

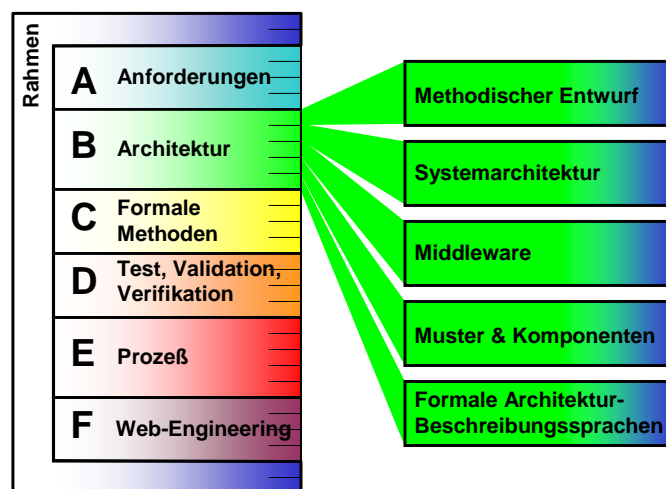
Vorlesung „Methoden des Software Engineering“

Block B „Software Architektur“ **Systemarchitektur**

Dr. Harald Störrle

Einheit B.2, 16.11.2004

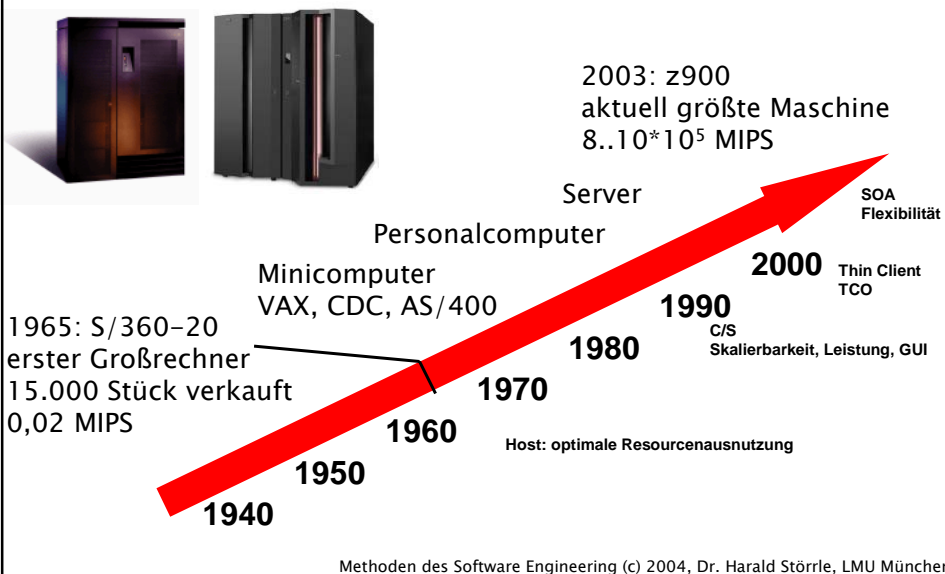
Gliederung Block B



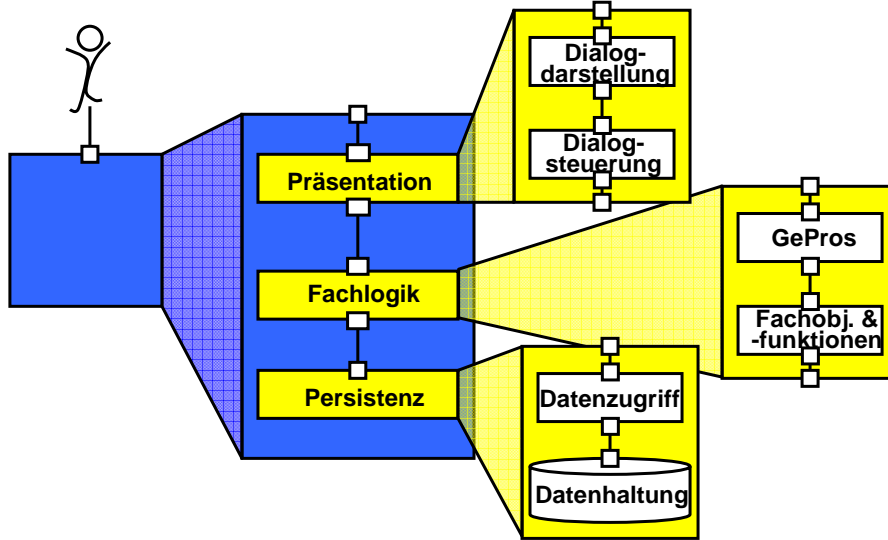
Gliederung / Themen heute

- **Systemarchitektur**
 - Was ist das?
 - Was hat das mit *Softwarearchitektur* zu tun?
 - Welche Arten von Systemarchitektur gab es historisch?
 - Wieso hat sich das geändert?
 - Wo geht die Reise hin?
- **Schichtenarchitektur**
 - Schichtung als Konstruktionsprinzip

Epochen der Systemarchitektur

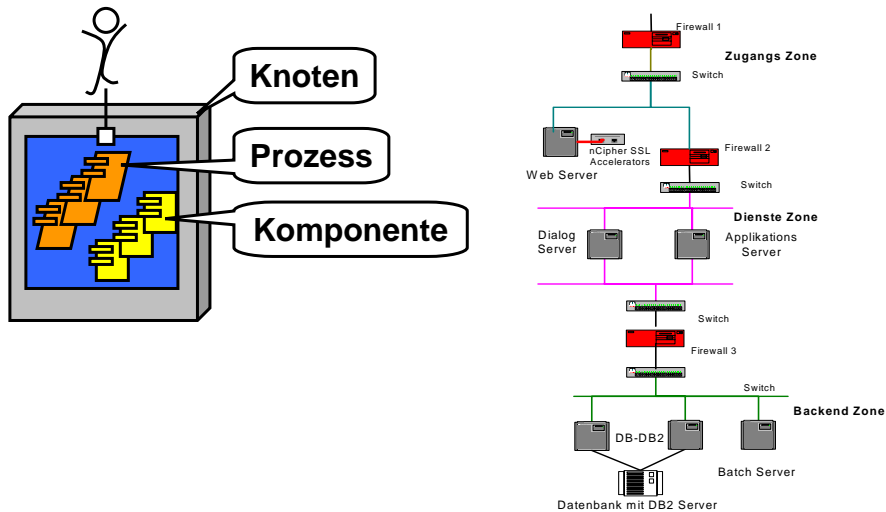


Softwaretechnische Gliederung Schichten („Layer“)



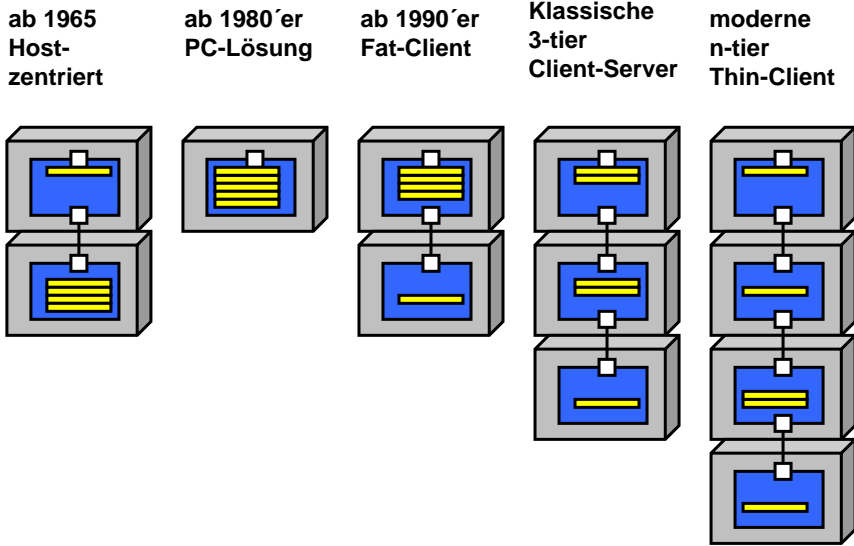
Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Notationen



Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

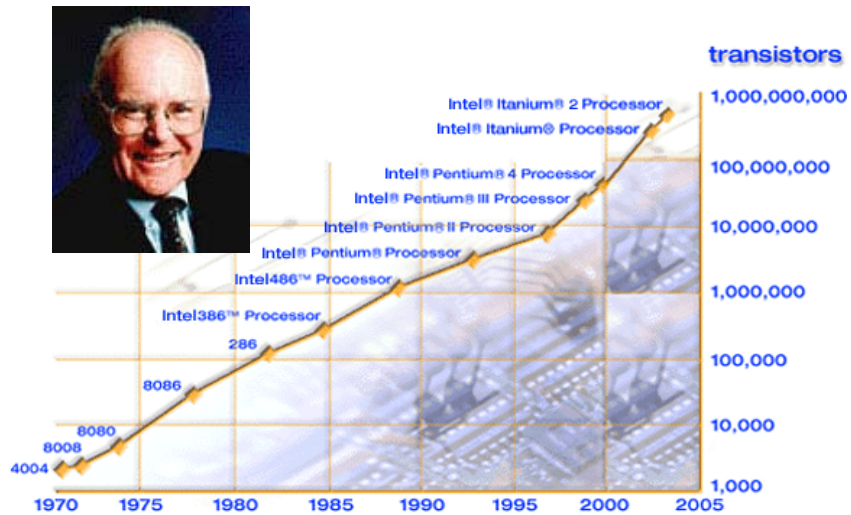
Systemtechnische Gliederung Ebenen („Tier“)



Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

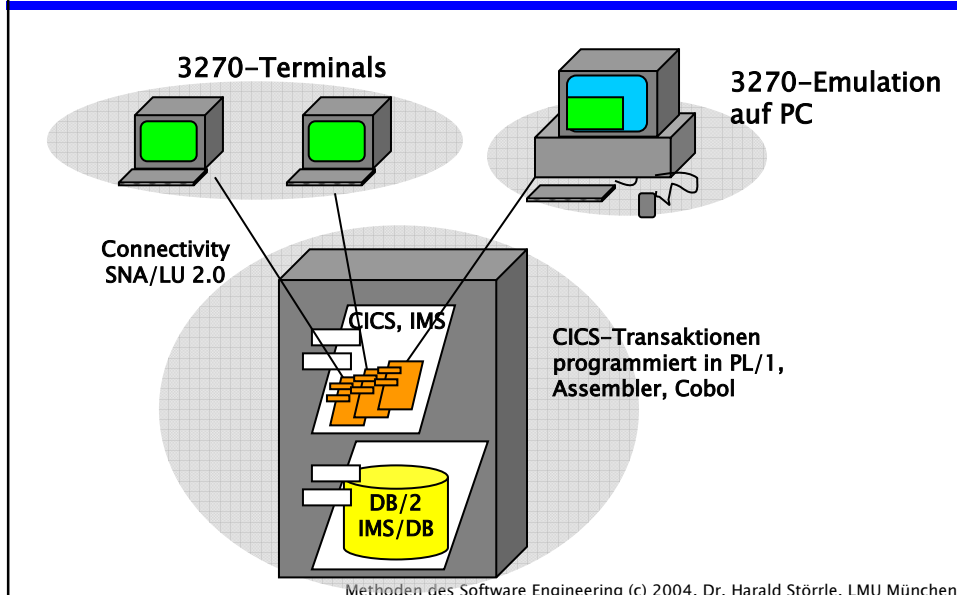
Moore's Gesetz

Die Integrationsdichte von ICs verdoppelt sich alle 12-18 Monate.



Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Klassische Host-zentrierte Systemarchitektur

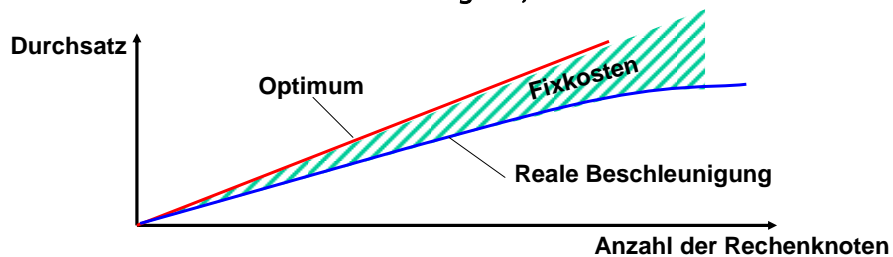


Vom Host zu Client/Server: Hin...

- **Kosten**
 - Lizenzen
 - Skalierung
 - kostengünstige gemäßigte Ausfallsicherheit
- **Der PC ermöglichte den Abteilungen größere Autonomie.**
- **Die Informatiker suchten die Herausforderung.**
- **C/S war schicker.**
- **IBM hat eine ungeschickte Politik betrieben.**

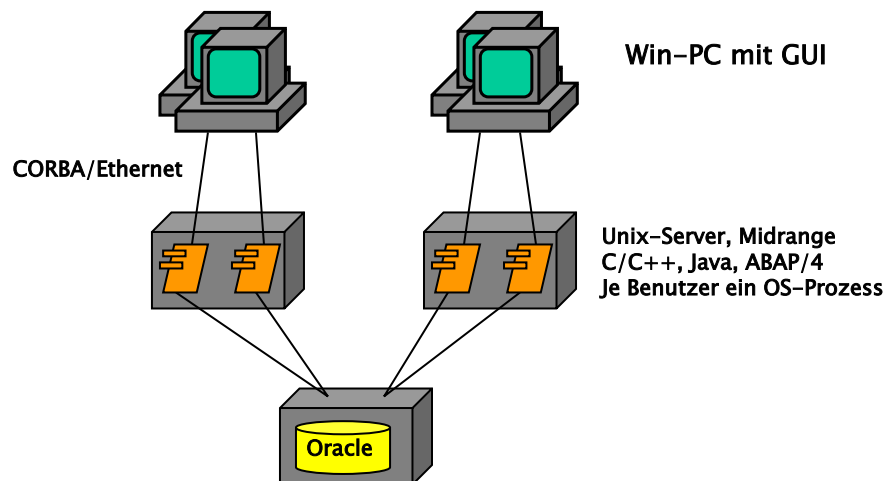
Skalierbarkeit

- Eine definierte Aufgabe kann von einer Recheneinheit des Typs R in k Zeiteinheiten bearbeitet werden.
- Die gleiche Aufgabe kann von n Recheneinheiten des Typs R in m Zeiteinheiten bearbeitet werden. Die Beschleunigung ist also k/m .
- Das Optimum ist k/n , wird aber wegen Fixkosten („Overhead“) und Artefakten (z.B. max. Zahl von DB-Verbindungen u.ä.) praktisch nicht erreicht. Es ist durchaus möglich, daß $m > n$.



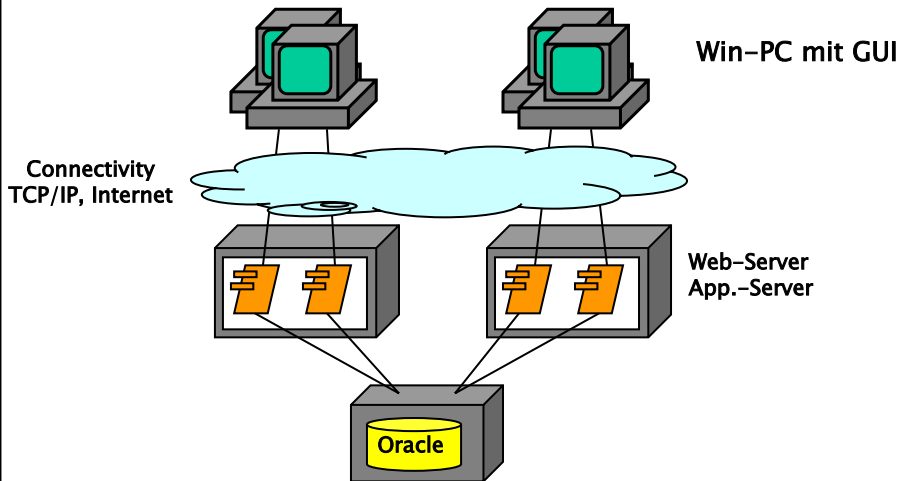
Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Klassische Client/Server-Konfiguration lokale Fat-Clients



Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Klassische Client/Server-Konfiguration J2EE



Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Ausfallsicherheit – Grundlagen

- Seien X und Y zwei stochastisch voneinander unabhängige Zufallsvariablen, die die Verfügbarkeit eines Rechenknotens in Prozent (bzw. als reelle Zahl zwischen 0 und 1) angeben.
- **Wahrscheinlichkeit, daß...**
 - beide verfügbar sind: $X \cdot Y$
 - beide ausfallen: $(1-X) \cdot (1-Y)$
 - genau einer ausfällt: $(1-X) \cdot Y + X \cdot (1-Y)$

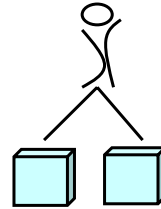
	Y	\bar{Y}
X		
\bar{X}		

Summe aller Felder: 1

Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Ausfallsicherheit – Anwendung

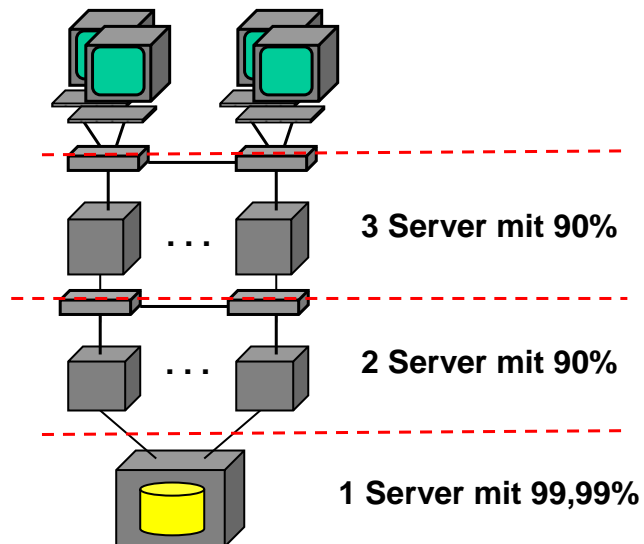
- Gestaltung eines Gesamtsystems
- 1) Redundante Verschaltung
 - Für Dienstbringung reicht ein funktionierender Server aus.
- 2) Schichtung
 - Für Dienstbringung müssen beide Server laufen



Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

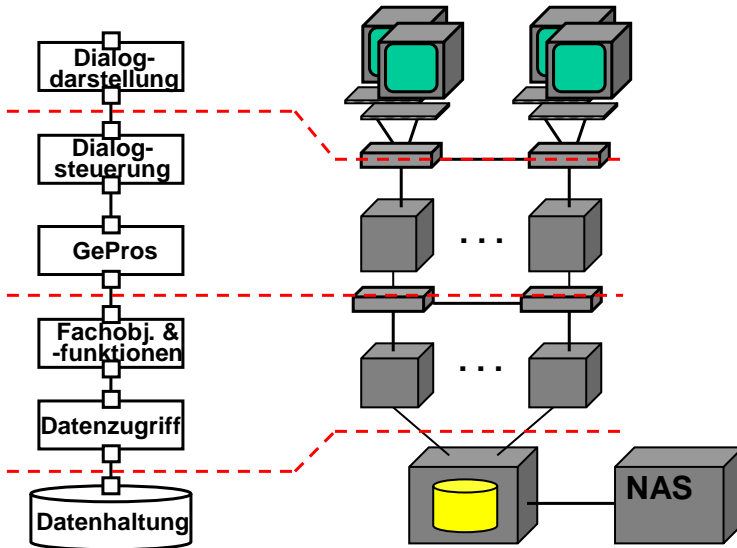
Ausfallsicherheit – Beispiel

Wie groß ist die Verfügbarkeit dieses System?



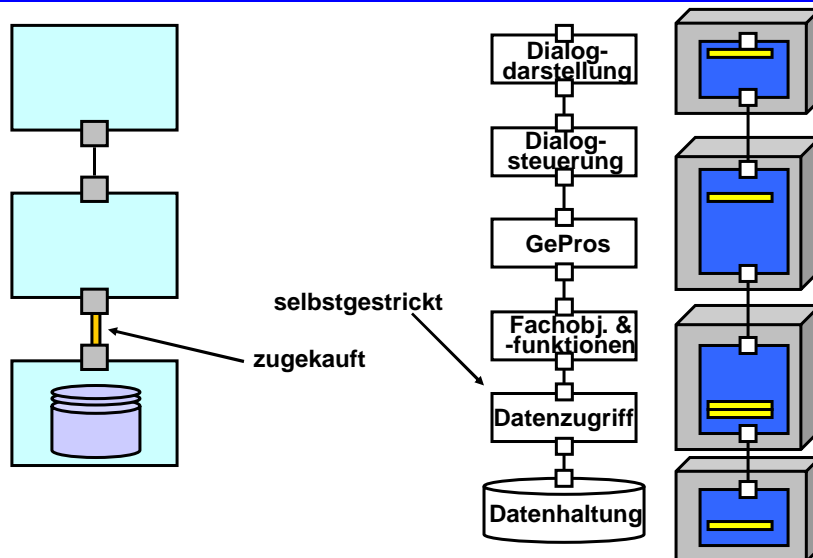
Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Klassische Client/Server-Konfiguration n-tier thin Client



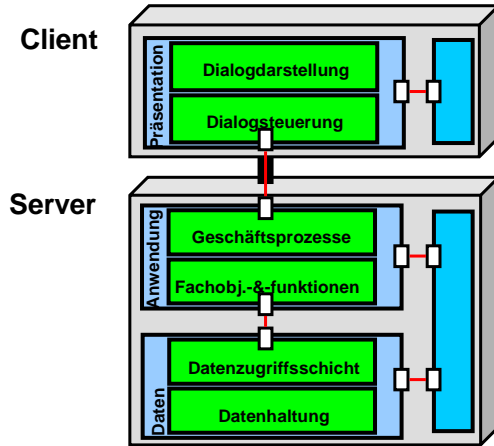
Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Middleware in der Persistenzschicht O2R-Mapping



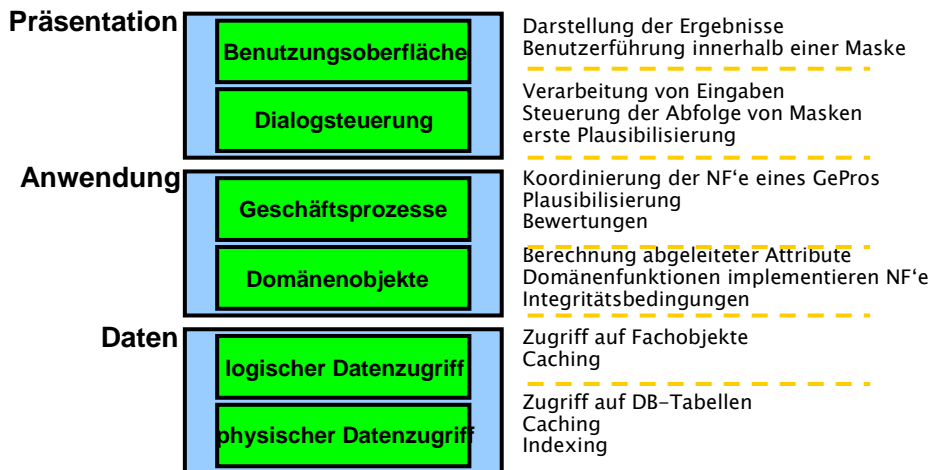
Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Kapselung von Middleware



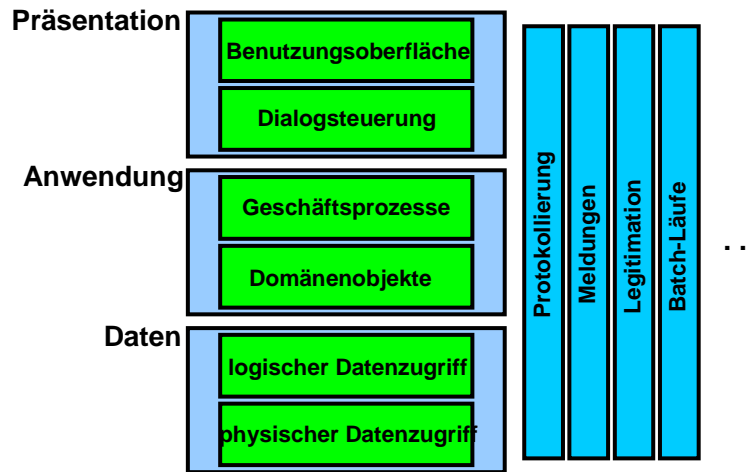
Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Schichtenarchitektur Aufgaben der einzelnen Schichten

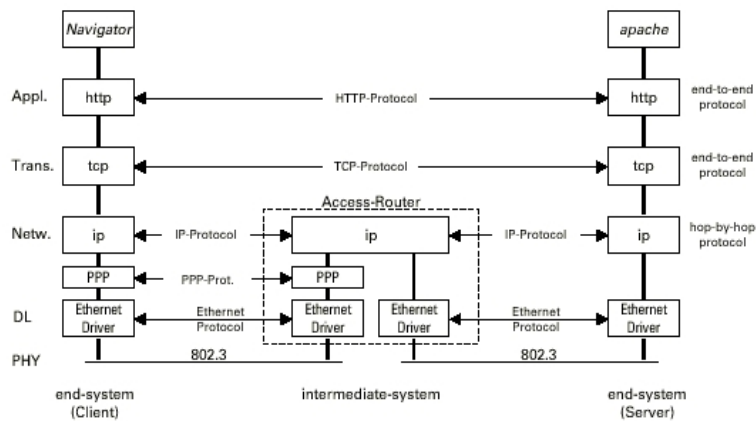


Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Schichtenarchitektur Übergreifende Dienste

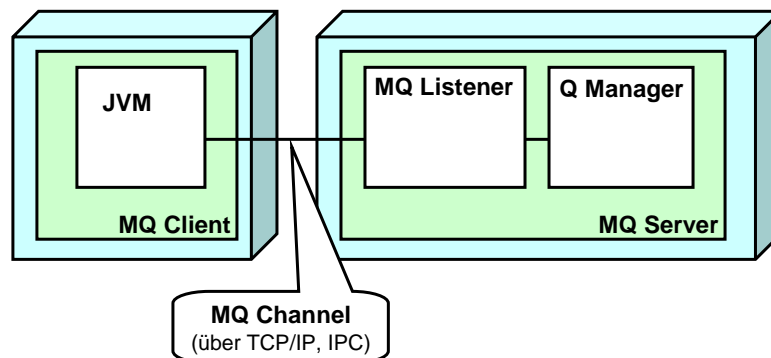


Beispiel: ISO/OSI-7-Schichten-Modell



MQSeries

- **Paradigmatisches Beispiel für Message-Oriented Middleware**
 - 85 % Marktanteil
 - auf 35 Plattformen verfügbar
 - heißt neuerdings WebSphere MQ



Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

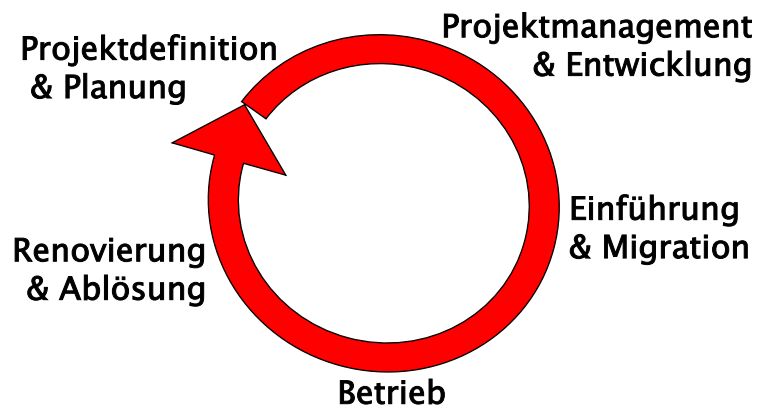
Vom Host zu Client/Server: ...und zurück

- **Kosten**
 - Stark steigende Betriebskosten rücken „Total Cost of Ownership“ (TCO) in den Mittelpunkt.
- **IBM hat sich nach der Krise Mitte der 90'er gewandelt.**
- **Im Internet-Zeitalter zählt extreme Ausfallsicherheit wieder**
 - Im 24-Stunden-Betrieb gibt es kein Batch-Fenster
 - Im 7-Tage-Betrieb gibt es kein Wartungswochenende
 - Im Online-Betrieb kostet jeder Ausfall Gewinn

Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Total Cost of Ownership (TCO)

- Die Kosten eines Systems über seine gesamte Lebensdauer setzen sich aus verschiedenen Faktoren zusammen.



Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

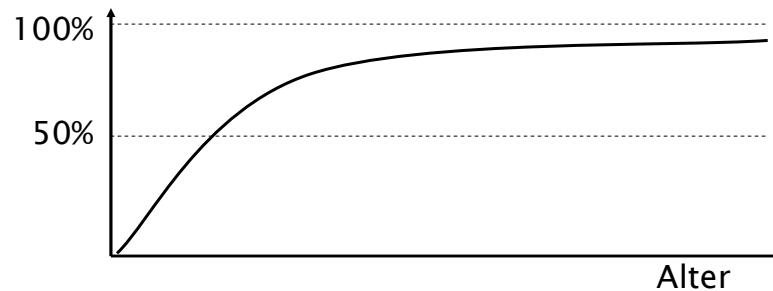
Relevante Themen im Betrieb von Systemen

- Ausfallsicherheit / Verfügbarkeit
- Durchsatz / Leistung / Skalierbarkeit
- Security / Safety
- Backup / Speicherlösungen
- Stapelverarbeitung („Batch-Betrieb“)
- Quality-of-Service (QoS)

Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Betriebskostenanteil vs. Lebensdauer

Anteil der Betriebskosten
an den gesamten Kosten
[%]



Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Bedeutung von Großrechnern heute Verbreitung

- **90 % der weltweit größten Unternehmen setzten S/390 als zentralen Server ein.**
 - [Karlheinz Kronauer, Marketing Manager Software AG in CZ
21.7.2003]
- **ca. 2/3 aller weltweit relevanten wirtschaftlichen Daten sind in EBCDIC-Format gespeichert**
- **60 % aller aus dem Web aufrufbaren Daten befinden sich auf Hosts**
- **16000 Unternehmen setzen CICS als Transaktionsmonitor ein**
- **führende Beratungsunternehmen wie z. B. Gartner Group bescheinigen der S/390-Hardware eine Spitzenstellung unter den Rechnern**

Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Bedeutung von Großrechnern heute

Die Meinung der Gartner Group

<i>Hersteller</i>	IBM	SUN	HP	Compaq	
<i>Rechner</i>	S/390	Exxx	9000	Alpha	Proliant
<i>Betriebssystem</i>	OS/390	Solaris	HP-UX	Tru64	NT 4.0
System Performance	15	15	15	12	6
Clustering Performance	5	2	2	3	1
Single System Availability	15	9	9	9	3
Multiple Systems Availability	15	9	12	12	3
Workload Management	15	6	9	6	3
Partitioning	10	6	2	2	2
Systems Management	10	6	8	6	6
Summe	85	53	57	50	24

Studie der Gartner-Group, 1999]

Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Gegenüberstellung der Technik

Host / Unix

	Host	Unix
Ursprung	EnterpriseComputing	WorkgroupComputing
Batch-Verarbeitung	JCL und RZ-Controllsysteme	Kaum, Script-Lösungen (SHELL oder PERL)
Online	TM IMS und CICS	Daemonprogrammierung 1 Benutzer 1 Prozeß Eigenprogrammierung Kaum TM (Tuxedo)
Oberflächen	Zeichenorientiert (3270) Browser	Zeichenorientiert (VT100) GUI (X/Windows) Browser
Programmiersprachen	PL/I, COBOL, Assembler C/C++ u. Java (eher wenig)	C/C++ u. Java
Systemcalls	Strikte Trennung von System- und Anwendungsprogrammierung	Mischung aus beiden
Workloadmangement	Integriert in OS/390 homogener Rechnerverbund (Sysplex)	Häufig heterogene Clusterlösungen
Verfügbarkeit	> 99,9 %	< 99
Datenbanken	IMS/DB, DB/2	Oracle, Informix, DB/2, ...
Präsentation	3270-Terminals	graphische Fenstermanager
Ablaufumgebung	CICS, IMS	unüblich (Tuxedo, App.-Server)

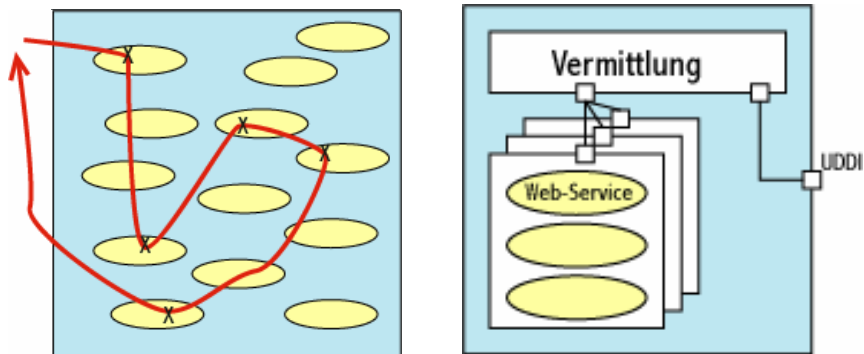
Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Vom Host zu Client/Server: ...und weiter Richtung SOA

- **SOA = Service Oriented Architecture**
- **Host-Architektur**
 - ist leicht zu betreiben,
 - aber ziemlich starr und nicht gerade billig
- **klassische C/S-Architektur**
 - ist sehr flexibel und skalierbar
 - aber sehr aufwändig in Herstellung, Betrieb und Wartung
- **Service Oriented Architecture**
 - flexibel und skalierbar
 - und abstrahiert von Plattform
 - Integraton/Konsolidierung nach Fusion, Kooperationen
 - Controlling der Betriebskosten, Outsourcing
 - aber großer Overhead und hohe Umstrukturierungskosten und unsicherer QoS

Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

Web Services



Methoden des Software Engineering (c) 2004, Dr. Harald Störrle, LMU München

