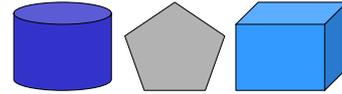


Datenintegration

Datenintegration



Kapitel 4: Architekturen von Integrationssystemen

Dr. Andreas Thor
Sommersemester 2009

Universität Leipzig
Institut für Informatik
<http://dbs.uni-leipzig.de>

1



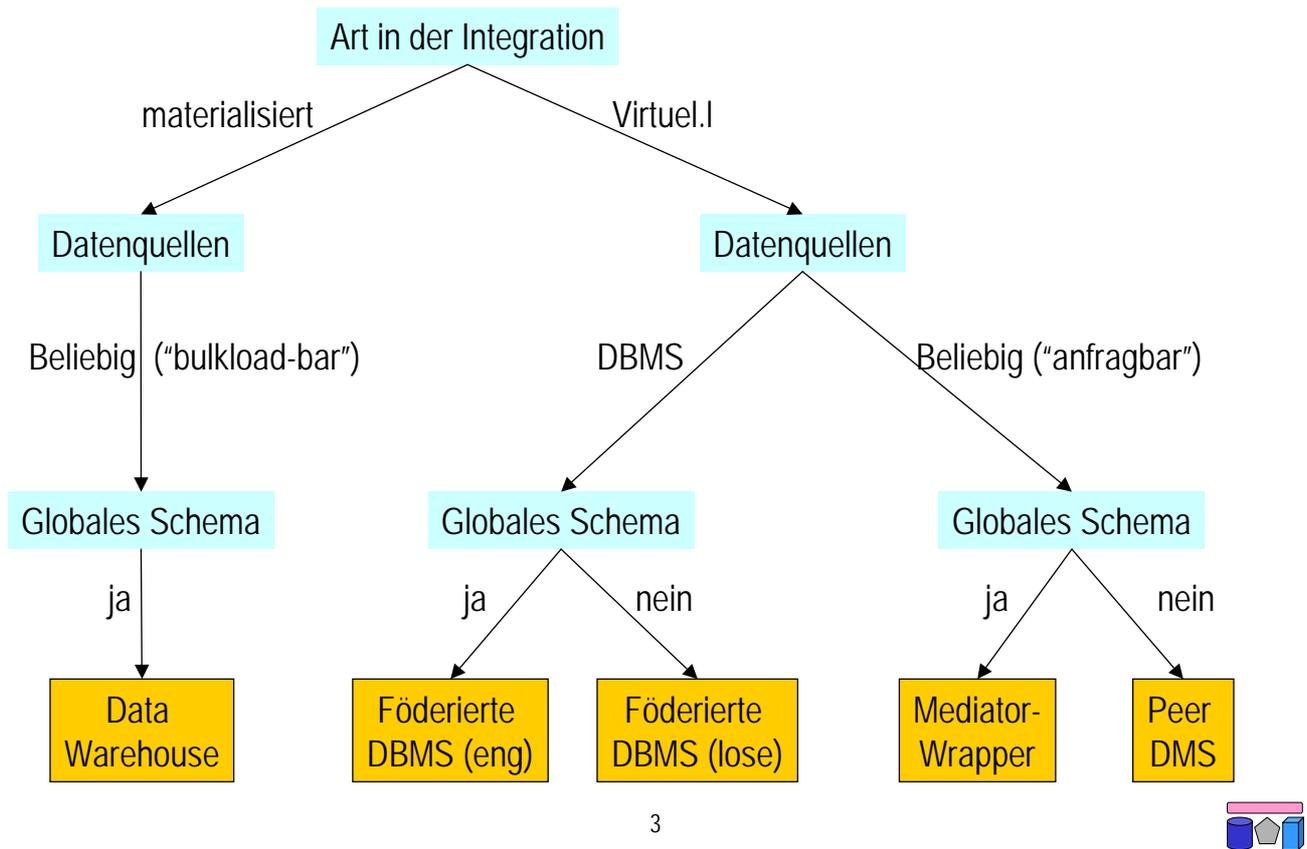
Inhalt

- Übersicht über Arten von Informationssysteme
- Data Warehouse
 - Definition und Architektur
 - Mehrdimensionale Datenmodellierung und -analyse
- Föderierte Datenbanken
 - Taxonomie von DBMS
 - Schichtenaufbau Integrierter Systeme
- Mediator-Wrapper-Ansatz
 - Architektur, Funktionen von Mediator & Wrapper
 - Web Wrapper
- Peer Data Management Systeme
 - Definition und Architektur
 - Vergleich zu P2P-Systemen
- Weitere Architekturen
- Architekturen vs. Integrations(teil)aufgaben

2

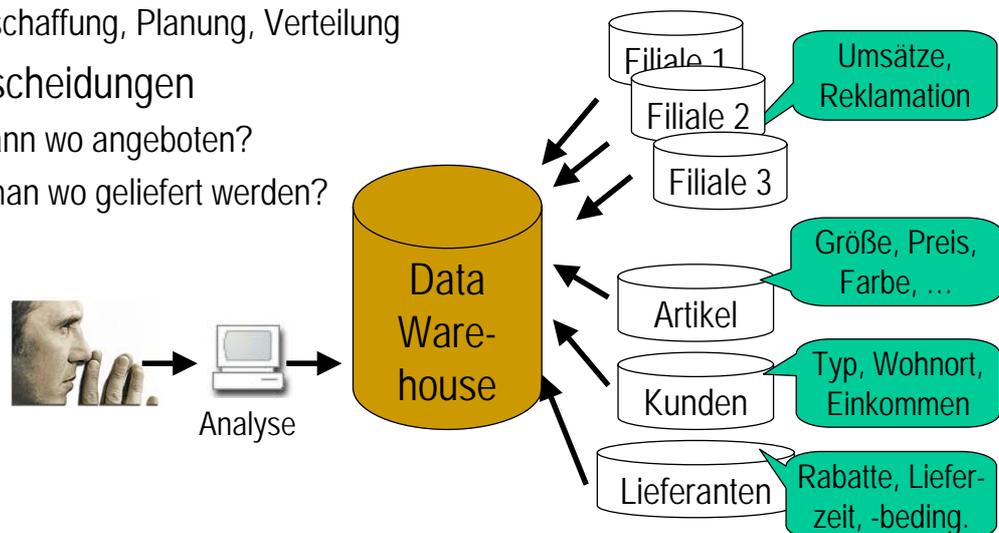


Integrationssysteme: Übersicht (Auswahl)



Data Warehouse: Szenario

- Handelshaus mit mehreren Filialen
 - Physikalische Datenverteilung (Niederlassungen, ...)
 - Zentrale Beschaffung, Planung, Verteilung
- Fragen & Entscheidungen
 - Was wird wann wo angeboten?
 - Was muss man wo geliefert werden?



- Analysen
 - Welches sind die Topkunden? Welches sind die Topfilialen?
 - Welche Produkte hatten im letzten Jahr im Bereich Leipzig einen Umsatzrückgang um mehr als 10%?
 - Haben Filialen einen höheren Umsatz, die gemeinsam gekaufte Produkte zusammen stellen?

Data Warehouse: Problem & Definition

- Problem: viele Unternehmen haben Unmengen an Daten, ohne daraus ausreichend Informationen und Wissen für kritische Entscheidungsaufgaben ableiten zu können
- Data Warehouse ist eine für Analysezwecke optimierte zentrale Datenbank, die Daten aus mehreren, i.a. heterogenen Quellen zusammenführt und verdichtet (Integration und Transformation)
- *A data warehouse is a subject-oriented, integrated, non-volatile, and time-variant collection of data in support of management's decision.* [In96]
 - Subject-oriented: Verkäufe, Personen, Produkte, etc. (*nicht* task-oriented)
 - Integrated: Erstellt aus vielen Quellen
 - Non-Volatile: Hält Daten unverändert über die Zeit
 - Time-Variant: Vergleich von Daten über die Zeit
 - Decisions: Wichtige Daten rein, unwichtige raus

[In96] W. H. Inmon, Building the Data Warehouse, 2nd ed., John Wiley 1996

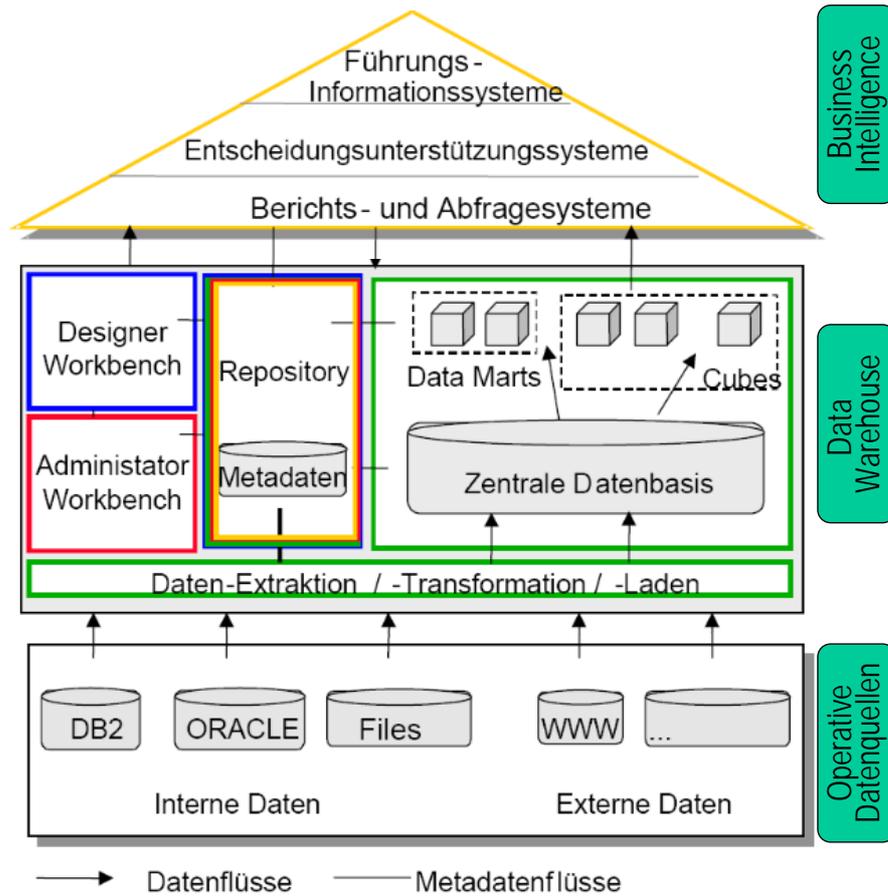


Operationale Datenbanken vs. DWH

	Operationale Datenbanken / OLTP (Online Transaction Processing)	Data Warehouse / OLAP (Online Analytical Processing)
Typische Operationen	Insert, Update, Delete, Select	
Transaktionen	Viele und kurz	
Typische Anfragen	Einfache Queries, Primärschlüsselzugriff, Schnelle Abfolgen von Selects/inserts/updates/deletes	
Daten pro Operation	Wenige Tupel	
Datenmenge in DB	Gigabyte	
Änderungen	Sehr häufig / stets aktuell	
Dateneigenschaften	Rohdaten	
Antwortzeiten (erwart.)	Echtzeit bis wenige Sek.	
Modellierung	Anwendungsorientiert	
Typische Benutzer	Sachbearbeiter	

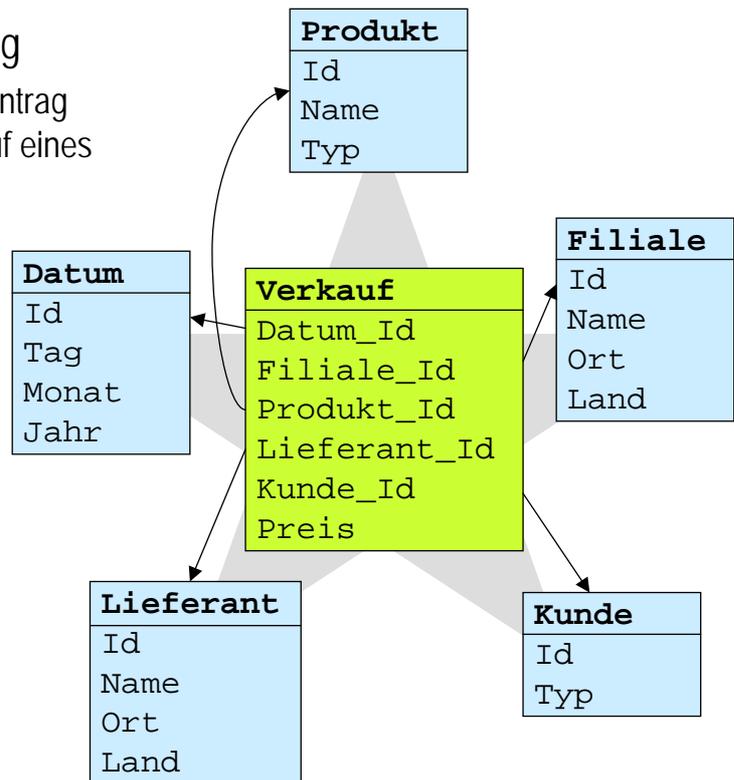


Grobarchitektur einer DWH-Umgebung



Datenmodellierung

- Multidimensionale Modellierung
 - Faktentabelle enthält je einen Eintrag pro "Analyseobjekt", z.B. Verkauf eines Produkts
 - Dimensionstabellen zur Beschreibung der Fakten
- Schemata
 - Häufig: Star-Schema (rechts)
 - Snowflake-Schema
 - Galaxienschema



Faktentabelle

Dimensionstabelle

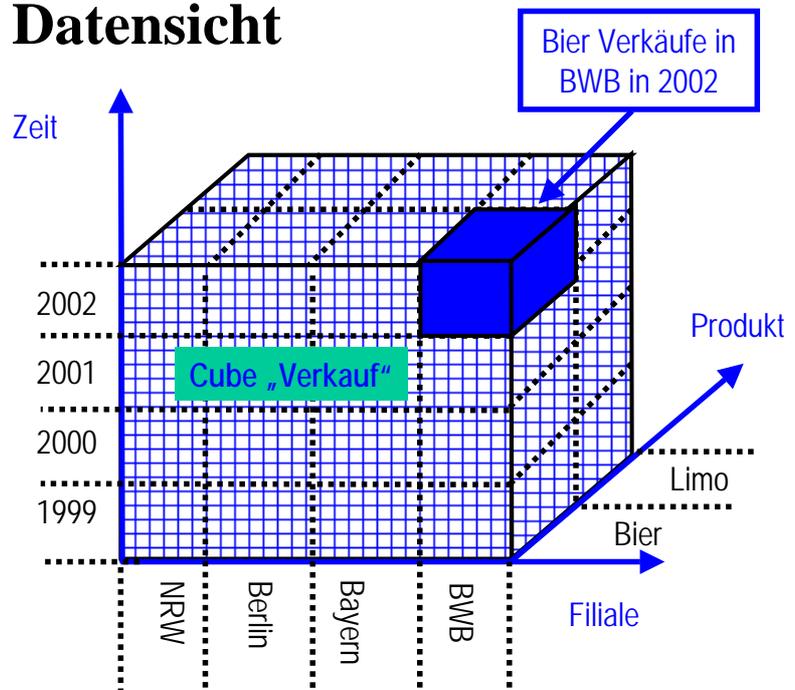


Mehrdimensionale Datensicht

- Data Cube:
Hyperwürfel
mit beliebig vielen
Dimensionen

- Kennzahlen
 - numerische Werte als
Grundlage für
Aggregationen /
Berechnungen

- Operationen
 - Aggregation der Kennzahlen über eine oder mehrere Dimension(en)
 - Slicing and Dicing: Bereichseinschränkungen auf Dimensionen
 - Drill-down and Roll-up: Wechsel der Dimensionsebenen



10

OLAP (Online Analytical Processing)

- interaktive multidimensionale Analyse auf konsolidierten Daten
- Merkmale / Anforderungen
 - multidimensionale, konzeptionelle Sicht auf die Daten
 - unbegrenzte Anzahl an Dimensionen und Aggregationsebenen
 - unbeschränkte dimensionsübergreifende Operationen
 - intuitive, interaktive Datenmanipulation und Visualisierung
 - transparenter (integrierter) Zugang zu heterogenen Datenbeständen mit logischer Gesamtsicht
 - Skalierbarkeit auf große Datenmengen
 - stabile, volumenunabhängige Antwortzeiten
 - Mehrbenutzerunterstützung
 - Client/Server-Architektur

11



Data Warehouse: Vor- und Nachteile

- Vorteile
 - Flexible, einfach zu nutzende, mehrdimensionale Datenanalyse
 - Hohe Datenqualität durch Integration, Bereinigung und Aggregation von Daten aus heterogenen Datenquellen
 - Performanz
 - Unabhängig von operativen Systemen
- Nachteile
 - Datenredundanz
 - Daten nicht vollständig aktuell
 - Hoher Administrationsaufwand
 - Hohe Kosten



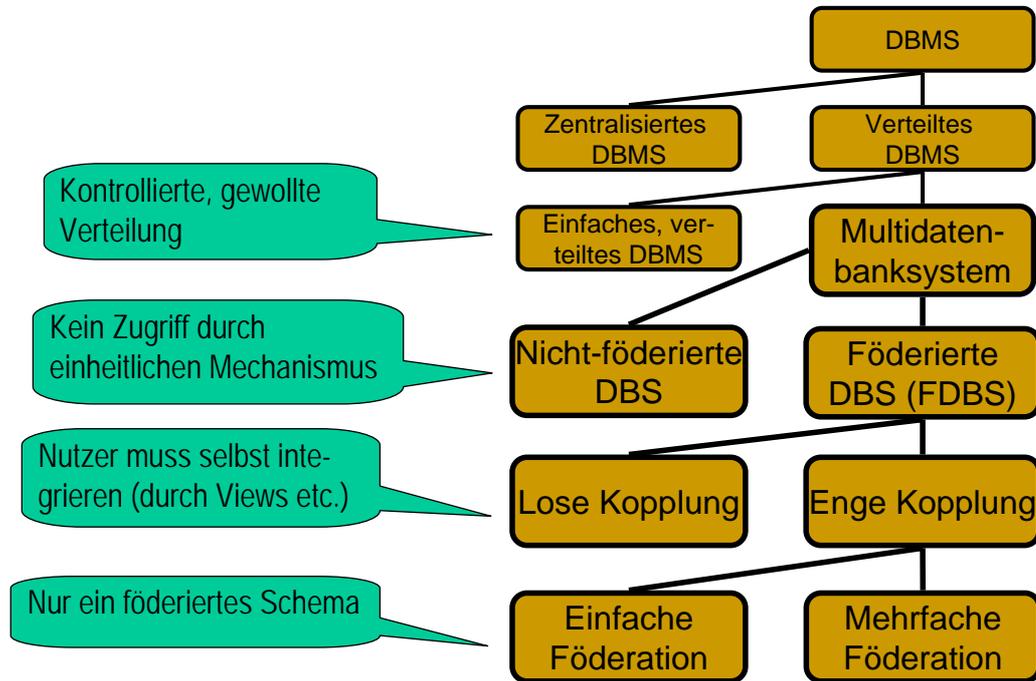
Föderierte Datenbanken

- Föderation = Bündnis
 - vergleiche politische Föderation (z.B. BRD, EU, USA)
- Föderierte Datenbank ist Gesamtsystem bestehend aus mehreren (teil-) autonomen Datenbanksystemen und ggf. einem zusätzlichen globalen (föderierten) Schema



Taxonomie von DBMS

- Nach: Amit P. Sheth and James A. Larson, Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases, ACM Computing Surveys, Vol. 22(3), 1990



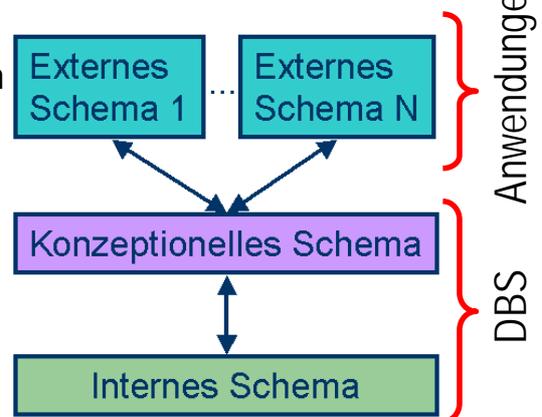
- Kontrollierte, gewollte Verteilung
- Kein Zugriff durch einheitlichen Mechanismus
- Nutzer muss selbst integrieren (durch Views etc.)
- Nur ein föderiertes Schema

14



3-Schichten-Architektur (Wdhg. DBS1)

- ANSI/SPARC 3-Schichten Architektur für zentralisierte DBMS
- Externe (logische) Sicht
 - Anwendungsprogramme
 - Nur auf die relevanten Daten
 - Enthält Aggregationen und Transformationen
- Konzeptionelle (logische) Sicht
 - Unabhängig von physischer Sicht
 - Definiert durch Datenmodell
 - Stabiler Bezugspunkt für interne und externe Sichten
- Interne (physische) Sicht
 - Speichermedium (Tape, Festplatte)
 - Speicherort (Zylinder, Block)

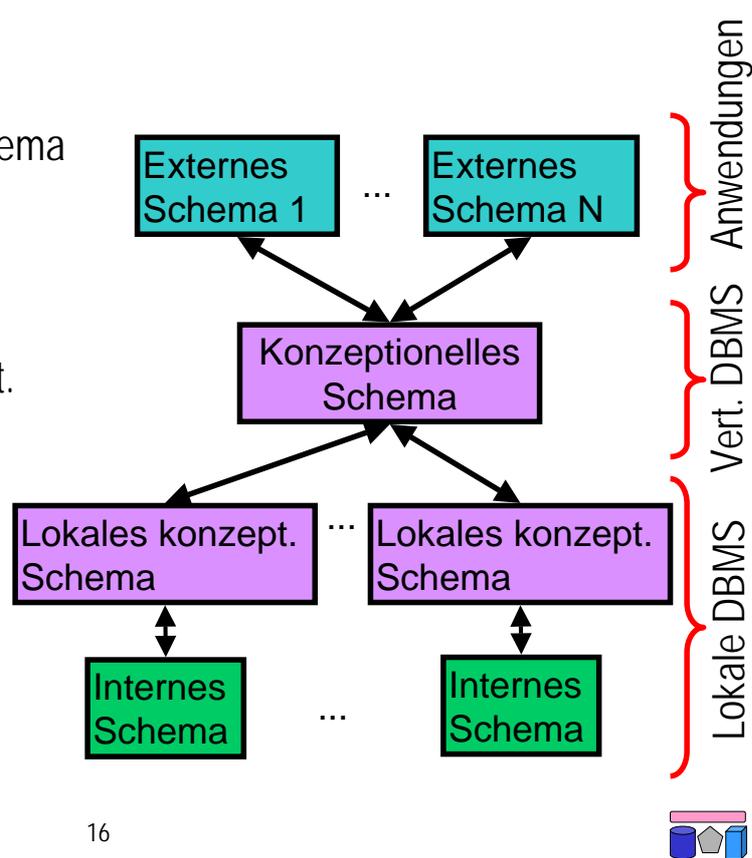


15



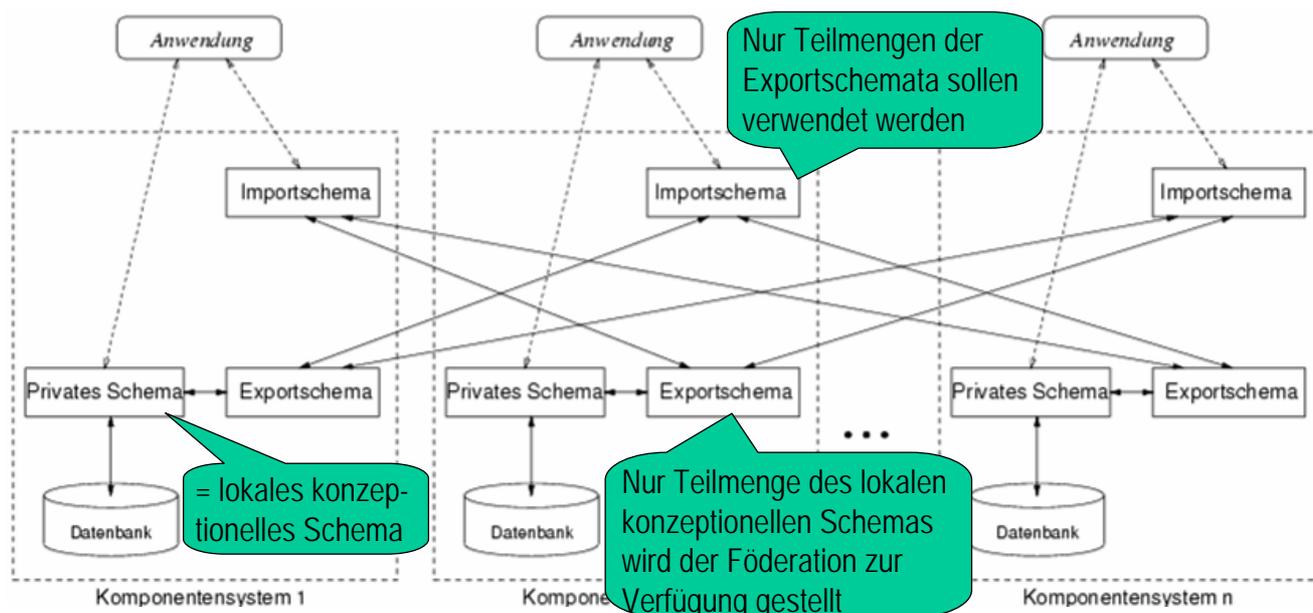
4-Schichten-Architektur (Verteilte DBMS)

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



16

Import/Export-Architektur [HM85]



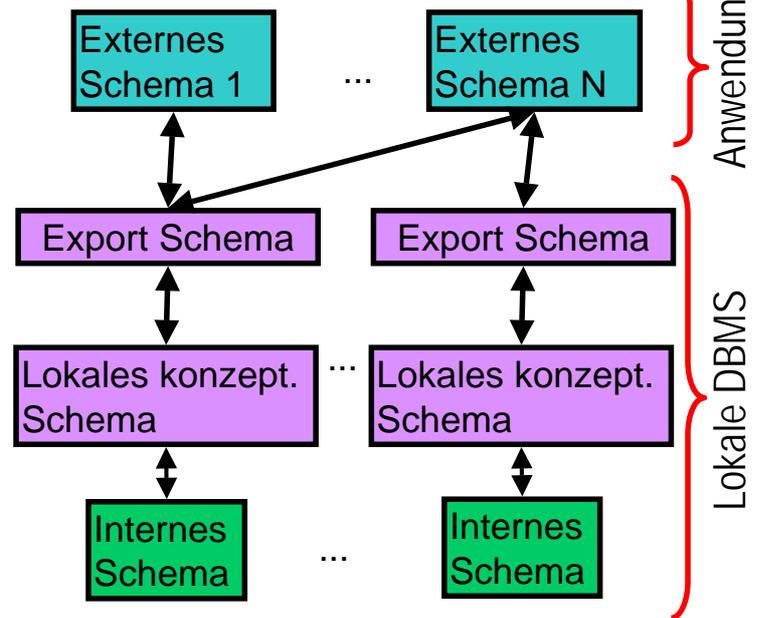
[HM85] Dennis Heimbigner, Dennis McLeod: *A Federated Architecture for Information Management*. ACM Trans. Inf. Syst. 3(3): 253-278 (1985)

17



4-Schichten-Architektur (Multidatenbanken)

- Auch: Multidatenbankarchitektur [LMR90]
- Voraussetzung
 - Nutzer kennen die jeweiligen Schemas
 - Multidatenbanksprache
- Anwendungen müssen selbst integrieren
- Lose Kopplung



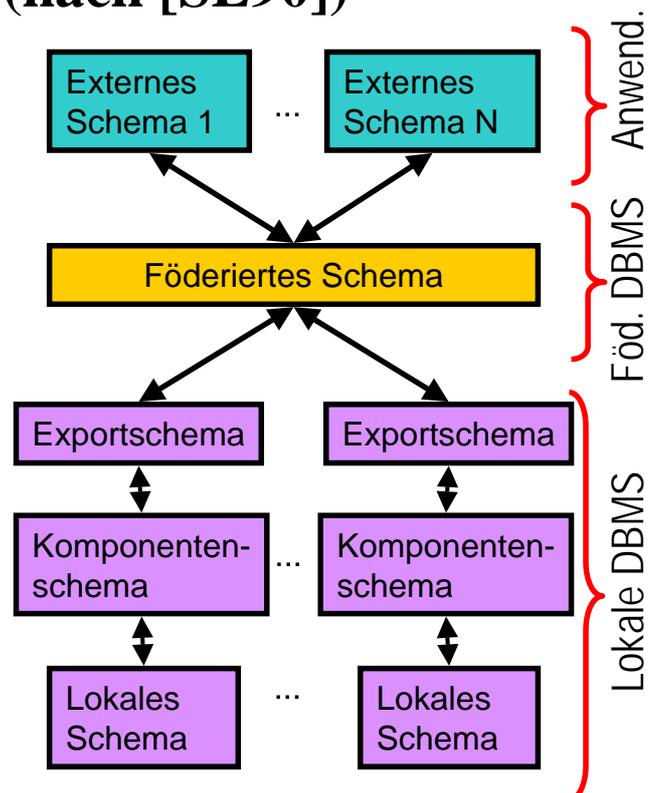
[LMR90] W. Litwin, L. Mark, N. Roussopoulos: *Interoperability of Multiple Autonomous Databases*, ACM Computing Surveys, Vol. 22(3), pp267-293, 1990.

18



5-Schichten Architektur (nach [SL90])

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

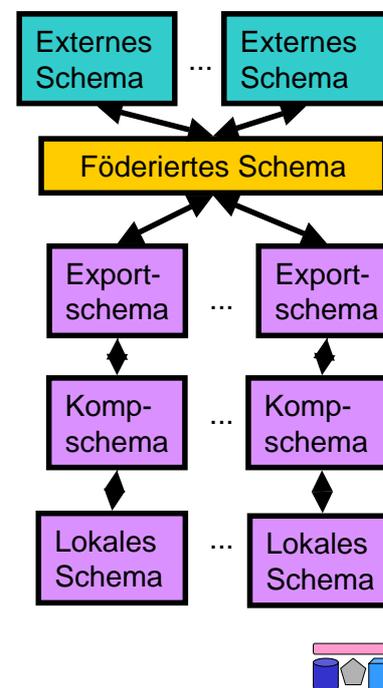


[SL90] Amit P. Sheth and James A. Larson, *Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases*, ACM Computing Surveys, Vol. 22(3), 1990



5-Schichten Architektur

- Externes Schema
 - Anwendungsabhängig, Zugangskontrollen
- Föderiertes Schema (auch globales, unified, Enterprise Schema)
 - Integriert aus den Exportschemata
 - Kennt Datenverteilung
 - Föderiertes Schema kann sehr groß sein
→ Vereinfachung im Exportschema
- Exportschemata
 - Teilmenge des jeweiligen Komponentenschemas (mit Zugangsberechtigungen)
 - Unnötig, wenn komplettes Schema exportiert wird
- Komponentenschemata
 - Kanonisches Datenmodell
 - Fügt fehlende Semantik hinzu; Übergang durch Mappings
 - Unnötig, wenn lokales = kanonisches Datenmodell
- Lokale Schemata
 - Konzeptionell



Vergleich: Föderierte Datenbanken

	Föderierte Architektur Import/Export	Föderierte Datenbank 4-Schichten	Föderierte Datenbank 5-Schichten
Art der Integration			
Integration durch			
Zugriff über			
DBMS-Funktionalität			

Föderierte Datenbanken: Vor- und Nachteile

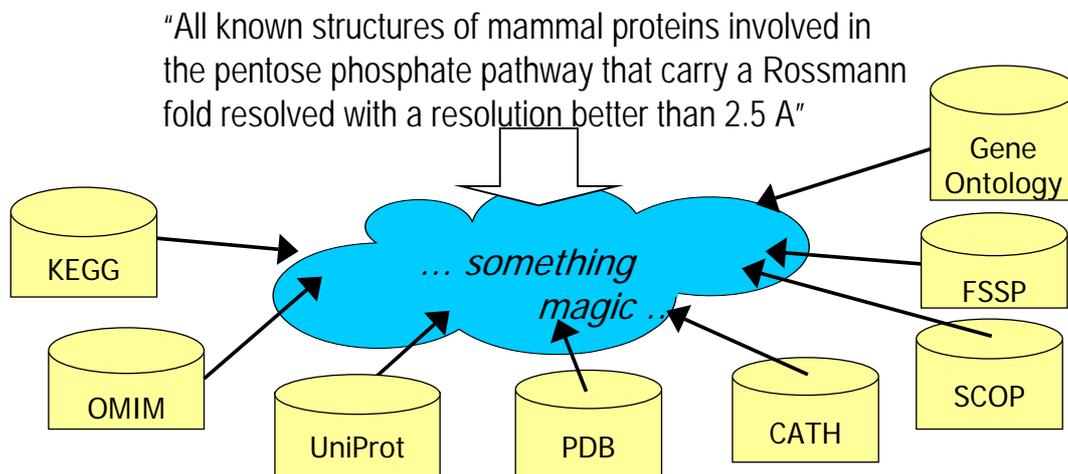
- Vorteile
 - Einheitlicher Zugriff durch Datenbanksprache (z.B. Schema-SQL)
 - Anfrageoptimierung
 - Volle Anfragemächtigkeit bzgl. Exportschema
 - Auch schreibender Datenzugriff (prinzipiell) möglich
- Nachteile
 - Nur DBMS als Quellen

23



Mediator-Wrapper-Architekturen

- Ein Mediator ist eine Softwarekomponente, die Wissen über bestimmte Daten benutzt, um Informationen für höherwertige Anwendungen zu erzeugen. (nach [Wie92])
- Wrapper sind Softwarekomponenten, die die Kommunikation und den Datenfluss zwischen Mediatoren und Datenquellen herstellen.



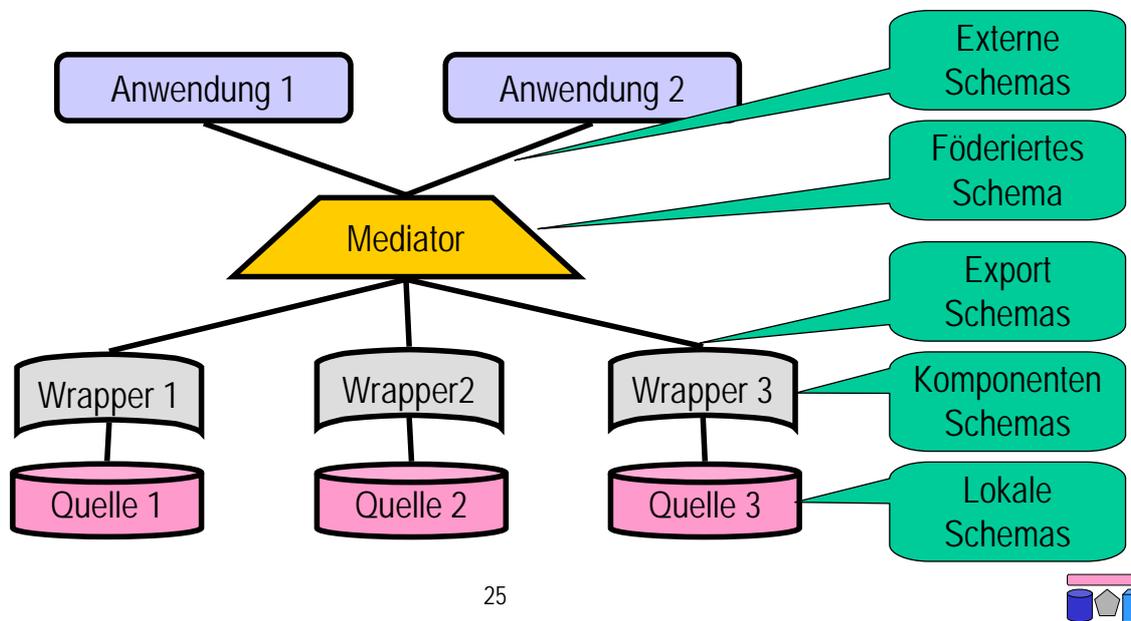
[Wie92] Gio Wiederhold: *Mediators in the Architecture of Future Information Systems*.
IEEE Computer Journal, 25(3), 38-49, 1992

24



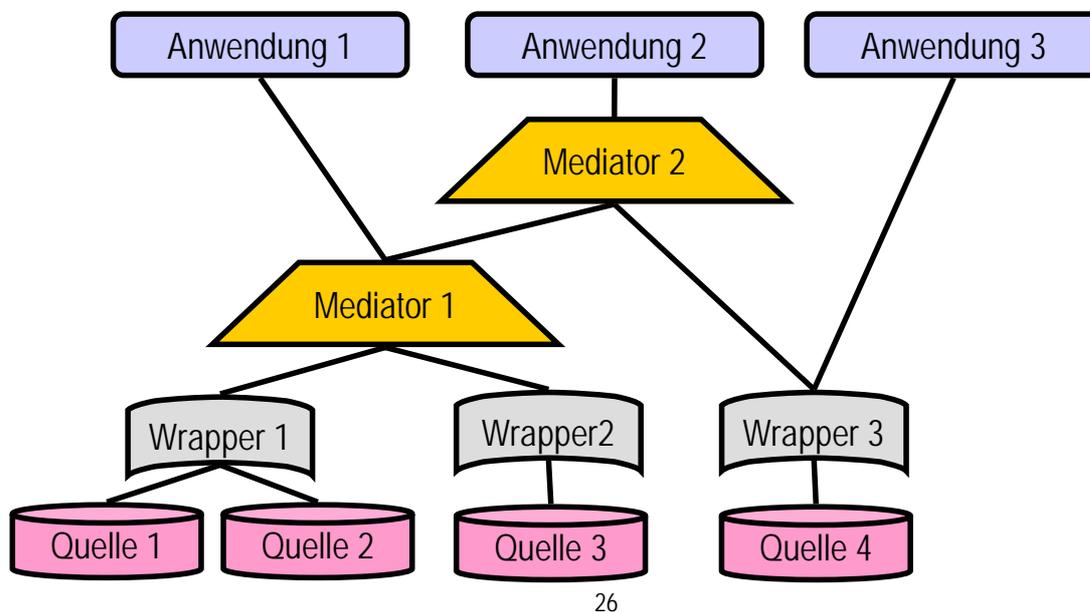
Aufbau u. Vergleich zur 5-Schichten-Architektur

- Schichtenaufbau
- Anwendungen, Mediator und Wrapper sind autonome Systeme
- Kommunikation zwischen Systemen = Anfrage senden + Ergebnis erhalten



Aufbau (verallgemeinert)

- Schematisch gleiche/ähnliche Quellen können sich Wrapper "teilen"
- Mediatoren dienen als Quelle für Mediatoren → gestufter Aufbau
- Anwendungen können direkt mit Wrapper (Quelle) kommunizieren



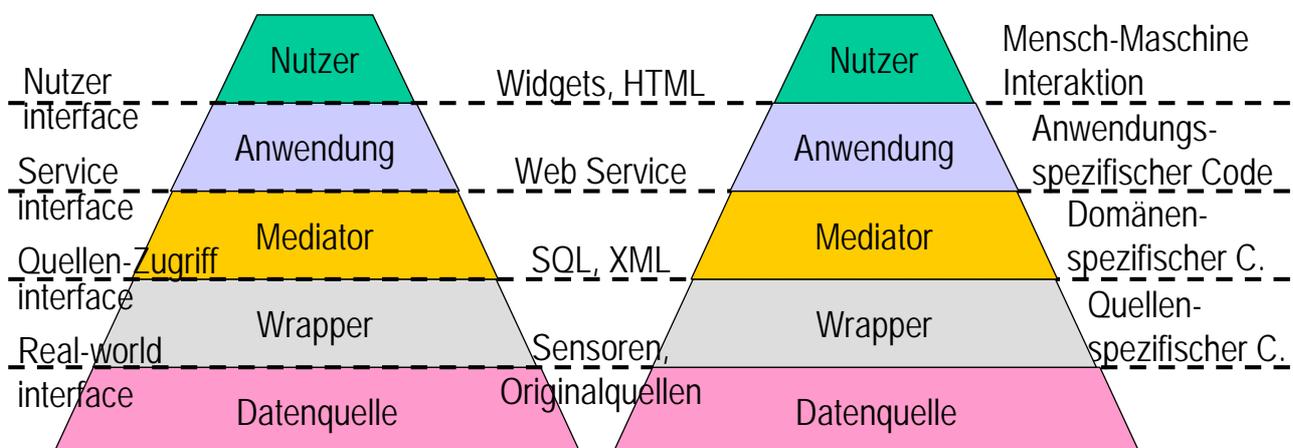
Vergleich: Mediator vs. Wrapper

- Mediator
 - Überwindet strukturelle und semantische Heterogenität
 - Führt Integration und Anfrageplanung aus
 - Benötigt oftmals Domänenwissen
 - Regelbasiert, deklarative Techniken
- Wrapper
 - Überwindet technische und Datenmodellheterogenität
 - Sollte im Rahmen wieder verwendbar sein
 - Schlanke Wrapper erwünscht
 - Quellspezifisch, kein Domänenwissen
 - Engineering, hoch-spezifisch

27



Funktionale Schichten und Schnittstellen



28



Mediatoren

- Ein Mediator sollte klein und einfach genug sein, um durch einen einzigen oder höchstens eine kleine Gruppe von Experten gewartet werden zu können
 - einfaches föderiertes Schema
 - begrenzte Domäne
 - einfache Schnittstellen
- Trennung unterschiedlicher Aufgabenbereiche
 - „Zerlege große Probleme in viele kleine Probleme“
 - Klare, begrenzte Aufgabenbereiche
 - Können unabhängig von einander entwickelt und betrieben werden
 - KISS – „Keep it as simple as possible“

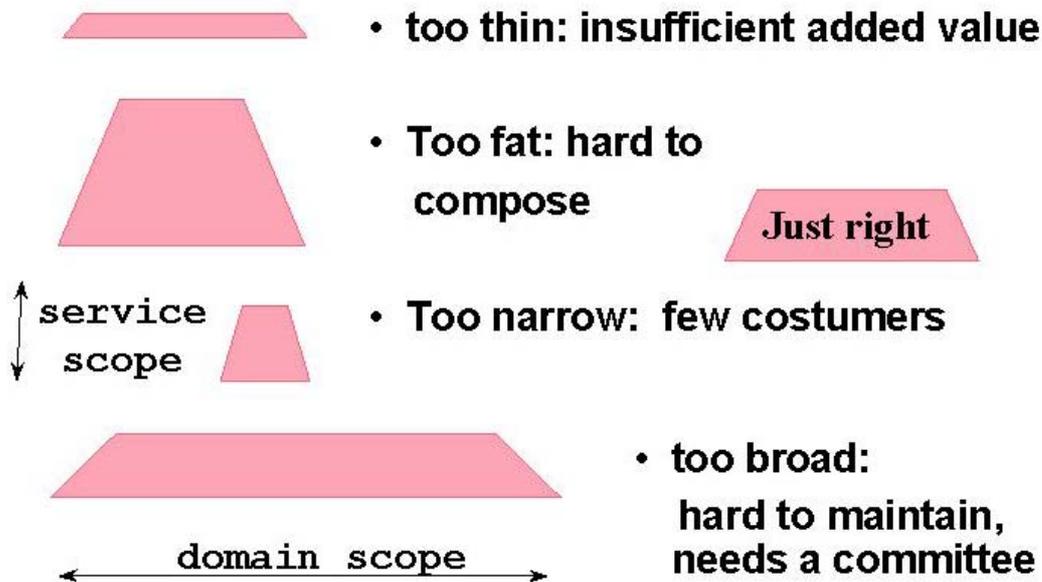


Mediatoren: Funktionen

- Erbracht durch Domänen-Experten
 - Suche und Auswahl relevanter Informationsquellen
 - Auswahl und Ranking gemäß Anforderungen und Qualität
 - Transformationen
 - Quell- und Zielformate, Konsistenzerhaltung
 - Anreicherung mit Metadaten
 - Einordnung im neuen Kontext
 - Bewertung der Qualität, Vollständigkeit, Konsistenz, ...
 - Abstraktion (zum besseren Verständnis, als Auswahl)
 - Aggregation, Auswahl relevanter Daten, Zusammenfassungen
 - Integration verschiedener Quellen
- Von allgemeinen Daten zu spezifischen Informationen
 - Anwendungs- und Kontextabhängigkeit



Dicke und dünne Mediatoren



Gio Wiederhold 1999 93

31



Wrapper: Definition und Eigenschaften

- Softwarekomponenten, die die Kommunikation und den Datenfluss zwischen Mediatoren und Datenquellen herstellen
- "vermitteln zwischen Mediator und Quelle"
- Eigenschaften
 - jeweils spezialisiert auf eine Ausprägung autonomer, heterogener Quellen
 - Haben ein Export-Schema
 - Übersetzen Anfragen und Ergebnisse zwischen Mediator (kanonisches Datenmodell, globale Anfragesprache) und Quelle (quellspezifisches Datenmodell und Anfragesprache)

32



Wrapper: Aufgaben

- Lösen Schnittstellenheterogenität
 - Technisch
 - Anfragesprache (SQL, HTML Formulare, http, CORBA, ...)
 - Z.T Mächtigkeit der Anfragesprache
- Lösen Datenmodellheterogenität
- Lösen evtl. schematische Heterogenität → kanonisches Schema
- Lösen *nicht* strukturelle und semantische Heterogenität
- Reduzieren Komplexität im Mediator
 - Reduzieren Anzahl der Datenmodelle
 - Reduzieren Anzahl der Schemata
- Einfache Datenkonvertierung (Übersetzungstabelle, Umrechnung, ...)
- Unterstützen globale Optimierung
 - Kostenmodell, Anfragefähigkeiten

33



Wrapper: Anforderungen

- Sollten schnell implementiert werden können (< 1 Woche)
 - Ziel: Schnelle einfache Implementierung (Proof-of-concept)
 - Möglichkeit zur sukzessiven Verbesserung
- Sollten wiederverwendbar sein
- Lokale Wartung (bei föderierten Systemen)

- An den Wrappern scheitern viele Projekte!
 - Geringer Automatisierungsgrad – zu quellspezifisch
 - Überwindung der Datenmodellheterogenität nicht einfach
 - Viele Quellen – viele Wrapper – viel Aufwand
 - Ohne „gute Wrapper“ keine guten integrierten Daten
 - Garbage in – garbage out
- Deshalb
 - Forschung zur schnellen, (semi-) automatischen Wrappergenerierung
 - Wrapperbibliotheken

34



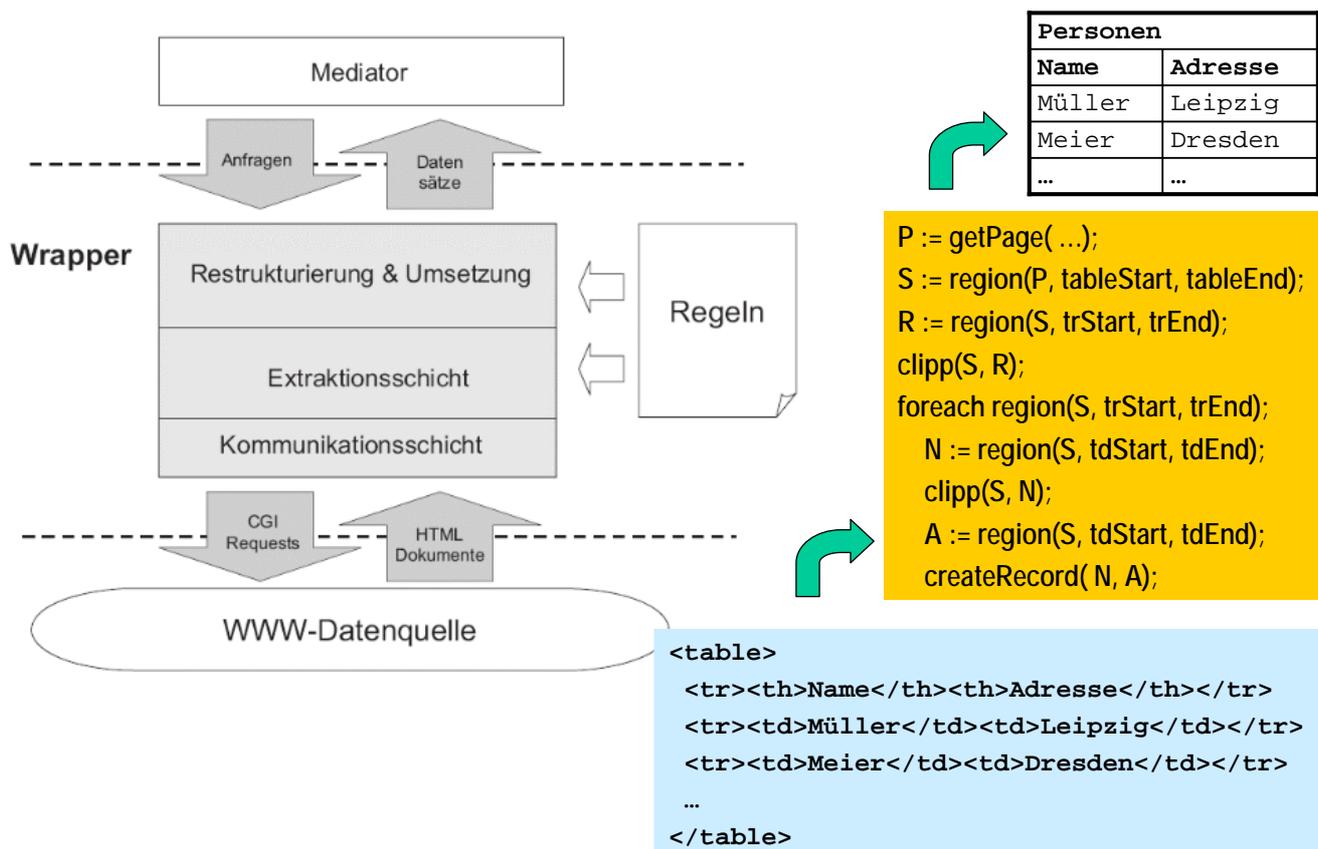
Beispiel: Web Wrapper

- „Screen Scraping“
- Umsetzung von Mediatoraufrufen in HTTP Requests
 - Die zu CGI/Servlet/PHP... Requests werden
 - Die meistens zu SQL Requests führen ...
- Extraktion der gewünschten Daten aus HTML Seiten
- Umwandlung in das Datenmodell des Mediators
- Kommerziell interessant
 - Trendanalysen (Blogs), Beobachtung der Konkurrenz, Business Intelligence, Meta-X (Produktkataloge, Reisen, Paper, ...)
 - Firmen: Lixto, W4F, ...
- Juristische Aspekte zu beachten

35



Web Wrapper: Aufbau und Funktionsweise



Web Wrapper: DOM vs. RegEx

- Zwei unterschiedliche Herangehensweisen zur Extraktion
- DOM (Document Object Model, hierarchische Struktur der HTML-Seite)
 - Kann automatisch erstellt werden
 - Speicherplatzintensiv
 - Langsam (insbesondere wenn nur wenige Daten extrahiert werden sollen)
 - Gut bei regulären Strukturen (datenbankerzeugte Webseiten)
 - Schwierigkeiten mit irregulärem HTML
 - Anfällig für Änderungen in der Reihenfolge oder Einschübe neuer Elemente
 - Kann Textfelder nicht weiter unterteilen (also doch reguläre Ausdrücke nötig)
- Regulärer Ausdrücke (RegEx, regular expressions)
 - Schnelle Ausführung
 - Unübersichtlich
 - Kompliziert, wenn geschachtelte Elemente benötigt werden
 - Benötigen stabile Anker Elemente (Überschriften etc.)

37



Web Wrapper

- Echte Web-Wrapper-Sprachen müssen mehr können
 - Webpage-Wrapping versus Website-Wrapping
 - Verfolgen von Links und Nachladen von Seiten
 - Rekursive Verfeinerungen
 - Sukzessive Anwendung immer genauerer regulärer Ausdrücke
 - Abbildung extrahierter Daten (Listen, Terme, Tabellen) auf ein Exportschema
 - Variablen, Funktionen, Schleifen, Fehlerbehandlung
 - Einfache Anfragefunktionalität
 - `Wrapper.get(„SELECT adresse FROM personen WHERE name=„Meier“);`
 - „Erraten“ von Wrappern (Wrapper Induktion)
 - ...
- Ganz schwierig
 - JavaScript, Web 2.0 (AJAX)

38



Mediator-Wrapper-Ansatz: Vor- und Nachteile

- Vorteile
 - Flexibler Integrationsansatz mit stets aktuellen Daten (virtuelle Integration)
 - Geringe Anforderungen an Quellen (hohe Autonomie)
 - Integrationssystem modular erweiterbar (neuer Wrapper, neuer Mediator)
- Nachteile
 - Nur begrenzt skalierbar
 - Anwendungen mit max. 7 ± 2 Mediatoren; Mediatoren für max. 7 ± 2 Quellen [Wie99]
 - Regelmäßige Wartung/Anpassung der Wrapper notwendig
 - Anfragemöglichkeit ggf. beschränkt durch Quellen
 - Sicherstellung der Datenqualität (u.a. Duplikaterkennung) zur Laufzeit (Performanz)



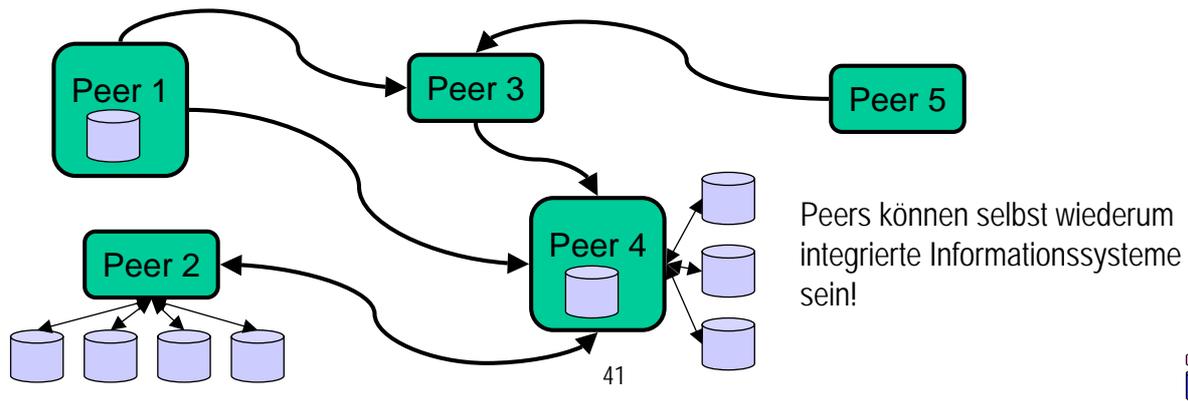
Peer-Data-Management-Systeme: Szenario

- Genomdaten
 - Forscher haben den Willen (und die Pflicht), Daten weltweit zu veröffentlichen.
 - Komplexe Schemata und komplexe Anfragen
 - Bekannte Zusammenhänge zwischen den Daten
 - Bildung eines globalen Schemas nicht immer einfach
- Gesundheitsinformationssystem
 - Krankenhausdaten auf vielen Systemen verteilt
 - Ärzte wollen manche Daten verbreiten, andere nicht.
 - Content-Management-artige Suche ist wichtig.
 - Verschiedenste und komplexe Schemata
 - Mehrwert (für Patienten) durch Teilen der Daten

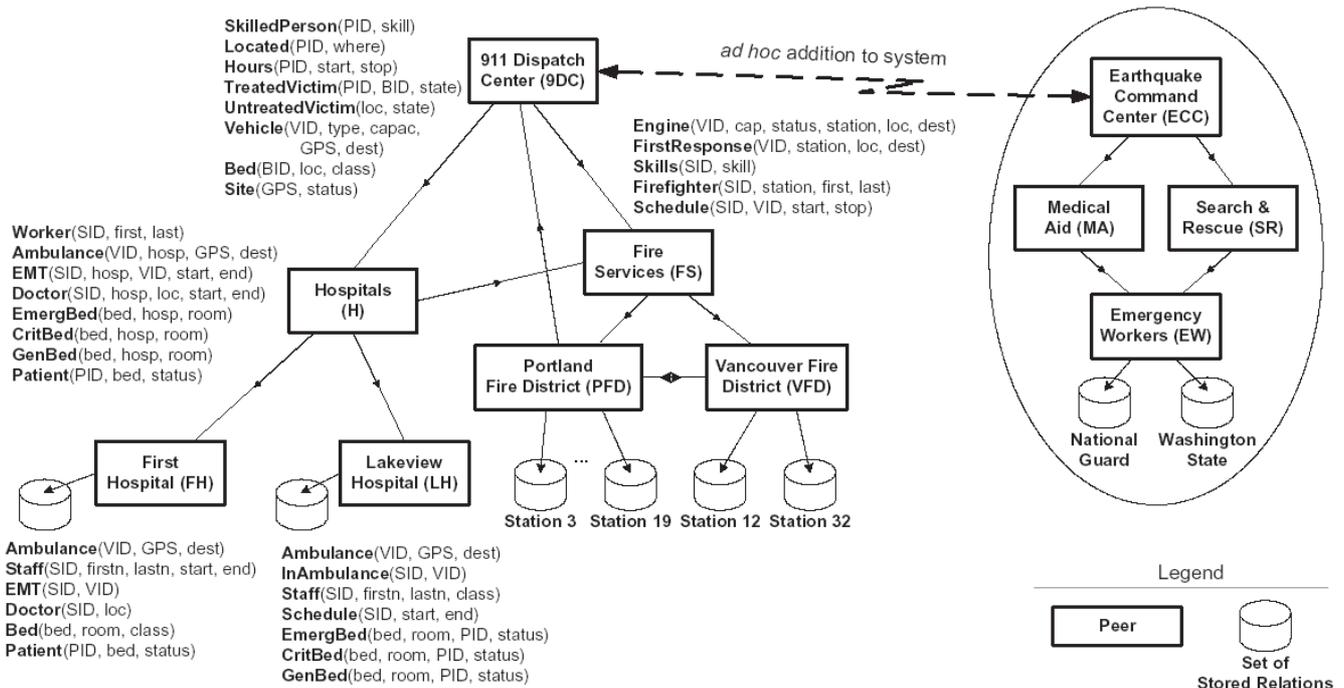


Peer-Data-Management-Systeme

- Grundidee: Strukturierte Peer-2-Peer Netzwerke ("P2P mit SQL")
- Jeder Peer kann
 - Daten exportieren (= Datenquelle)
 - Sichten auf Daten zur Verfügung stellen (= Wrapper)
 - Anfragen anderer Peers entgegennehmen und weiterleiten (= Mediator)
 - Anfragen stellen (Benutzer)
- Verknüpfungen nicht zwischen lokalen und globalem Schema, sondern zwischen Paaren von Peer-Schemata (kein globales Wissen)



Peer-DMS am Beispiel von Piazza [HIST03]



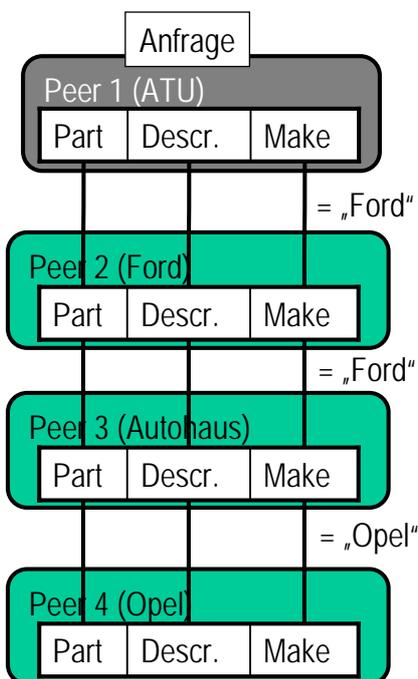
Peer-to-Peer-Systeme vs. Peer-DMS

	Peer-to-Peer	Peer-Data-Management-System
Granularität der Daten	Niedrig, nur ganze Dateien	
Anfragen	Einfache Keyword-Suche auf Dateinamen und -metadaten	
Anfrageergebnis	Unvollständig (best-effort)	
Anfrageverteilung	Gemäß (proprietärem) Protokoll	
Schema	Einfach (Dateiname, dateiabhängige Metadaten)	
Dynamik	Sehr hoch, ständiger Bei- und Austritt von Peers	
Skalierbarkeit	Millionen von Peers ⁴³	



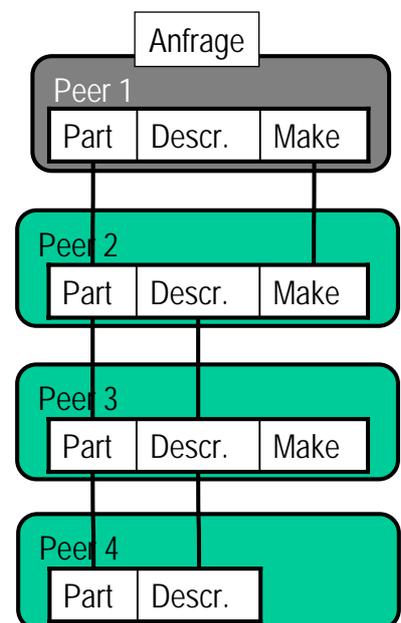
Mapping-Komposition

- Komposition zur Propagation von Anfragen
 - Bei jedem Mapping/ Zwischenschritt geht potentiell Information verloren
 - Kann nicht durch spätere Schritte ausgeglichen werden



Probleme

- Kumulierte Selektion (links)
- implizit in Schemata
 - explizit in Mappings
 - Punkt- und Bereichs-Selektionen
- Kumulierte Projektion (rechts)
- in Schemata
 - in Mappings



Peer-DMS: Vor- und Nachteile

- Vorteile
 - Nutzer müssen nur eigenes Schema kennen
 - Dennoch sind weitere Daten (über transitive Hülle der Mappings) verfügbar
 - Neue Schemas können leicht und inkrementell hinzugefügt werden.
 - Mapping nur zum ähnlichsten Schema nötig
- Nachteile / Probleme
 - Mapping Komposition
 - Effizienz (bei vielen Zwischenstationen)
 - Informationsqualität: Schleichender Informationsverlust über mehrere Mappings
 - Vollständigkeit des Ergebnisses nicht gewährleistet
 - Vertrauen in alle Peerdaten



Vergleich von Integrationsarchitekturen

	Data Warehouse	Föderierte Datenbank	Mediator-Wrapper	Peer-DMS
Instanzintegration				
Zeit Duplikat-erkennung				
Globales Schema				
Zeit Schema-integration				
Anfrage bzgl.				
Datenqualität				
Skalierbarkeit (#Quellen)				
Analyse großer Datenmengen				
Autonomie der Quellen				
Aktualität der Daten				



Weitere Architekturen

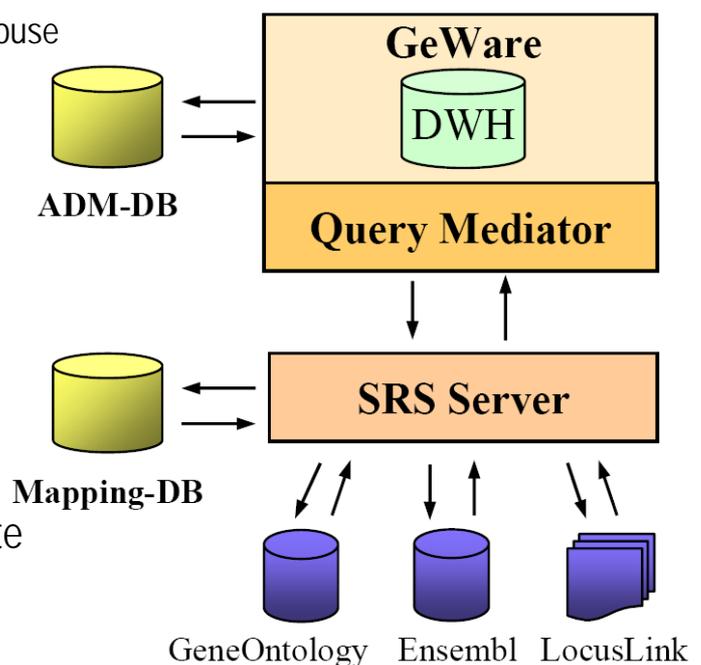
- Jede Architektur hat Vor- und Nachteile bzw. Stärken und Schwächen
- Kombination oder Modifikation von "Standard-Ansätzen" für bestimmte Anwendungen sinnvoll
- Beispiel 1: Materialisierte und virtuelle Integration
 - DWH kombiniert mit Mediator
 - Prototyp: GeWare-Erweiterung
- Beispiel 2: Instanzbasierte P2P-artige Verknüpfung
 - P2P-Verknüpfung kombiniert mit Mediator-Steuerung und generischem Warehouse
 - Prototyp: iFuice

49



Beispiel 1: Materialisiert + Virtuell

- GeWare-Erweiterung [KDKR05]
 - GeWare = Gene Expression Warehouse
- Data Warehouse enthält experimentelle Daten für performante Analysen
- Aktuelle Annotationsdaten können bedarfsgesteuert über Mediator inzugezogen werden
- Mapping-Datenbank enthält vorberechnete Verbindungen zwischen Objekten für performante Join-Berechnung

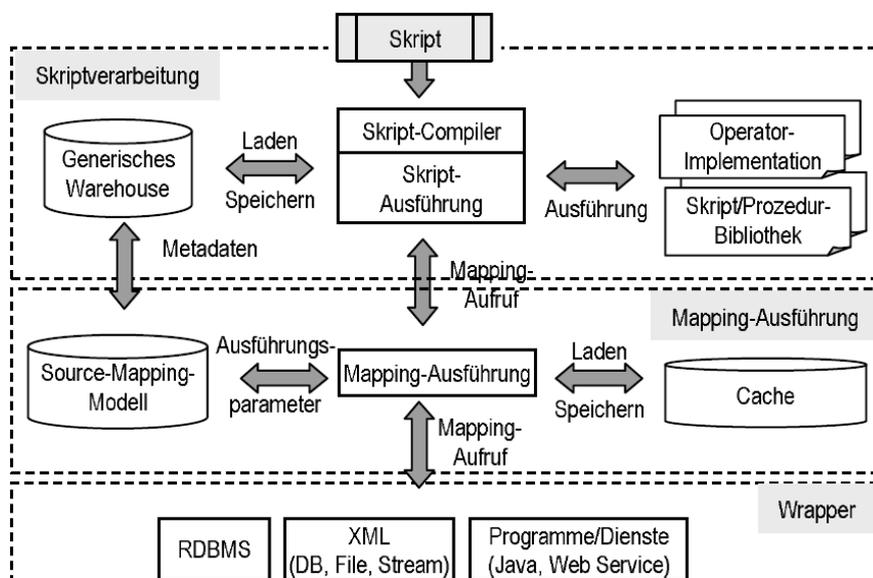


[KDKR05] Kirsten, T., Do, H. H., Körner, Ch., Rahm, E.: *Hybrid Integration of molecular-biological Annotation Data*. Proc. of DBILS, 2005



Beispiel 2: Instanzbasierte P2P-artige Verknüpf.

- iFuice [RT05] basiert auf P2P-Verknüpfung von Objekten via Mappings
 - Information Fusion utilizing Instance Correspondences and Peer Mappings
- Mediator ermöglicht bedarfsgesteuerte Verarbeitung in Skripten
 - keine deklarative Anfragesprache sondern Datenfluss (ähnlich ETL bei DWH)
- Speicherung aller Daten (Objekte, Mappings) in generischen Warehouse



[RT05] Rahm, Thor, et. al. :
iFuice - Information Fusion utilizing Instance Correspondences and Peer Mappings. In Proc. of WebDB, 2005



Architekturen vs. Integrationsaufgaben

- Integrations(teil)aufgaben
 - Anfrageübersetzung: Überführung von Anfragen von Schema A nach Schema B
 - Schemaintegration: Bildung eines globalen Schemas aus Quellschemata
 - Schema Matching: Ermitteln von Korrespondenzen zwischen Elementen zweier Schemata (=Schema Mapping)
 - Datentransformation: Überführung von Instanzdaten mit Schema A nach Schema B
 - Duplikaterkennung: Identifikation von "gleichen" Datensätzen
 - Konfliktbehandlung: Auflösung widersprüchlicher/fehlender Informationen

	DWH	Föd-DBMS	Med.-Wrap.	Peer-DMS
Anfrageübersetzung				
Schemaintegration				
Schema Matching				
Datentransformation				
Duplikaterkennung				
Konfliktbehandlung				

+ relevant
 o evtl. relevant
 – nicht relevant

Kapitel 5-7
 betrachten
 Integrations-
 aufgaben



Zusammenfassung

- Übersicht über Arten von Informationssysteme
 - U.a. bzgl. Integrationsart und Verwendung eines globalen Schemas
- Vorstellung und Vergleich typischer Integrationsansätze
 - Data Warehouse
 - Föderierte Datenbanken
 - Mediator-Wrapper-Ansatz
 - Peer Data Management Systeme
- Hybride Ansätze
 - Kombination der Stärken einzelner Architekturen für bestimmte Einsatzzwecke
- Architekturen vs. Integrations(teil)aufgaben
 - Anfrageverarbeitung → Kap. 5
 - Schemamanagement → Kap. 6
 - Informationsfusion → Kap. 7

