

Hochschule Darmstadt  
Fachbereich Informatik

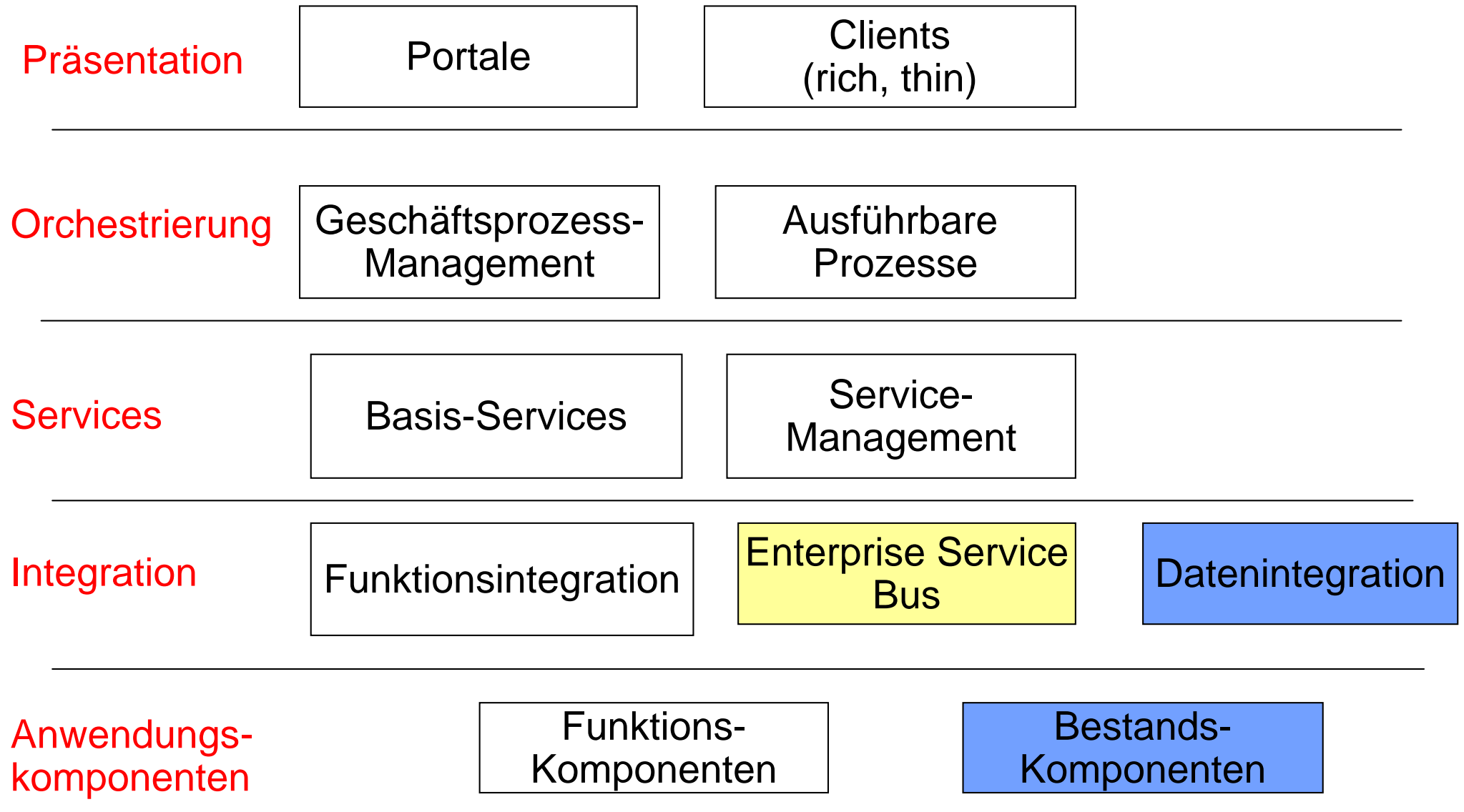
# **Daten- und Systemintegration**

**Kap. 10 Weitere Integrationsarten**

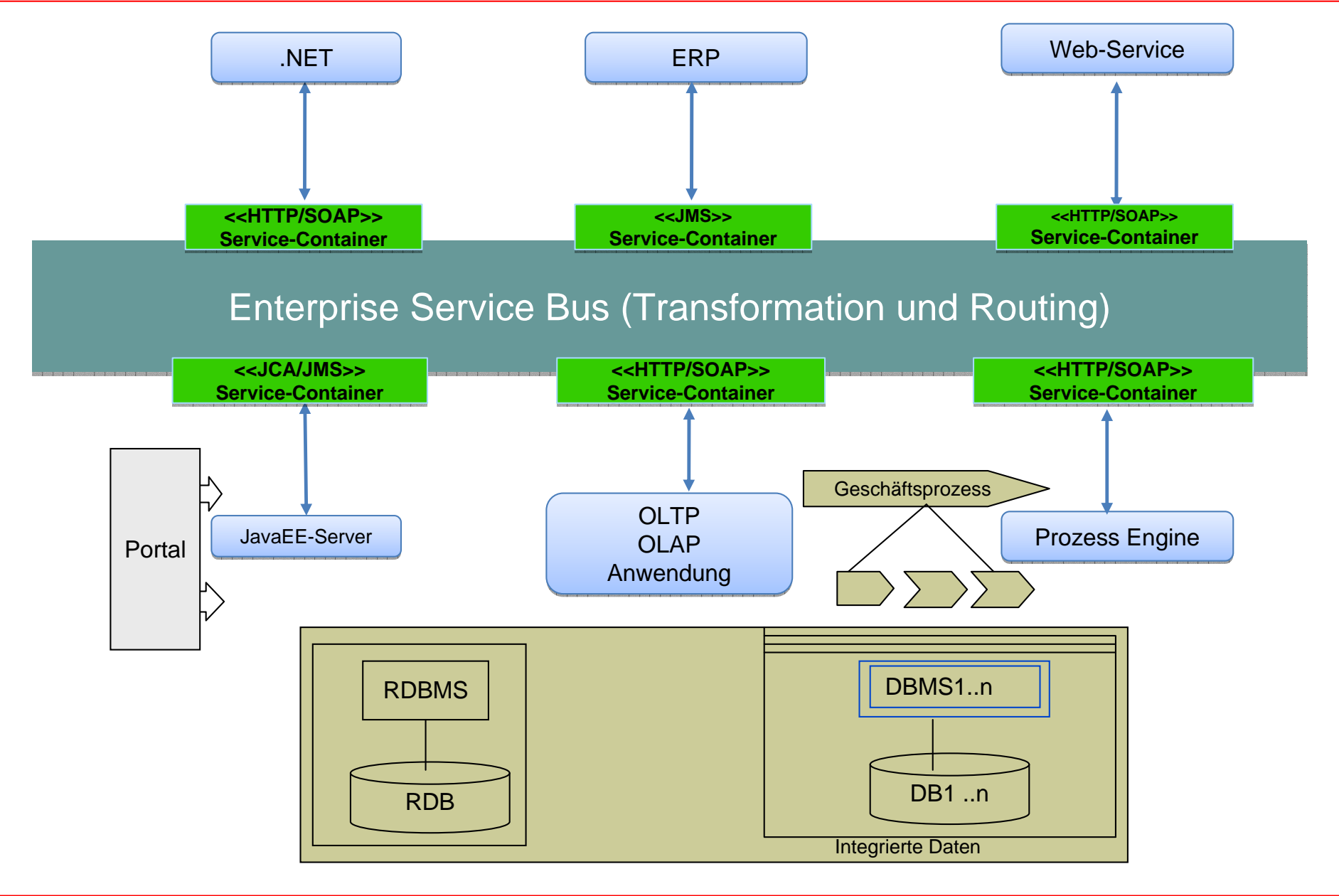
- Datenintegration**
- Portaltechnologien**

Prof. Dr. Frank Bühler

# Standardarchitektur SOA



# Alles auf einem Blick



# Datenintegration - Fälle

## Fall 1: Client-Server Datenbanken (isolierte, nicht-integrierte Datenbestände)

Die Anwendungsprogramme kennen den DB-Server und greifen direkt auf ihn zu (z.B. über JDBC). Dieser Fall ist für unsere Betrachtungen weniger interessant!

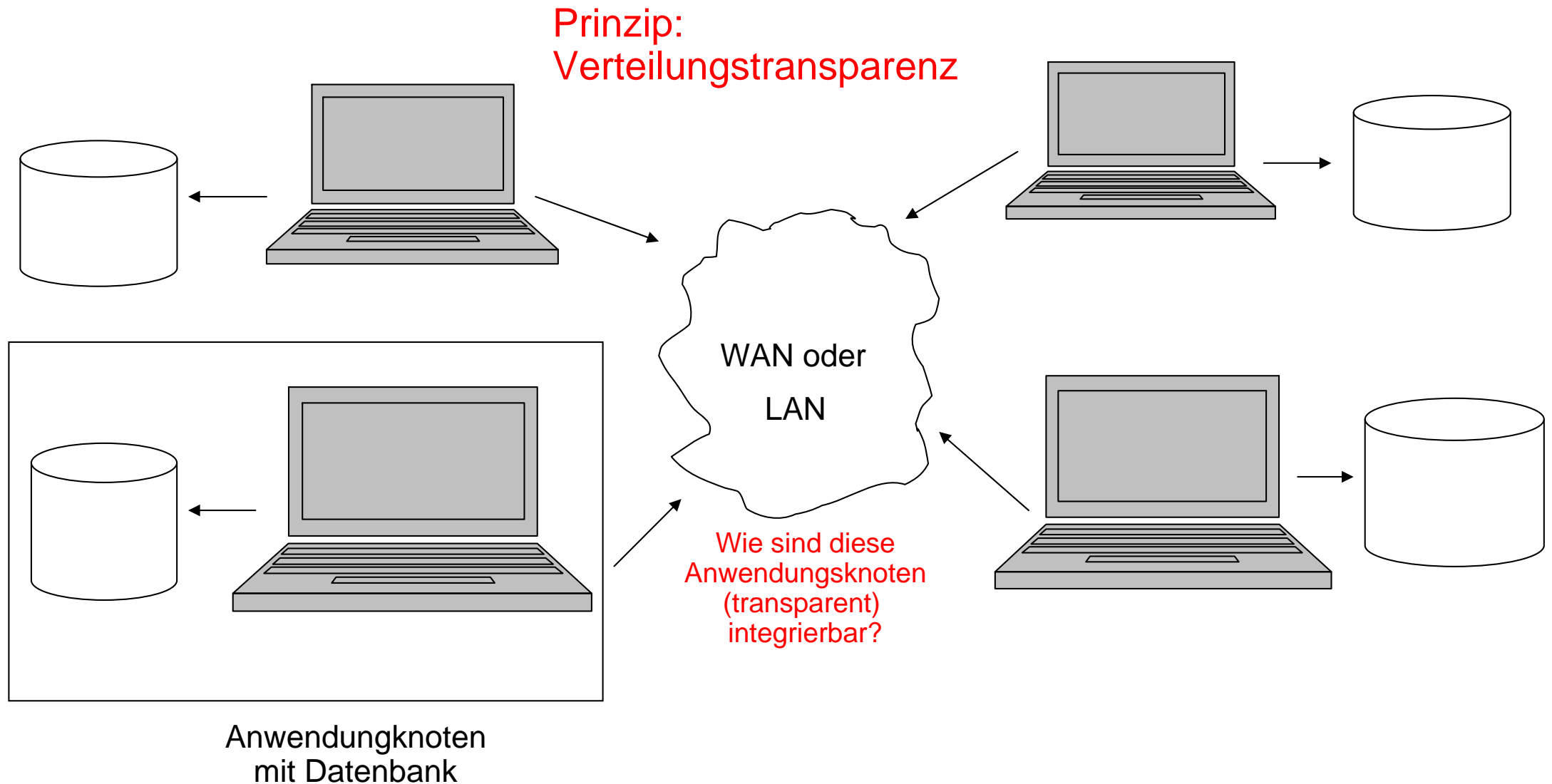
## Fall 2: Verteilte Datenbanken / Föderierte Datenbanken

Die einzelnen DB-Server sind für die Anwendungsprogramme transparent, die Anwendungen führen globale Transaktionen aus, die in lokale Transaktionen der einzelnen DB-Server aufgelöst werden

- a) prä-integrativ - **Verteilte Datenbanken**: Die Verteilung der Daten auf die Knoten wird vorab entworfen und festgelegt. In den Knoten laufen i.d.R. **homogene Datenbanksysteme (z. B. DB2 oder Oracle)**.
- b) post-integrativ - **Föderierte Datenbanken**: Die Verteilung der Daten auf die Knoten ist zufällig und historisch gewachsen. In den Knoten laufen i.d.R. **heterogene Datenbanksysteme (z. B. DB2, Oracle und Sybase bzw. unterschiedliche Datenbanktechnologien)**.

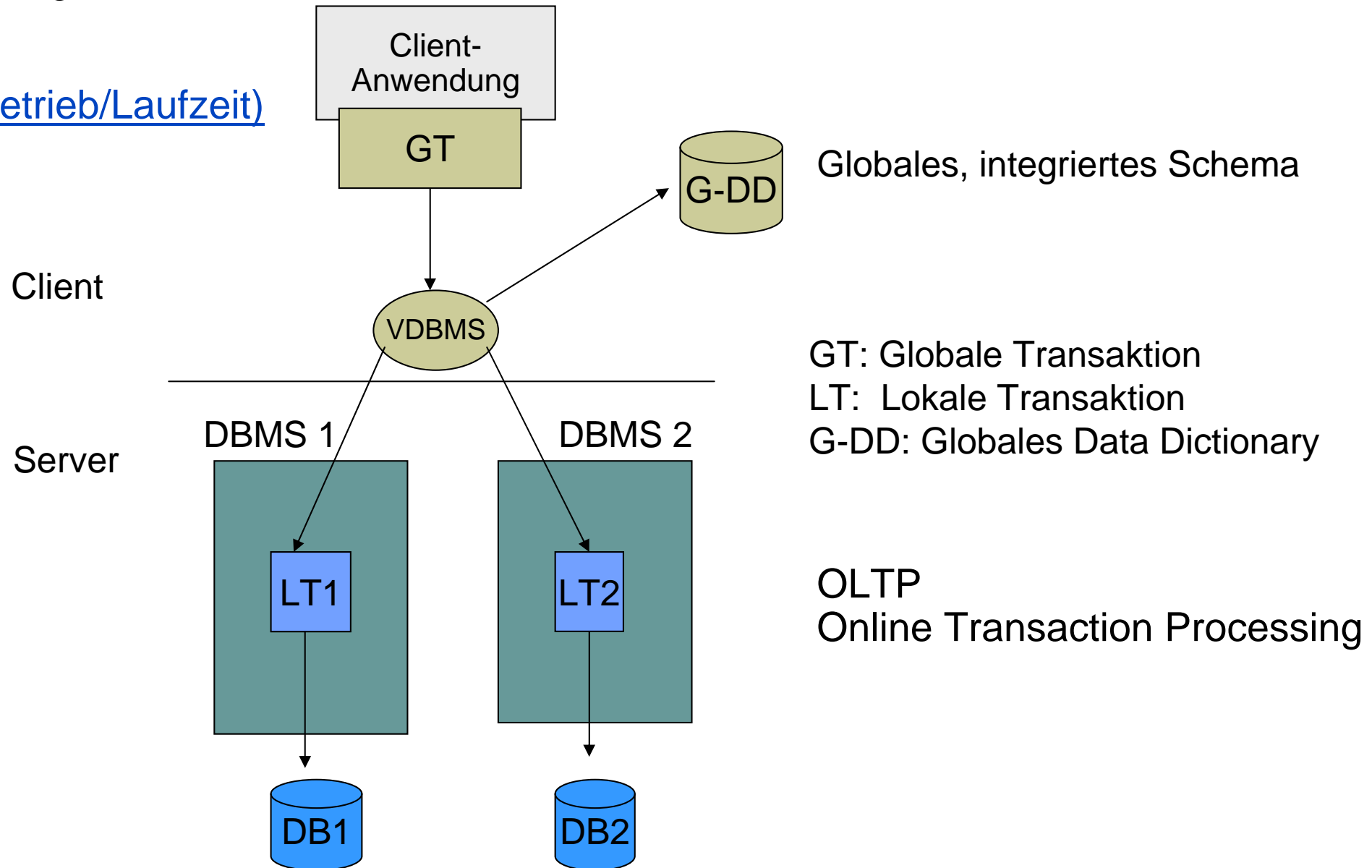
## Fall 3: OLAP / Business Intelligence

## Fall 2: Grundmodell verteilte DB (mit n Knoten)



# Datenintegration

## Fall 2 (Betrieb/Laufzeit)



# Datenintegration

## Beispiel zu Fall 2:

### **Funktion Verfügbarkeitsprüfung mit Nachbestellung für Artikel.**

Die Prüfung erfolgt als globale Transaktion transparent in  $n$  **verteilten Lager-Datenbanken**, kontrolliert durch  $n$  **homogene oder heterogene Datenbanksysteme**.

Die Transaktionsklammer umfasst die Reservierung für die Artikel und die rechtsverbindliche Bestellung der Artikel.

### Fall 2a):

Verteilungsentwurf (vor Implementierung der Datenbanken) für  $n$  auf verschiedene Standorte verteilte, relationale Lager-Datenbanken, um folgende **Ziele** zu realisieren:

- Bessere Abbildung der dezentralen Organisation in Hauptwerk/Niederlassungen (Horizontale/vertikale FrAGMENTIERUNG)
- Erhöhung der Systemverfügbarkeit durch geplante Redundanz (Replikationen)
- Senkung der DÜ-Kosten
- Verbesserung der Performanz

# Datenintegration

## Fall 2b)

Nachträgliche Integration von bestehenden, zumeist heterogenen Lager-Datenbanken (**relationale DB, Netzwerk-DB, hierarchische DB**), um folgende **Ziele** zu erreichen:

- Globale Verfügbarkeitsprüfung bei weltweit verteilten Zulieferern mit heterogenen Datenhaltungssystemen
- Investitionssicherung von bestehenden Lagerdatenhaltungssystemen im eigenen Unternehmen

Datenbankfunktionalitäten wie verteilte Anfragebearbeitung und globale Transaktionssteuerung sind wesentlich komplexer zu implementieren als im Fall 2a).

# Datenintegration

## Ein Integrationsszenario zur Datenintegration mit OLTP

Ein großer **Finanzdienstleister** setzt zur Abwicklung des Handels von Wertpapieren und die Verwaltung der entsprechenden Konten konzernweit eine Lösung auf Basis von JavaEE ein.

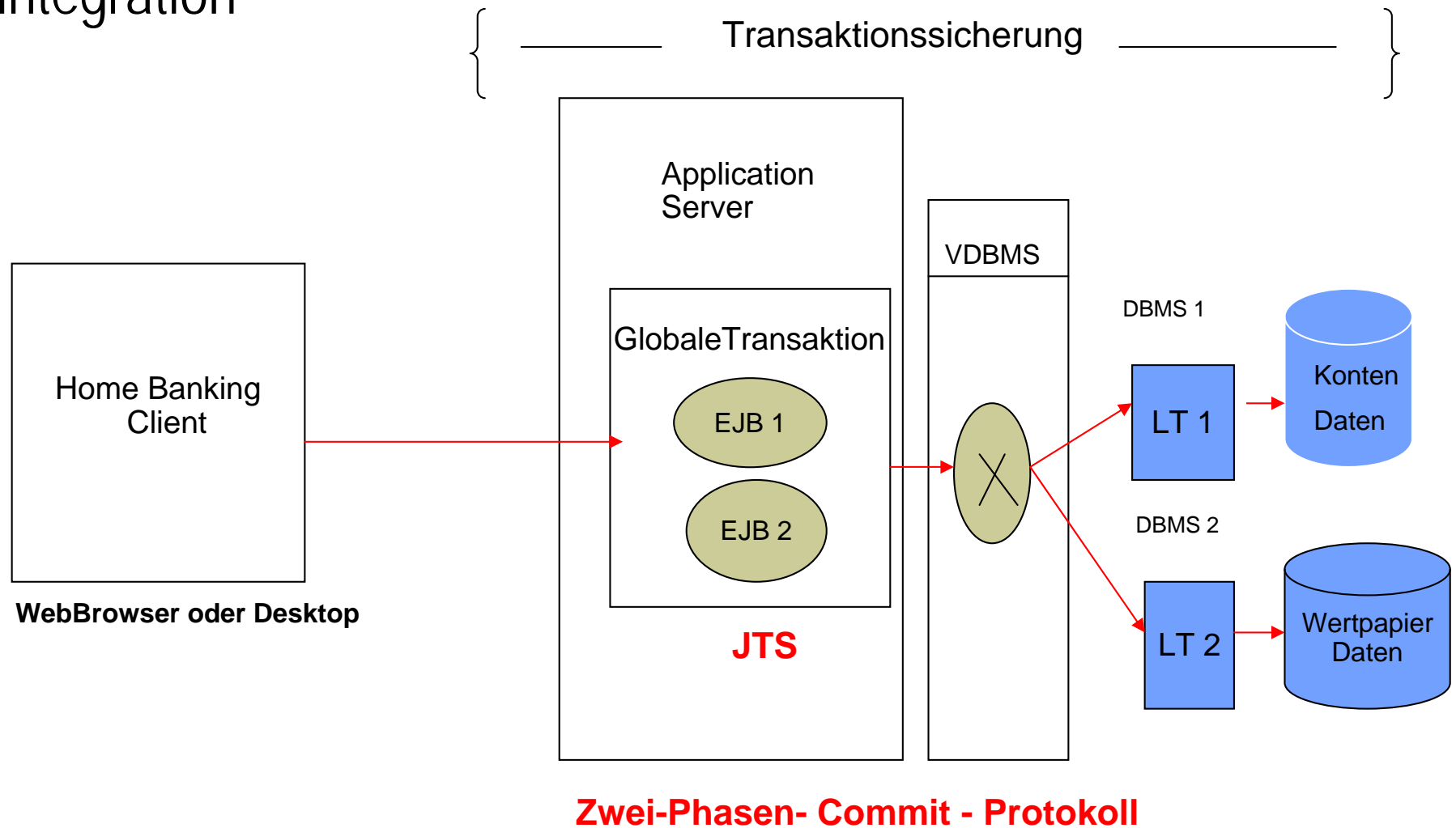
Die Kontenbuchungen für die Kunden und der An- und Verkauf von Wertpapieren werden durch EJB-Business-Komponenten (Session Beans) abgebildet. Diese greifen direkt auf ein **verteiltes Datenbankmanagementsystem** zu, das die verteilten Datenbestände zu den Girokonten und zu den Investmentfonds verwaltet.

Ein Kunde will per **Home Banking** eine globale Transaktion durchführen, die die Verarbeitungsschritte

- Kauf eines Fonds
- Einzug des Kaufbetrags vom Konto umfasst.

Die Transaktion wird am Home Banking – Client gestartet und beendet. Dazwischen durchläuft sie die Application Server- und DBMS – Schicht und muss gesichert werden.

# Datenintegration



# Datenintegration

## Zwei-Phasen-Commit – Protokoll für verteilte Transaktionen

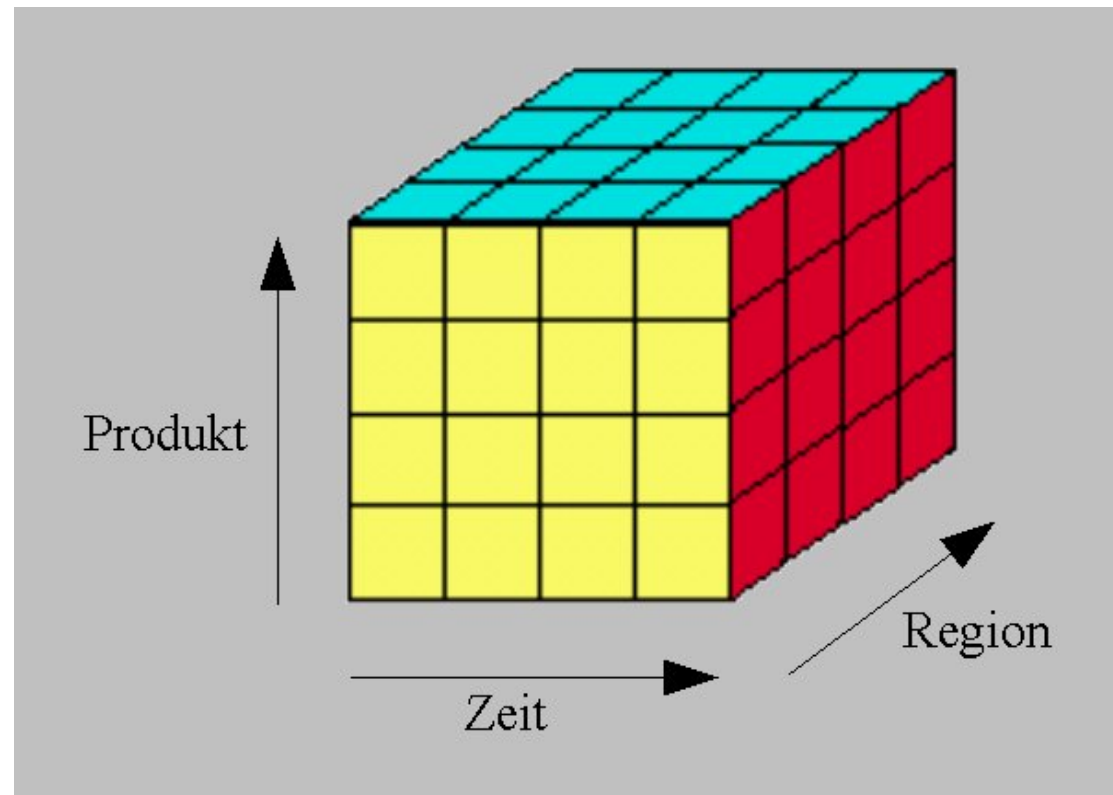
**Erste Phase:** Jeder entfernte Knoten meldet dem Supervisor die Bestätigung seiner Änderungen, hält sie aber noch in der Schwebe.

**Zweite Phase:** Wenn der Supervisor die (positiven) Bestätigungen aller Knoten erhalten hat, meldet er dies allen Knoten. Diese können dann ihre lokalen Änderungen abschließen. Wenn nicht alle Knoten positiv bestätigen, setzen alle Knoten ihre Änderungen zurück (lokales Rollback).



# Datenintegration

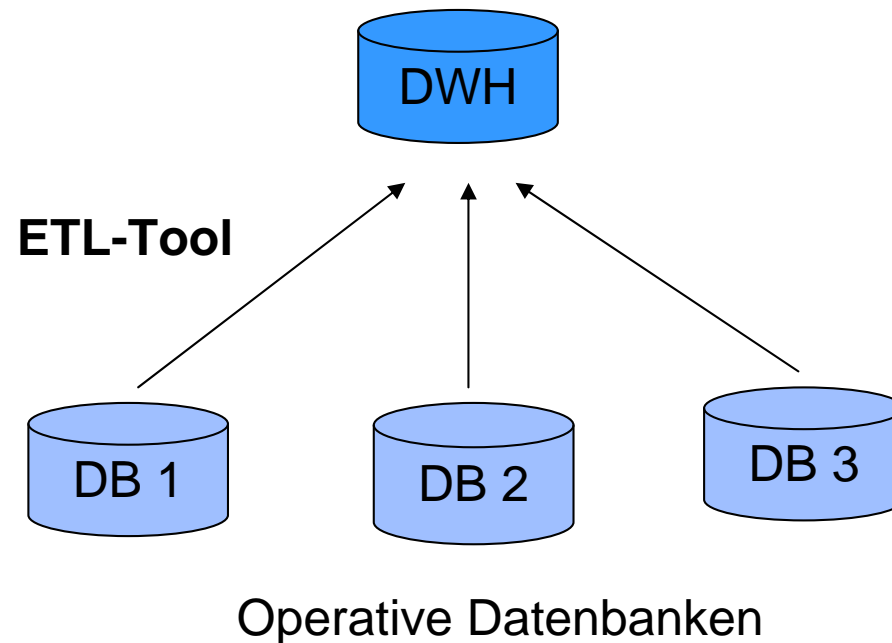
## Fall 3: On-Line Analytical Processing (OLAP) Datenintegration für Auswertungen



Dimension	Menge von Attributen
Produkt	Warengruppen: Lebensmittel, Kleidung, Elektronik,
Zeit	Jahr: 2005, 2006, 2007, 2008, 2009, 2010
Region	Regionen: Nord, Süd, Ost, West

# Datenintegration

Basis dieser Auswertungen ist die Datenintegration in einem Data Warehouse (DWH). Dies ist post-integrativ – mit allen Folgen (Data Cleansing, Datentransformation etc.)!



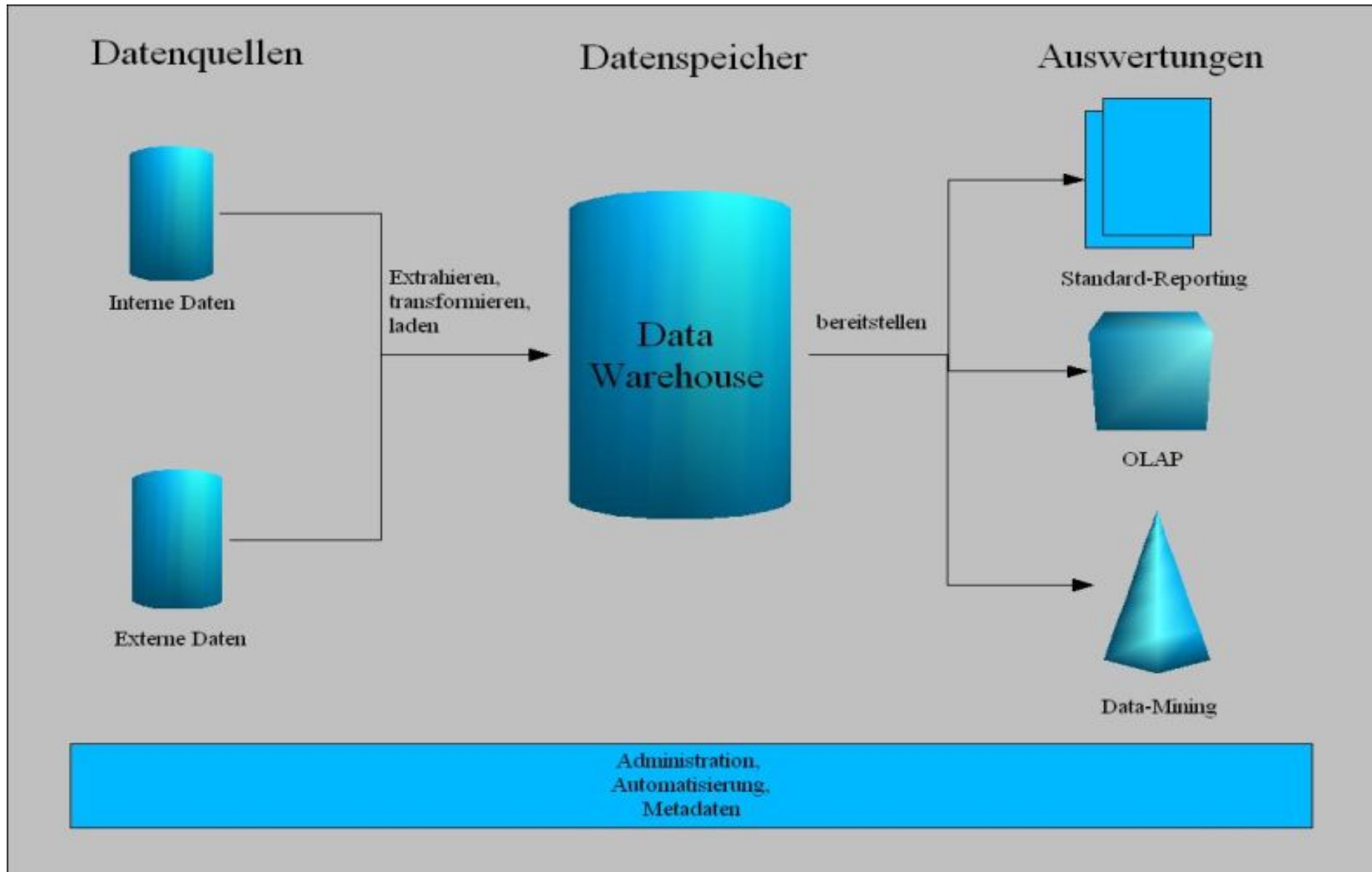
**Extraktion** der relevanten Daten aus verschiedenen Quellen

**Transformation** der Daten mittels Strukturanpassungen und Bereinigung

**Laden** der Daten in das Data Warehouse

- Ziele:**
- Verdichtung von operativen Daten in entscheidungsrelevante Daten
  - Bereitstellung von Sichten (z.B. slice and dice)
  - Basis für OLAP - Auswertungen

# Data Warehouse



# Datenintegration

Ein **Data-Warehouse-System** besteht aus ...

- .. dem Datenbeschaffungsbereich (engl. **staging area**), in dem die aus den Datenquellen extrahierten Daten zusammengeführt und gegebenenfalls bereinigt und transformiert werden
- .. der zentralen **Data-Warehouse-Datenbank**, dem eigentlichen Data Warehouse
- .. weiteren Komponenten für die Bereitstellung von bereichs- oder auswertungsspezifischen Sichten
- .. dem Metadaten-Repository
- .. dem DWH-Manager für die Ablaufsteuerung (inkl. z. B. Timer-Steuerungsmodulen)

# Datenintegration

## Data Warehouse

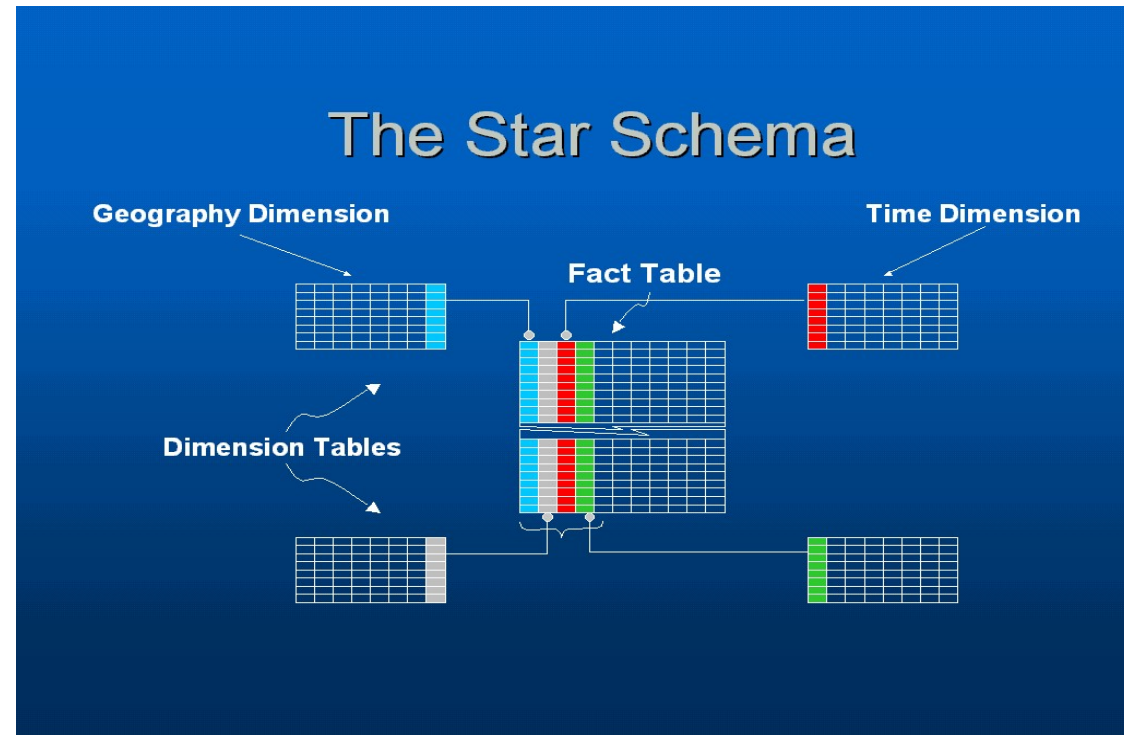
- Logisch zentrale Datenbank
- Enthält potentiell entscheidungsunterstützende Daten
- Daten stammen aus verschiedenen Quellsystemen
- Daten sind historisiert
- Daten sind bereinigt und für Analysen vorverarbeitet
- Daten sind strukturiert

## Hauptmerkmale

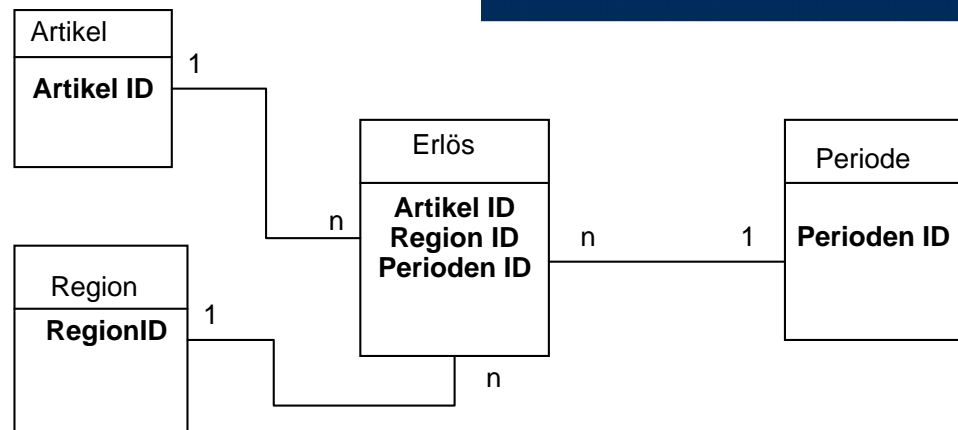
- Integration
- Zeitraumbezug
- Zwischen zwei Ladevorgängen konstant

# Datenintegration

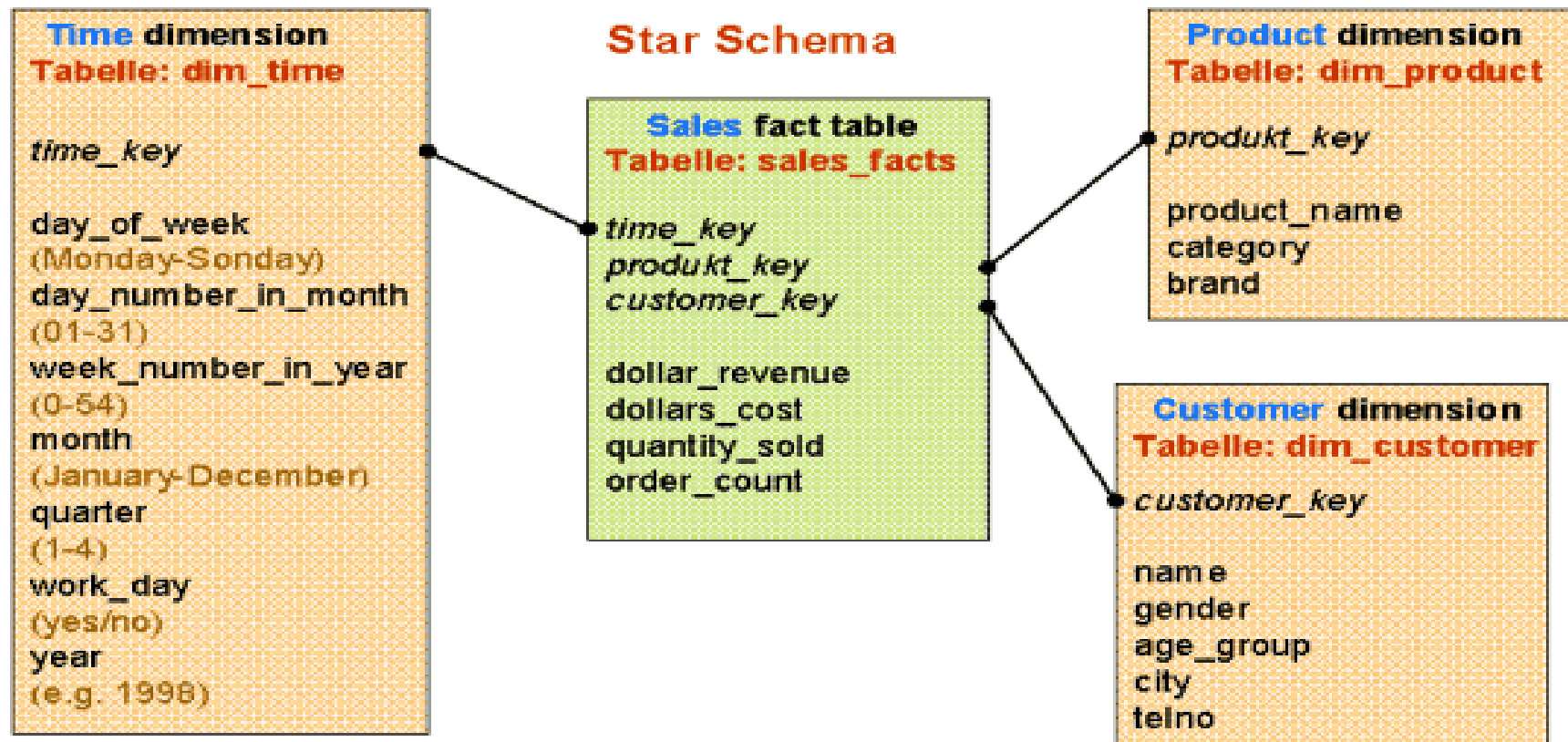
## Das Star-Schema für ein Data Warehouse



Beispiel:



# Datenintegration



# Datenintegration

## ETL Extraction Transformation Loading

### Extraktion:

Aufgabe der Extraktionskomponente ist es, die Daten aus den Quellen in den Arbeitsbereich (staging area) des DWH zu übertragen (periodisch, zumeist nachts, auf Anfrage oder ereignisgesteuert).

Extraktionskomponenten werden unter Nutzung von Standardschnittstellen (z. B. ODBC/JDBC) realisiert.

# Datenintegration

## ETL Extraction Transformation Loading

### Transformation

Die Transformationskomponente passt die Daten für die DWH-Datenbank an und bereitet das Laden vor.

Strukturelle Transformation: Schemaintegration

Inhaltliche Transformation: Daten- bzw. Objektinstanzintegration

Außerdem führt diese Komponente eine Bereinigung der Daten durch ( **Data Cleaning**) – Bereinigung von fehlerhaften Werten, Redundanzen, veralteten Werten usw.

# Datenintegration

## ETL Extraction Transformation Loading

### Laden

Aufgabe der Ladekomponente ist es, die bereinigten und aufbereiteten (z.B. aggregierten) Daten in die DWH-Datenbank zu laden. Dazu werden spezielle Ladewerkzeuge, z.B. SQL Loader von Oracle, eingesetzt.

Der Ladevorgang kann on- oder offline erfolgen. Beim offline-Ladevorgang geschieht das in vereinbarten Zeitfenstern, z.B. nachts oder am Wochenende.

# Datenintegration

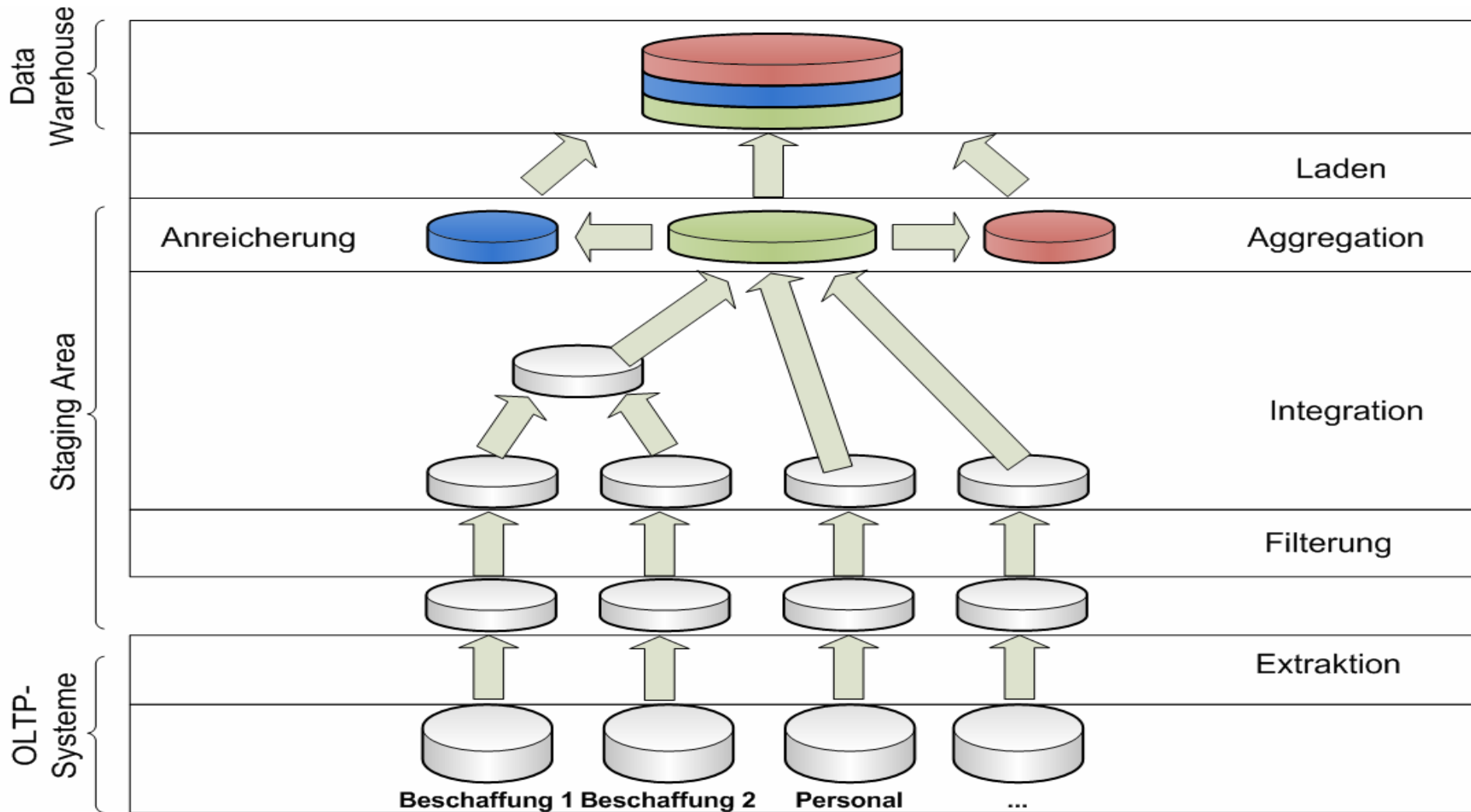
## ETL



Extraktion	<b>Erfassung der Daten aus den Quellsystemen</b>
Filterung	<b>Bereinigung syntaktischer Fehler</b>
Integration	<b>Vereinheitlichung und Zusammenführung</b>
Aggregation	<b>Verdichtung der Daten</b>
Anreicherung	<b>Bildung von Kennzahlen</b>
Laden	<b>Schreiben der Daten in das Zielsystem (DWH)</b>

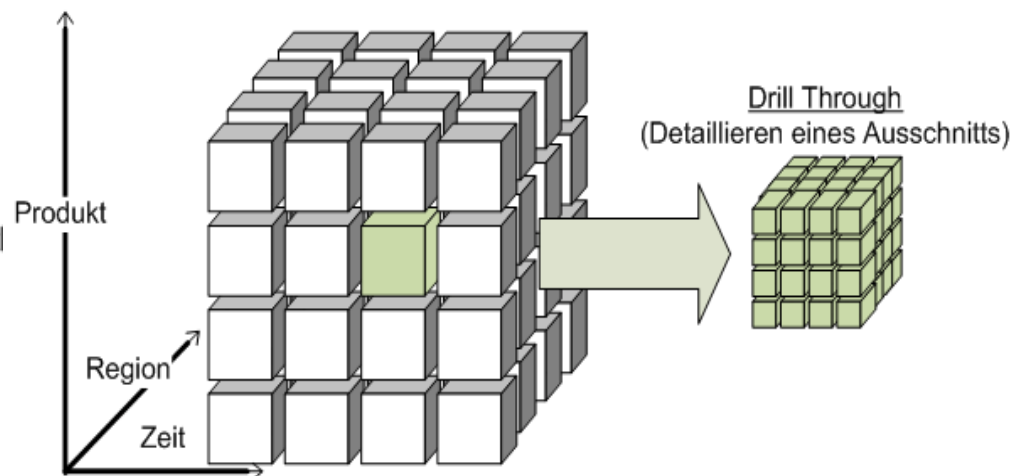
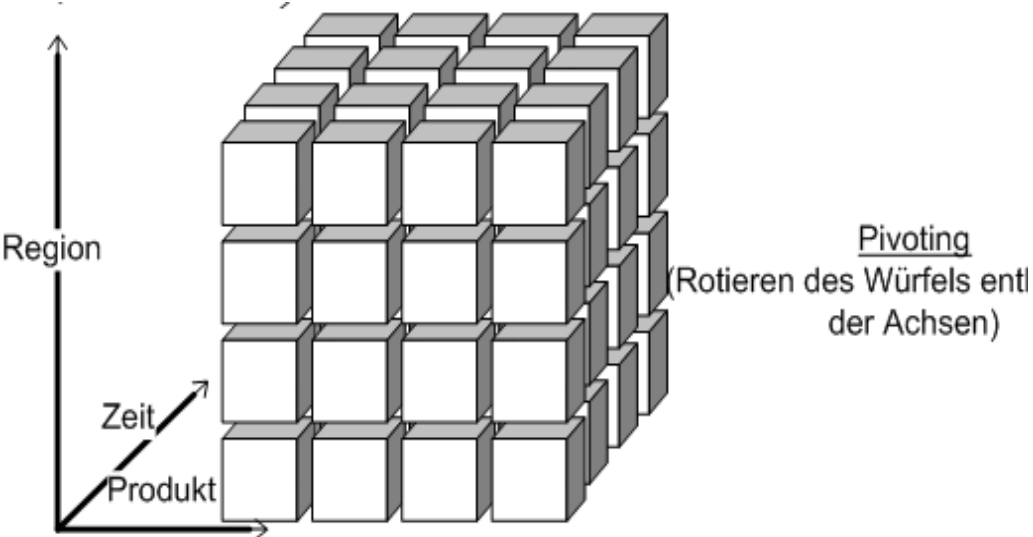
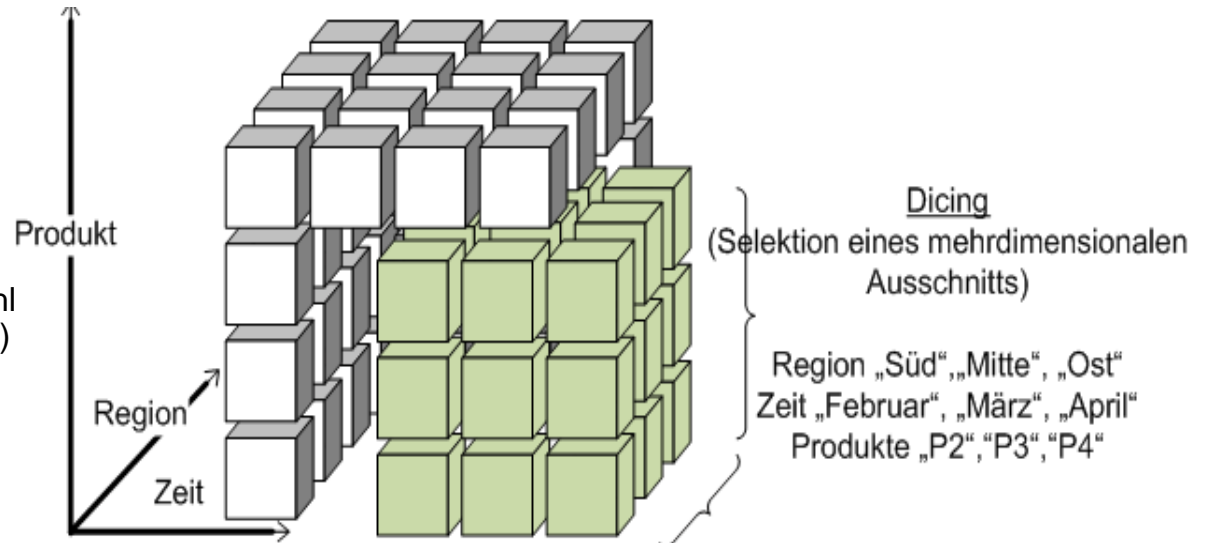
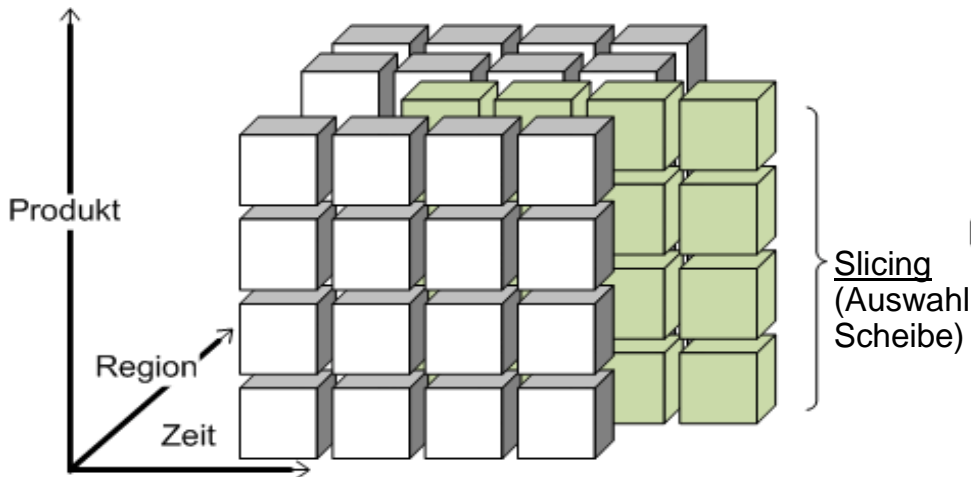
# Datenintegration

## ETL



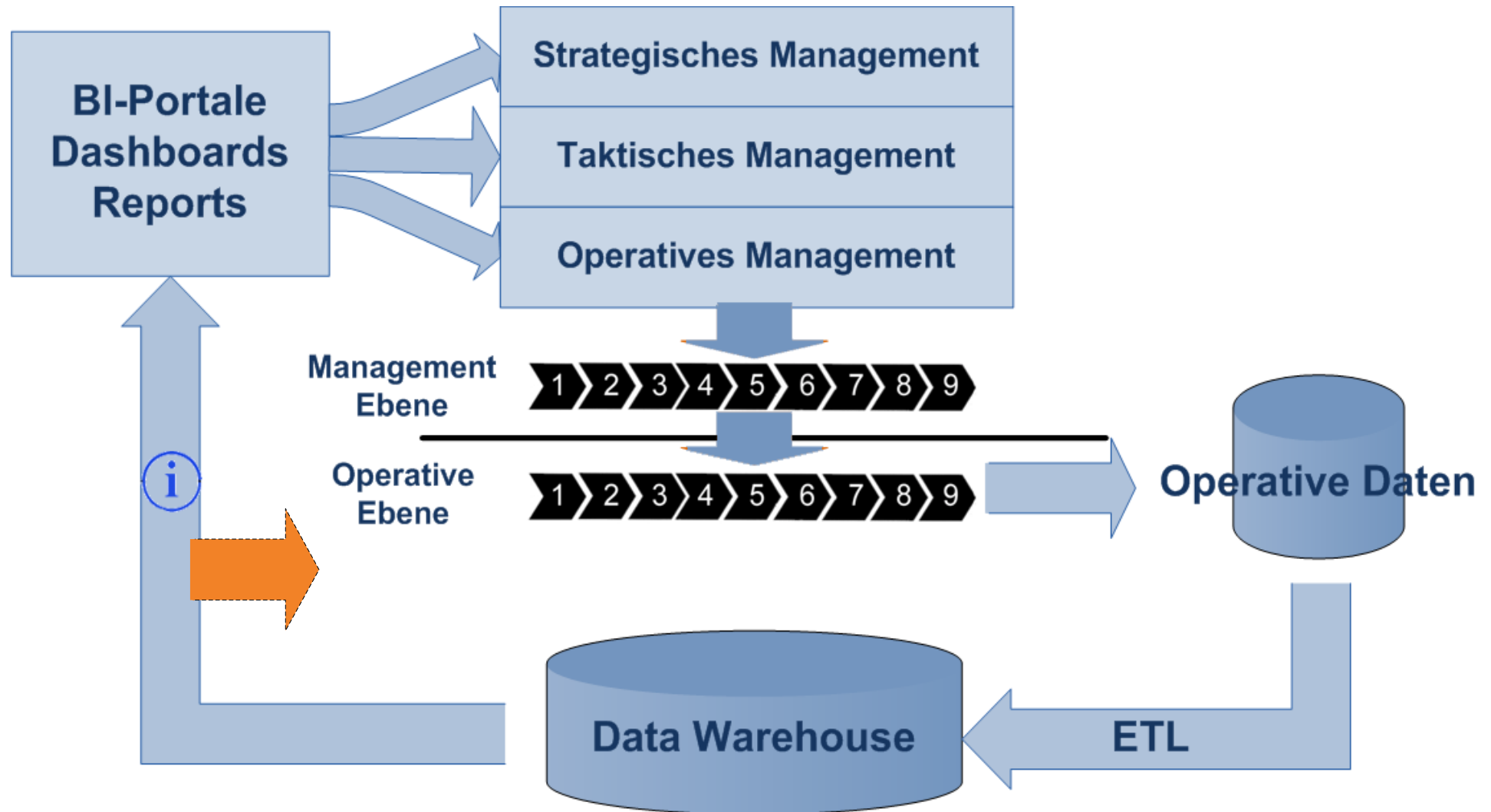
# Datenintegration

## Auswertungsspezifische Sichten



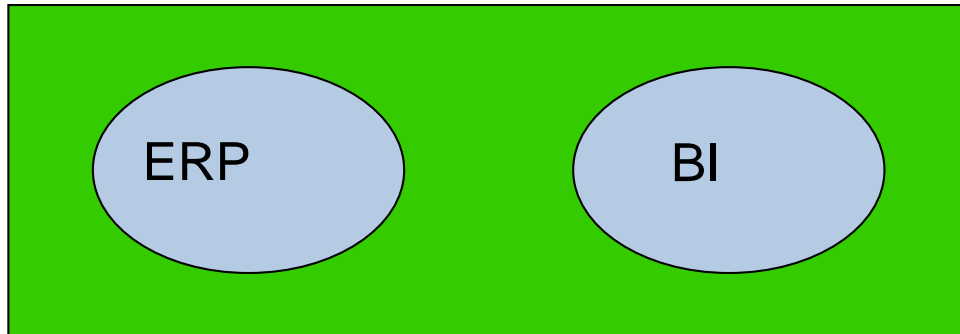
# Datenintegration

## Fall 4: Closed Loop of Information Supply Chain



# Datenintegration

## Beispielszenario: Zuverlässigkeitsbewertung von Lieferanten



*Szenario*

- Einige Materialien werden von *mehreren Lieferanten* angeboten
- Preisunterschiede zwischen den Lieferanten gering
- Nur zuverlässige Lieferanten sollen beauftragt werden

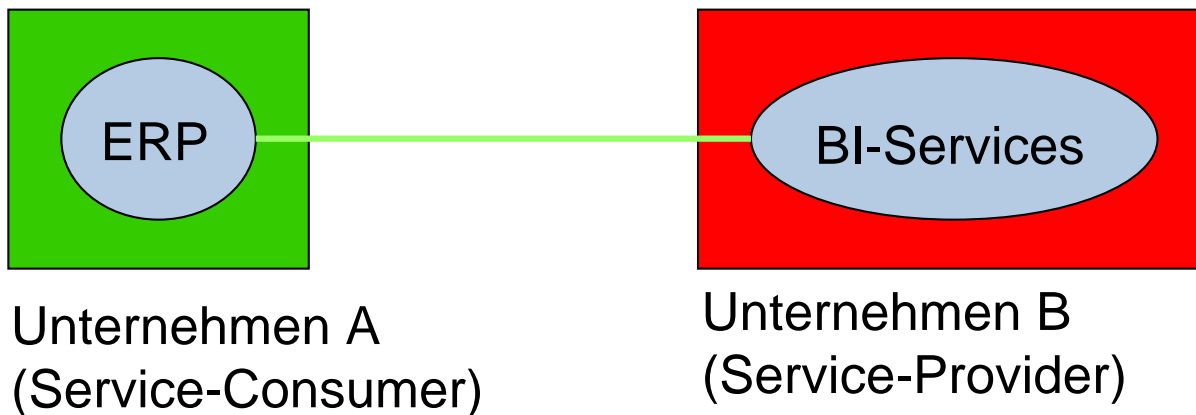
**Lieferanten sollen bewertet werden**

**Bewertungsinformationen sollen in das ERP-System integriert und so den Mitarbeitern im Einkauf zugänglich gemacht werden**

# Datenintegration

## Beispielszenario: Zuverlässigkeitsbewertung von Lieferanten

Szenarioerweiterung: BI as a Service



Externer BI-as-a-Service-Anbieter übernimmt Lieferantenbewertung  
Benötigte Daten werden regelmäßig extrahiert und als Parameter an Web Service des Service-Providers übergeben  
Lieferantenbewertung wird durchgeführt und ist über Webservice abrufbar  
ERP-System u. -DB von Unternehmen A werden angepasst, um die Bewertungen anzeigen zu können

# Datenintegration

## Beispielszenario: Zuverlässigkeitsbewertung von Lieferanten

Mögliche Kriterien zur Bewertung der Lieferantenzuverlässigkeit

Primäre Bewertungskriterien	Bewertungsgrundlage
Reaktionszeit	Zeitdifferenz zwischen Angebotsanfrage und Angebot; Zeitdifferenz zwischen Auftrag und Auftragsbestätigung.
Termintreue	Verhältnis von verspäteten zu rechtzeitigen Lieferungen.
Warenqualität	Anzahl der Reklamationen.



# Datenintegration - Integrationskonflikte

Zusammenstellung der Konflikte, die durch Heterogenität entstehen können

1 Unterschiedliche Rechnerhardware und Betriebssysteme

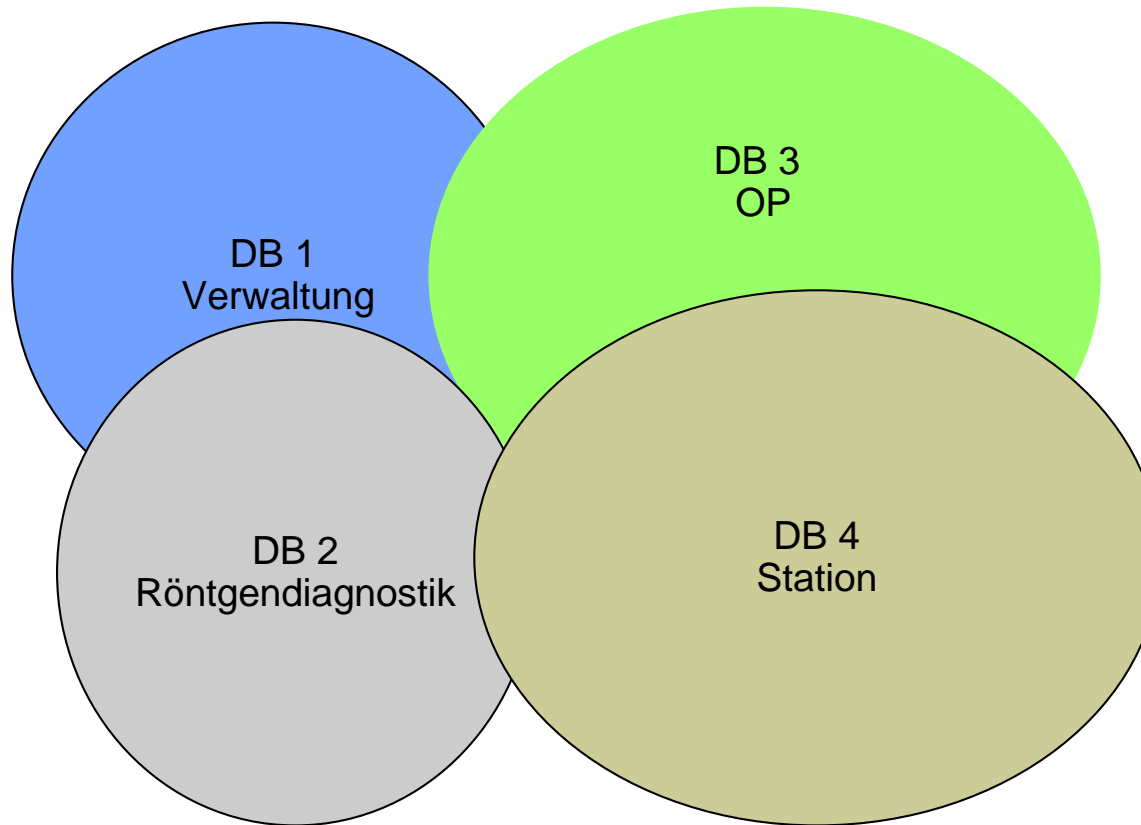
**2 Nicht vollständig übereinstimmende Modellierungsbereiche**

**3 Unterschiedliche Datenmodelle**

**4 Unterschiedliche Modellierungen bzw. Schemata**

**5 Heterogenität der eigentlichen Daten**

# Datenintegration - Integrationskonflikte



2 Nicht vollständig  
übereinstimmende  
Modellierungsbereiche

3 Unterschiedliche Datenmodelle

4 Unterschiedliche  
Modellierungen bzw.  
Schemata

5 Heterogenität der eigentlichen  
Daten

Beispiel für eine Ausgangssituation:  
Datenbestände in einem Krankenhaus-Informationssystem

# Datenintegration - Integrationskonflikte

## Zu 2) Konflikte durch unterschiedliche Modellierungsbereiche

Jede der zu integrierenden Datenbanken bildet z.B. eine bestimmte Sicht auf den Patienten ab, diese Sichten stimmen aber nicht vollständig überein.

## Zu 3) Datenmodellkonflikte

Den zu integrierenden Datenbanken liegen verschiedene Datenmodelle (Technologien) zu Grunde.

Beispiel: DB 1 verwendet ein objektorientiertes Modell,  
(OO-Modellierungskonstrukte: Vererbung, Aggregation, Assoziation, ...)  
DB 2 ein relationales Modell  
(Relationales Modellierungskonstrukt: Beziehungen zwischen Relationen  
mittels Primär-/Fremdschlüsseln)

# Datenintegration - Integrationskonflikte

## Zu 4) Schemakonflikte

Die zu integrierenden Datenbanken verwenden das gleiche Datenmodell

Die Konfliktfälle betreffen:

### 4.1. Objektmengen:

Zwei zu integrierende Entitätstypen verwenden unterschiedliche Objektmengen:

- Teilmengenbeziehung
- Überlappung
- Disjunktheit

### 4.2 Struktur:

Innerhalb eines Datenmodells gibt es verschiedene Möglichkeiten einen Sachverhalt strukturell abzubilden, z.B. mit einfachen/mehrwertigen Attributen oder eigenständigen Entitätstypen

# Datenintegration - Integrationskonflikte

## 4.3. Beschreibungen

z. B.

- Beschreibung eines Entitätstyps mit unterschiedlichen Attributen
- Homonyme und synonyme Bezeichnungen von Entitätstypen, Attributen und Beziehungen:

Synonym – der gleiche Entitätstyp hat in zwei lokalen Schemata unterschiedliche Bezeichnungen (Personal/Mitarbeiter)

Homonym – verschiedene Entitätstypen haben in zwei lokalen Schemata die gleiche Bezeichnung (Schloss=Türschloss, Schloss=Gebäude)

- Datentypkonflikte

USW.

# Datenintegration - Integrationskonflikte

## Zu 5) Datenkonflikte

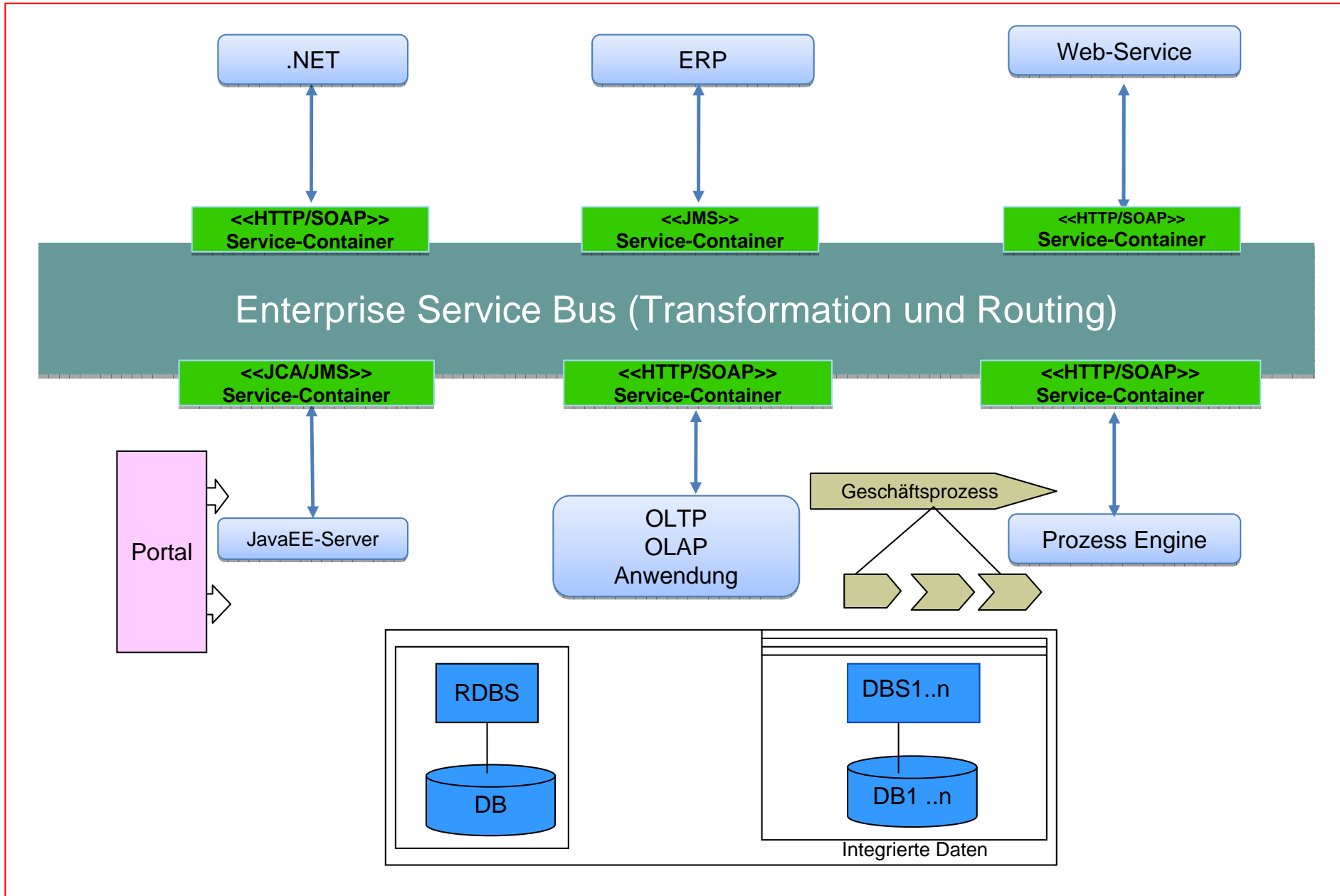
- Inkorrekte Einträge
- Unterschiedlich aktuelle Einträge
- Unterschiedliche Repräsentation von Werten  
(z.B. numerisch/nicht-numerisch, Genauigkeiten/Stellen)

## Fazit:

Der Weg zu einem integrierten Datenbankschema ist lang und steinig.



# Zur Einordnung



# Agenda

## Portale im Überblick

- Aufbau und Referenzarchitektur
- Portalstandards
- Nutzen und Einordnung in die Anwendungsintegration

# Portale im Überblick

## Definition

„Ein Portal stellt basierend auf einer oder mehreren **Zugangstechnologien** (z.B. Web-Technologien, Sprache) nach einer **einmaligen Authentifizierung** (Single Sign On) einen zentralen Zugriff auf **personalisierte Inhalte** (Content und ggf. Dokumente), **Prozesse und Anwendungen** bereit.“



Port|al, das; -s, -e ([Haupt]eingang), [prunkvolles] Tor; ...  
(aus „Der Duden. Die deutsche Rechtschreibung“)

# Portale im Überblick

## Charakterisierung

Ein Portal ist ein gemeinsamer, integrierter **Einstiegspunkt** für personalisierten Zugriff auf relevante unternehmensweite Informationsquellen:

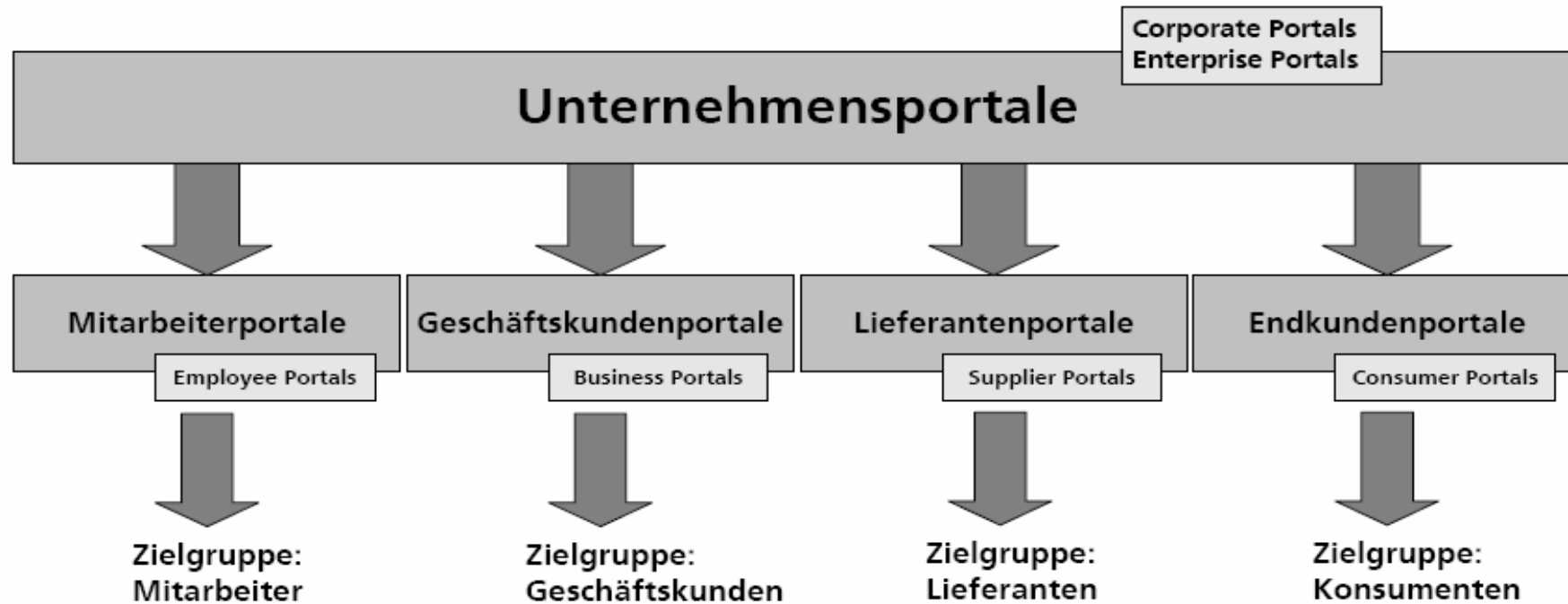
- Unternehmensapplikationen und -prozesse
- Business Intelligence
- Internetinhalte und -dienste
- Andere Online-Ressourcen

## Weitere Merkmale

- Verbindung zahlreicher Informationsquellen in einer gemeinsamen Oberfläche (-> **Oberflächenintegration**)
- Prozessunterstützung durch Integration der beteiligten Oberflächen
- Individuelle Zusammenstellung der Portalinhalte auf Benutzerebene möglich
- Abgrenzung zu CMS durch Prozessorientierung der Inhalte
- Nutzt zentrale Dienste

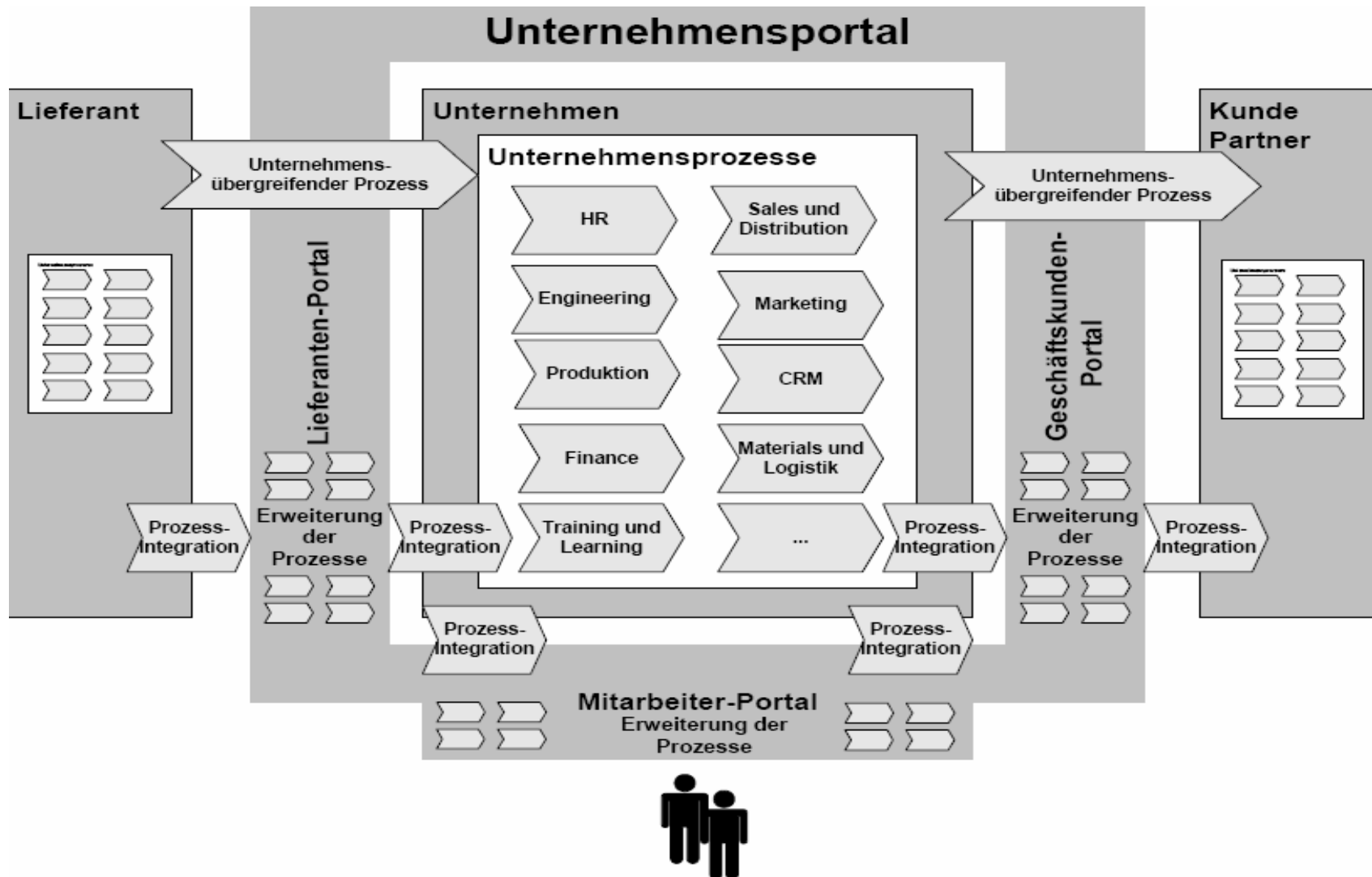
# Portale im Überblick

## Portaltypen



# Portale im Überblick

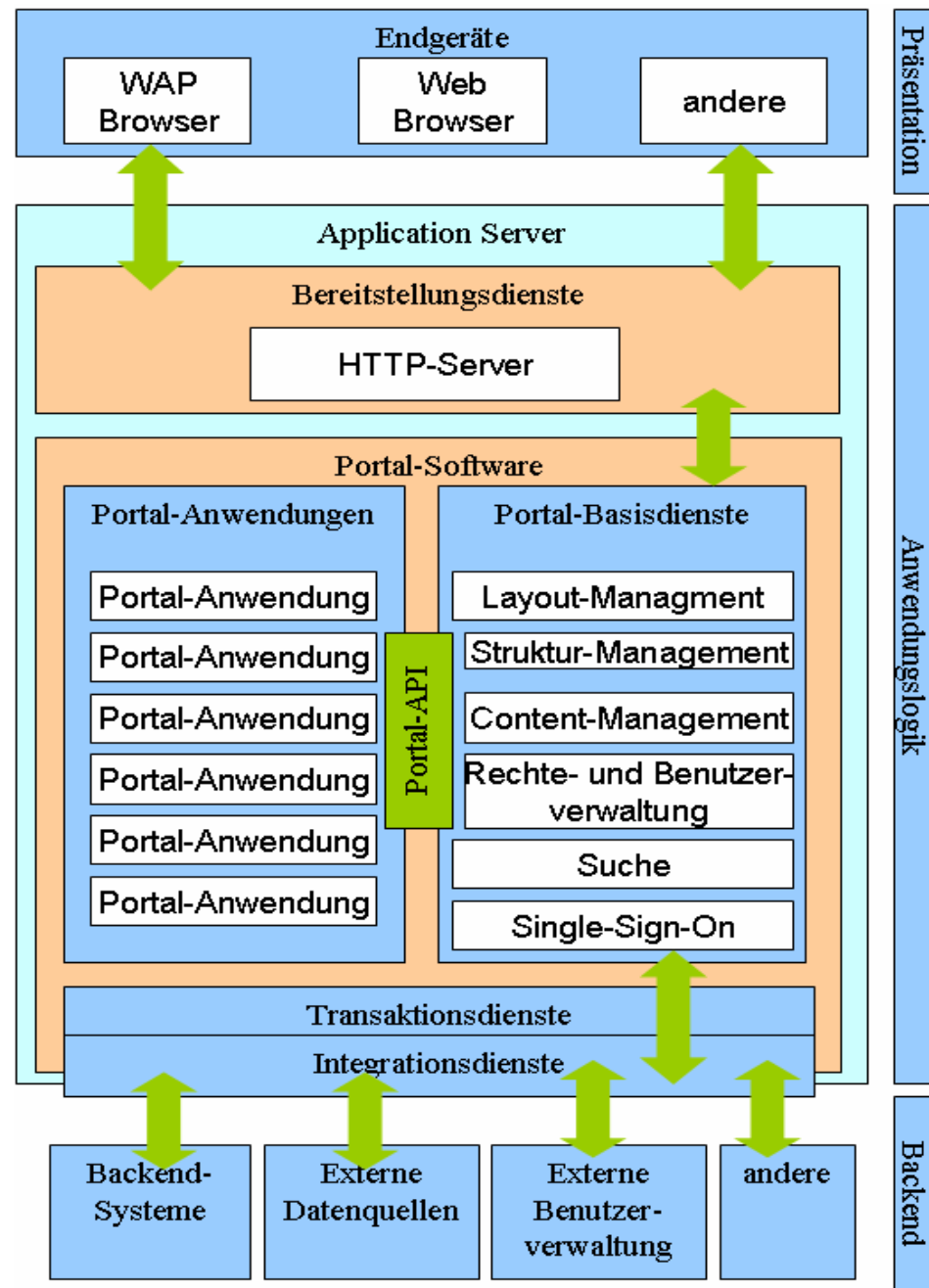
## Portale im betrieblichen Prozessumfeld



# Aufbau und Referenz-architektur

## Portal

### Portalanwendungen und Basisdienste



# Aufbau und Referenzarchitektur

## Referenzarchitektur

Die **Komponenten der Referenzarchitektur** sind nach den Technologien und Schwerpunkten der Hersteller verschieden stark ausgeprägt

**Application Server** können auf Scripting-Technologien (PHP) oder JavaEE basieren

Die Bandbreite der **Backend-Schnittstellen** reicht von einfachen DB-Schnittstellen (JDBC, ODBC) bis hin zu komplexen EAI-Funktionen

Am Markt verfügbare **Portalanwendungen** existieren in verschiedenen Formen: Portlets, I-Views, Widgets/Gadgets. Sie werden über die Portal-API aufgerufen und können selbst über diese API Basisdienste nutzen

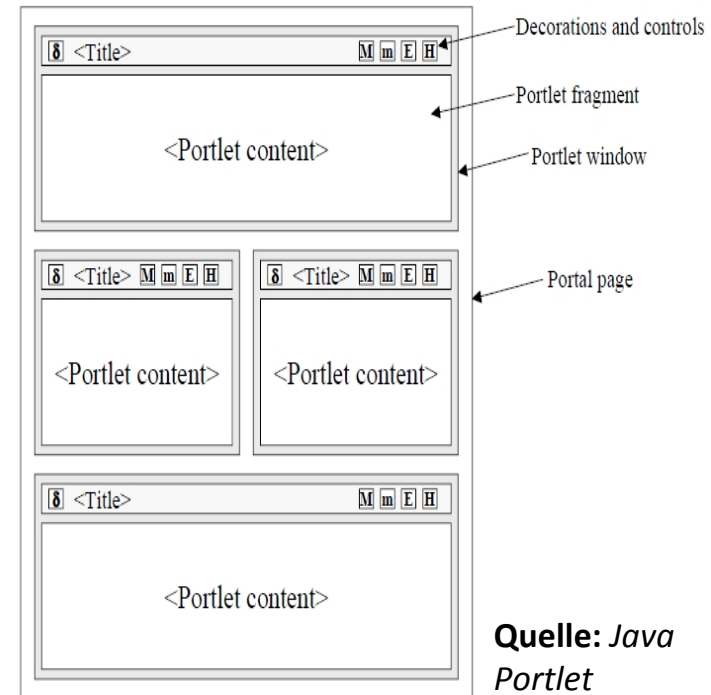
# Aufbau und Referenzarchitektur

## Basisdienste

- **Strukturmanagement:** Struktureller Aufbau und Navigierbarkeit des Portals
- **Layoutmanagement:** Rendering der Portalseiten und Erzeugung der Endgerät-spezifischen Ausgaben
- **Suche** über verschiedene, heterogene Datenbestände
- **Rechteverwaltung** erfolgt durch die Portalsoftware, die Benutzerverwaltung wird häufig durch einen externen Verzeichnisdienst realisiert
- **Content-Management** kann durch ein extern angebundenes CMS erweitert werden
- **Eränzende Module** sind z.B. Workflow-oder Groupwarefunktionalität

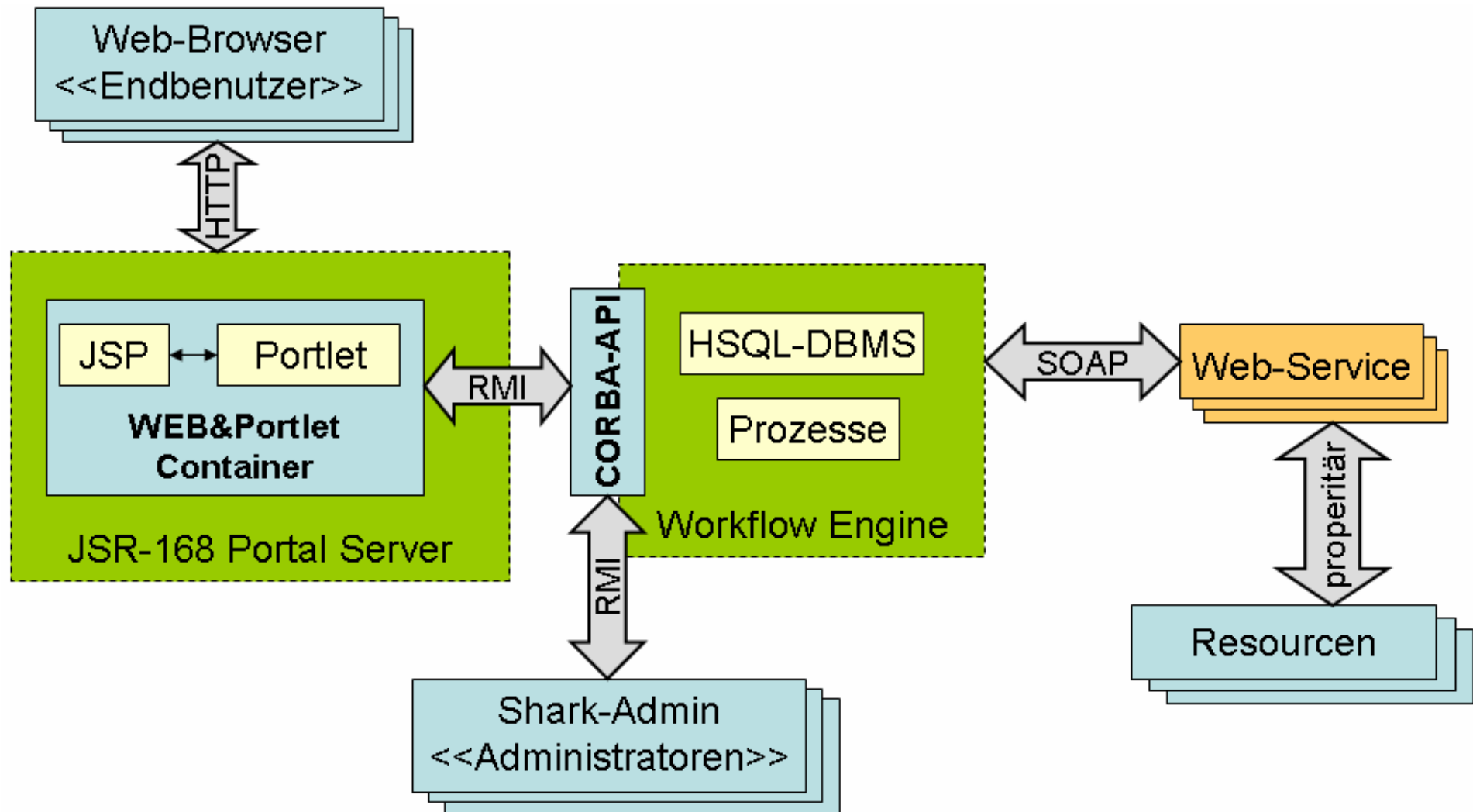
# Portalstandards

- Ziel: Wiederverwendung der Portalkomponenten zur Oberflächenintegration
- JAVA Standards:
  - Portlets 1.0* (JSR-168) und *Portlets 2.0* (JSR-286)
  - SAP iViews*
- Sprachübergreifend:
  - WSRP 1.0* und *2.0* (*Web Services for Remote Portlets*)
- Definieren:
  - Lebenszyklusmethoden und Zustände
  - Anzeige- / Datenverarbeitungsmethoden
  - Erlaubte HTML Elemente / Designklassen



**Quelle:** *Java Portlet Specification, 2003*

# Einsatz von Liferay in einem aiDa-Projekt (Enhydra Shark Workflow - XPDL Engine)



# Einordnung in die Anwendungsintegration

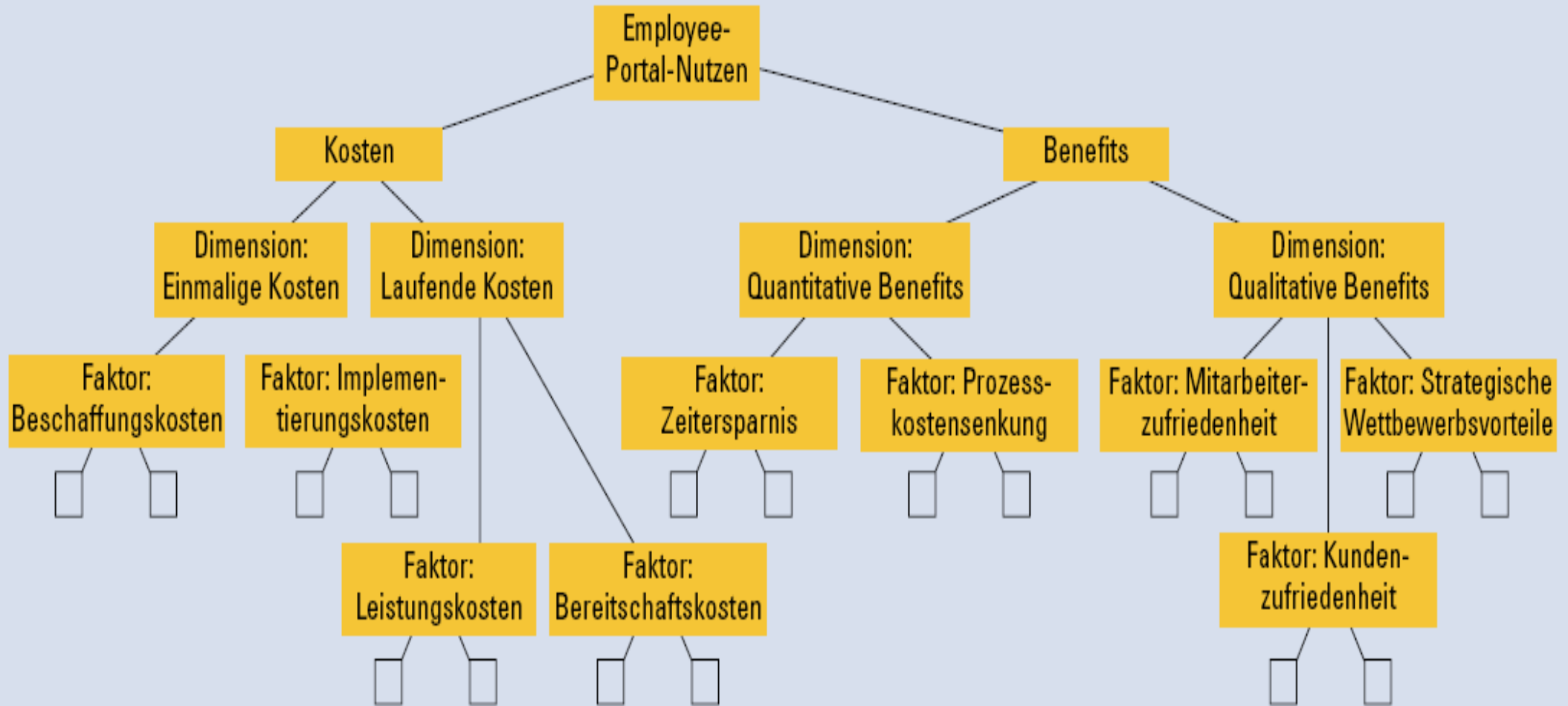
Auf der Portalseite eines Mitarbeiterportals sollen jeweils in einem Portlet folgende Informationen visualisiert werden:

- 1 Ein BPEL-Workflow für die Auftragsabwicklung
- 2 Eine Übersicht über offene Posten für belieferte Kunden aus dem Mahnwesen eines ERP-Systems
- 3 Eine OLAP-Auswertung von Lieferanten auf Basis eines Data Warehouse
- 4 Die Internetseite eines Marktplatzes für Zulieferprodukte

Stellen Sie die Beziehungen zwischen Oberflächen-, Prozess-, Funktions- und Datenintegration dar. Benutzen Sie zur Darstellung das nachfolgende SOA-Bild.

# Nutzen von Portalen

## Empirisch ermitteltes Nutzenmodell eines Employee-Portals



# Zusammenfassung - Schichtenmodell

