

Verteilte Systeme

Vorlesung 3 vom 22.04.2004
Dr. Sebastian Iwanowski
FH Wedel

Inhaltsverzeichnis für die Vorlesung

Zur Motivation: 4 Beispiele aus der Praxis

Allgemeine Anforderungen an Verteilte Systeme

Konzepte verteilter Hardware

→ Die Client-Server-Beziehung und daraus entstehende Fragestellungen

Grundlagen der Kommunikation in verteilten Systemen

Nebenläufigkeitstechniken

Entfernte Aufrufe / Objektmigration

Namensverwaltung / Namenssuche

Dienstevermittlung

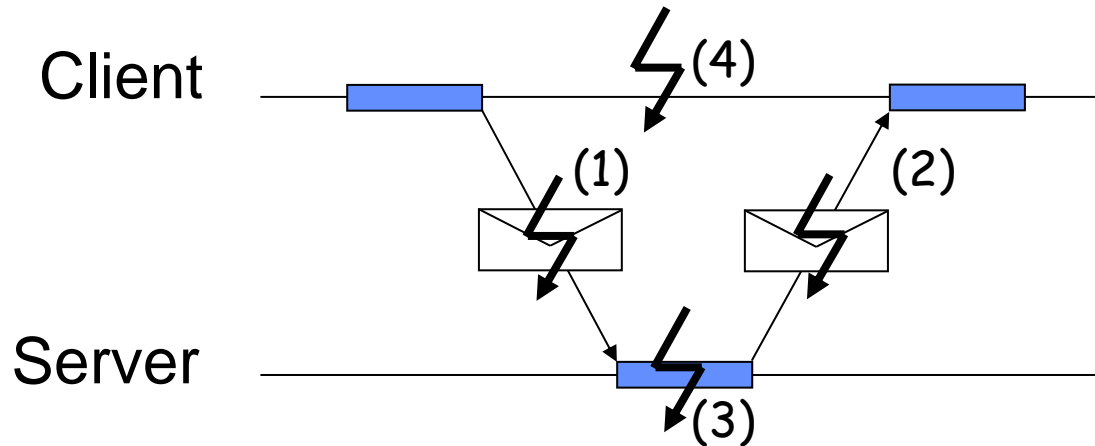
Synchronisation von Daten und Prozessen

Konzepte zur Erzielung von Fehlertoleranz

Sicherheit

Ausblick auf konkrete Software: J2EE, SOAP,...

Probleme in der Client-Server-Kommunikation



Client wartet und versucht...

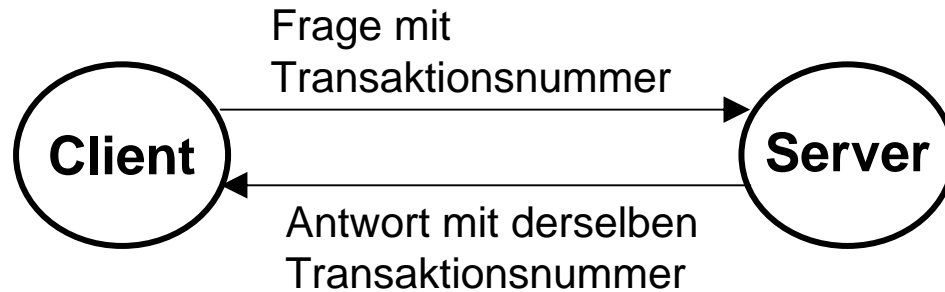
... nach Timeout ein erneutes Senden,

kann aber nicht zwischen verschiedenen Fehlersituationen unterscheiden.

Erneutes Senden führt zur erneuten Ausführung.

Problem: Wie erkennt der Server, dass der Client dieselbe Anfrage noch einmal gestellt hat und nicht eine neue gestellt hat ?

Lösungsansatz: Protokolle / Transaktionskonzept



wichtig:

- Transaktionsnummer ist im ganzen Netzwerk eindeutig !

Was ist eine Transaktion ?

Eine **Transaktion** ist eine Folge von Operationen, die entweder alle vollständig oder alle überhaupt nicht durchgeführt werden sollen.

Was ist ein Protokoll ?

Ein **Protokoll** ist ein Regelwerk für Kommunikationsaktionen. Es legt die Abfolge der Aktionen und die zu benutzenden Formate fest, unterscheidet die Rollen der beteiligten Kommunikationspartner und legt eventuell weitere logische Zusammenhänge fest.

Lösungsansatz: Protokolle / Transaktionskonzept

Warum brauchen wir Transaktionen ?

Bsp. Bankkonto: Umbuchung eines Betrages von Konto A nach Konto B

geplant:


Umbuchung

```
read (A, a1)
a1 := a1 - 300
write (A, a1)
read (B, b1)
b1 := b1 + 300
write (B, b1)
```

tatsächlicher Verlauf:

Umbuchung

```
read (A, a1)
a1 := a1 - 300
write (A, a1)
read (B, b1)
```

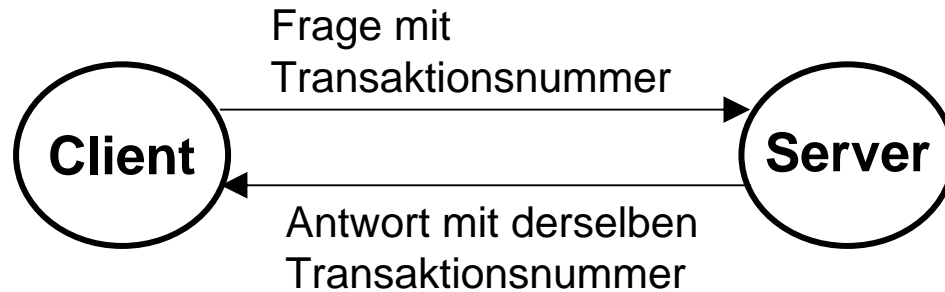
Störung 

Wo sind die 300 € geblieben?

Warum brauchen wir Protokolle ?

- zur effizienten Beschreibung eines gewünschten Kommunikationsschemas
- weitere Beispiele im Abschnitt „Grundlagen der Kommunikation“

Lösungsansatz: Protokolle / Transaktionskonzept



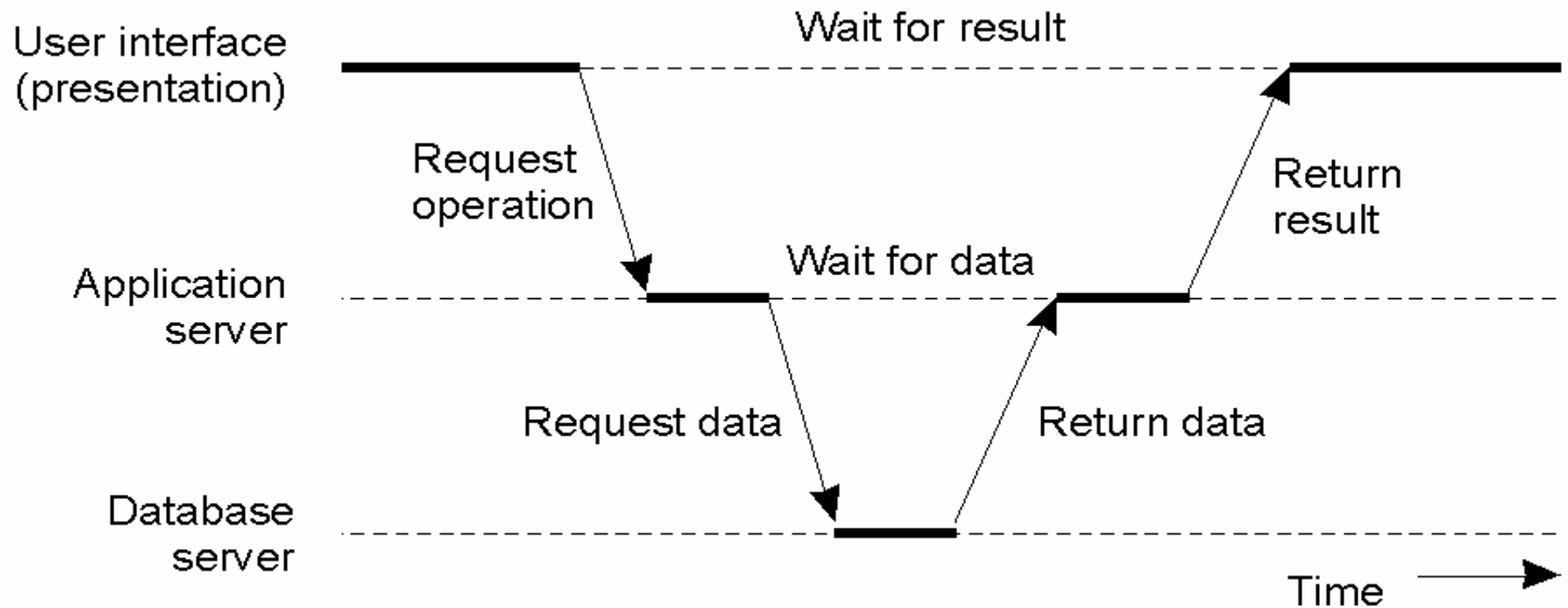
wichtig:

• **Transaktionsnummer ist im ganzen Netzwerk eindeutig !**

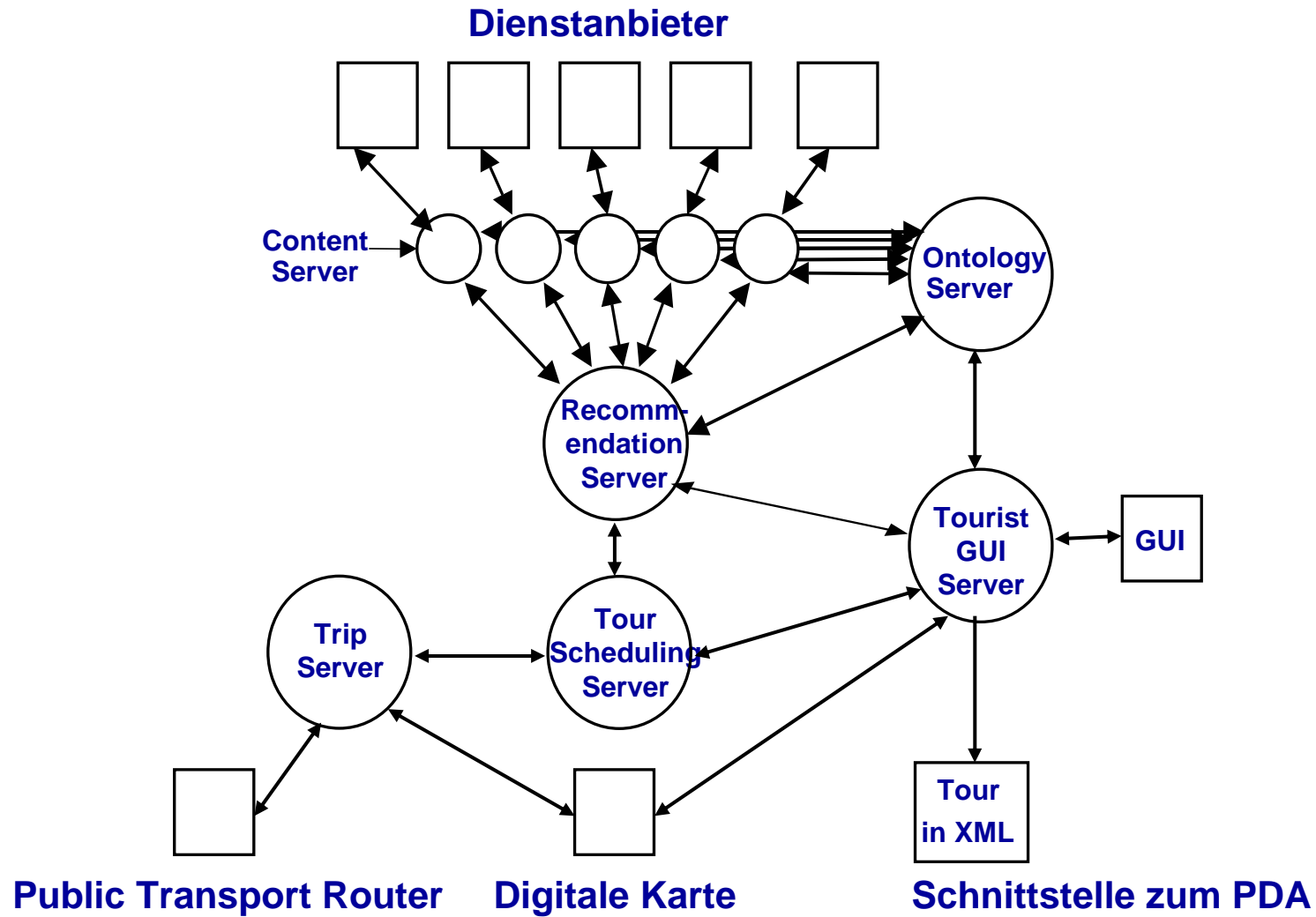
- Die Transaktionen werden innerhalb eines Protokolls ausgeführt
- Jede Operation derselben Transaktion hat dieselbe Transaktionsnummer und eine Ausführungsnummer, die seine Stellung innerhalb des Protokolls beschreibt
- Wenn eine Transaktion nicht ordnungsgemäß zu Ende geführt wurde, werden alle Operationen dieser Transaktion rückgängig gemacht.

Mehrschichten-Architektur (Multitiered architecture)

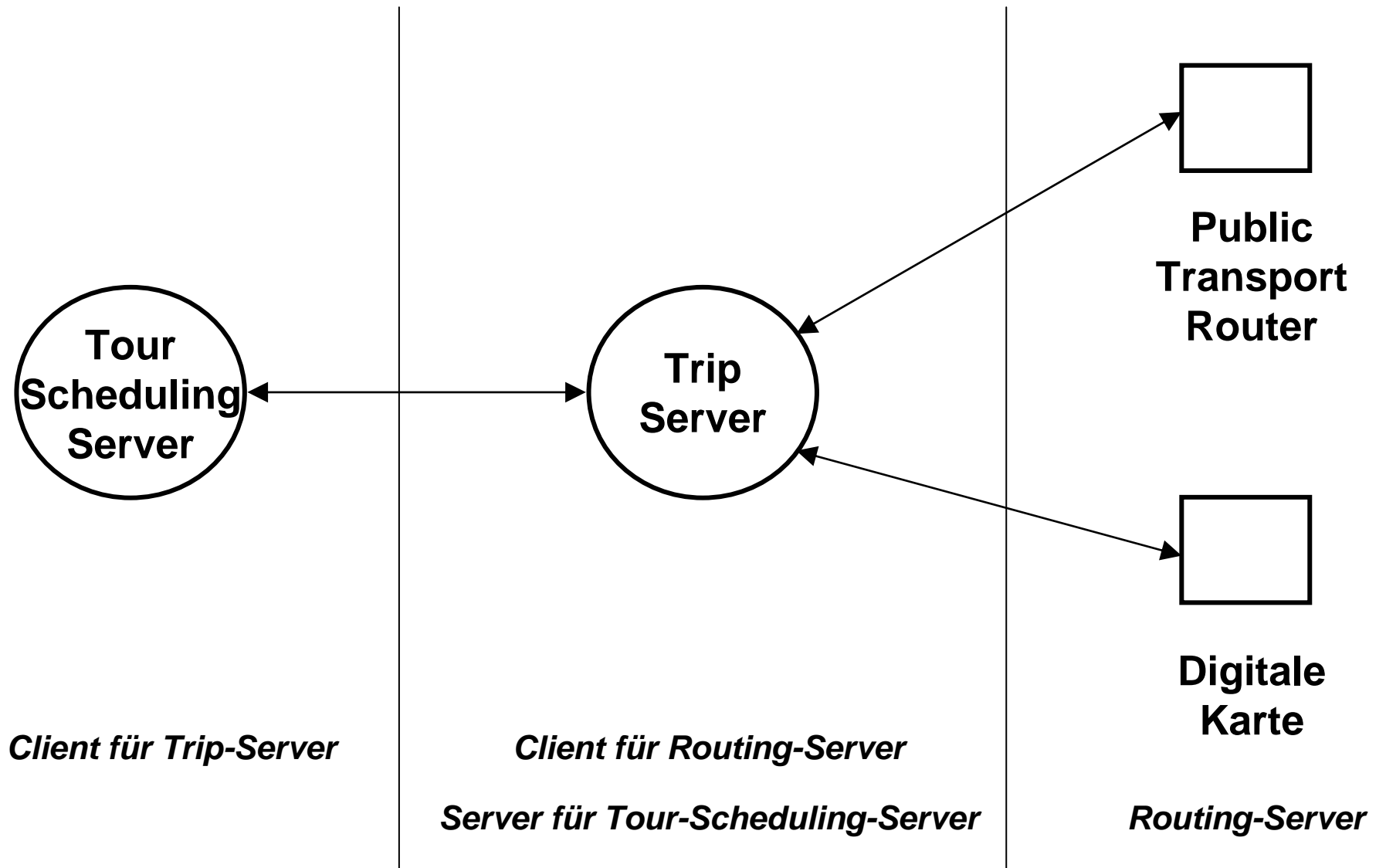
Beispiel für 3 Schichten:



Beispiel: Touristeninformationssystem



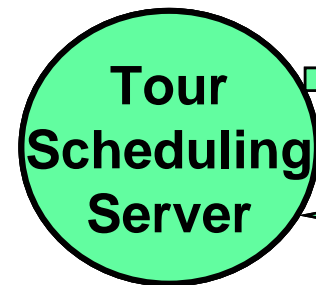
Bsp. für mehrere Schichten im Touristeninformationssystem



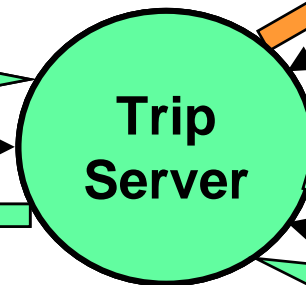
Bsp. für die Blockierung eines Prozesses

Anfrage 1:
erfordert Fahrt durch
die ganze Stadt

Anfrage 2:
erfordert kurzen Fußweg



Client für Trip-Server



Client für Routing-Server
Server für Tour-Scheduling-Server



**Public
Transport
Router**



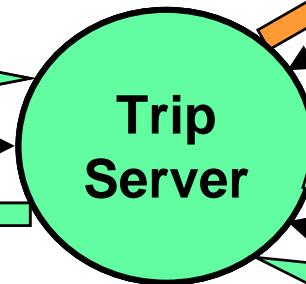
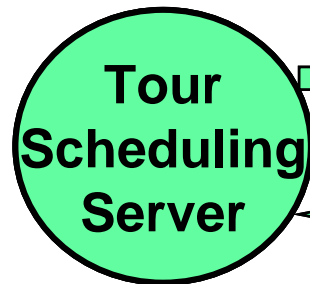
**Digitale
Karte**

Routing-Server

Bsp. für die Aufhebung der Blockade durch Multithreading

Anfrage 1:
erfordert Fahrt durch
die ganze Stadt

Anfrage 2:
erfordert kurzen Fußweg



**Public
Transport
Router**



**Digitale
Karte**

Client für Trip-Server

Client für Routing-Server
Server für Tour-Scheduling-Server

Routing-Server

→ Details im Abschnitt „Nebenläufigkeitstechniken“

Weitere Schwierigkeiten beim Touristeninformationssystem

Client und Server befinden sich an unterschiedlichen Orten

→ Lösungen im Abschnitt „Entfernte Aufrufe“

Heterogene Datenwelt in unterschiedlichen Servern

→ Lösungen im Abschnitt „Dienstevermittlung“

Grundlagen der Kommunikation in Verteilten Systemen

Bewertungskriterien für Kommunikationsnetze

Leistung (Performanz)

Verzögerungsrate (Latency)

Bandbreite (Data Transmission Rate)

Kapazität, Skalierbarkeit

Zuverlässigkeit (Reliability)

Sicherheit (Security)

Mobilitätsfähigkeit

Multicastingfähigkeit

Netzwerktypen

Local Area Networks (LAN)

Wide Area Networks (WAN)

Metropolitan Area Networks (MAN)

Wireless Networks (WLAN und WWAN)

Internet

	<i>Reichweite Bandbreite (Mbps) Verzögerung (ms)</i>		
LAN	1-2 km	10-1000	1-10
WAN	weltweit	0.010-600	100-500
MAN	2-50 km	1-150	10
Wireless LAN	0.15-1.5 km	2-11	5-20
Wireless WAN	weltweit	0.010-2	100-500
Internet	weltweit	0.010-2	100-500

Kommunikationstypen

verbindungsorientiert vs. verbindungsunabhängig

synchron vs. asynchron

Verschaltungstechnik

Massenübertragung (broadcast)

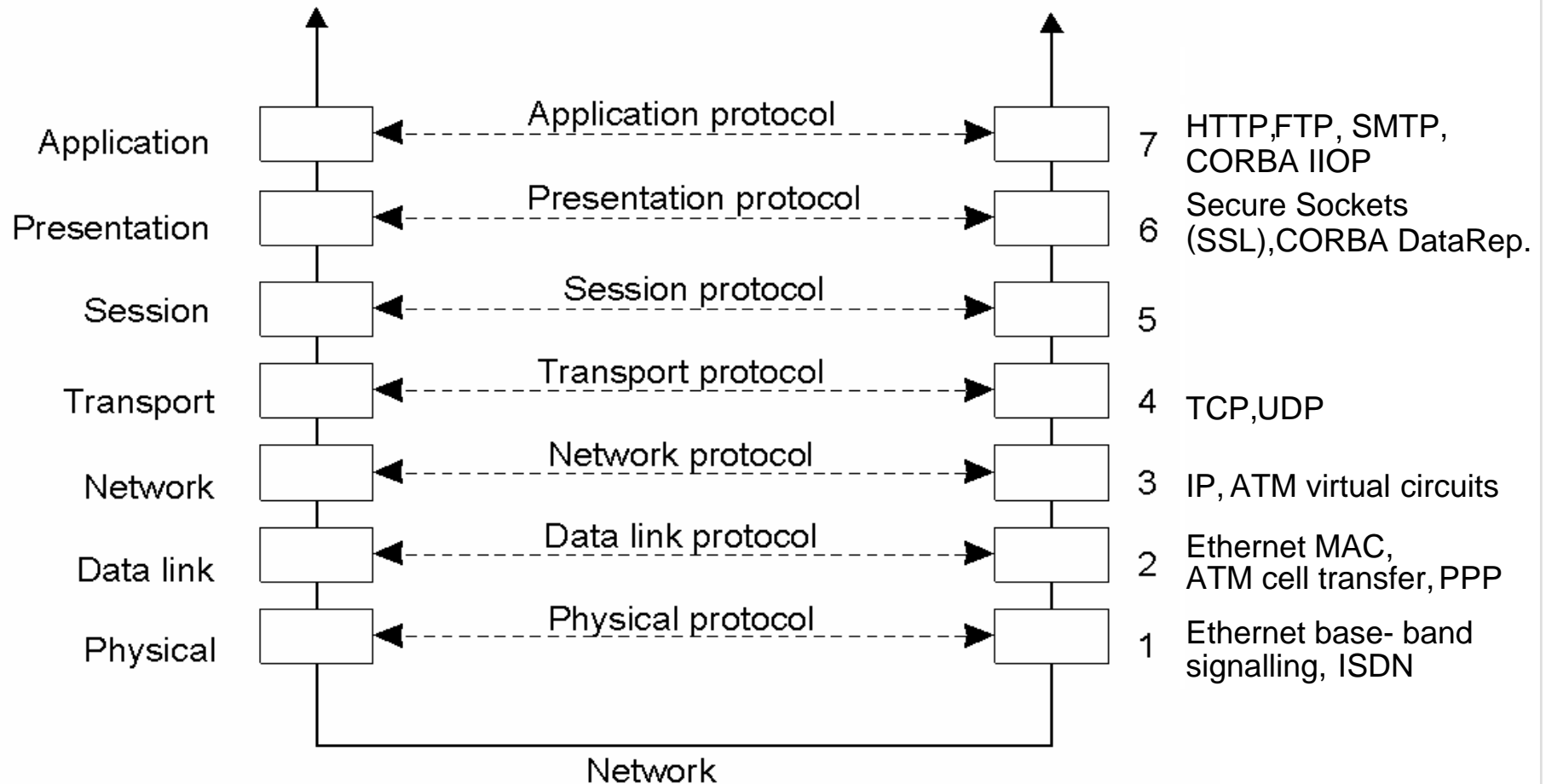
Kreisschaltung (circuit switching)

Bündelschaltung (packet switching)

ATM (asynchronous transfer networks)

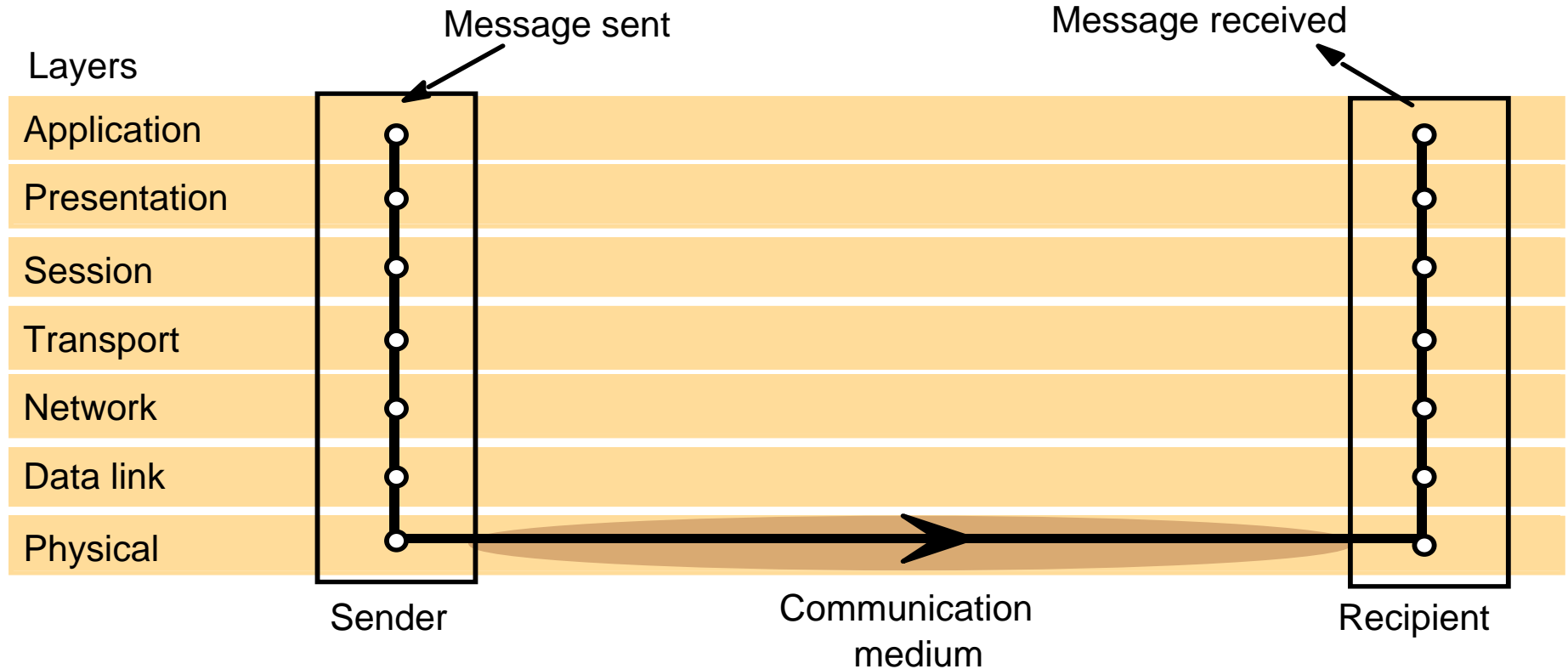
Kommunikationsprotokolle

Die Protokolle sichern Kommunikationseigenschaften auf verschiedenen Schichten (layers) ab:



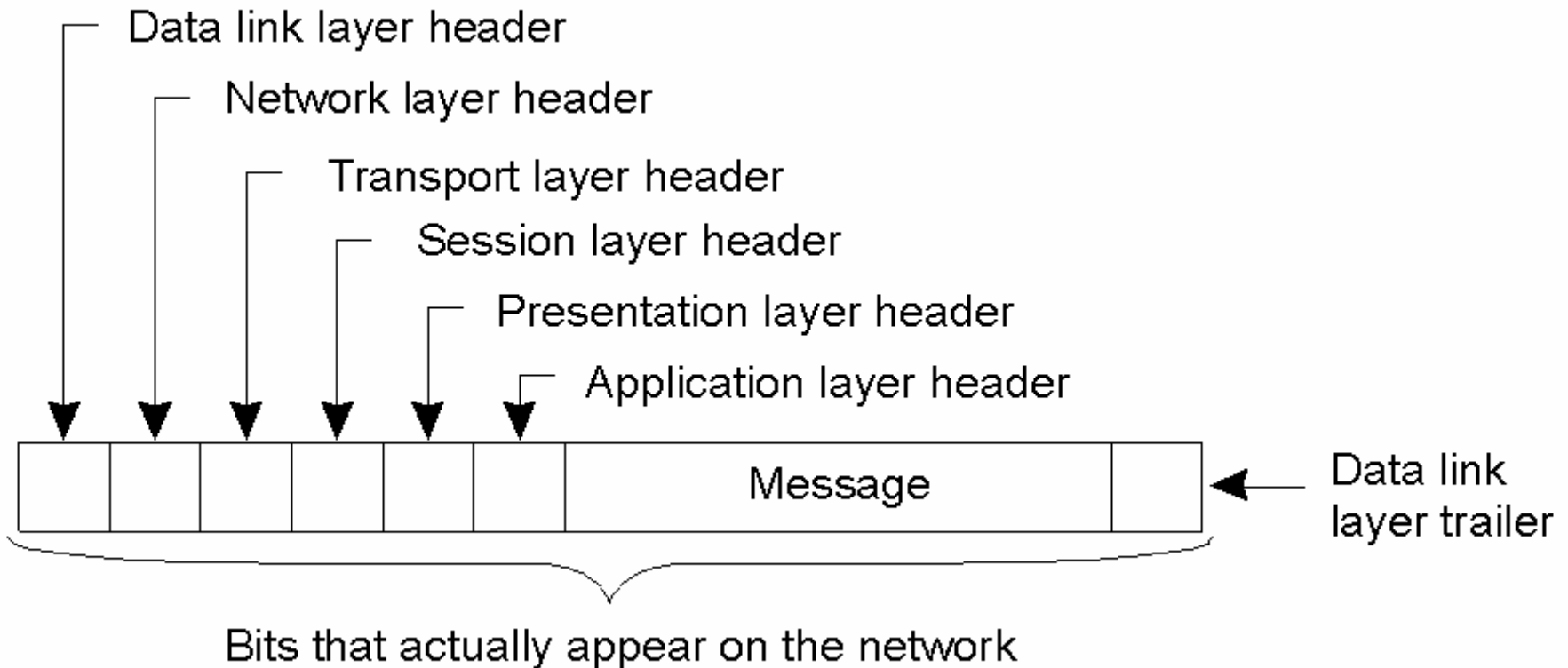
Kommunikationsprotokolle

Jedes Protokoll reichert die Nachricht beim Sender um zusätzliche Kontrollsequenzen an, die vom Empfänger wieder entfernt werden müssen:



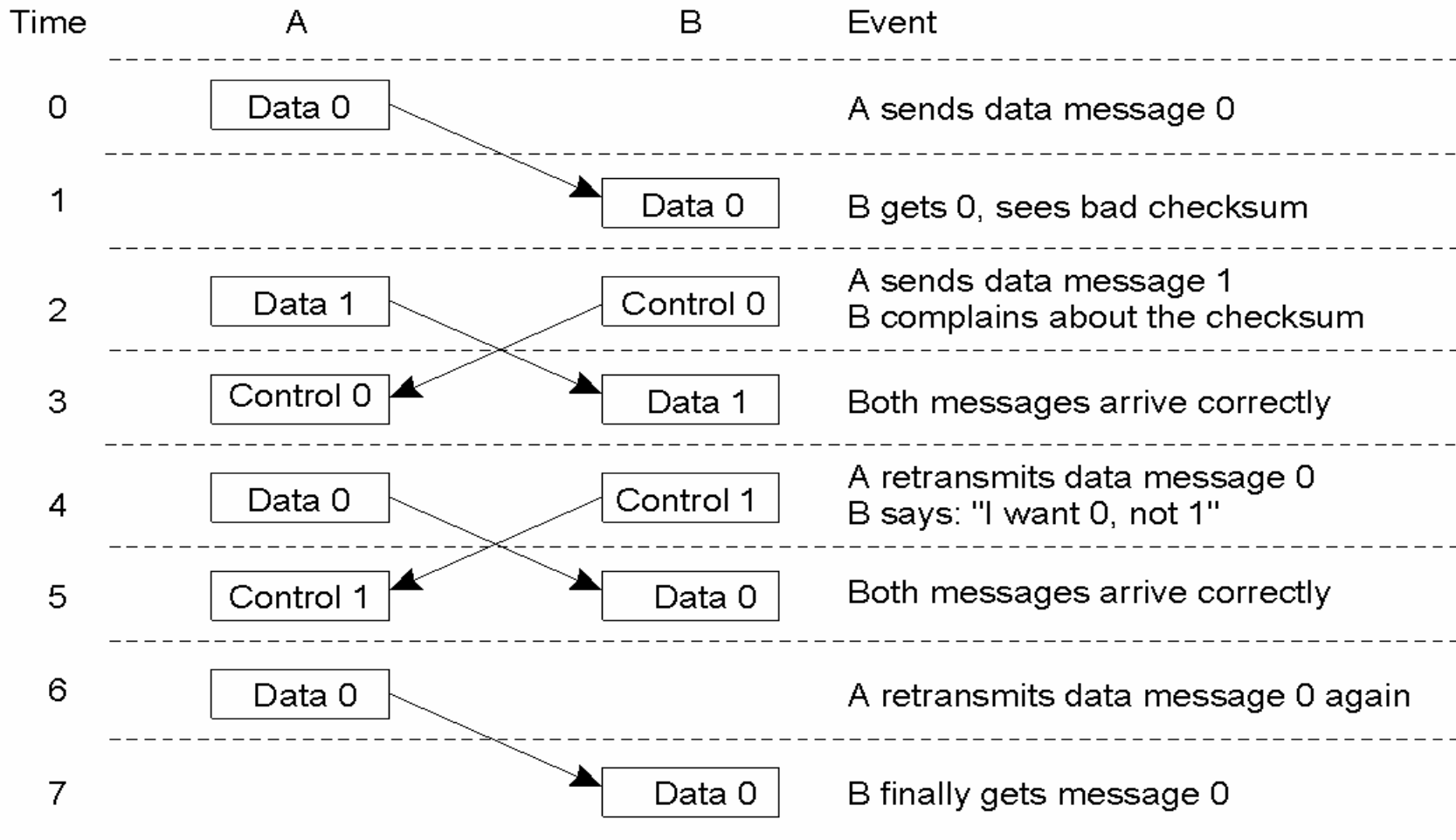
Kommunikationsprotokolle

Eine Nachricht sieht dann folgendermaßen aus:



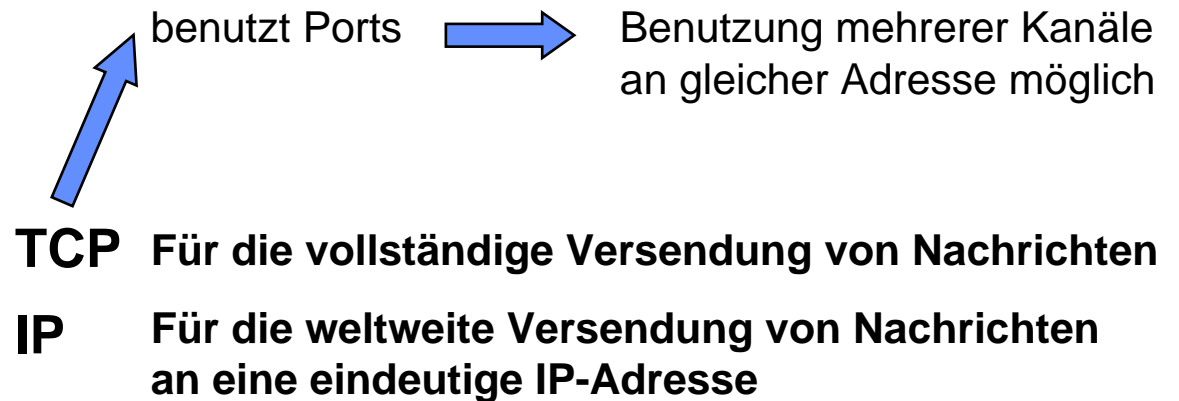
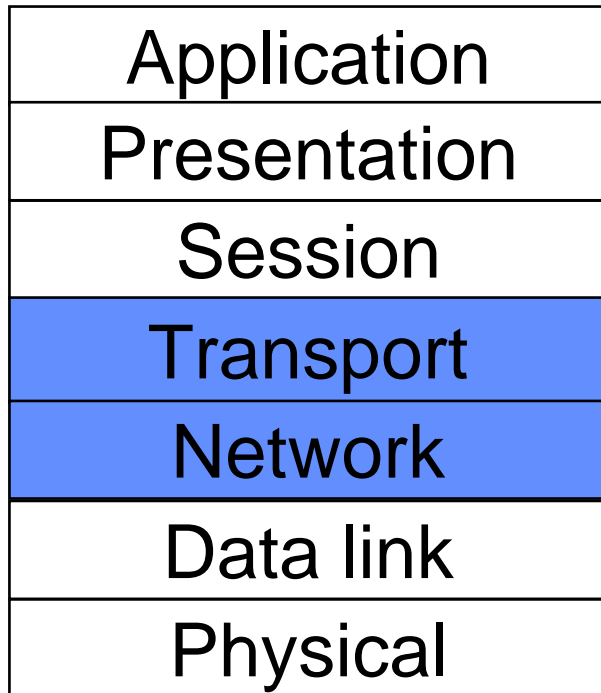
Kommunikationsprotokolle

Beispiel einer Kommunikation auf dem Data Link Layer (Schicht 2):



Kommunikationsprotokolle

Das wichtigste Kommunikationsprotokoll für die Erstellung von Software in verteilten Systemen: TCP / IP



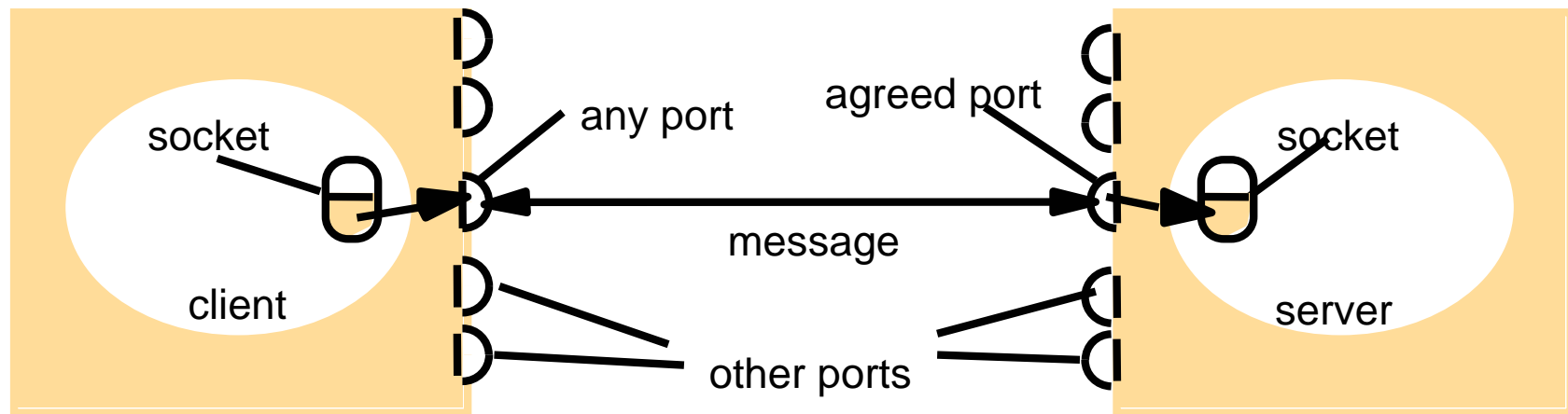
→ Näheres über die Funktionsweise von TCP/IP sowie der Alternative UDP/IP in der Vorlesung „Rechnernetze“ bei Herrn Kaleck

Die Benutzung einer TCP / IP – Verbindung („Socket-Schnittstelle“)

Adressierung eines Knotens (Computer)

- IP-Adresse (z.B. 134.100.12.135) oder Name
- Port (z.B. 8088)

Namensdienste: DNS



IP-Adresse = 138.37.94.248

IP-Adresse = 138.37.88.249
Netz-Name = „Simulator“

Die Benutzung einer TCP / IP – Verbindung („Socket-Schnittstelle“)

Sockets sind prozessspezifisch:

Gleichzeitig kann nur ein Prozess ein Socket benutzen

**Jeder Prozess darf mit mehreren Sockets in
Verbindung stehen**

Die Benutzung einer TCP / IP – Verbindung („Socket-Schnittstelle“)

Welches Datenformat sollte gewählt werden ?

Antwort abhängig von Homogenität der Partner:

Alle Beteiligten sollten das gleiche Verständnis des Datenformats haben !

Bei beliebigen Partnern: ASCII-Zeichenketten

**Für Partner aus der selben Programmierwelt:
Spezifischere Objekte**

→ Java bietet vielfältige Möglichkeiten

**Beim nächsten Mal:
Socketkommunikation in Java
Nebenläufigkeitskonzepte**