Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Informationsintegration
Architekturen

17.4.2012

Felix Naumann

Überblick

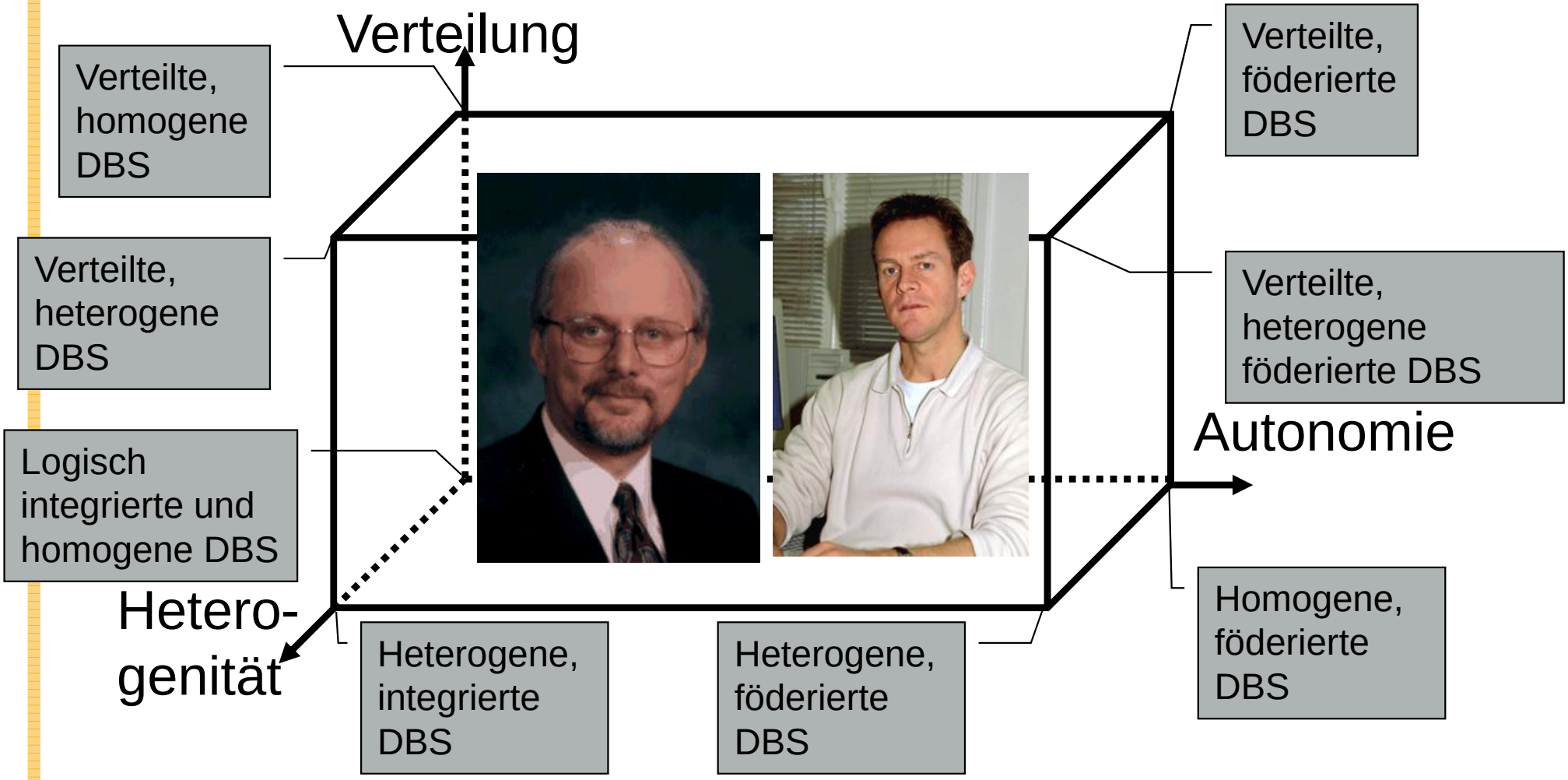
- Überblick über Informationssysteme
  - Klassifikation
  - Materialisiert vs. virtuell
- Architekturen
  - 3 Schichten Architektur
  - 4 Schichten Architektur
  - 5 Schichten Architektur
- Mediator-Wrapper Architektur
  - Gio Wiederholds Definitionen
  - Konfigurationen
  - Mediatoren
  - Wrapper
- Peer-Data-Management
  - Architektur
  - Anwendungen

Klassifikation von Informations-systemen nach [ÖV99]

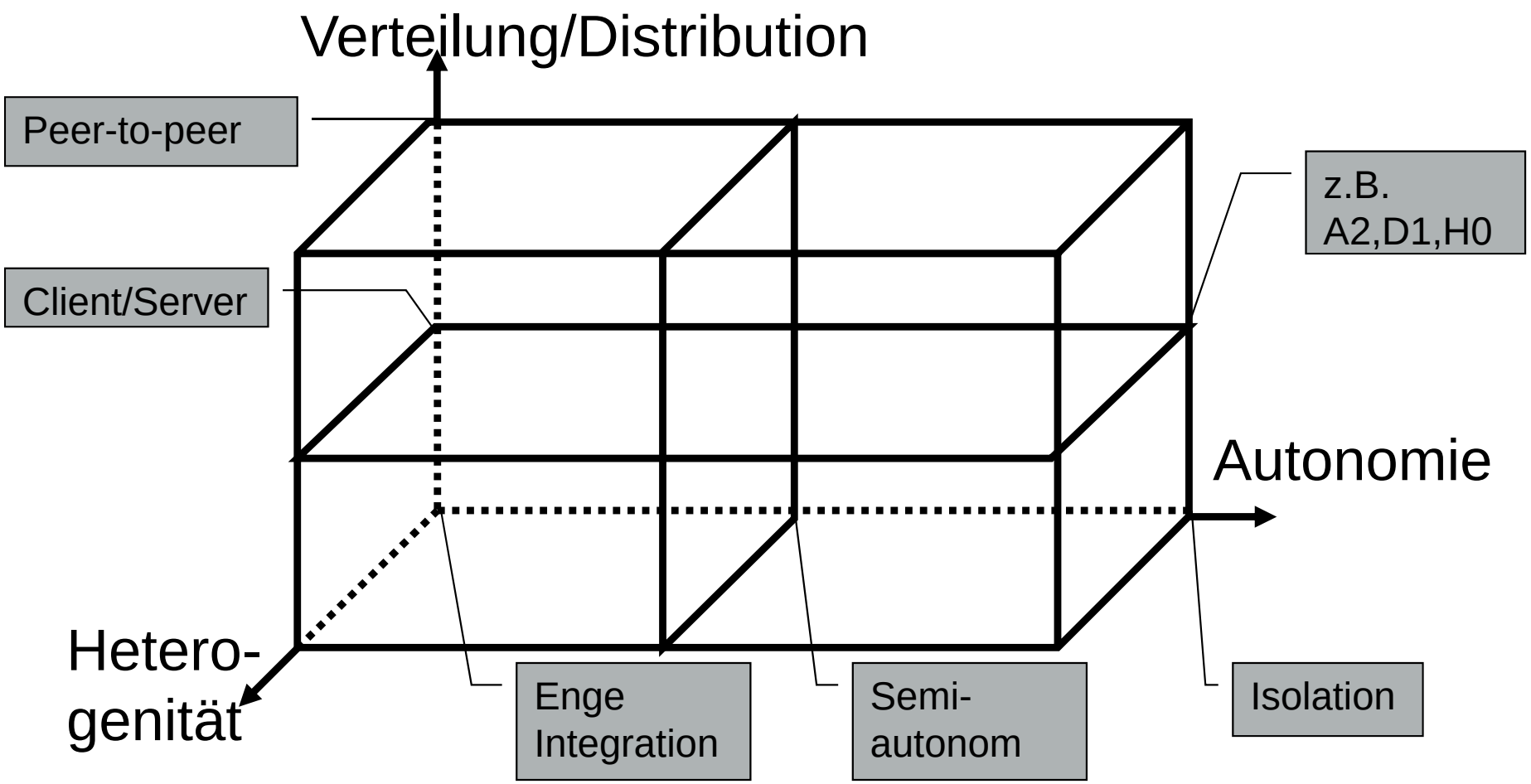
- Orthogonale Dimensionen
  - Verteilung
  - Autonomie
  - Heterogenität

- Orthogonal in der Lösung der jeweiligen Probleme
- Nicht unbedingt orthogonal in ihrer Ursache

Klassifikation von Informations-systemen nach [ÖV91]

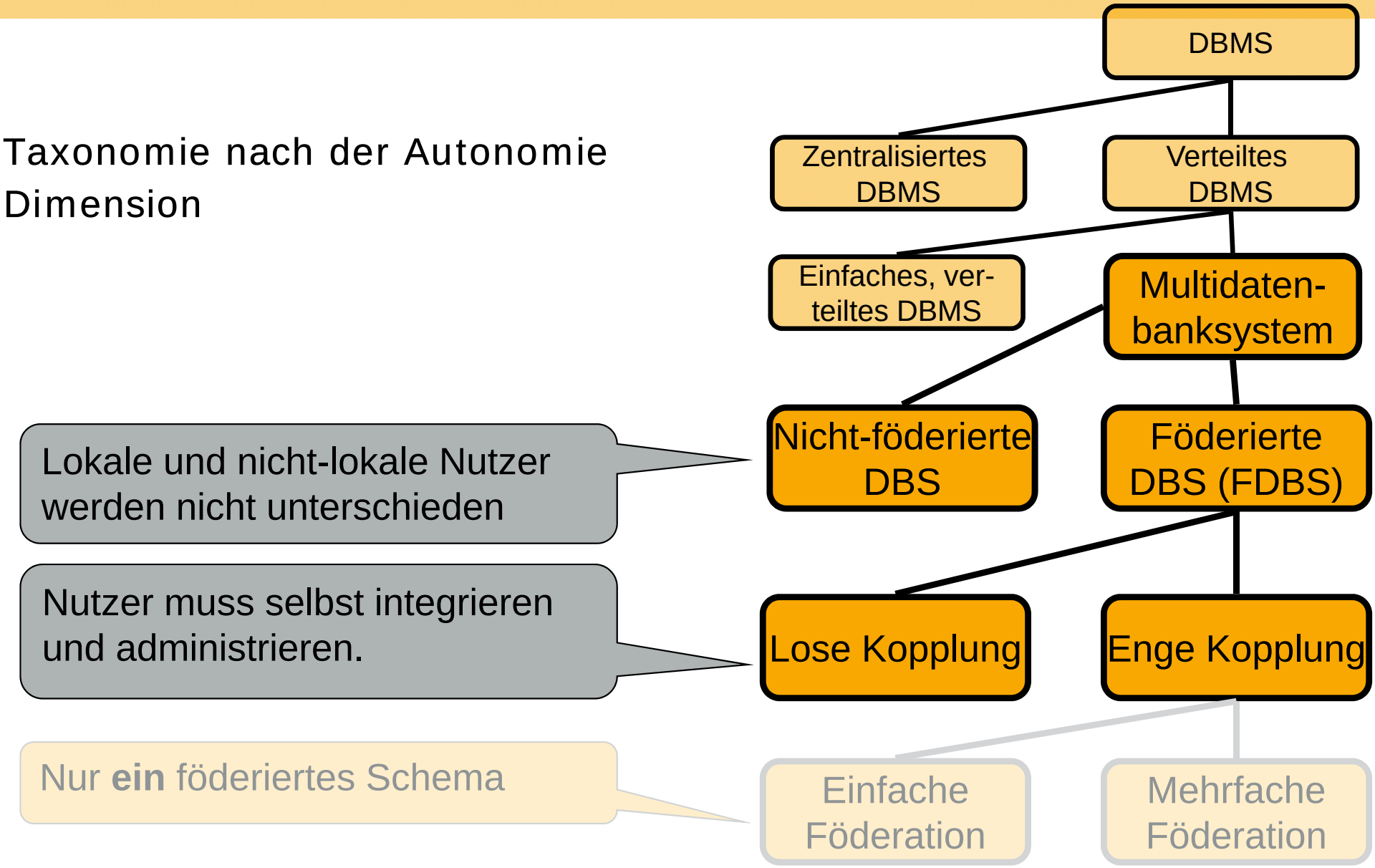


Erweiterung der Klassifikation nach [ÖV99]



Taxonomie nach [SL90]

Taxonomie nach der Autonomie Dimension



Verteilung (*Distribution*)

Ein verteiltes Informationssystem ist eine Sammlung mehrerer, logisch verknüpfter Informationssysteme, die über ein gemeinsames Netzwerk verteilt sind. [ÖV91]

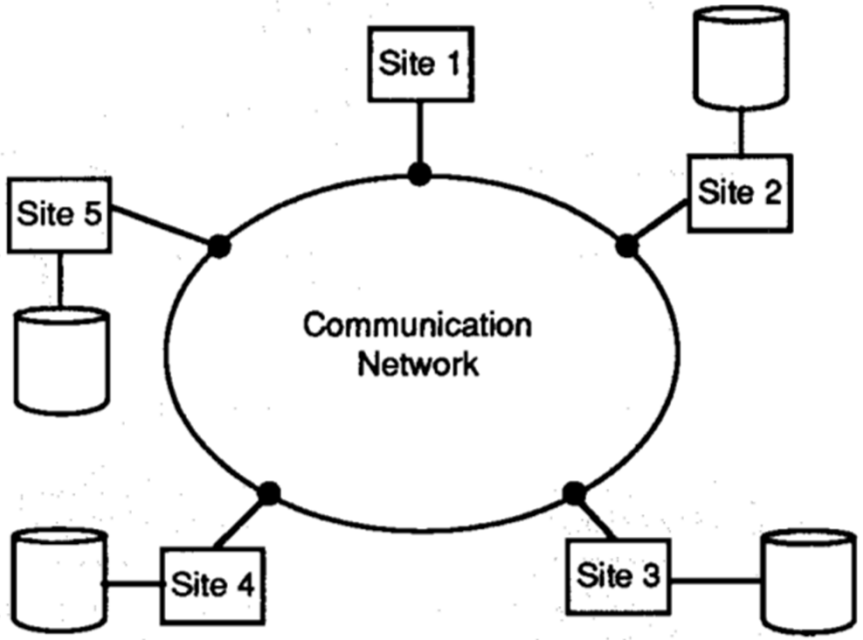


Figure 1.7 DDBS Environment

Physikalische Verteilung

- Motiviert durch Hardwareanforderungen (Hardwarebeschränkungen)
- Server stehen an unterschiedlichen Orten
  - Gleicher Raum, anderer Raum
  - Anderes Gebäude
  - Andere Stadt, anderes Land
- Shared Nothing
  - Server haben keine gemeinsamen, abhängigen Hardwarekapazitäten
    - Memory
    - Disk
    - CPU
  - Mit Ausnahme des Netzwerks
  - Im Gegensatz zu shared-disk und shared-memory

Logische Verteilung

- Motiviert durch Anwendungsanforderungen
  - Zuverlässigkeit
    - Bei Ausfall eines Servers
  - Verfügbarkeit
    - Bei Ausfall eines Netzwerkteils
  - Effizienz
- Redundanz
  - Replikation
  - Caching
- Partitionierung
  - Vertikal
  - Horizontal

Verteilung – Vor- und Nachteile

Vorteile aus Sicht der Quellen und des IIS

- Autonomie (gleich genauer)
- Performance: Kapazität dort, wo sie gebraucht wird
- Verfügbarkeit: Bei Ausfall eines Standorts
- Erweiterbarkeit
- Teilbarkeit (Verantwortung bei anderen Organisationseinheiten)

Nachteile aus Sicht des IIS

- Komplexität (Verwaltung, Optimierung)
- Kosten
- Sicherheit
- Autonomie

Autonomie (*Autonomy*)

Der Grad zu dem verschiedene DBMS unabhängig operieren können.

Bezieht sich auf Kontrolle, nicht auf Daten.

Klassen nach [ÖV99]

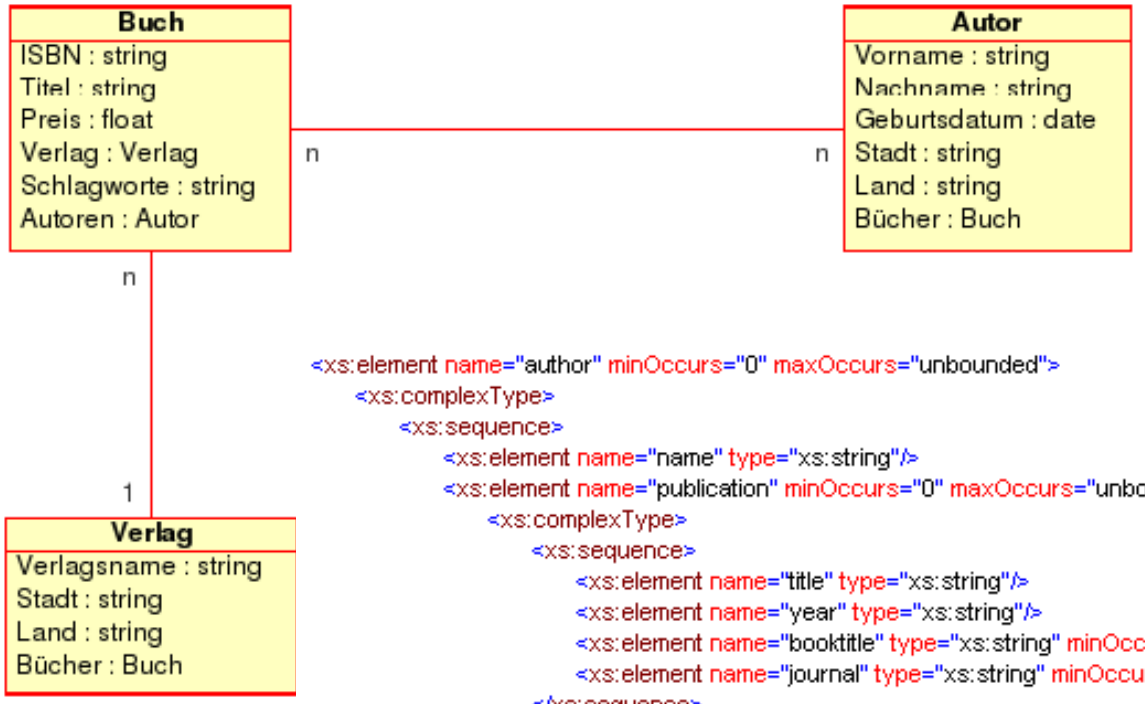
- Design-Autonomie
- Kommunikations-Autonomie
- Ausführungs-Autonomie

Design-Autonomie

- Auch: Entwurfsautonomie
- Freiheit des lokalen DBMS bezüglich
  - Datenmodell
    - Relational, hierarchisch, XML
  - Schema
    - Abdeckung der Domäne (*universe of discourse, miniworld*)
    - Grad der Normalisierung
    - Benennung
  - Transaktionsmanagement
    - Sperrprotokolle
- Freiheit dies jederzeit zu ändern.
  - Besonders problematisch!

Design-Autonomie – Beispiel

- Schema und Datenmodell 1
  - (Fast) relational
  - Flach
- Schema und Datenmodell 2
  - XML
  - hierarchisch



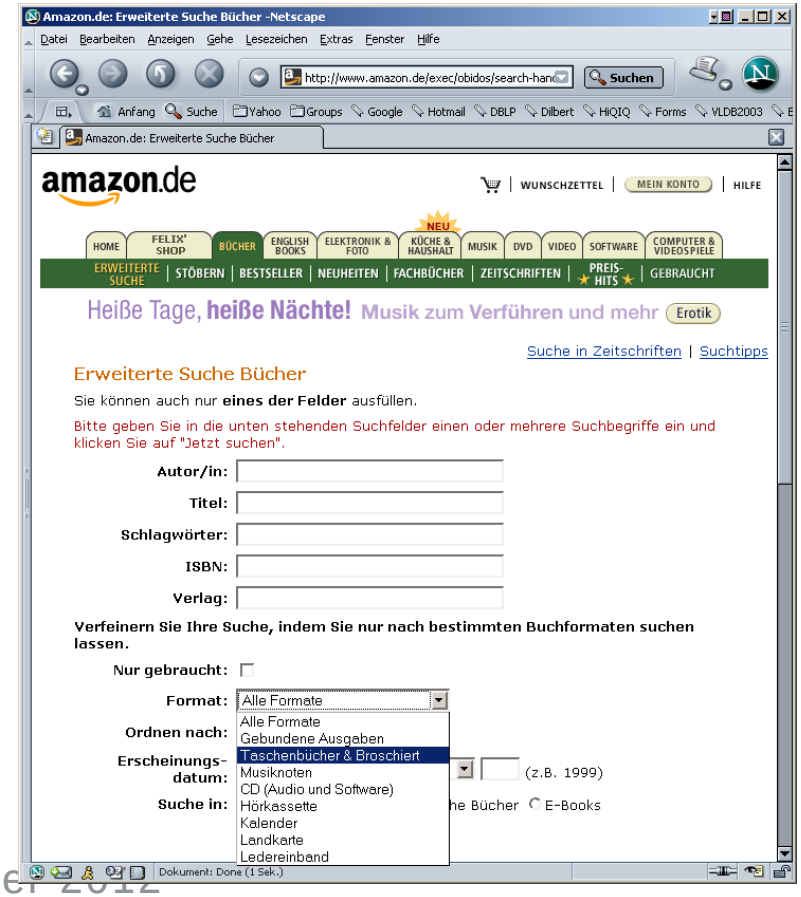

```
<xs:element name="author" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="publication" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="title" type="xs:string"/>
                        <xs:element name="year" type="xs:string"/>
                        <xs:element name="booktitle" type="xs:string" minOccurs="0"/>
                        <xs:element name="journal" type="xs:string" minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Kommunikations-Autonomie

- DBMS frei bezüglich
  - Wahl mit welchen Systemen kommuniziert wird
  - Wahl wann mit anderen Systemen kommuniziert wird
    - Jederzeit Eintritt/Austritt aus integriertem System
  - Wahl was (welcher Teil der Information) kommuniziert wird
  - Wahl wie mit anderen Systemen kommuniziert wird
    - Anfragesprache
  - Wahl welcher Teil der Anfragemöglichkeiten zur Verfügung gestellt werden
    - Prädikate
    - Sortierung
    - Write
    - ...

Kommunikations-Autonomie – Beispiel

- Extrem 1: Voller SQL Zugang
  - z.B. via JDBC
  - Transaktionen
  - Optimierung
  - Lesend (und Schreibend?)
  - Schemaveränderungen?
  - Antwort als Ergebnisrelation
- Extrem 2: HTML Formular
  - Nur ein (oder mehr) Suchfelder
  - Antwort als HTML Text
  - Nur Teile der Daten (public area)

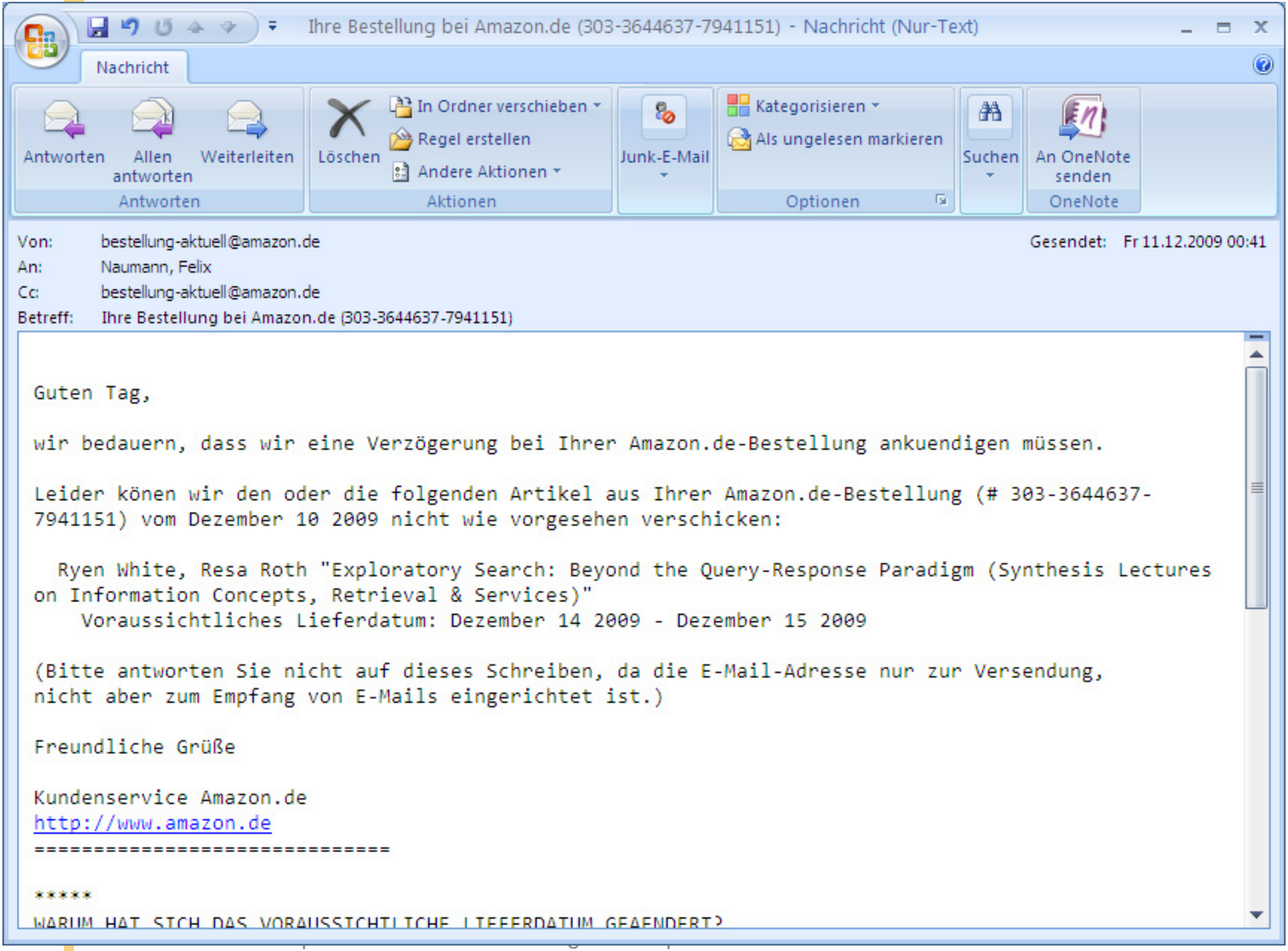


Ausführungs-Autonomie

- Informationssystem frei bezüglich
  - Wahl wann Anfragen ausgeführt werden
  - Wahl wie Anfragen ausgeführt werden
  - Wahl der Scheduling-Strategien
  - Wahl Optimierungs-Strategien
  - Wahl ob globale Transaktionen unterstützt werden

Ausführungs-Autonomie – Beispiel

- Optimierung und Scheduling
  - Behandlung externer vs. lokaler Anfragen
  - *Golden customers*
  - Garantierte Antwortzeiten
- Transaktionen
  - Dirty-read egal?
  - Amazon-Beispiel


Ihre Bestellung bei Amazon.de (303-3644637-7941151) - Nachricht (Nur-Text)
Nachricht
Antworten
Allen antworten
Weiterleiten
Antworten
Löschen
In Ordner verschieben
Regel erstellen
Andere Aktionen
Aktionen
Junk-E-Mail
Kategorisieren
Als ungelesen markieren
Optionen
Suchen
An OneNote senden
OneNote
Von: bestellung-aktuell@amazon.de
Gesendet: Fr 11.12.2009 00:41
An: Naumann, Felix
Cc: bestellung-aktuell@amazon.de
Betreff: Ihre Bestellung bei Amazon.de (303-3644637-7941151)
Guten Tag,
wir bedauern, dass wir eine Verzögerung bei Ihrer Amazon.de-Bestellung ankuendigen müssen.
Leider könen wir den oder die folgenden Artikel aus Ihrer Amazon.de-Bestellung (# 303-3644637-7941151) vom Dezember 10 2009 nicht wie vorgesehen verschicken:
Ryen White, Resa Roth "Exploratory Search: Beyond the Query-Response Paradigm (Synthesis Lectures on Information Concepts, Retrieval & Services)"
Voraussichtliches Lieferdatum: Dezember 14 2009 - Dezember 15 2009
(Bitte antworten Sie nicht auf dieses Schreiben, da die E-Mail-Adresse nur zur Versendung, nicht aber zum Empfang von E-Mails eingerichtet ist.)
Freundliche Grüße
Kundenservice Amazon.de
http://www.amazon.de
==========
*****
WARUM HAT SICH DAS VORAUSSICHTLICHE LIEFERDATUM GEAENDERT?

Autonomie → Heterogenität

- Verteilung als „Ursache“ für Autonomie
- Autonomie als Ursache für Heterogenität:
  - Autonome Systeme
  - ⇒ Gestaltungsfreiheit
  - ⇒ Unterschiedliche Entscheidungen
  - ⇒ Heterogenität

Heterogenität (*Heterogeneity*)

Heterogenität herrscht, wenn sich zwei miteinander verbundene Informationssysteme syntaktisch, strukturell oder inhaltlich unterscheiden.

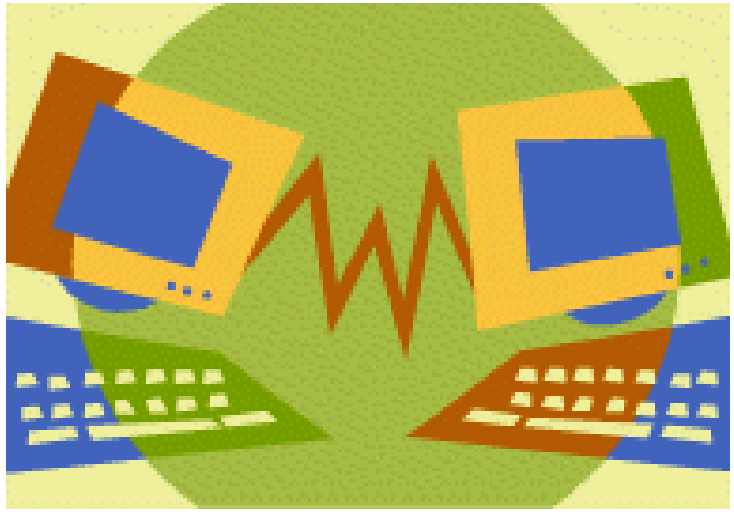
- Syntaktische Heterogenität
  - Auch: „Technische Heterogenität“
- Strukturelle Heterogenität
- Semantische Heterogenität

Die 3 S

Heterogenitäten zu überbrücken ist
die Kernaufgabe der Informationsintegration.

Syntaktische Heterogenität

- Hardware-Heterogenität
- Software-Heterogenität
- Schnittstellen-Heterogenität



Schnittstellen Heterogenität

- In einzelnen Systemen kein Problem
- Probleme für integrierte Systeme
  1. Globale Anfragesprache ist mächtiger als lokale Anfragesprache
     - Anfragen eventuell nicht ausführbar
     - Oder globales System muss kompensieren
  2. Lokale Anfragesprache ist mächtiger als globale Anfragesprache
     - Verpasste Chance, lokale (effiziente) Ausführung auszunutzen
  3. Gebundene und freie Variablen sind inkompatibel
     - Anfragen eventuell nicht ausführbar

Mächtige globale Anfragesprache

SQL

```
SELECT    * FROM Books
WHERE     Author = „Defoe“
AND       PubYear = 1979
```

HTML Form

Erweiterte Suche Bücher

Sie können auch nur **eines der Felder** ausfüllen.

Bitte geben Sie in die unten stehenden Suchfelder einen oder mehrere Suchbegriffe ein und klicken Sie auf "Jetzt suchen".

Autor/in:	
Titel:	
Schlagwörter:	
ISBN:	
Verlag:	

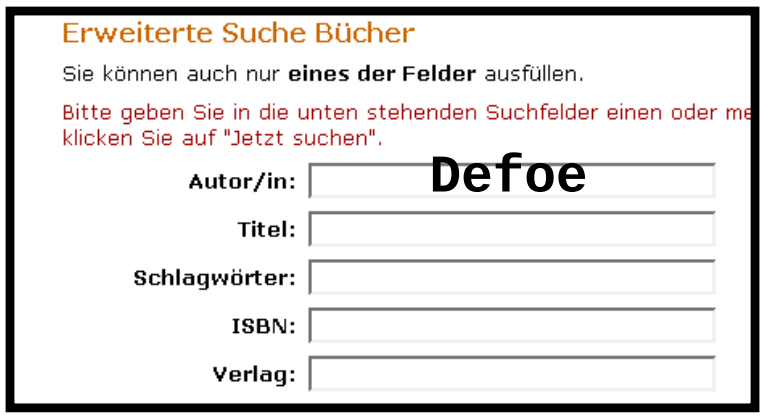
Mächtige globale Anfragesprache


```
SELECT * FROM Books
WHERE  Author = „Defoe“
AND    PubYear = 1979
```

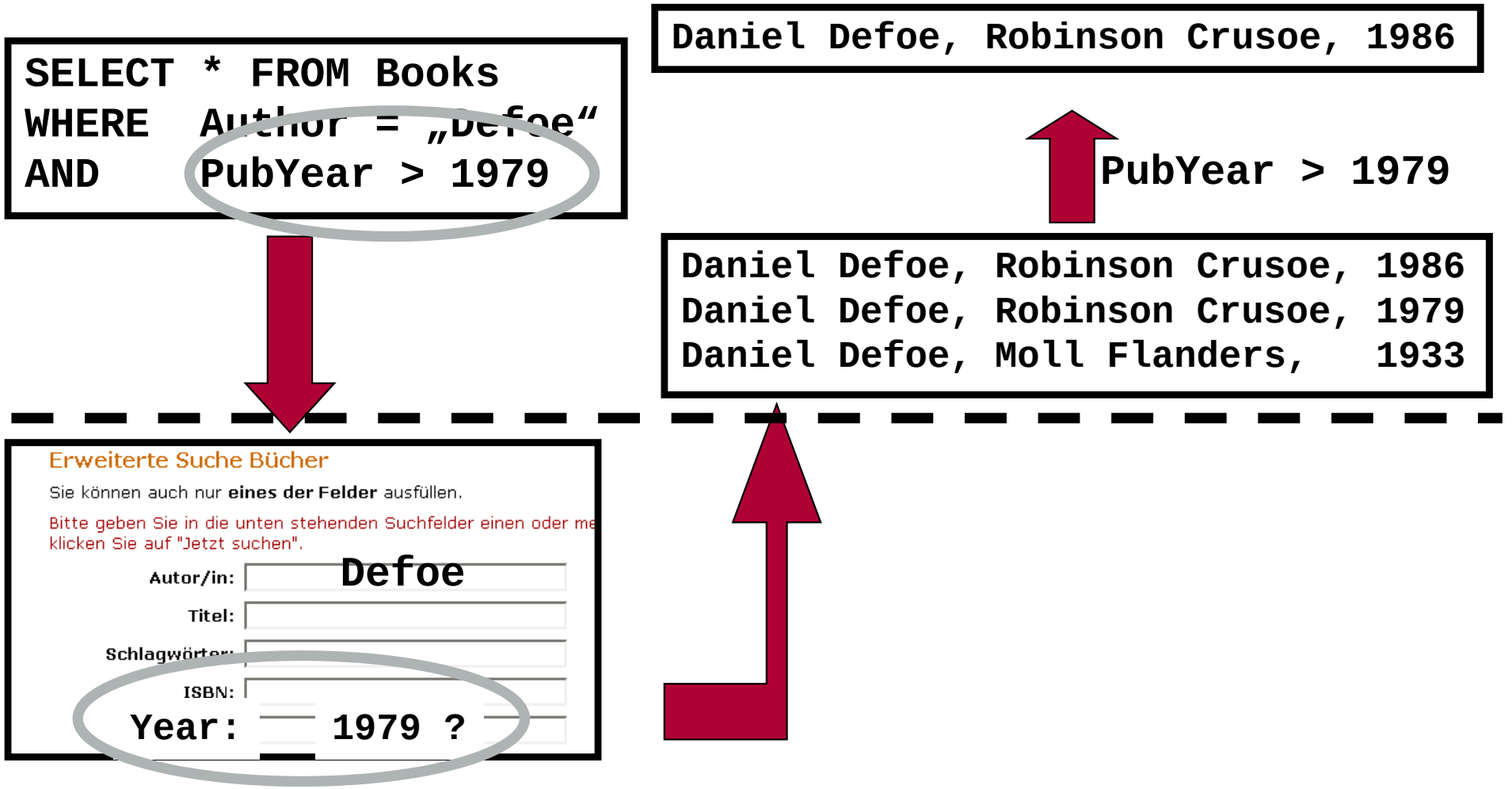
```
Daniel Defoe, Robinson Crusoe, 1979
```

PubYear = 1979

```
Daniel Defoe, Robinson Crusoe, 1986
Daniel Defoe, Robinson Crusoe, 1979
Daniel Defoe, Moll Flanders,   1933
```

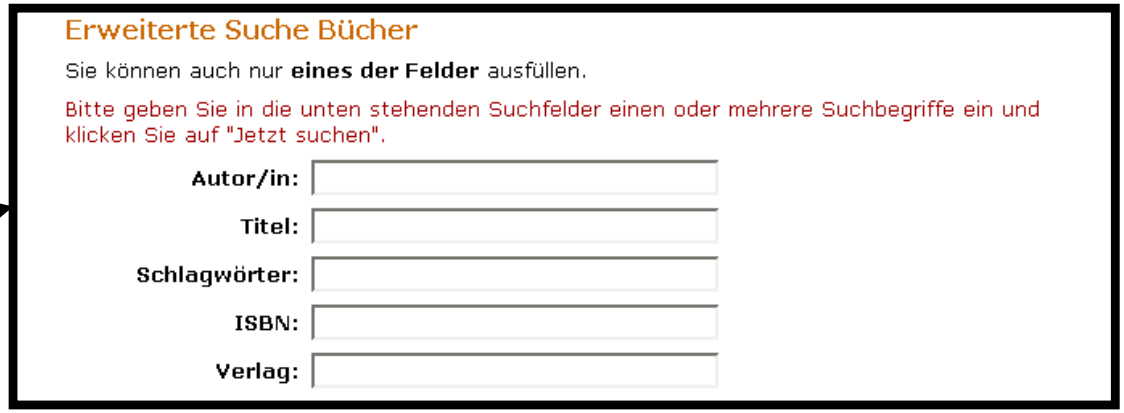


Mächtige globale Anfragesprache



Mächtige lokale Anfragesprache

HTML Form



SQL

```
SELECT    * FROM Books
WHERE     Author = „Defoe“
```

Verpasste Chancen.

Gebundene & Freie Variablen

- **Gebundene Variablen** müssen bei einer Anfrage gebunden werden.
  - z.B.: „Search“-Feld bei Google
- **Freie Variablen** müssen nicht gebunden werden.
  - z.B. „Autor“-Feld bei Amazon.de, falls Titel gebunden ist.

Gebundene & Freie Variablen – Beispiel & Ausblick

SONGS	Song	CD
	Friends	Life
	Friends	Love

CDs	CD	Künstler	Preis
	Love	Lucy	15
	Story	Snoopy	14

Künstler	CD	Künstler	Preis
	Story	Lucy	13
	Love	Snoopy	10
	Life	Charlie	8

Bastelaufgabe 1:
Wie teuer ist die billigste CD
mit einem Song namens “Friends”?

Quelle: [LC00]

Gebundene & Freie Variablen – Beispiel & Ausblick

SONGS	Song	CD
	Friends	Life
	Friends	Love

CDs	CD	Künstler	Preis
	Love	Lucy	15
	Story	Snoopy	14

Künstler	CD	Künstler	Preis
	Story	Lucy	13
	Love	Snoopy	10
	Life	Charlie	8

Unterstrichen = gebundene Variable

Bastelaufgabe 2:
Welches ist die billigste CD mit einem Song namens “Friends”, *die Sie anfragen können*?

Mehr später…

Syntaktische Heterogenität - Zusammenfassung

- Hardware Heterogenität
  - Bandbreite, CPU, ...
- Software Heterogenität
  - Protokolle, Sicherheit, ...
- Schnittstellen Heterogenität
  - Mächtigkeit der Anfragesprachen
  - Gebundene & freie Variablen

Strukturelle Heterogenität

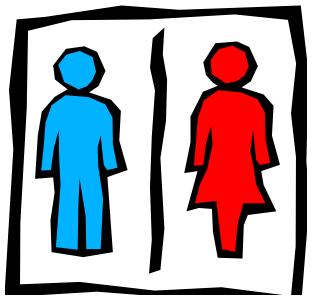
- Datenmodell-Heterogenität
  - Unterschiedliche Semantik
  - Unterschiedliche Struktur
- Schematische Heterogenität
  - Integritätsbedingungen, Schlüssel, Fremdschlüssel, etc.
  - Schema (Attribut vs. Relation etc.)
  - Struktur (Gruppierung in Tabellen)

Schematische Heterogenität

- Struktur
  - Modellierung
    - Relation vs. Attribut
    - Attribut vs. Wert
    - Relation vs. Wert
  - Benennung
    - Relationen
    - Attribute
    - Jeweils Homonyme und Synonyme
  - Normalisiert vs. Denormalisiert
  - Geschachtelt vs. Fremdschlüssel

- Diese Probleme sogar bei gleichem Datenmodell!

Schematische Heterogenität



Männer(Id, Vorname, Nachname)
Frauen(Id, Vorname, Nachname)

Relation vs. Attribut

Person(Id, Vorname, Nachname,männlich, weiblich)

Relation vs. Wert

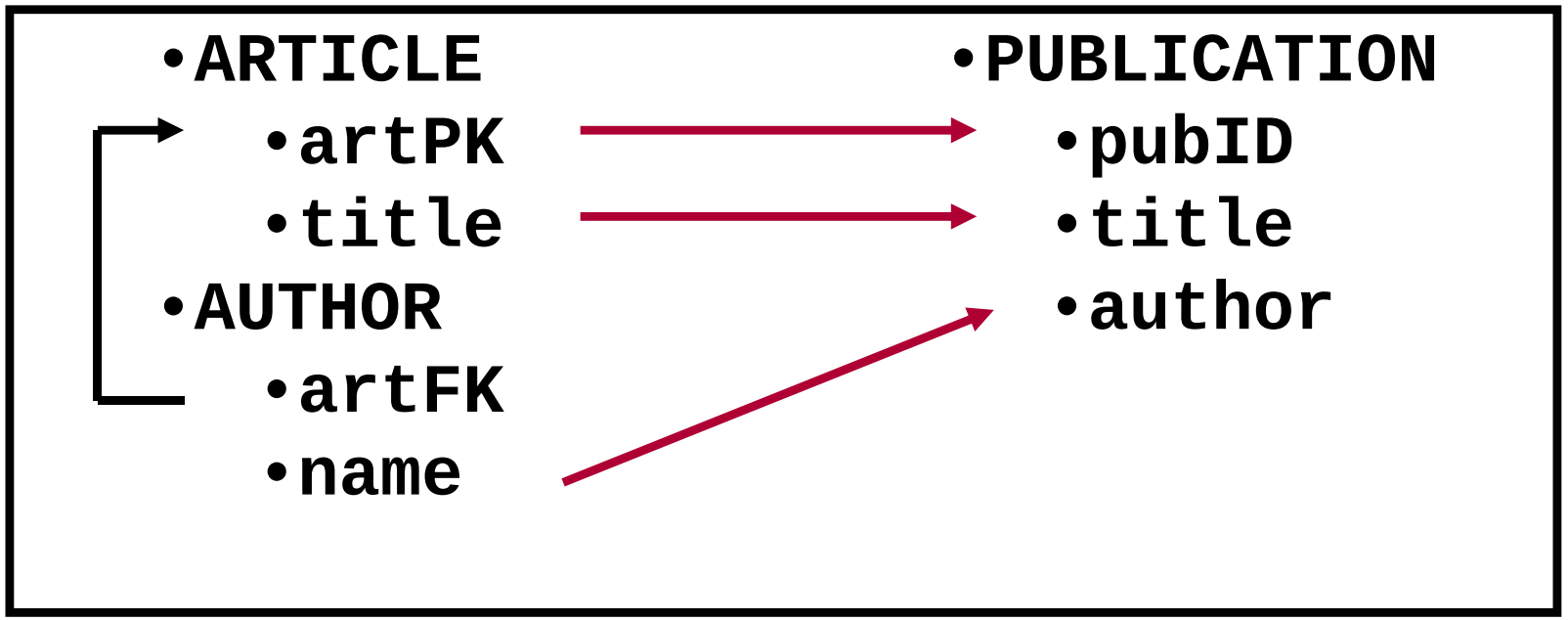
Attribut vs. Wert

Person(Id, Vorname, Nachname, Geschlecht)

Schematische Heterogenität - Beispiel

Normalisiert vs. Denormalisiert

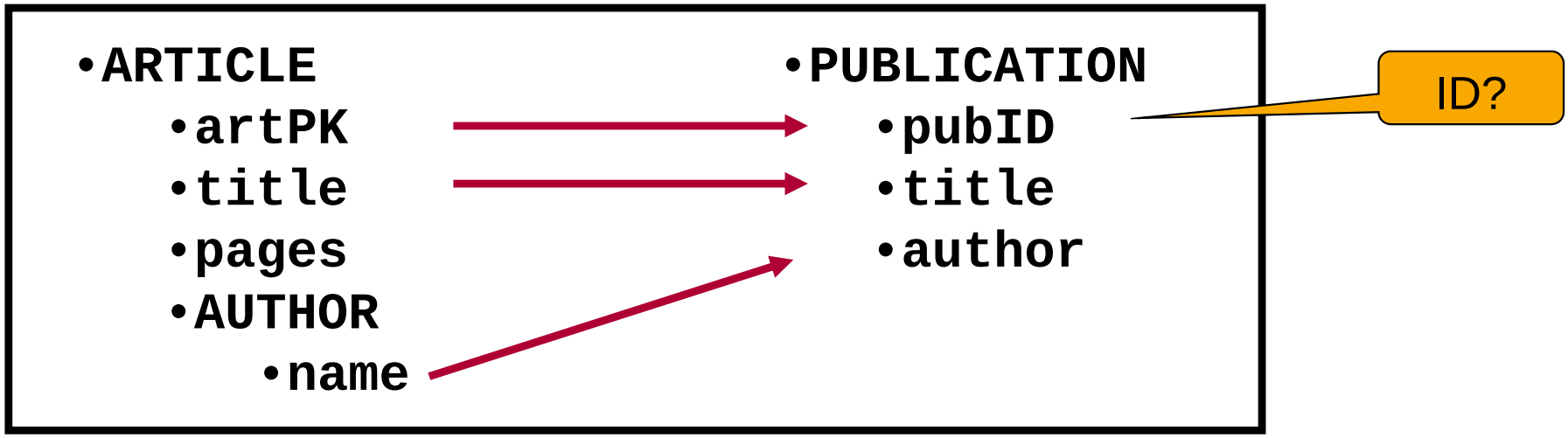
- 1:1 Assoziationen zwischen Werten wird unterschiedlich dargestellt
  - Durch Vorkommen im gleichen Tupel
  - Durch Schlüssel-Fremdschlüssel Beziehung



Schematische Heterogenität - Beispiel

Geschachtelt vs. Flach

- 1:n Assoziationen werden unterschiedlich dargestellt
  - Als geschachtelte Elemente
  - Als Schlüssel-Fremdschlüssel Beziehung



Semantik

Fremdwörterduden “Semantik”:

- Teilgebiet der Linguistik, das sich mit den Bedeutungen sprachlicher Zeichen und Zeichenfolgen befasst
- Bedeutung, Inhalt eines Wortes, Satzes oder Textes

„*Semantische Heterogenität ist ein überladener Begriff ohne klare Definition. Er bezeichnet die Unterschiede in Bedeutung, Interpretation und Art der Nutzung.*“ [ÖV91]

Semantik vs. Struktur

Strukturelle Heterogenität

- Betrifft Schemas
- Bedeutung der Labels im Schema egal
- Annahme bisher: Gleiche Label -> Gleiche Semantik

Männer(Id, Vorname, Nachname) **Frauen(Id, Vorname, Nachname)**	**A(Id, X, Y)** **B(Id, X, Y)**
Person(Id, Vorname, Nachname, Männlich, weiblich)	**P(Id, X, Y, a, b)**

Semantische Heterogenität

- Betrifft Daten
- Betrifft „Bedeutung“

Konzept

- Definition eines Konzepts
  - Noch nicht einmal hier sind sich immer alle einig.
  - Gen, Transaktion, Bestellung, Mitarbeiter
- Semantisch überlappende Weltausschnitte mit einander entsprechenden Klassen
- Korrespondenzarten zwischen Klassenextensionen:
  - $A = B$ Äquivalenz
  - $A \subseteq B$ Inklusion
  - $A \cap B$ Überlappung
  - $A \neq B$ Disjunktion

Wie viele Mitarbeiter hat IBM?

- Definition Mitarbeiter:
  - temporäre MA
  - Diplomanden
  - Berater
  - Studentische Mitarbeiter
  - Stellen oder Köpfe?
- Definition IBM
  - Welche Region? Welcher Geschäftsbereich?
  - Informix?
  - PWC?
- Welcher Zeitpunkt?
- Definition der Zählung:
  - Doppelte Zählung bei mehreren Anstellungen?

Wie viel Hardware haben wir an das HPI verkauft?

Zusammenfassung

- Verteilung
- Autonomie
  - Design-Autonomie
  - Kommunikations-Autonomie
  - Ausführungs-Autonomie
- Heterogenität
  - Syntaktische Heterogenität
    - Hardware Heterogenität
    - Software Heterogenität
    - Schnittstellen Heterogenität
  - Strukturelle Heterogenität
    - Datenmodell-Heterogenität
    - Schematische Heterogenität
  - Semantische Heterogenität
    - Konzepte & Konflikte

Überblick

- Überblick über Informationssysteme
  - Klassifikation
  - Materialisiert vs. virtuell
- Architekturen
  - 3 Schichten Architektur
  - 4 Schichten Architektur
  - 5 Schichten Architektur
- Mediator-Wrapper Architektur
  - Gio Wiederholds Definitionen
  - Konfigurationen
  - Mediatoren
  - Wrapper
- Peer-Data-Management
  - Architektur
  - Anwendungen

Integration

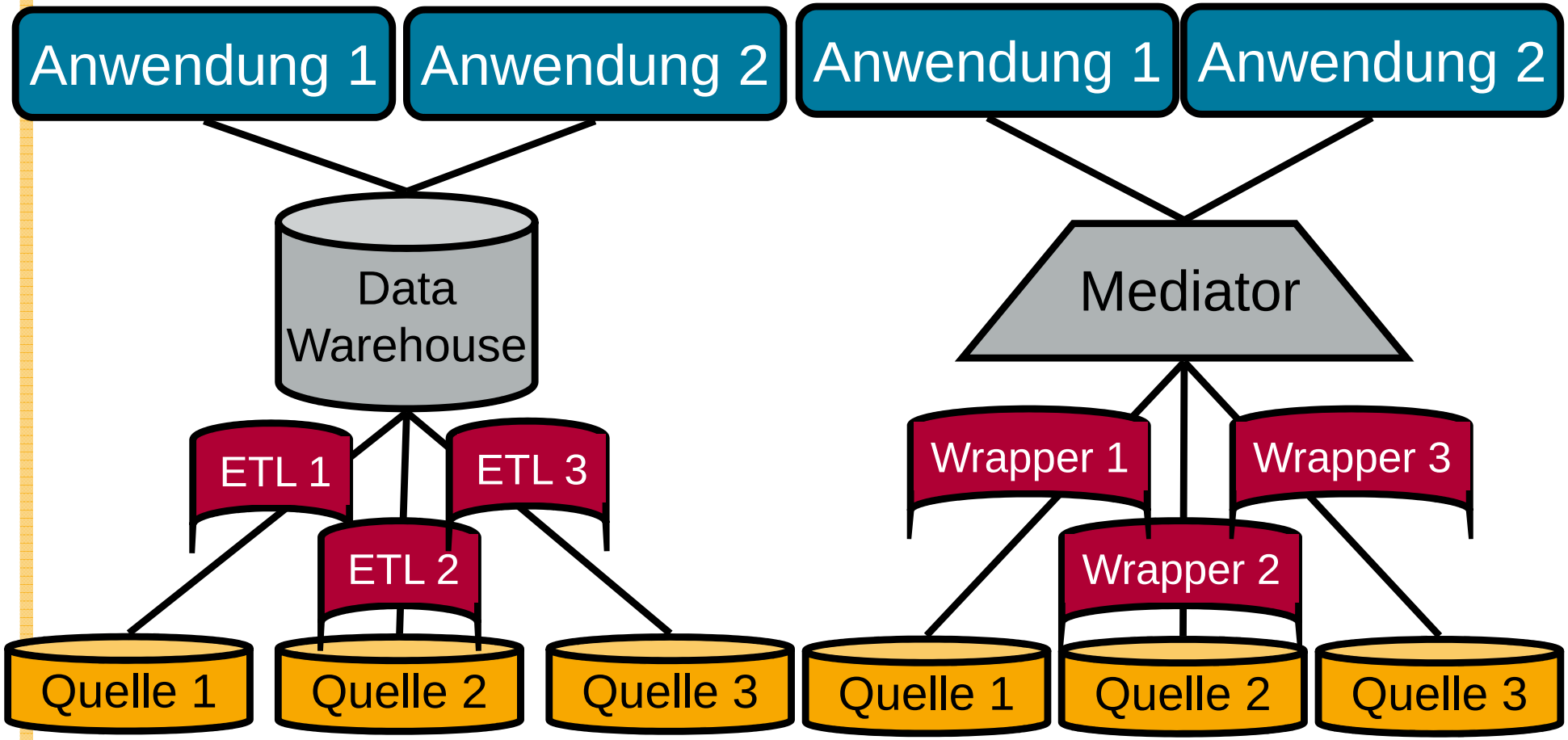
Materialisiert

- A priori Integration
- Zentrale Datenbasis
- Zentrale Anfragebearbeitung
- Typisches Beispiel: Data Warehouse

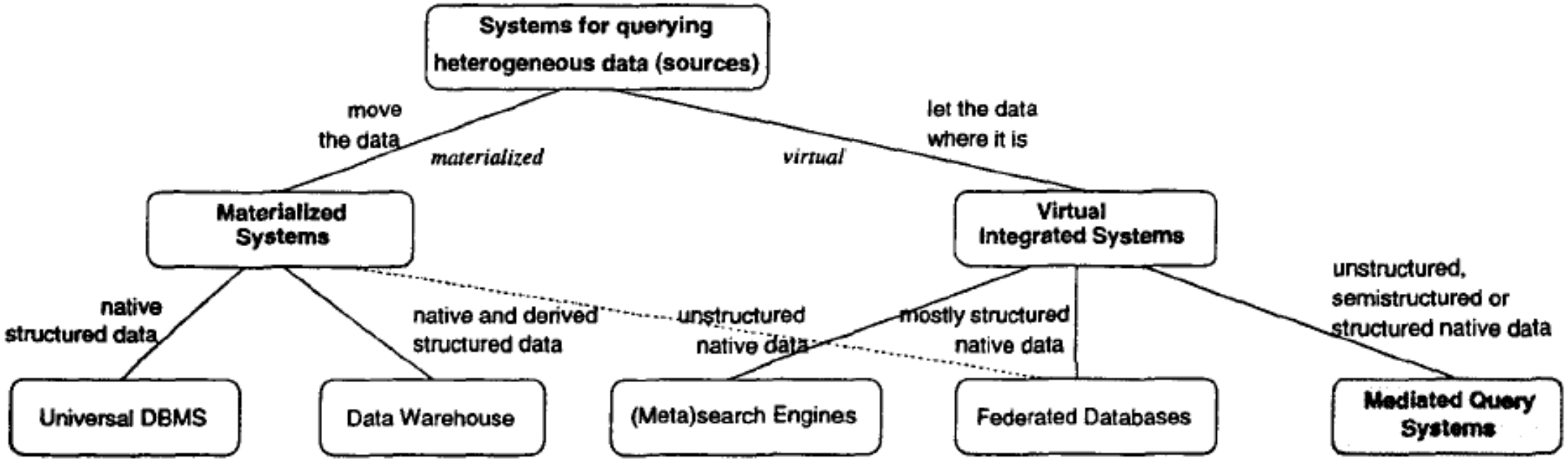
Virtuell

- On demand Integration
- Dezentrale Daten
- Dezentrale Anfragebearbeitung
- Typisches Beispiel: Mediator-basiertes Informationssystem

Data Warehouse vs. Mediator



Taxonomie nach [DD99]



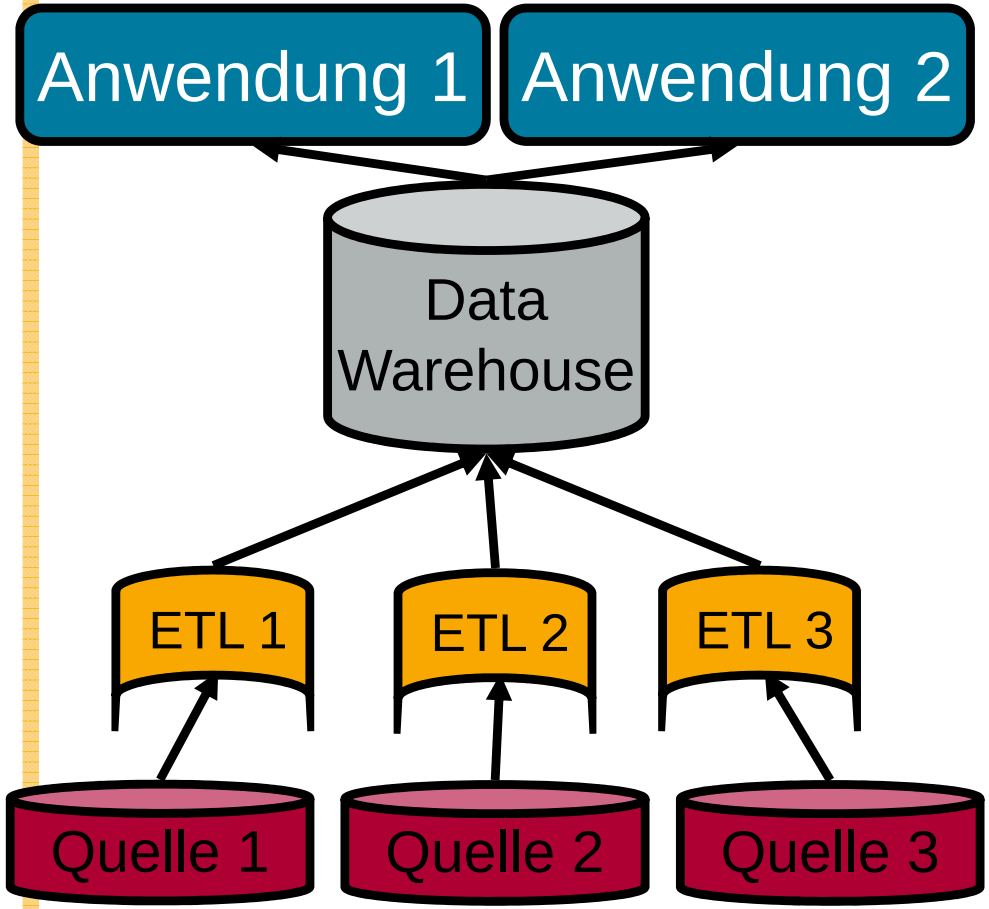
Data Warehouse vs. Mediator

Jetzt jeweils kurzer Überblick

- Datenfluss
- Anfragebearbeitung
- Entwurf und Entwicklung (Schema)

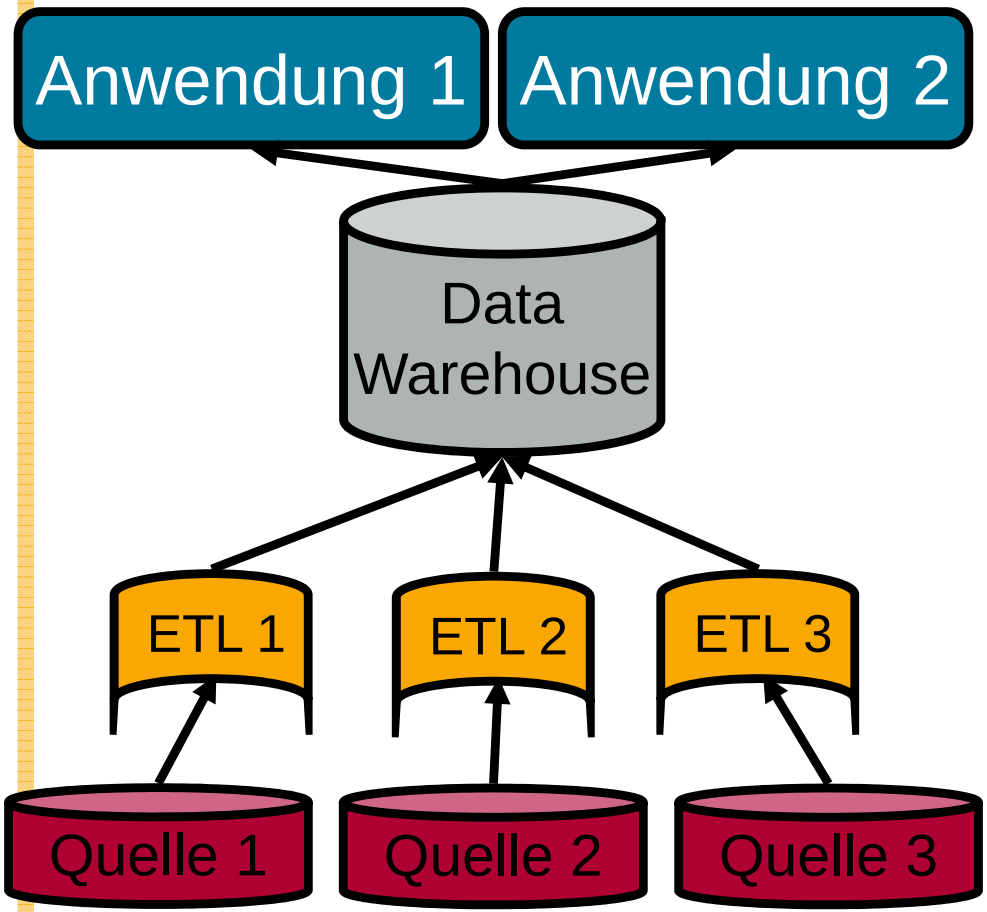
Details in den folgenden Wochen

Materialisierte Integration - Datenfluss



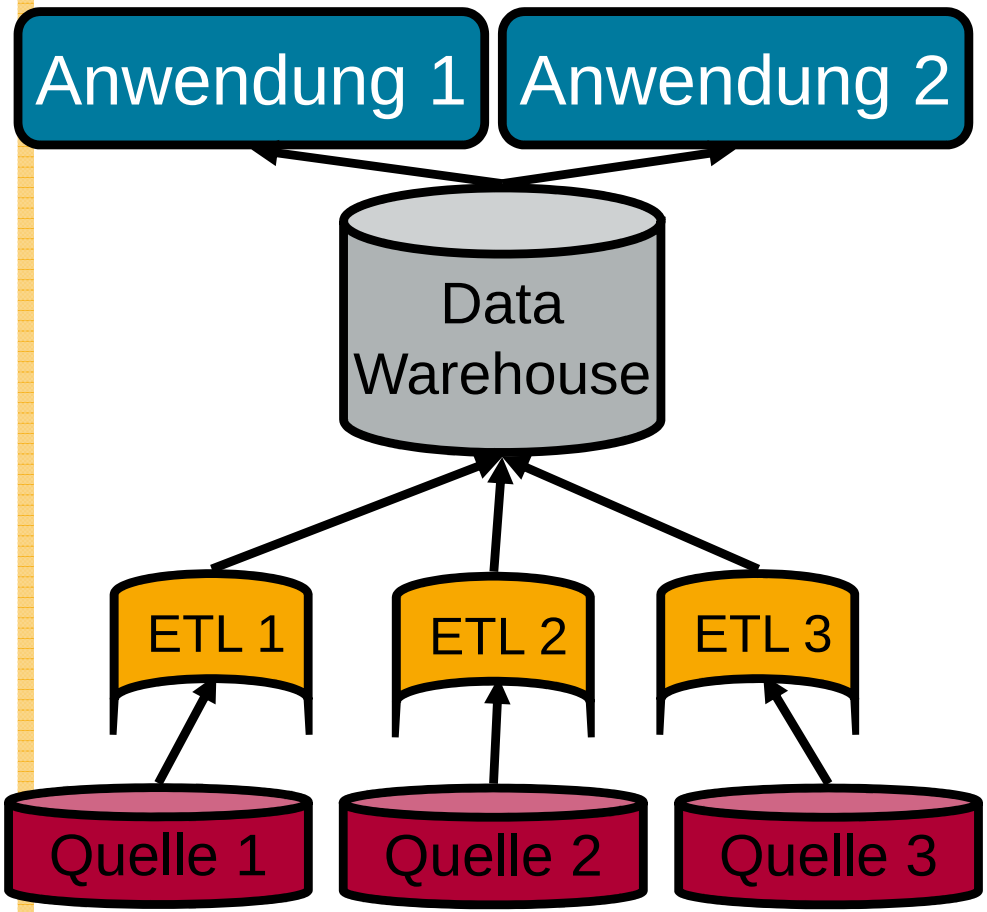
- Push
- Erstmalige „Bevölkerung“ (population) des DW
  - Data Cleansing
- Periodischer Datenimport
  - Stündlich / Täglich / Wöchentlich
  - Materialisierte Sichten / Sicht-Updates
- Redundante Datenhaltung
- Aggregation und Löschung alter Daten
  - Je älter, desto „aggregierter“

Materialisierte Integration - Anfragebearbeitung



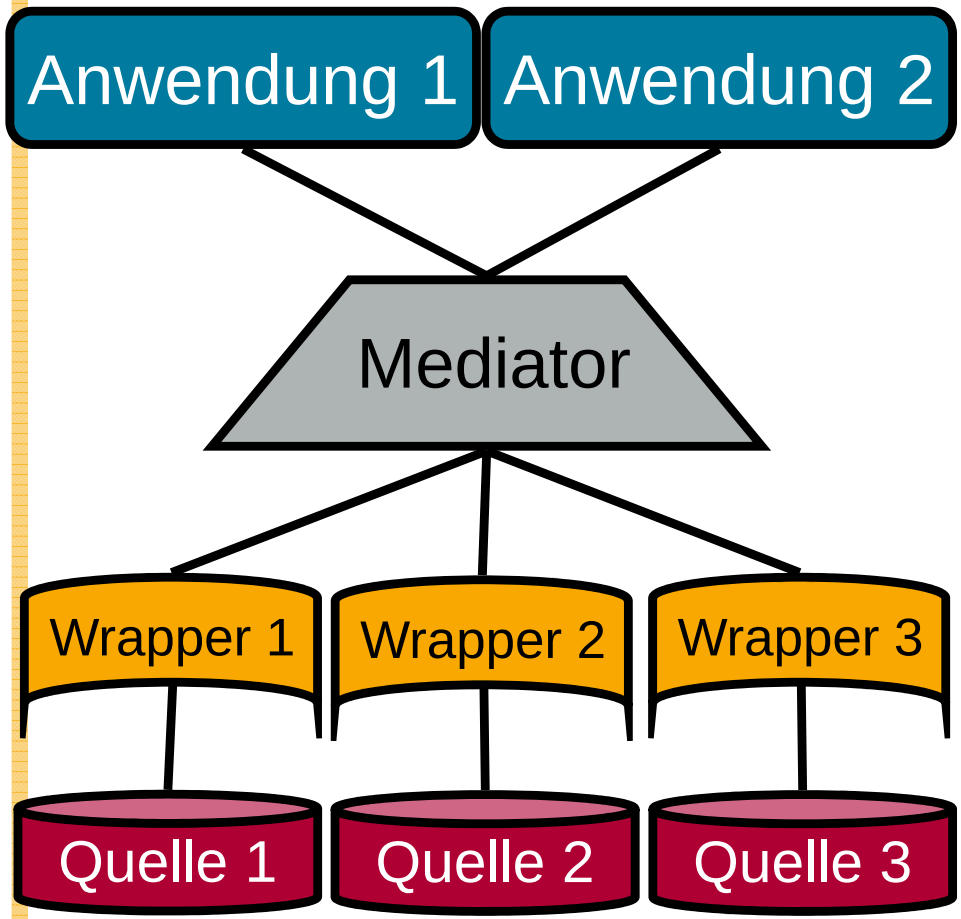
- Wie „normale“ DBMS
- Besonderheiten
  - Star Schema
  - Aggregation
  - Decision Support
- Siehe auch VL DWH

Materialisierte Integration - Schema



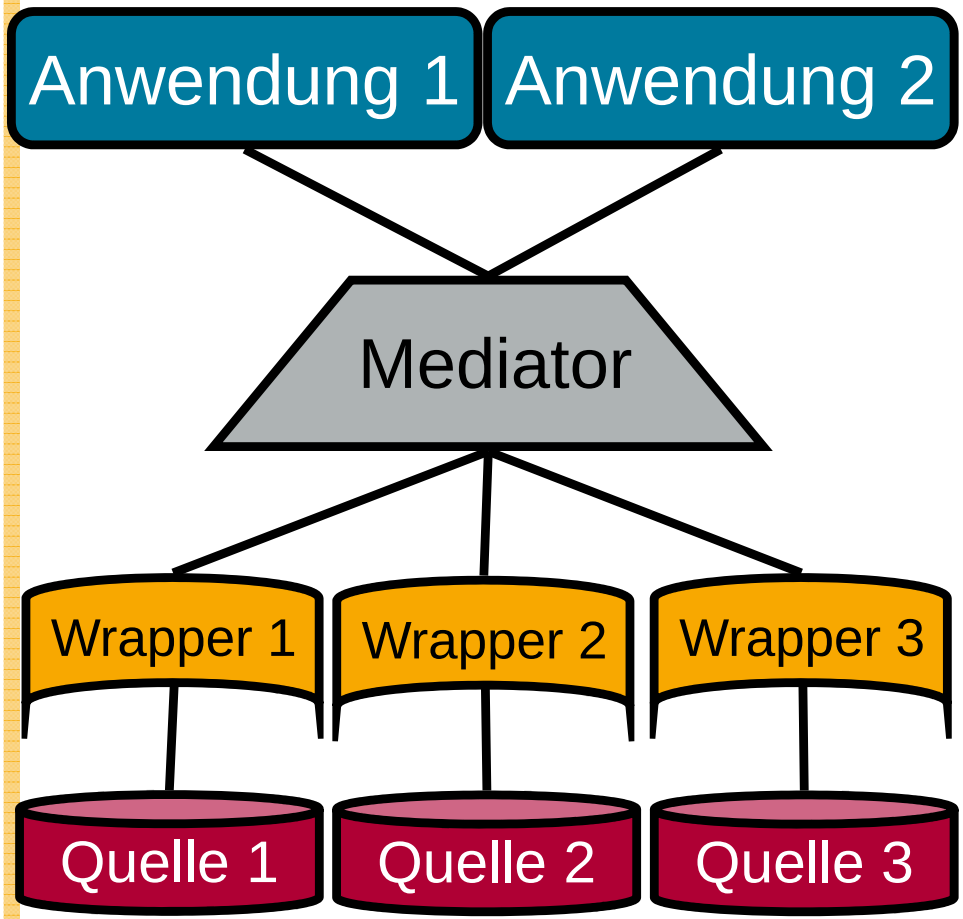
- Bottom-Up Entwurf
- Schemaintegration
- Star-Schema
  - *Fact-Table*
  - *Dimension Tables*

Virtuelle Integration - Datenfluss



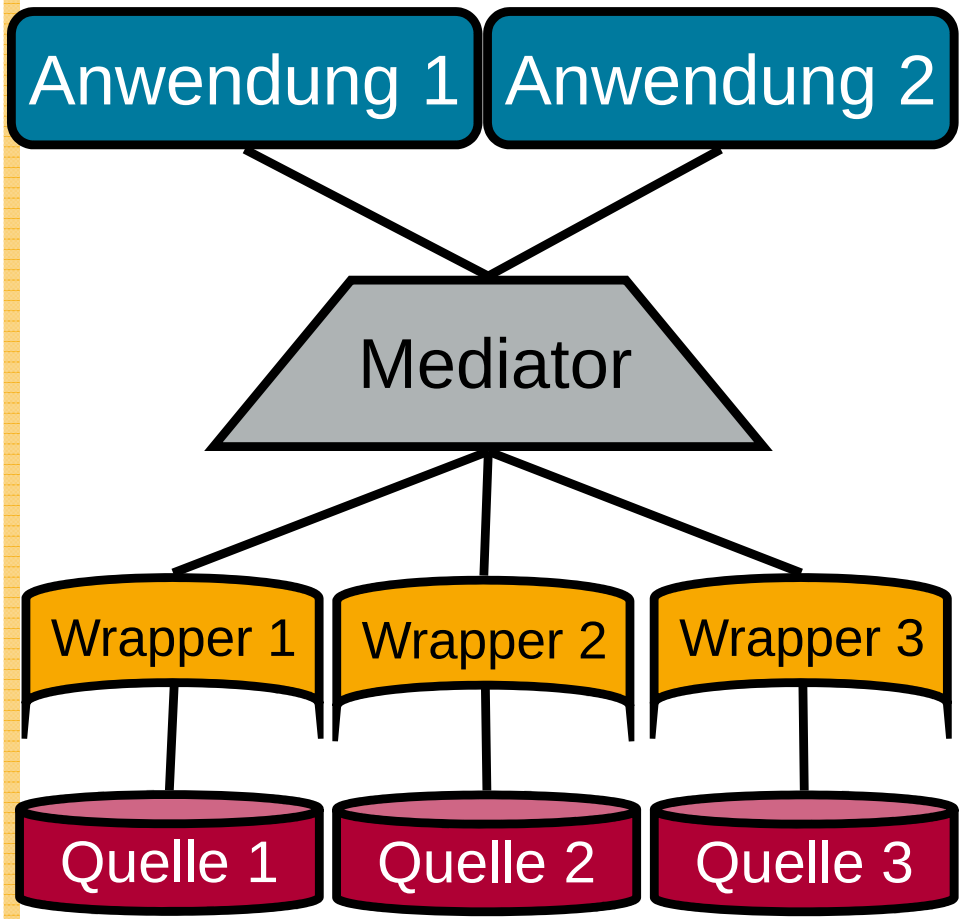
- Pull
- Daten sind in Quellen gespeichert.
- Nur die zur Anfragebeantwortung notwendigen Daten werden übertragen.
- Data Cleansing nur online möglich.

Virtuelle Integration - Anfragebearbeitung



- Optimierung schwierig
  - Fähigkeiten der Quellen
  - Geschwindigkeit der Quellen
- Viele mögliche Pläne
  - Redundante Quellen
  - Redundante Pläne
- Dynamisch, um ausfallende Quellen auszugleichen

Virtuelle Integration - Schema



- Top-Down Entwurf
- Leicht erweiterbar
  - Global: Neue Quellen suchen
  - Lokal: Nur ein *mapping* verändern.
- Schema Mapping statt Schema-integration

Zusammenfassung Vor- und Nachteile

	Materialisiert	Virtuell
Aktualität	- (Cache)	+
Antwortzeit	+	-
Flexibilität	- (GaV)	+ (LaV)
Komplexität	-	--
Autonomie	-	+
Anfragemächtigkeit	+	-
Read/Write	+/+	+/-
Größe	-	+
Ressourcenbedarf	? (workload)	? (workload)
Vollständigkeit	+	? (OWA, CWA)
Datenreinigung	+	-
Informationsqualität	+	-

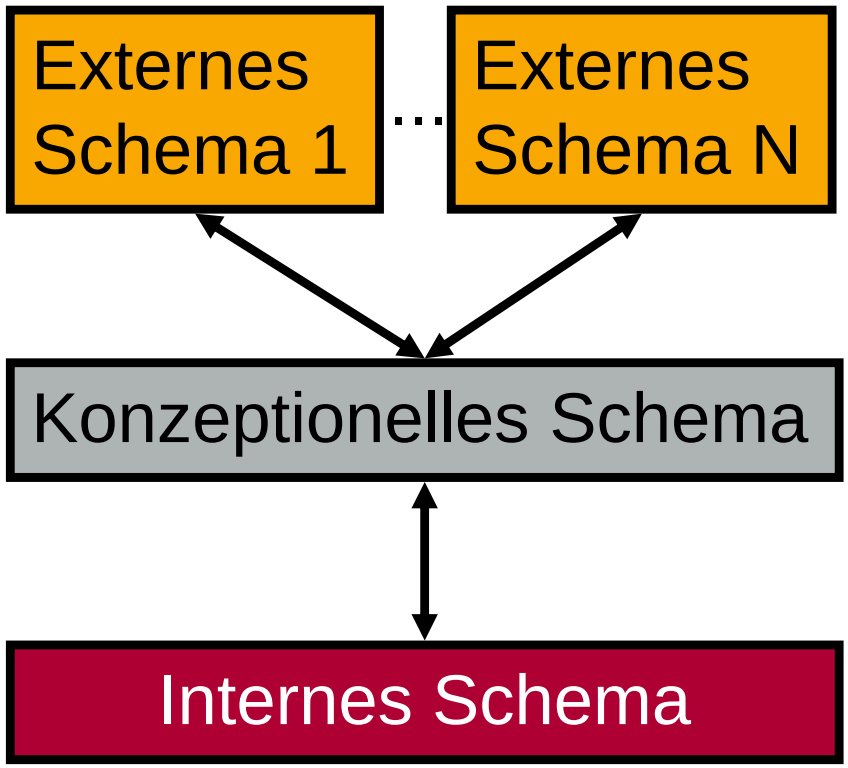
Überblick

- Überblick über Informationssysteme
  - Klassifikation
  - Materialisiert vs. virtuell
- Architekturen
  - 3 Schichten Architektur
  - 4 Schichten Architektur
  - 5 Schichten Architektur
- Mediator-Wrapper Architektur
  - Gio Wiederholds Definitionen
  - Konfigurationen
  - Mediatoren
  - Wrapper
- Peer-Data-Management
  - Architektur
  - Anwendungen

3-Schichten Architektur

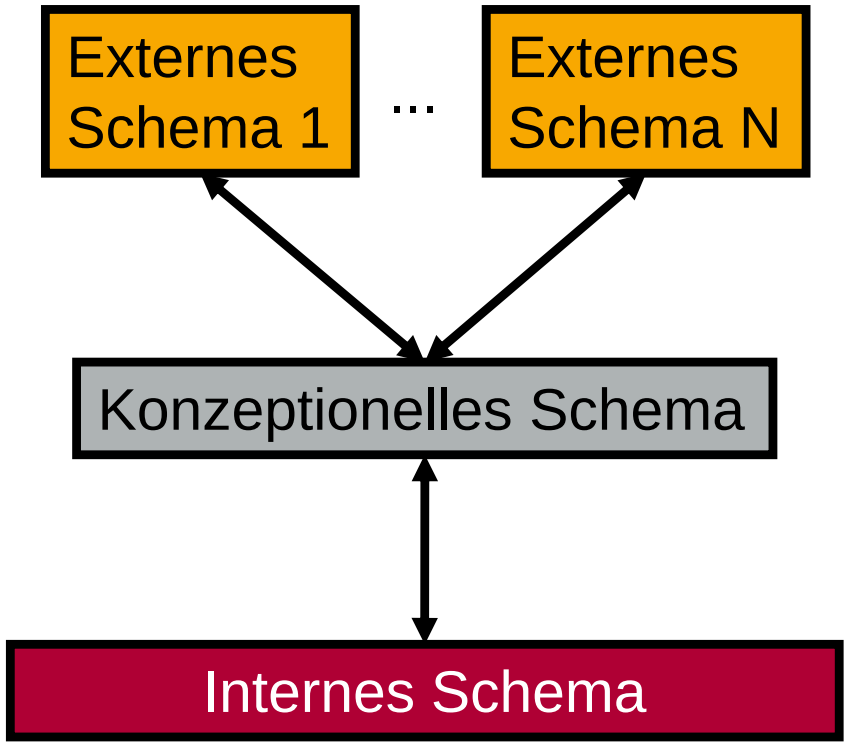
ANSI/SPARC

3-Schichten Architektur für zentralisierte DBMS

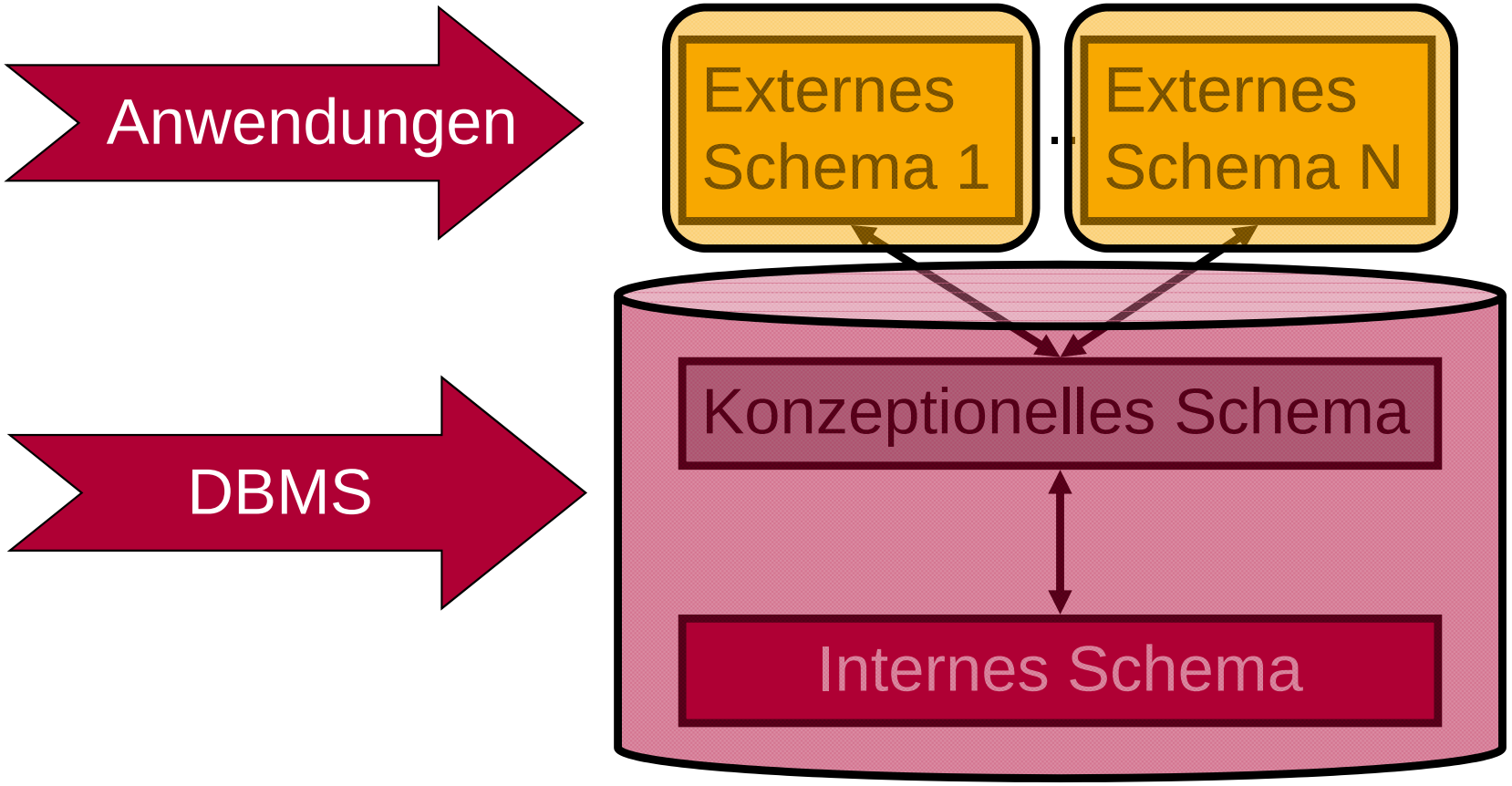


Wdh: Das Schichtenmodell

- Interne (physische) Sicht
  - Speichermedium (Tape, Festplatte)
  - Speicherort (Zylinder, Block)
- Konzeptionelle (logische) Sicht
  - Unabhängig von physischer Sicht
  - Definiert durch Datenmodell
  - Stabiler Bezugspunkt für interne und externe Sichten
- Externe (logische) Sicht
  - Anwendungsprogramme
  - Nur auf die relevanten Daten
  - Enthält Aggregationen und Transformationen

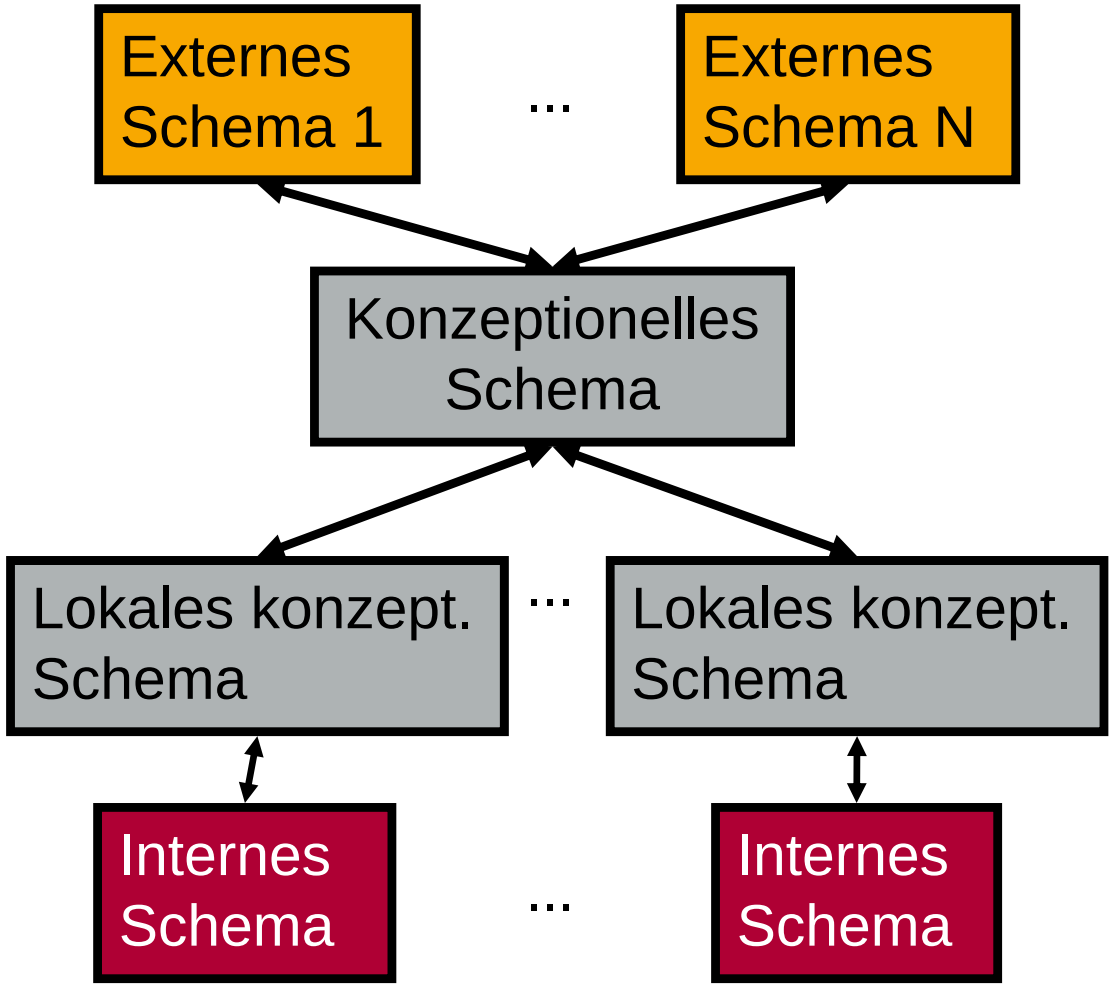


3-Schichten Architektur

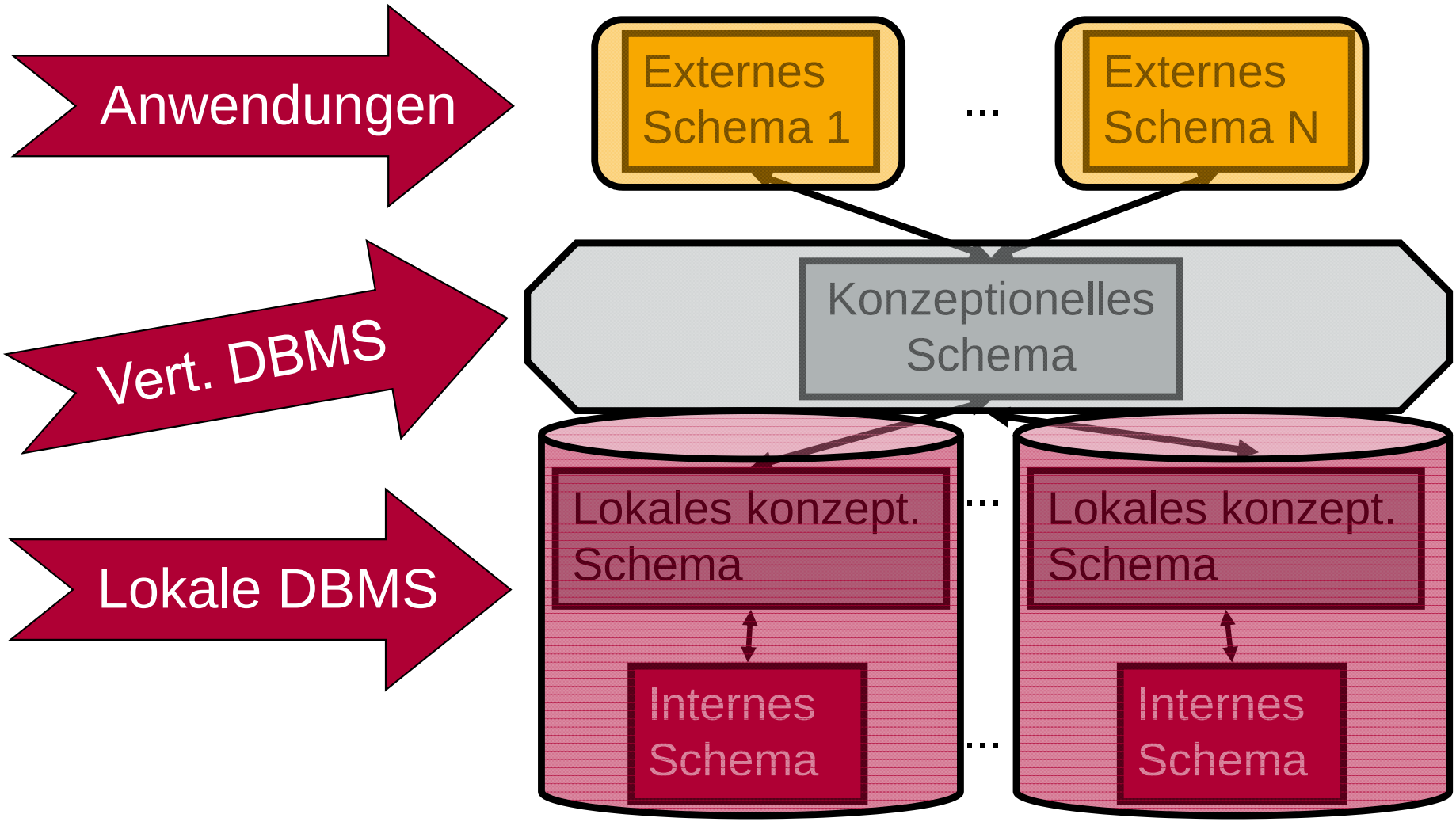


4-Schichten Architektur

- Für verteilte DBMS
- Neu: Trennung lokales vs. globales konzeptionelles
- Globales Konzeptionelles Schema ist integriert aus den lokalen konzeptionellen Schemas.
- Lokales und globales konzept. Schema kann gleich sein.

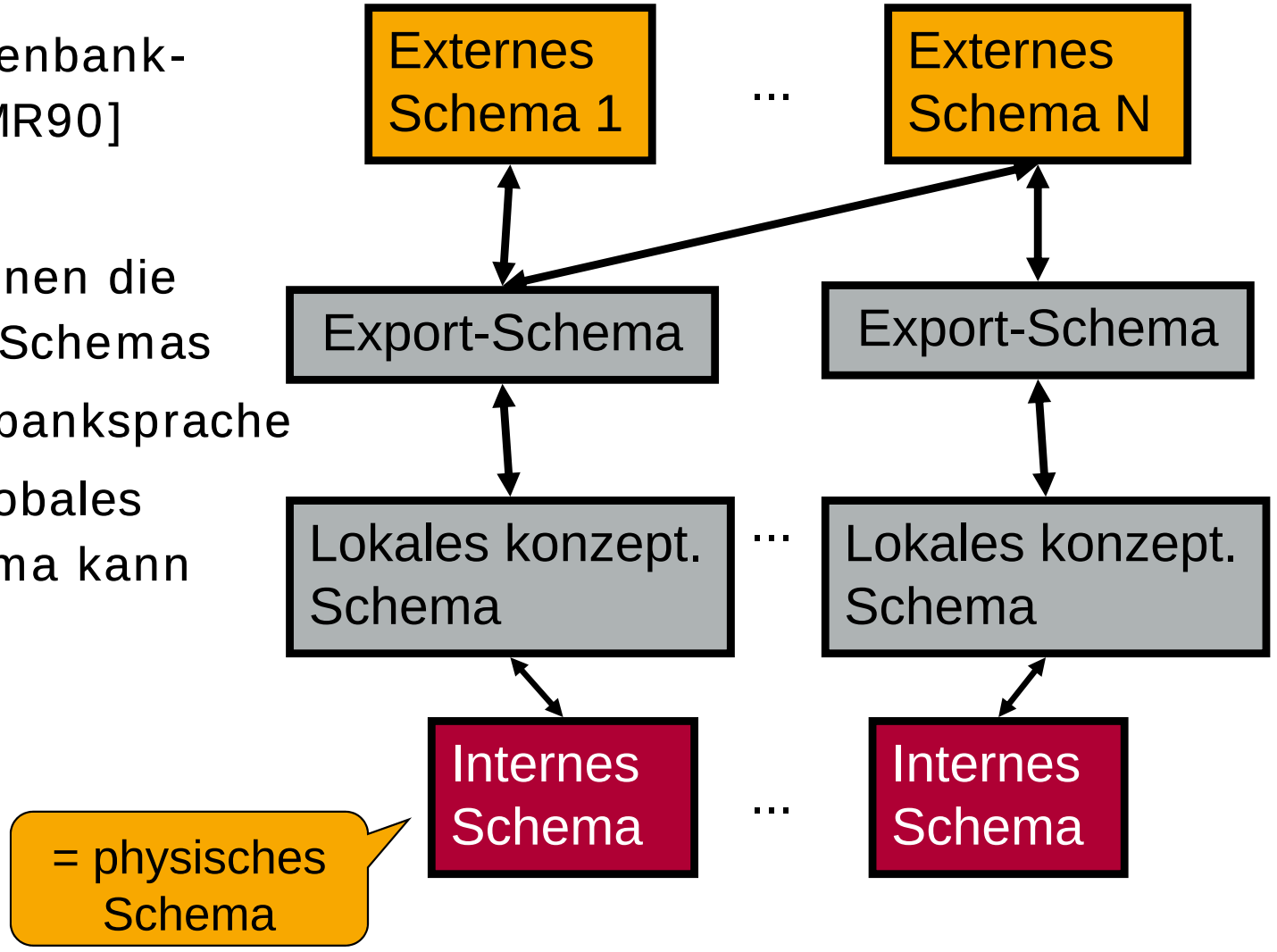


4-Schichten Architektur

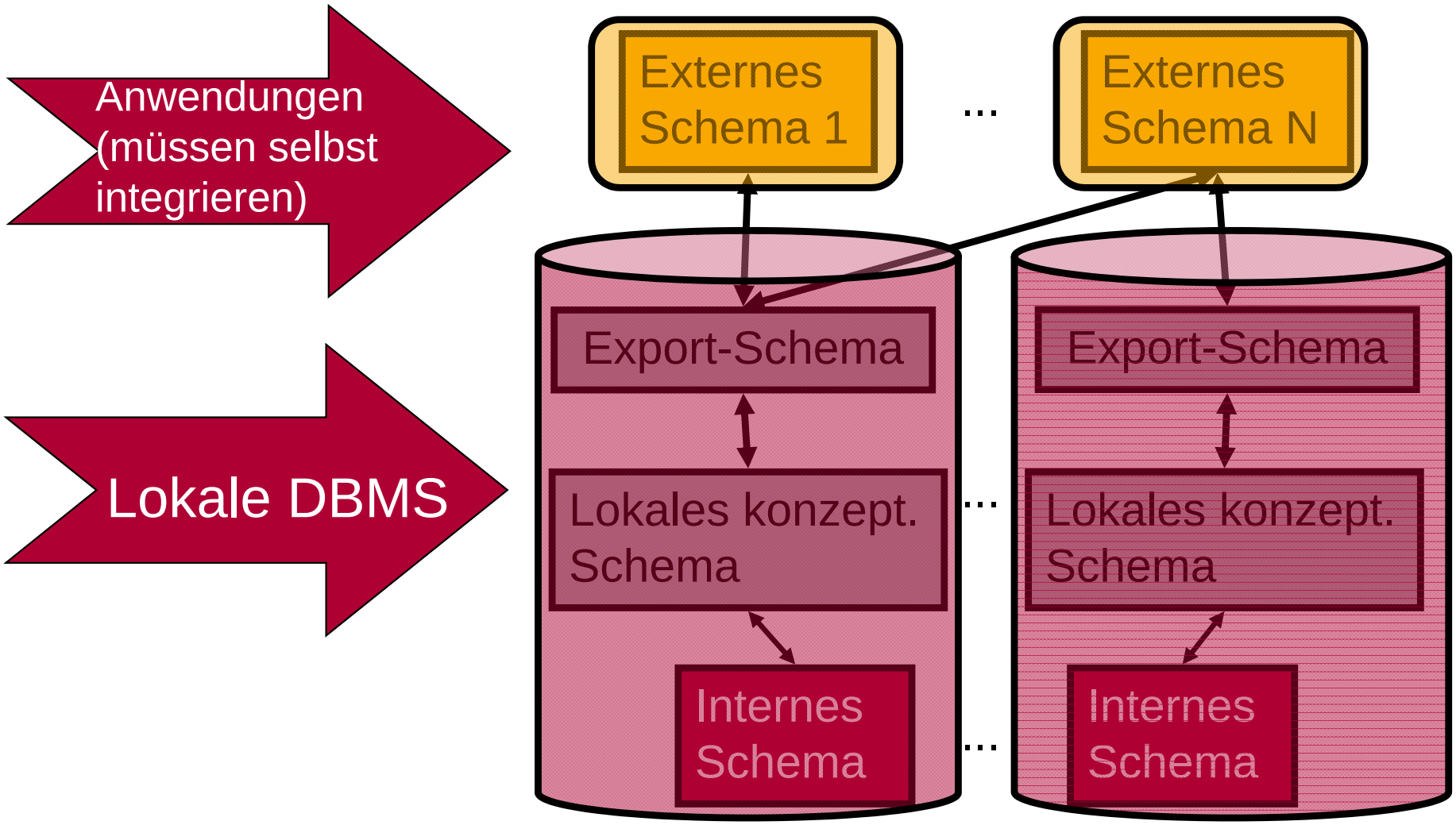


4-Schichten Architektur

- Auch: Multidatenbank-architektur [LMR90]
- Voraussetzung
  - Nutzer kennen die jeweiligen Schemas
  - Multidatenbanksprache
- Lokales und globales konzept. Schema kann gleich sein.
- Lose Kopplung

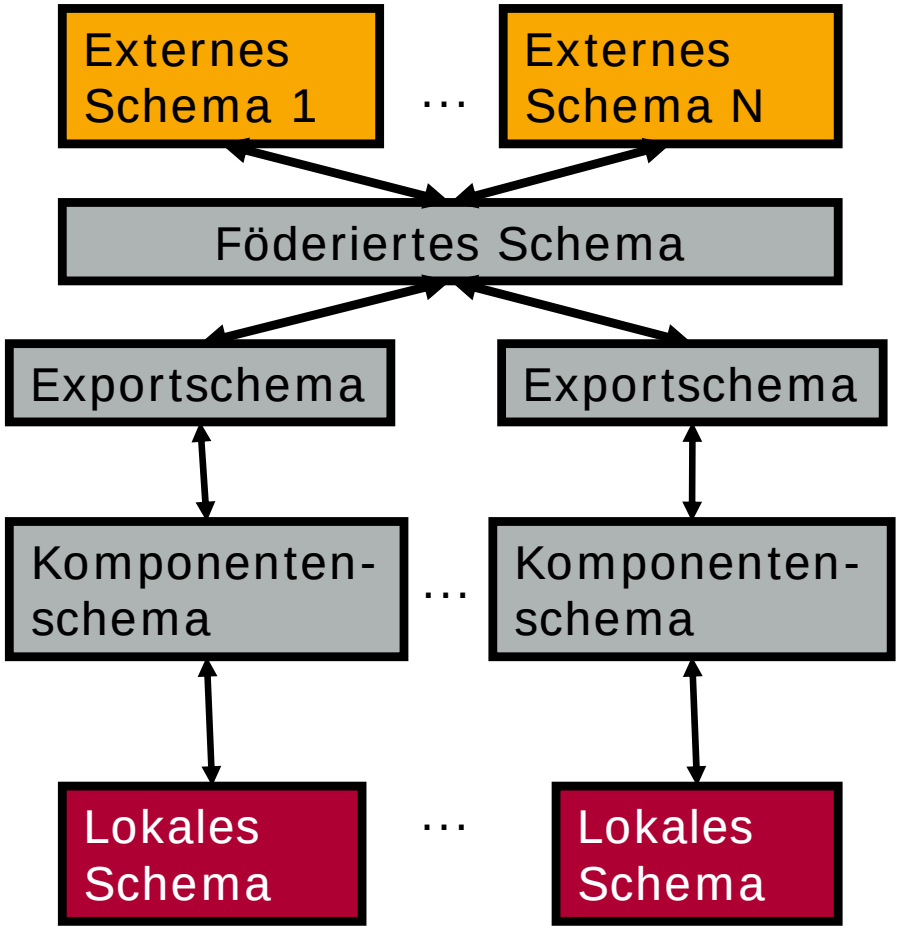


4-Schichten Architektur

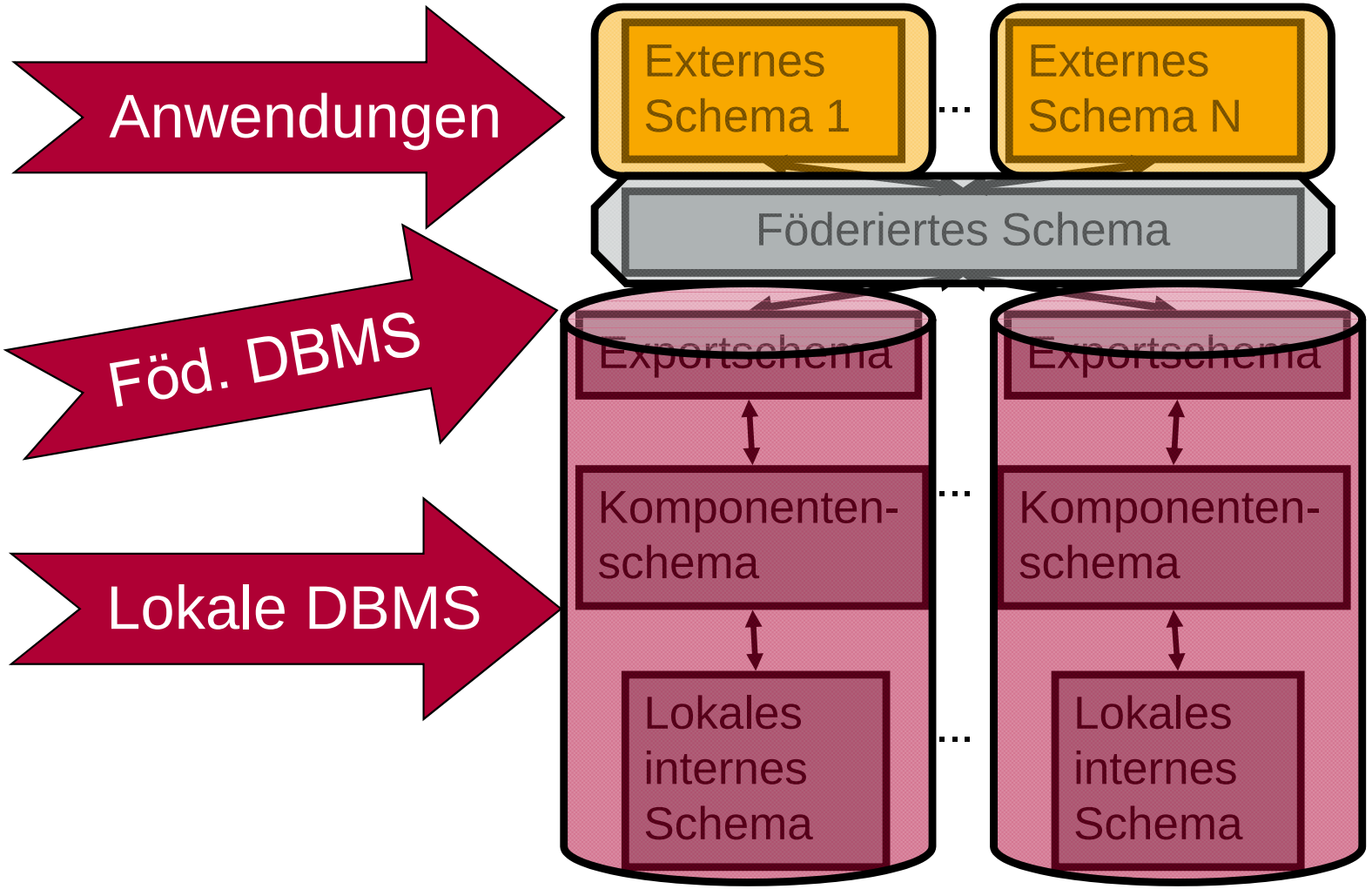


5-Schichten Architektur [SL90]

- Neu:
  - Interne Schemas werden nicht mehr betrachtet.
  - Exportschemas
  - Integriertes, föderiertes Schema
- Terminologie
  - Komponentenschema = lokales konzept. Schema
  - Föderiertes Schema = globales konzept. Schema

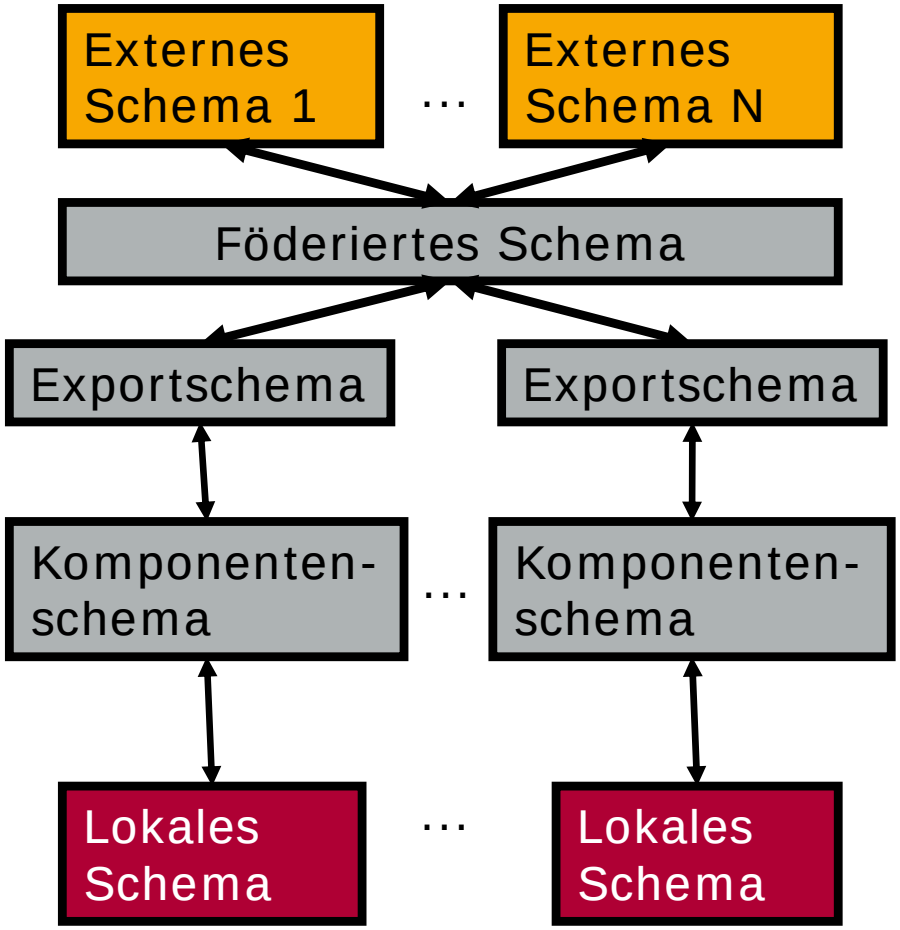


5-Schichten Architektur [SL90]



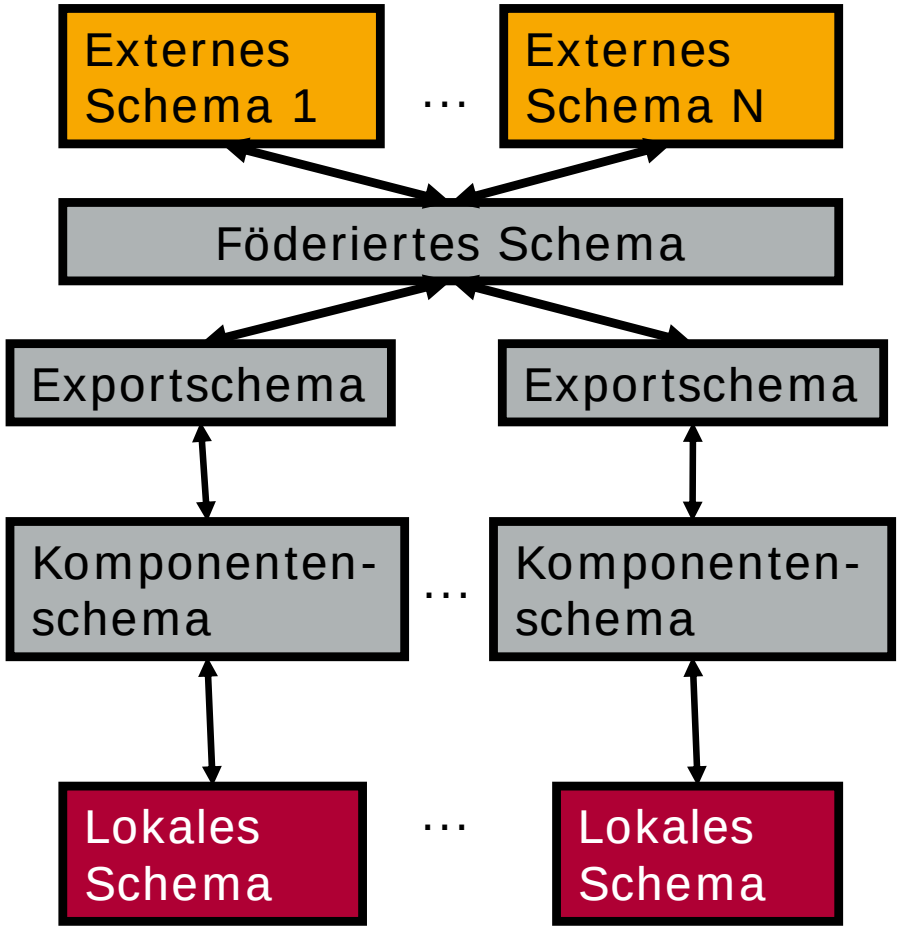
5-Schichten Architektur [SL90]

- Lokale Schemas
  - Konzeptionell
- Komponentenschemas
  - Kanonisches Datenmodell
  - Fügt fehlende Semantik hinzu.
  - Übergang durch Mappings.
- Exportschemas
  - Teilmenge des Komponentenschemas
  - Verwaltet Zugangsberechtigungen



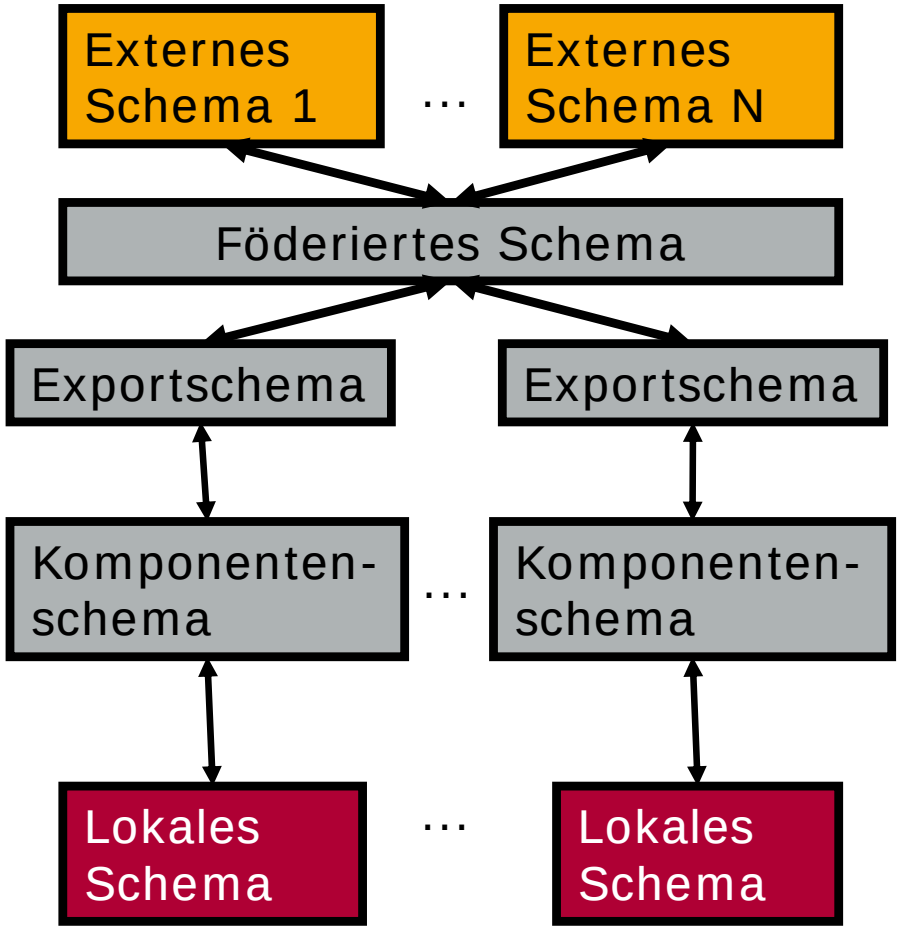
5-Schichten Architektur [SL90]

- Föderiertes Schema
  - Integriert aus den Exportschemas
  - Kennt Datenverteilung
  - Andere Namen:
    - Import Schema
    - Globales Schema
    - Enterprise Schema
    - Unified Schema
    - Mediator Schema
- Externes Schema
  - Föderiertes Schema kann sehr groß sein → Vereinfachung im Exportschema
  - „Schema Evolution“ leichter
  - Zusätzliche Integritätsbedingungen
  - Zugangskontrollen



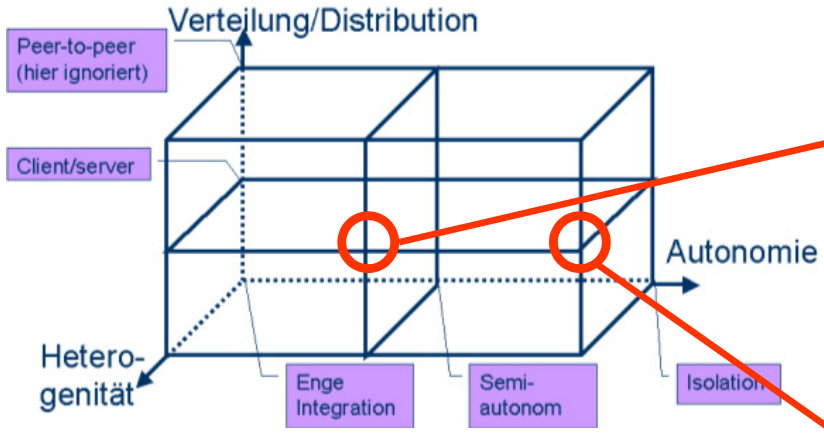
5-Schichten Architektur [SL90]

- Mischformen
  - Einige Schichten nicht immer nötig.
    - Z.B. wenn lokales und Komponentenschema gleich sind.
    - Z.B. wenn komplettes Komponentenschema exportiert werden soll.
  - Ein Komponentenschema kann mehrere Exportschemas haben.
  - Große FDBS können mehrere föderierte Schemas haben.
- Föderation!
  - Nur semi-autonom
  - Lokale DBMS müssen bereits kanonisches Datenmodell unterstützen.

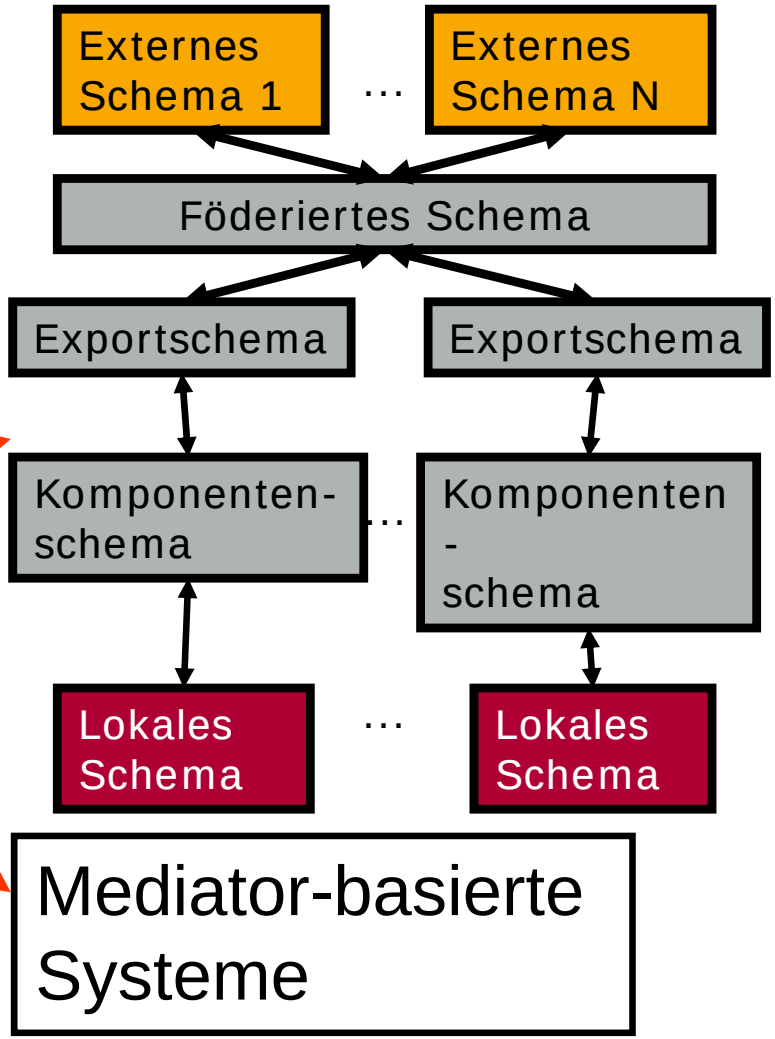


Zusammenfassung

- 1.



- 2.

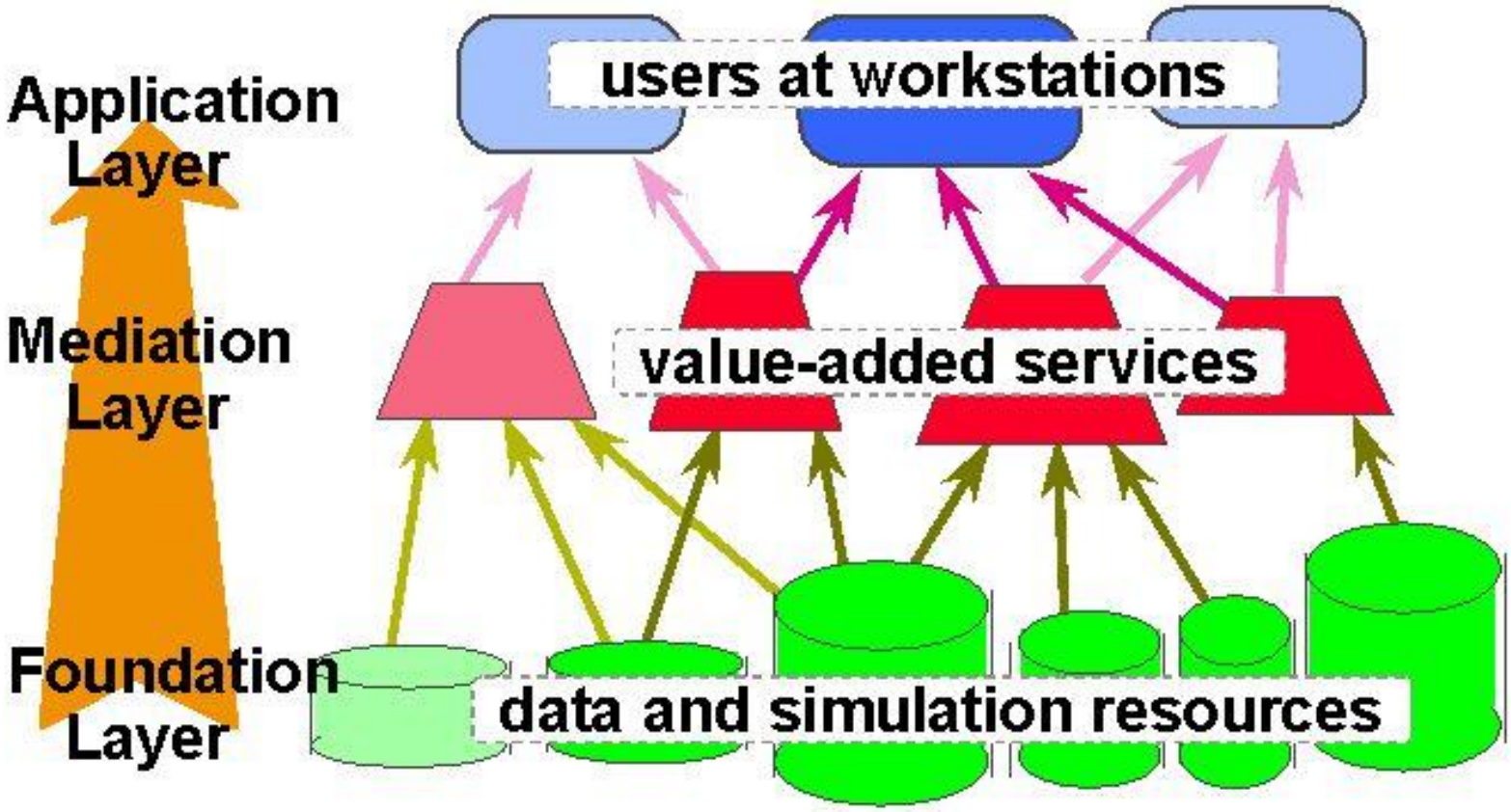


Überblick

- Überblick über Informationssysteme
  - Klassifikation
  - Materialisiert vs. virtuell
- Architekturen
  - 3 Schichten Architektur
  - 4 Schichten Architektur
  - 5 Schichten Architektur

- Mediator-Wrapper Architektur
  - Gio Wiederholds Definitionen
  - Konfigurationen
  - Mediatoren
  - Wrapper
- Peer-Data-Management
  - Architektur
  - Anwendungen

Daten werden zu Informationen



Mediatoren

- „A mediator is a software module that exploits encoded knowledge about certain sets or subsets of data to create information for a higher layer of applications“ [Wie92]

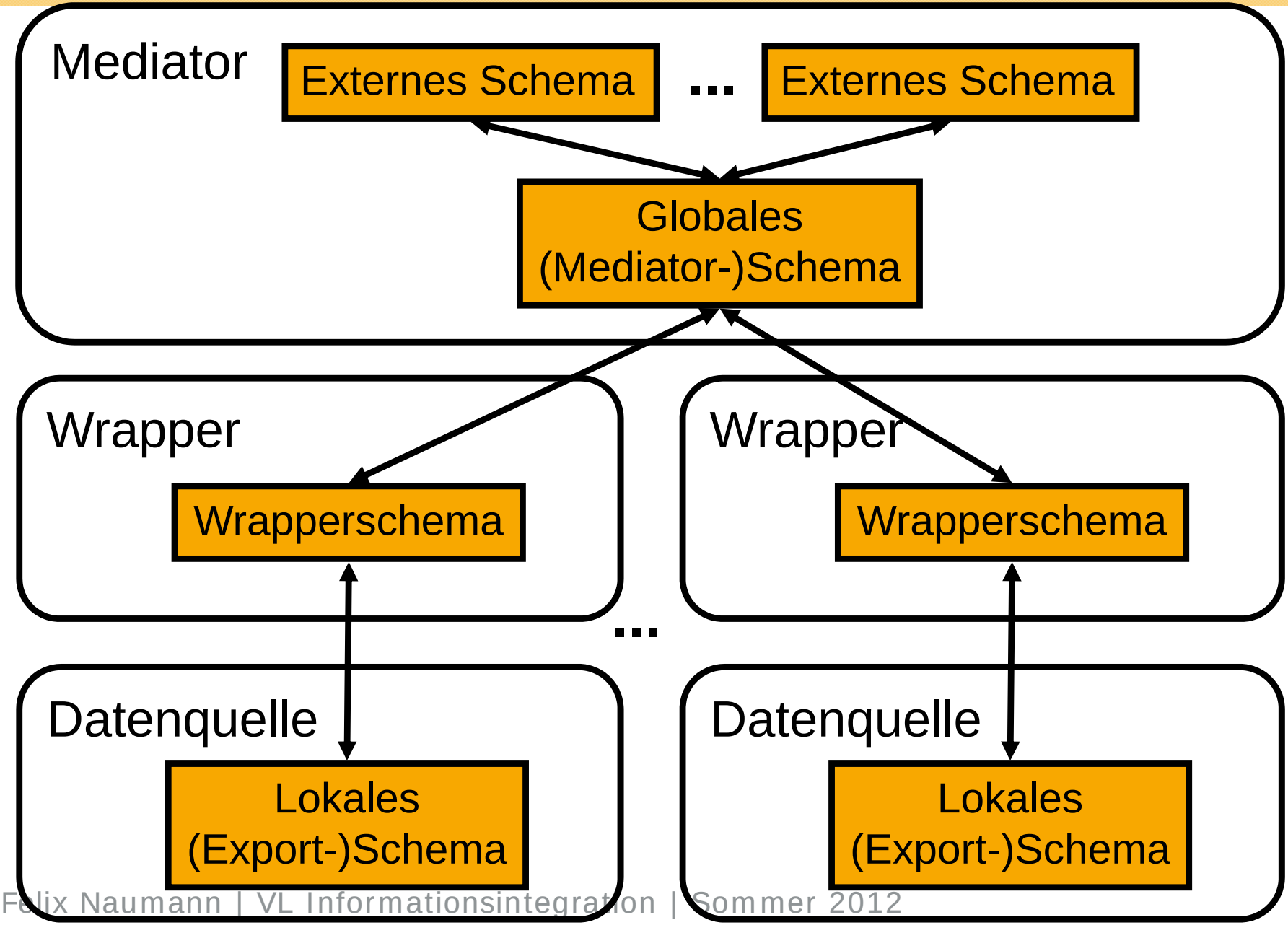
- Ein Mediator ist eine Softwarekomponente, die Wissen über bestimmte Daten benutzt, um Informationen für höherwertige Anwendungen zu erzeugen.

Überblick

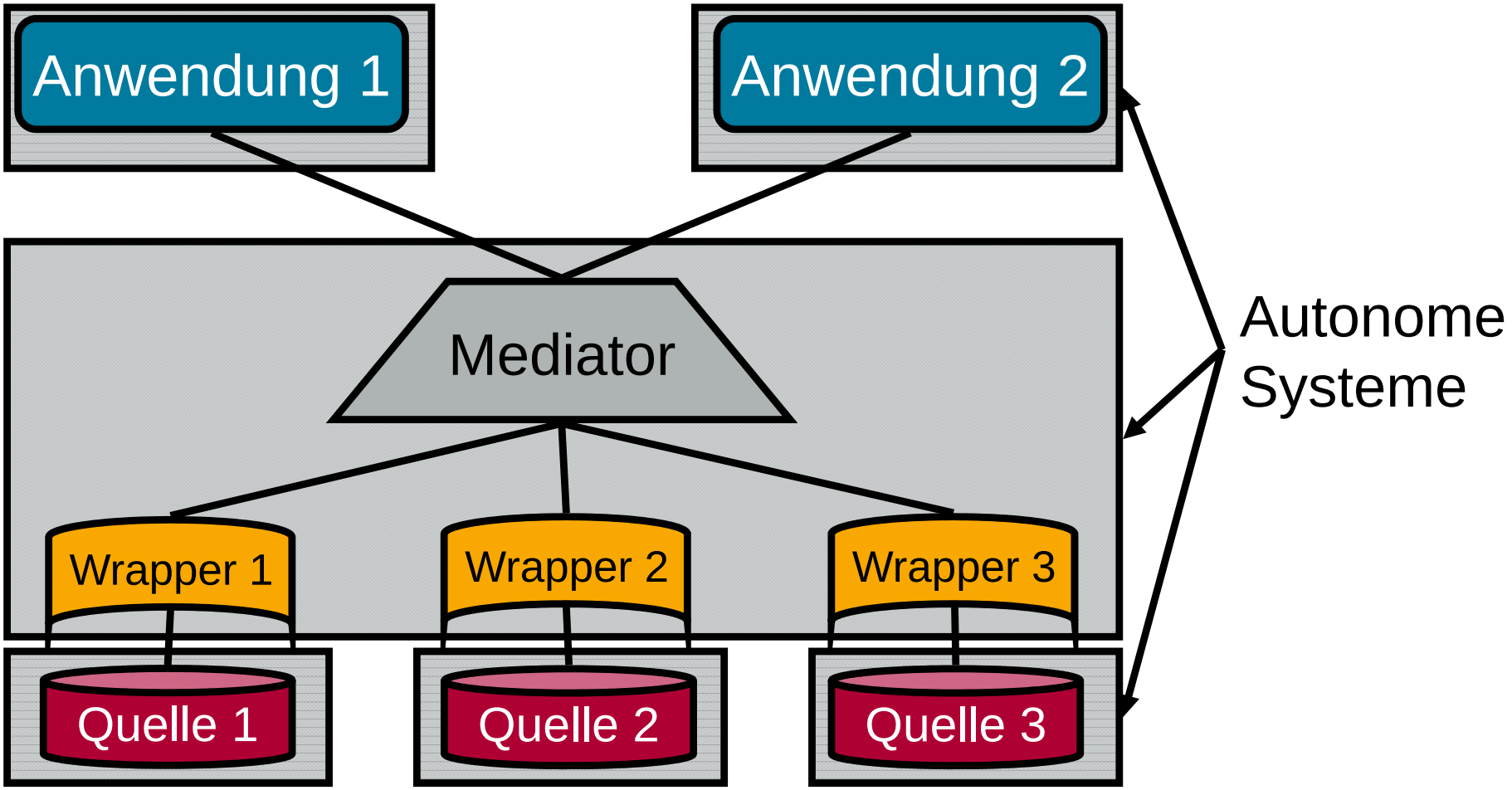
- Überblick über Informationssysteme
  - Klassifikation
  - Materialisiert vs. virtuell
- Architekturen
  - 3 Schichten Architektur
  - 4 Schichten Architektur
  - 5 Schichten Architektur

- Mediator-Wrapper Architektur
  - Gio Wiederholds Definitionen
  - Konfigurationen
  - Mediatoren
  - Wrapper
- Peer-Data-Management
  - Architektur
  - Anwendungen

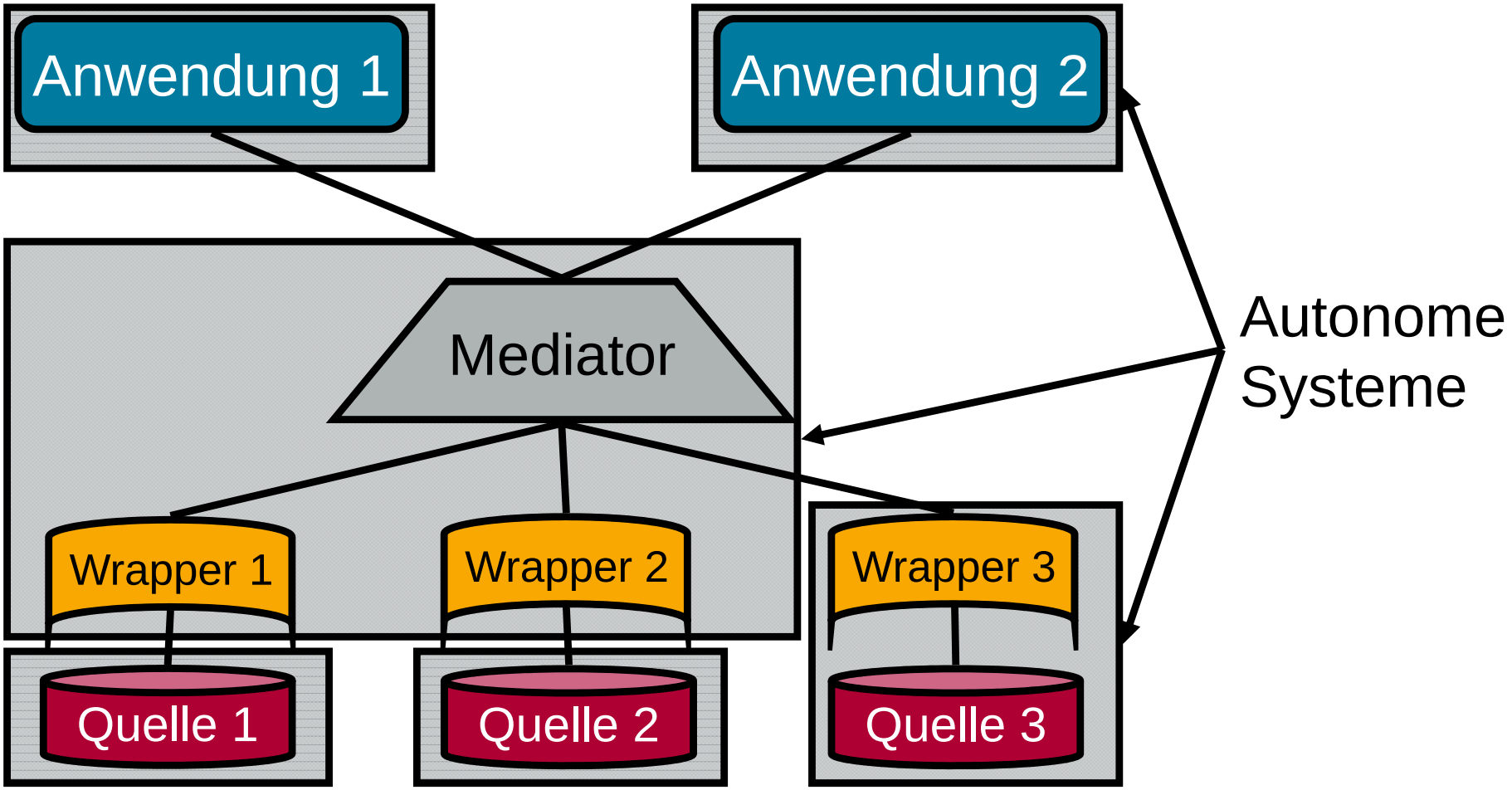
Die Architektur



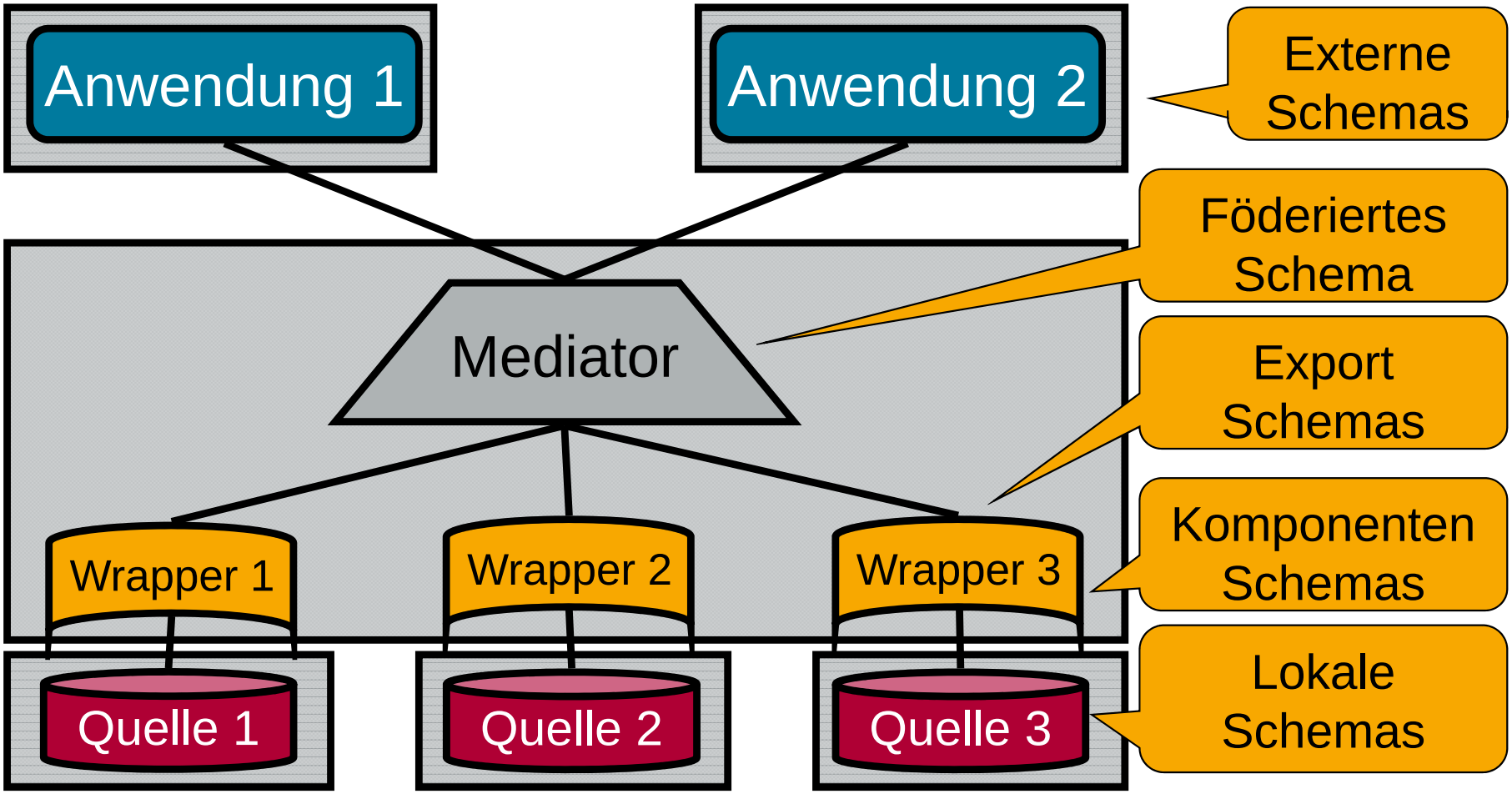
Mediator-Wrapper Architektur



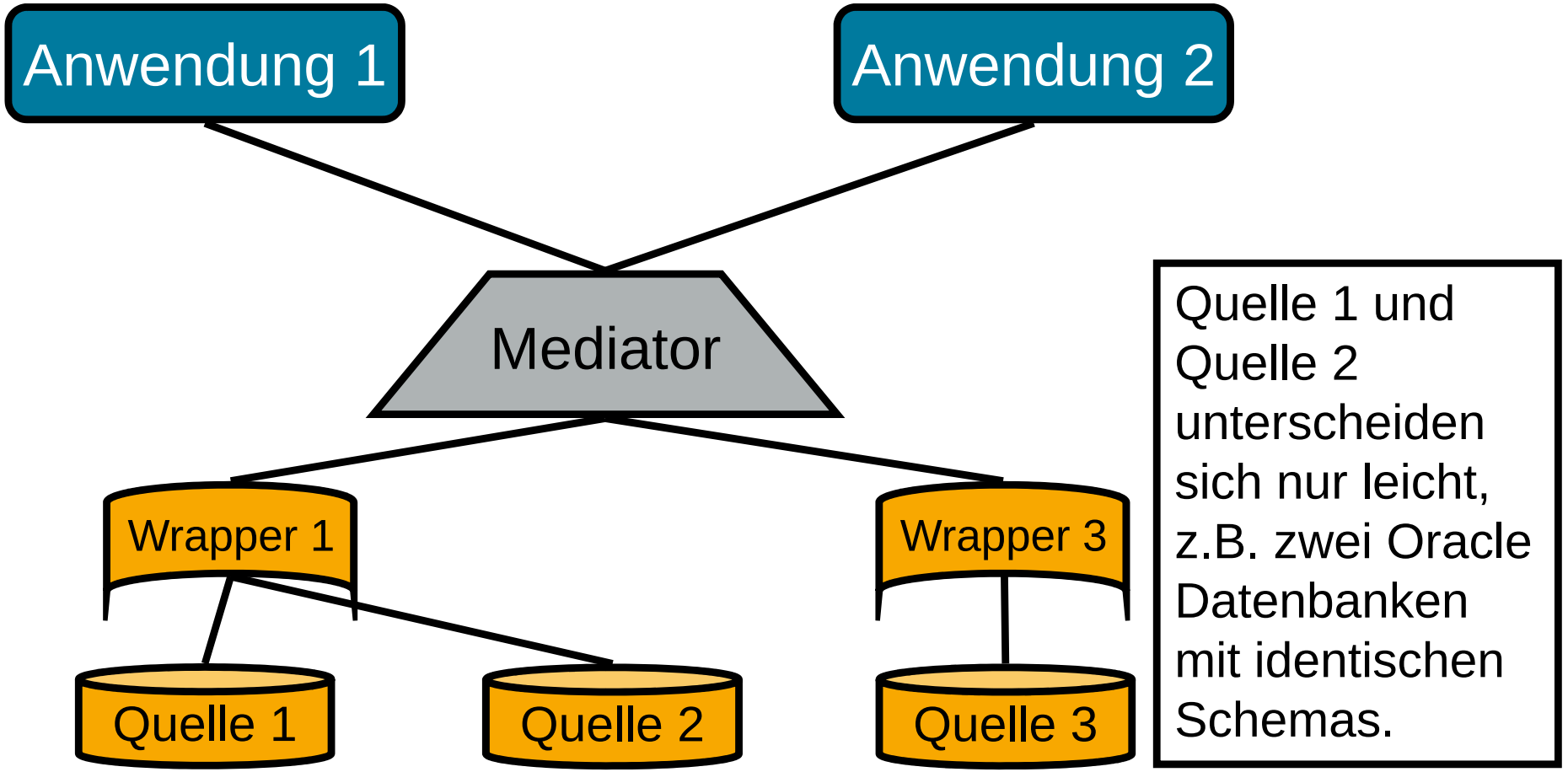
Mediator-Wrapper Architektur



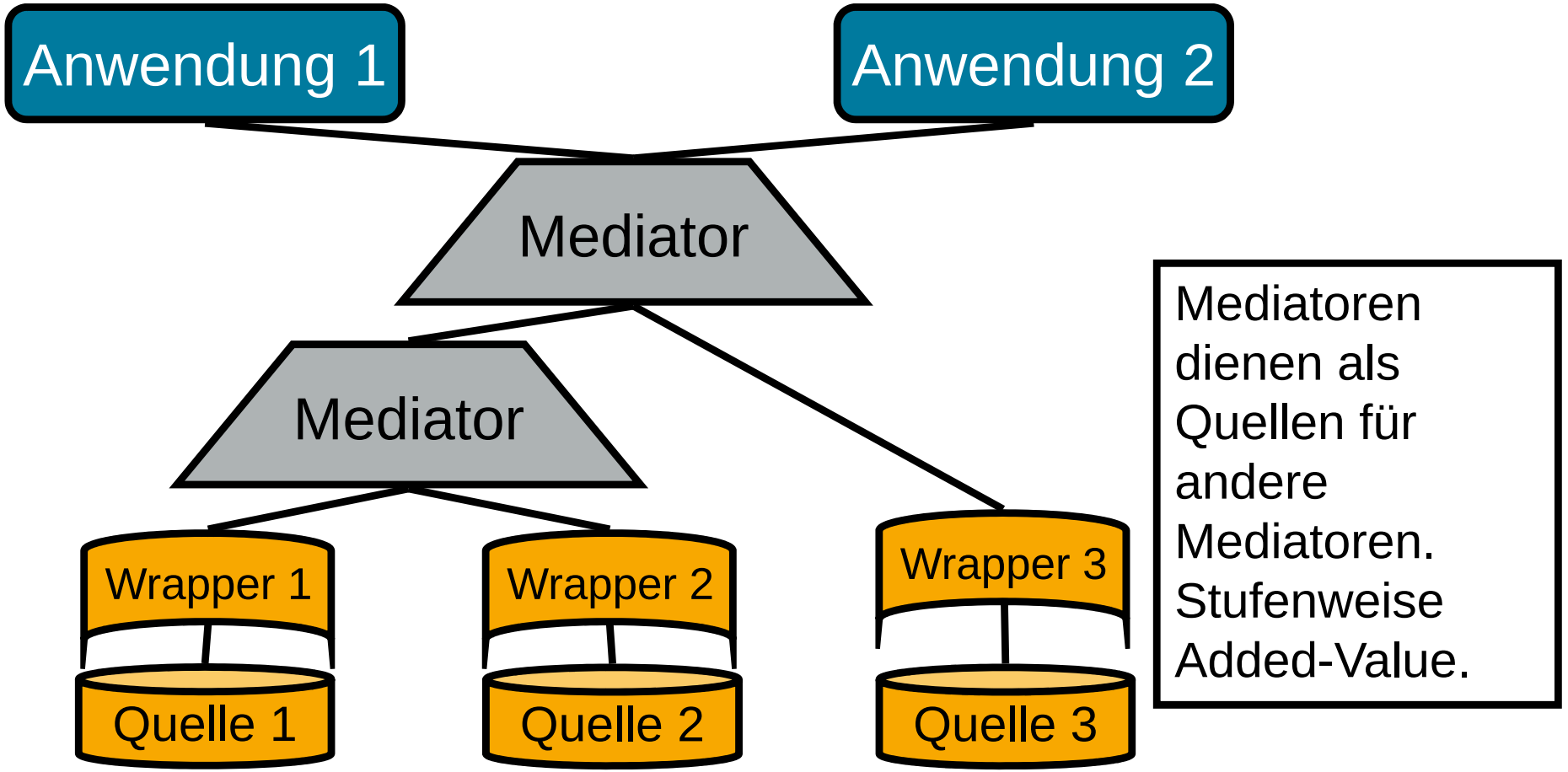
Mediator-Wrapper Architektur



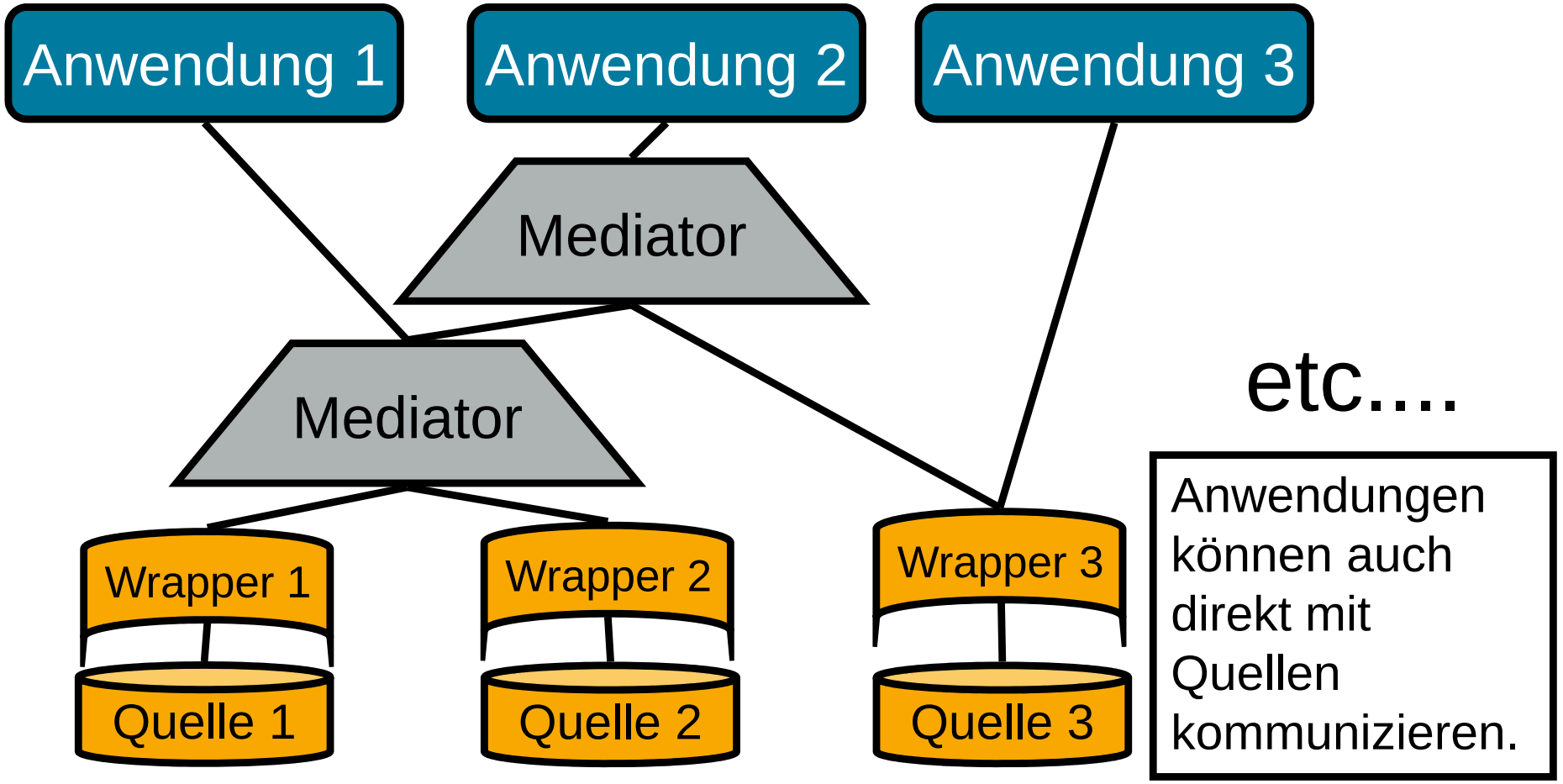
Mediator-Wrapper Architektur



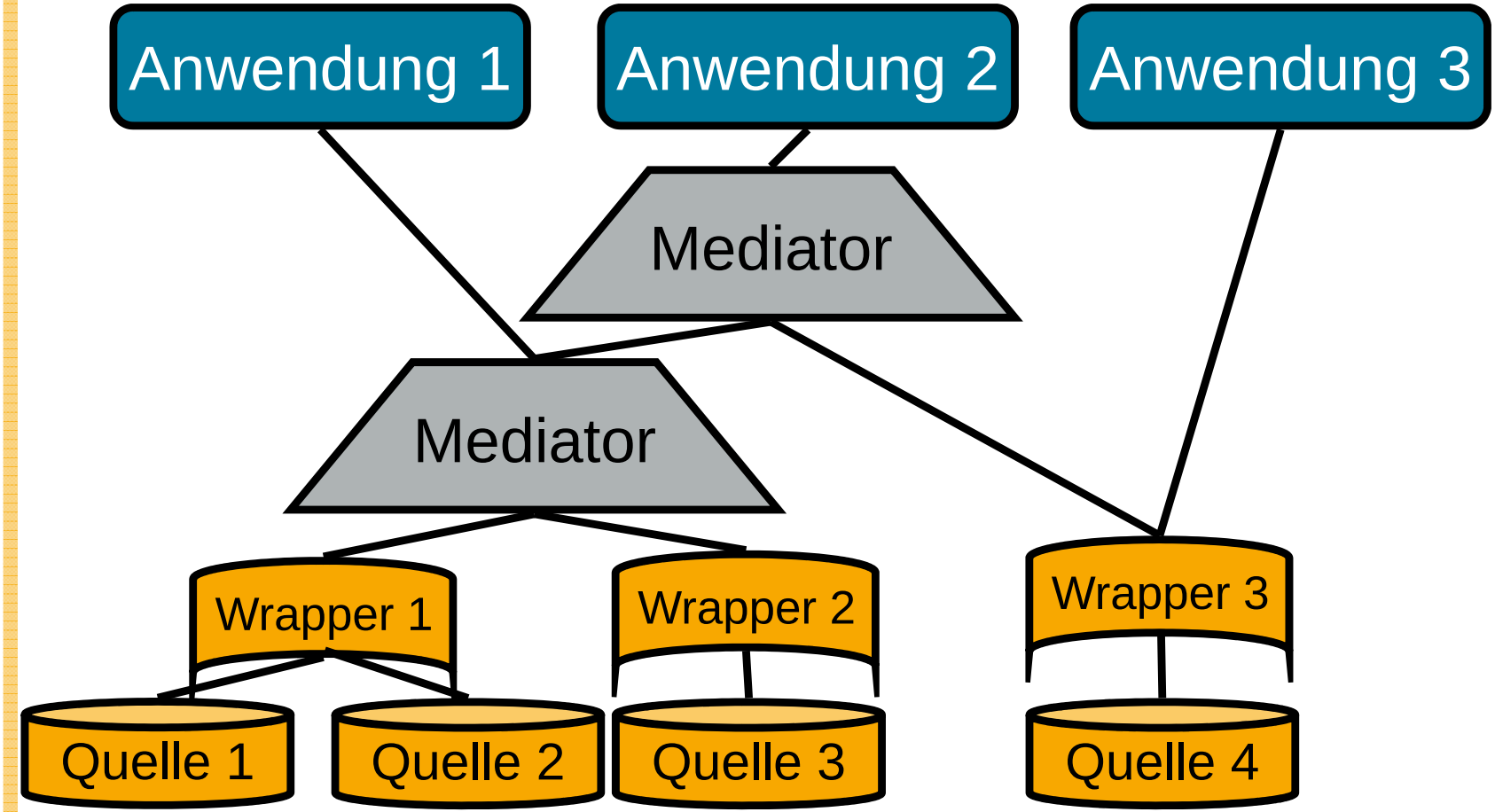
Mediator-Wrapper Architektur



Mediator-Wrapper Architektur



Mediator-Wrapper Architektur



Überblick

- Überblick über Informationssysteme
  - Klassifikation
  - Materialisiert vs. virtuell
- Architekturen
  - 3 Schichten Architektur
  - 4 Schichten Architektur
  - 5 Schichten Architektur

- Mediator-Wrapper Architektur
  - Gio Wiederholds Definitionen
  - Konfigurationen
  - Mediatoren
  - Wrapper
- Peer-Data-Management
  - Architektur
  - Anwendungen

Einfache Mediatoren

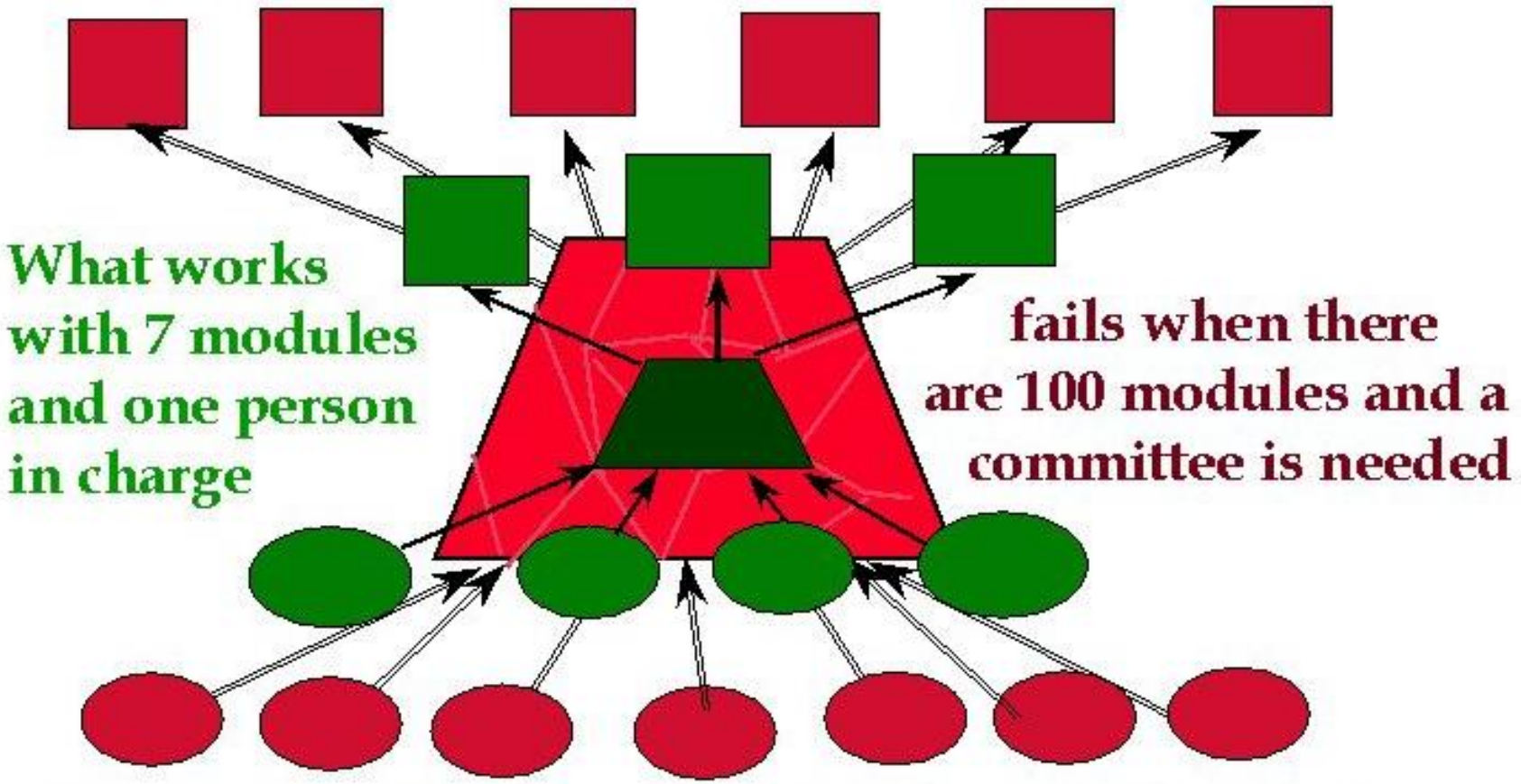
„[A mediator] should be small and simple, so that it can be maintained by one expert or, at most, a small and coherent group of experts.“ Wiederhold `92

Ein Mediator sollte klein und einfach genug sein, um durch einen einzigen oder höchstens eine kleine Gruppe von Experten gewartet werden zu können.

D.h.: Einfaches föderiertes Schema, begrenzte Domäne, einfache Schnittstellen

Erfahrung: Suchmaschinen ändern wöchentlich ihre Schnittstelle

Einfache Mediatoren



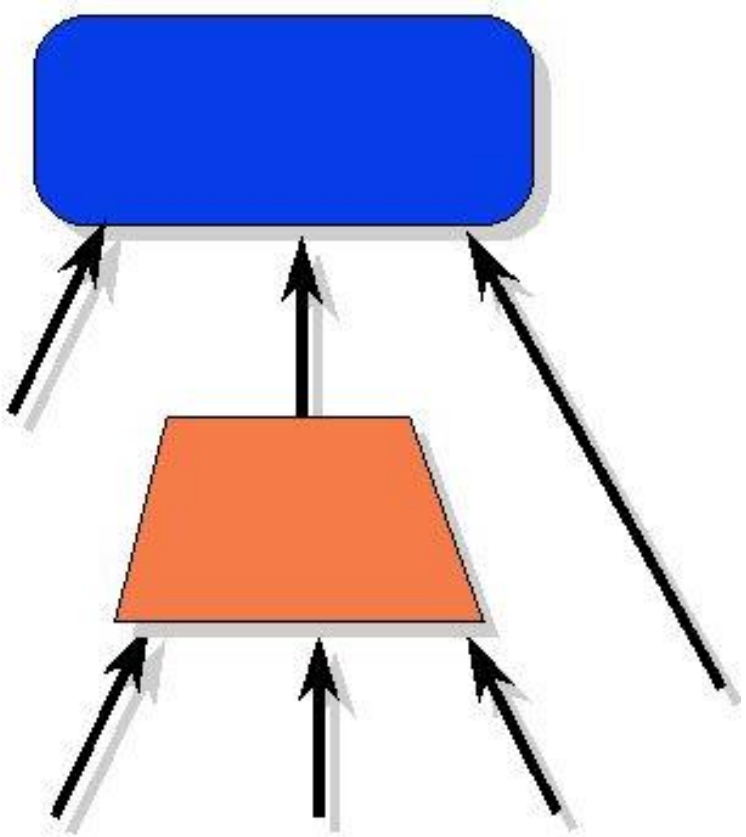
Integration mit Mediatoren

Application

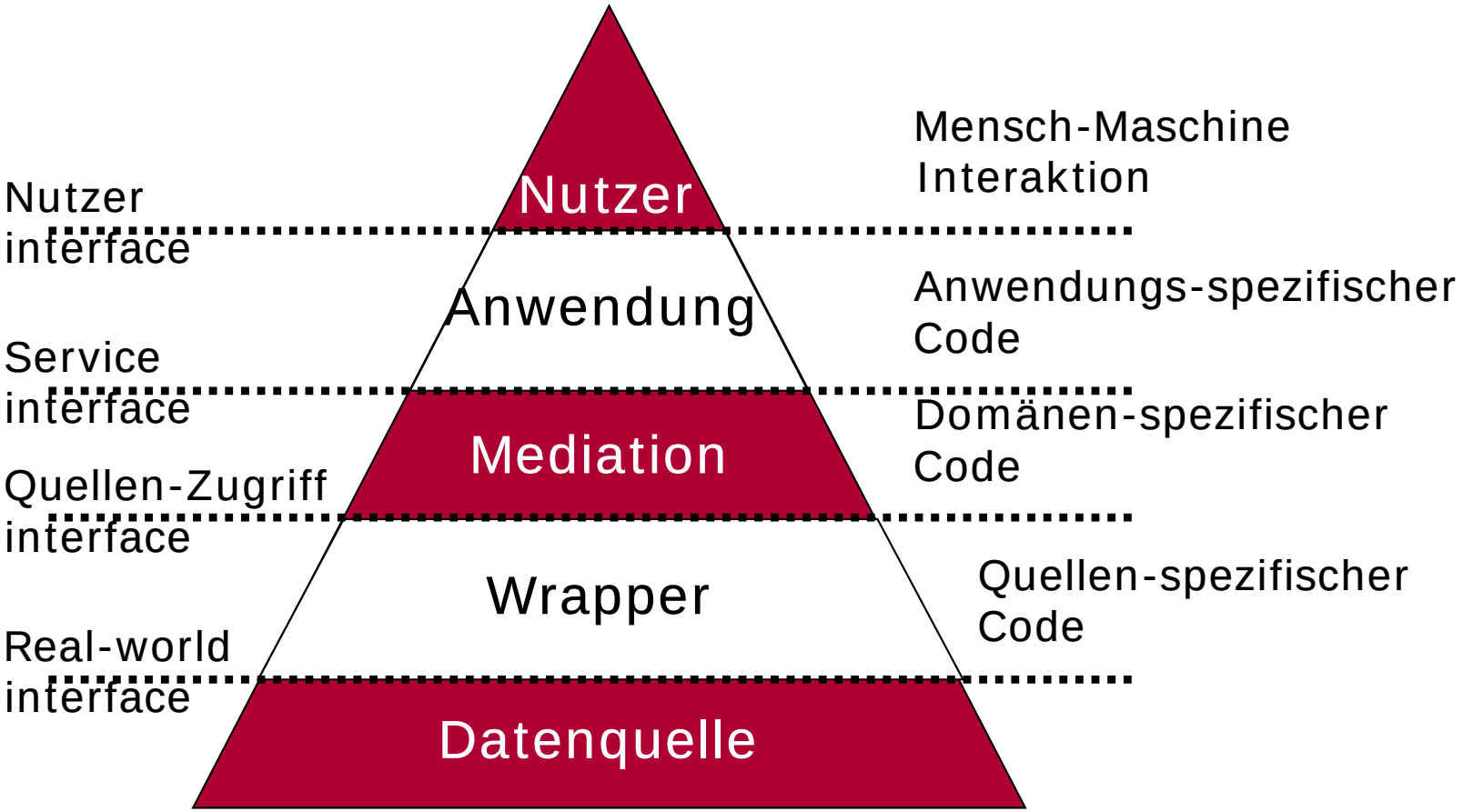
- Informal, pragmatic
- Client-control
- Use up to 7±2 mediators

Mediation

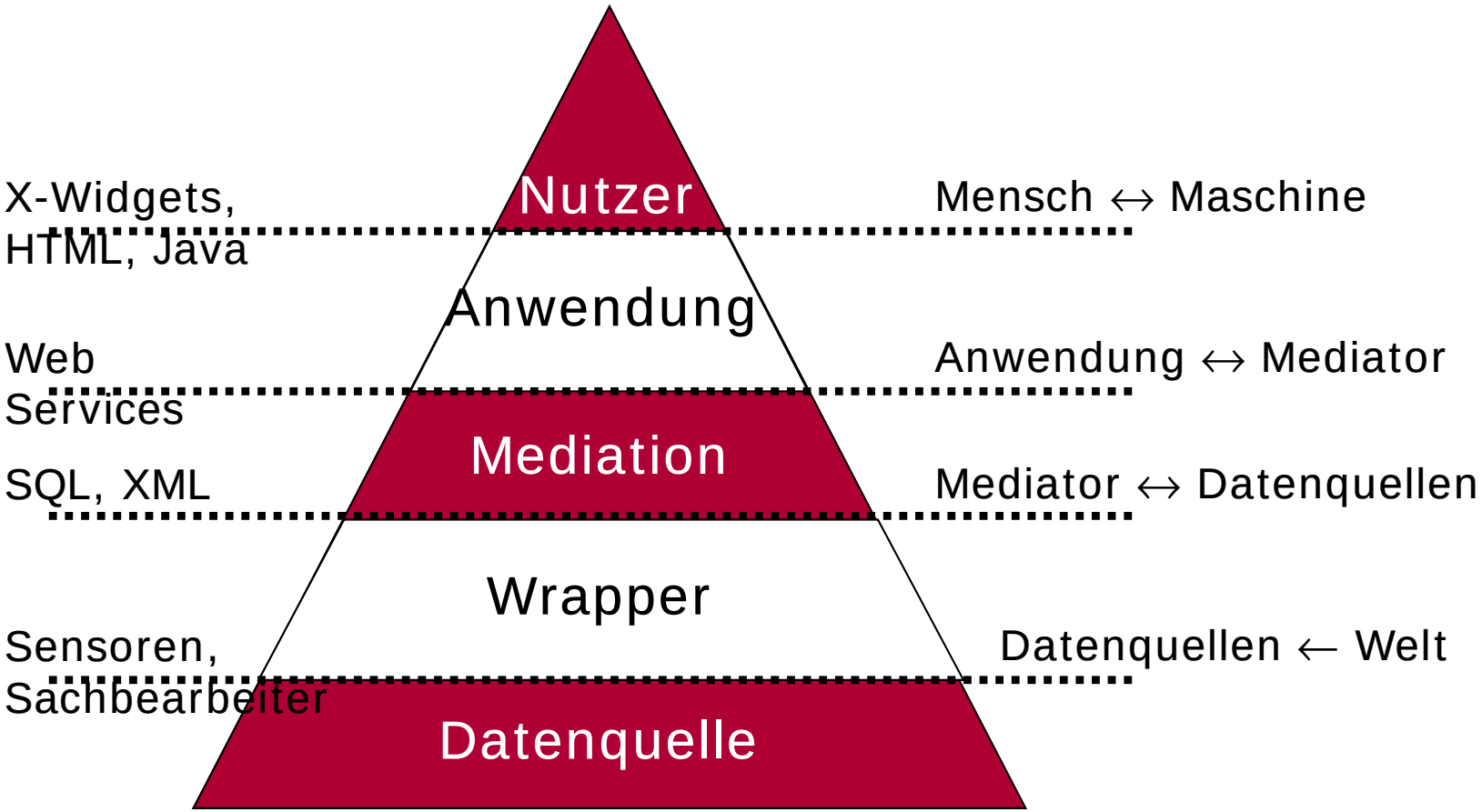
- Formal, reliable service
- Domain-Expert control
- Use up to 7±2 sources



Funktionale Schichten



Schnittstellen



Funktionen der Mediation

- Erbracht durch Domänen-Experten
  - Suche und Auswahl relevanter Informationsquellen
    - „Source selection“
  - Transformationen zur Konsistenzerhaltung
  - Metadaten zur Verarbeitung
  - Abstraktion zum Verständnis
  - Integration verschiedener Quellen
  - Zusammenfassung zur Präsentation
- All dies transformiert Daten zu Informationen.

Dicke und Dünne Mediatoren

Überblick

- Überblick über Informationssysteme
  - Klassifikation
  - Materialisiert vs. virtuell
- Architekturen
  - 3 Schichten Architektur
  - 4 Schichten Architektur
  - 5 Schichten Architektur

- Mediator-Wrapper Architektur
  - Gio Wiederholds Definitionen
  - Konfigurationen
  - Mediatoren
  - Wrapper
- Peer-Data-Management
  - Architektur
  - Anwendungen

Wrapper

- Wrapper sind Softwarekomponenten, die die Kommunikation und den Datenfluss zwischen Mediatoren und Datenquellen herstellen.
- Wrapper sind jeweils spezialisiert auf eine Ausprägung autonomer, heterogener Quellen.
- Wrapper vermitteln zwischen Mediator und Quelle.

Wrapper – Aufgaben

- Lösen Schnittstellenheterogenität
  - technisch
  - SQL, HTML Formulare, http, CORBA, …
  - Mächtigkeit der Anfragesprache
- Lösen Datenmodellheterogenität
- Lösen schematische Heterogenität
  - Liefern globales Schema
- Reduzieren Anzahl der Datenmodelle (mit denen das IIS umgehen muss)
- Reduzieren Anzahl der Schemata
- Unterstützen globale Optimierung
  - Kostenmodell
  - Anfragefähigkeiten

Wrapper – Anforderungen

- Sollten schnell implementiert werden können
  - (< 1 Woche)
- Sollten wiederverwendbar sein
- Lokale Wartung (bei föderierten Systemen)

- An den Wrappern scheitern viele Projekte!
  - Deshalb Forschung zur schnellen oder sogar automatischen Wrappergenerierung.
  - Wrapperbibliotheken

Garlic Wrapper Generierung nach [TS97]

Praktische Anforderungen aus [TS97]

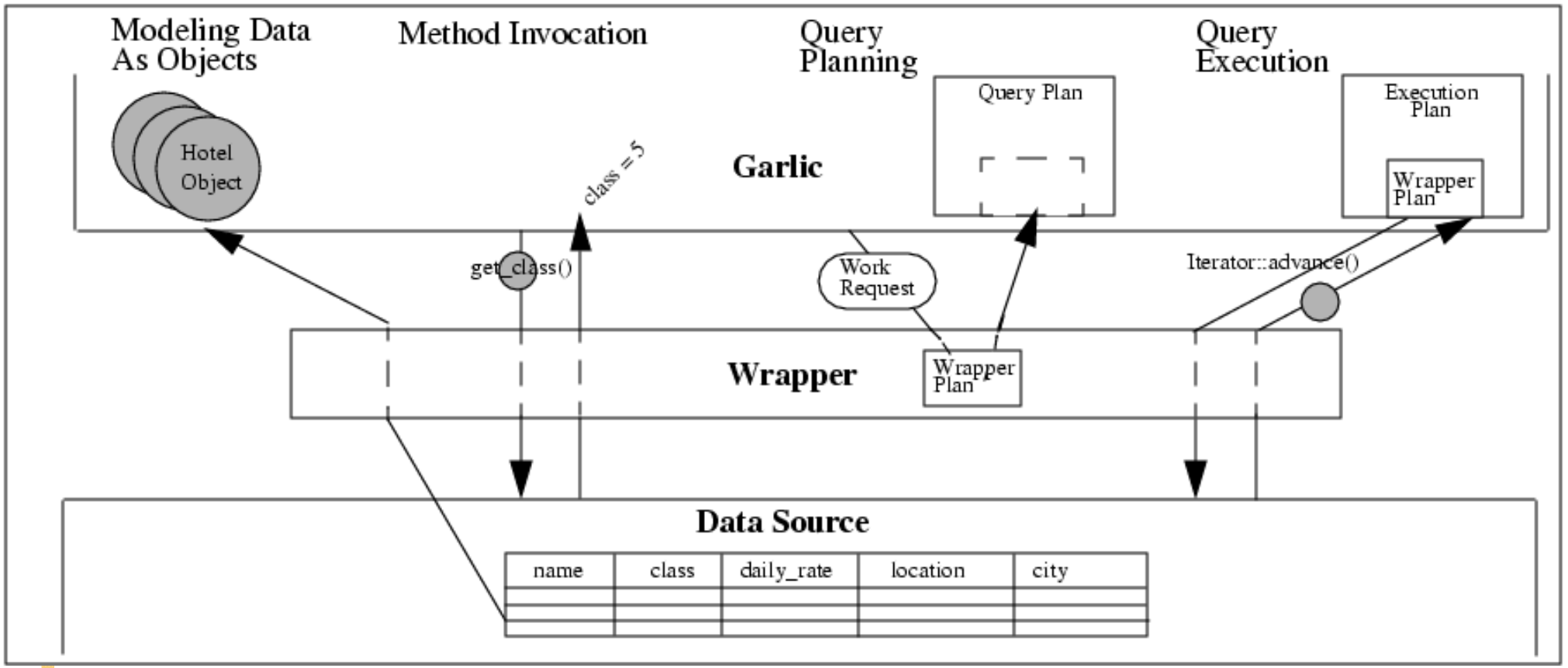
- Start-up Kosten gering (Stunden)
- Erweiterbarkeit
  - Einfacher Start
  - Später Fähigkeiten der Quellen hinzufügen
- Flexibilität
  - Möglichst breites Spektrum an Quellen abdecken
  - Neue Quellen stören Architektur nicht.
- Optimierung
  - Nicht durch Autoren sondern durch Garlic

Garlic Wrapper Generierung

Vier Grund-Services

1. Modellierung und Zugriff auf die Daten in der Quelle
2. Aufruf von Methoden in der Quelle
3. Mithilfe bei der Anfrageplanung
4. Anfrageausführung

Garlic Wrapper Generierung



Quelle: [TS97]

Garlic Wrapper Generierung

Modellierung und Zugriff auf die Daten

- Garlic nutzt OO Modell
- Wrapper stellt Daten als Objekte mit Interface (globales Schema) und Implementierung (lokales Schema) dar.
- Stellt Identität von Objekten her.

Garlic Wrapper Generierung

Aufruf von Methoden in der Quelle

- Implizit immer: Get_attr() für jedes Attribut
- Implizit immer: Set_attr() für jedes nicht-read-only Attribut
- Um auch Quellen abzudecken, die nur über Methoden zu erreichen sind.
- Um besondere Fähigkeiten von Quellen auszuschöpfen.
- Beispiel: display_Image(ImageID)

Garlic Wrapper Generierung

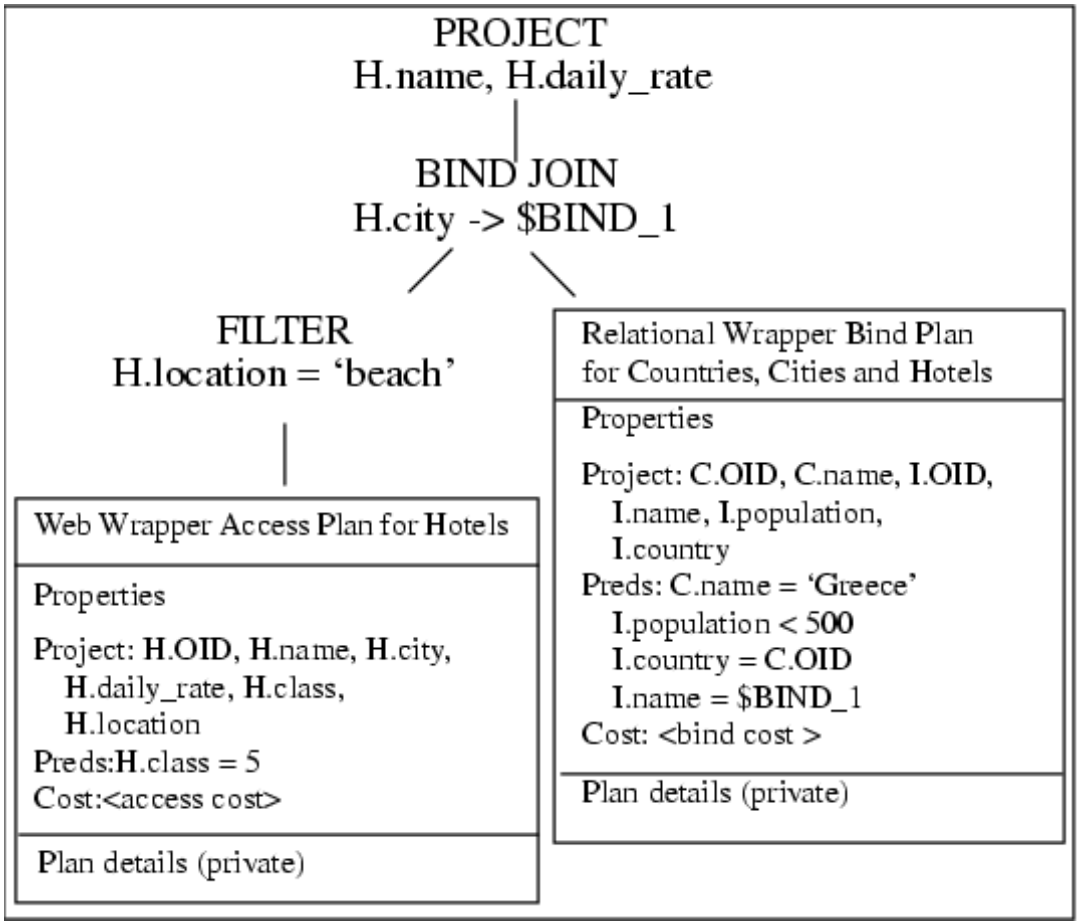
Mithilfe bei der Anfrageplanung (query planning)

- Garlics Anfrageplanung betrachtet alternative Pläne und sucht den besten heraus.
  - Kostenbasiert
- Mediator verschickt „Teilaufgaben“ an Wrapper.
  - Wrapper kann Teile davon ablehnen (je nach Fähigkeiten der Quelle).
  - Mediator gleicht aus.
    - Preprocessing
    - Postprocessing
- Wrapper liefert null oder mehr Teilpläne zurück.
- Teilpläne werden in Gesamtplan eingebaut.

Garlic Wrapper Generierung

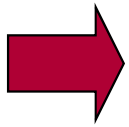
Anfrageausführung (query execution)

- Mediator produziert Operatorbaum.
- Wrapper-Teilpläne sind Blätter in dem Baum.
- Pläne werden in Iteratoren umgewandelt
- Pipelining



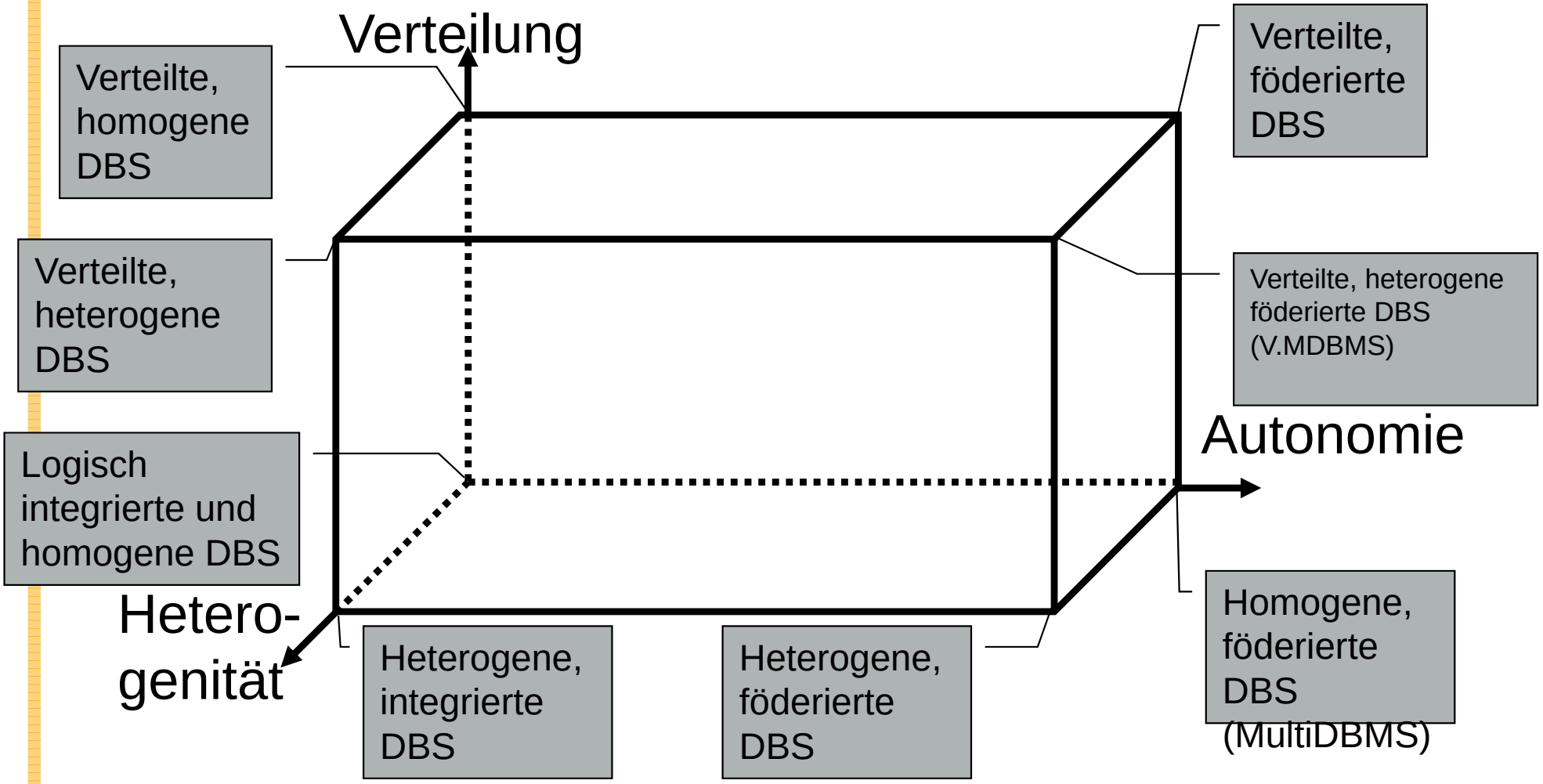
Überblick

- Überblick über Informationssysteme
  - Klassifikation
  - Materialisiert vs. virtuell
- Architekturen
  - 3 Schichten Architektur
  - 4 Schichten Architektur
  - 5 Schichten Architektur

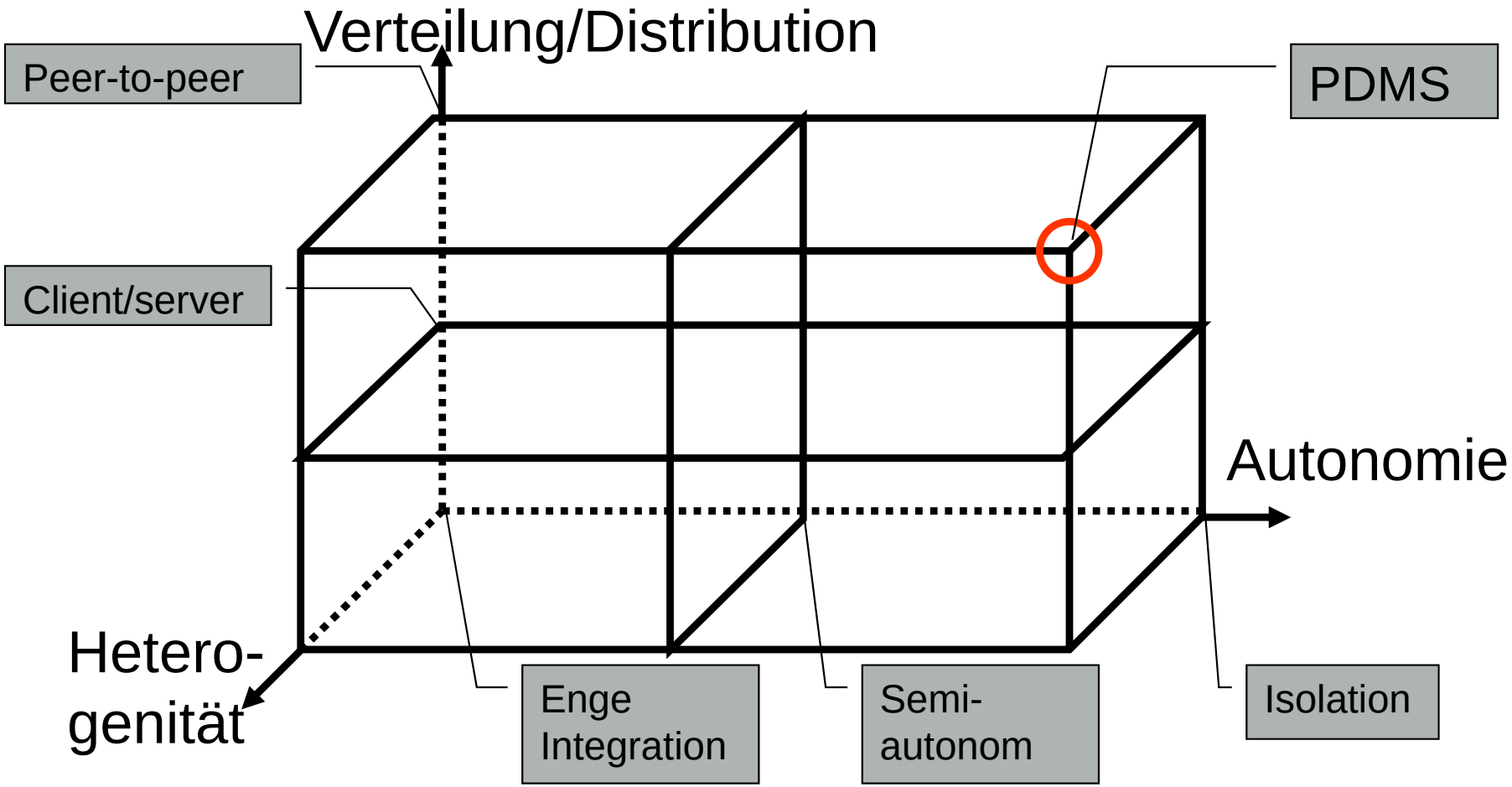


- Mediator-Wrapper Architektur
  - Gio Wiederholds Definitionen
  - Konfigurationen
  - Mediatoren
  - Wrapper
- Peer-Data-Management
  - Architektur
  - Anwendungen

Wdh: Klassifikation von Informationssystemen nach [ÖV91]



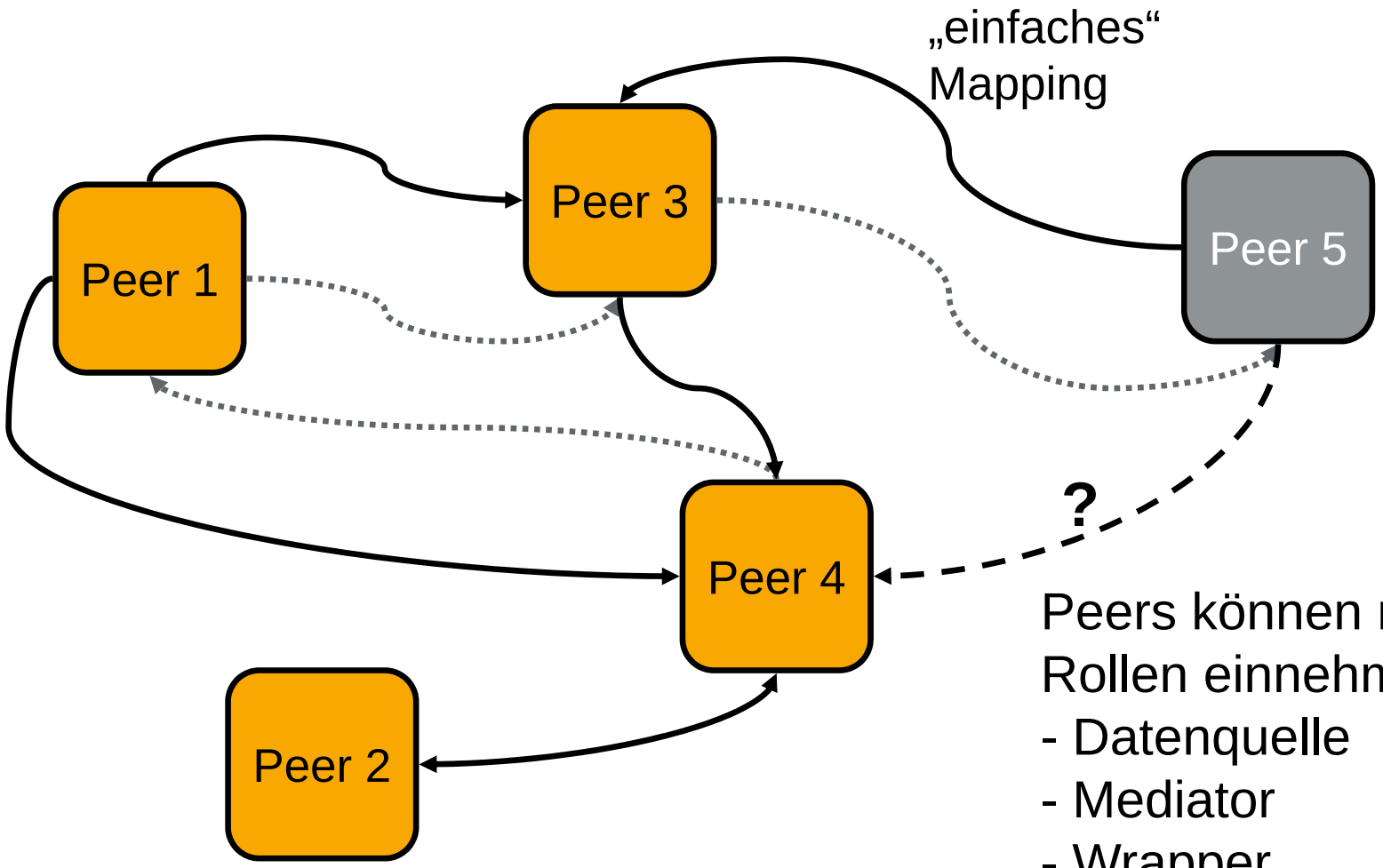
Wdh: Erweiterung der Klassifikation nach [ÖV99]



PDMS – Idee

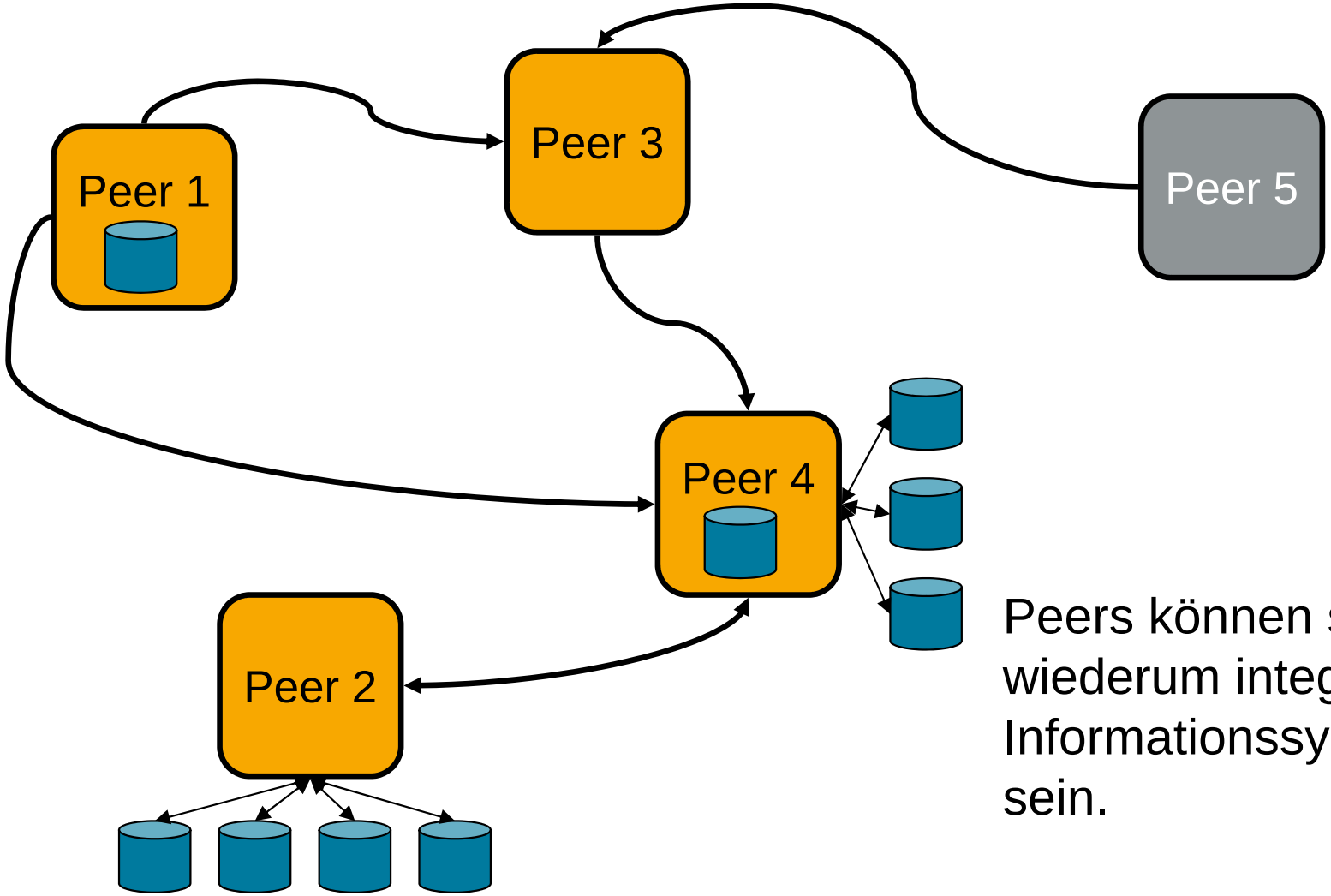
- Idee: Peer Netzwerk (P2P)
  - [HIST03], [HIMT03], [BGK+02]
- Jeder Peer kann
  - Daten exportieren (= Datenquelle)
  - Sichten auf Daten zur Verfügung stellen (= Wrapper)
  - Anfragen anderer Peers entgegennehmen und weiterleiten (= Mediator)
  - Anfrage stellen
- Verknüpfungen nicht zwischen lokalen und globalem Schema, sondern zwischen Paaren von Peers.

Peer-Data-Management Systeme (PDMS)



Peers können mehrere Rollen einnehmen:
- Datenquelle
- Mediator
- Wrapper
- Anfrager

Peer-Data-Management Systeme (PDMS)



Peers können selbst wiederum integrierte Informationssysteme sein.

PDMS vs. P2P Dateiaustausch

- P2P
  1. Nur ganze Dateien (niedrige Granularität)
  2. Einfachste Anfragen
     - Dateinamen, Keywords
  3. Unvollständige Anfrageergebnisse
  4. Einfaches Schema
  5. Hoch dynamisch
  6. Millionen Peers
  7. Datenübertragung direkt

- PDMS
  1. Objekte (hohe Granularität)
  2. Komplexe Anfragen
     - Anfragesprache (SQL, etc.)
  3. Vollständige Anfrageergebnisse (zumindest erwartet)
  4. Schema
  5. Kontrollierte Dynamik
  6. Zig peers
  7. Datenübertragung entlang des Mapping-Pfads

PDMS vs. FDBMS

- Vorteile
  - Nutzer/Anwendungen müssen nur das eigene Schema kennen.
  - Alle Daten sind erreichbar (über die transitive Hülle der Mappings).
  - Neue Schemata und Peers können leicht hinzugefügt werden (inkrementell).
  - Mappings nur zu ähnlichsten Schemata
- Nachteile / Probleme
  - Mapping-Erstellung ist schwierig.
  - Mapping Komposition ist schwierig.
  - Viele Mappingschritte durch das Netzwerk:
    - Effizienz
    - Skalierbarkeit
    - Datenqualität
  - Effiziente Platzierung von Daten?
  - Updates?

Literatur

Wichtige Literatur

- [ÖV99] Principles of Distributed Database Systems. M. Tamer Özsu, Patrick Valduriez, Prentice Hall, 1999.
- [SL90] Amit P. Sheth and James A. Larson, Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases, ACM Computing Surveys, Vol. 22(3), pp183-236, 1990.
- [Con97] Stefan Conrad, Föderierte Datenbanksysteme. Springer, Heidelberg 1997.
- [HIST03] Alon Y. Halevy, Zachary G. Ives, Dan Suciu, Igor Tatarinov. Schema Mediation in Peer Data Management Systems, ICDE conference 2003
- [Wie92] Mediators in the Architecture of Future Information Systems, Gio Wiederhold, IEEE Computer Journal, 25(3), 38-49, 1992

Weitere Literatur zu Architekturen

- [LMR90] W. Litwin, L. Mark, N. Roussoupoulos, Interoperability of Multiple Autonomous Databases, ACM Computing Surveys, Vol. 22(3), pp267-293, 1990.
- [HM85] Dennis Heimbigner, Dennis McLeod: A Federated Architecture for Information Management. ACM Trans. Inf. Syst. 3(3): 253-278 (1985)

Literatur

Weitere Literatur zu Mediator Wrapper Architektur

- [Gar95] Michael J. Carey, Laura M. Haas, Peter M. Schwarz, Manish Arya, William F. Cody, Ronald Fagin, Myron Flickner, Allen Luniewski, Wayne Niblack, Dragutin Petkovic, Joachim Thomas II, John H. Williams, Edward L. Wimmers: Towards Heterogeneous Multimedia Information Systems: The Garlic Approach. RIDE-DOM 1995: 124-131
- [JS03] Querying XML data sources in DB2: the XML Wrapper, Vanja Josifovski and Peter Schwarz, ICDE conference, Bangalore, India, 2003
- [TS97] Don‘t Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources, Mary Tork Roth and Peter Schwarz, VLDB Conference 1997 Athens, Greece, 1997.

Weitere Literatur zu PDMS

- [BGK+02] Philip A. Bernstein, Fausto Giunchiglia, Anastasios Kementsietsidis, John Mylopoulos, Luciano Serafini, Ilya Zaihrayeu: Data Management for Peer-to-Peer Computing : A Vision. WebDB 2002: 89-94
- [ÖV91] & [ÖV99] Principles of Distributed Database Systems. M. Tamer Özsu, Patrick Valduriez, Prentice Hall, 1991/1999.
- [HIMT03] Alon Y. Halevy, Zachary G. Ives, Peter Mork, Igor Tatarinov. Peer Data Management Systems: Infrastructure for the Semantic Web. WWW Conference, 2003.
- [NOTZ03] W.S.Ng, B.C. Ooi, K.L. Tan und A. Zhou: PeerDB: A P2P-based System for Distributed Data Sharing. ICDE 2003
- [Len04] Maurizio Lenzerini: Quality-aware data integration in peer-to-peer systems. Invited talk at IQIS workshop, Paris, 2004.