

Softwaretechnik 2

Einführung



Struktur

1. Einführung

1.1 Einführung und Begriffe

1.2 Verteilungsmodelle und
Middleware-Technologien (im Überblick)

1.3 Kriteriengeleitete Technologieevaluation

Wo stehen wir jetzt?

OOA-Modell beschreibt **WAS** erstellt werden soll, aber nicht **WIE**.

Was muss geklärt werden?



OOA-Modell



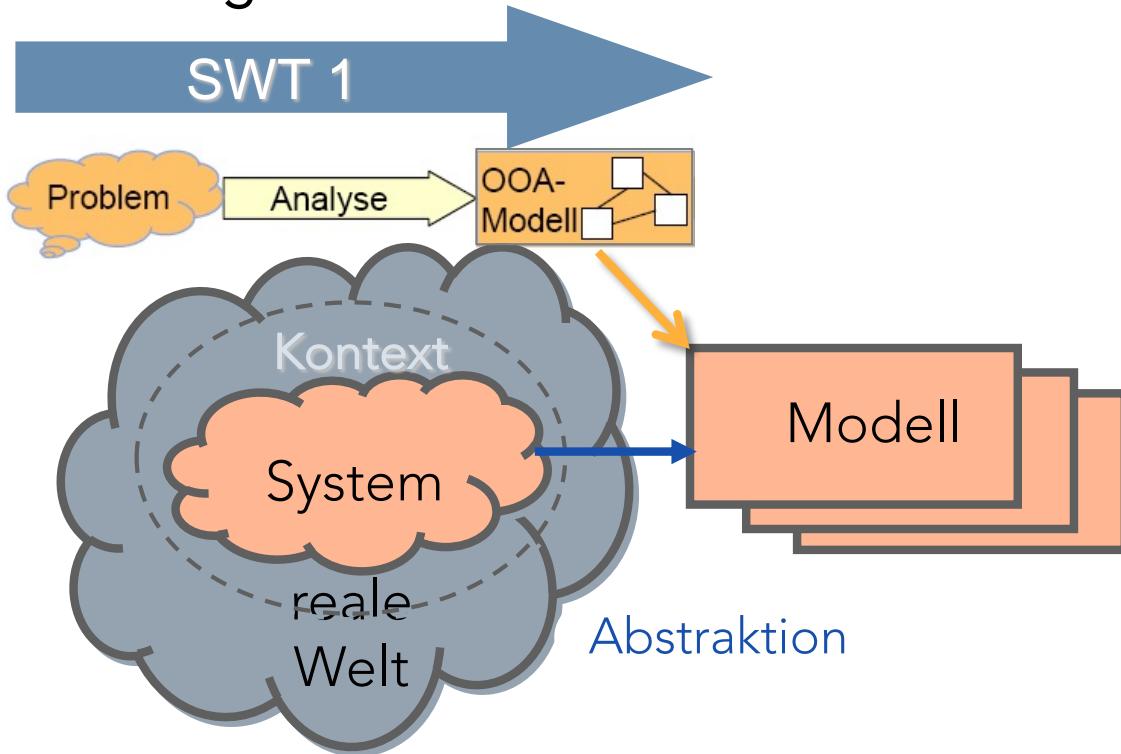
Implementierung/
Realisierung

Vorgehen

Beispiel: Hausbau



Vorgehen

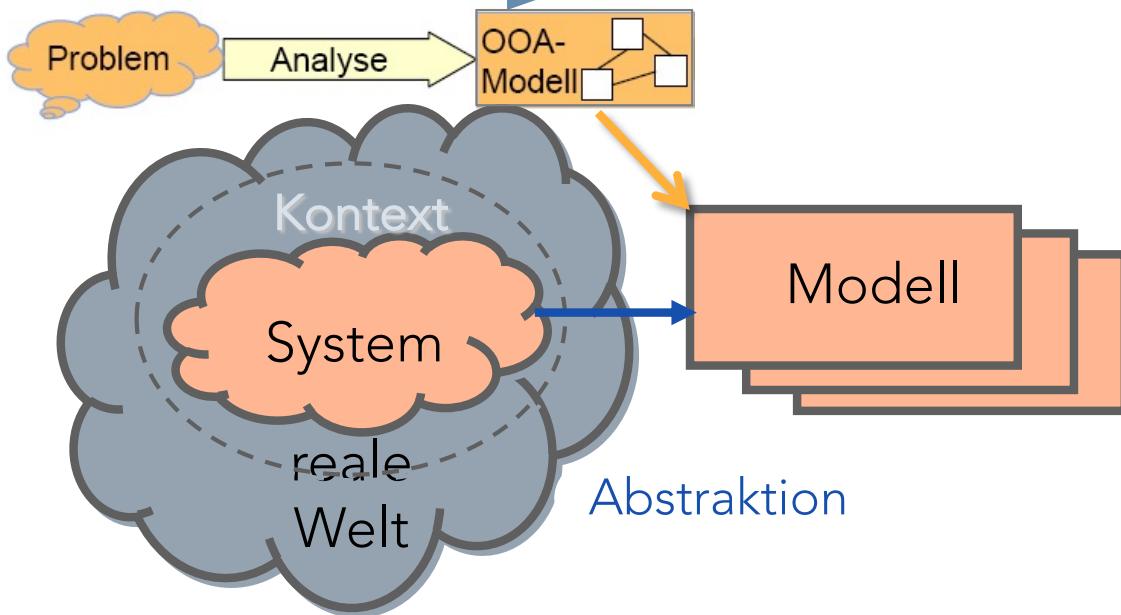


Beispiel: Hausbau



Vorgehen

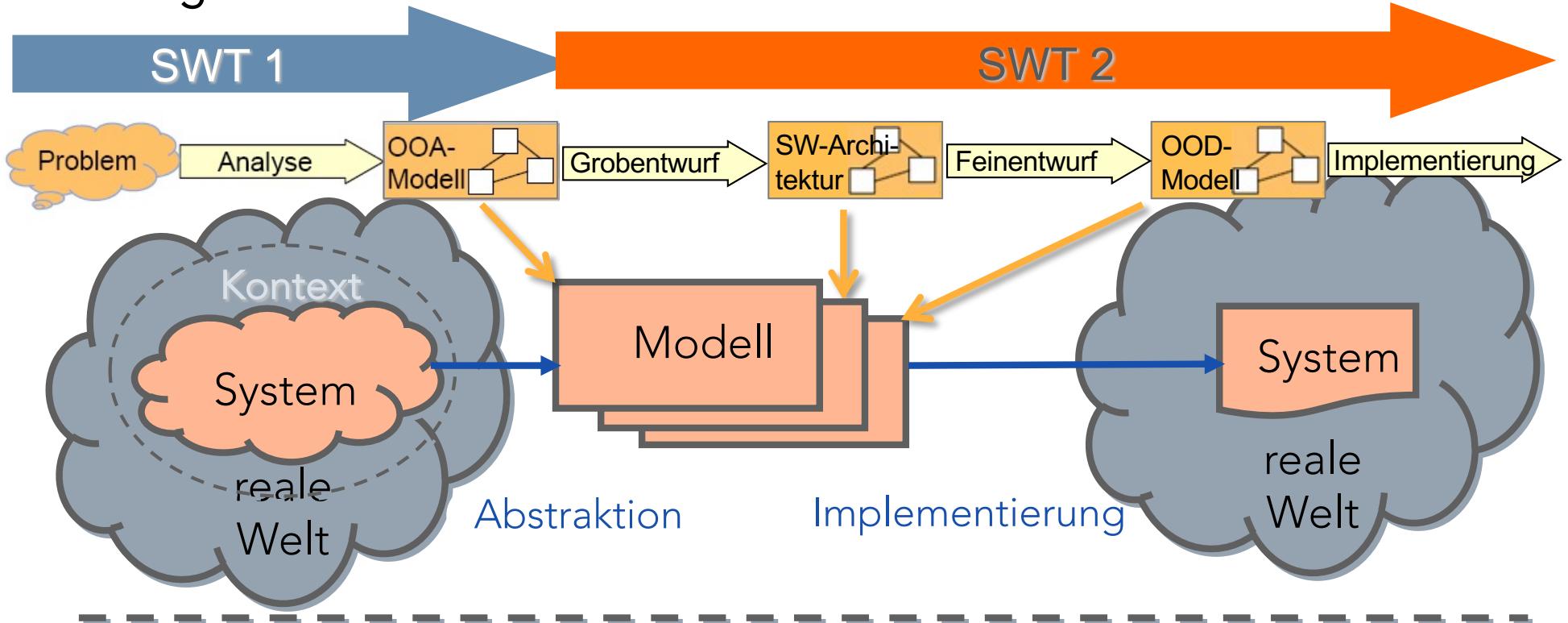
SWT 1



Beispiel: Hausbau



Vorgehen



Beispiel: Hausbau



Begriffe

Architektur

- Architektur ist die Kombination von utilitas, firmitas und venustas.
frei nach Vitruvius (Römischer Architekt 90-20 v. Chr.)
 - Funktionalität (Zweckmäßigkeit)
 - Stabilität (Beständigkeit)
 - Ästhetische Gestaltung (Eleganz)



Titusbogen

(am Eingang zum
Forum Romanum)



Pont du Gard

(Aquädukt)



Amphitheatrum Flavium

(heute bekannt als Kolosseum)

Zielgerichtete Architekturen



Zielgerichtete Architekturen



Burj Khalifa, 560 Meter

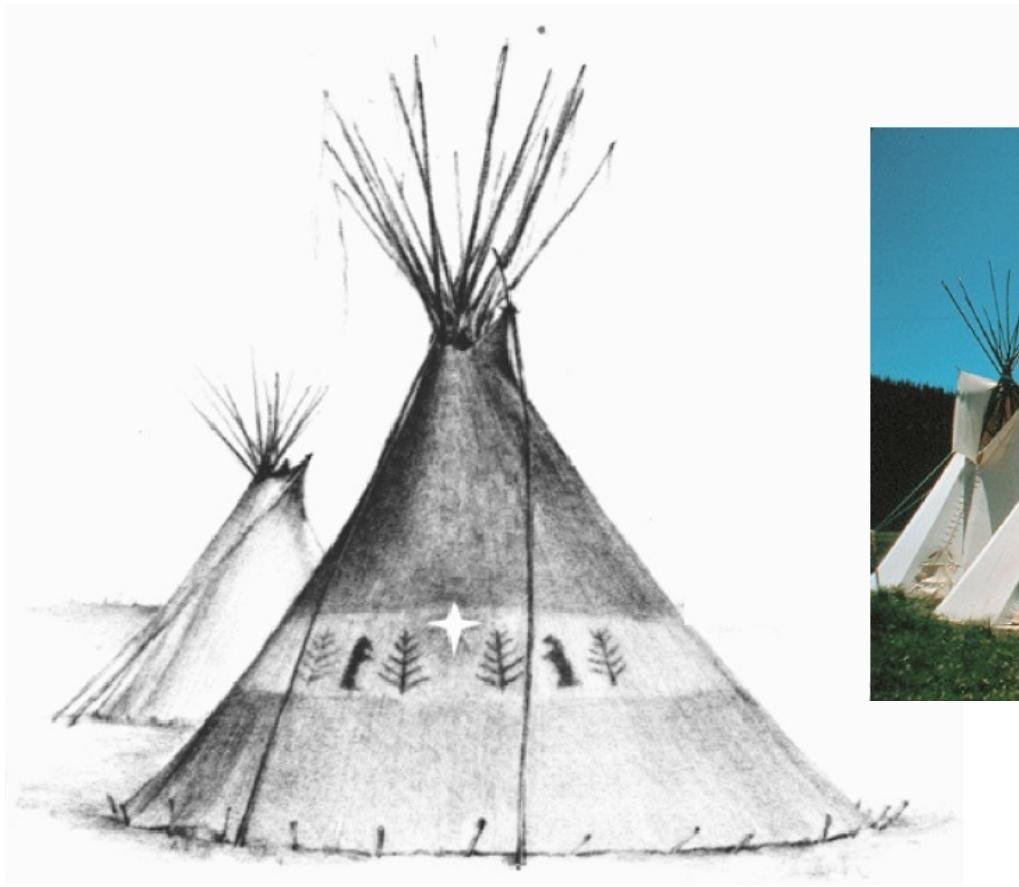


Burj al Arab, 7 Sterne Hotel

Zielgerichtete Architekturen



Zielgerichtete Architekturen



Zielgerichtete Architekturen

Und hier ???

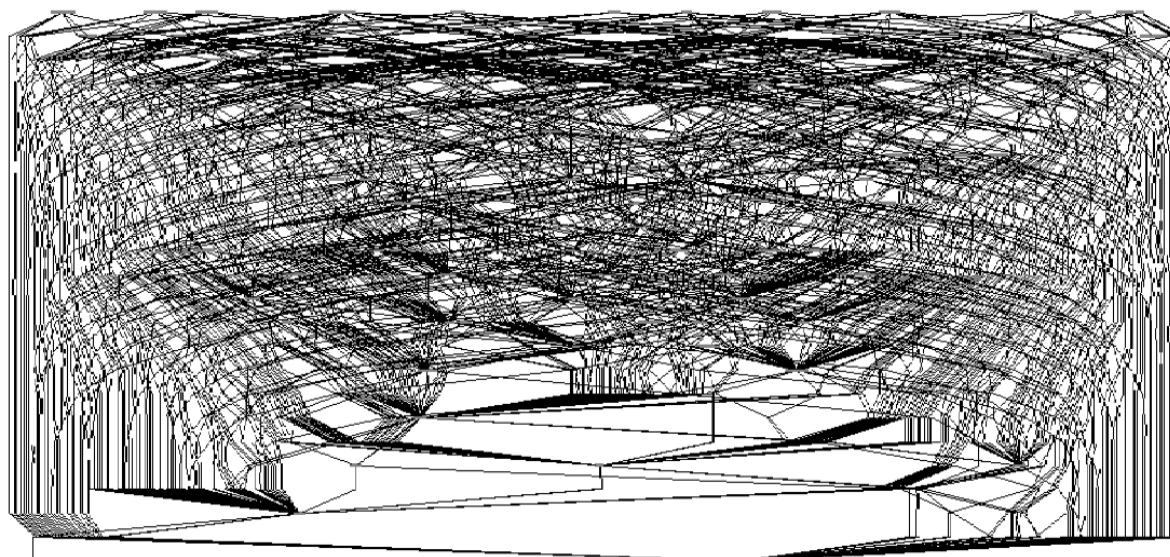


Begriffe

Softwarearchitektur (1)

- Softwarearchitektur ist die Kombination von utilitas, firmitas und venustas.
frei nach Vitruvius (Römischer Architekt 90-20 v. Chr.)
 - Funktionalität: das Softwaresystem erfüllt seine Anforderungen
 - Stabilität: das Softwaresystem ist langlebig und stabil gegenüber Veränderungen
 - Ästhetische Gestaltung:
die Strukturen des Softwaresystems weisen eine Klarheit auf, die das Verständnis und die Kommunikation im Team erleichtert

[vgl. Rausch 2006]



Softwarearchitektur
oder
Chaos ?

[Quelle: Erich Gamma, "100 OO Frameworks, Pitfalls and Lessons Learned", 1997]

Begriffe

Softwarearchitektur (2)

- Der Name „Softwarearchitektur“ ist als Analogie zu sehen, als eine programmatische Aussage, eine Absichtserklärung:
„da, wo die Architektur heute steht, da wollen wir auch hin“ [Vgl. Rausch 2006]

Softwarearchitektur (3)

- „Softwarearchitektur:
Das ist die Königsdisziplin des Software Engineering.“

[Ernst Denert, 2004]

Begriffe

Softwarearchitektur (4)

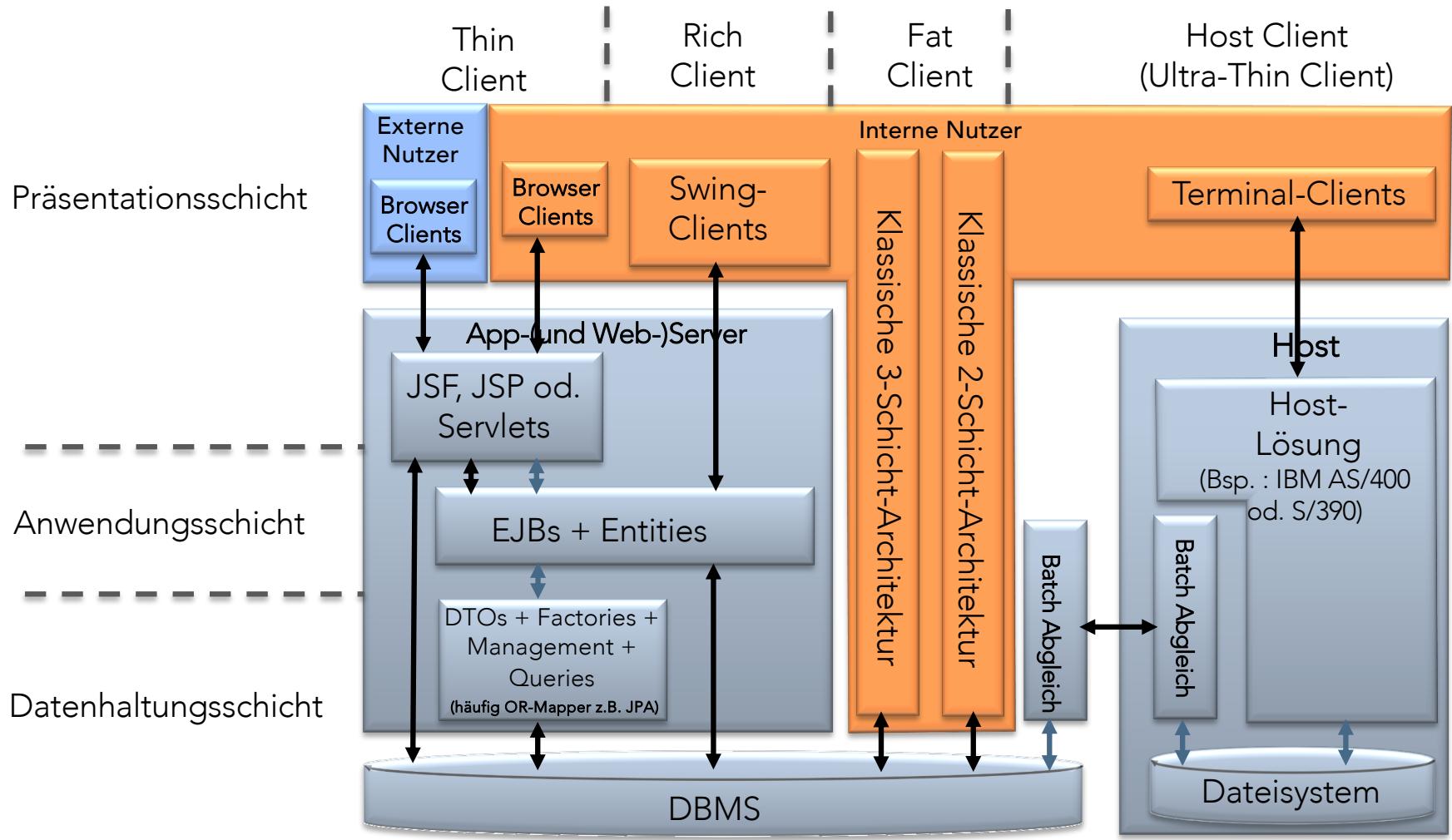
- „Die Softwarearchitektur eines Systems ist die Menge an Strukturen, die erforderlich sind, um über das System nachzudenken. [Konkret die Strukturen], die **Softwareelemente**, **Beziehungen** zwischen ihnen und **Eigenschaften** von beiden umfassen.“
[Vgl. Bass et al. 20012]

Weitere Definitionen

- Es existieren über 90 Definitionen des Begriffs „Softwarearchitektur“

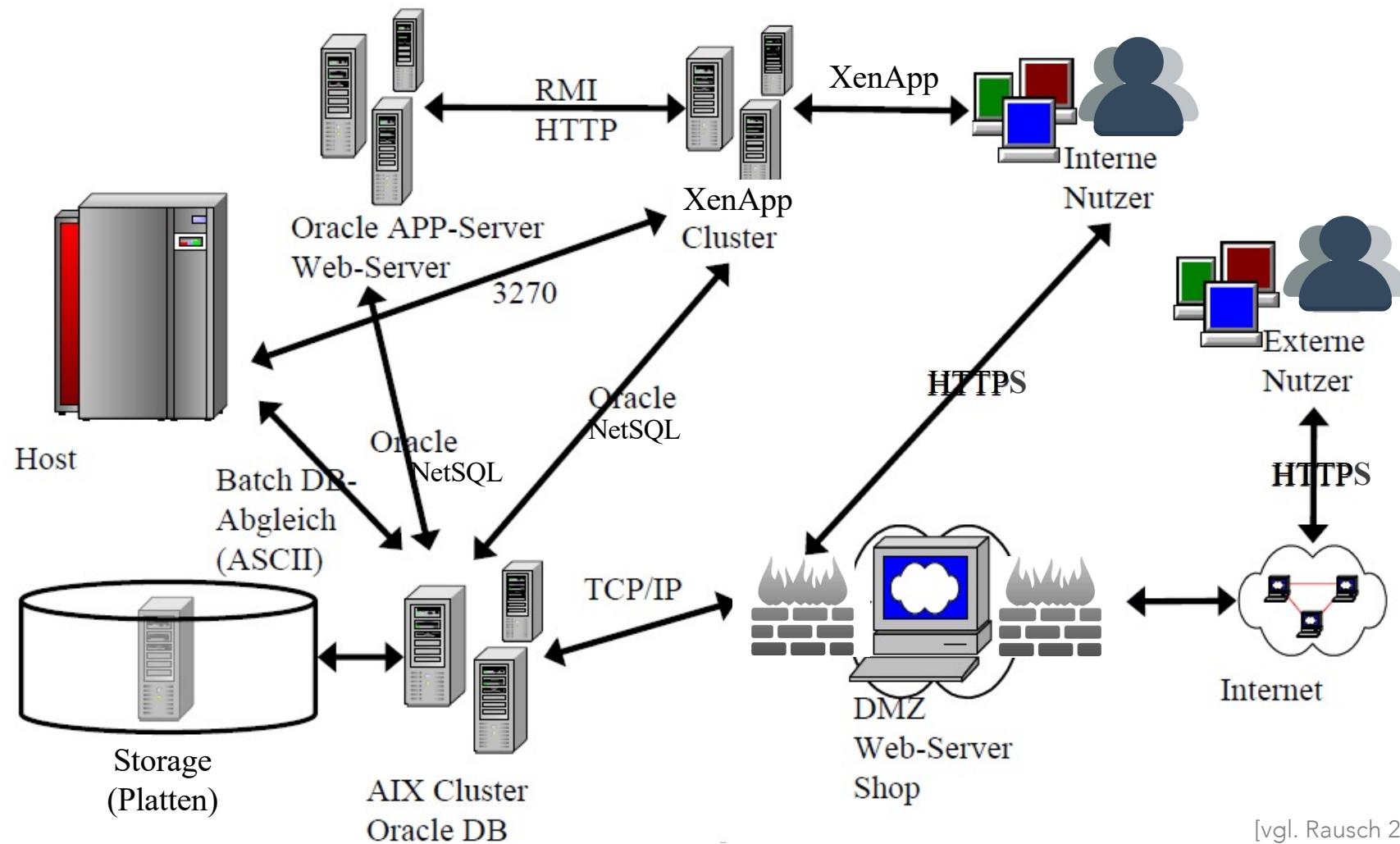
[vgl. <http://www.sei.cmu.edu>]

Beispiel: Softwarearchitekturen



[angelehnt an Rausch 2006]

Beispiel: Systemarchitektur



[vgl. Rausch 2006]

Struktur

1. Einführung

- 1.1 Einführung und Begriffe
- 1.2 Verteilungsmodelle und
Middleware-Technologien (im Überblick)
- 1.3 Kriteriengeleitete Technologieevaluation

Technologien

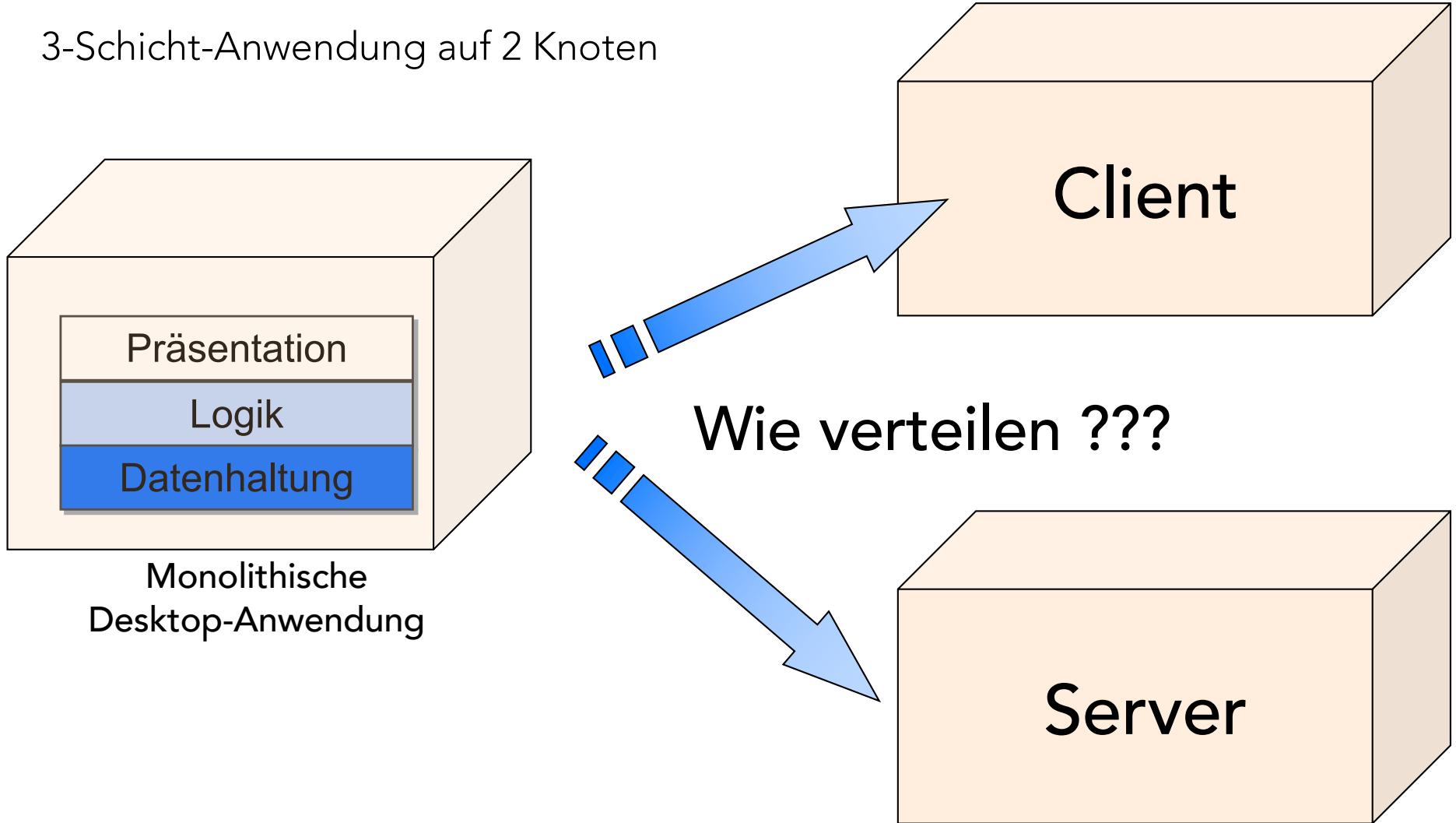
Entscheidung: Monolithisches oder verteiltes System

- **monolithische Desktop-Anwendung** = Anwendung, die
 - lokal auf dem Arbeitsplatzrechner (engl.: Desktop) installiert ist.
 - Beispiele:
 - die meisten Office-Programme,
 - e-Mail-Programme,
 - CAD-Systeme,
 - Computerspiele,
 - Integrierte Entwicklungsumgebungen
 - Web-Browser ...

- **Verteilte-Anwendungen** = komplexe Anwendung, die
 - in einem verteilten System abläuft (→ nächste Folie).
 - Beispiele:
 - Reisebuchungssysteme (z.B. ab-in-den-urlaub),
 - SAP,
 - ebay, amazon, facebook, ...

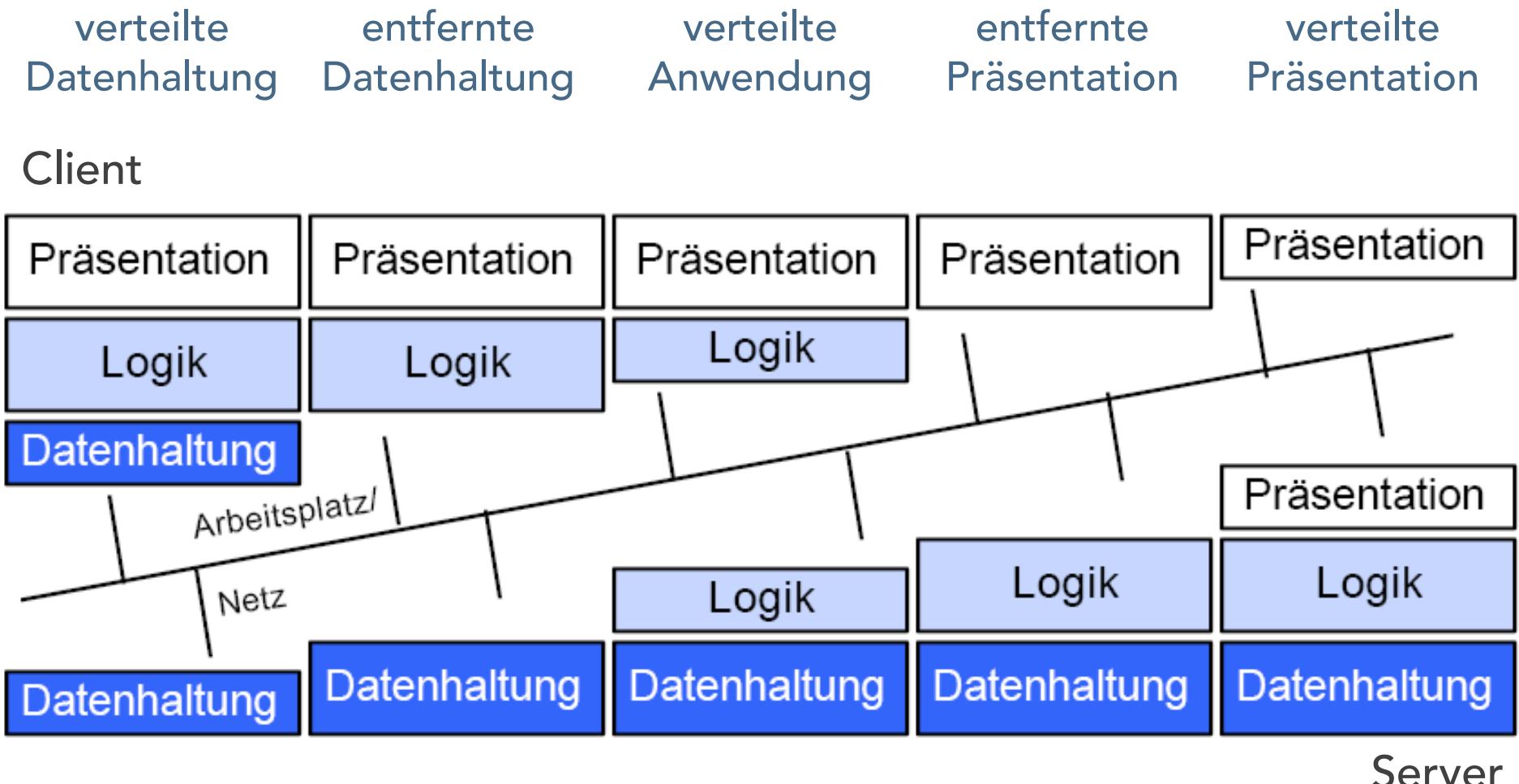
Technologie-Entscheidung: Verteilungsmodell

3-Schicht-Anwendung auf 2 Knoten



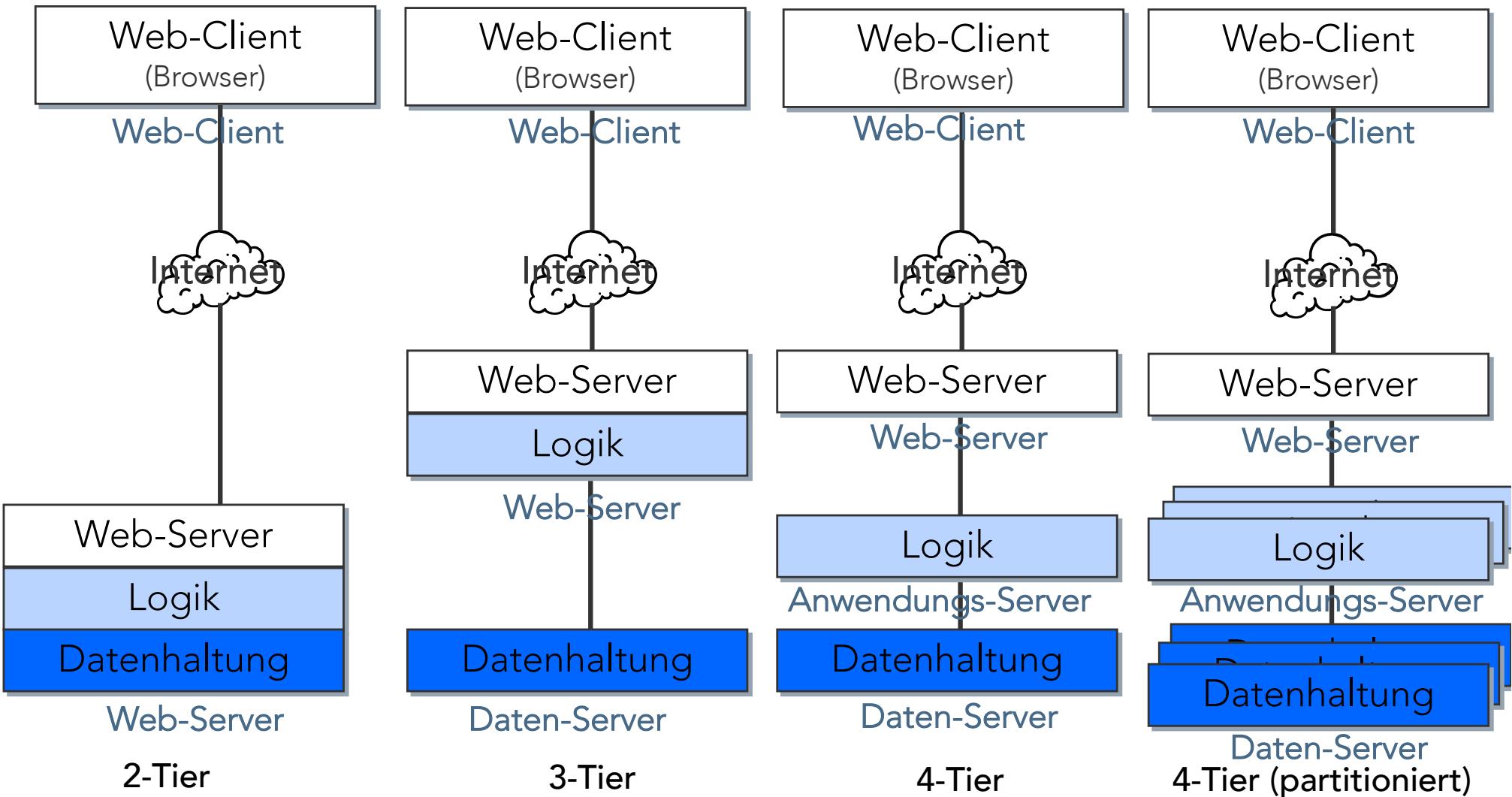
Technologien

Verteilungsmodelle in einer 2-Tier-Architektur



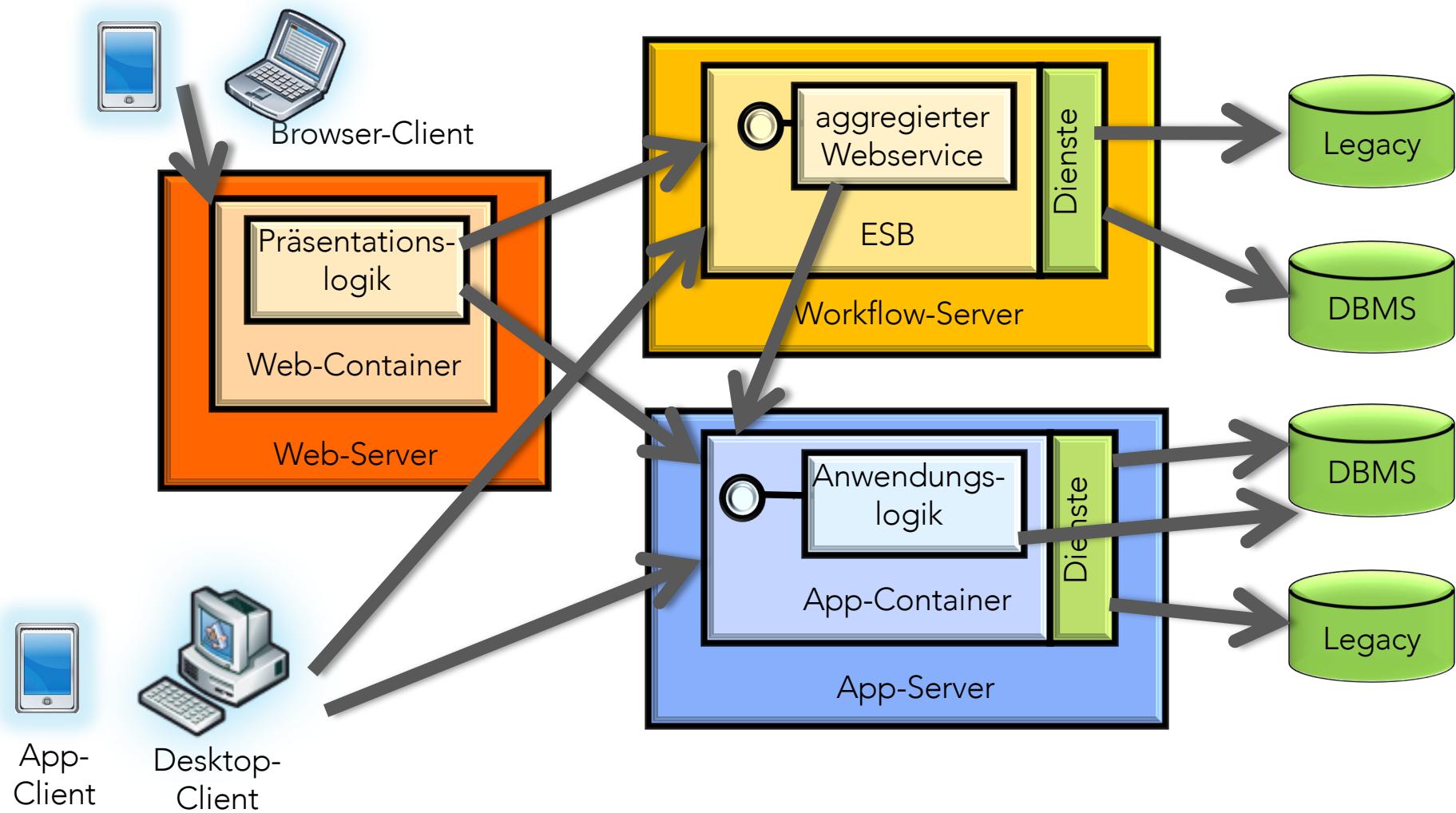
Technologien

Verteilungsmodelle in einer n-Tier Architektur (einige Beispiele)



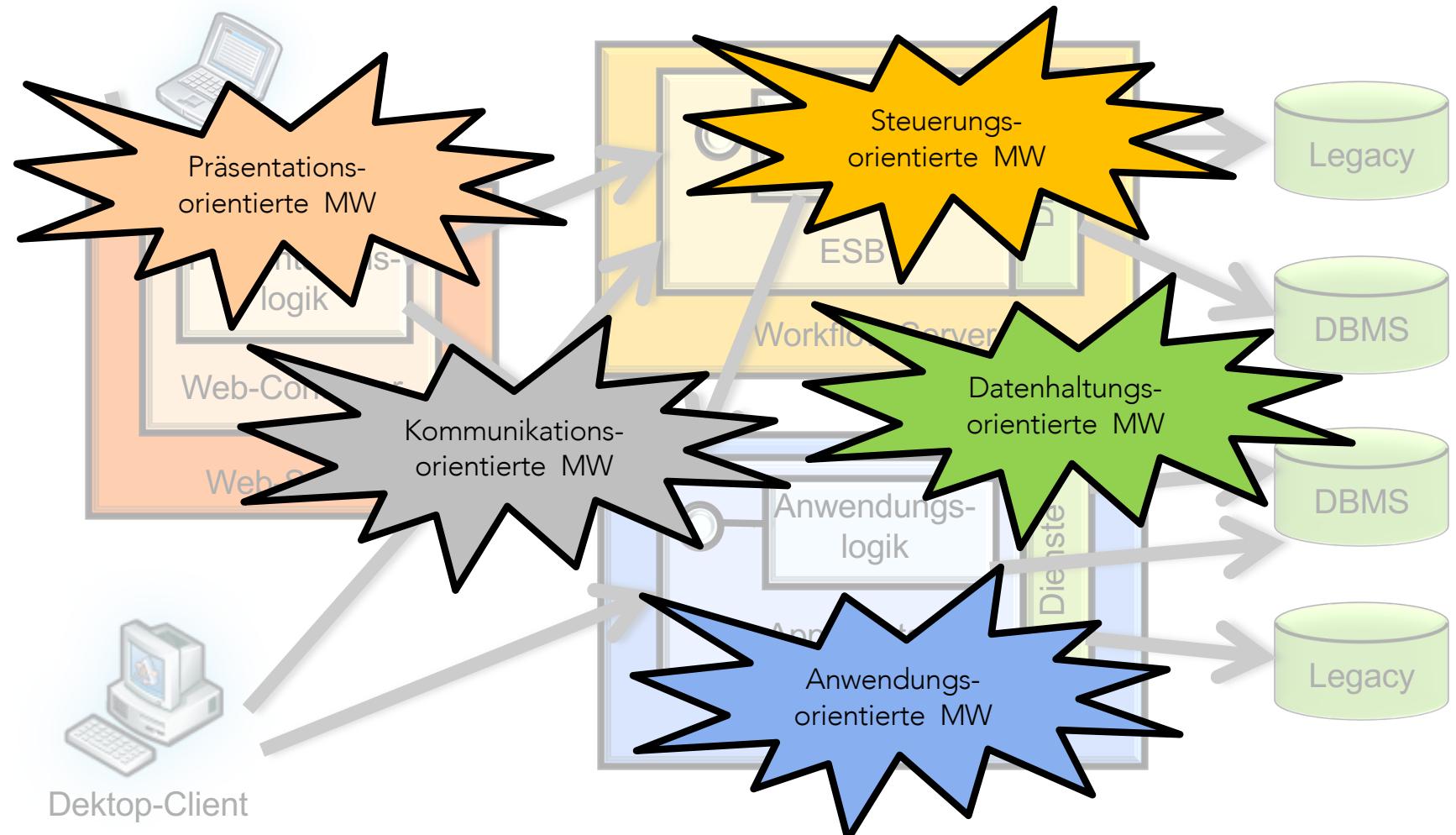
Technologien

Generisches Modell einer n-Tier Architektur



Technologien

Generisches Modell einer n-Tier Architektur



Technologien

Middleware-Varianten

- Kommunikationsorientierte Middleware
 - RPC, RMI, Message-Oriented Middleware (MOM)
- Präsentationsorientierte Middleware
 - Beispiele: Java Servlets , JSP und JSF
- Anwendungsorientierte Middleware
 - Beispiele: CORBA, Java EE, .Net
- Steuerungs- bzw. Serviceorientierte Middleware
 - Beispiele: WfMC (SOA → BPEL)
- Datenhaltungsorientierte Middleware
 - Beispiele: relationaler, graph-basierter, dokumentenzentrierter, ... Zugriff

Struktur

1. Einführung

- 1.1 Einführung und Begriffe
- 1.2 Verteilungsmodelle und
Middleware-Technologien (im Überblick)
- 1.3 Kriteriengeleitete Technologieevaluation

Grundlegende Technologie(auswahl)entscheidungen

- Wahl der Plattform
 - Microsoft-Plattform oder Nicht-Microsoft-Plattform?
- Wahl der Programmiersprache(n)
 - In Bezug zur Plattform
 - Objektorientierte Programmiersprachen üblich
- Wahl des GUI-Systems
 - nativ oder Web?
 - Technologie
 - Konkretes Framework
- Wahl der Datenhaltung bzw. der Datenbank
 - Relationale oder NoSQL-DBMS?
 - OR-Mapper einsetzen?
- Wahl der Kommunikationskanäle
 - RMI
 - WebServices (REST oder SOAP)
 - MOM

Evaluation

Wortentwicklung

valere (lat.) = wert sein, bei Kräften sein

valoir (franz.) = kosten, wert sein

value (engl.) = Wert, Nutzen

evaluation (engl.) = Schätzung, Bewertung

Definition Evaluation

Die systematische Untersuchung von Güte oder Nutzen eines Gegenstands bzw. das Ergebnis einer solchen Untersuchung.

Kriteriengeleitete Evaluation

1. Festlegung der Auswahlkriterien
 - Flipchart etc.
2. Ermittlung der K.O.-Kriterien
 - Liste K.O.-Kriterien
3. Gewichtung der Bewertungskriterien
 - Liste Bewertungskriterien + Gewichtungsverfahren
4. Marktrecherche: Welche Ansätze gibt es?
 - Bücher + Internetrecherche
5. Eingrenzung der Anbieter/Systeme (Stufe 1)
 - Liste Marktrecherche
6. Evaluation der Ansätze/Anbieter (Stufe 2)
 - Aufbau/Installation einer Testumgebung + Prototyp (z.B. Hello World)
7. Vergleichende Bewertung der Ansätze (Zusammenfassung Stufe 2)
 - Bewertungsmatrix

Beispiel: 1. Festlegung der Auswahlkriterien

- Ziel ist die Evaluation von Workflow-Werkzeugen für eine Microsoft Sharepoint Lösung
- Definition der Kriterien
 - Verfügbarkeit und Qualität von Dokumentation und Tutorials
 - Unterstützung einer Versionsverwaltung
 - Wiederverwendbarkeit der Workflows
 - Export – Lauffähigkeit außerhalb der Entwicklungsumgebung
 - Debugging Möglichkeiten
 - Verlaufshistorie
 - Zugriffsrechte
 - Erforderliche Vorkenntnisse
 - Benutzerfreundlichkeit
 - Funktionsumfang
 - Erweiterbarkeit

Beispiel: 2. + 3. Gewichtung der Kriterien

- 2. Ermittlung der K.O.-Kriterien
(z.B. Export oder Zugriffsrechte)
- 3. Gewichtung der Kriterien nach Niklas [easy mind: <http://community.easymind.de>]

Zeile wichtiger als Spalte (grüner Bereich)?	
nein	0
gleichwichtig	1
ja	2

Kriterien	1	2	3	4	5	6	7	8	9	10	11	Summe	Faktor	Prozent
1. Dokumentation, Tutorials	1	0	0	1	1	0	0	1	0	0	0	4	0,04	4%
2. Versionsverwaltung	1	0	0	1	1	0	1	0	0	0	0	4	0,04	4%
3. Wiederverwendbarkeit	2	2	2	2	2	2	2	2	2	2	2	20	0,18	18%
4. Export	2	2	0	2	2	1	2	1	1	1	1	14	0,13	13%
5. Debugging Möglichkeiten	1	1	0	0	1	0	1	0	0	0	0	4	0,04	4%
6. Verlaufshistorie	1	1	0	0	1	0	1	0	0	0	0	4	0,04	4%
7. Zugriffsrechte	2	2	0	1	2	2	2	1	1	1	1	14	0,13	13%
8. Vorkenntnisse	2	1	0	0	1	1	0	0	0	0	0	5	0,05	5%
9. Benutzerfreundlichkeit	1	2	0	1	2	2	1	2	1	1	1	13	0,12	12%
10. Funktionsumfang	2	2	0	1	2	2	1	2	1	1	1	14	0,13	13%
11. Erweiterbarkeit	2	2	0	1	2	2	1	2	1	1	1	14	0,13	13%
Summe												110	1,00	100%

Beispiel: 4. Marktrecherche + 5. Eingrenzung

- 4. Marktrecherche: Welche Ansätze gibt es?
 - Bücher
 - Internetrecherche
 - Messen
 - Hersteller anschreiben
 - Telefoninterviews
 - Vorführtermine vereinbaren
 -
- 5. Eingrenzung der Anbieter (Stufe 1)
 - Anbieter die k.o.-Kriterien nicht erfüllen raus nehmen
 - Bewertung der Kriterien auf Dokumentationsbasis, d.h. ohne ausprobieren => evt. erste Evaluation

Beispiel: 6. Evaluation der Ansätze/Systeme (Stufe 2)

- Aufbau/Installation einer Testumgebung
- Ansatz/System ausprobieren
(am Besten auf Basis eines zuvor definierten einheitlichen Beispiels)
- Beurteilungswerte definieren:

Beurteilungswerte	
Wert	Beschreibung
1	Ungenügend, nicht unterstützt
2	Mangelhaft
3	Ausreichend
4	Befriedigend
5	Gut
6	Sehr gut, voll unterstützt

- Beurteilungswerte sind sinnvoll zu definieren!

Beispiel: 7. Vergleichende Bewertung der Ansätze (Nutzwertanalyse)

Kriterien	Faktor	Entwicklungswerkzeuge					
		SharePoint Designer 2010		Nintex Workflow 2010		Visual Studio 2010	
		Bewertung	Nutzwert	Bewertung	Nutzwert	Bewertung	Nutzwert
1. Dokumentation, Tutorials	0,04	6	0,24	3	0,12	4	0,16
2. Versionsverwaltung	0,04	1	0,04	5	0,2	3	0,12
3. Wiederverwendbarkeit	0,18	2	0,36	6	1,08	6	1,08
4. Export	0,13	2	0,26	4	0,52	6	0,78
5. Debugging Möglichkeiten	0,04	1	0,04	1	0,04	6	0,24
6. Verlaufshistorie	0,04	4	0,16	6	0,24	3	0,12
7. Zugriffsrechte	0,13	5	0,65	6	0,78	4	0,52
8. Vorkenntnisse	0,05	6	0,3	5	0,25	1	0,05
9. Benutzerfreundlichkeit	0,12	4	0,48	6	0,72	2	0,24
10. Funktionsumfang	0,13	4	0,52	4	0,52	6	0,78
11. Erweiterbarkeit	0,13	4	0,52	5	0,65	6	0,78
Ergebnis:			3,57		5,12		4,87

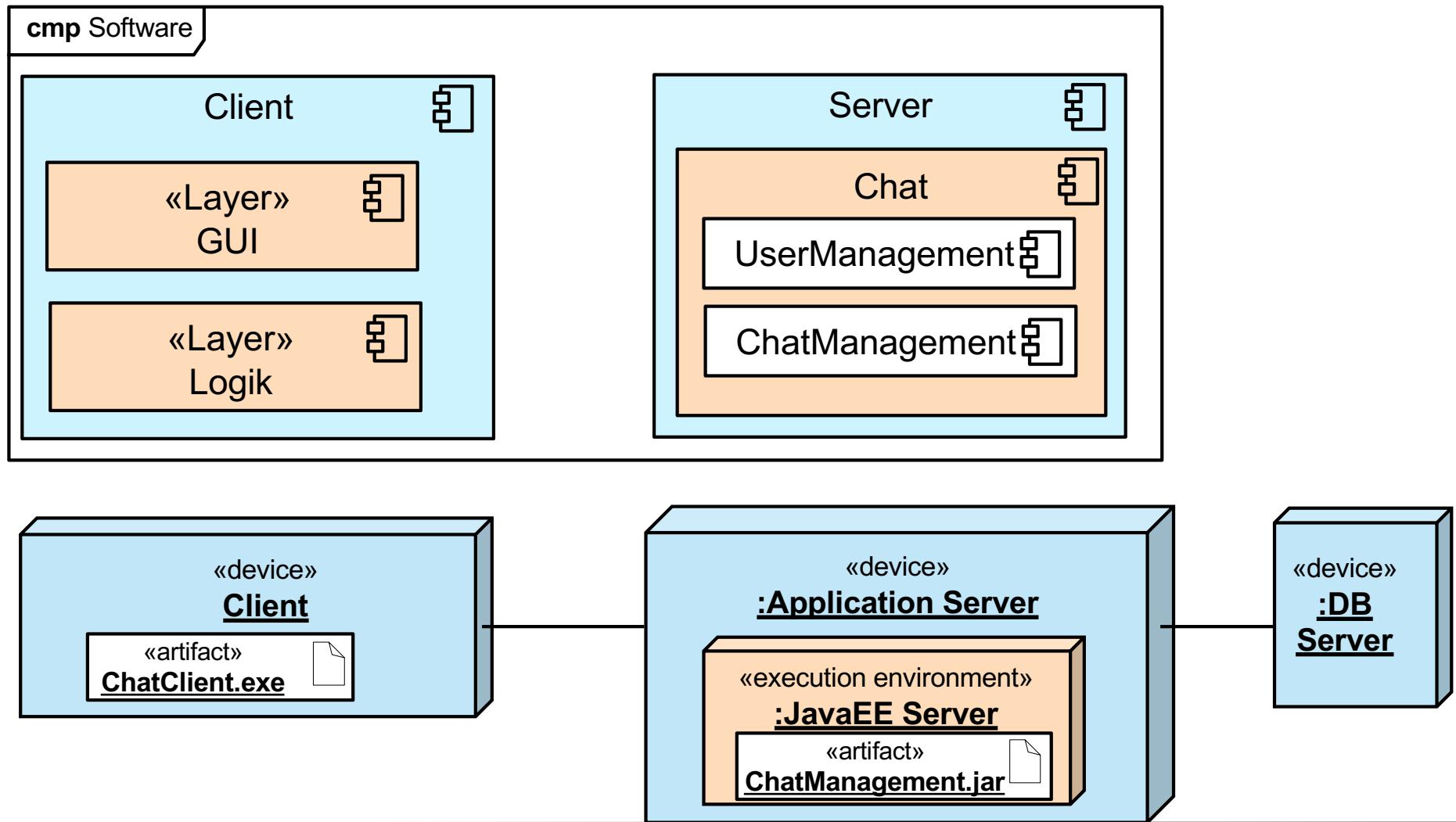
Struktur

1. Einführung

- 1.1 Einführung und Begriffe
- 1.2 Verteilungsmodelle und
Middleware-Technologien (im Überblick)
- 1.3 Kriteriengeleitete Technologieevaluation

Ausblick

Struktur vs. Verteilung → Architekturmodellierung



Fragen

