Informationsintegration

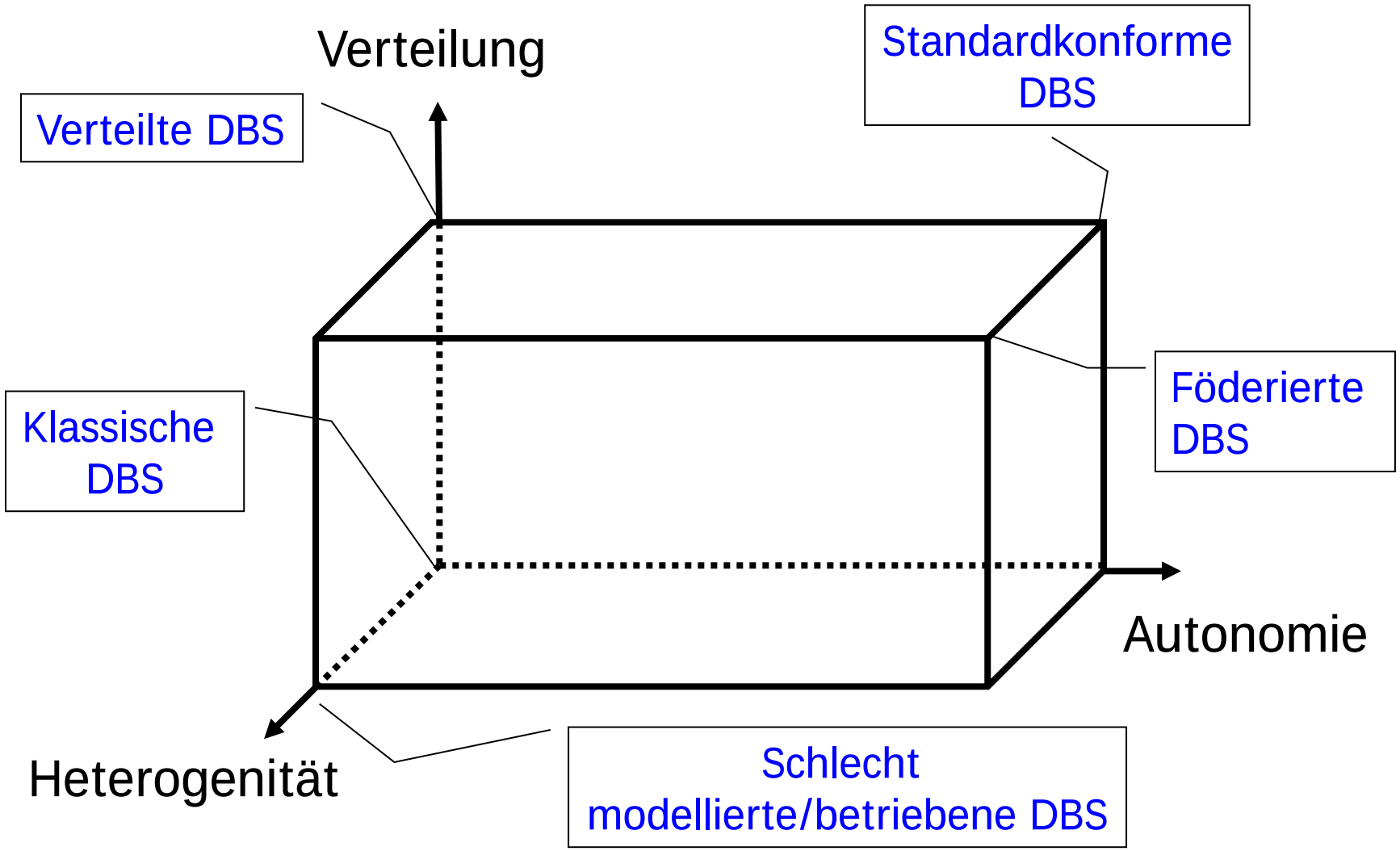
Grundlegende Architekturen

Ulf Leser

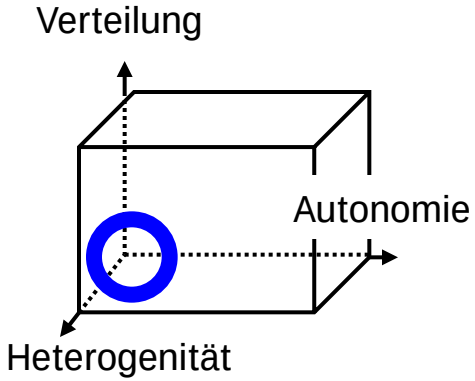
Inhalt diese Vorlesung

- Klassifikation verteilter, autonomer, heterogener Systeme
- Weitere Klassifikationskriterien
- Schichtenaufbau integrierter Systeme

Klassifikation [ÖV91]

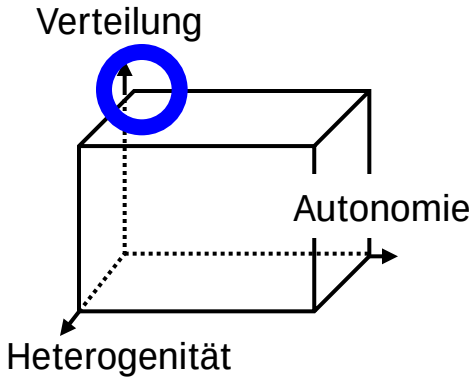


Zentrale Datenbank



- Normalfall – homogene, zentrale Datenbank
- Daten/Berechnung können trotzdem begrenzt verteilt sein
  - Partitionierung, RAID, SAN
  - Parallele Datenbanken
- Datenbank entsteht aus homogenem Entwurf
  - Wenn Redundanz / Heterogenität, dann mit Absicht und kontrolliert
  - Problem: Weiterentwicklung (Evolution)
- Zentrale Kontrolle und Administration

Verteilte Datenbanken



- Daten sind physisch verteilt
  - Absichtsvolle, kontrollierte, a-priori Verteilung
  - Homogenes aber verteilt implementiertes Schema
- Knoten haben keine Autonomie
- Heterogenität wird unterdrückt
- Ortstransparenz, aber keine Verteilungstransparenz
  - Aliase und Proxy kapseln entfernte Orte
  - Verteilungstransparenz durch Sichten möglich, aber nicht durch System erzeugt

Einschub

- Oracle-DBs können auf andere Oracle-DBs zugreifen
- Database Links
  - **CREATE [PUBLIC] DATABASE LINK <link_name>**
    CONNECT TO <user_name> <IDENTIFIED BY <password>
    USING '<service_name>';
    - service_name muss über Konfigurationsfiles aufgelöst werden
  - **SELECT col1, col2, … FROM tab1@link_name;**
    - Zugriff wie auf lokale Tabelle (Joins, Selektion, Projektion, ...)
    - Transparenz durch Sicht möglich
      - **CREATE VIEW myview AS SELECT * FROM tab1@link_name;**
- Anwendung z.B. zur automatische Replikation

Gateways

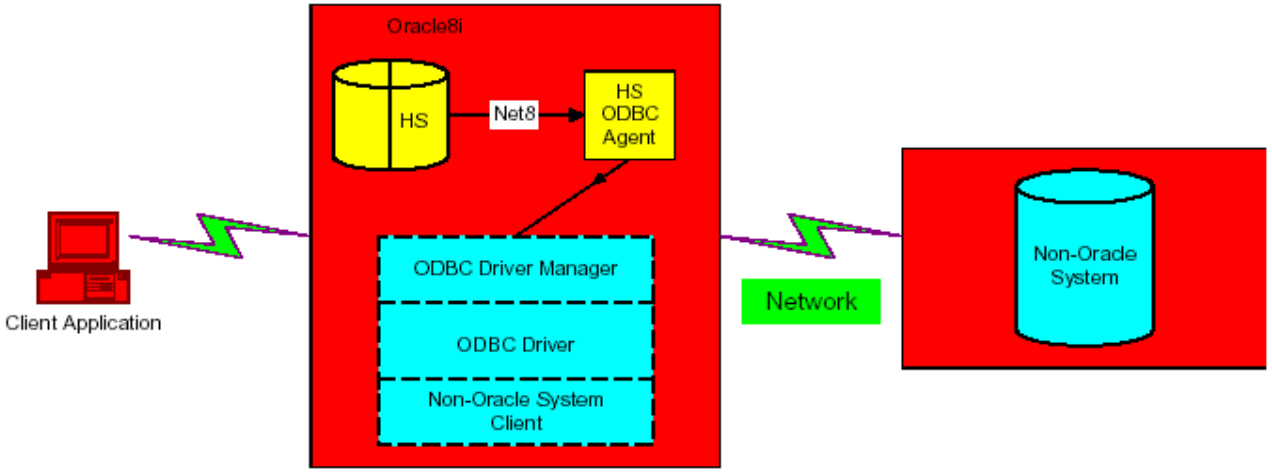
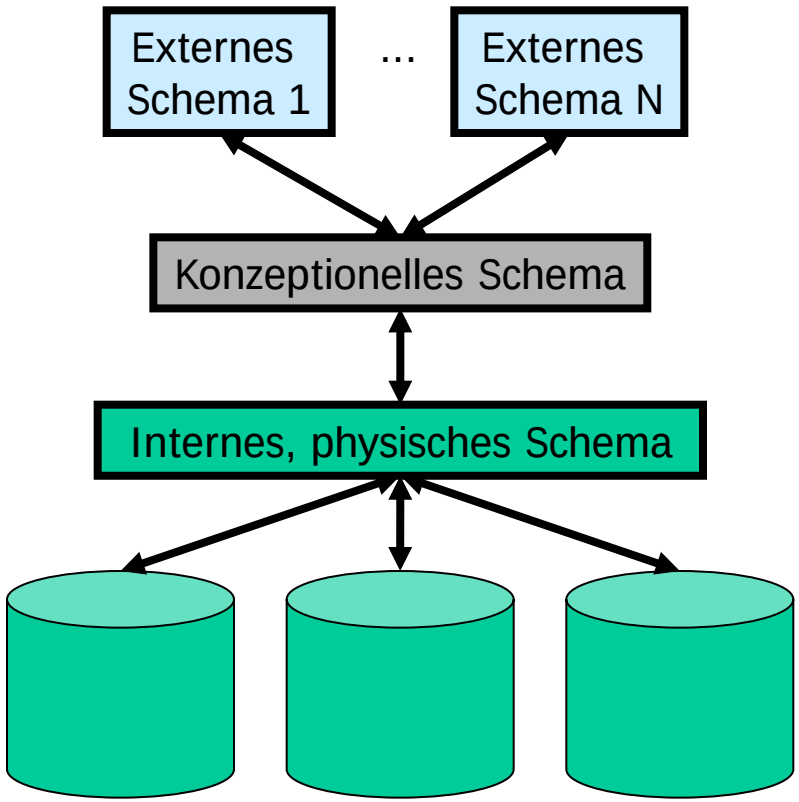


Figure 1: Oracle8*i* Release 8.1.6 ODBC Generic Connectivity Configuration Example

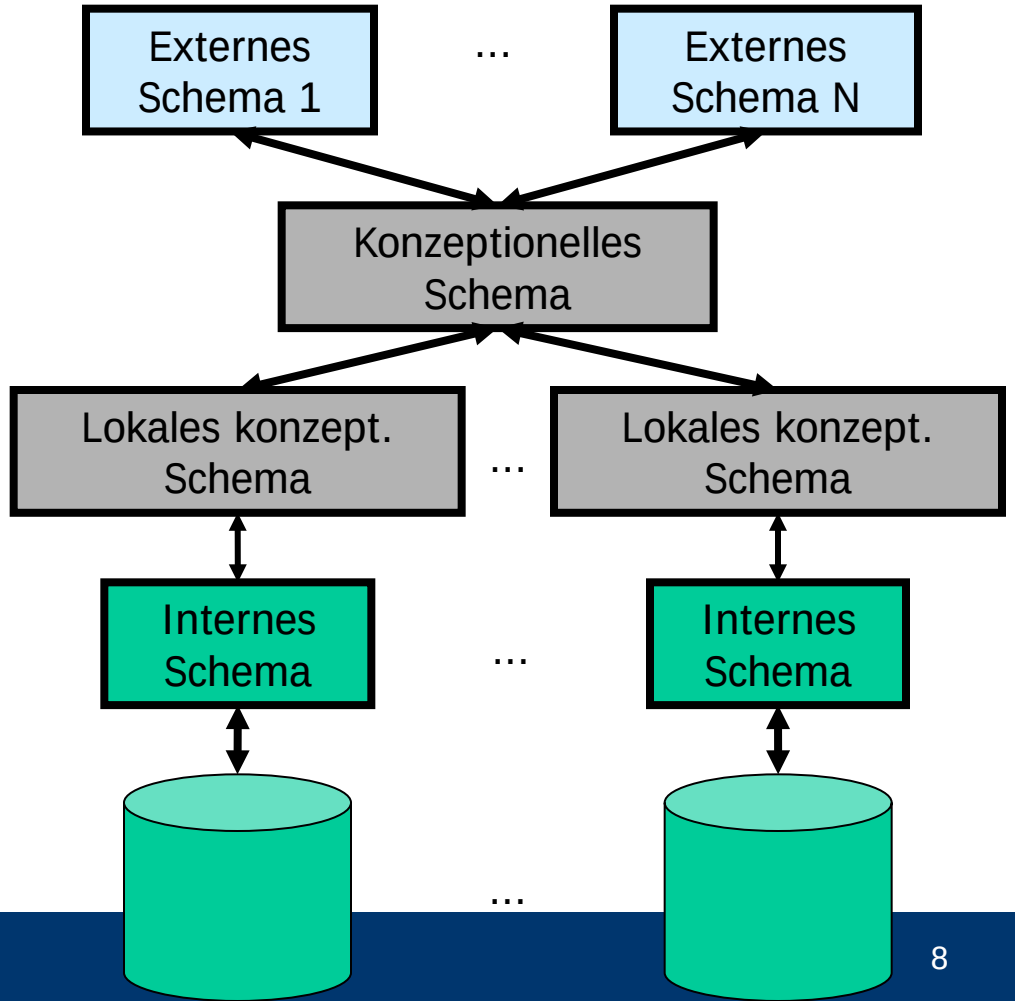
- Zugriff z.B. über ODBC
- Umwandlung von Datentypen
- Unterstützung verschiedener SQL-Dialekte

Verteilte versus parallele Datenbanken

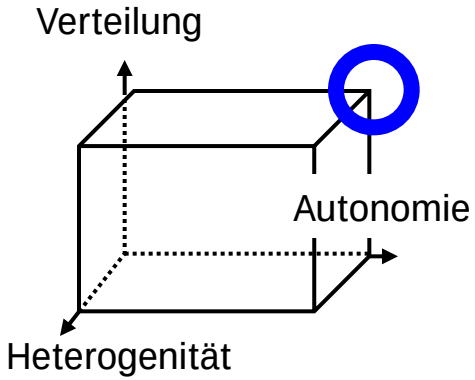
Parallele Datenbank



Verteilte Datenbank

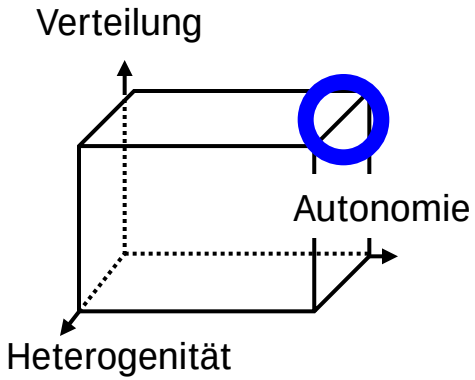


Verteilte & autonome DB



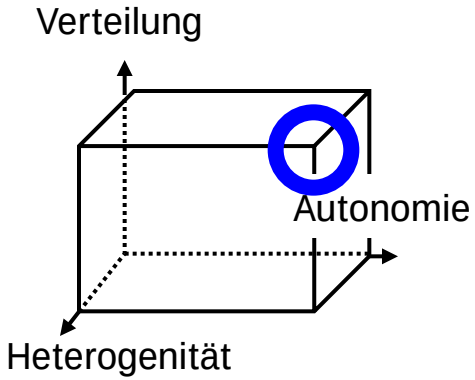
- Verteilte, aber homogene Datenbestände
- Entsteht durch freiwillige Übernahme von Regeln
  - Standards, Verträge, ...
- Autonomie wird teilweise aufgegeben
  - Z.B. Aufgabe von Designautonomie
  - Z.B. nicht Kommunikationsautonomie
  - Z.B. nicht juristische Autonomie

Multidatenbanken



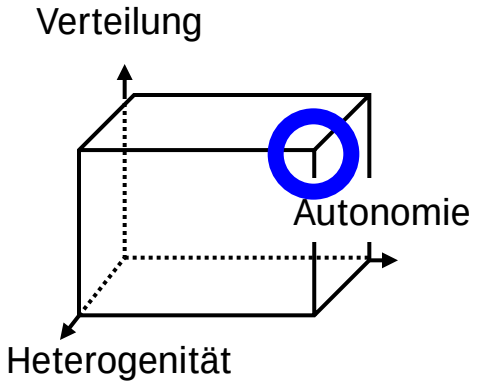
- Verteilt, autonom, „etwas“ heterogen
  - Keine technische Heterogenität
  - Keine Datenmodellheterogenität
  - Zugriff über einheitliche Sprache möglich
- Autonomie bleibt bewahrt
  - Aber Zugriff muss möglich sein (Kommunikationsautonomie)
- Multidatenbanksprachen
  - Qualifizierung von Tabellennamen mit Datenbanknamen
  - Meist spezielle Sprachelemente zur Überbrückung struktureller Heterogenität
  - Überbrückung technischer Heterogenität (siehe Gateways)

Föderierte Datenbanken

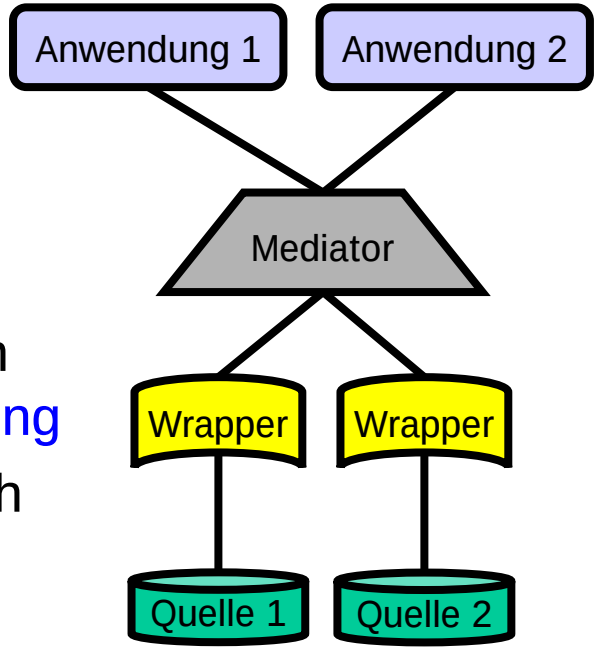


- Häufig verwendeter Begriff ohne klare Definition
- In der Forschung: Enge, virtuelle Koppelung
  - Integrierte Schemata, bereinigte Daten
  - Strukturierte Daten, Anfragen
- In der Industrie: Virtuelle relationale Datenbankintegration
  - Allgemeiner Begriff für integrierten Zugang ohne Datenreplikation
  - Realisierung über Multidatenbanksprachen
  - Definition von Sichten zur Erstellung (teil-)integrierter Schema
  - Datenbankhersteller meiden semantische Integration

Mediator-Wrapper Systeme



- Ebenfalls keine klare Definition
- Trennung der Aufgaben
  - Wrapper verantwortlich für technische / syntaktische Integration
  - Mediatoren verantwortlich für strukturelle/ semantische Integration
- Unterschiede
  - FDBS setzt meist (relationale) Datenbanken voraus und fokussiert auf Anfrageoptimierung
  - Mediator-basierte Systeme adressieren auch semi- / unstrukturierte Daten (Web) und fokussieren auf semantische Probleme



Überblick

- Klassifikation
- Weitere Klassifikationskriterien
- Schichtenaufbau integrierter Systeme

Kriterien föderierter Informationssysteme

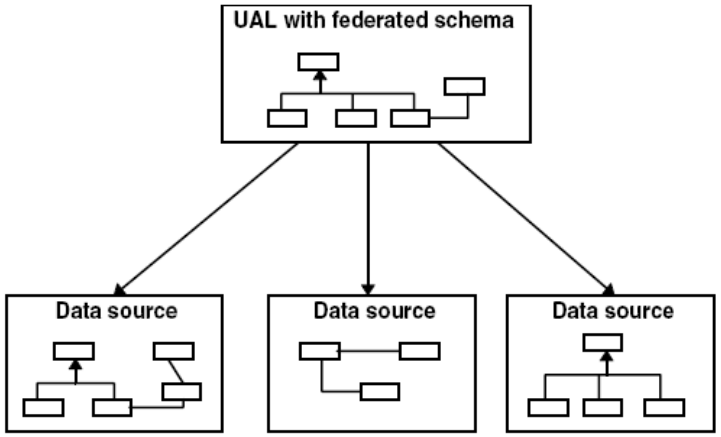
- Föderierte Informationssysteme als Oberbegriff
- Weitere (nicht-orthogonale) Kriterien [BKLW99]
  - Strukturiertheit der Komponenten
  - Enge und lose Kopplung
  - Datenmodell
  - Art der semantischen Integration
  - Transparenz
  - Anfrage-Paradigma
  - Bottom-up oder Top-down Entwurf
  - Virtuell oder materialisiert
  - Read-only oder read-&-write

Strukturiertheit der Komponenten

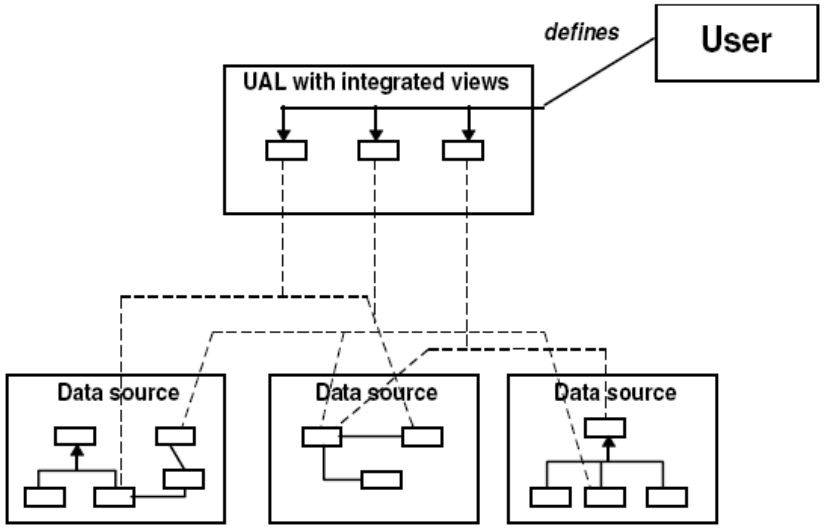
- Strukturiert: Festes Schema, festes Format
- Semi-strukturiert: Feste Elemente, erweiterbar
  - Struktur nur teilweise festgelegt
  - Beispiel: XML/RDF ohne/mit Schemata
- Unstrukturiert
  - Beispiel: Textuelle Daten, Webseiten, Berichte

Enge versus lose Kopplung

- Enge Kopplung
  - Festes und integriertes Schema
  - Für Benutzer einheitlicher Zugang
  - Automatische Anfrageübersetzung
  - System muss Änderungen der Quellen kompensieren



- Lose Kopplung
  - Kein integriertes Schema
  - Struktur / Semantik: Nutzer integrieren selber
    - Nur technische / Datenmodellhetero-genität wird vom System gelöst
  - Änderungen gelangen zum Benutzer



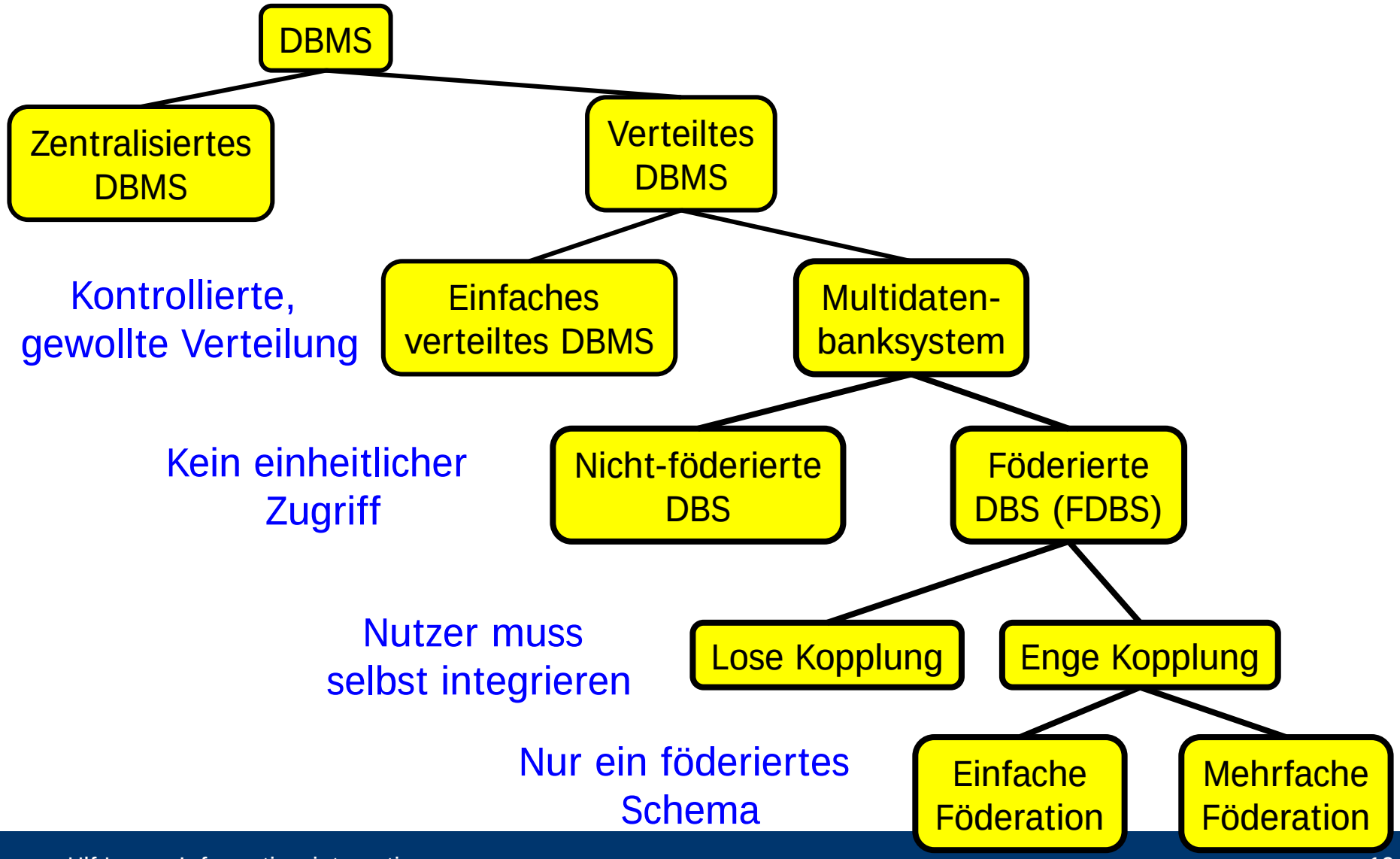
Art der semantischen Integration

- Vereinigung
  - Simple „Konkatenation“ von Objekten
  - Erzeugt redundante Daten
- Anreicherung
  - Mit Metadaten; keine Konfliktauflösung
  - Erzeugt mehr, aber nicht notwendigerweise bessere Daten
- Datenfusion
  - Objektidentifizierung
  - Re-Strukturierung
  - Komplementierung
  - Konfliktlösung

Bottom-up oder Top-down Entwurf

Bottom-up	Top-down
Bedarf nach Integration einer festen Menge von Quellen	Ausgelöst durch „globalen“, quell-unabhängigen Informationsbedarf
Globales Schema durch Schemaintegration	Neuentwurf globales Schema
Änderungen in Quellen i.d.R. nicht vorgesehen (Neuintegration)	Quellen werden nach Bedarf und Eignung hinzugefügt
Meist enge Koppelung, wenige Quellen	Meist lose Koppelung, viele Quellen
Hohe Ansprüchen an Vollständigkeit und Qualität	Geringere Ansprüche an Vollständigkeit und Qualität
Data Warehouse, Merging von Unternehmensdatenbanken	Webquellen, sehr große Integrationssysteme

Taxonomie nach [SL90]

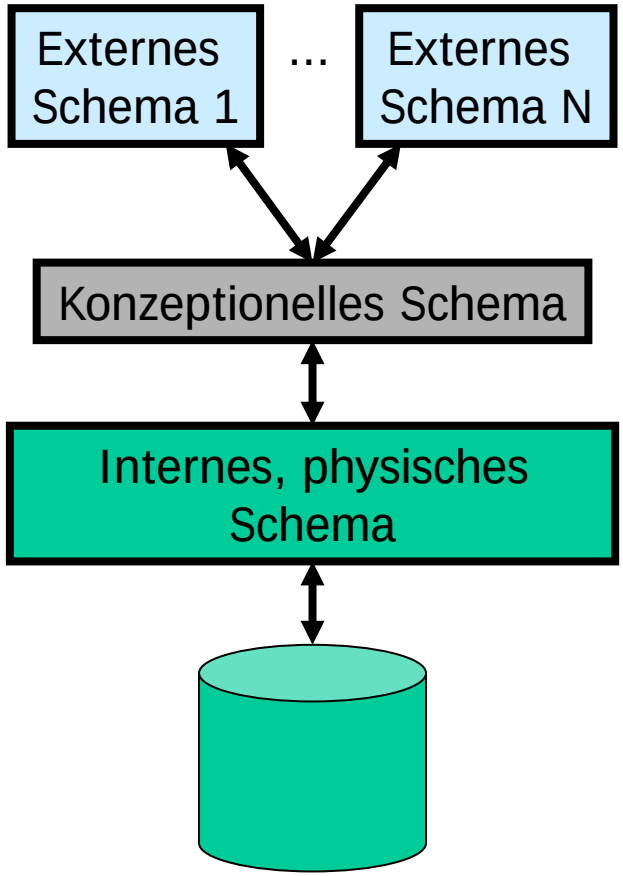


Überblick

- Klassifikation
- Weitere Klassifikationskriterien
- Schichtenaufbau integrierter Systeme

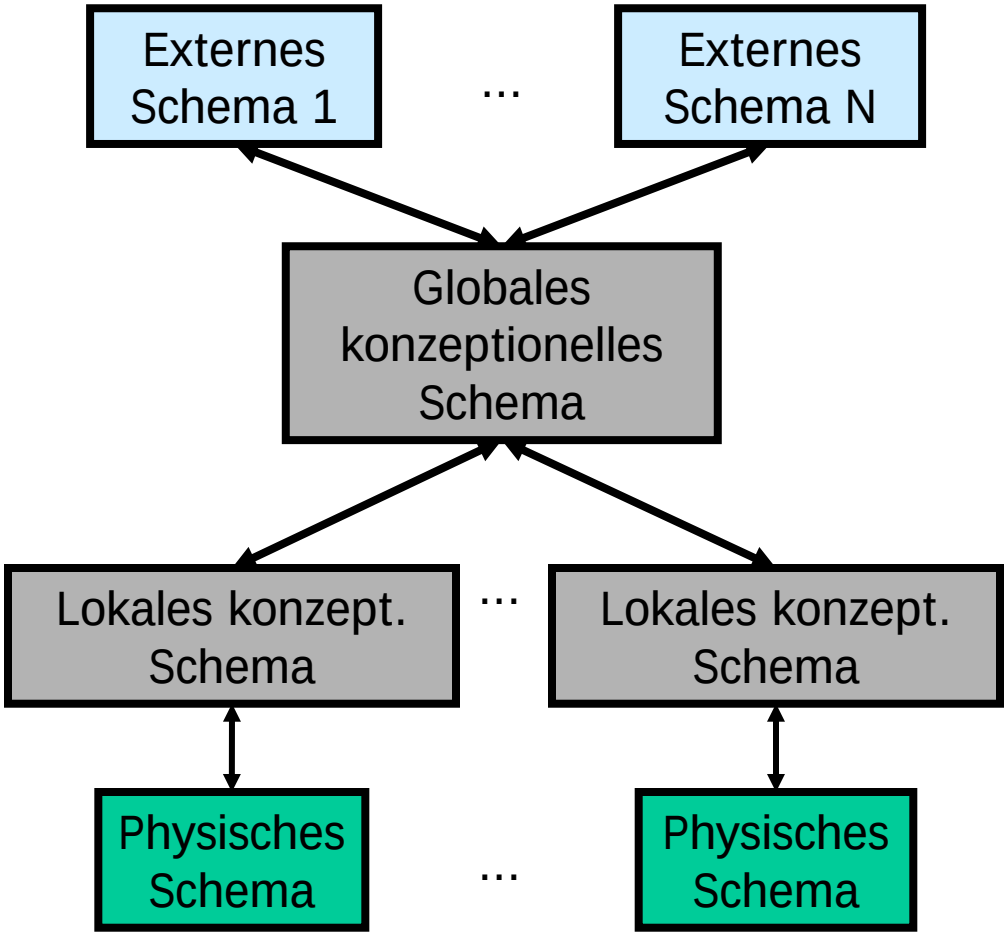
ANSI/SPARC 3-Schichten Architektur

- Externe (logische) Sicht
  - Je nach Anwendung
  - Nur relevante Daten
  - Sichten (Views)
- Konzeptionelle (logische) Sicht
  - Unabhängig von physischer Sicht
  - Definiert durch Datenmodell
  - Stabiler Bezugspunkt
- Interne (physische) Sicht
  - Dateistruktur
  - Speicherort (Zylinder, Block)
  - Indexe, Partitionen, ...



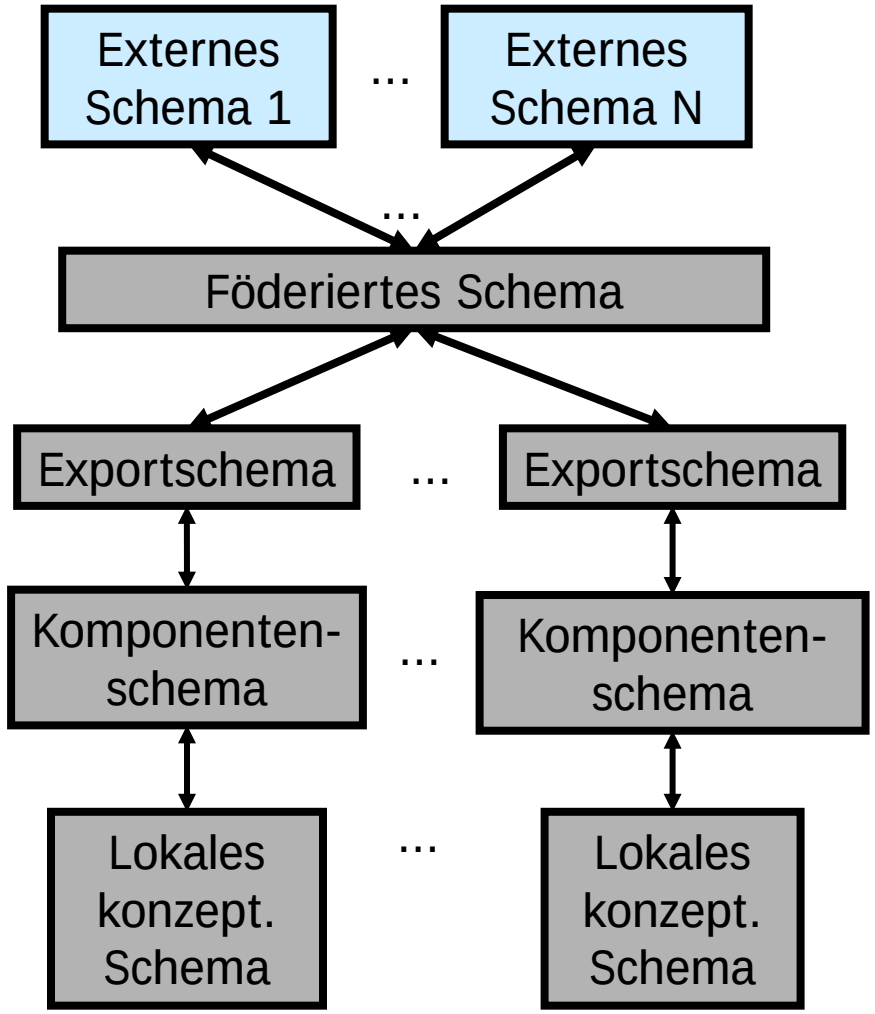
4-Schichten Architektur

- Globales konzeptionelles Schema integriert die lokalen konzeptionellen Schemata
- Lokales und globales konzeptionelles Schema können gleich sein
  - Aber Datenbestände unterschiedlich
- Verteilte Datenbank

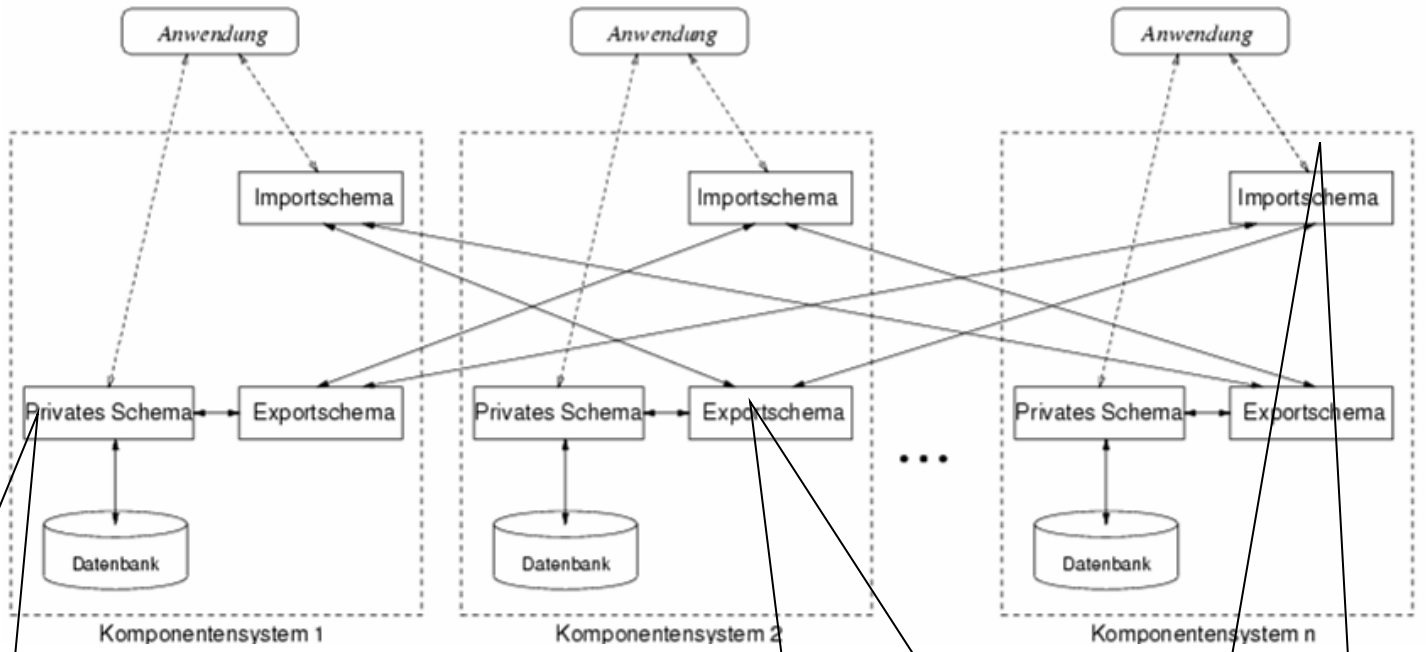


5-Schichten Architektur [SL90]

- Föderiertes Schema = globales konzeptionelles Schema
- Exportschema: Für die Föderation zugänglicher Ausschnitt des Komponentenschemas
- Komponentenschema = lokales konzeptionelles Schema im kanonischen Datenmodell



Weniger Hierarchisch: Import-/Export-Schema-Architektur [HM85]



= lokales konzeptionelles Schema

Teilmenge des lokalen konzeptionellen Schemas wird der Föderation zur Verfügung gestellt

Ausgewählte Teilmengen der Exportschemata werden importiert