

---

物理学硕士，数学硕士

**安诺夫·丹策** 博士

---

**SIEMENS**

西门子股份公司  
信息和移动通信  
移动通信

通信地址：

Siemens AG, ICM N-MR TS SE 2  
D-81359 Munich 联邦德国

办公地址：

Hofmannstr.51  
D-81379 Munich

物理学硕士，数学硕士

**安诺夫·丹策** 博士

系统工程 TD-SCDMA

电话: +49 89 722 - 37212

传真: +49 89 722 - 54889

Arnulf.Deinzer@icn.siemens.de

---

# Dr. Arnulf Deinzer

---

44a, verheiratet, 1 Schulkind (6a)

1980-86 Mathematik&Physik Uni Würzburg

1998 Promotion Informatik Uni d. BW

14.5a Siemens AG

- 8a **SW-Entwicklung (OS Vermittlungsr.\*)**
- 1a Integrationstest (Boca Raton FL, USA)
- 5a **Systems Engineering mobiler Systeme**



Dipl.-Phys., Dipl.-Math.

**Dr. Arnulf Deinzer**

Systems Engineering TD-SCDMA  
Director Call Processing

Siemens AG  
Information and  
Communication Mobile

Postal Address:  
Siemens AG, ICM N MR TS SE 2  
D-81359 Munich

Office Address:  
Hofmannstr. 51  
D-81379 Munich

Tel. +49 89 722-37212

Fax +49 89 722-54889

arnulf.deinzer@icn.siemens.de

(\*)

**Betriebssystem:**

- **hoch sicher,**
- **realzeitfähig,**
- **für ein Verteiltes System**

# Veranstaltungen

MAT1	SWT	MAT1	SWT	MAT1	SWT/it3	MAT1
MAT1	SWT-Ü	MAT1	SWT-Ü	MAT1-Ü	SWT-Ü	MAT1-Ü
MAT1	SWT-Ü	MAT1	SWT-Ü	MAT1-Ü	SWT-Ü	VS
MAT1	BSS1	MAT1	BSS1	VS	BSS1	VS-P
MAT1-Ü	BSS1-Ü	MAT1-Ü	BSS1-Ü	VS-P	BSS1-Ü	BSS2
MAT1-Ü	BSS1-Ü	MAT1-Ü	BSS1-Ü	VS-P	BSS1-Ü	BSS2-P
MAT1-Ü	FÜ M	MAT1-Ü	RNT	VS-P	RNT	BSS2-Ü
MAT1-Ü		MAT1-Ü	RNT-P	BSS2	RNT-P	RT
it3		SWT/it3	RNT-P	BSS2-P	it3-Ü	RT (M)
it3		SWT-Ü	RNT-P	BSS2-P	it3-Ü	ARN
PRO-Ü		SWT-Ü		BSS2-P		
		it3-Ü		RT		
		it3-Ü		RT (M)		

Aktive Teilnahme an  
Übungen/Praktikum!

---

# Veranstaltung

---

## Vorlesung:

- 0 Einführung
- 1 Grundlagen
- 2 Middleware
- 3 Shared Data
- 4 System-Infrastruktur
- 5 Verteilte Algorithmen

## Praktikum:

- 1to1(Linux)
- Peer to Peer (Linux)
- Homepage KiGa
- SSL (Klausurergebnisse im Internet?)
- LAMP?
- weitere?

**Ausblick: Mobilfunkprotokolle (3G, Summer school), BSS2 (RT-OS)**

---

# Veranstaltung (detailliert)

---

## **0 Einführung**

## **1 Grundlagen**

- 1.1 Charakteristische Eigenschaften verteilter Systeme
- 1.2 Systemmodelle
- 1.3 Netzwerke
- 1.4 Interprozesskommunikation

## **2 Middleware**

- 2.1 Verteilte Objekte und externer Aufruf
- 2.2 Sicherheit
- 2.3 Namensdienste

## **3 Shared Data**

- 3.1 Transaktionen und Nebenläufigkeitskontrolle
- 3.2 Verteilte Transaktionen
- 3.3 Replikation

## **4 System-Infrastruktur**

- 4.1 OS-Unterstützung
- 4.2 Verteilte Dateisysteme
- 4.3 Verteilter gemeinsame genutzter Speicher

## **5 Verteilte Algorithmen**

- 5.1 Zeit und globale Zustände
- 5.2 Koordination der Übereinstimmung

---

# Veranstaltung (detailliert)

---

## **0 Einführung**

## **1 Grundlagen**

- 1.1 Charakteristische Eigenschaften verteilter Systeme /Coulouris02/, Kap.1
- 1.2 Systemmodelle /Coulouris02/, Kap.2
- 1.3 Netzwerke /Coulouris02/, Kap.3
- 1.4 Interprozesskommunikation /Coulouris02/, Kap.4

## **2 Middleware**

- 2.1 Verteilte Objekte und externer Aufruf /Coulouris02/, Kap.5
- 2.2 Sicherheit /Coulouris02/, Kap.7
- 2.3 Namensdienste /Coulouris02/, Kap.9

## **3 Shared Data**

- 3.1 Transaktionen und Nebenläufigkeitskontrolle /Coulouris02/, Kap.12
- 3.2 Verteilte Transaktionen /Coulouris02/, Kap.13
- 3.3 Replikation /Coulouris02/, Kap.14

## **4 System-Infrastruktur**

- 4.1 OS-Unterstützung /Coulouris02/, Kap.6
- 4.2 Verteilte Dateisysteme /Coulouris02/, Kap.8
- 4.3 Verteilter gemeinsame genutzter Speicher /Coulouris02/, Kap.16

## **5 Verteilte Algorithmen**

- 5.1 Zeit und globale Zustände /Coulouris02/, Kap.10
- 5.2 Koordination der Übereinstimmung /Coulouris02/, Kap.11

---

# Praktikumstermine

---

Gruppe 1	Gruppe 2	Gruppe 3
1.V 05.10.04		
	1.V 12.10.04	
		1.V 19.10.04
2.V 26.10.04		
	2.V 02.11.04	
		2.V 09.11.04
3.V 16.11.04		
	3.V 23.11.04	
		3.V 30.11.04
4.V 07.12.04		
	4.V 14.12.04	
		4.V 21.12.04
5.V 11.01.05		
	5.V 18.01.05	
		5.V 25.01.05





---

# Voraussetzungen

---

Für Informatik-Studenten des Hauptstudiums gegeben:

- Vertrautheit mit Algorithmen und algorithmischer Denkweise
- Vertrautheit mit einfachen Datenstrukturen (Arrays, verkettete Listen, Bäume)
- Kenntnis einer höheren Programmiersprache (z.B. C++, Beispiele häufig in Pseudocode)
  - einschließlich Programmiererfahrung, dabei
  - sicherer Umgang mit einem modernen OS (z.B. NT, User-Niveau)
- Anfangskenntnisse der Nachrichtentechnik
  - Information (Alphabet, Semantik, Syntax)
  - Informationsgehalt, Entropie, Redundanz
  - Nachrichtenkanal (mit Gedächtnis  $k$ -ter Ordnung, gedächtnislos)
  - Verluste, Rauschen
  - Abtasttheorem
  - Quellen- und Kanalkodierung
- Erfolgreiche Teilnahme an Veranstaltung „Rechnernetze und Telekommunikation“

# „Rechnernetze“ vs. „Verteilte Systeme“

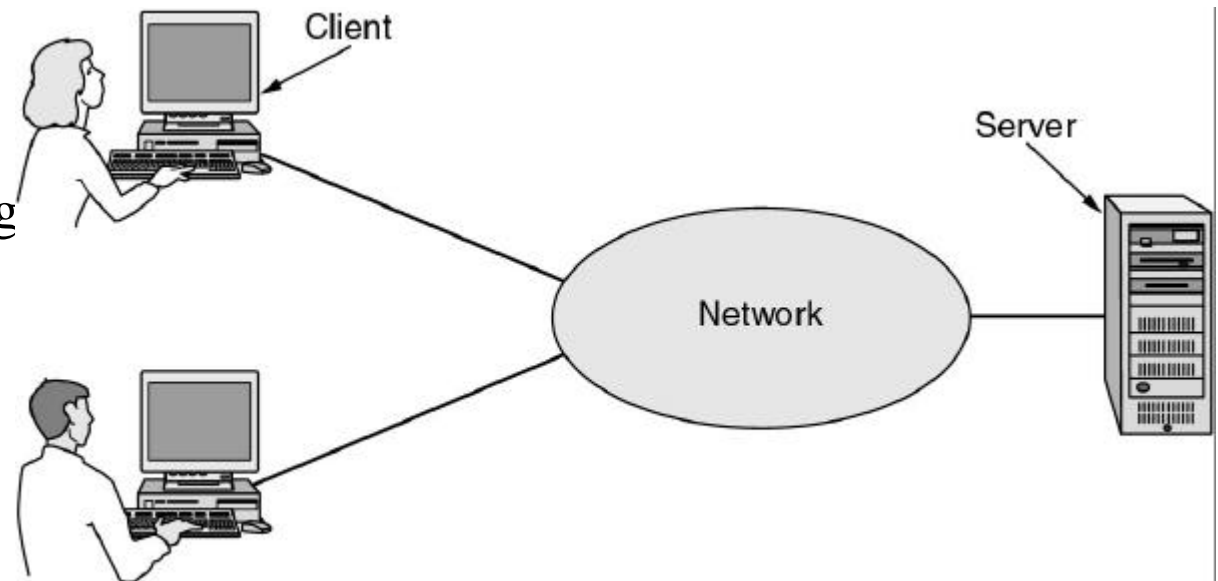
Definition: **Rechnernetze** := (mehrere) miteinander verbundene autonome Computer, d.h. BenutzerIn verteilt Aufgaben auf Netzwerkkomponenten.

**Nicht:**

- Master/Slave
- CAN

Definition: Verteiltes System := Rechner unter „gemeinsamer Führung“ d.h. OS verteilt Aufgaben, transparent für Benutzer.

Viele Überschneidungen!



---

# „Rechnernetze“ vs. „Verteilte Systeme“

---

Definition /Coulouris02/:

Bei einem **verteilten System** arbeiten Komponenten zusammen, die sich auf **vernetzten Computern** befinden und die ihre Aktionen durch den **Austausch von Nachrichten** koordinieren.

Daraus folgend Eigenschaften:

- Nebenläufigkeit der Komponenten
- keine globale Uhr
- Komponenten können unabhängig voneinanderausfallen

Beispiele:

- Internet
- Intranet eines Unternehmens
- mobiles und allgegenwärtiges Rechnen

Motivation: u.a. Nutzung gemeinsamer Ressourcen (HW u. SW!)

Viele Überschneidungen RN/VS!

---

## Die Welt als „Verteiltes System“

---



Auch die "reale Welt" ist ein verteiltes System:

- Viele gleichzeitige ("parallele") Aktivitäten
- Exakte globale Zeit nicht erfahrbar / vorhanden
- Keine konsistente Sicht des Gesamtzustandes
- Kooperation durch explizite Kommunikation
- *Ursache* und *Wirkung* zeitlich (und räumlich) getrennt

---

## Weitere Definitionen „Verteilte Systeme“

---

Definition /Tanenbaum95/:

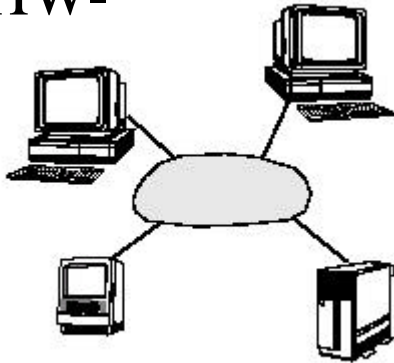
Ein **verteiltes System** ist eine Sammlung voneinander unabhängiger Computer, die dem Benutzer des Systems den Eindruck vermitteln, es handele sich um einen einzigen Computer.

A distributed computing system consists of multiple autonomous processors that do not share primary memory, but cooperate by sending messages over a communication network.

-- *H. Bal*

# Sichten auf „Verteilte Systeme“

„HW-“



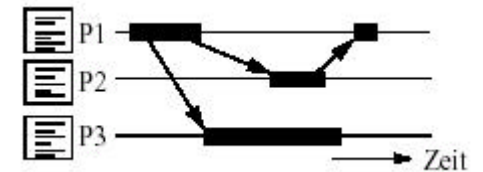
„Programmierer-“



Objekte eines Betriebssystems bzw. einer Programmiersprache

==> "Programmiersicht" (Client, Server...)

„Algorithmen-/Protokoll-“



- Aktionen, Ereignisfolgen  
- Konsistenz, Korrektheit

Rechnernetz mit Rechenknoten:

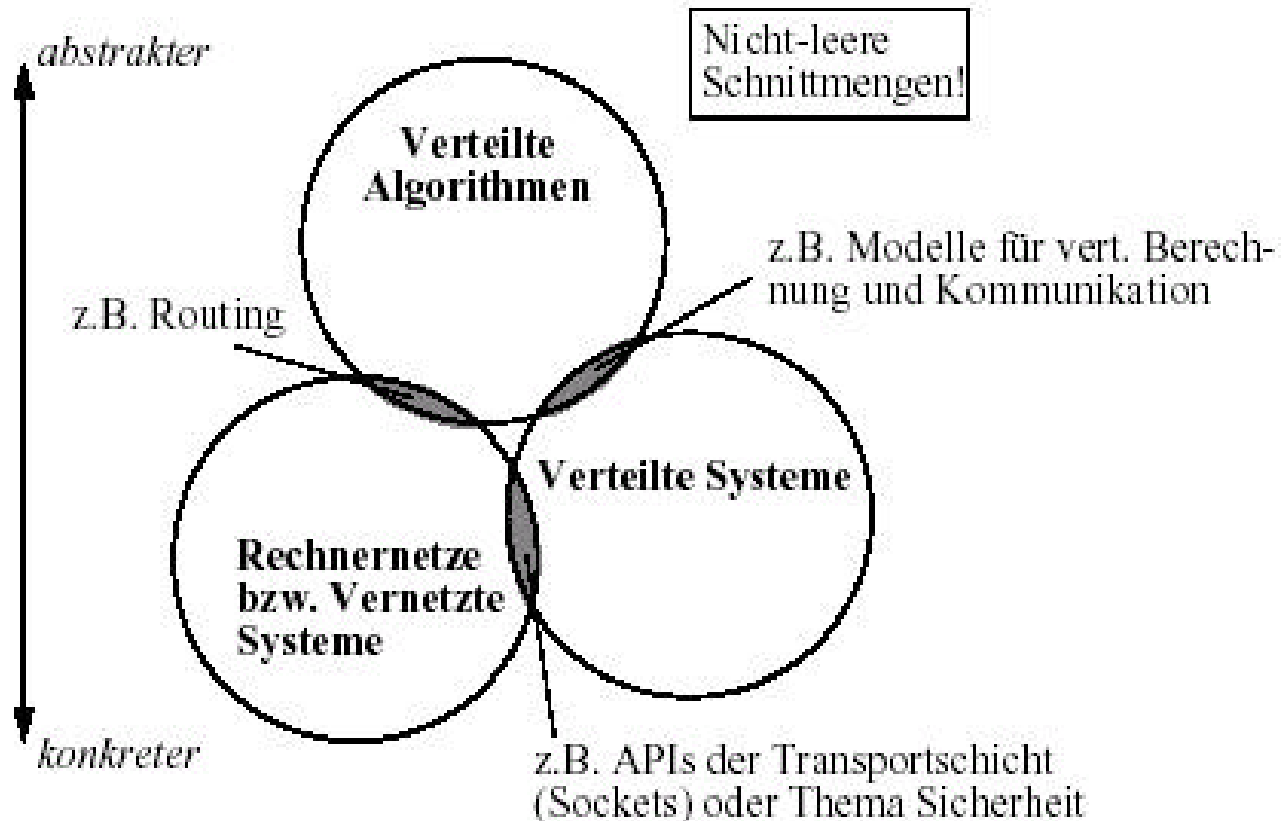
- LAN = Local Area Network
- WAN = Wide Area Network
- Multicomputer (Parallelrechner)
- Routing, Adressierung....

**zunehmende Abstraktion**

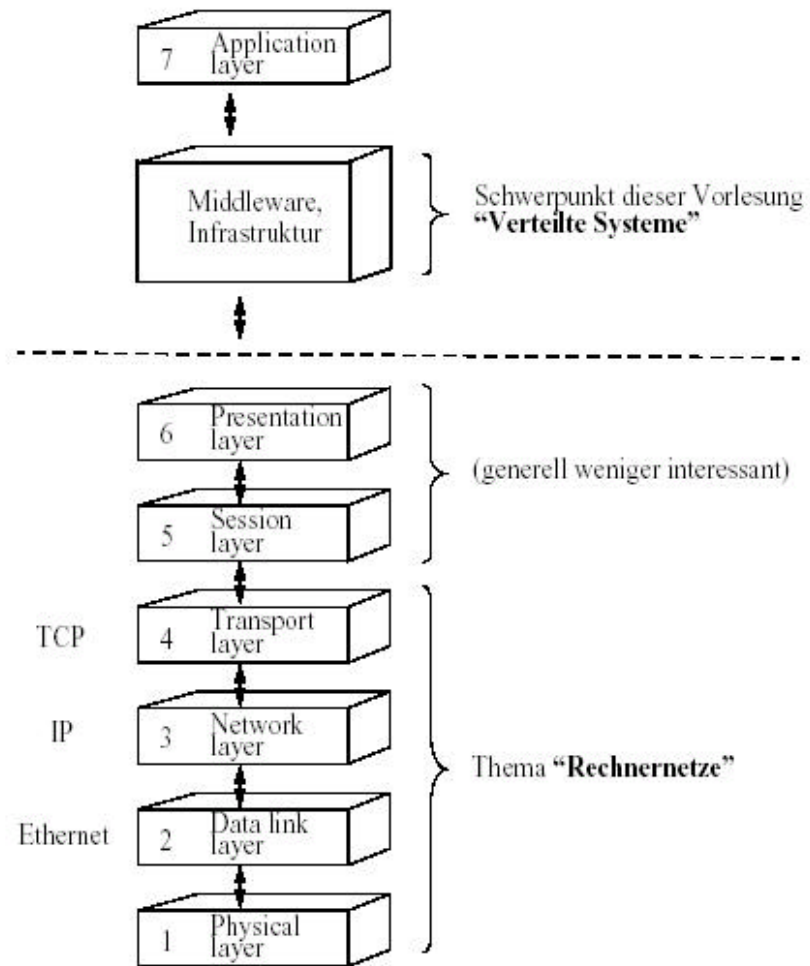
---

# Warum „Verteilte Systeme“ – wir haben doch schon „Rechnernetze und Telekommunikation“ gehört!

---

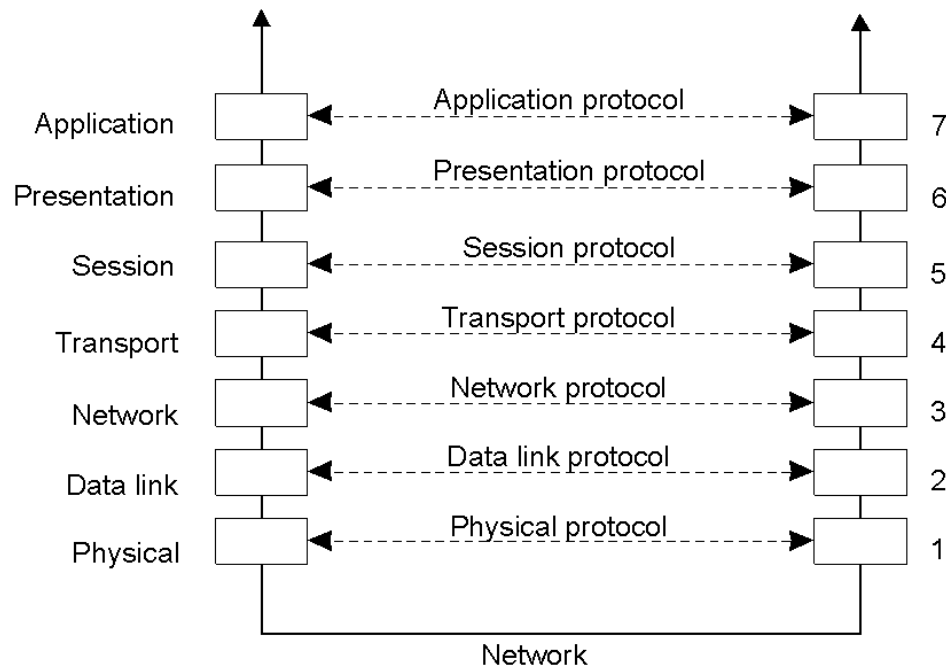


# Schichten 6.1 bis 6.9

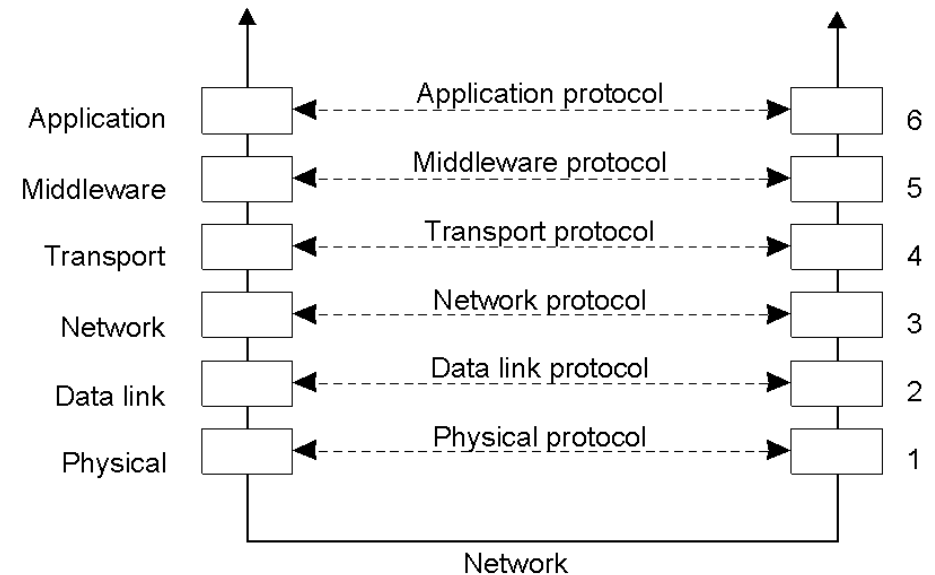




# Schichten 6.1 bis 6.9



/Tanenbaum03/



Layers, interfaces,  
and protocols in the OSI model.

An adapted reference model  
for networked communication.

---

## Probleme/Aufgabenstellungen „Verteilte Systeme“

---

- Heterogenität der Komponenten
- Offenheit (erlaubt, dass Komponenten hinzugefügt oder ersetzt werden)
- Sicherheit
- Skalierbarkeit (funktionieren auch noch, wenn Anzahl der Benutzer steigt)
- Fehlerverarbeitung
- Nebenläufigkeit der Komponenten
- Transparenz

---

## Techniken/Lösungsansätze „Verteilte Systeme“

---

- Interprozesskommunikation
- entfernte Aufrufe (u.a. RPC)
- verteilte Namensgebung
- verschlüsselnde Sicherheit
- verteilte Dateisysteme
- Datenreplikation
- verteilte Transaktionsmechanismen
- ...

stellen Laufzeit-Infrastruktur bereit, die für VS benötigt werden.

---

## „Middleware“ für „Verteilte Systeme“

---

Entwicklung VS über Middleware-Unterstützung, SW-Umgebungen, die Abstraktionen erlauben:

- verteilte gemeinsam genutzte Objekte
- verteilte Dienste:
  - sichere Kommunikation
  - Authentifizierung
  - Zugriffskontrolle
  - mobiler Code
  - Transaktionen
  - persistente Speichermechanismen

---

## (nahe) Zukunft „Verteilte Systeme“

---

Verteilte Applikationen:

enge Zusammenarbeit Benutzer über

- replizierte Daten
- Multimedia-Datastreams

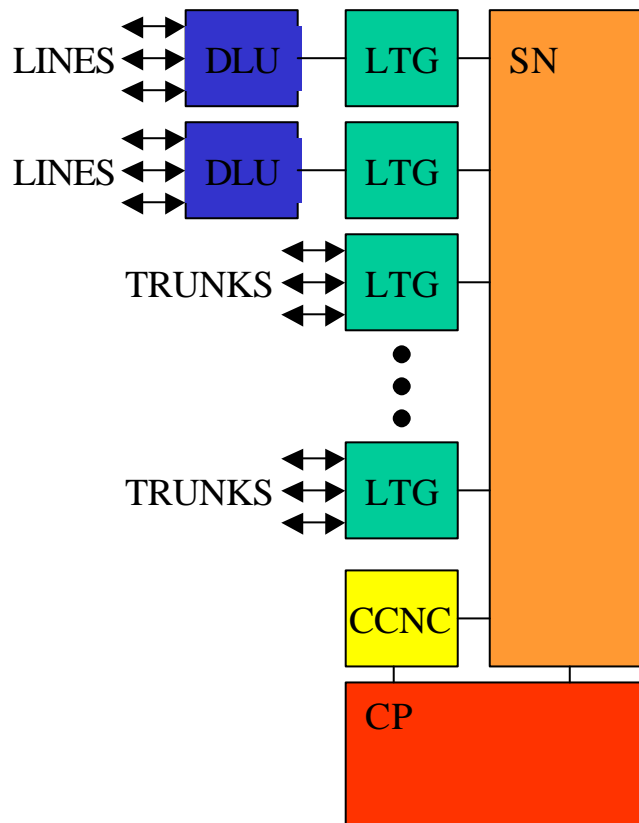
Unterstützung der Mobilität von

- Benutzern und
- Geräten

unter Verwendung von

- Funk- (z.B. UMTS) und
- spontanen Netzwerken (z.B. WLAN)

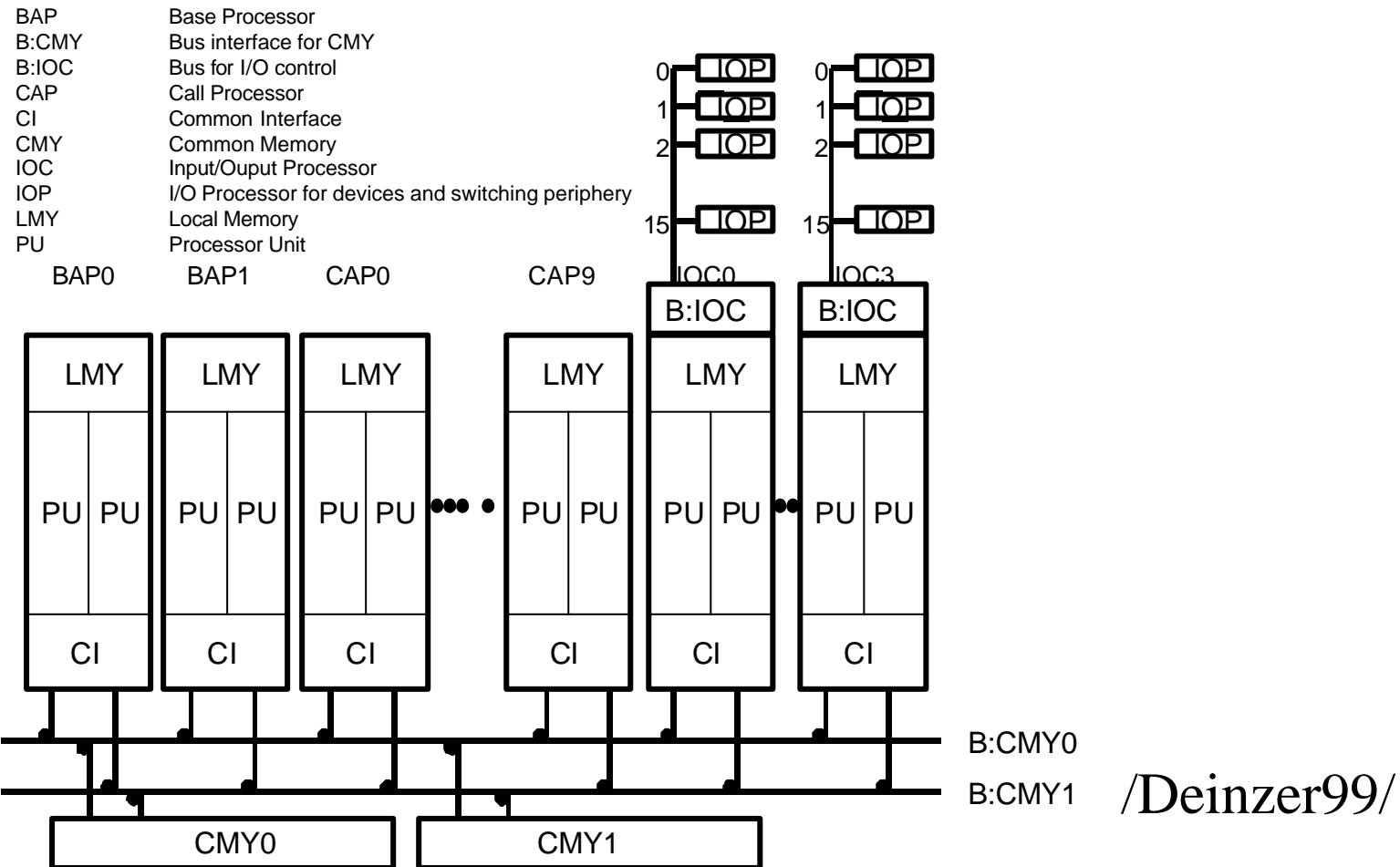
## (Extremes) Beispiel „Verteilte Systeme“



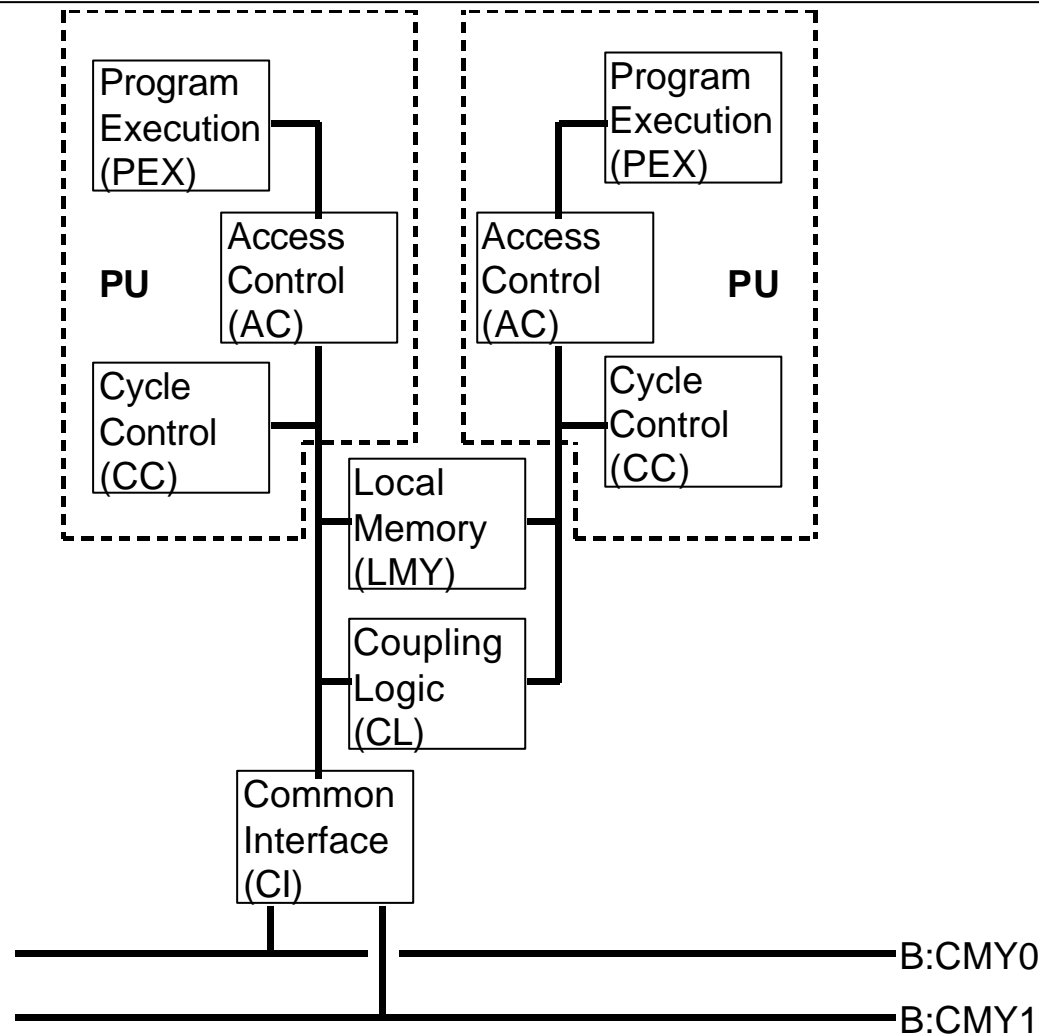
CCNC: Common Channel  
Signaling Network Control  
CP: Coordination Processor  
DLU: Digital Line Unit  
LTG: Line Trunk Group  
SN: Switching Net

/Deinzer99/

# Beispiel „Verteilte Systeme“ – Feinstruktur CP



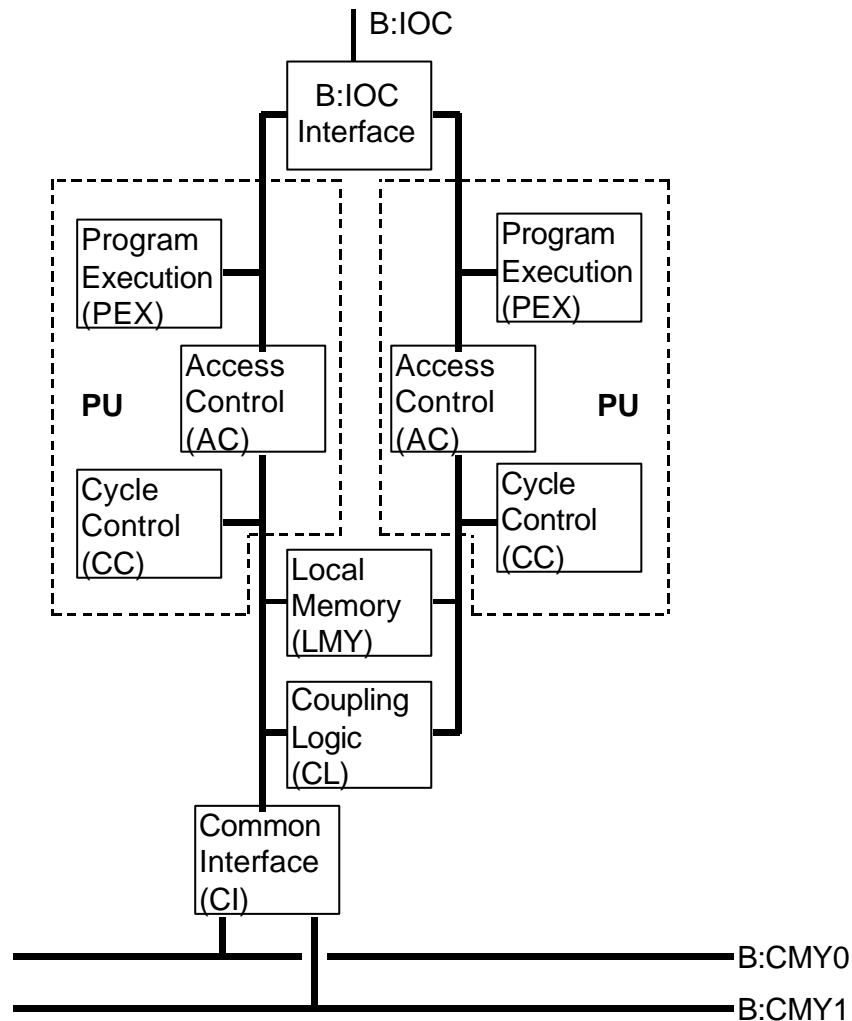
# Beispiel „Verteilte Systeme“ – Feinstruktur CP.BAP/CAP



/Deinzer99/

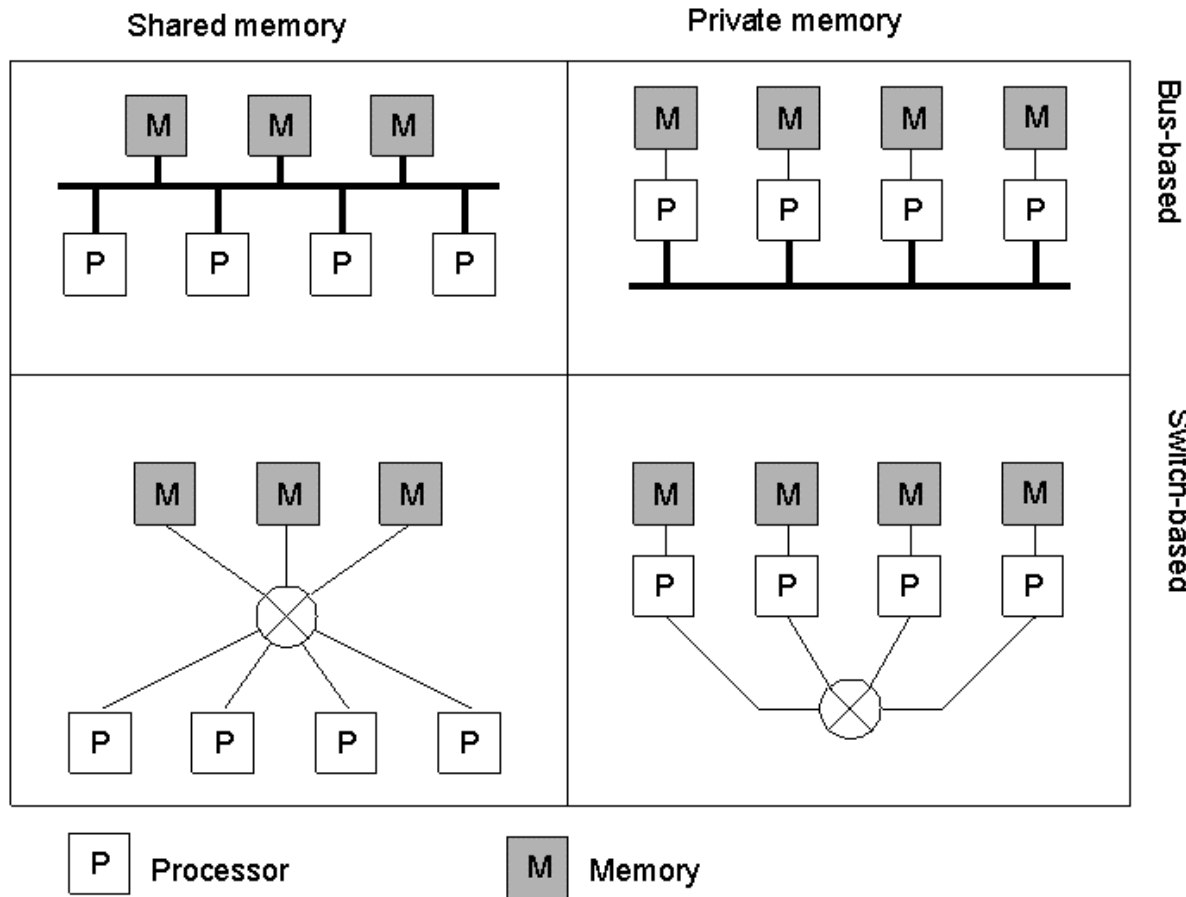


# Beispiel „Verteilte Systeme“ – Feinstruktur CP.IOC



/Deinzer99/

# HW-Konzepte Multiprozessoren



Bus-based

Switch-based

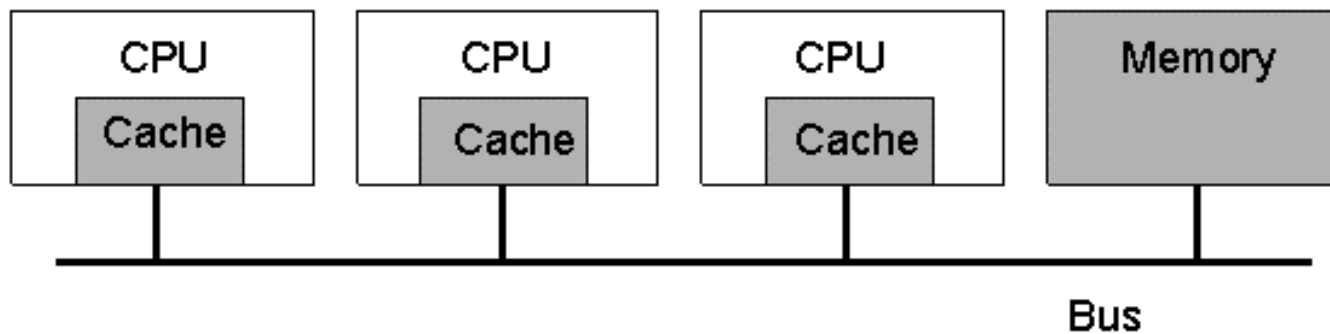
Different basic organizations and memories in distributed computer systems

/Tanenbaum03/

---

# HW-Konzepte Multiprozessoren - vgl. CP113!

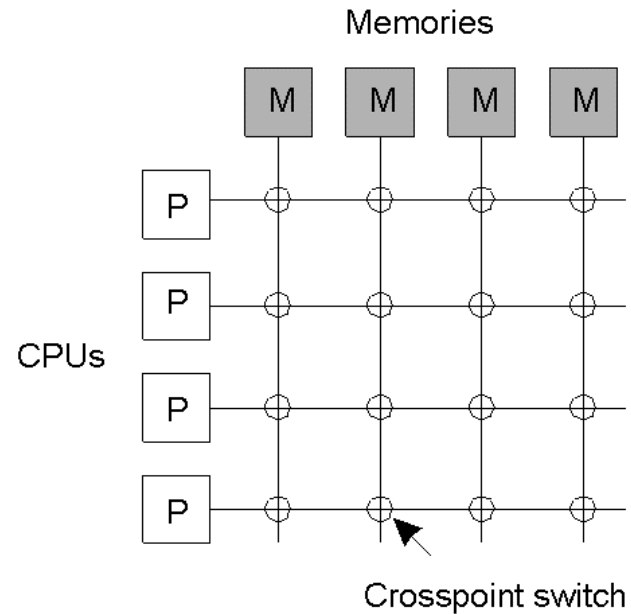
---



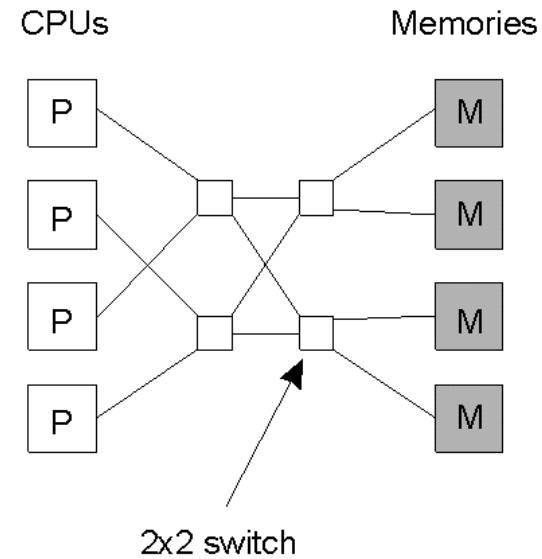
A bus-based multiprocessor.

/Tanenbaum03/

# HW-Konzepte Multiprozessoren



(a)



(b)

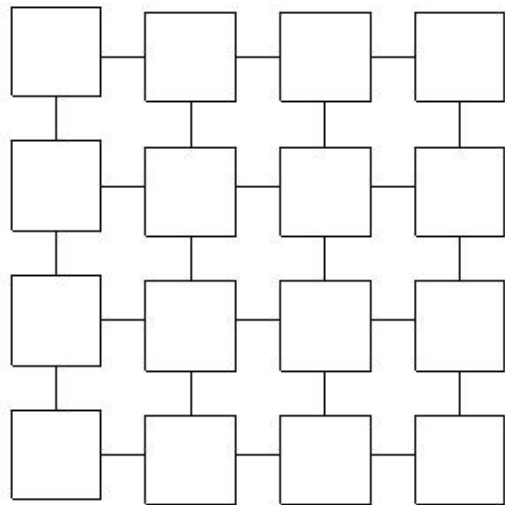
- a) A crossbar switch
- b) An omega switching network

/Tanenbaum03/

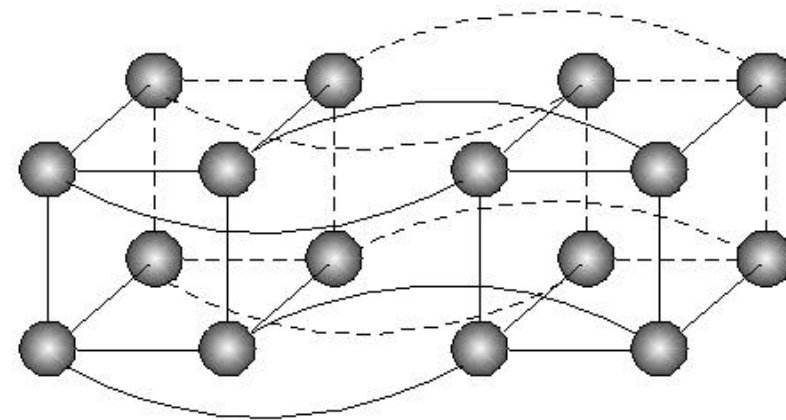
---

# HW-Konzepte Multicomputer

---



(a)



(b)

- a) Grid ("Matrix"!)
- b) Hypercube

/Tanenbaum03/

---

## Noch ein VS-Beispiel

---



LAN-Party  
31.05.2003  
FHK

---

## Noch ein VS-Beispiel

---



LAN-Party  
31.05.2003  
FHK

---

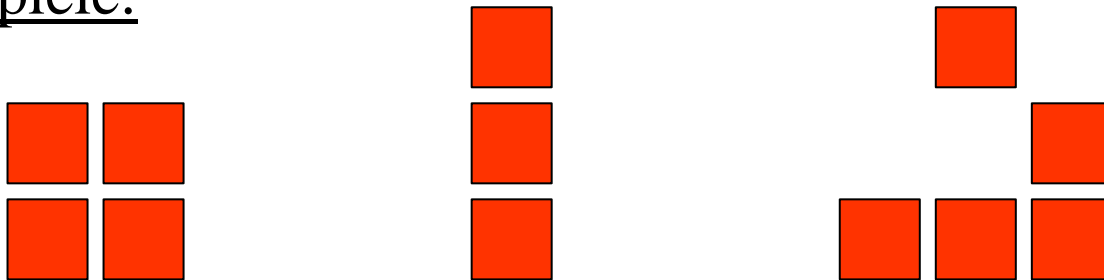
## Noch ein VS-Beispiel

---

Spiel des Lebens: Kästchentopologie, jede Zelle hat 8 Nachbarn

- lebendige Zelle bleibt lebendig, wenn sie 2 oder 3 lebendige Nachbarzellen hat
- eine tote Zelle wird lebendig, wenn sie genau 3 lebendige Nachbarzellen hat

Beispiele:



Großes VS: jede Zelle CPU, mit 8 Nachbarn verbunden (vgl. "Grid, 0.28), alle Zellen führen synchron den einfachen Überlebensalgorithmus aus.

Es ist **nicht** möglich, einer gegebenen Zellgeneration "anzusehen", was aus ihr wird (z.B. ob sie ausstirbt oder sich unbegrenzt ausbreitet) - Halteproblem.



---

## Literatur - vgl. auch Literaturliste RNT!

---

Tanenbaum, Andrew S.; van Steen, Marten  
Verteilte Systeme (1. Auflage), Pearson 2003

Tanenbaum, Andrew S.  
Moderne Betriebssysteme (2. Auflage), Pearson 2002

Coulouris, George; Dollimore, Jean; Kindberg, Tim  
Verteilte Systeme (3. Auflage), Pearson 2002

Bengel, Günther  
Grundkurs Verteilte Systeme (3. Auflage), Vieweg 2004

Badach A.; Rieger S.; Schmauch M.  
Web-Technologien, Hanser 2003

Herrtwich R.G.; Hommel G.  
Kooperation und Konkurrenz, Springer 1989

Zöbel, Dieter  
Echtzeitsysteme, Thomson Publishing 1995