, dass ihre Autos zu teuren „Rasenornamenten“ verkommen sind, da sie keine Möglichkeit haben, diese wieder in Gang zu setzen.

Abb. 9: Cloud-Abhängigkeit im Internet of Things und Internet of Cars Rechte s. Anhang.

Problem der Abo-Modelle (1)



Abb. 10: Kritische Karikatur zu Marketing Modellen. Das Problem bei KI ist zusätzlich, daß sie zu (1) einer **Degeneration bestehender Fähigkeiten** (Bsp: Sprachfähigkeit, Rechenfähigkeit), (2) zu einer **Rückentwicklung dazu erforderlicher Infrastrukturen** (Bsp: Suchmaschinen die nur suchen, klassische Bibliotheken mit Print-Angeboten) und damit zu (3) einer **fundamentalen Abhängigkeit** führen kann. Rechte s. Anhang.

Problem der Abo-Modelle (2)



Anthropic, Open AI und Meta füttern uns gerade an – bis wir nicht mehr ohne sie können: «KI wird viel teurer werden»

Netflix, Spotify – und nun KI. Internetdienste sind nur am Anfang günstig, danach erhöhen sie die Preise massiv. Zahlen wir bald Hunderte Franken pro Monat für Abos?

Moritz Kaufmann 22.11.2026, 05.30 Uhr 5 Leseminuten

Abb. 11: Die kritische Einstellung zu den Marketing-Mechanismen der KI wird auch von der Tagespresse geteilt. Rechte s. Anhang.

Rentabilität der Abhängigkeit



Abb. 12: Karikatur: Die kognitive Abhängigkeit von KI ist als Marketing-Konzept oft überraschend hilfreich.
Rechte s. Anhang.

Grundproblem der Cloud



Abb. 13: Das Grundproblem der Cloud. Die ökonomische Situation muß differenziert gesehen werden. Wie die Quelle, der dieses Bild entnommen ist, argumentiert: Für drei Jahre Hosting-Dienstleistung ist die Nutzung eigener Hardware fast drei Mal kostengünstiger als die Cloud Lösung. Aber: Der Beitrag stammt von 2019. Gleichwohl: Selber nachrechnen. Wenn: Man das noch kann. Siehe 34. Rechte s. Anhang.

Mögliche Auswege aus der Abhängigkeit

Open Data Lösungen:

- Datenformate sind dokumentiert und offen.
- Erlauben im Problemfall den Wechsel.
- Aber: Kompatibilität nicht immer praktisch gewährleistet. (Bsp: Powerpoint Format)
- Aber: Abwehrstrategie massiver Komplexitätserhöhung. (siehe Slide 37).

Open Source / **On Premise** Lösungen:

- Ermöglichen Installation auf eigener Hardware.
- Kombination mit Dual-Licensing Konzepten möglich
  Private Nutzung: Kostenfrei
  Kommerzielle Nutzung: Lizenzpflichtig
- Bsp: OnlyOffice.

Offene Dokumentenformate im Vergleich

Direkter Vergleich

Merkmal	Office Open XML (OOXML)	OpenDocument Format (ODF)
Spezifikationsumfang	Über 6.000 Seiten	Ca. 800–1.000 Seiten
Entwicklungsziel	Abwärtskompatibilität zu alten MS-Formaten	Schlanker, herstellerunabhängiger Standard
Verwendete Standards	Viele eigene XML-Dialekte (z. B. DrawingML)	Wiederverwendung offener Standards (z. B. SVG, MathML)
Implementierungsaufwand	Sehr hoch für Drittanbieter	Geringer, leichter vollständig umzusetzen

Durch diese enorme Detailtiefe ist OOXML für andere Programme (wie LibreOffice) schwerer vollständig fehlerfrei nachzubauen, garantiert aber innerhalb von Microsoft Office, dass selbst älteste Dokumente exakt so aussehen wie früher.

Abb. 14: Antwort von Gemini auf das Prompt: Welches der beiden [offenen Formate von Microsoft] ist das Komplexere? Rechte s. Anhang.

Umfrage

Denken Sie an ein System, in dem Sie die meisten und wichtigsten Notizen festhalten.

Nehmen Sie nun an, der Dienst wird plötzlich nicht mehr angeboten.

1. Ich habe ein Problem.
2. Ich habe ein Backup, das ich woanders einpflegen kann.
3. Ich habe ein Backup, das ich aber nicht woanders einpflegen kann.
4. Ich kontrolliere das System voll selber (Format und Source).



https://twbk.de/2vxg

Aufgabe: Kostenvergleich Cloud

Machen Sie einen Kostenvergleich für die Server-Lösung eines kleinen Unternehmens.

Treffen Sie dazu Annahmen über die Leistungsanforderung.

Recherchieren Sie (selber) bei Amazon EC2 die Cloud Kosten.

Recherchieren Sie (selber) die Kosten der Hardware.

Rechnen Sie das über die angenommene Lebensdauer der Hardware hoch.

Vergleichen Sie die eigene Berechnung mit den Antworten einer KI.

Wirtschaftlicher Vergleich von Unternehmen

Kennzahl	**Anthropic (NASDAQ: ANT)**	**Samsung Electronics (KRX: 005930)**	**Walmart Inc. (NASDAQ: WMT)**
Börsenwert (Market Cap)	**~1,16 Bio. USD** *(voll verwässert ~1,6 Bio. USD)*	**~867 – 888 Mrd. USD** *(~1.188 Bio. KRW)*	**~1,02 Bio. USD**
Letzter Jahresgewinn (Netto)	**–41,97 Mrd. USD** *(GJ 2025)*	**+31,79 Mrd. USD** *(GJ 2025 / 45,21 Bio. KRW)*	**+21,89 Mrd. USD** *(GJ 2026, per 31.01.2026)*
Nettoergebnis 9M 2026	**–104,00 Mrd. USD** *(Verlust)*	**~140 Mrd. USD** *(~193,58 Bio. KRW, Speicherboom)*	**+16,64 Mrd. USD** *(9M GJ 2027, per 31.10.2026)*
Jahresumsatz (Referenz)	**~54,5 Mrd. USD** *(Prognose 2026)*	**~220 – 230 Mrd. USD** *(GJ 2025)*	**~713,2 Mrd. USD** *(GJ 2026)*
Kurs-Gewinn-Verhältnis (KGV)	**Negativ** *(kein Jahresgewinn)*	**~5,7** *(TTM)*	**~45 – 48** *(TTM)*
Kurs-Buchwert-Verhältnis (KBV)	**Nicht ausweisbar** *(negatives Eigenkapital vor IPO)*	**~2,2 – 2,4**	**~9,7 – 10,3**

Abb. 15: Vergleich von Anthropic nach IPO mit Unternehmen der Halbleiter-Branche und der “normalen” Konsumgüte. Themen: Too-big-to-fail, reale Diskussion von KI-Gefahren und lokale Modelle (cloud-freie KI-sovereignty). Rechte s. Anhang.

3. Home Automation

Ein kurzer Überblick über Kommunikations-Protokolle der Home Automation zeigt:
Hier bestehen ganz andere architekturelle Konzepte, die sich aber auf ISO-OSI abbilden lassen.

1. Komponenten eines Netzwerks

2. Cloud

3. Home Automation

Home Automation

1. Bluetooth Low Energy
2. Lorawan
3. Zigbee
4. Z-Wave
5. DECT
6. Wifi
7. Enocean
8. KNX
9. Ethernet
10. Ethernet Zweidraht
11. Thread
12. ...

Selbe Phase wie beim Beginn der Rechnerkommunikation.

Proliferation proprietärer Standards und erster Experimente.

Verstehen Lernen der wesentlichen Themen

Thread

Drahtloses Mesh-Netzwerk

- Offener Standard
- WPAN (wireless personal area network)
- IEEE 802.15.4 auf dem 2.4 Ghz Band.

Sicherheit:

- AES-128 Verschlüsselung
- Erlaubt den Netzwerk-Beitritt nur für authentifizierte Geräte

Architektur:

- Benötigt keinen speziellen Hub
- Unterstützt Mesh Strukturen
- Selbstheilend: Wenn einzelne Geräte oder Pfade ausfallen.

Architektur von Thread (1)

Zwei **Rollen** von Knoten in der Vermittlung von Paketen:

1. **Router**: Leiten Daten weiter, Transceiver immer aktiv.
2. **End Device**: Kommuniziert mit einem Router (parent), Transceiver darf aus sein.

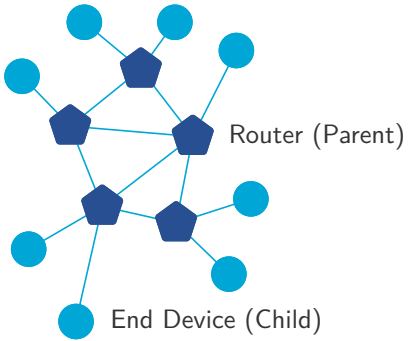


Abb. 16: Die zwei Rollen von Thread Knoten Rechte s. Anhang.

Architektur von Thread (2)

Zwei **Typen** von Knoten:

1. **Full Thread Device**: Transceiver muß immer an
   Hört auf Router Multicast Adresse
   - Router: Agiert als Router
   - Router Eligible end Device: Kann zum Router ernannt werden
   - Full End Device: Kann nicht zum Router ernannt werden
2. **Minimal Thread Device**: Transceiver muß nicht immer an
   Hört nicht auf Router Multicast Adresse
   - Minimal End Device: Transeiver immer an, pollt nicht den Parent
   - Sleepy End Device: Transceiver typischerweise aus, pollt gelegentlich

Architektur von Thread (3): Weitere Aufgaben

Leader: **Verwaltet** das Netzwerk

- Weiß wer Router ist.
- Stets ein Leader in jeder **Netz-Partition**.
- Rolle kann dynamisch vergeben werden.

Border Router: **Verbindet** mit anderen Netzen, etwa: Wifi, Ethernet.

Matter

Ziele:

- Herstellerübergreifende Interoperabilität
- Gemeinsamer Sicherheits-Standard
- Offen für weitere Hersteller und neue Konzepte
- Geringer Overhead

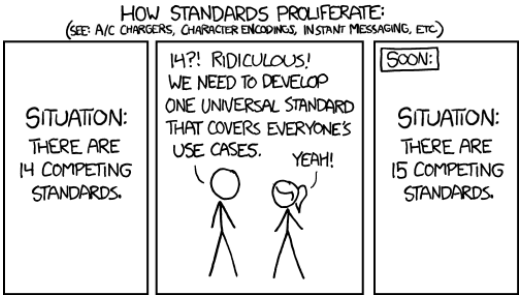


Abb. 17: xkcd 927: Das Schicksal übergreifender Interoperabilitäts-Standards. Rechte s. Anhang.

Matter: Schichten-Architektur der Kommunikation (1)

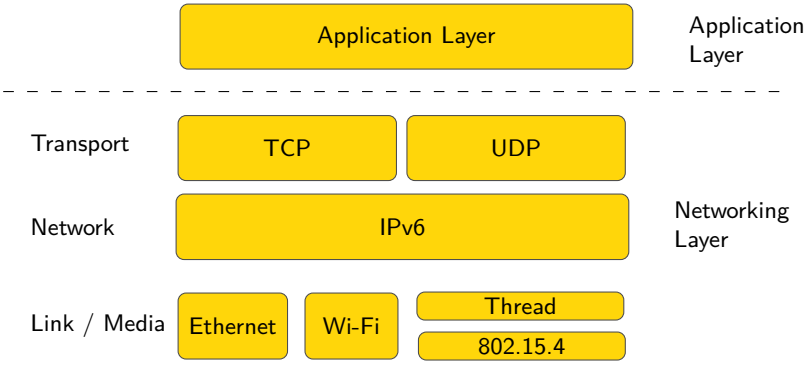


Abb. 18: Schichten-Architektur der Kommunikations-Protokolle in Matter.

Matter: Schichten-Architektur der Kommunikation (2)

Offenes Protokoll, 2021 neu eingeführt.

Anbindung nach unten:

- Auf OSI 4: TCP, UDP
- Auf OSI 3: IPv6
- Auf OSI 1/2: Ethernet, Wifi, Thread
- BLE: Für die Ersteinrichtung

Anbindung an andere Standards:

- Mit einer geeigneten **translational bridge**
  - Verbindet unterschiedliche Teilnetze (*bridge*)
  - übersetzt die dort gesprochenen Protokolle (*translational*)
- Beispiel: An Zigbee: Beispielsweise mit einer Philips Hue Bridge
- Beispiel: An Z-Wave: Beispielsweise mit einem SmartThings Hub

Matter: Schichten-Architektur der Konzepte (1)

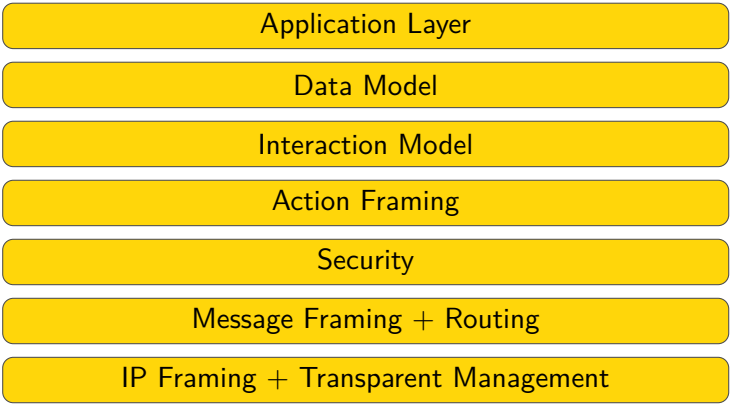


Abb. 19: Schichten-Architektur der Konzepte in Matter.

Matter: Schichten-Architektur der Konzepte (2)

Application:	Geschäfts-Logik Bsp: Lampe ist ein/aus, Farbwahl, Helligkeitswahl
Data Model:	Abb. Application Schicht auf *Daten* und *Verben*
Interaktion:	Abbildung der Data Model Schicht auf Client-Server Sicht bestehend aus read/write von Attributen.
Action Framing:	Serialisierung der Interaktion in ein Binärformat.
Security:	Verschlüsseln und signieren des Binärformats
Message Framing + Routing:	Header hinzufügen zum cryptogr. Binärformat
IP Framing + Transport Mgmt:	Übergabe des Frames an das Transport-Protokoll.

Appendix

Übersicht

Übersicht 53

Verzeichnis aller Abbildungen Abb 54

Rechtsnachweise © 56

Rechtliche Hinweise § 58

Zitierweise dieses Dokuments → 60

Verzeichnis aller Folien 61

Verzeichnis aller Abbildungen (1/2)

1 Aufgabe des Router 12
2 Network Address Translation 15
3 Caching Server als transparentes Proxy geschaltet. 18
4 Intransparentes Proxy. 19
5 3 Arten von Cloud Architekturen und diverse Schichten. 22
6 Ausfall von Outlook 28
7 Ausfall der Flugsicherung 28
8 Cloud-Abhängigkeit im Internet of Things und Internet of Cars 30
9 Cloud-Abhängigkeit im Internet of Things und Internet of Cars 31

10 Kritische Karikatur zu Marketing Modellen 32
11 headline NZZ November 2026 33
12 Karikatur zur Profitabilität kognitiver Abhängigkeit 34
13 Das Grundproblem und ökonomische Situation der Cloud 35
14 Offene Formate im Vergleich 37
15 Unternehmensvergleich 40
16 Die zwei Rollen von Thread Knoten 44
17 xkcd 927: Das Schicksal übergreifender Interoperabilitäts-Standards. 47
18 Schichten-Architektur der Kommunikations-Protokolle in Matter. 48
19 Schichten-Architektur der Konzepte in Matter. 50

Rechtsnachweise (1/2)

Abb. 1 Eigene Abbildung

Abb. 2 Eigene Abbildung

Abb. 3 Eigene Abbildung

Abb. 4 Eigene Abbildung

Abb. 6 Quelle: https://www.tagesschau.de/wirtschaft/unternehmen/microsoft-stoerung-fehler-100.html. Eigener Screenshot nach Zitatrecht.

Abb. 7 Quelle: https://www.flugrevue.de/zivil/faa-anordnung-landesweites-grounding-wegen-atc-problemen/. Eigener Screenshot nach Zitatrecht.

Abb. 8 Quelle: https://www.derstandard.de/story/3000000240487/fisker-bankrott-macht-3300-e-autos-moeglicherweise-zu-elektroschrott

Abb. 9 Quelle: https://www.all-electronics.de/automotive-transportation/warum-fisker-elektroautos-sich-in-schrott-verwandeln-519.html

Abb. 10 Eigene Karikatur, Visualisierung durch Eigenes Bild, Visualisierung durch ChatGPT, AI

Abb. 11 Quelle: Neue Zürcher Zeitung vom 22. 11. 2026 Eigener Screenshot nach Zitat-Recht.

Abb. 12 Eigene Karikatur, Visualisierung mit Eigenes Bild, Visualisierung durch ChatGPT, AI

Abb. 13 Quelle:Zitiert aus: https://blog.codinghorror.com/the-cloud-is-just-someone-elses-computer/

Abb. 14 Eigener Screenshot auf Gemini-Anfrage

Abb. 15 Daten einer Gemini-Abfrage, von Claude nach LaTeXübertragen.

Abb. 16 Quelle: https://openthread.io/guides/thread-primer/node-roles-and-types

Abb. 17 Quelle: https://xkcd.com/927/, Nutzung nach CC-ANC 2.5

Rechtliche Hinweise (1)

Die hier angebotenen Inhalte unterliegen deutschem Urheberrecht. Inhalte Dritter werden unter Nennung der Rechtsgrundlage ihrer Nutzung und der geltenden Lizenzbestimmungen hier angeführt. Auf das Literaturverzeichnis wird verwiesen. Das Zitatrecht 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, hier nicht angeführten 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.

Use of Logos and Trademark Symbols: The logos and trademark symbols used here are the property of their respective owners. The YouTube logo is used according to brand request 2-9753000030769 granted on November 30, 2020. The GitHub logo is property of GitHub Inc. and is used in accordance to the GitHub logo usage conditions `https://github.com/logos` to link to a GitHub account. The Tweedback logo is property of Tweedback GmbH and here is used in accordance to a cooperation contract.

Rechtliche Hinweise (2)

Disclaimer: Die sich immer wieder ändernde Rechtslage für digitale Urheberrechte erzeugt ein nicht unerhebliches Risiko bei der Einbindung von Materialien, deren Status nicht oder nur mit unverhältnismäßig hohem Aufwand abzuklären ist. Ebenso kann den Rechteinhabern nicht auf sinnvolle oder einfache Weise ein Honorar zukommen, obwohl deren Leistungen genutzt werden.

Daher binde ich gelegentlich Inhalte nur als Link und nicht durch Framing ein. Lt EuGH Urteil 13.02.2014, C-466/12 (Pressemitteilung, Blog-Beitrag, Urteilstext). ist das unbedenklich, da die benutzten Links ohne Umgehung technischer Sperren auf im Internet frei verfügbare Inhalte verweisen.

Wenn Sie diese Rechtslage stört, dann setzen Sie sich für eine Modernisierung des völlig veralteten Vergütungs- und Anreizsystems für urheberrechtliche Leistungen ein. Bis dahin klicken Sie bitte auf die angegebenen Links und denken Sie darüber nach, warum wir keine für das digitale Zeitalter sinnvoll angepaßte Vergütungs- und Anreizsysteme digital erbrachter Leistungen haben.

Zu Risiken und Nebenwirkungen fragen Sie Ihren Rechtsanwalt oder Gesetzgeber.

Weitere Hinweise finden Sie im Netz hier und hier oder hier.

Zitierweise dieses Dokuments

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

Clemens H. Cap: Architekturen (3). Electronic document, 22. 11. 2026 https://iuk.one/RNDS-ARCH-3.pdf

Bibtex Information: https://iuk.one/RNDS-ARCH-3.bib

```
@misc{doc:RNDS-ARCH-3,
    author       = {Clemens H. Cap},
    title        = {Architekturen (3)},
    year         = {2026},
    month        = {11},
    howpublished = {Electronic document},
    url          = {https://iuk.one/RNDS-ARCH-3.pdf }
}
```

Typographic Information:
Typeset on 2026-11-22 at 18:58
This is pdfTeX, Version 3.141592653-2.6-1.40.29 (TeX Live 2026) kpathsea version 6.4.2
This is pgf in version 3.1.11a
This is preamble-slides.tex ©C.H.Cap
Jobtype flag=foreground
Active modes (if any): slides, biblio
Keywords: keywords
Subject: subject

Verzeichnis aller Folien

Titelseite . . . 1
Übersicht . . . 2

1. Komponenten eines Netzwerks . . . 3

Bridge: Aufgabe . . . 4
Bridge: Definition . . . 5
Bridge: Mechanismen . . . 6
Transceiver, Bridge, Switch, Hub & Co (1) . . . 7
Transceiver, Bridge, Switch, Hub & Co (2) . . . 8
Weiterleitung bei Fehlern und Kollisionen . . . 9
Generelle Strategie bei Problemen (1) . . . 10
Generelle Strategie bei Problemen (2) . . . 11
Router . . . 12
Gateways . . . 13
NAT (Network Address Translation) Gateway (1) . . . 14
NAT (Network Address Translation) Gateway (2) . . . 15
Firewall . . . 16
Proxy . . . 17
Transparenz von Komponenten . . . 18
Intransparenz von Komponenten . . . 19

2. Cloud . . . 20

3 Arten von Cloud Architekturen (1) . . . 21
3 Arten von Cloud Architekturen (2) . . . 22
Verwandter Begriff: Zero Install . . . 23
Vorteile . . . 24
Nachteile (1) . . . 25
Nachteile (2) . . . 26
Nachteile (3) . . . 27
Beispiele für Abhängigkeit (1) . . . 28
Umfrage . . . 29
Beispiele für Abhängigkeit (2) . . . 30
Beispiele für Abhängigkeit (3) . . . 31
Problem der Abo-Modelle (1) . . . 32
Problem der Abo-Modelle (2) . . . 33
Rentabilität der Abhängigkeit . . . 34
Grundproblem der Cloud . . . 35
Mögliche Auswege aus der Abhängigkeit . . . 36
Offene Dokumentenformate im Vergleich . . . 37
Umfrage . . . 38
Aufgabe: Kostenvergleich Cloud . . . 39
Wirtschaftlicher Vergleich von Unternehmen . . . 40

3. Home Automation 41
Home Automation 42
Thread 43
Architektur von Thread (1) 44
Architektur von Thread (2) 45
Architektur von Thread (3): Weitere Aufgaben 46
Matter 47
Matter: Schichten-Architektur der Kommunikation (1) 48
Matter: Schichten-Architektur der Kommunikation (2) 49
Matter: Schichten-Architektur der Konzepte (1) 50
Matter: Schichten-Architektur der Konzepte (2) 51

Legende:
Fortsetzungsseite
Seite ohne Überschrift
Bildseite