

# Verteilte Systeme und Netzwerke



**JProf. Dr. Gunnar Stevens**  
Human Computer Interaction  
University of Siegen  
[gunnar.stevens@uni-siegen.de](mailto:gunnar.stevens@uni-siegen.de)

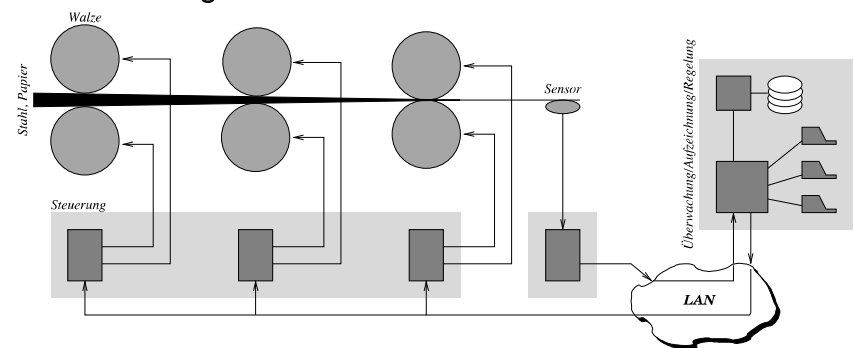
# Motivation

2

- Gemeinsame Nutzung von Betriebsmitteln
  - {Druck, Datenbank, Datei, Namens...}dienst
- Geographisch verteilte, kooperative Anwendungen

Transaktionssysteme - Geldautomaten

Industrieanlagen - Walzwerk



# Definition

3

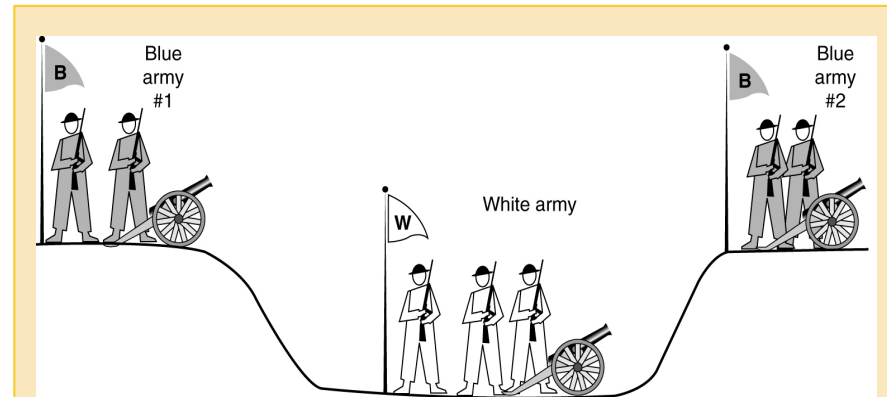
## □ Definition I

- “Ein verteiltes System ist eine Kollektion unabhängiger Computer, die den Benutzern als Einzelcomputer erscheinen” – A. Tannenbaum

## □ Definition II

- Mehrere, physikalisch bzw. logisch verteilte, autonome Prozesse, die
- über (unzuverlässige) Kanäle/(offene) Netzwerke

kommunizieren bzw. sich koordinieren



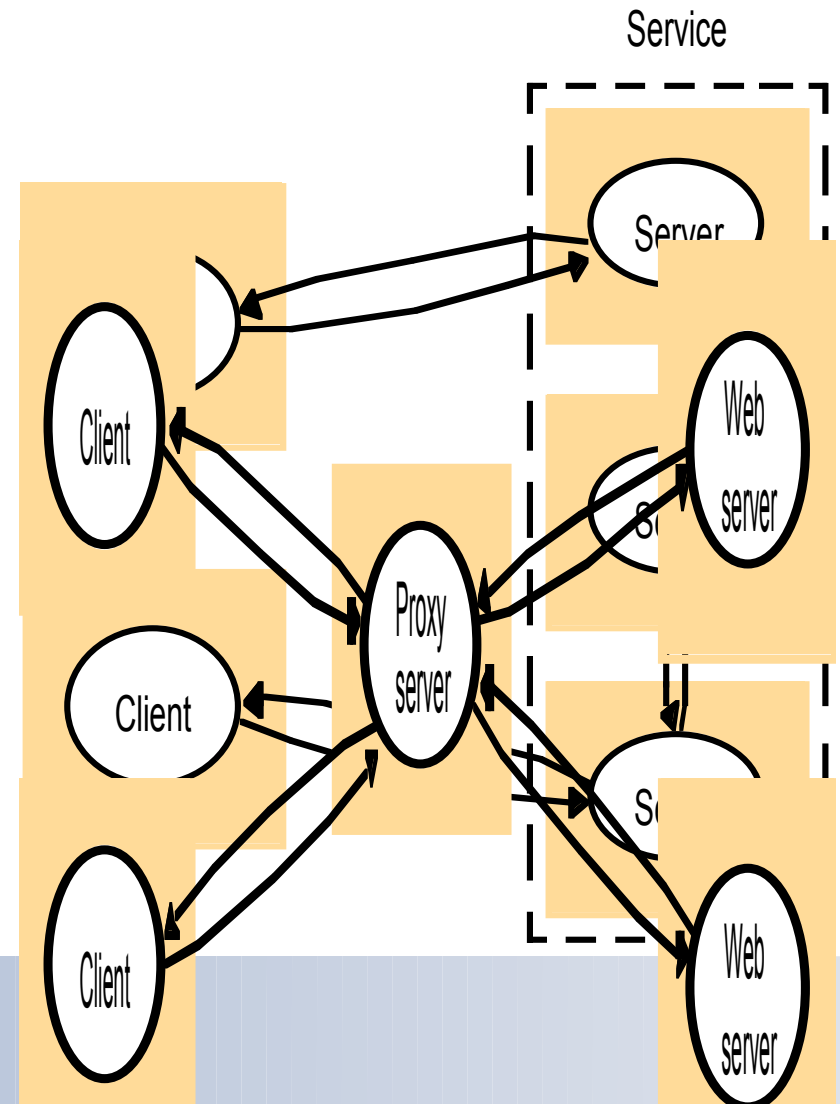
### **Beispiel “The Two Generals' Problem”:**

Um die weiße Armee koordiniert anzugreifen, müssen die zwei Generäle der blauen Armee Boten durch das Tal schicken, dass von der weißen Armee kontrolliert wird

# Client/Server

4

- Client/Server sind Rollen von Prozessen innerhalb einer Dienstleistung eines verteilten Systems
- Server: Reagierender Prozess
  - bearbeitet Anfragen
  - ständige Ausführung
  - Adresse bekannt
- Beispiel: Webshop, Anrufbeantworter
- Client: Initiierender Prozess
  - stellt Anfragen



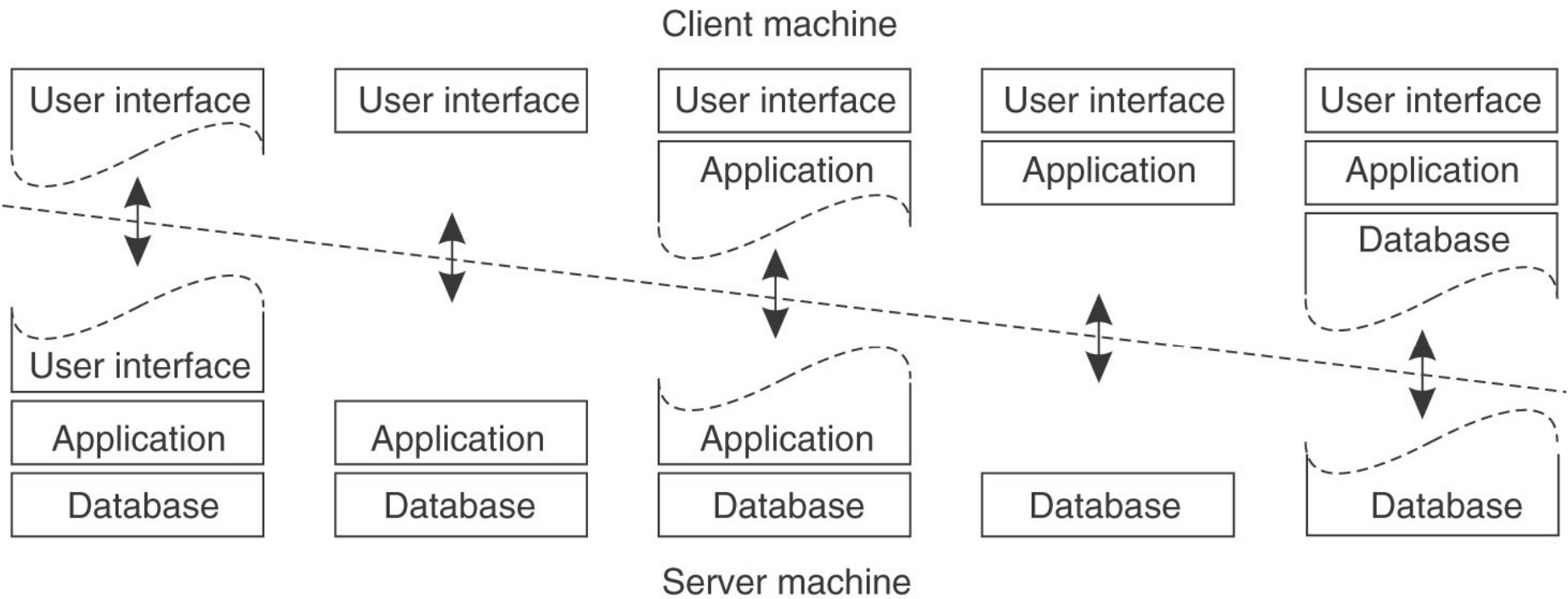
# Drei Ebenen Architektur:

5

„*three-tier*“

# Ebenen im Client/Server Model

6



Thin Client



Fat Client

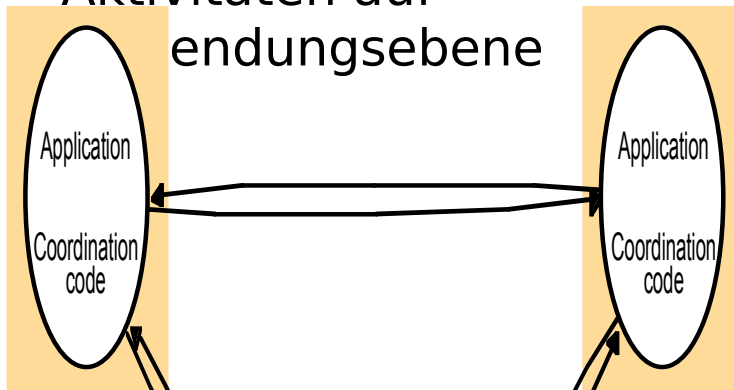


# Gleichrangige Prozesse

## Aufgaben der

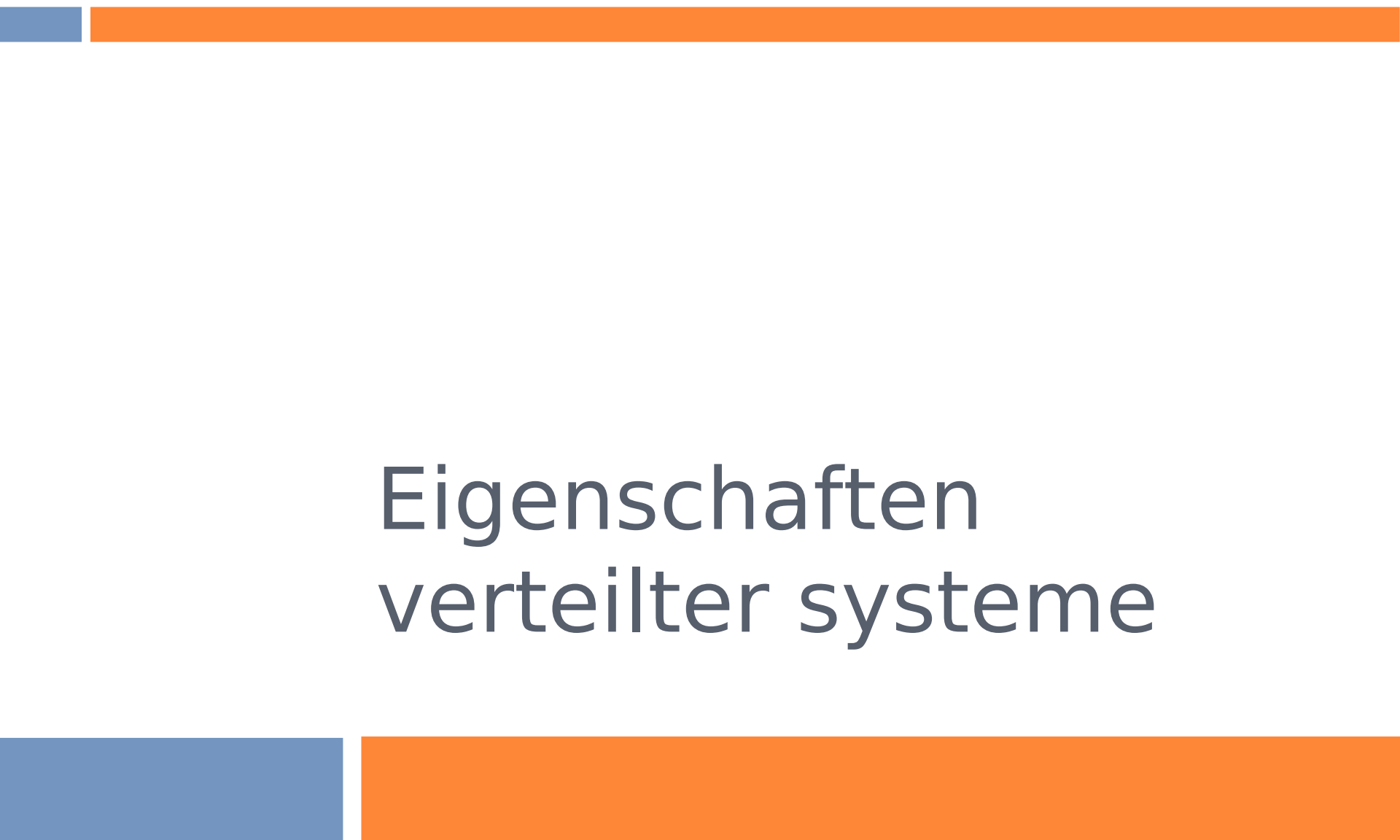
## Peer-Prozesse

- Konsistenz der Ressourcen auf Anwendungsebene
- Synchronisation der Aktivitäten auf Anwendungsebene



## □ Eigenschaften

- Vermeidet Flaschenhals-Prozesse
- Meist höhere Komplexität
- Beispiele
  - File-Sharing-Systeme: Jedes System teilt den anderen Peer-Prozessen, welche Dateien zur Verfügung stehen
  - P2P basierte Groupware (z.B. Groove, Sykpe): Nachrichten eines Gruppenchats werden vom initierenden Prozess an alle Teilnehmer geschickt



# Eigenschaften verteilter systeme



# Eigenschaften

9

- Charakteristische Eigenschaften
  - Heterogenität
  - Nebenläufigkeit
  - Fehlerverarbeitung
- Wünschenswerte Eigenschaften:
  - Sicherheit
  - Skalierbarkeit
  - Offenheit
  - Transparenz

# Heterogenität

10

- Bereiche von Heterogenität
  - ▣ Netzwerke und Kommunikationsprotokolle
  - ▣ Prozesse und Betriebssysteme
  - ▣ Personen und Organisationen
  - ▣ ...
- Umgang Heterogenität
  - ▣ Nutzung gemeinsamer, offener Standards
  - ▣ Nutzung von Middleware zum Verbergen von H.
  - ▣ Virtuelle Maschinen und mobiler Code
  - ▣ ...

# Nebenläufigkeit

11

- Koordination unabhängiger Prozesse bzw.
- Zugriff auf gemeinsame Ressource

- Gestaltungsoptionen
  - Exklusive Nutzung
  - Parallele Nutzung
  - Nutzung von Replikaten

Exklusive Nutzung

Parallele Nutzung

Nutzung von Replikaten

# Fehlerverarbeitung

12

## Fehler

*An error is a manifestation of a fault in a system, which can lead to system failure*

- Failure (Funktionsausfall)
  - Keine Dienstleistung bzw. weicht von Spezifikation ab
- Error (Fehler)
  - Systemzustand der zu ein Funktionsausfall führen kann
- Fault (Fehlerursache)
  - Fehlerhafte

## Software/Hardware

## Fehlerverarbeitung

- Fehler erkennen
  - z.B. durch Prüfsummen-Check
- Maskieren/Verbergen
  - Erkannte Fehler nach außen verbergen (z.B. verlorene Nachrichten wiederholen) oder abschwächen (z.B. fehlerhafte Nachrichten verwerfen).
- Tolerieren
  - Nicht erkannte bzw. maskierte Fehler sollten toleriert werden können

# Fehlerarten

13

- Gutmütige Fehler (benign faults)
  - Crash Stop: Prozess bzw. der Zugriff darauf fällt komplett aus
    - Fail Stop: Andere Prozesse erkennen Ausfall innerhalb endlicher Zeit
    - Fail Silent: Keine garantierte Ausfallerkennung möglich
  - Crash Recovery: Ein korrekter Knoten kann endlich oft ausfallen und wiederanlaufen
- Bösertige Fehler (malicious/byzantine faults)
  - Prozess/Kanal weist zufälliges Verhalten auf
  - Fehlerhafte Prozesse können beliebige Aktionen (koordiniert) ausführen
  - Umfasst sämtliche Arten von Fehler, z.B. auch gezielte Angriffe von außen auf das System

# Sicherheit

14

## □ **Motivation**

- Datenübertragung über öffentliche Netze, sowie
- Nutzung fremder Dienste bzw. öffnen von Diensten für Fremde

erhöht das Risiko der missbräuchlichen Nutzung

## □ **Beispiele**

- Nachrichten können abgefangen, verfälscht bzw. gefälscht und/oder mehrfach geschickt werden (z.B. Kontoüberweisungen)
- Dienste und Nutzer können vorgetäuscht werden (z.B. Onlinebank-Attrappe)

## □ **Schutzziele CIA**

- **Confidentiality** (Vertraulichkeit)
  - nur Berechtigte dürfen auf Daten zugreifen
- **Integrity** (Integrität)
  - keine unbemerkte Veränderung der Daten
- **Availability** (Verfügbar)
  - Daten müssen dann verfügbar sein, wenn ein autorisierter Benutzer diese benötigt

# Schutzziele: Sortierung

	Inhalte	Umfeld / Kontext
Vertraulichkeit	Vertraulichkeit Verdecktheit	Anonymität Unbeobachtbarkeit
Korrektheit	Integrität	Zurechenbarkeit
	Verfügbarkeit	Erreichbarkeit Verbindlichkeit

# Abwehr von Bedrohungen

Benutzer



authentifiziert

authentifiziert sich an



Server

16

- Identifizierung
  - ▣ Zuordnung einer Identität zu einer Entität
- Authentisierung
  - ▣ Entität behauptet Identität und erbringt den zugehörigen Nachweis
- Authentifizierung
  - ▣ Prüfung des Nachweises zur Identität
- Autorisierung
  - ▣ Berechtigung auf eine Ressource zuzugreifen
- Verschlüsselung
  - ▣ Nachrichten verschlüsseln, so dass Inhalt nur von autorisierten Prozessen/Personen gelesen werden kann



# Skalierbarkeit

17

## □ Definition

- Ein System skaliert, wenn es bei steigender Anzahl an Ressourcen und Benutzer effizient bleibt.

## □ Kernprobleme beim Entwurf skalierbarer verteilter Systeme

- Kostenkontrolle für die physischen Ressourcen: Bei steigender Nachfrage soll eine Erweiterung zu vernünftigen Kosten möglich sein
- Kontrolle des Leistungsverlustes und Vermeidung von Leistungsengpässen: Hierarchische Lösungen (Beispiel DNS) bevorzugen, so dass die Menge zu ladender/verarbeitender Daten nicht zu groß wird
- Erschöpfung der SW-Ressourcen verhindern: Ressourcen

so anlegen, dass sie für zukünftige Erweiterungen voraussichtlich ausreichen werden. Aktuelles Problem:

# Offenheit

18

## □ Definition

- Offenheit beschreibt den Grad, zu dem neue Dienste hinzugefügt und zur Verwendung von unterschiedlichen Clients bereitgestellt werden

## □ Offene verteilte Systeme

- basieren auf dem Vorhandensein eines einheitlichen Kommunikationsmechanismus und veröffentlichten Schnittstellen für den Zugriff auf gemeinsam genutzte Betriebsmittel;
- Können aus heterogener Hard- und Software aufgebaut sein, die insbesondere auch von unterschiedlichen Herstellern stammen können.
- Beispiel Internet-Protokolle

JProf Dr. Gunnar Stevens  
Human Computer Interaction, University of Siegen  
gunnar.stevens@uni-siegen.de  
RFC Dokumente (Requests For Comment), laufend nummeriert, enthalten Diskussionen und Spezifikationen (siehe auch RFC 1918)

# Transparenz

19

## □ Definition

- Das verteilte System wird vom Anwender/Programmierer als einheitliche/lokale Einheit wahrgenommen und nicht als Sammlung voneinander unabhängiger, verteilter Komponenten.

## □ Acht Formen der Transparenz (ANSI/ISO):

1. **Zugriffs-T.** ermöglicht den Zugriff auf lokale und entfernte Ressourcen unter Verwendung identischer Operationen.
2. **Positions-T. (Ortstransparenz)** erlaubt den Zugriff auf die Ressourcen, ohne dass man ihre Position/ihren Ort kennen muss.
3. **Nebenläufigkeits-T.** erlaubt, dass mehrere Prozesse gleichzeitig mit denselben gemeinsam genutzten Ressourcen arbeiten, ohne sich gegenseitig zu stören.

# Transparenz

20

4. **Replikations-T.** erlaubt, dass mehrere Instanzen von Ressourcen verwendet werden, um die Zuverlässigkeit und die Leistung zu verbessern, ohne dass die Benutzer oder Applikationsprogrammierer wissen, dass Repliken verwendet werden.
5. **Fehler-T.** erlaubt das Verbergen von Fehlern, so dass Benutzer und Applikationsprogrammierer ihre Aufgaben erledigen können, auch wenn Hardware- oder Softwarekomponenten ausgefallen sind.
6. **Mobilitäts-T.** erlaubt das Verschieben von Ressourcen und Clients innerhalb eines Systems, ohne dass die Arbeit von Benutzern oder Programmen dadurch beeinträchtigt wird.
7. **Leistungs-T.** erlaubt, dass das System neu konfiguriert wird, um die Leistung zu verbessern, wenn die Last variiert.
8. **Skalierungs-T.** erlaubt, dass sich Systeme und Applikationen vergrößern, ohne dass die Systemstruktur oder die Applikationsalgorithmen geändert werden müssen.

# Zusammenfassung

21

- *Es ist die Eigenart verteilter Systeme, dass es auf Dauer keine zwei Prozesse gibt, die zur gleichen Zeit, die gleiche zutreffende Sicht des Systems haben*
  - Keine gemeinsame Uhr
    - Uhrensynchronisation aufgrund nicht bekannter Verzögerungszeiten bei der Kommunikation (meist Nutzung logischer statt physikalischer Uhren)
  - Kein gemeinsamer Speicher
    - Kein einzelner Knoten kennt den vollständigen globalen Zustand. Es ist daher schwierig, globale Eigenschaften des Systems zu beobachten
  - Keine akkurate Ausfallerkennung
    - Es ist in einem asynchronen verteilten System (d.h. keine obere Schranke für Kommunikationszeiten bekannt) unmöglich, langsame von ausgefallenen Prozessen zu unterscheiden

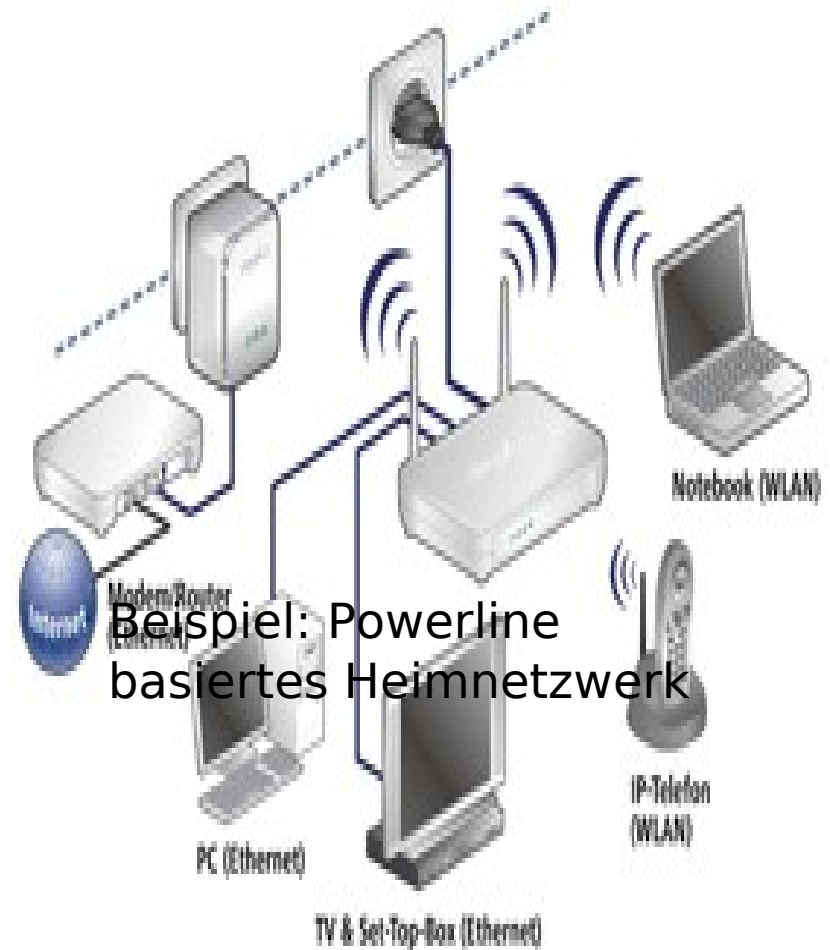


# Netzwerke

# Definition

23

- “Ein Rechnernetz
  - ▣ besteht aus mehreren autonomen Rechnern (Computern),
  - ▣ welche zum Zweck des Datenaustauschs miteinander verbunden sind” –



# Netzwerkarten

## □ Typen

- Personal Area Networks (PAN)
- Local Area Networks (LAN)
- Metropolitan Area Networks (MAN)
- Wide Area Networks (WAN)
- Inter-Networks

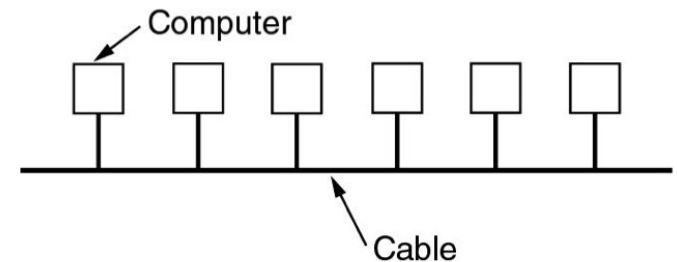
## □ Merkmale

- Übertragungsweg
  - Wired
  - Wireless
- Kennzahlen
  - Latenz
  - Bandbreite
  - Jitter

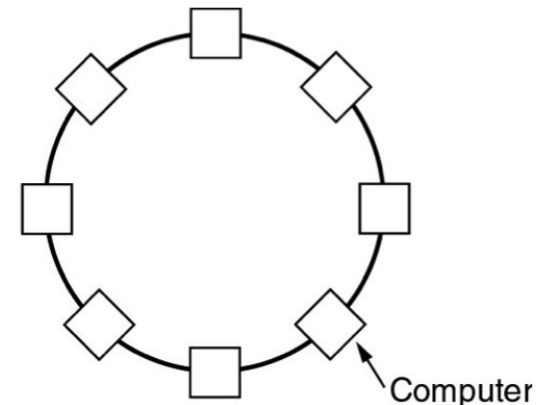


# Local Area Networks (LAN)

- Ein LAN ist
  - ein lokales Netz, das Endsysteme innerhalb eines oder benachbarter Gebäude verbindet.
  - Häufig basieren lokale Netze auf Ethernet oder drahtlosen Technologien



(a) Bus-Topologie

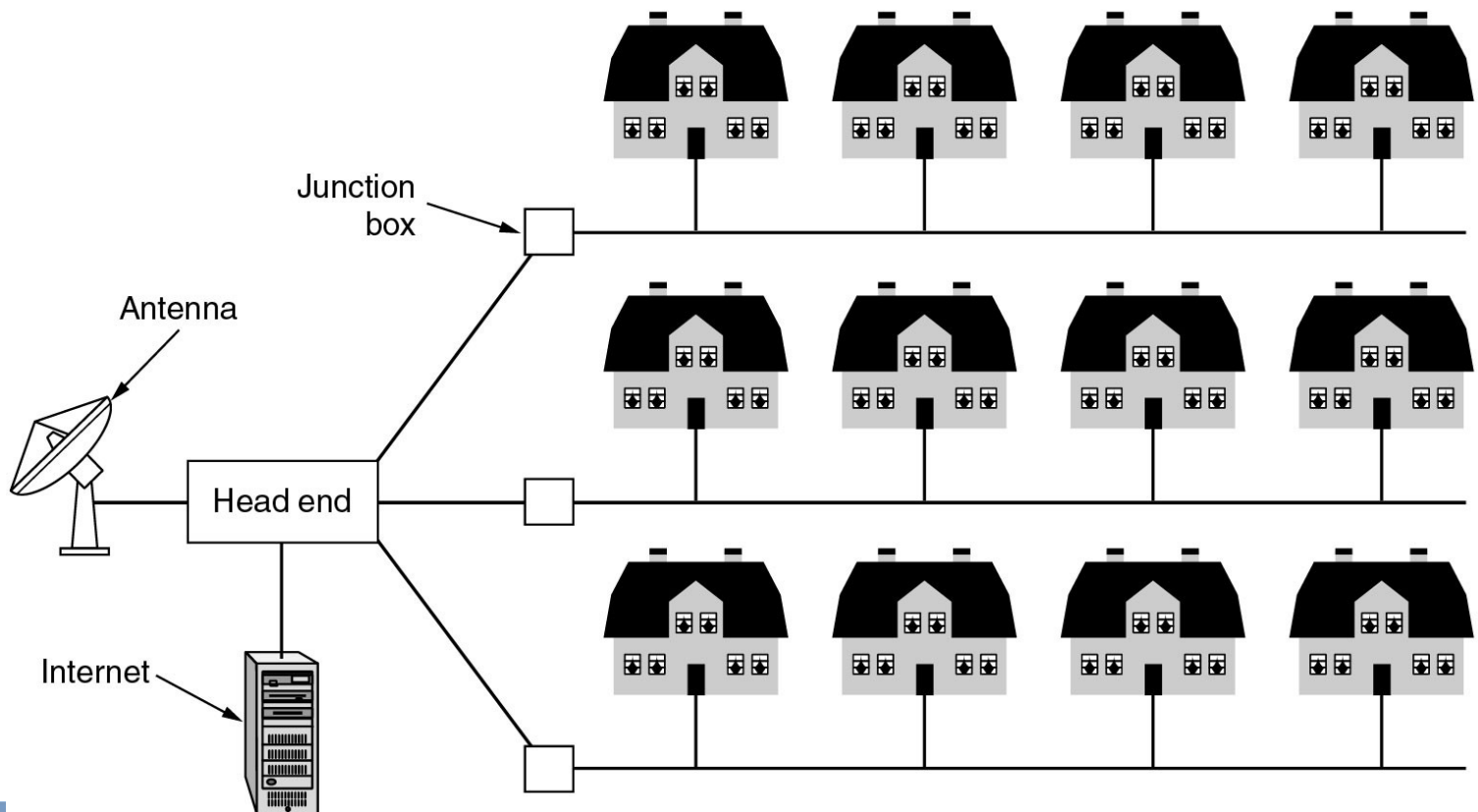


(a) Ring-Topologie

- Netzwerktopologie
  - Bus (z.B. Ethernet)
  - Ring (z.B. Token Ring)

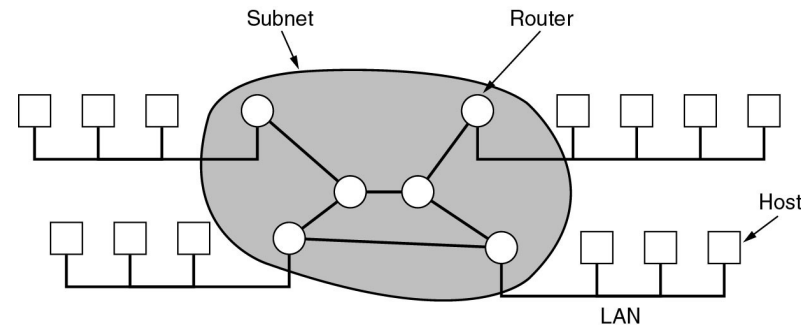
# Metropolitan Area Networks

## □ TV-Kabel basiertes MAN



# Wide Area Networks (WAN)

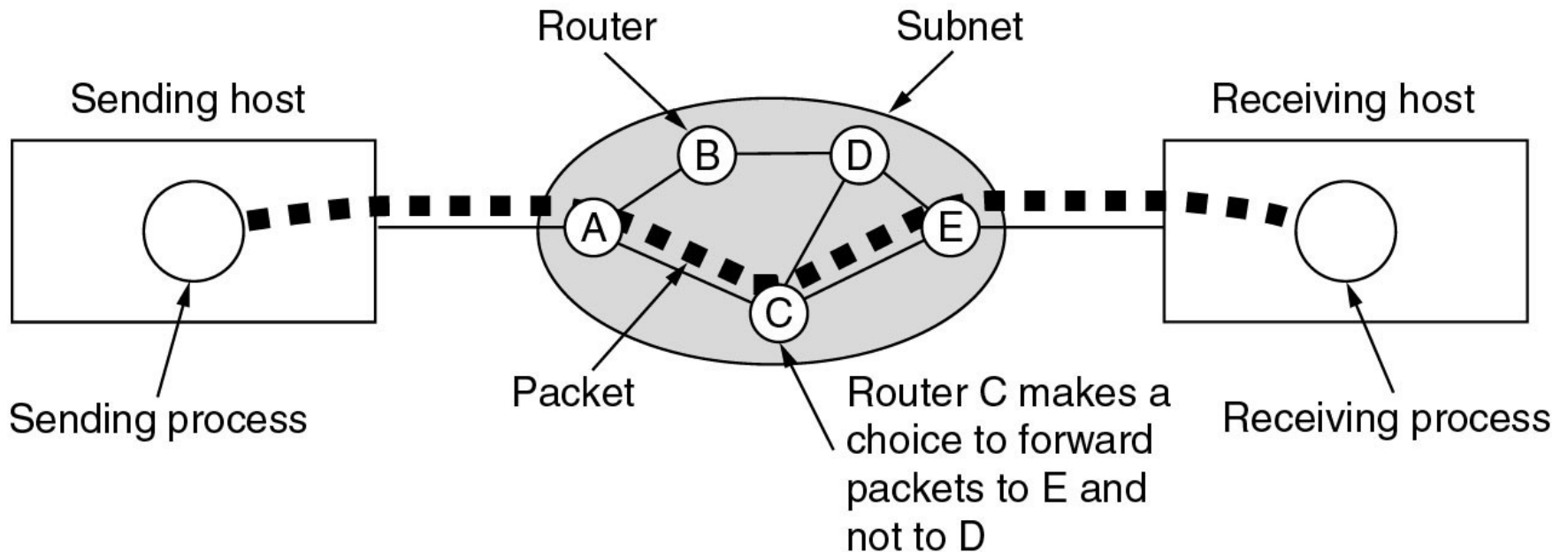
- Ein Weitverkehrsnetz (WAN) ist ein Rechnernetz, dass
  - einen sehr grossen geografischen Bereich (Land, Kontinent oder die ganze Welt) abdeckt.
  - oft setzt es sich aus vielen einzelnen, kleineren, von mehreren Organisationen betriebenen Rechnernetzen zusammen,
  - die über Vermittlungssysteme miteinander verbunden werden.



Zusammenschluß mehrerer LAN zu einem Subnetz eines WAN

## Vermittlungssysteme

- Weiterleitung eines Paket-Strom (stream of packets) vom Sender zum Empfänger



# Kennzahlen

29

<i>Wired:</i>	<i>Example:</i>	<i>Range:</i>	<i>Bandwidth (Mbps):</i>	<i>Latency (ms):</i>
LAN	Ethernet	1–2 kms	10–1000	1–10
WAN	IP routing	worldwide	0.010–600	100–500
MAN	ATM	2–50 kms	1–150	10
Internetwork	Internet	worldwide	0.5–600	100–500
<i>Wireless:</i>				
WPAN	Bluetooth (IEEE 802.15.1)	10–30m	0.5–2	5–20
WLAN	WiFi (IEEE 802.11)	0.15–1.5 km	2–54	5–20
WMAN	WiMAX (IEEE 802.16)	5–50 km	1.5–20	5–20
WWAN	GSM, 3G phone nets	worldwide	0.010–2	100–500

Quelle: Coulouris et al. 2005



# VERMITTLUNGSTECHNI EN

# Vermittlung (Switching)

31

- **Leitungsvermittlung (circuit-switching)**
  - Zu Beginn wird eine Leitung zwischen Sender und Empfänger aufgebaut. Die Leitung dient dann zur Übertragung von Datenströmen
  - Synchrone Übertragung, Zeit orientierte Tarife, geringes Jitter
  - Beispiel: Analoges Telefon
- **Paketvermittlung (packet-switching)**
  - Aufteilung der Nachricht in Pakete, die unabhängig voneinander zum Ziel geschickt werden. Sie müssen nicht die gleiche Route nehmen!
  - Asynchrone Übertragung, Volumen orientierte Tarife, hohes Jitter

Prof. Dr. Gunnar Stevens

Human Computer Interaction, University of Siegen

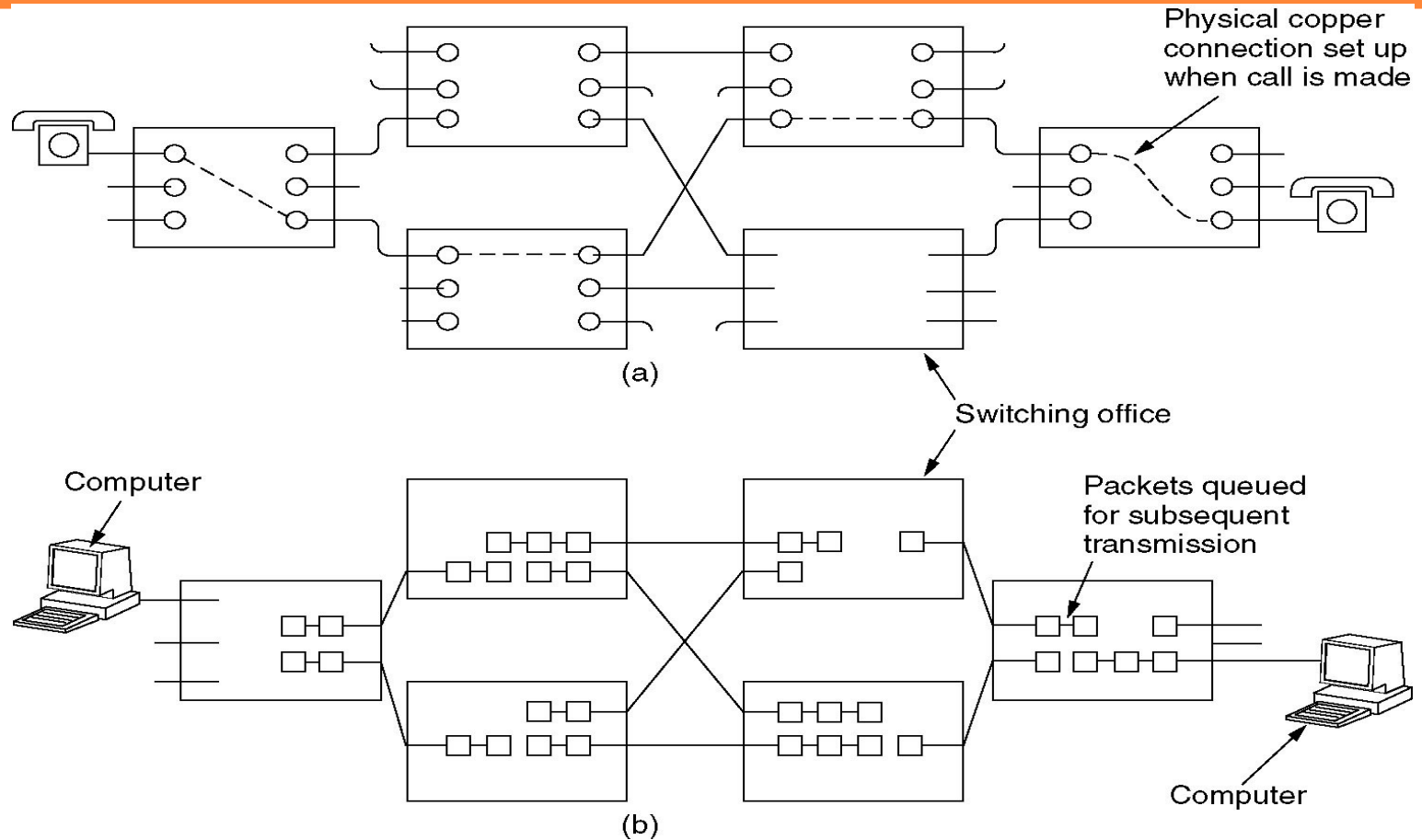
gunnar.stevens@uni-siegen.de

□ Beispiel: TCP/IP, Voice over IP (VoIP)

**Nachrichtenvermittlung (message switching)**



# Strukturdiagramm

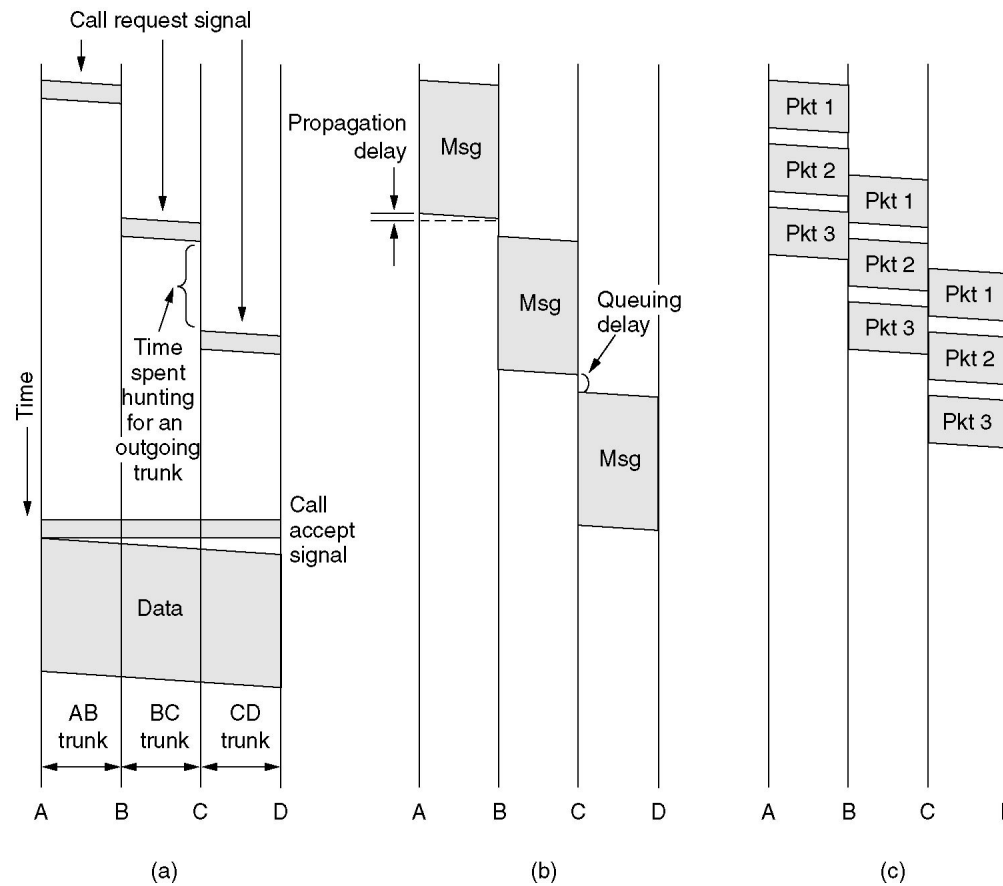


(a) Leitungsvermittlung (b) Paketvermittlung

## Paketvermittlung



# Sequenzdiagramm



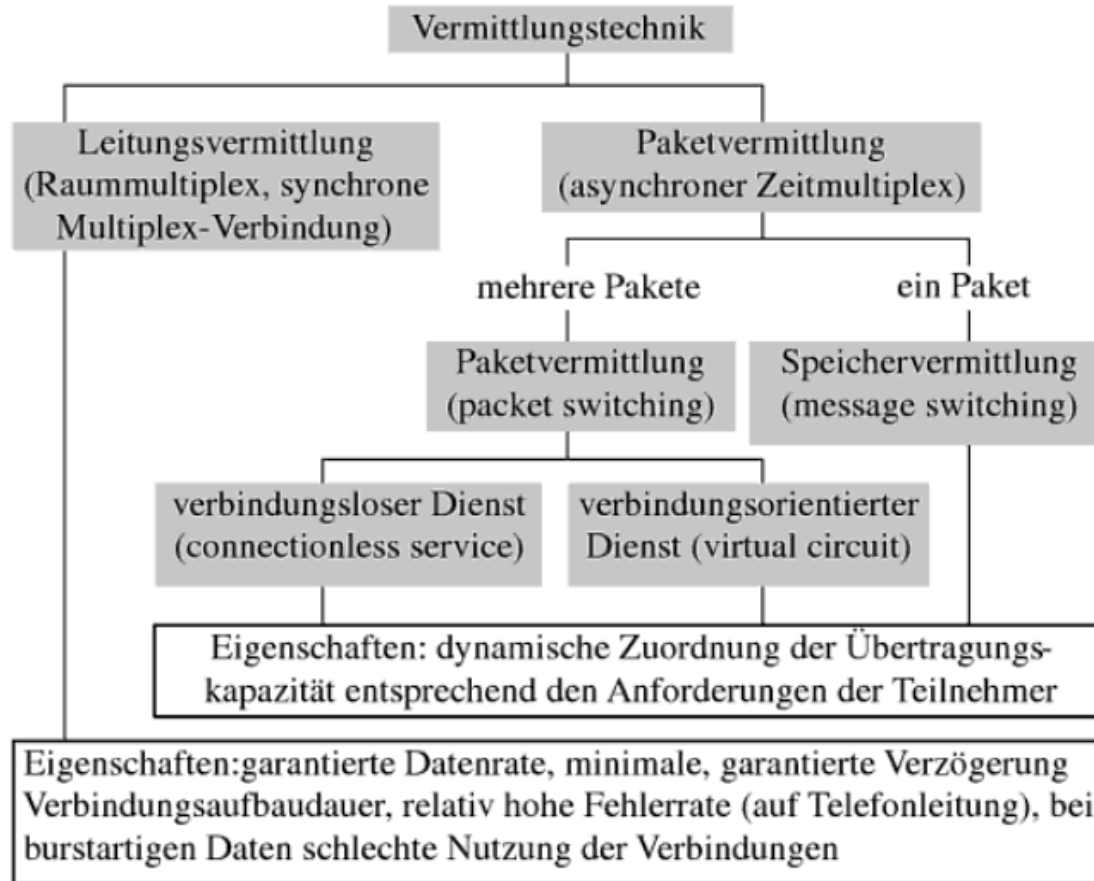
(a) Leitungsvermittlung (b) Nachrichtenvermittlung (c) Paketvermittlung

# Vergleich

Item	Circuit-switched	Packet-switched
Call setup	Required	Not needed
Dedicated physical path	Yes	No
Each packet follows the same route	Yes	No
Packets arrive in order	Yes	No
Is a switch crash fatal	Yes	No
Bandwidth available	Fixed	Dynamic
When can congestion occur	At setup time	On every packet
Potentially wasted bandwidth	Yes	No
Store-and-forward transmission	No	Yes
Transparency	Yes	No
Charging	Per minute	Per packet

# Zusammenfassung

35



Quelle: Stein (2008) "Rechnetze und Internet"

Prof. Dr. Gunnar Stevens

Human Computer Interaction, University of Siegen

gunnar.stevens@uni-siegen.de



# Netzwerk software

# Gestaltung von Netzwerksoftware

## Allgemeine Aufgaben

- Addressierung
  - Identifizierung der Netzknoten
- Fehlerbearbeitung
  - Erkennen, Maskieren
- Datenfluss-Kontrolle (Flow control)
  - Abstimmung von Senden/Empfangen
- Multiplexing
  - Nutzung einer Leitung durch mehrere Akteure
- Routing

□ Festlegen eines Weges vom  
Empfänger zum Ziel

## Gestaltungsprinzip

- Separation of Concerns
- Kapselung von Diensten und deren Realisierung

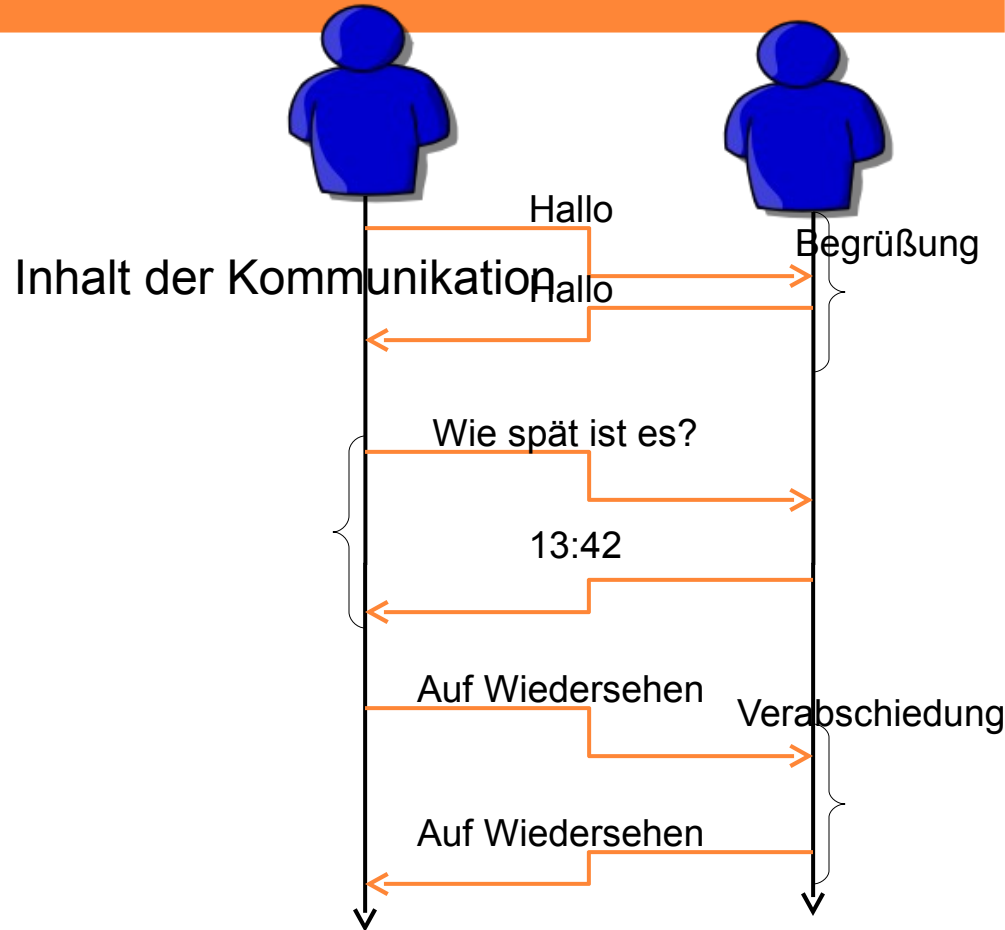
## Gestaltungsansatz

- Referenzmodelle von Schichtenarchitekturen

# Protokoll und Dienste

38

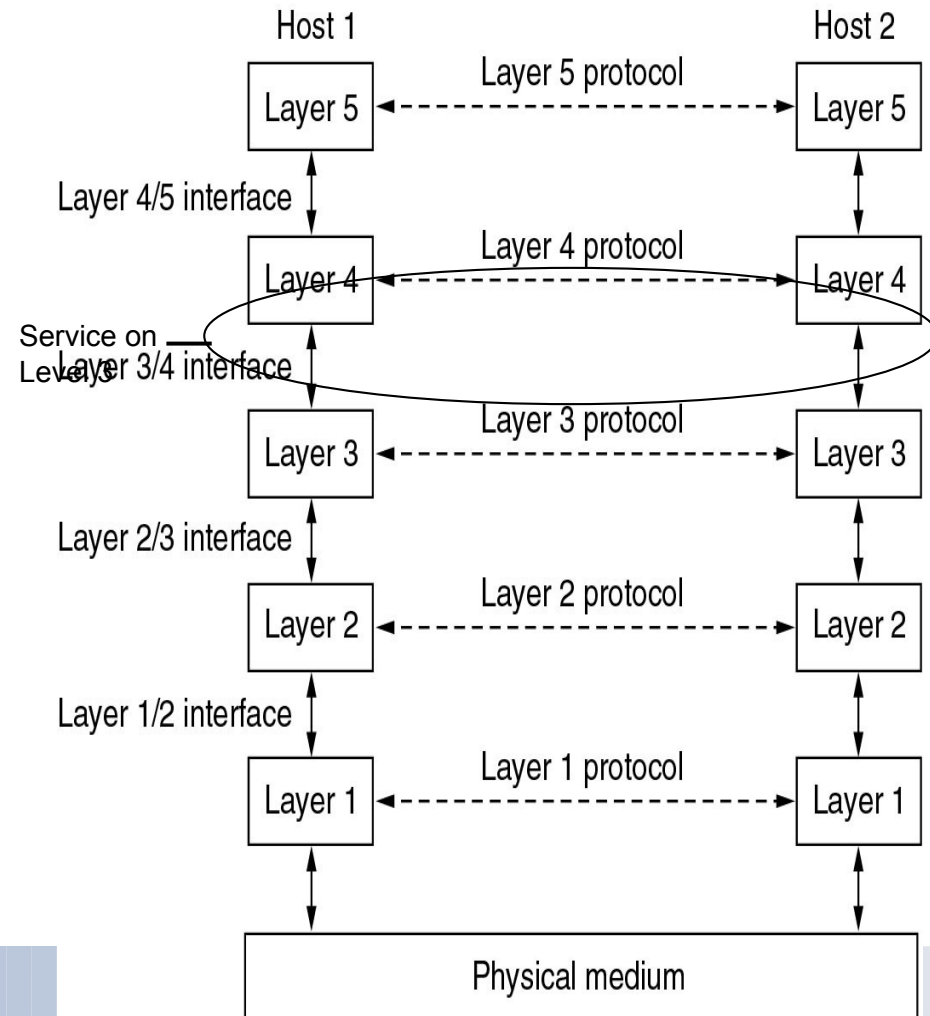
- Protokoll
  - definiert das Format und die Reihenfolge ausgetauschter Nachrichten zwischen zwei oder mehr Kommunikationspartnern, sowie die Aktionen die durch das Senden bzw. Empfangen ausgeführt werden müssen
- Dienste
  - ist die Erbringung einer Leistung nach einem spezifizierten Protokoll



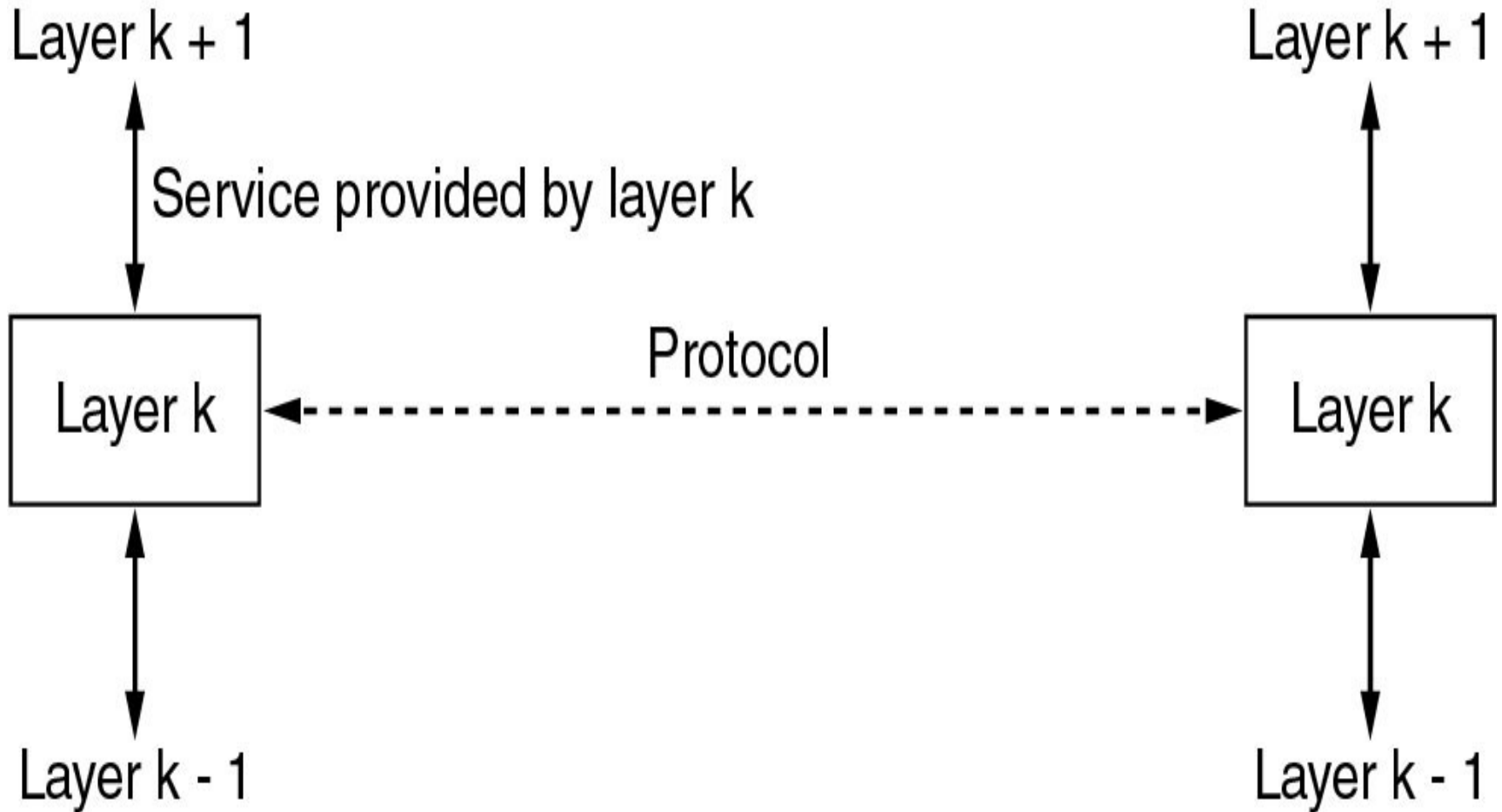
von einem Dienstleistungsgeber für ein

# Protokoll-Hierarchien

- **Protokolstapel (protocol stack)**
  - **Schichten (layers)**
    - bieten Dienste an
  - **Schnittstellen (interfaces)**
    - legen Zugriff auf Dienste fest
  - **Protokolle (protocols)**
    - spezifizieren Dienste auf der jeweiligen Netzwerkschicht



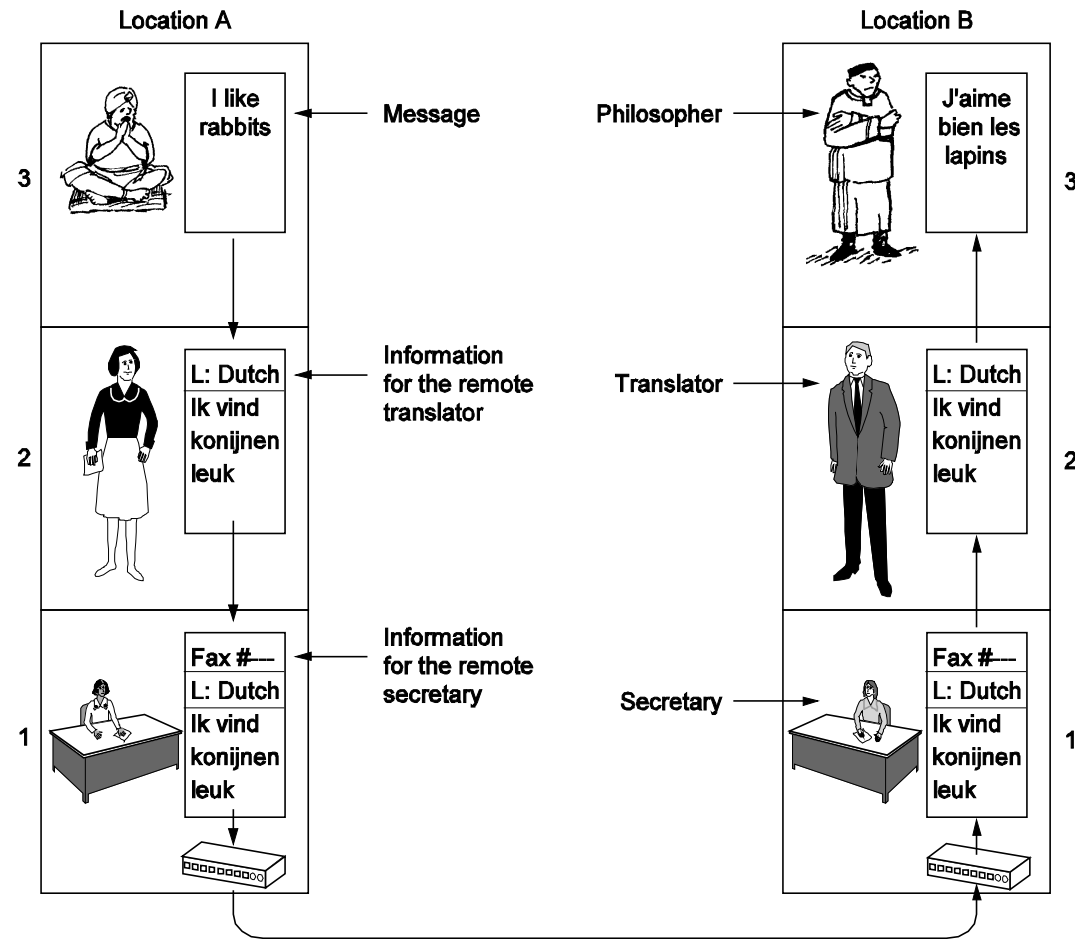
# Dienst-Protokoll Zusammenhang





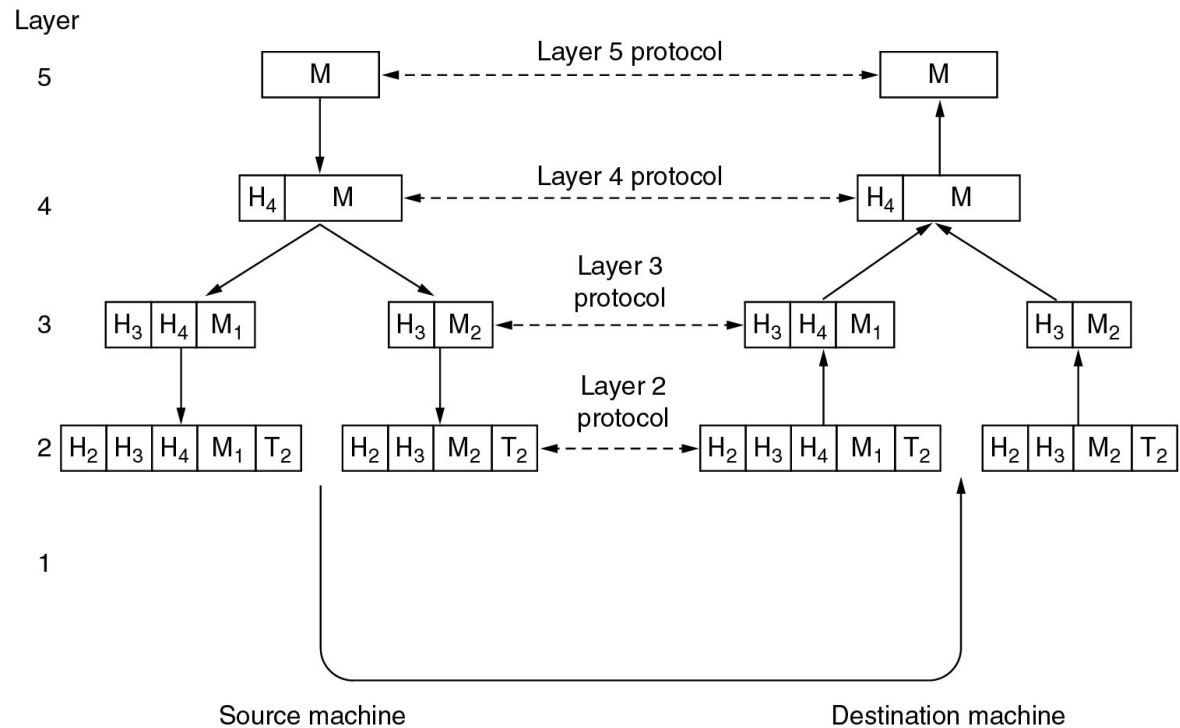
# Beispiel 1

- Beispiel einer Dreischichten Architektur
  - ▣ Philosophen
  - ▣ Übersetzer
  - ▣ Sekretariat



# Beispiel 2

- Sender
  - Einpacken
    - Zerlegen
    - Zusammenlegen
  - Addressieren
- Empfänger
  - Auspacken
    - Zusammensetzen
    - Aufteilen
  - Adresse entfernen



# Angebotene Dienste einer Schicht

- Verbindungs-orientiert (connection orientend)
  - Modelliert nach Telefon-Prinzip
    - Aufbau einer Verbindung
    - Sender der Nachrichten (plural)
    - Schliessen der Verbindung)
- Verbindungslos (connectionless, datagram oriented)
  - Modelliert nach Post-Prinzip
    - Jede Nachricht wird einzeln mit Empfänger-

	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Remote login
	Unreliable connection	Digitized voice
Connection-less	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Registered mail
	Request-reply	Database query

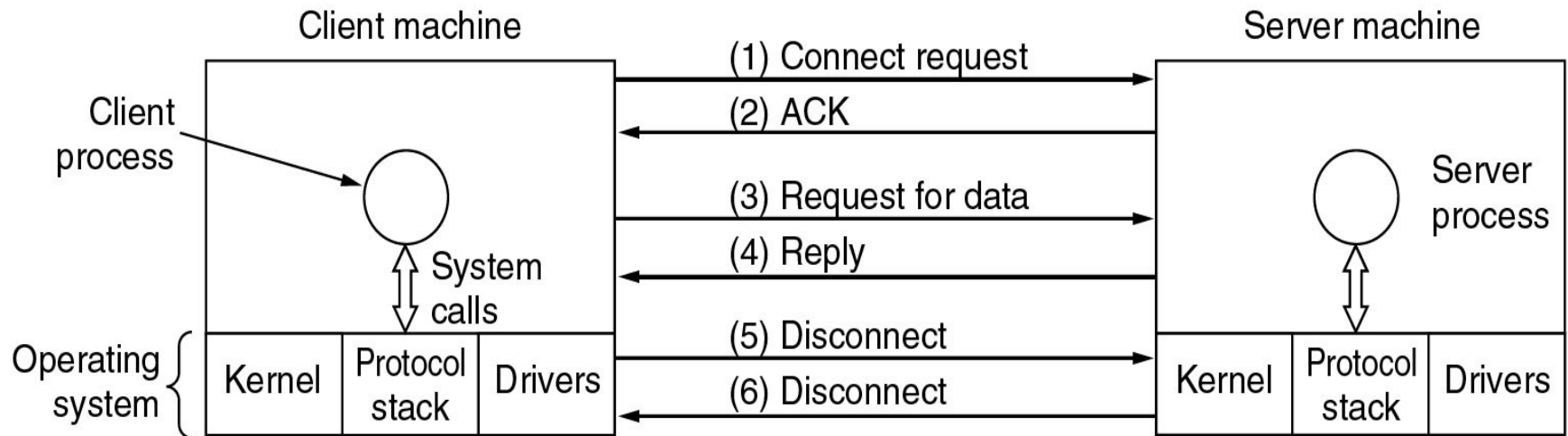
Beispiel von sechs verschiedene Typen von Diensten (Services)

# Dienstprimitive

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

- Fünf Dienstprimitive um einen einfachen verbindungs-orientierten Dienst zu implementieren

# Sequenzdiagramm



Paketversendung in einer einfachen Client-Server Interaction über ein verbindungsorientiertes Netzwerke.



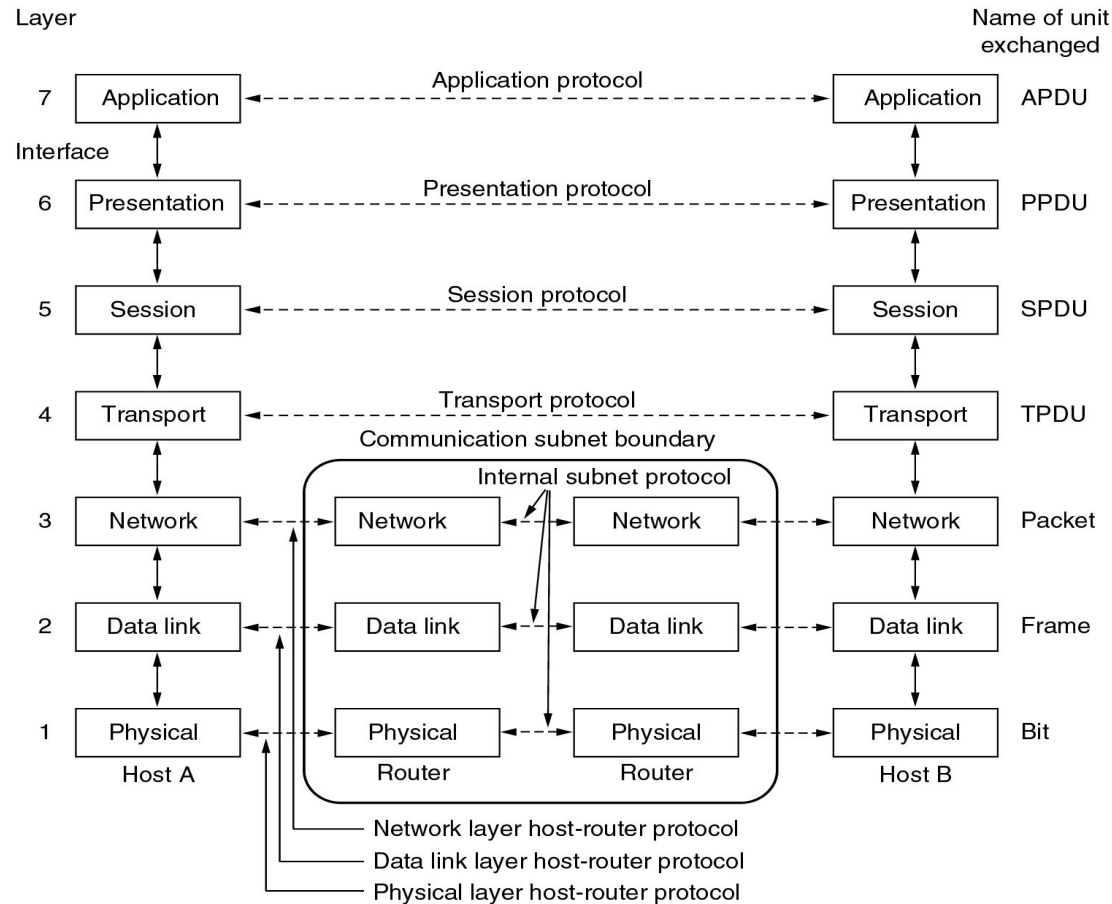
# Referenzmodelle

- OSI
  - TCP/IP
- 

# 7 Schichten OSI Model

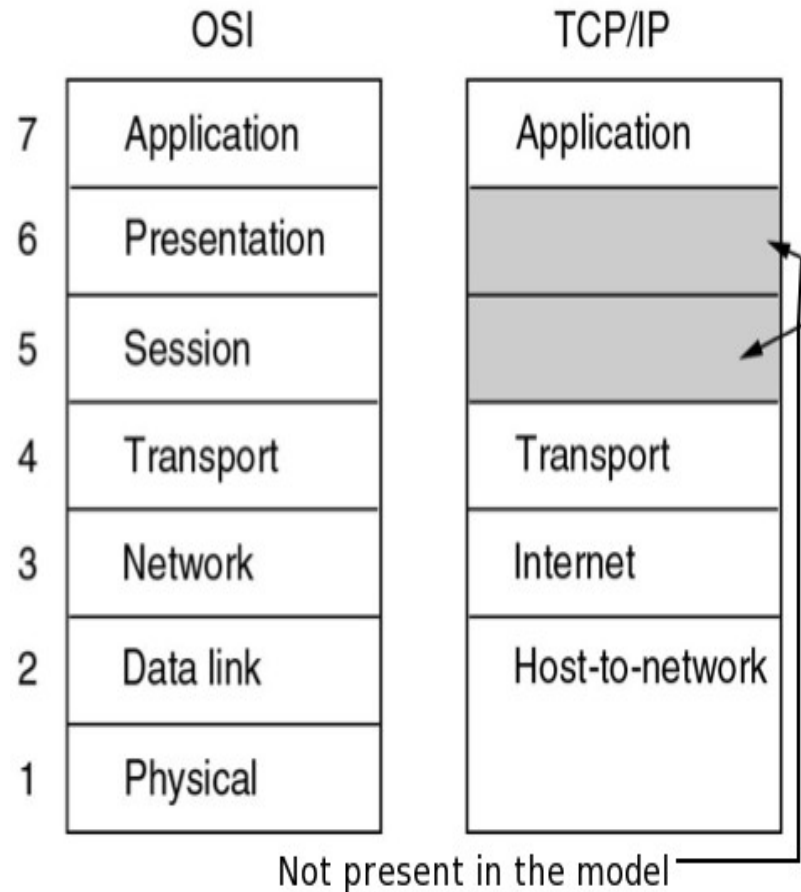
- Open Systems Interconnection (OSI) Reference Model
- 1979 entwickelt
- 1983 ISO Standard Referenzmodell
- Unterteilt Aufgaben in sieben

aufeinander  
aufbauende  
Schichten (layers)



# TCP/IP Model

- Familie von Netzwerkprotokollen
- Grundzüge bereits 1974 festgelegt
- Version 4 bis 1981 in den RFCs definiert



- Hardware-unabhängig

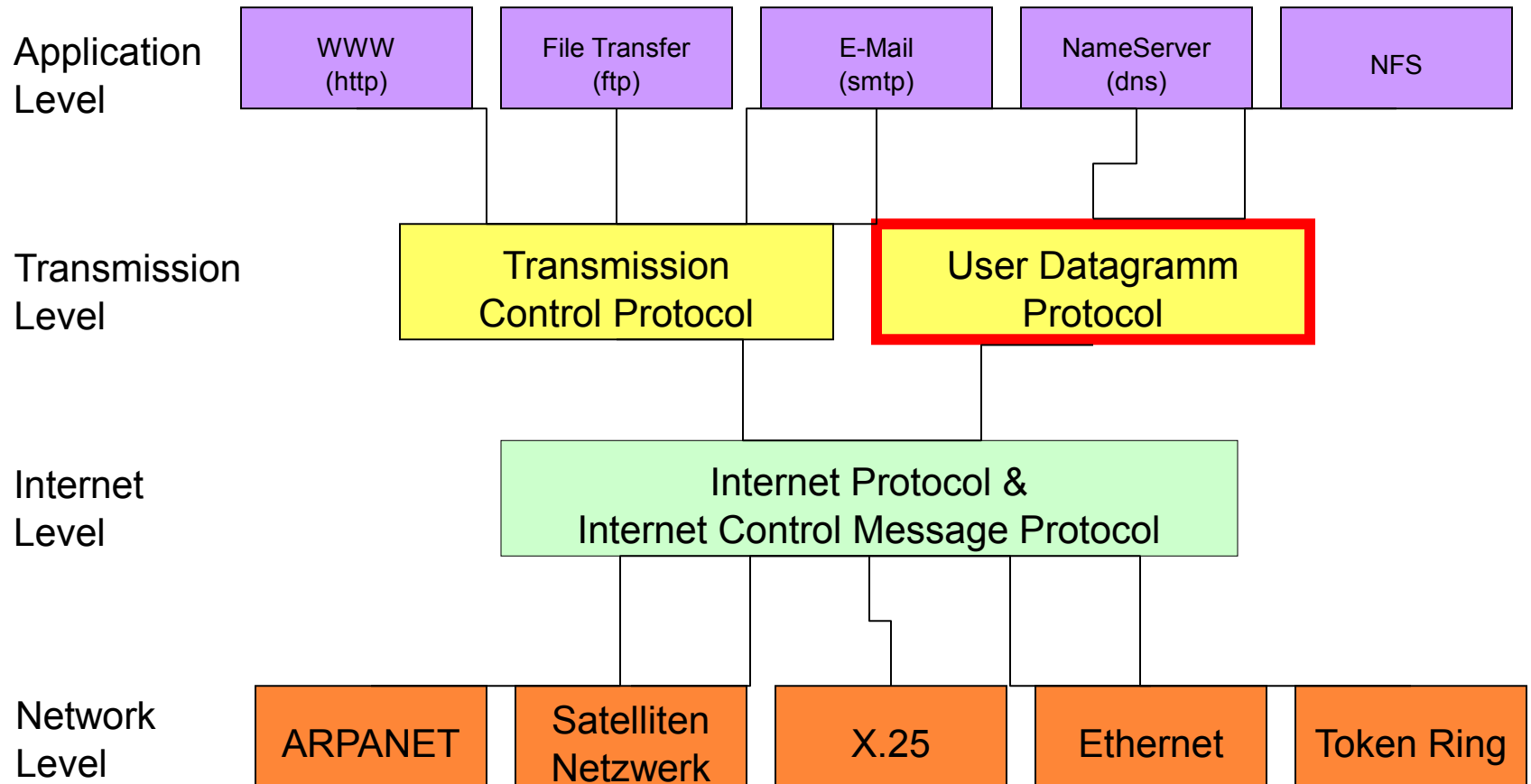
JProf. Dr. Gunnar Stevens

Human-Computer Interaction, University of Siegen  
gunnar.stevens@uni-siegen.de



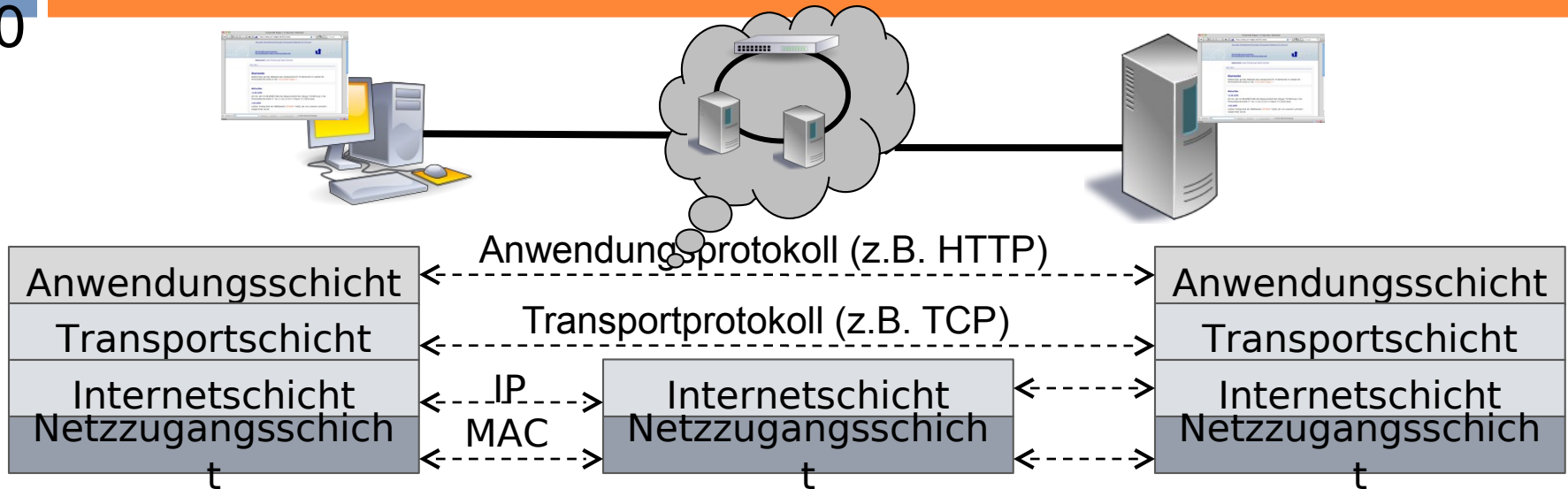
# TCP/IP Schichtmodell

(OSI Namen)



# Zusammenfassung

50



- Schichten können transparent miteinander kommunizieren
  - Jede Schicht hat ihr eigenes Protokoll
  - Kein Wissen über die Implementierung der Dienste der unteren Schichten nötig
  - Daten werden einfach an die nächste Schicht weitergereicht



# Ausblick: NetzwerkSchichten