

Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik

Middleware für verteilte industrielle Umgebungen

Dr. Matthias Riedl
ifak e.V. Magdeburg, Werner-Heisenberg-Str. 1
39106 Magdeburg
matthias.riedl@ifak.eu
Tel.: 0391 / 9901460



Ziel

- ❑ Konzept und Entwurf verteilter Systeme
 - Erlangung allgemeiner Grundlagenkenntnisse
- ❑ Eigenschaften verteilter Betriebssysteme
 - Unterschiede zum normalen Betriebssystem
- ❑ Aspekte verteilter Echtzeitsysteme erkunden
 - kurzer Einblick in Automatisierungstechnik (AT)
 - Middleware für AT



Vorlesungsinhalt

- ❑ Einführung
- ❑ Begriffe der AT
- ❑ Einführung Steuerungskonzepte
- ❑ Eigenschaften / Entwurfsziele verteilter Systeme
- ❑ RPC inkl. IPC
- ❑ (D)COM, .Net
- ❑ OPC, OPC UA
- ❑ FDT
- ❑ DPWS
- ❑ DOME



Struktur

- Vorlesung
 - 2 SWS
 - Overhead
 - Anregungen durch Beispielaufgaben
- Selbststudium
 - Windows-OS
 - C++
 - Aufbereitung des Vorlesungsstoffes
 - Besprechung der Beispielaufgaben



Weitere Informationsquellen

□ Weiterführende Literatur

- Tanenbaum, Stehen: Verteilte Systeme, Pearson Studium, ISBN 3.8273-7057-4
- Coulouris, Dollimore, Kindberg: Verteilte Systeme – Konzepte und Design, Pearson Studium, ISBN 3-8273-7022-1
- Hofmann, J.; Jobst, F.; Schabenberger, R.: Programmieren mit COM und CORBA, Einführung in die Architekturen für verteilte Anwendungen, Carl Hanser Verlag, München, Wien, ISBN 3-446-21479-8
- B. Stroustrup. The C++ Programming Language, Addison-Wesley, 1997, ISBN 0-201-53992-6

□ Fachzeitschrift(en):

- iX, Magazin für professionelle Informationstechnik, Heise Zeitschriften Verlag GmbH & Co. KG, Hannover, <http://www.heise.de/ix/>
- ct, Magazin für Computer Technik, Heise Zeitschriften Verlag GmbH & Co. KG, Hannover, <http://www.heise.de/ct/>



Quellen

- ❑ Literatur
- ❑ Vorlesung Prof. Schröder-Preikschat, Uni Magdeburg
- ❑ Vorlesung Prof. Simon, FH Harz
- ❑ Vorlesung Prof. Schill, TU Dresden
- ❑ Vorlesung Prof. Diedrich, Uni Magdeburg



Motivation

□ Arbeitsplatzrechner:

- Multitasking
- Vernetzung
- Direktmanipulation, graphische Schnittstelle
- Hohe Leistung (CPU, Übertragung)
- Großer Haupt- und Hintergrundspeicher

□ Einsatzbereiche:

- Management / Entwicklung
- Teamarbeit (CSCW - Computer Supported Collaborative Work)
- Gruppenkommunikation
- Prozesssteuerung



Verteiltes Unix

- ❑ Unix, das z.Z. verbreitetste Mehrbenutzerbetriebssystem, . . .
 - massiv bedrängt durch Windows NT, 2000, XP, Vista, ...
- ❑ . . . lieferte die Basis zur Verwaltung vieler vernetzter Rechner
 - Grundkonzepte wurden übernommen und erweitert
 - Leistung
 - Zuverlässigkeit
 - Erweiterbarkeit
 - Kommunikation
 - Vorreiter waren die Versionen von 4BSD
 - Berkeley Software Distribution der späten 70 er Jahre
- ❑ **Netzwerkdateisystem** und **Fernaufrufmechanismus**
 - zentrale Komponenten heutiger Unix Implementierungen



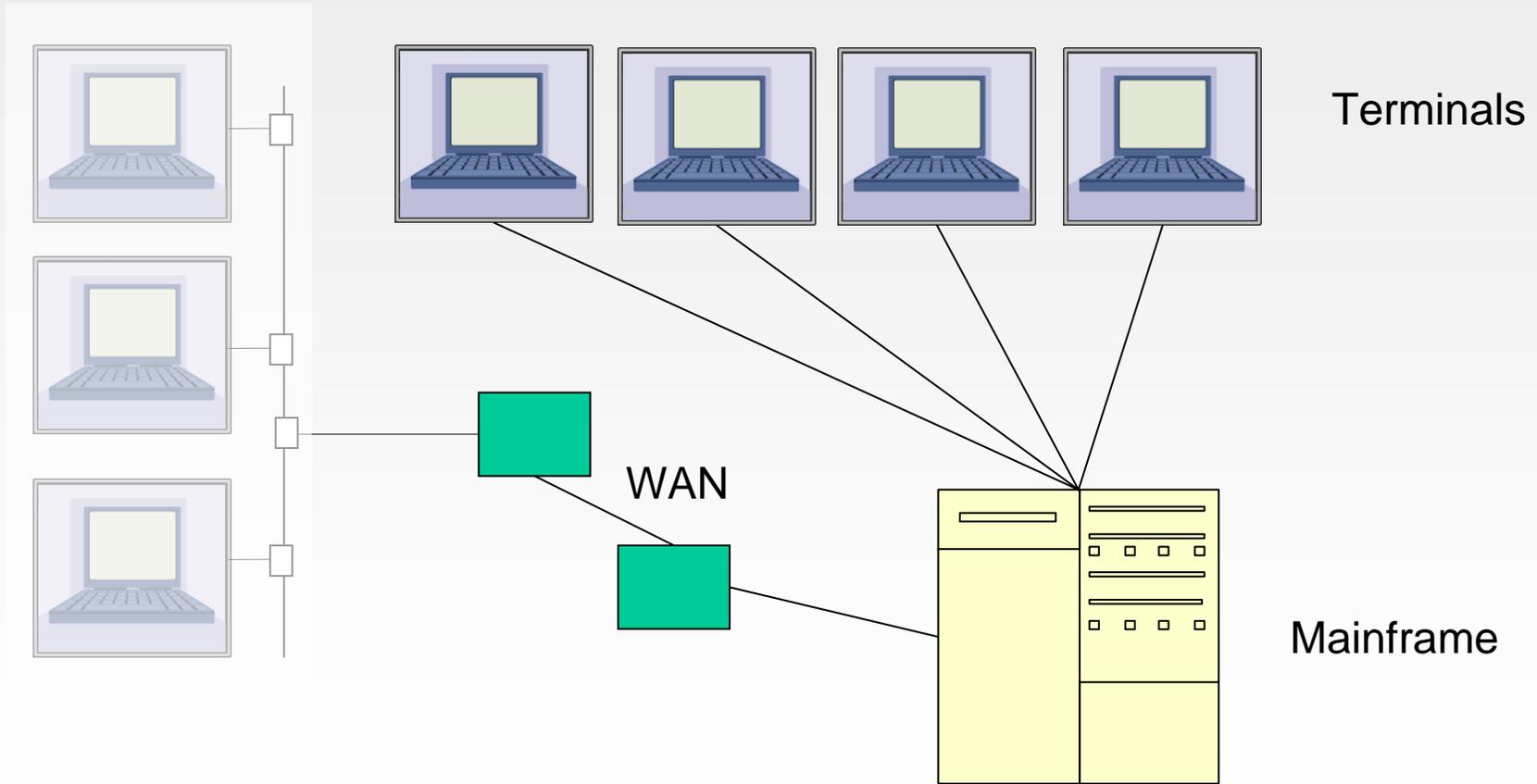
Dezentrale Struktur

- ❑ kein einzelner, zentraler, gemeinsam genutzter Rechner . . .
 - Großrechner ([main frame](#))
- ❑ . . . sondern mehrere "kleine", individuell genutzte Rechner
 - Arbeitsplatzrechner ([workstation](#)), PC
- ❑ [Kunden](#) teilen sich mehrere "Bediener", d.h. [Server](#)
 - Client/Server-Systeme
- ❑ "Client" steht allgemein für Benutzerprogramme
 - Editor, Übersetzer, . . .
- ❑ "Server" steht für gemeinsam nutzbare Betriebsfunktionen
 - Platten
 - Drucker
 - Netzwerkübergänge
 - Authentifikation
- ❑ arbeitsteilige Systemorganisation



Beispieltopologie

Zentralisiertes System



Verteilung

□ Verteiltes System

- Softwarekomponenten auf vernetzten Computern
 - Austausch von Nachrichten
 - kommunizieren, Aktionen koordinieren
- z. B. Internet
 - IP (Schicht 3)
 - TCP / UDP (Schicht 4)

□ Verteilte Anwendung

- Anwendungsproblemlösung durch Nutzung des verteilten Systems
- bestehend aus verschiedenen Komponenten
 - kommunizierend mit
 - Komponenten des verteilten Systems
 - Anwendern
- z. B.
 - FTP, Telnet, ...
 - Internetzeitung, -shops, ...
 - Automatisierungssoftware
 - SMS-Dienste (Handy)
 - Grid



Verteilung (2)

- ❑ Physikalische Rechnerknoten (Prozessor + Speicher)
- ❑ Rechnerkopplung direkt / indirekt
 - Lokale Netze (Ethernet (CSMA/CD), Token Ring, Token Bus)
 - Hochleistungsnetze (Gigabit Ethernet, ATM)
 - Gateways / Bridges
 - Funknetze (GSM, UMTS)
- ❑ Transportorientierte Kommunikationsprotokolle (TCP/IP, UDP/IP, ...)
- ❑ Kommunizierende Betriebssystemprozesse
 - Volle logische Vermaschung
 - Keine volle physische Vermaschung (Kommunikation über Zwischenkomponenten)
- ❑ Systemorientierte Betriebsmittel (Dateisystem, Threads, Systemprogramme)
- ❑ Verteilter Speicher, dezentral, kooperativ
- ❑ Hierauf aufbauend: Verteilte Anwendung (bereichsspezifisch)



Zielsetzung

- ❑ Verteilung von Daten / Funktionen und Last
- ❑ Dezentralisierung und Kooperation
- ❑ Lokalitätseigenschaften und Effizienz
- ❑ Integration von Teilanwendungen
- ❑ Entfernter Betriebsmittelzugang
- ❑ Fehlertoleranz: Ausfallsicherheit und Verfügbarkeit



Definition Verteilter Systeme

- Lt. Duden Informatik, Dudenverlag, 1993

Ein System heißt verteilt, wenn sich seine Komponenten an **räumlich getrennten Stellen** befinden oder befinden könnten, hierdurch aber die **Funktionalität** des Gesamtsystems **nicht beeinträchtigt** wird. Fast alle in der menschlichen Gesellschaft eingeführten Systeme sind verteilt (z.B. Behörden, Vertriebsorganisationen). Es müssen **Informationen** zwischen den verteilt liegenden Institutionen **ausgetauscht** werden, die einzelnen Institutionen führen aber ihre Aufgabe in eigener Verantwortung unabhängig voneinander aus und **sprechen sich in Einzelfällen direkt untereinander ab**.



Definition Verteilter Anwendung

□ Lt. Uwe M. Berghoff, Johann Schlichter

Eine Anwendung A heißt verteilt, wenn deren Funktionalität in eine Menge von kooperierenden Teilkomponenten A_1, \dots, A_n , $n > 1$ zerlegt ist; Jede Teilkomponente hat ihren eigenen internen Zustand. Die Teilkomponenten A_i sind autonome Verarbeitungseinheiten, die auf verschiedene Funktionseinheiten F_i abgebildet werden. Die Teilkomponenten A_i tauschen untereinander Informationen mittels des Netzes aus.



Kommerzielle Anwendungen

- ❑ Datenverarbeitungs- und Informationssysteme
 - Reiseunternehmen
 - Reservierungssysteme (Flüge, Mietautos, . . .)
 - Banken
 - Zweigstellen- und Geldautomatenverwaltung
 - Versicherungen
 - Schadensmeldungen und -regulierungen
 - Supermärkte bzw. Kaufhausketten
 - Kassenüberwachung, Lagerverwaltung
- ❑ nahezu das gesamte **Dienstleistungsgewerbe**



Kommerzielle Anwendungen (ff)

- typische Anforderungen:
 - sehr hohe Verfügbarkeit
 - Sicherheit
 - Datenschutz
 - sehr gute Skalierbarkeit
- typische Randbedingungen:
 - nebenläufiger/paralleler Datenbankzugriff
 - garantierte Antwortzeiten
 - geographisch weit verstreute Datenstationen
 - heterogene Hardware- und Software-Systeme
- typische Erscheinungsformen:
 - dedizierte Hardware, Software, Kommunikationseinrichtungen
 - zentralisierte Datenbank
 - mobile Datenbankanfragen



Fernnetze

- ❑ relativ langsame Verbindungen, extrem hohe Skalierung . . .
 - Millionen von Rechnern "hängen" im Internet
- ❑ . . . dennoch Raum für verteilte Anwendungen:
 - elektronische Post ([email](#))
 - Namens- und Adreßauflösung, Wegewahl, Zustellung
 - extrem hohe Dynamik
 - Namen/Adressen hinzufügen, ändern und löschen
 - unterbrechungsfreier Betrieb
- ❑ Netznachrichten ([netnews](#))
 - Millionen weltweit verstreute Benutzer
 - Tausende von News-Gruppen verwalten
- ❑ "Seitenblättern" ([information browsing](#))
 - Ein "Webster elektronischer Dokumente"
 - Text, (Fest-/Bewegt-) Bild, Ton
 - Gopher und nicht zuletzt [WWW](#)
- ❑ die physikalische Verteilung ist (noch) nicht transparent
 - d.h. die Verteilung von Hardware und Information



Multimedia

- ❑ digitale Darstellungen von:
 - (photographischen) Bildern
 - Audio-Sequenzen
 - Video-Sequenzen
- ❑ Anwendungen im Geschäfts- und Privatbereich:
 - rechnergestütztes Lernen (tele learning)
 - Tele-Konferenzen (tele conferencing)
 - kooperativer Entwurf
 - ...
 - Shopping
 - Spiele
- ❑ aufwendige Datenspeicherung, -übertragung und -darstellung:
 - aufgezeichnete Audio-/Video-Sequenzen
 - video on demand
 - per Video-Kamera on-line gelieferte Bilder
 - rechner-synthetisierte Animationen
 - virtual reality
 - von Simulationen



Multimedia (ff)

- entwurfsbeeinflussende, spezielle Anforderungen:
 - Audio-/Video-Daten sind **kontinuierliche Daten**
 - zeitbasierte Daten (time-based data)
 - originalgetreue, "verständliche" Reproduktion von Ton
 - Synchronisation von Bild und Ton
 - Reproduktion mit einer Rate von 16 Rahmen pro Sekunde
 - ca. 512 x 512 16-Bit-Pixel pro Rahmen
 - Datenrate von ca. 100 MBits/sek
 - komprimiert ca. 1 MBits/sek
 - Interaktion erfordert **kleine Antwortzeiten**
 - kleiner 100ms zwischen Produktion und Reproduktion
 - "verzögerungsfreie" Übertragung
- **Echtzeitfähigkeit** ist maßgeblich für die Dienstgüte
 - Kommunikationsprotokolle und Betriebssysteme



Verteilte Echtzeitsysteme

- bestimmen mehr und mehr das menschliche Umfeld:
 - Fabrikautomatisierung
 - Stahlgießerei, Walzwerk, Gerätebau, Energieversorgung, . . .
 - Telefonvermittlung
 - hunderte vernetzter Spezialrechner . . .
 - . . . schalten einige hunderttausend Anrufe pro Stunde
 - Automotive Systeme
 - für Wasser-, Luft-, Straßen- und Schienenverkehr
 - "verteilte Systeme mit Fügeln" (fly by wire)
 - "verteilte Systeme auf Rädern" (drive by wire)
 - "intelligente" Produkte
 - autonome Geräte mit speziellen Aufgaben
 - mechanisches Teilsystem
 - Sensoren, Aktuatoren
 - Steuereinheit
 - Benutzerschnittstelle
 - zusammengesetzt aus dedizierten VLSI-Chips
 - Funktionalität festgelegt durch Software

□ übernehmen mehr und mehr **lebenswichtige Aufgaben**



Wichtige Eigenschaften

- ❑ **Ortstransparenz** – der Ort, an dem sich ein Objekt befindet, ist für den Benutzer transparent; zwischen lokalen und im Netz verteilten Objekten wird nicht unterschieden.
- ❑ **Zugriffstransparenz** – auf alle Objekte wird in gleicher Weise zugegriffen
- ❑ **Nebenläufigkeitstransparenz** – mehrere Anwender oder Anwendungsprogramme greifen gleichzeitig auf gemeinsame Objekte (z.B. Daten) zu



Software-Strukturen

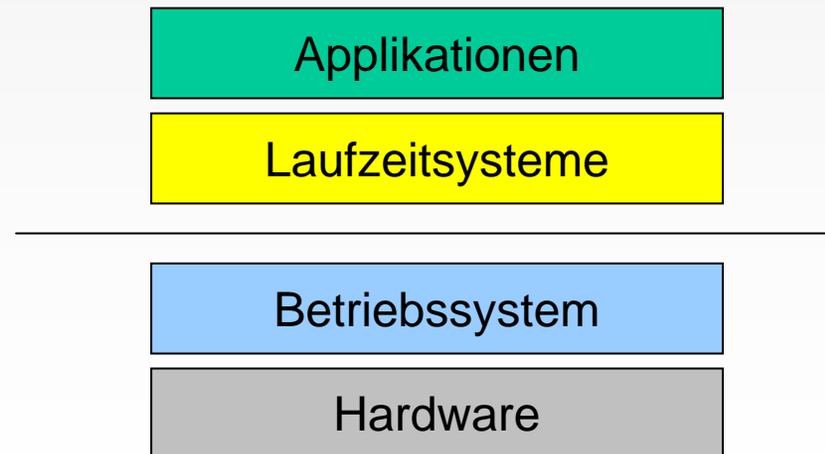
- ❑ Betriebssysteme zentralisierter Systeme sind (meist) monolithisch
 - eine feste Menge von Systemabstraktionen . . .
 - . . . angeboten über eine **unveränderliche Schnittstelle**
 - . . . eingebunden in einen **gemeinsamen Adreßraum**
- ❑ offene Systeme erfordern **offene Schnittstellen**
 - eine feste Menge weniger **Basisabstraktionen** . . .
 - . . . ergänzt durch **problembezogene Erweiterungen**
- ❑ Betriebssysteme dezentralisierter System sind nicht monolithisch
 - sie sind **modular** aufgebaut
 - sie sind in ihrer Funktion **erweiterbar**



Software-Strukturen

Zentralisierte Systeme

- das **Betriebssystem** ist die "Haupt-Software-Schicht" . . .
 - sorgt für die grundlegende **Betriebsmittelverwaltung**
 - stellt **Benutzer-/Anwendungsdienste** zur Verfügung
- . . . ergänzt durch die **Laufzeitsysteme**
 - Funktionen zur Unterstützung von Programmiersprachen



Software-Strukturen

Dezentralisierte Systeme

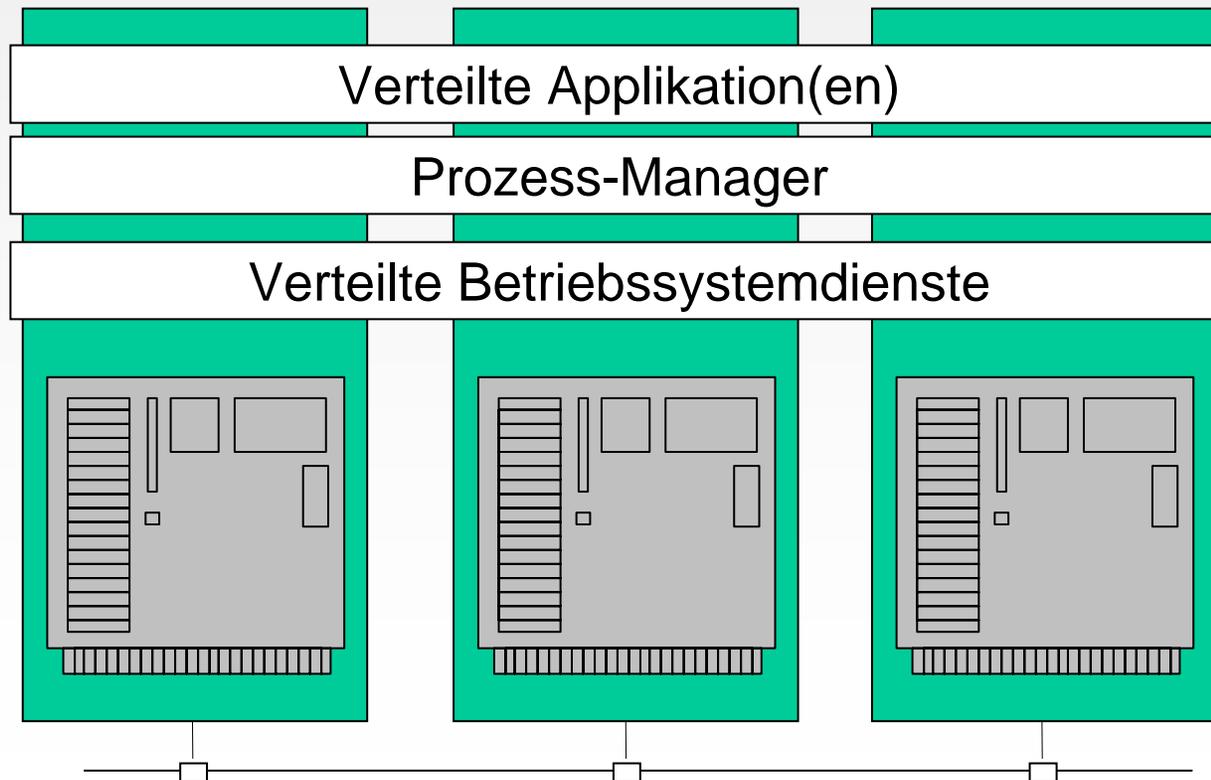
- ❑ das Betriebssystem teilt sich grob in zwei Bereiche auf:
 1. grundlegende Funktionen im **Betriebssystemkern**
 - Prozeß-/Speicherverwaltung und Schutz
 - Interprozeßkommunikation, Gerätebehandlung
 2. problem- und anwendungsorientierte **offene Dienste**
- ❑ verteilte Laufzeitsysteme bieten Programmierunterstützung



Software Strukturen

Verteiltes Betriebssystem

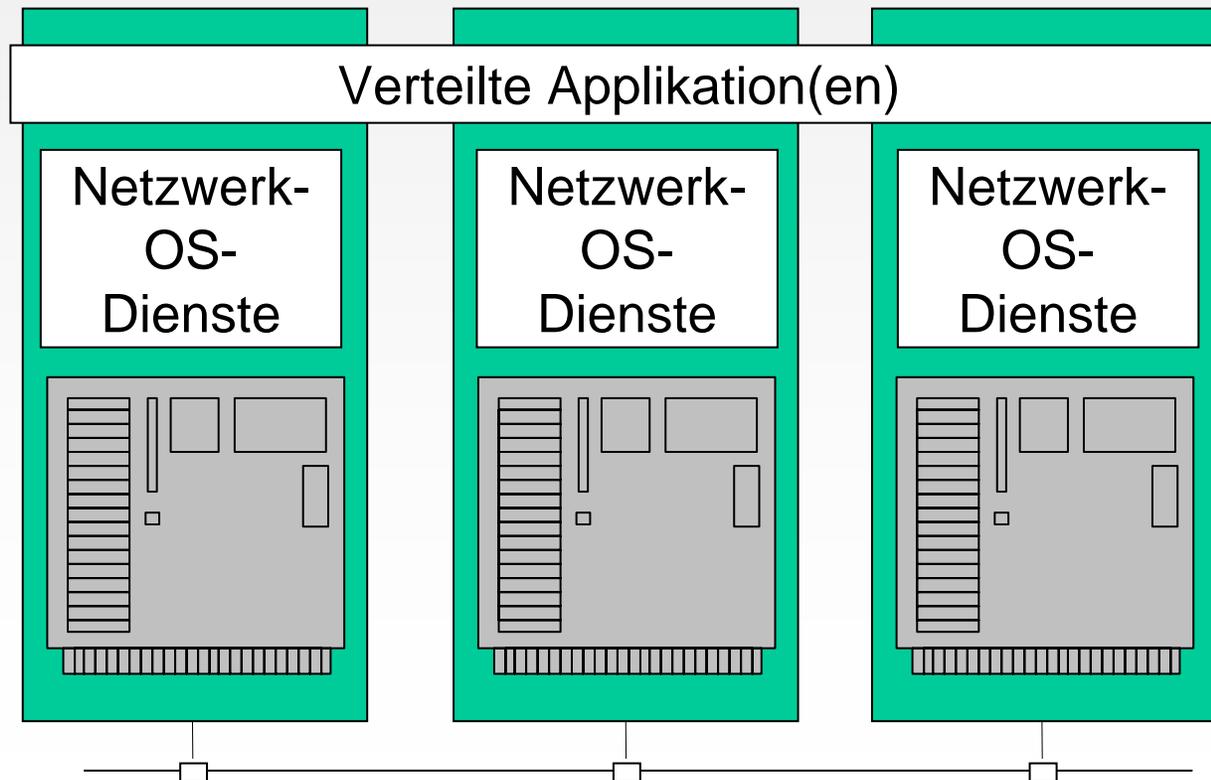
- ❑ Einziges Systemabbild
- ❑ Kontrolle über alle Knoten



Software Strukturen

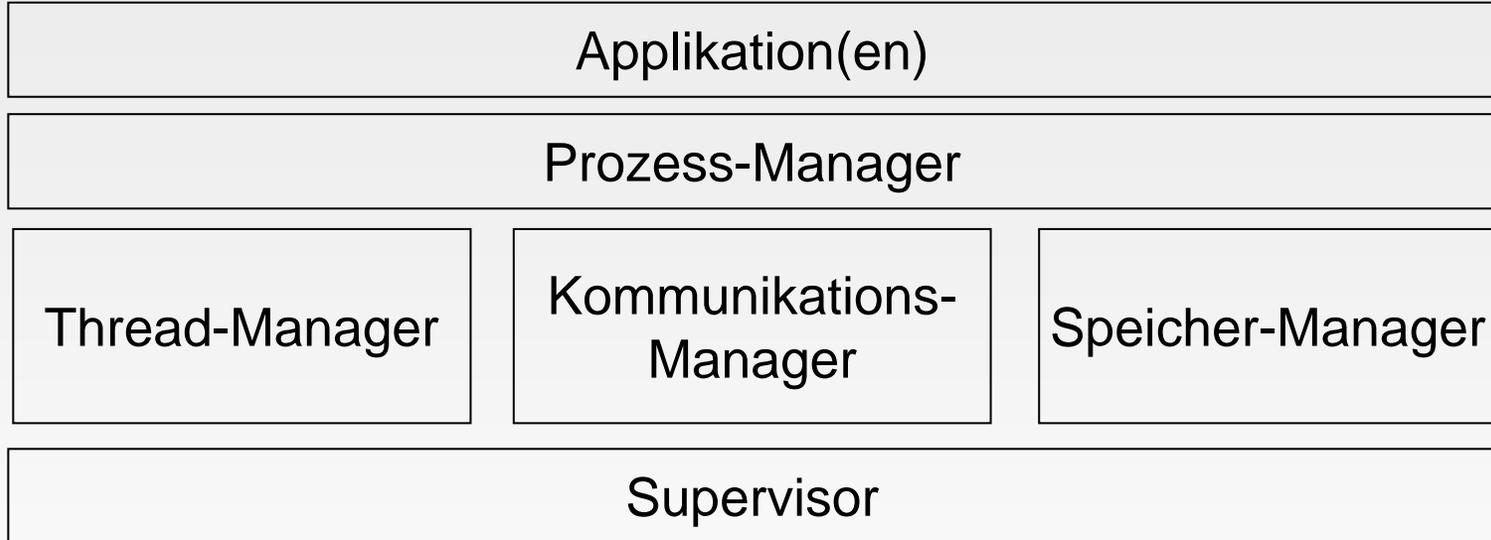
Netzwerkbetriebssystem

- ❑ verfügt über eine eingebaute Netzwerkfunktionalität
- ❑ Zugriff auf entfernte Ressourcen



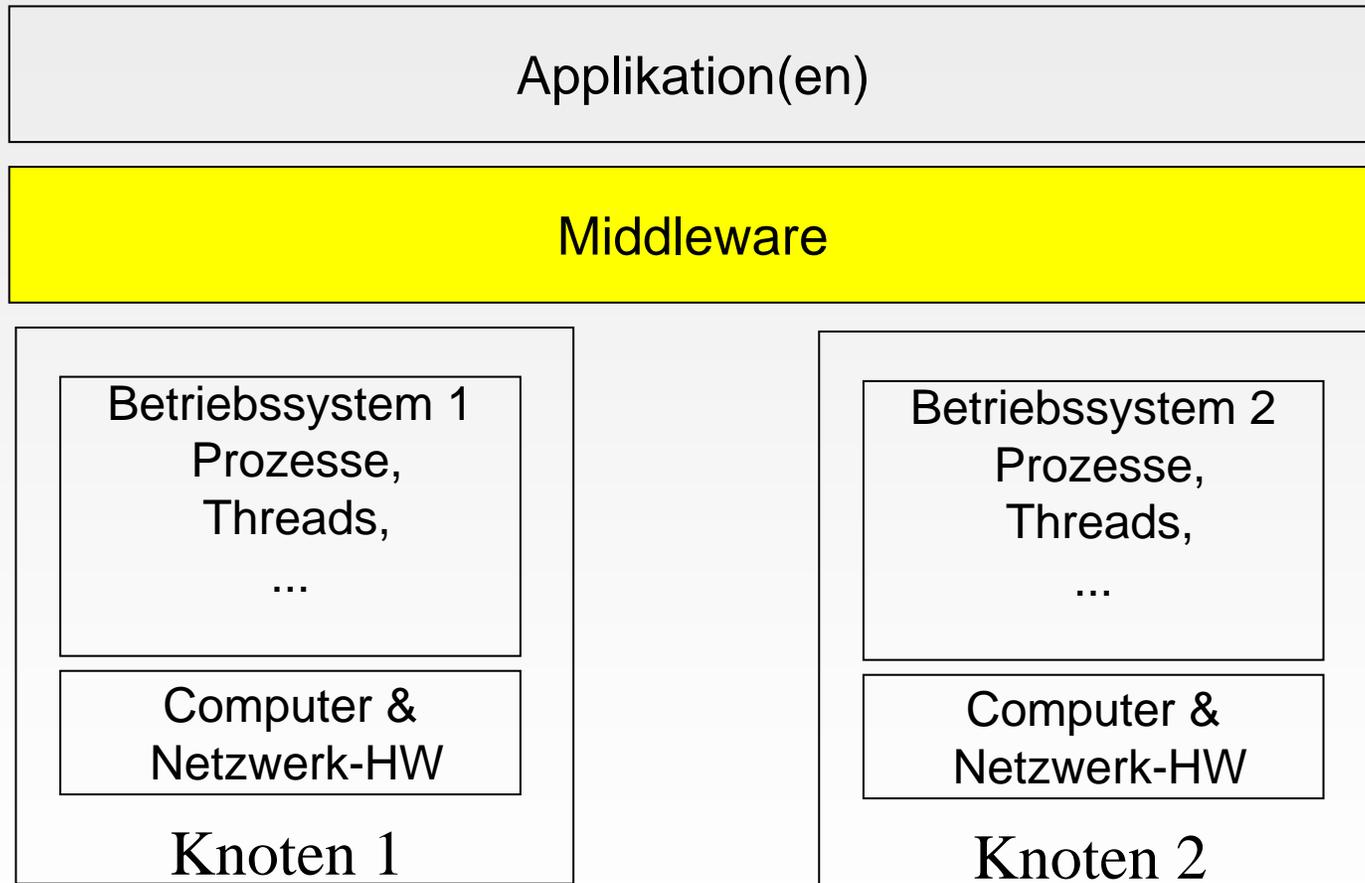
Software Strukturen

Netzwerkbetriebssystem

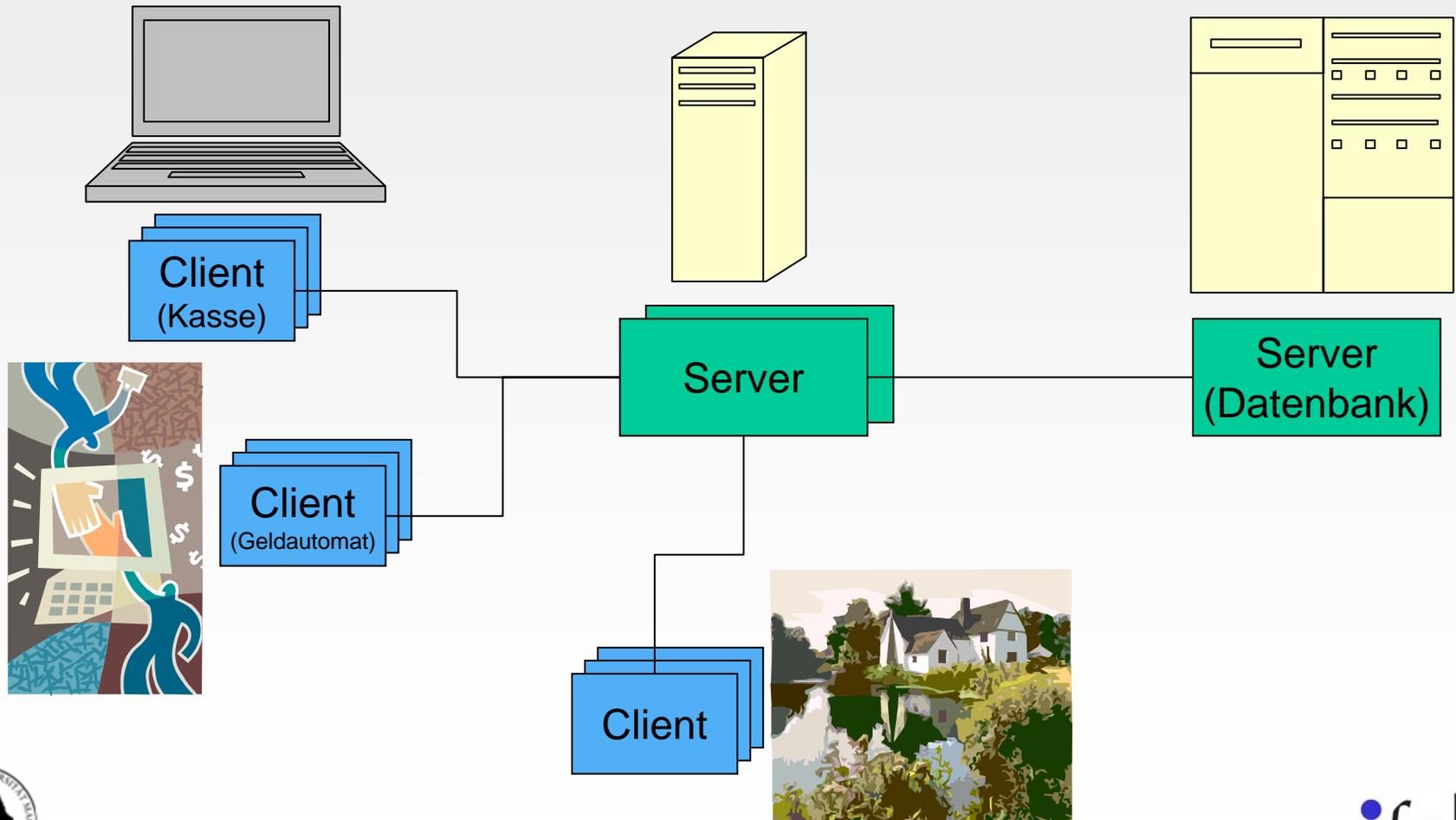


Software Strukturen

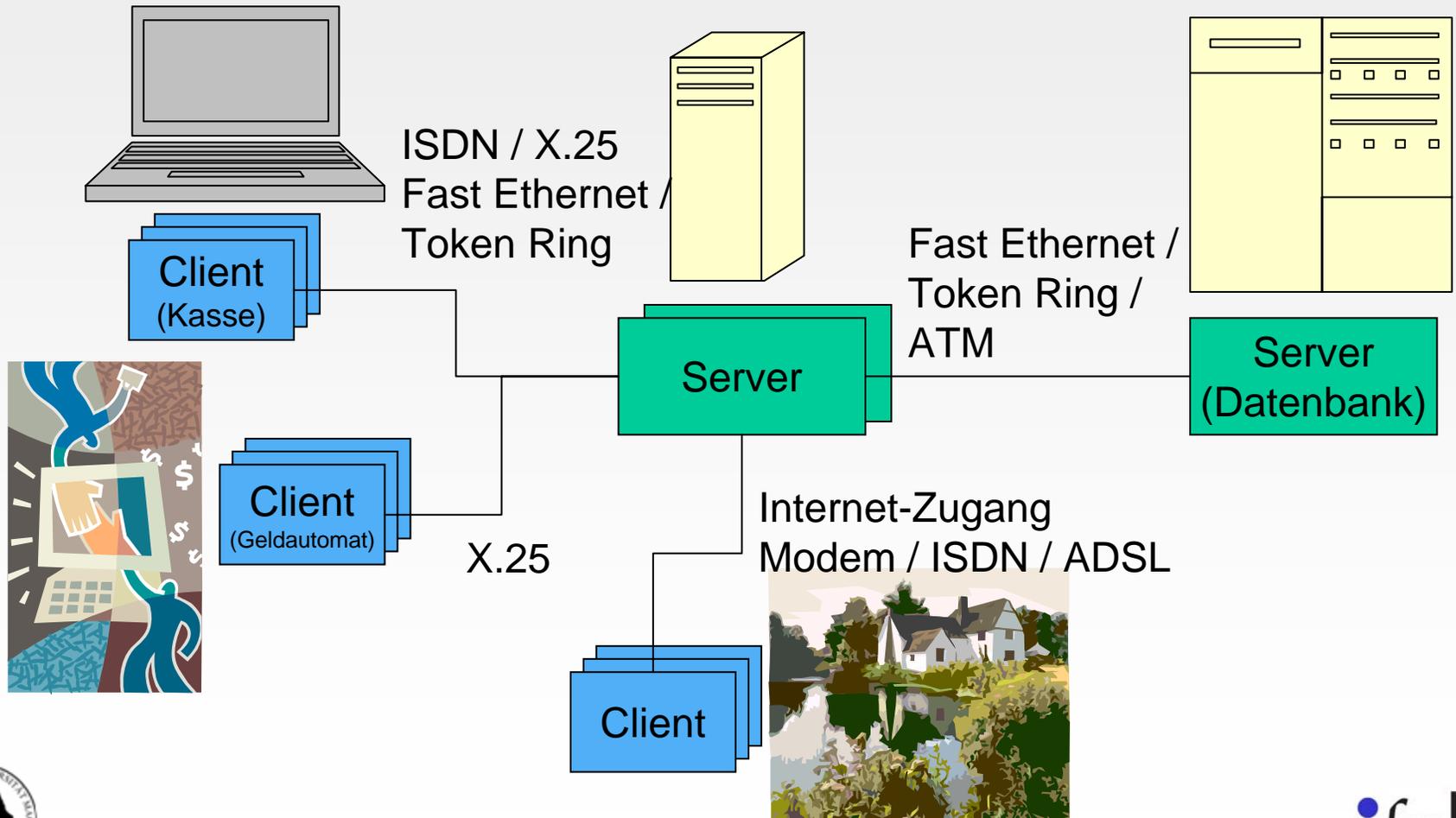
Middleware basiertes verteiltes System



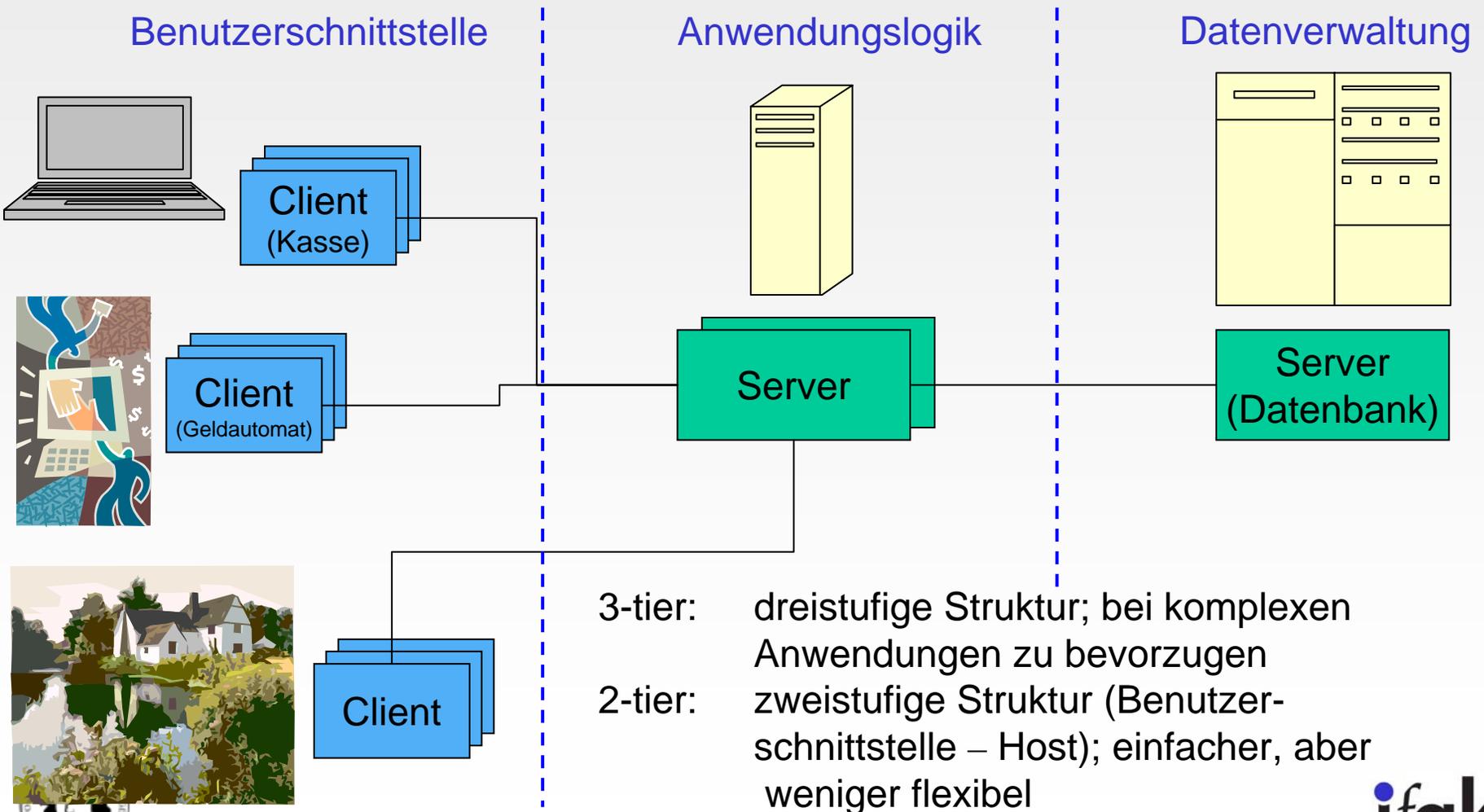
Anwendungsbeispiel



Anwendungsbeispiel - Netzinfrastruktur



Anwendungsbeispiel – Mehrstufige Architekturen

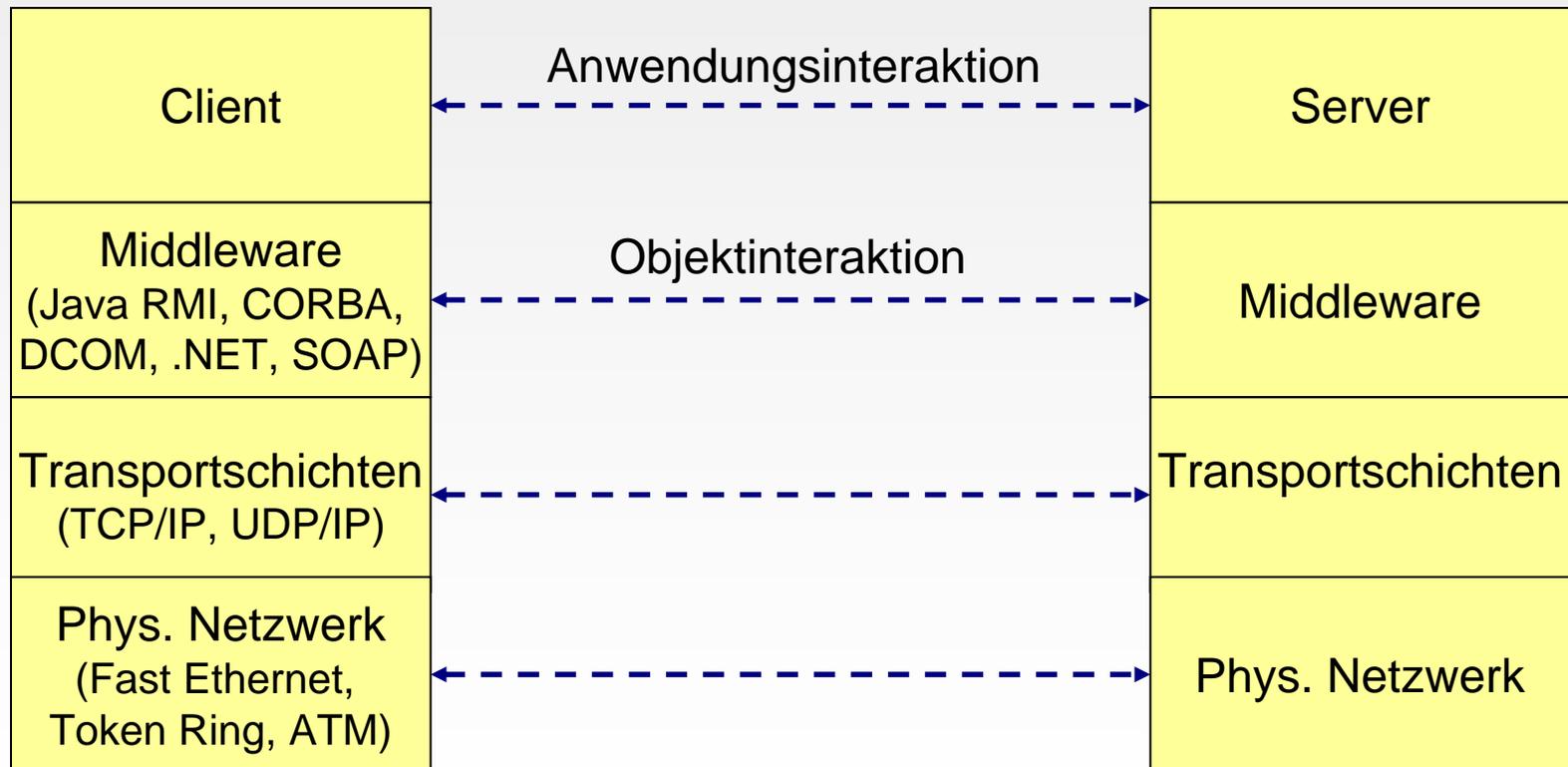


3-tier: dreistufige Struktur; bei komplexen Anwendungen zu bevorzugen

2-tier: zweistufige Struktur (Benutzerschnittstelle – Host); einfacher, aber weniger flexibel



Einordnung: Middleware bei Client /Server

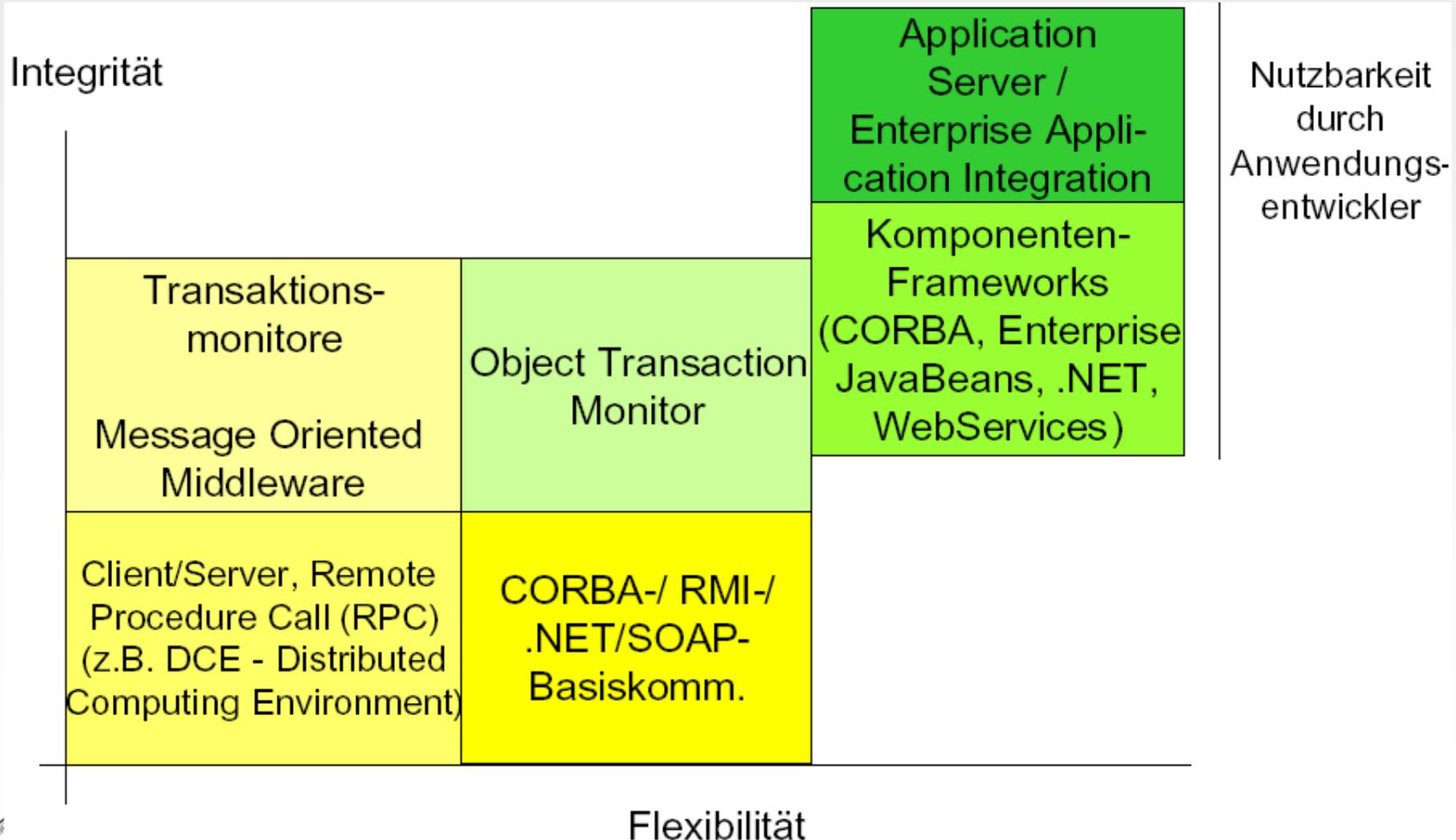


Bekannte Middleware Basistechnologien

- ❑ Java
 - Programmiersprache, Applets
 - Remote Method Invocation (RMI)
 - Enterprise JavaBeans (EJB): Komponenten
- ❑ CORBA
 - Standard der Object Management Group (OMG)
 - Objektorientiert, sprachunabhängig; relativ low-level
- ❑ DCOM / .Net
 - Objektorientiert, COM+ relativ proprietär, .NET offener
 - Entwicklung von Microsoft
- ❑ Weitere
 - MOM (Message Oriented Middleware)
 - SOAP (Simple Object Access Protocol), Web Services
 - Transaktionsmonitore, Application Server

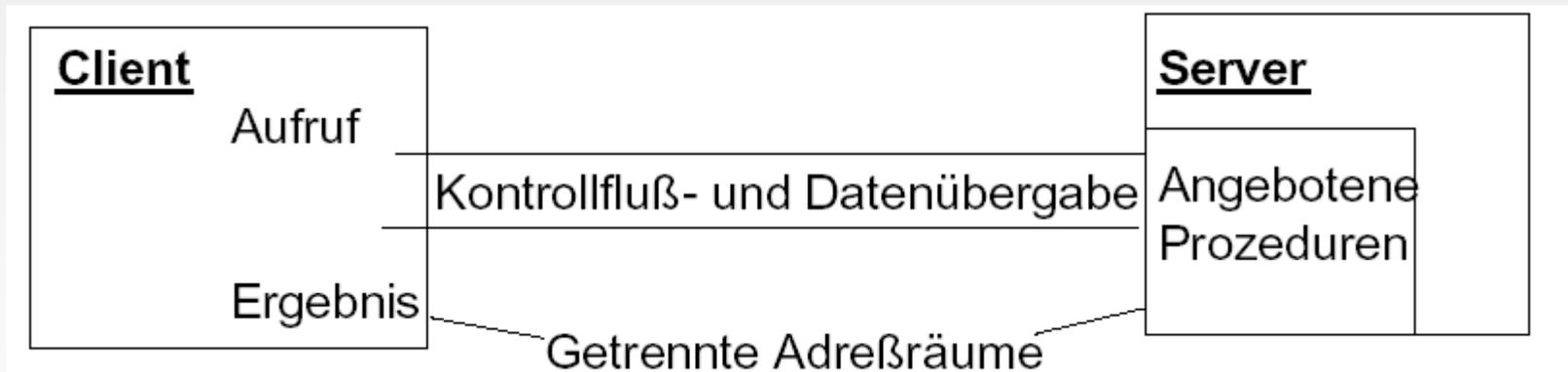


Einordnung Middleware



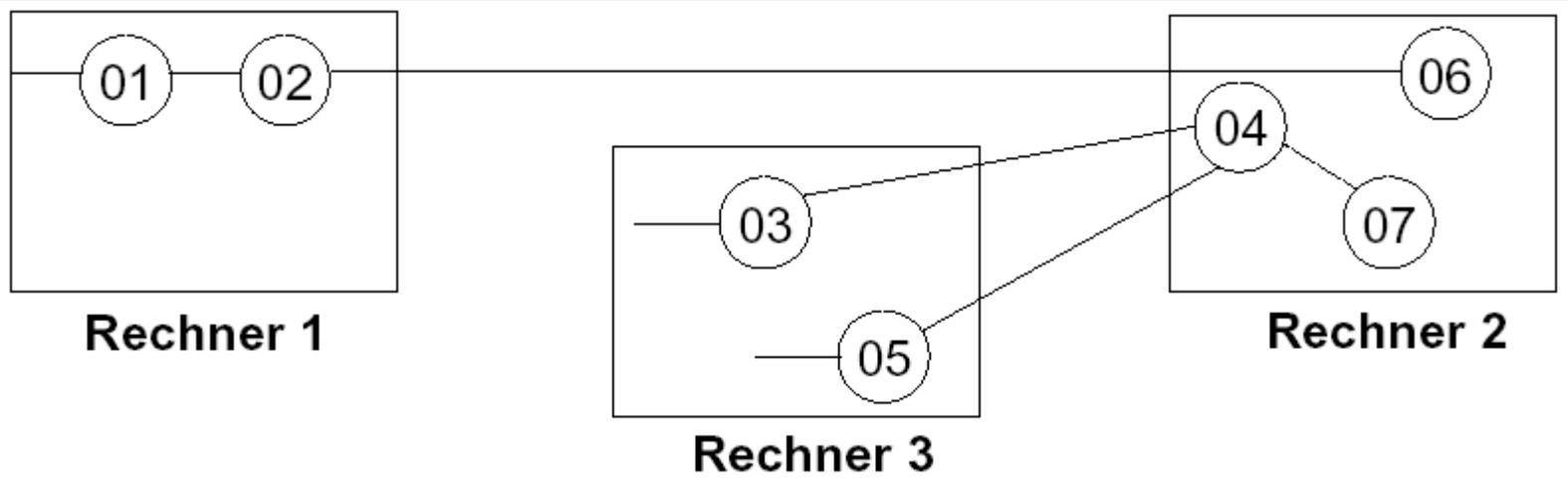
Systemmodelle

Remote Procedure Call



Systemmodelle

Objektorientierte Kommunikation



Vergleich der Systemmodelle

	Client/Server-Modell	Verteiltes objekt-orientiertes Modell
Verarbeitungsmodell	Prozedurales Modell mit Einschränkungen	Objektkommunikation
Zugriffsweise auf Daten	Datenzugriff indirekt über RPC-Server	Direktzugriff auf Objekte
Datenübergabe	Wertparameter-Semantik	Referenzparameter-Semantik
Identität	Nicht systemweit eindeutig	Systemweit eindeutig
Granularität	Server grober Granularität	Objekte beliebiger Granularität
Plazierung	Feste Plazierung	Modifizierbare Plazierung

⇒ Höherer Transparenzgrad und verbesserte Einflußnahme auf Verteilung bei objektorientiertem Modell

