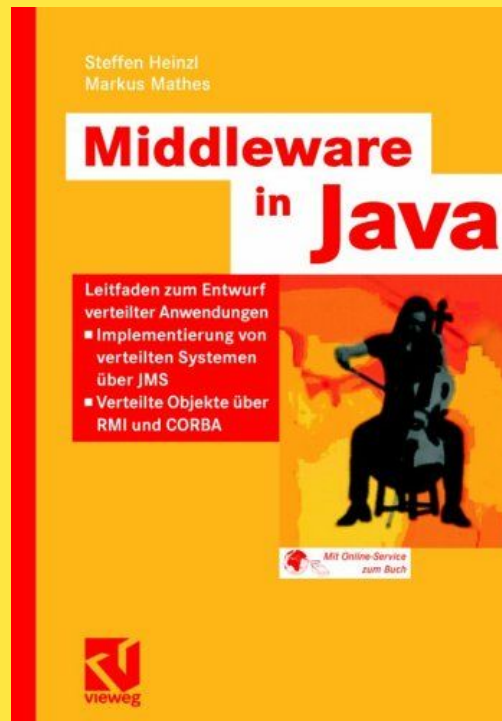


Kapitel 1:

Architektur verteilter Systeme



Middleware in Java
vieweg 2005

© Steffen Heinzl, Markus Mathes

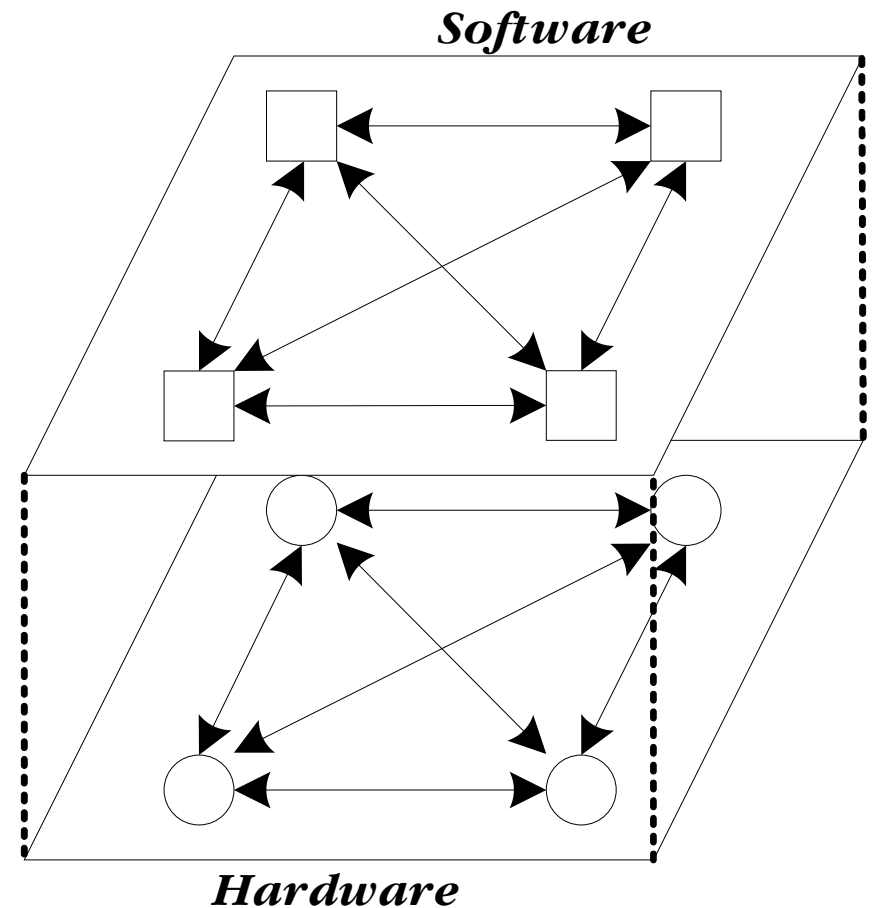
Struktur eines verteilten Systems

Ein verteiltes System besteht aus

- Menge unabhängiger Rechnersysteme
- Kommunikationsnetz

und erbringt eine

- verteilte Anwendung



Teil der verteilten Anwendung



Rechnersystem

Vor- und Nachteile eines verteilten Systems

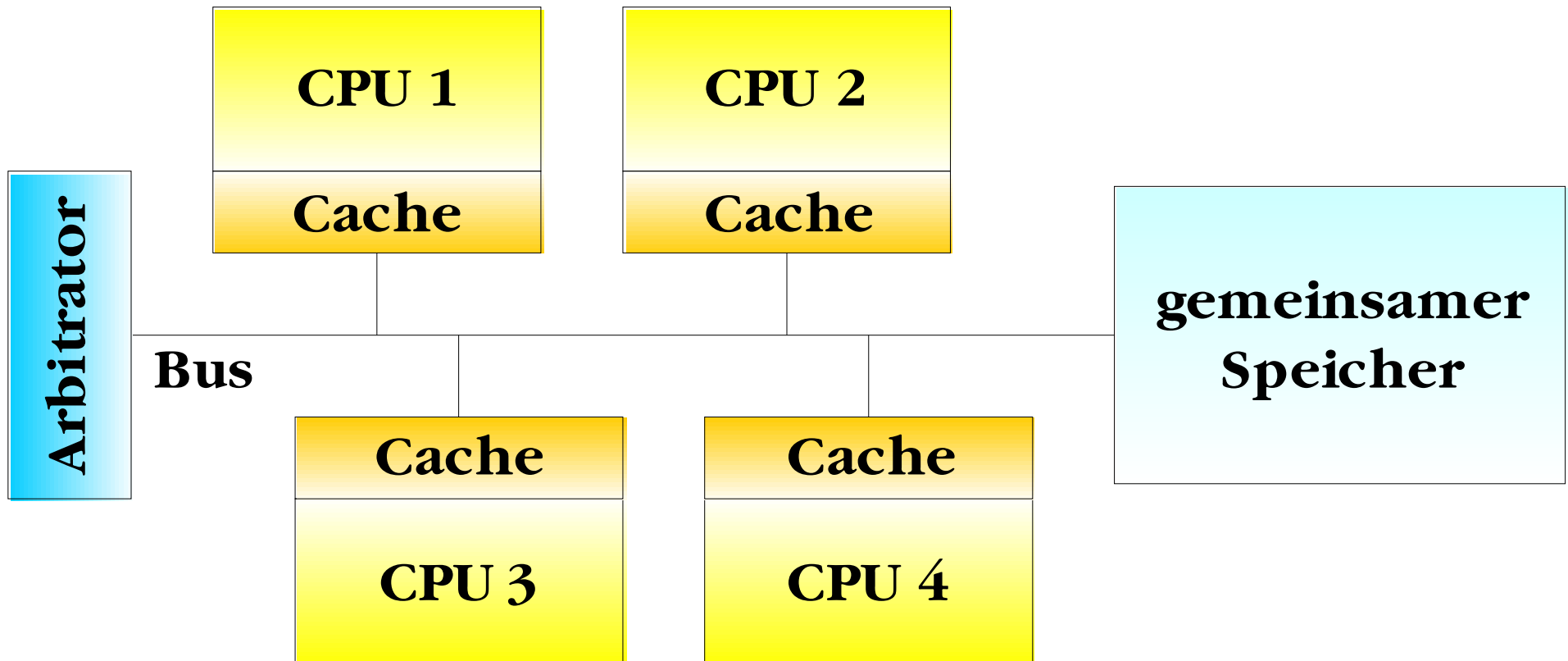
Vorteile:

- gemeinsame Nutzung von Hard- und Software-Ressourcen
- erhöhte Ausfallsicherheit
- schnellerer Zugriff auf Daten
- neue Anwendungen

Nachteile:

- neue Sicherheitsrisiken
- verminderte Ausfallsicherheit
- Konsistenzprobleme
- Performance-Schwankungen
- Wartung und Instandhaltung schwierig

Aufbau eines Multiprozessors (1)



Aufbau eines Multiprozessors (2)

- **CPU**
 - symmetrische Multiprozessorsysteme
 - asymmetrische Multiprozessorsysteme
- **Cache**
 - Zwischenspeicher
 - Kompensation unterschiedlicher Zugriffsgeschwindigkeiten
- **Bus**
 - Verbindungsleitung zwischen einzelnen Komponenten

Aufbau eines Multiprozessors (3)

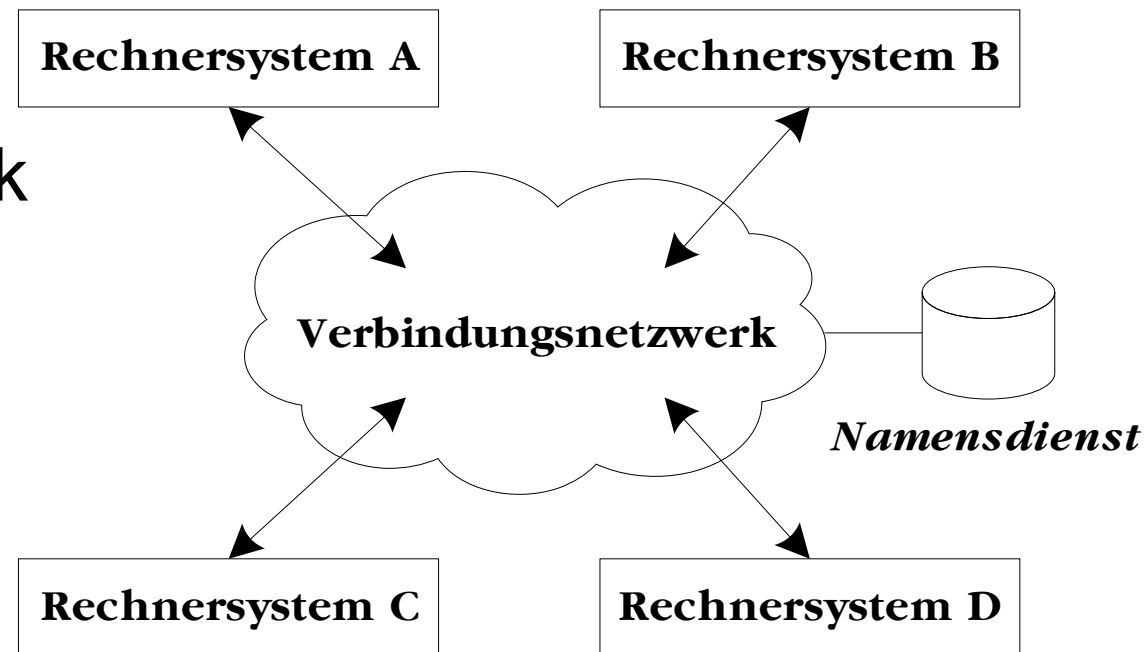
- ***Arbitrator***
 - regelt Zugriff auf gemeinsamen Bus
- ***Speicher***
 - Ablegen von Daten
 - Interprozesskommunikation

Struktur einer Multicomputers

Ein Multicomputersystem besteht aus

- Rechnersystemen
- Verbindungsnetzwerk
- Namensdienst

Die Rechnersysteme besitzen **keinen** gemeinsamen Speicher.



Kommunikation innerhalb eines Multicomputers

- Da kein gemeinsamer Speicher vorhanden ist, kann die Kommunikation nur über ***Austausch von Nachrichten (message passing)*** erfolgen.
- Eine Nachricht besteht aus
 - Senderadresse
 - Empfängeradresse
 - Umschlag
- Ein ***Namensdienst*** realisiert die Abbildung der Sender- und Empfängeradressen auf die physikalischen Adressen der Rechnersysteme (mapping).

Verteilungs- und Programmierparadigmen

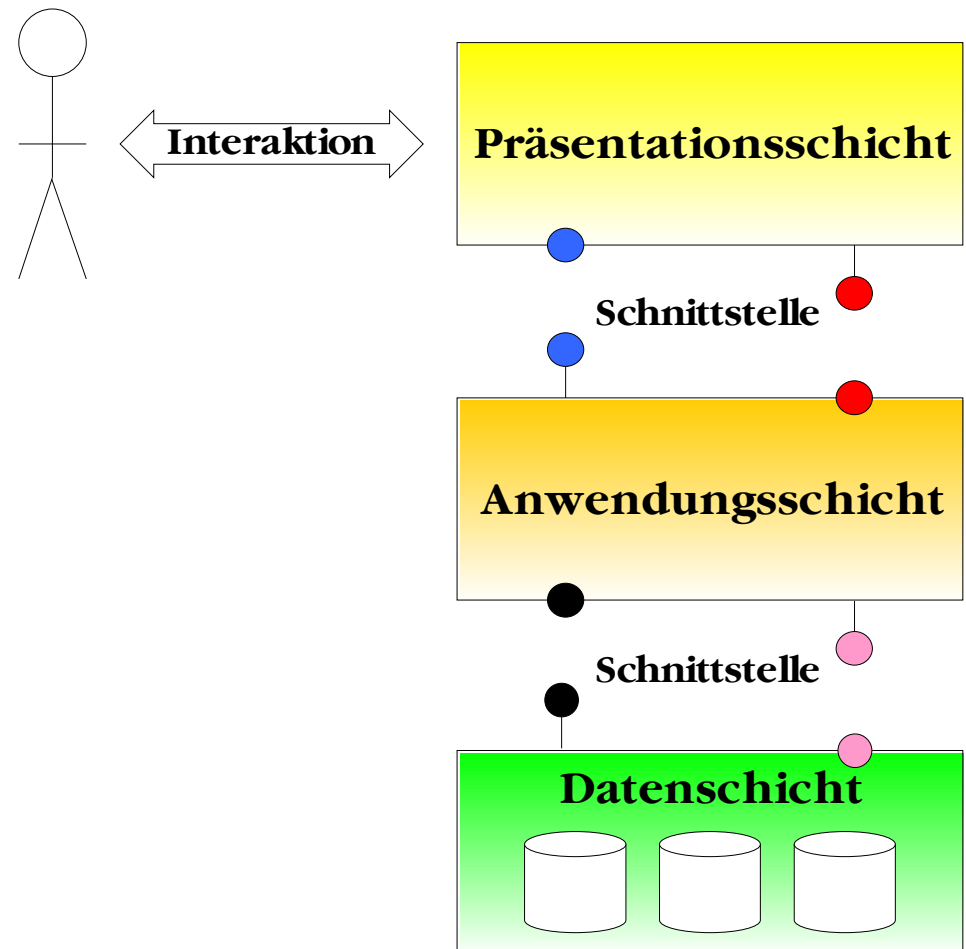
Verteilungsparadigmen: ***Programmierparadigmen:***

- Dateien
 - Dokumente
- Funktionalität
 - Prozeduren
 - Objekte
 - Berechnungen

- Sockets
- Remote Procedure Call (RPC)
- Remote Method Invocation (RMI)
- nachrichtenbasierte Kommunikation

3-tier-architecture

- Zerlegung der Anwendung in drei disjunkte Schichten
 - Präsentationsschicht
 - Anwendungsschicht
 - Datenschicht
- Austausch von Daten zwischen Schichten erfolgt durch definierte Schnittstellen



Transparenz

- Eine wichtige Eigenschaft verteilter Systeme ist die Realisation von verschiedenen ***Transparenzarten***.
- Unter Transparenz versteht man das Verstecken einer bestimmten ***Eigenschaft*** bzw. ***Funktion*** vor dem Benutzer, nicht aber deren ***Effekt***.
- Man unterscheidet verschiedene ***Arten*** von Transparenz.

Transparenzarten (1)

- ***Positionstransparenz***
 - Speicherort einer Ressource muss nicht bekannt sein
- ***Migrationstransparenz***
 - Ressourcen können zwischen Rechnersystemen verschoben werden
- ***Replikationstransparenz***
 - replizierte Ressourcen erscheinen als eine einzige
- ***Skalierungstransparenz***
 - verteiltes System kann einfach wachsen
- ***Fehlertransparenz***
 - Systemfehler werden verborgen

Transparenzarten (2)

- **Relokationstransparenz**
 - Ressource kann während Umzug zwischen Rechnersystemen genutzt werden
- **Parallelitätstransparenz**
 - versteckt gleichzeitigen Zugriff mehrerer Benutzer auf gleiche Ressource
- **Leistungstransparenz**
 - Verteilung von Aufgaben auf verschiedene Rechnersysteme (load balancing)
- **Zugriffstransparenz**
 - Heterogenität von Betriebs- und Rechnersystemen verstecken

Literatur

- Günther Bengel: *Grundkurs Verteilte Systeme* (3. Auflage); vieweg 2004; <http://www.vts.fh-mannheim.de>
- George Coulouris, Jean Dollimore, Tim Kindberg: *Verteilte Systeme – Konzepte und Design* (3., überarbeitete Auflage); Pearson Studium 2002
- Schahram Dustdar, Harald Gall, Manfred Hauswirth: *Software-Architekturen für Verteilte Systeme*; Springer 2003
- Andrew S. Tanenbaum: *Computernetzwerke* (3. überarbeitete Auflage); Pearson Studium 2002; <http://www.cs.vu.nl/~ast>
- Andrew Tanenbaum, Marten van Steen: *Verteilte Systeme – Grundlagen und Paradigmen*; Pearson Studium 2003
- Michael Weber: *Verteilte Systeme*; Spektrum Akademischer Verlag 1998

Aufgaben

In „*Middleware in Java*“ finden Sie

- Wiederholungs-,
- Vertiefungs-,
- Programmieraufgaben zu den vorgestellten Themen.

Zur Festigung und Vertiefung des Erlernten wird eine Bearbeitung der Aufgaben empfohlen.

