INFORMATIONSINTEGRATION UND WEBPORTALE

Informationsintegration: Architekturen und Systeme
Dr. Simone Braun

Klick-And-Bau

Informationsintegration und Webportale

bisher...

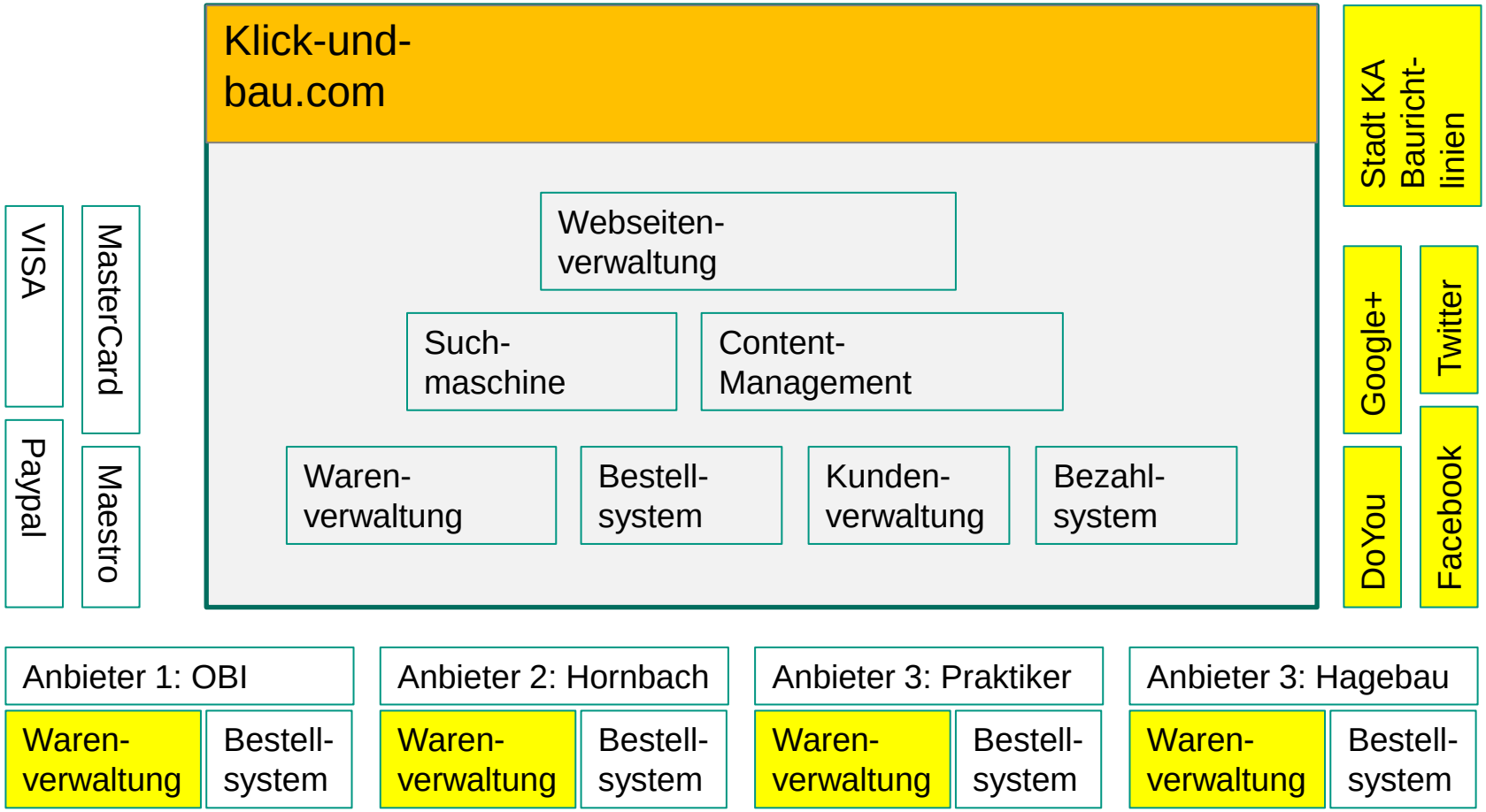
- Integration von Diensten innerhalb eines Unternehmens mittels Middleware
  - Applikationsserver: J2EE als Technologie
  - Enterprise Application Integration als Methode
- Anbindung der eigenen Systeme an das Web
  - Web-Technologien
  - Mobile Web Apps
- Immer bestand eine relativ große Kontrolle über die zu integrierenden Systeme bzw. die Systemumgebung.
  - Wir konnten etwas verändern
  - Beteiligte Systeme relativ verlässlich (LAN)

jetzt...

- Um das Angebot zu vergrößern und zu verbessern, sollen weitere Anbieter in das Portal mit aufgenommen werden.
  - Anbieter mit anderen Schwerpunkten
  - Verbraucherurteile über Produkte
  - Behördeninformationen zu Bauvorschriften

  → Letztes Mal auf Datenbankebene, heute Informationsebene

Baumärkteportal: Inhalte und Quellen



Randbedingungen

- Die Einbindung der neuen Angebote soll integriert erfolgen, sodass sich für den Benutzer eine einheitliche Sicht ergibt.
- Die Systeme der eingebundenen Partner bleiben autonom und können sich unabhängig vom Portal weiterentwickeln.
- Für die Einbindung sollen keine großen Änderungen an den einzubindenden Dienste notwendig werden.

Neu: Aspekt der Autonomie

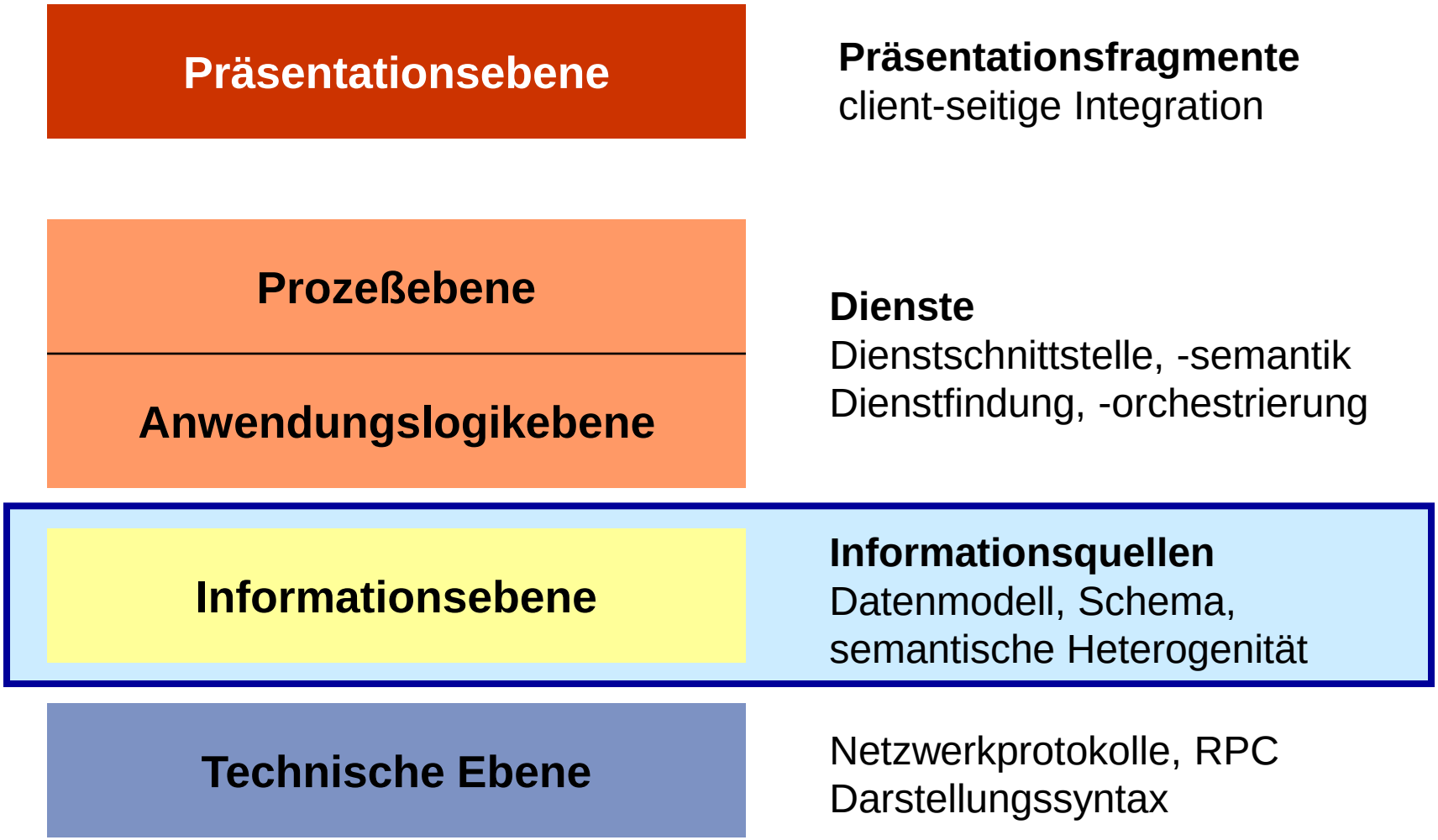
- Bezieht sich auf Kontrolle nicht auf Daten
- Klassen nach Özsu & Valduriez 1999
  - Design-, Kommunikations- und Ausführungs-Autonomie

- Beschränkter Lösungsraum
  - Keine Eingriffsmöglichkeiten in Infrastruktur oder Informationsmodelle
  - Insbesondere: keine Optimierungen bzgl. Performanz
- Unzuverlässigkeit
  - Netzwerk
  - Dienste
- Dynamik
  - Unabhängige (unkoordinierte) Weiterentwicklung der einzelnen Partner
    - Existenz mit eingeschlossen!

Übersicht

- Integrationsebenen
- Informationsintegration: Architekturen
  - Materialisierte Integration
  - Virtuelle Integration (I^3)
    - Referenzarchitektur
    - Semistrukturiertes Datenmodell
    - Problembereiche
- Informationsintegration: Semantische Integration
  - Datenmodellkonflikte
  - Schemakonflikte
  - Konflikte auf Instanzebene
- Vergleich & Fazit

Integrationsebenen (1)



Integrationsebenen (2)

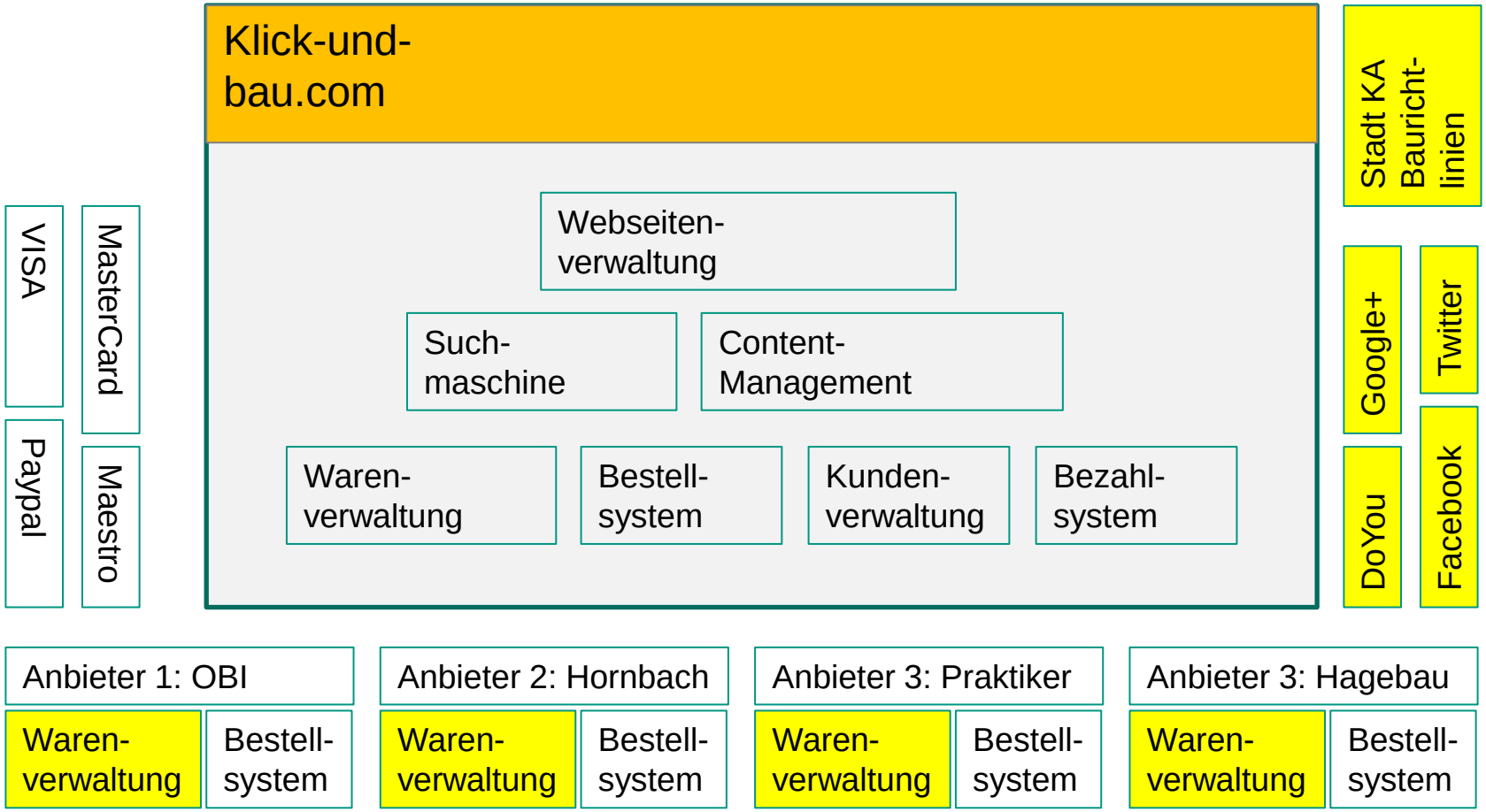
- Wir wollen heute nur eine **Informationsperspektive** einnehmen
  - Einzubindende Anbieter sind **Informationsquellen**
  - Interaktion mit den externen Quellen läuft über das Anfrage-Ergebnis-Paradigma
  - Es existiert eine zentrale Stelle, an der alle Informationen zusammenlaufen
  - Hauptprobleme
    - Wie überwinde ich die technische Heterogenität?
    - Wie überwinde ich die semantische Heterogenität?

Integrationsebenen (2)

- Andere Möglichkeit: **Dienstintegration**
  - Einzubindende Anbieter sind **Dienste**
  - Autonomen Dienste soll durch eine Infrastruktur die gegenseitige Nutzung ermöglicht werden

    → Dienstorientierte Architekturen
  - Hauptprobleme
    - Wie mache ich Dienste interoperabel?
    - Wie finde ich benötigte Dienste?
    - Wie beschreibe ich Dienste?
  - Wird in der Vorlesung über Dienstorientierte Integration von Komponenten am 8.12.2014 behandelt
- Perspektiven schließen sich nicht gegenseitig aus, sondern ergänzen sich!

Baumärkteportal: Inhalte und Quellen



Szenario – Beispiel-Probleme (1)

- Die einzubindenden Systeme sind sehr heterogen:
  - Anbieter 1 OBI besitzt eine Oracle-Datenbank, auf die man über JDBC/Spring zugreifen könnte.
  - Anbieter 2 Hornbach bietet eine Webservice-Schnittstelle an, über die wir auf den Informationsbestand zugreifen könnten.
  - Anbieter 3 Praktiker setzt ein XML-Datenbanksystem ein. Wir könnten mittels XML-Standards (XPath, XQuery) darauf zugreifen.
  - Anbieter 4 Hagebau hat ein Angebot auf der Basis von HTML-Seiten, auf die nur mittels HTTP zugegriffen werden kann.
- Alle verwenden unterschiedliche Schemata...

Szenario – Probleme (2)

- Heterogenität hinsichtlich Zugriff
  - Anfragemöglichkeit
    - Anfragesprache, parametrisierte Funktionen, Formulare
  - Austauschformat
    - Binärdaten, XML, HTML etc.
  - Zugriffsprotokoll
    - HTTP
    - JDBC
    - SOAP

→ Technische Heterogenität

Szenario – Probleme (3)

- Heterogenität hinsichtlich Darstellung
  - Datentypen
    - Boolean vs. Bit, float vs. decimal
  - Zeichensätze und Kodierungen
    - Unicode, UTF-8, ASCII etc.
  - Datenformate
    - Datum: 01.12.2014 vs. 1. Dezember 2014 vs. 2014-12-04
    - Währungen: EUR vs. Euro vs. €
  - Dateiformate
    - XML, CSV, TXT etc.

→ Syntaktische Heterogenität

Szenario – Probleme (4)

- Heterogenität hinsichtlich Datenmodell und Schema
  - Datenmodell
    - Relational vs. XML vs. Objektorientiert
  - Schemata
    - Unterschiedliche Modellierungsebene → als Wert vs. Attribut vs. Relation
    - Unterschiedliche Verteilung von Attributen auf Tabellen
    - Fehlende Attribute

→ Datenmodell- und Strukturelle Heterogenität

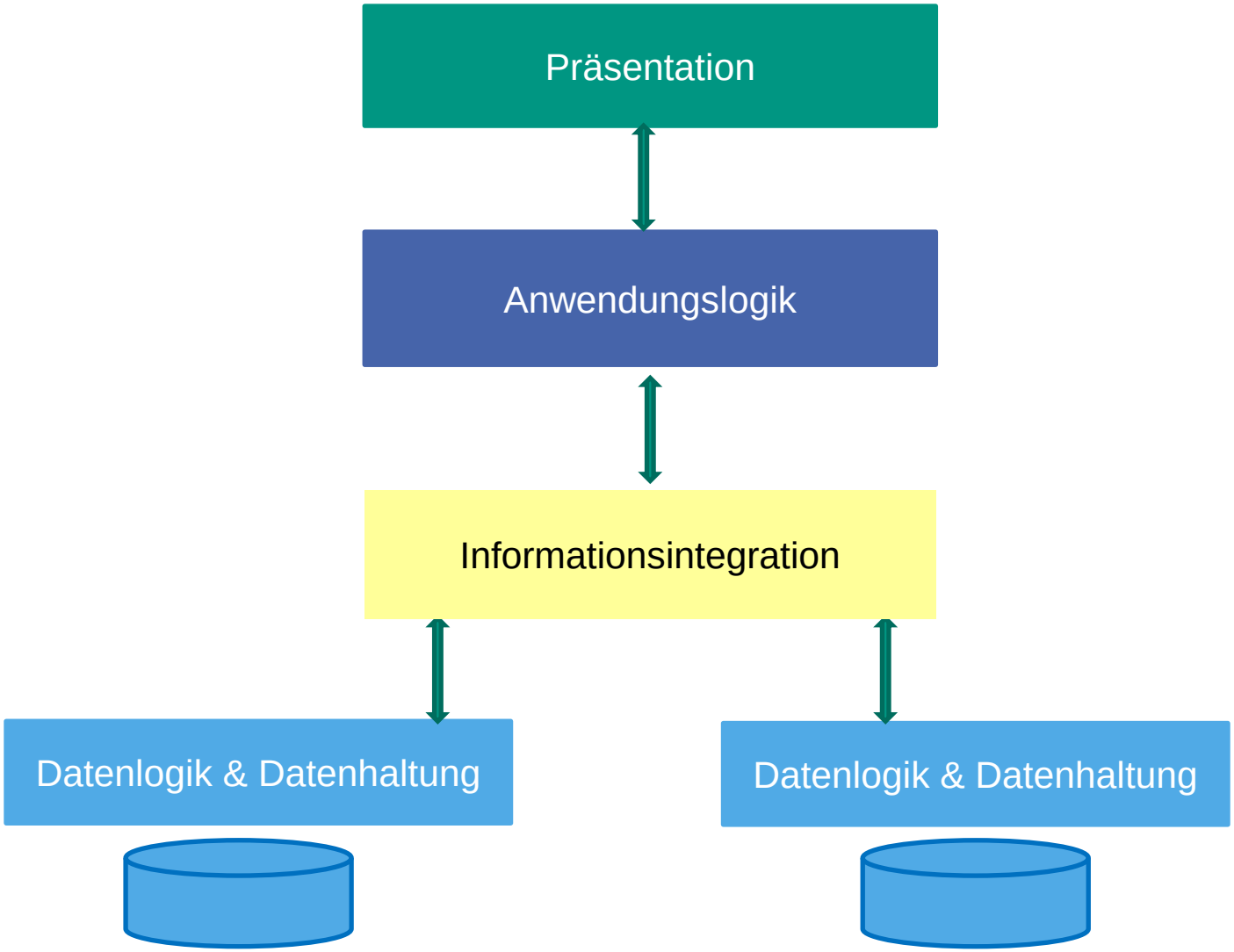
Szenario – Probleme (4)

- Heterogenität hinsichtlich Bedeutung
  - Definition von Konzepten
    - Was zählt zu „Mitarbeiter“? Was bedeutet „NULL“?
  - Synonymie, Homonymie, Polysemie etc.
    - Fahrstuhl/Aufzug/Lift, Bank, Feder
  - Unterschiedliche Skalen
    - km/h vs. mph, $ vs. €

→ Semantische Heterogenität

Anbindung: Naiver Ansatz

- Die Portalanwendung greift direkt auf die einzubindenden Datenquellen zu
  - Anpassung von Protokoll, Format, Schema und Anfragesprache in der Portalanwendung selbst
- Nachteil
  - Jede neue Quelle und jede Änderung an bestehenden Quellen zieht eine Änderung an der Portalanwendung nach sich
  - Kaum wartbare und beherrschbare Lösung
    - keine Skalierbarkeit
- Deshalb: Entkopplung durch eine Zwischenschicht, die eine integrierte Sicht zur Verfügung stellt


Präsentation
Anwendungslogik
Informationsintegration
Datenlogik & Datenhaltung
Datenlogik & Datenhaltung

Anbindung: Virtuell vs. materialisiert

- Aufbau einer zentralen Datenbasis im Portal, in die die Inhalte der angeschlossenen Anbieter in einem Vorverarbeitungsschritt importiert werden. Diese Datenbasis stellt die integrierte Sicht dar

  materialisierter (physischer) Ansatz
  a priori Integration

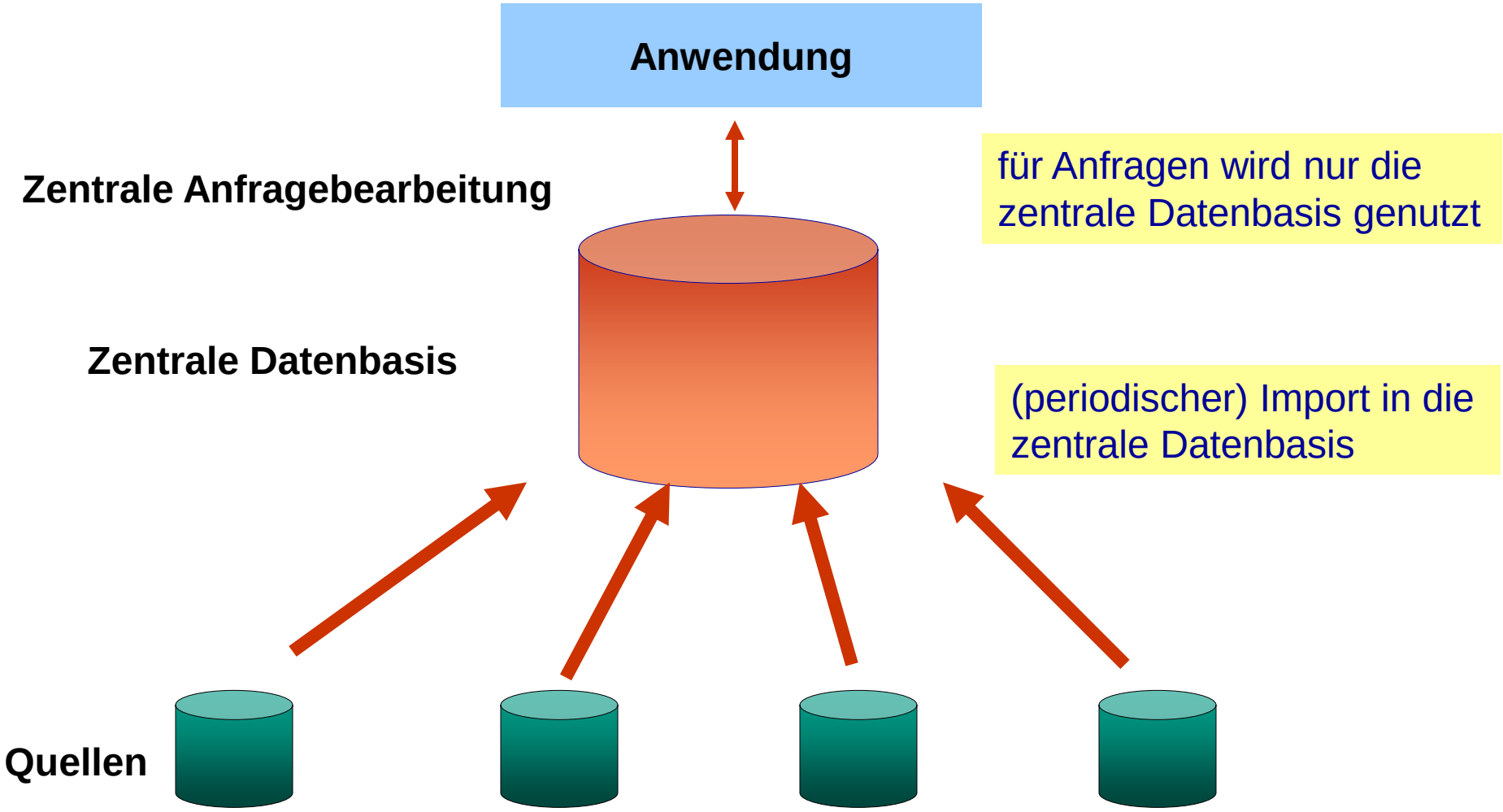
- Nutzung der Systeme der Fremdanbieter für die Anfrageauswertung (Durchreichen von Anfragen). Die integrierte Sicht ist physisch nicht vorhanden; sie wird von Mediatoren dynamisch bereitgestellt:

  virtueller (logischer) Ansatz
  a posteriori Integration bei Bedarf

MATERIALISIERTER ANSATZ

Materialisierter Ansatz



Materialisierter Ansatz – Aufgaben

- Wie kommen die Daten in die Datenbasis?

- **Push**: Die Datenquellen liefern die Daten in einem bestimmten Austauschformat
  - UN/EDIFACT bzw. X.12, XML, ...
  - Werden über ETL Prozesse geladen
  - Für B2B wichtigste Alternative
  - Einigung der kooperierenden Partner erforderlich
    - Starker Eingriff in die Autonomie
- **Pull**: Die Daten werden durch Dienste der Datenquellen gesammelt (Crawling, vgl. Suchmaschinen)
  - Ähnliche Probleme wie bei virtueller Integration

- Effiziente Importmechanismen für große Datenmengen

Materialisierter Ansatz – Aufgaben

- Wie werden die Daten aktuell gehalten?
  - Erkennung von Änderungen
    - meist organisatorische und technische Eingriffe in die Systeme erforderlich
    - Oder: Suchmaschinenstrategie
  - Effiziente Durchführung der Aktualisierung der Datenbasis

Materialisierter Ansatz – Vorteile

- Einfach zu realisieren
  - Anwendungsentwicklung unterscheidet sich durch zentrale Datenbasis kaum vom Ein-Quellen-Fall
  - Mehr Informationen über vorhandene Daten
- Performant
  - Direkte Datenbankzugriffe für Anfrageauswertung
  - Entkopplung von (evtl. langsamen, nur teilweise verfügbaren) externen Systemen
  - gezielte Optimierungen möglich
- Nachbearbeitungsoperationen möglich
  - Von den Fremdanbietern gelieferte Daten können (auch aufwendig) geprüft und bereinigt werden
  - Aggregation von Daten ebenfalls leicht möglich

Materialisierter Ansatz: Nachteile

- (redundante) Speicherung evtl. großer Datenmengen
  - leistungsfähige Infrastruktur auf Portal-Seite erforderlich
- Aktualität der Daten ist nicht gewährleistet
  - klassisches Caching-Problem
- Aktualisierung
  - auf Initiative der Datenquellen
    - organisatorische Maßnahmen erforderlich
    - Insbesondere: wir brauchen ein Austauschformat!
  - Aktualisierung auf Initiative des Portals
    - bei großen Datenmengen häufig impraktikabel
    - keine Information, was geändert wurde
- Keine Kontrolle des Urhebers mehr über die Daten!

Data Warehousing

- Auch unternehmensintern kann eine lose Kopplung ansonsten autonomer Teilsysteme sinnvoll sein.
- Entkopplung der operationalen Systeme von Systemen zur Auswertung und zur Entscheidungsunterstützung

- zentrale Datenbasis heißt dort **Data Warehouse**
- Weitere Aggregationsebenen für spezifische Auswertungen: **Data Marts**

- Zentrale Probleme dort: Performanz bei großen Datenvolumina

→ Vorlesung Prof. Böhm: Analyse großer Datenbestände

VIRTUELLE INTEGRATION

Virtueller Ansatz

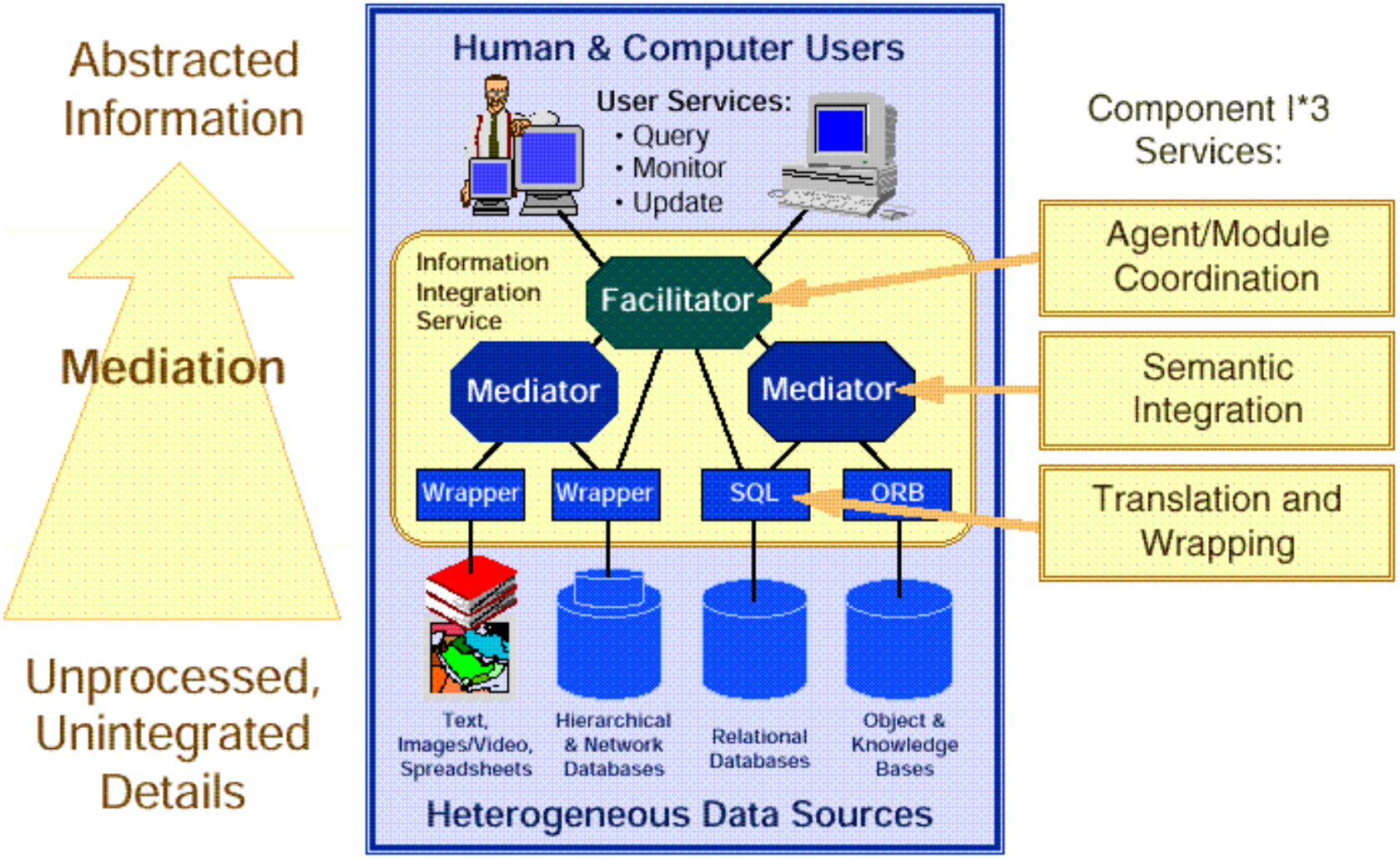
- Idee: Lasse die Daten in den Quellsystemen und benutze für jede Benutzeranfrage deren Anfragemöglichkeiten

- Entkopplung von Datenquellen und Anwendungen durch eine virtuelle Sicht:

Mediatorarchitektur

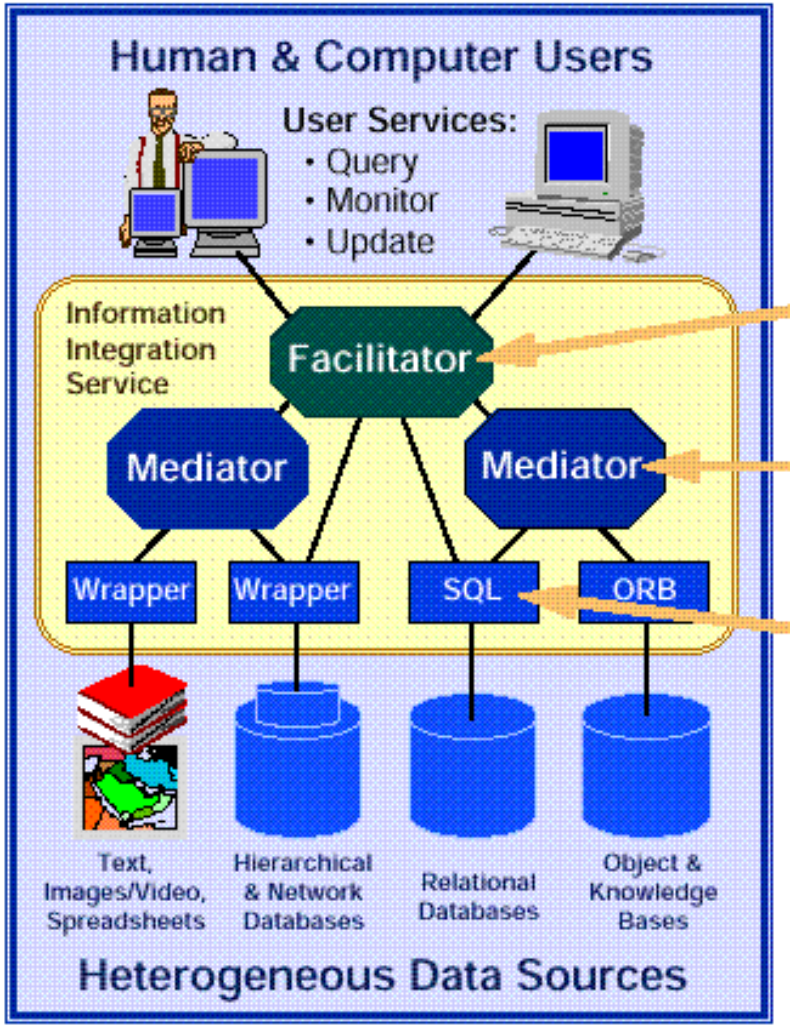
- Konzept der Mediator-Wrapper-Architektur von Gio Wiederhold 1992

I³-Referenzarchitektur nach [AHK95]



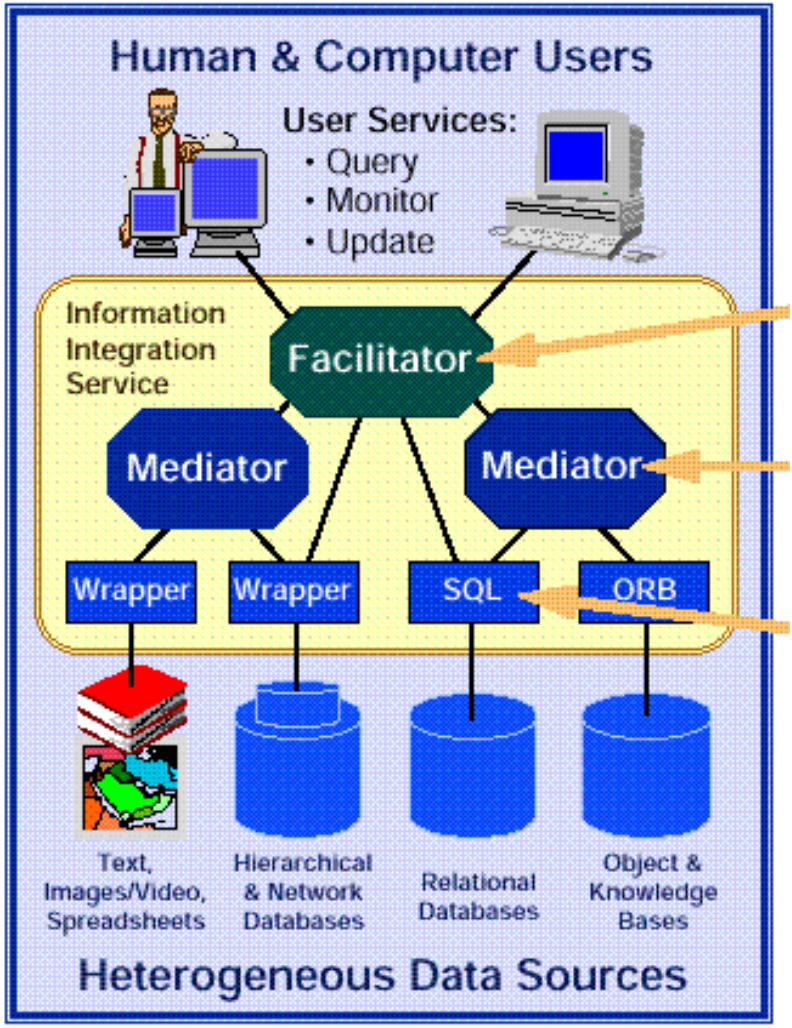
I³-Referenzarchitektur nach [AHK95]

- Facilitator:
  - Anlaufstelle für Anwendungen
  - Koordinator
  - Zuständig für Suche und Auswahl relevanter Quellen
  - Zusammenfassung zur Präsentation



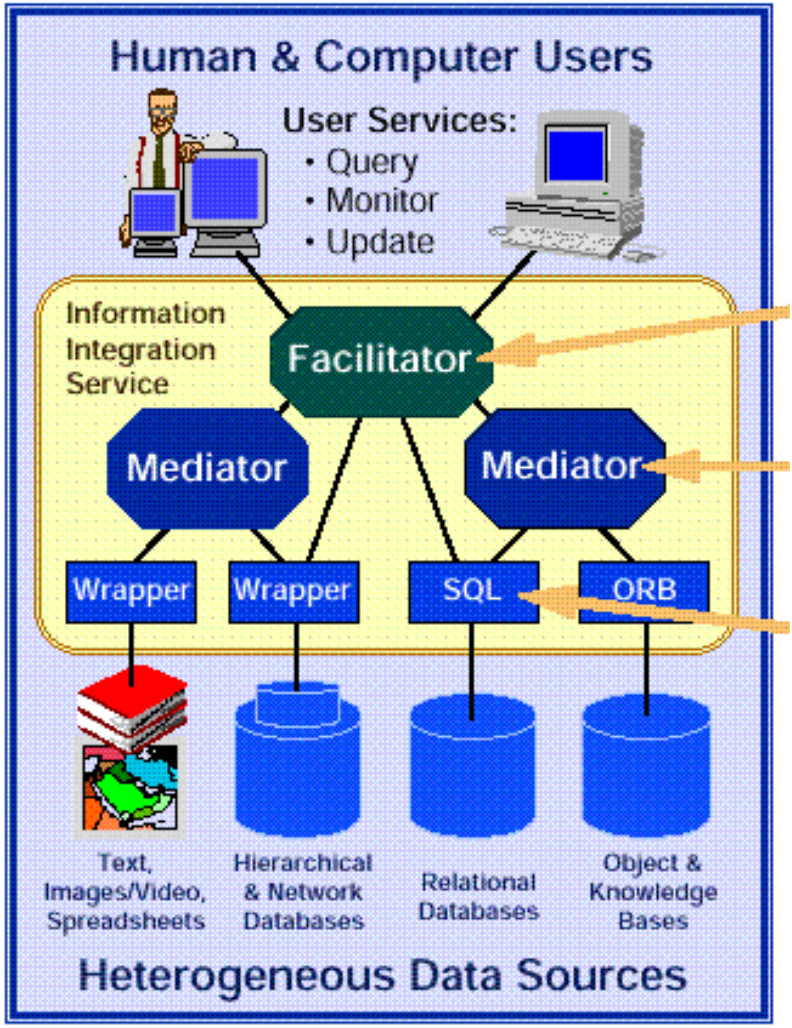
I³-Referenzarchitektur nach [AHK95]

- Mediator:
  - Überwindet semantische Heterogenität
  - Middleware zw. Informationsquelle und Anwendung
  - Beinhaltet Domänenspezifischen Code
  - Macht Anfragezerlegung
  - Transformiert Daten zu Informationen
  - Sollte klein & einfach sein, um von kleiner Expertengruppe gewartet werden zu können
  - → Einfaches globales Schema, begrenzte Domäne, einfache Schnittstellen

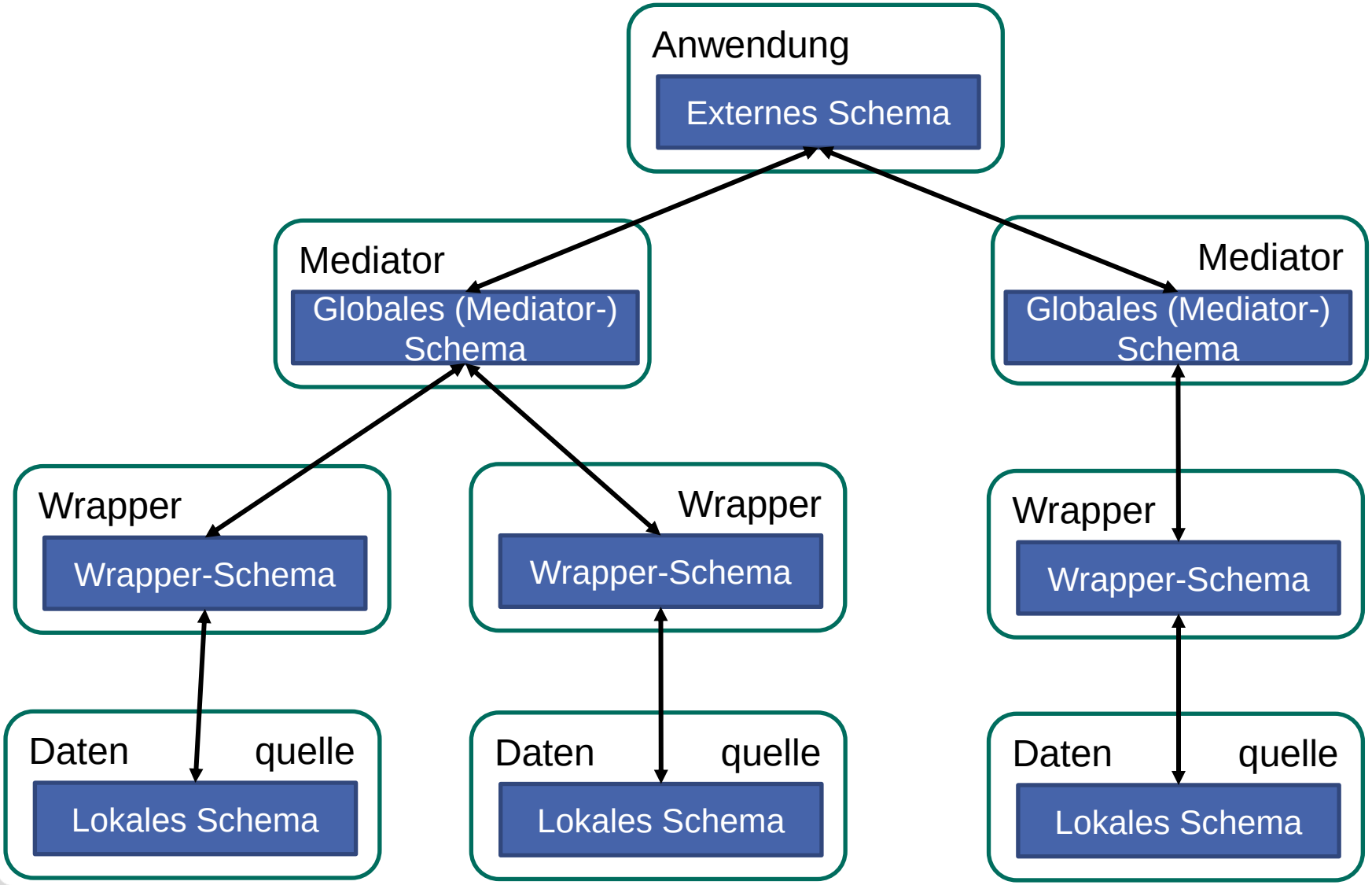


I³-Referenzarchitektur nach [AHK95]

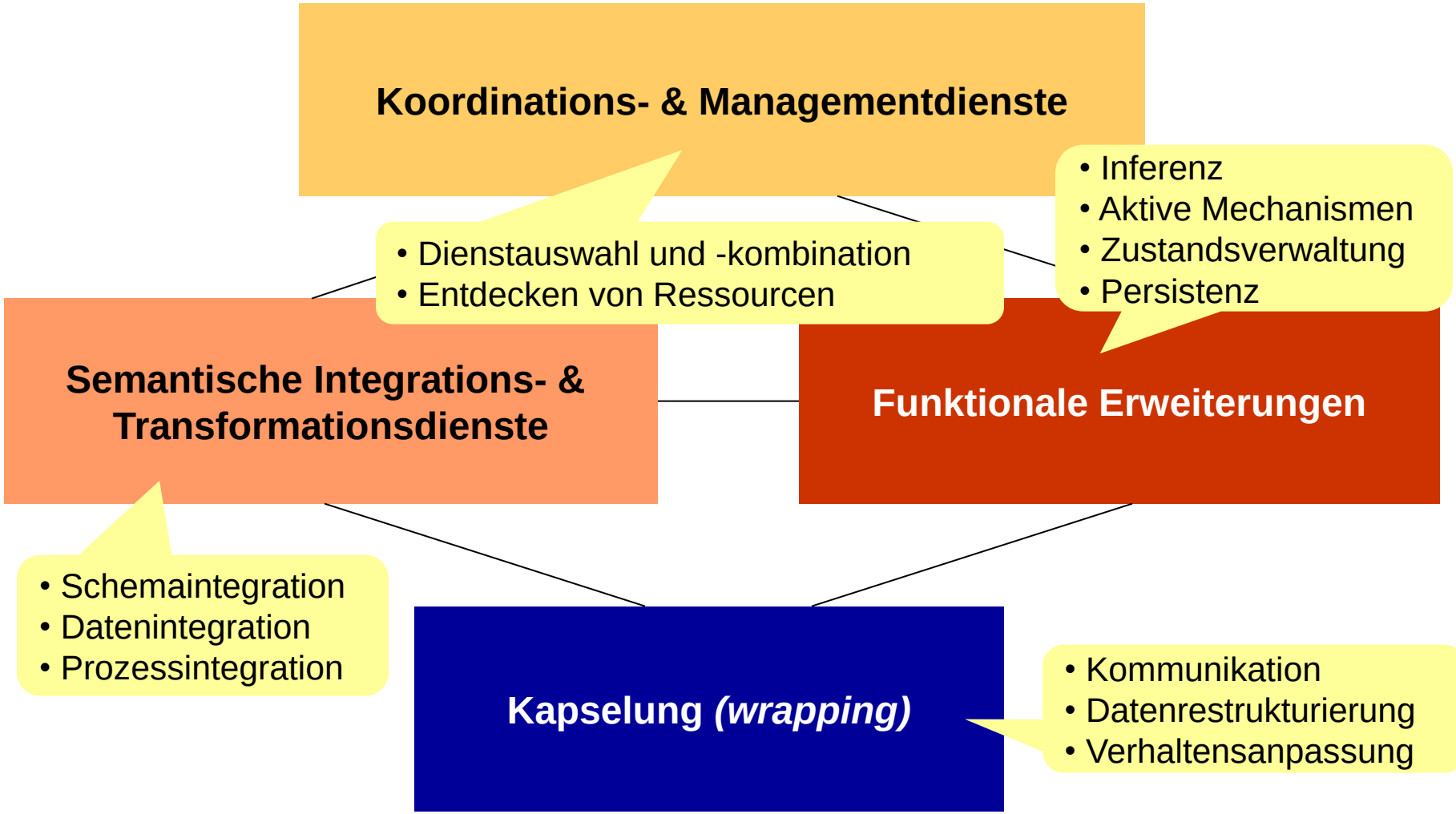
- Wrapper:
  - Überwindet technische Heterogenität
  - Vereinheitlicht Schnittstellen durch Abtraktion
  - Softwarekomponente zur Vermittlung zw. Mediator und Quelle
  - Enthält Quellen-spezifischen Code
  - Mapping vom lokalen Schema der Quelle ins globale Schema
  - Sollte schnell implementiert, wiederverwendbar und lokal wartbar sein



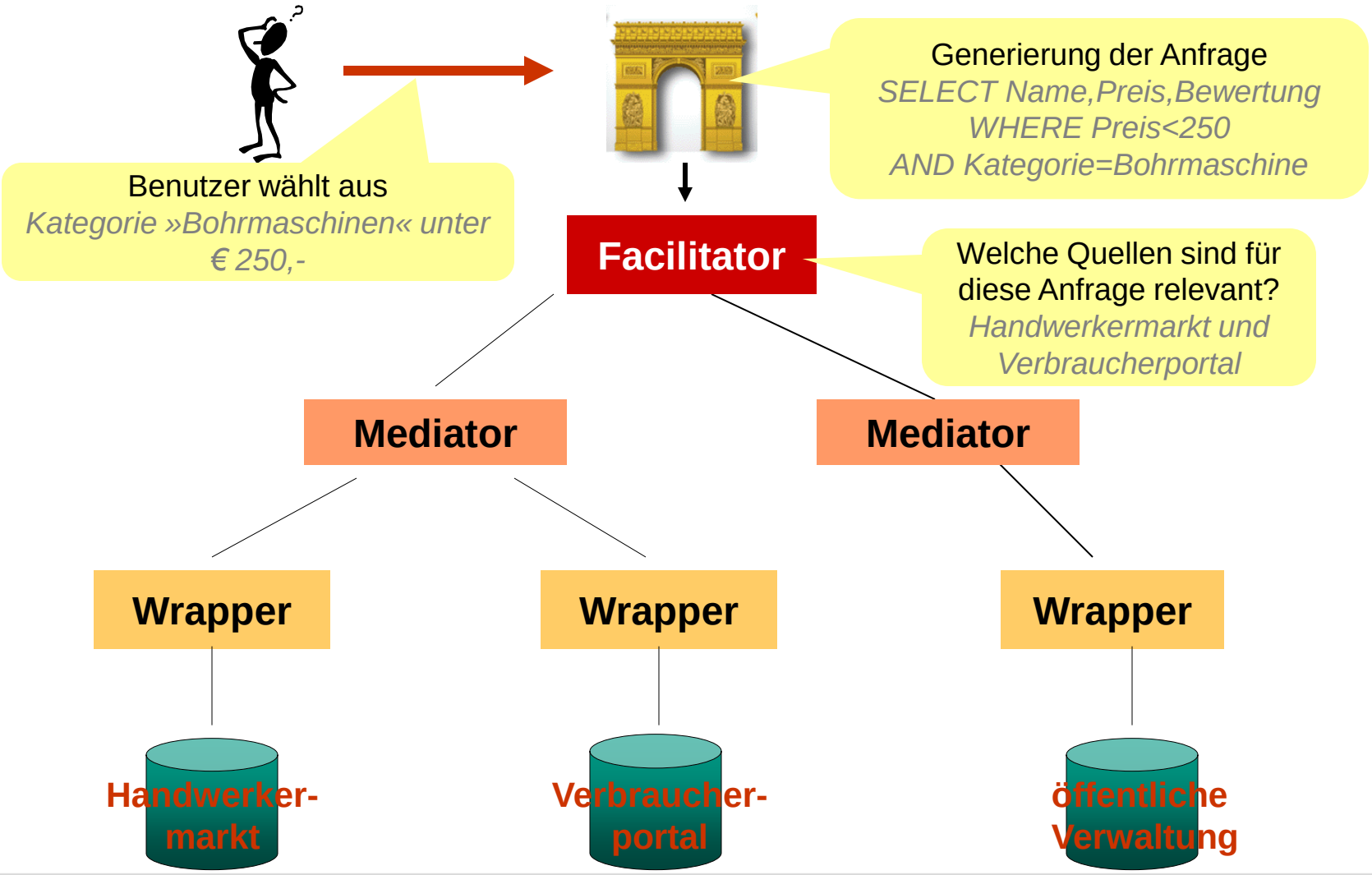
Intelligent Integration of Information



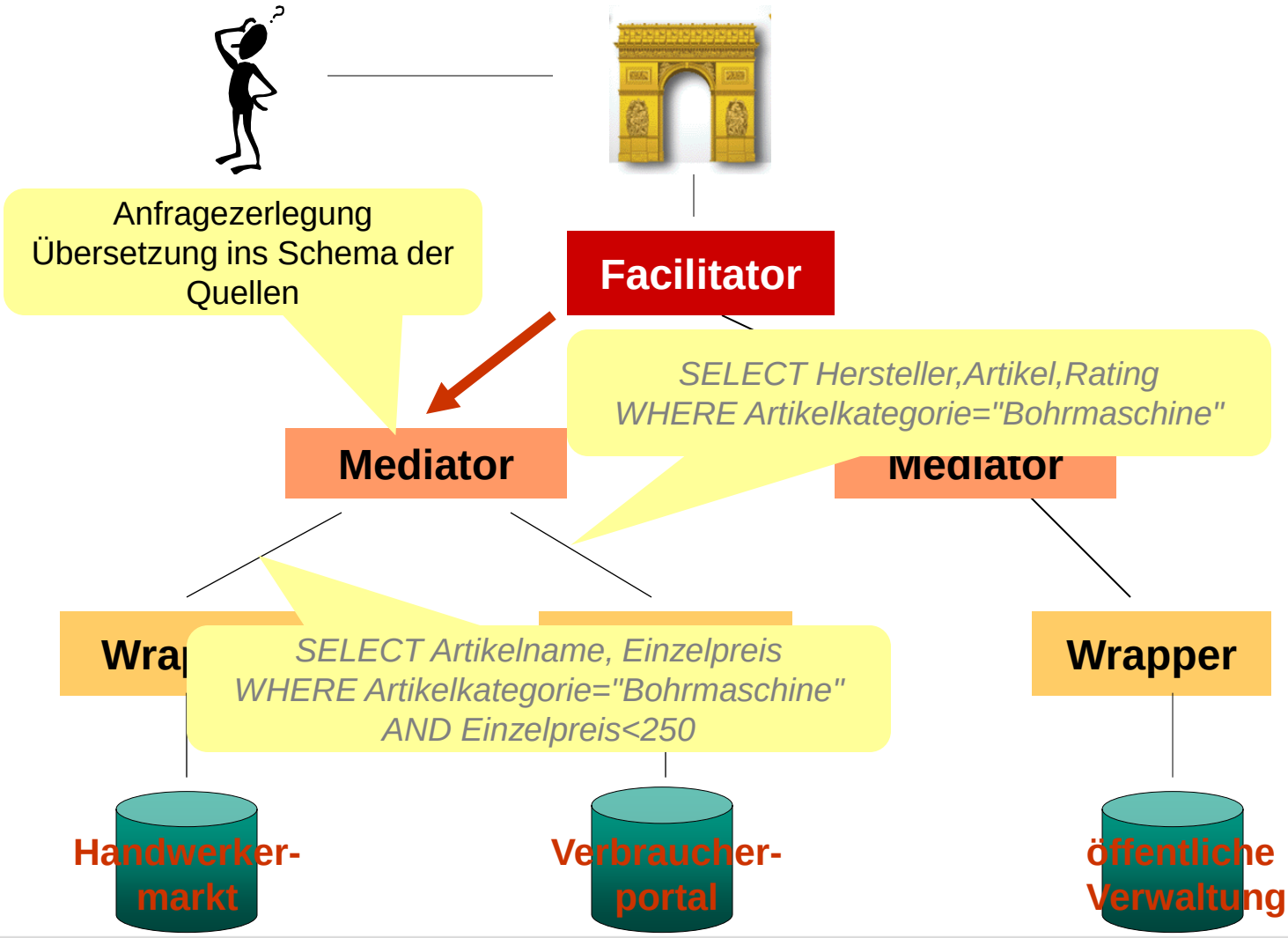
Intelligent Integration of Information (I³)



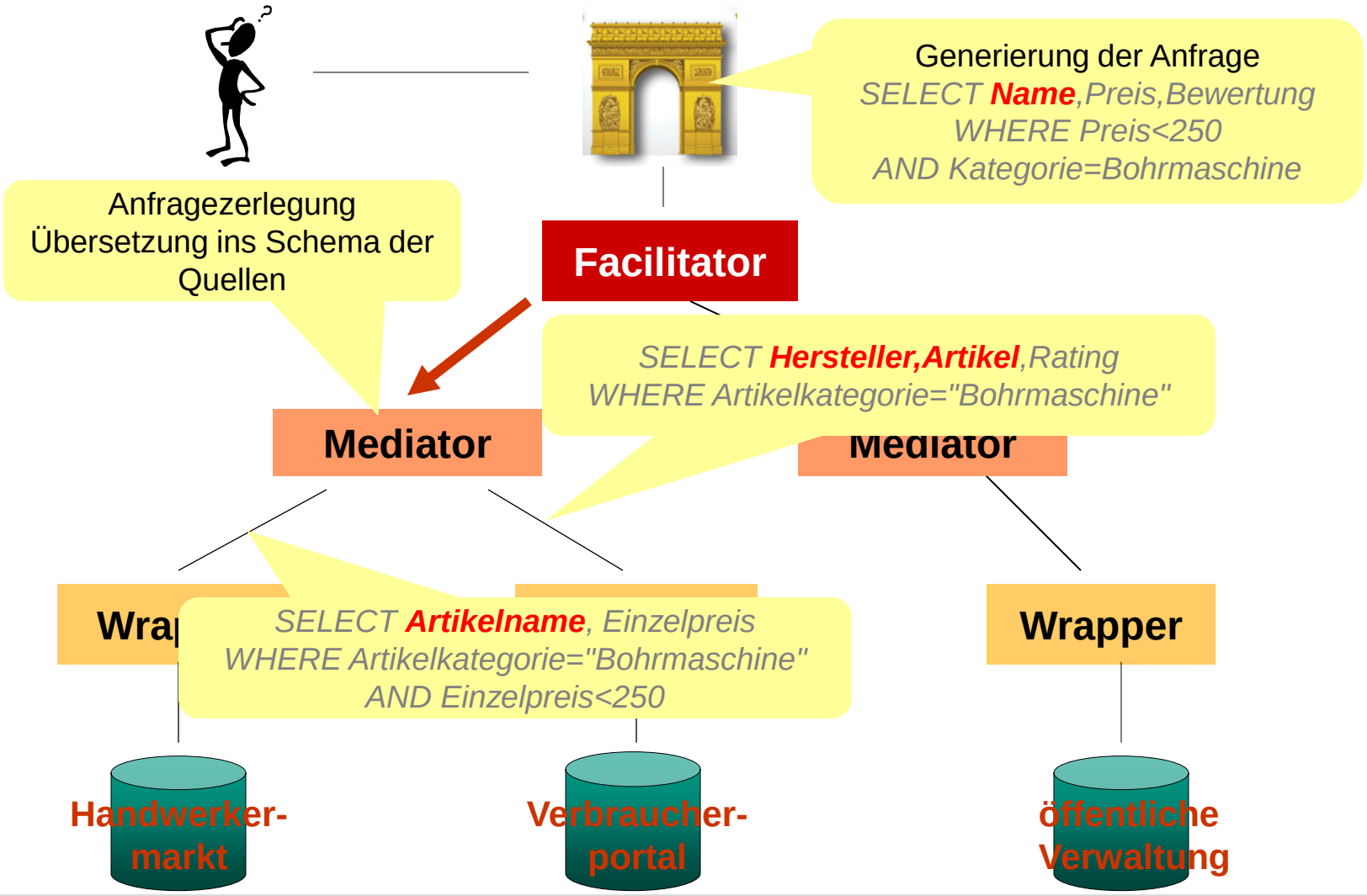
Virtuelle Integration – Beispiel



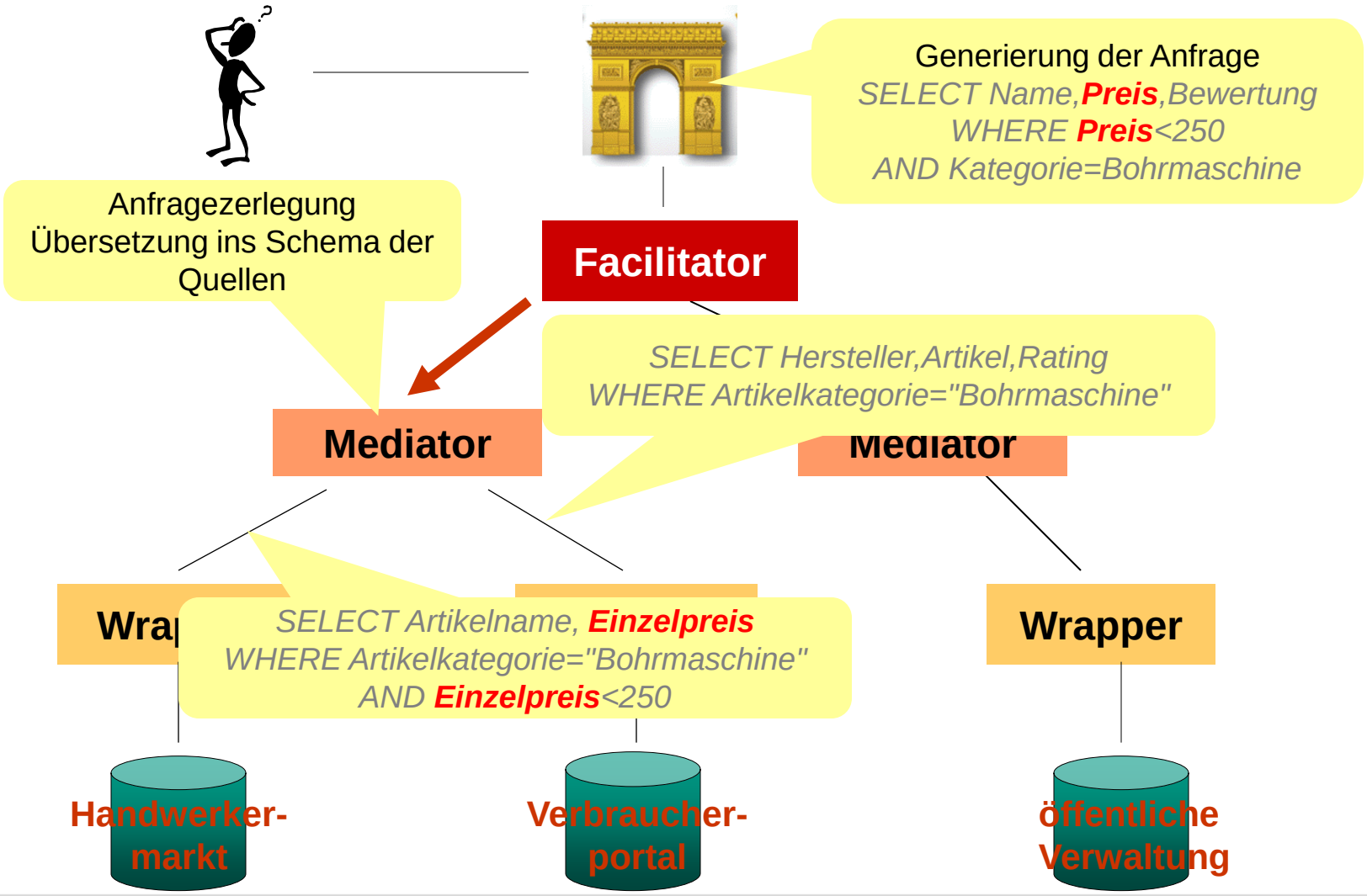
Virtuelle Integration – Beispiel



Virtuelle Integration – Beispiel



Virtuelle Integration – Beispiel



Virtuelle Integration – Beispiel

Generierung der Anfrage
SELECT Name,Preis,Bewertung
WHERE Preis<250
*AND **Kategorie**=Bohrmaschine*

Anfragezerlegung
Übersetzung ins Schema der Quellen

Facilitator

SELECT Hersteller,Artikel,Rating
*WHERE **Artikelkategorie**="Bohrmaschine"*

Mediator

Mediator

Wra

SELECT Artikelname, Einzelpreis
*WHERE **Artikelkategorie**="Bohrmaschine"*
AND Einzelpreis<250

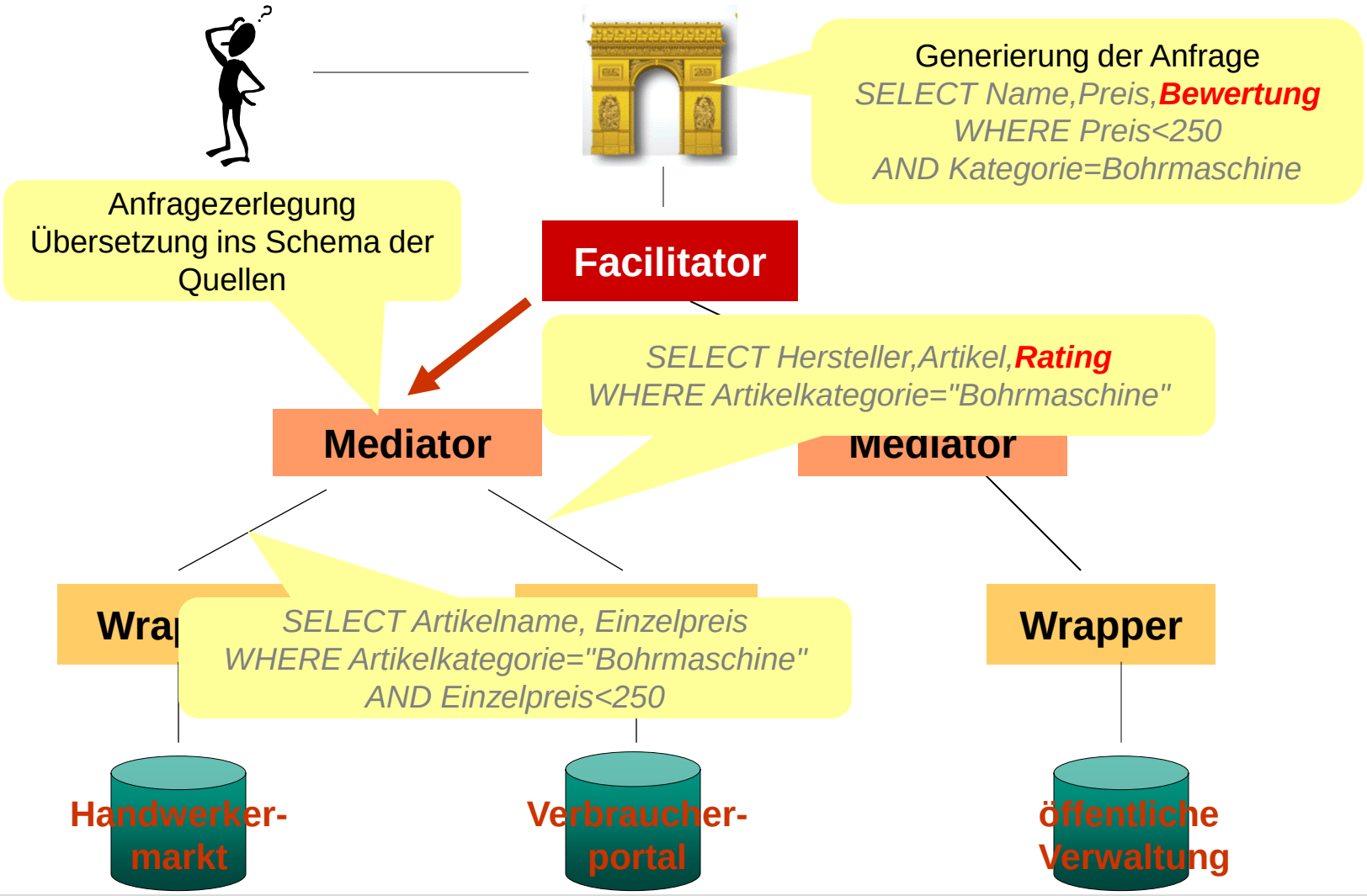
Wrapper

Handwerker-markt

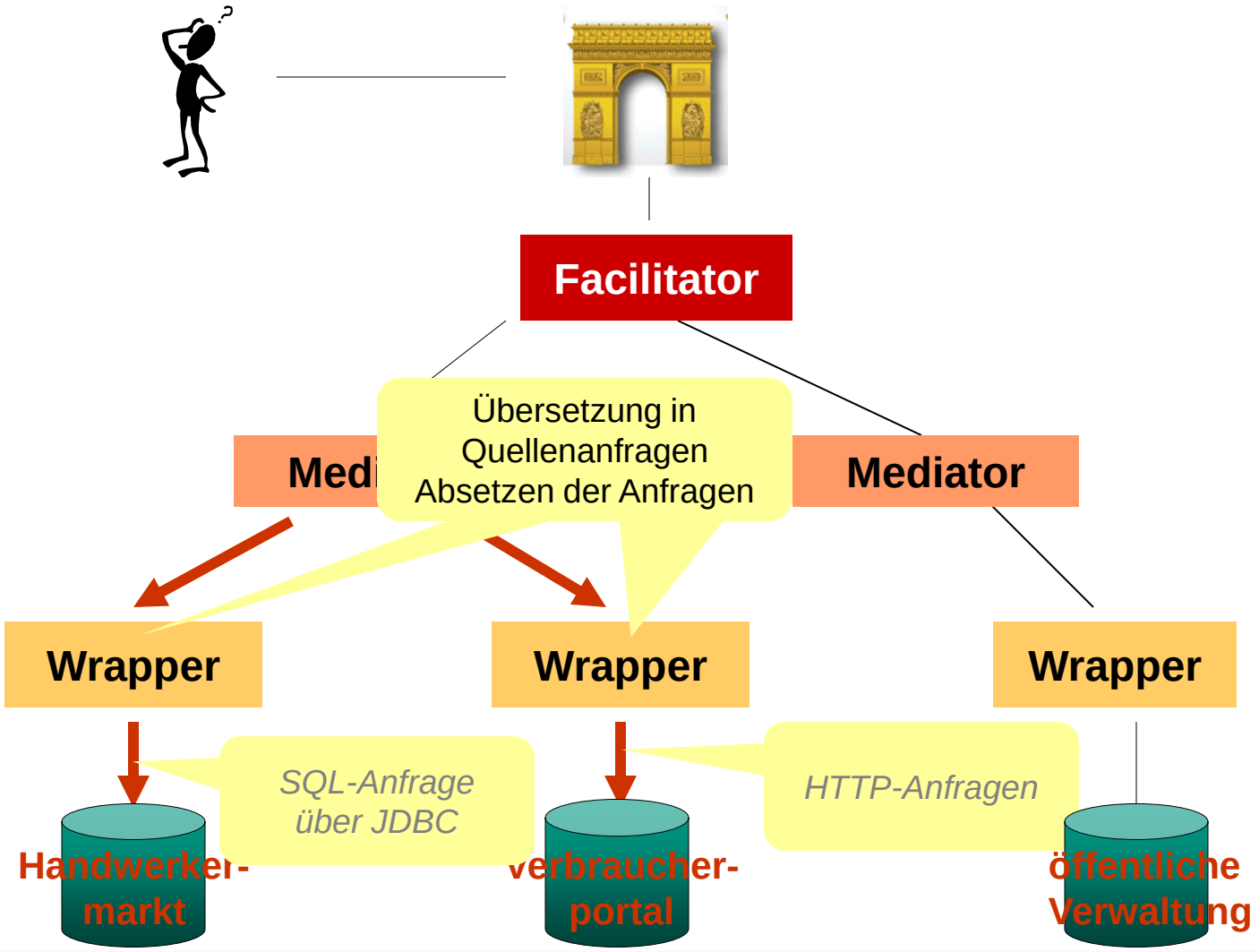
Verbraucher-portal

öffentliche Verwaltung

Virtuelle Integration – Beispiel



Virtuelle Integration – Beispiel



Virtuelle Integration – Beispiel

Facilitator

Zusammenführung der Ergebnisse einer Quelle
Transformation ins gemeinsame Datenmodell
Ausführung von Filteroperationen

Wrapper | **Wrapper** | **Wrapper**

JDBC-ResultSet

Quellen liefern Ergebnis zurück

HTML-Seite

Handwerker-markt | **Verbraucher-portal** | **öffentliche Verwaltung**

Virtuelle Integration – Beispiel

Aufbereitung der Ergebnisse für den Benutzer

Übersetzung ins Informationsmodell des Portales
z.B. Artikelname -> Name
Verschmelzen der Ergebnismengen

Facilitator

Sammeln der Ergebnisse

Mediator

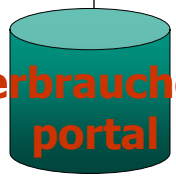
Mediator

Wrapper

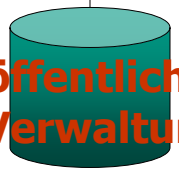
Wrapper

Wrapper

Handwerker-markt

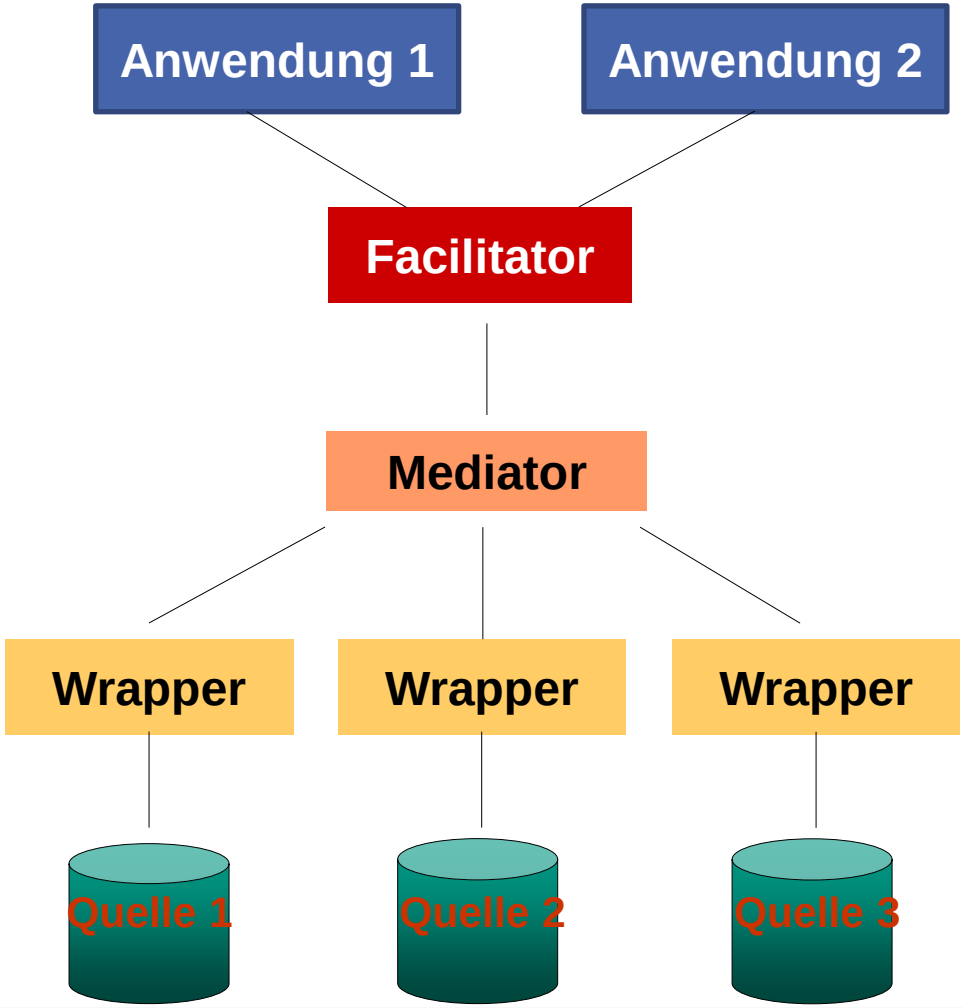


Verbraucher-portal

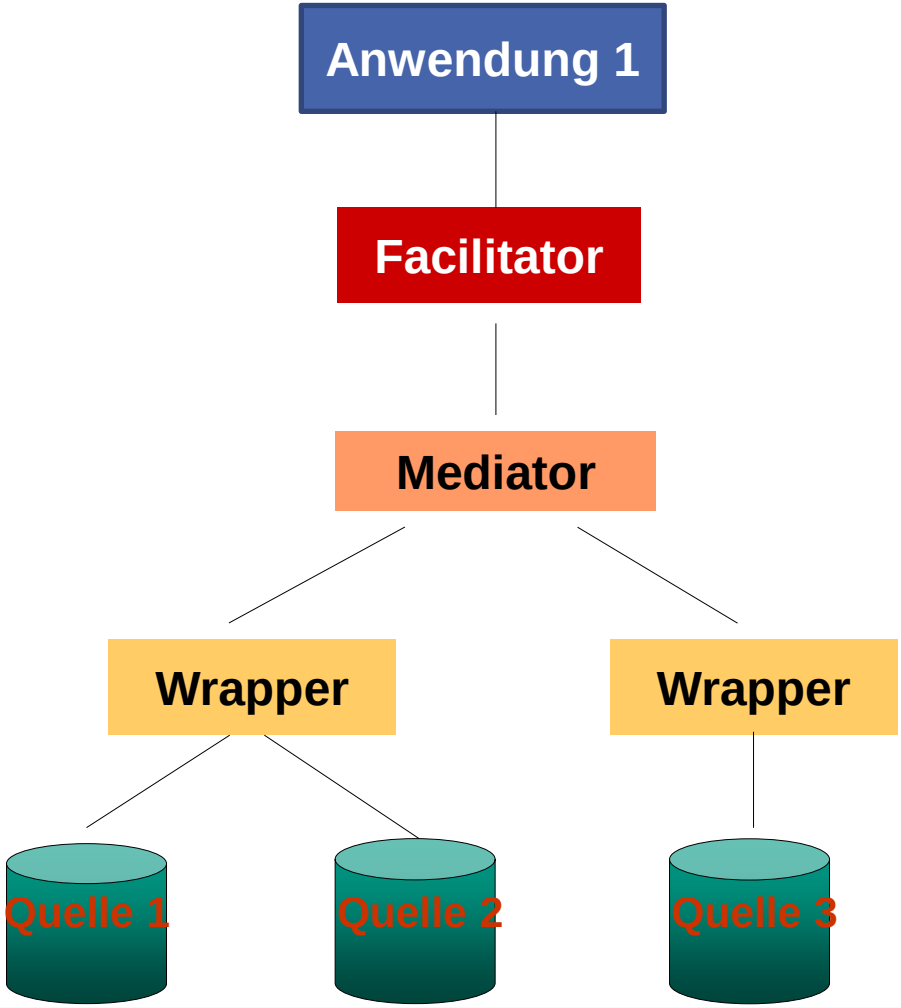


öffentliche Verwaltung

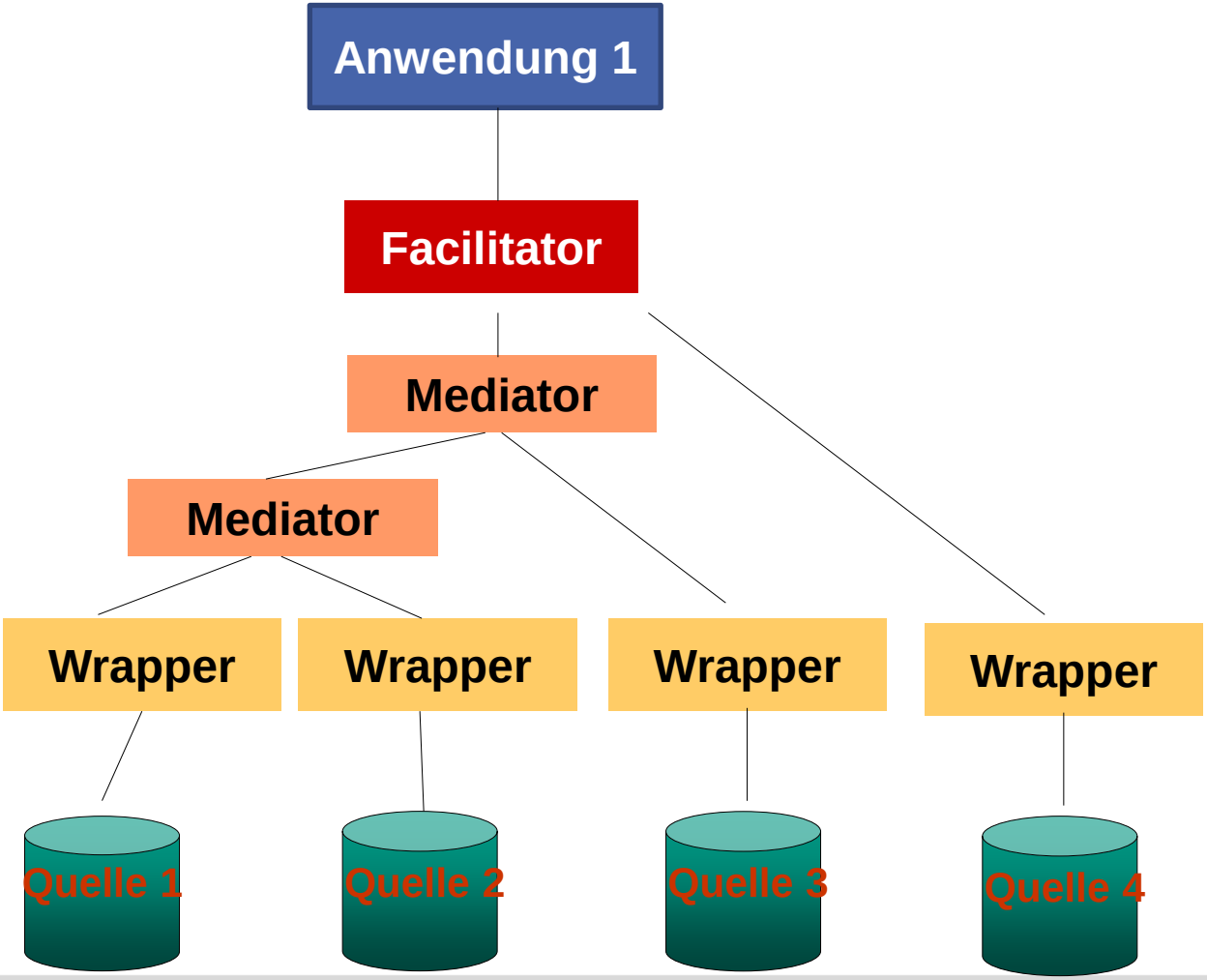
I³-Architekturen



I³-Architekturen



I³-Architekturen



Virtuelle Integration – Vorteile

- Aktualität
  - Daten sind in den Quellen gespeichert

  → Es wird immer auf den aktuellen Datenbestand zugegriffen.

  → Eine Aktualisierung ist nicht erforderlich.
- Ressourcenverbrauch
  - Durch die Nutzung der Fremdsysteme ist der Ressourcenverbrauch auf Portalseite niedriger.
  - Nur für Anfragebeantwortung notwendige Daten werden übertragen
- Leichte Erweiterbarkeit
- Autonomie
  - Nutzt man ausschließlich vorhandene Zugänge (z.B. WWW), so ist keinerlei Eingriff in die Datenquellen erforderlich.
  - Datenquellen wissen oft nicht von ihrer Integration

Virtuelle Integration – Nachteile

- Performanz
  - abhängig von den Fremdsystemen und der Netzwerkverbindung
  - keine gezielten Optimierungen möglich
- Mangelnde Kenntnisse über den Datenbestand
  - Komfortfunktionen sind erschwert, da keine Analyse des vollständigen Datenbestandes möglich ist
- Nachbereitungsoperationen
  - alle Transformationen der Daten müssen ausgeführt werden, wenn die Anfrage ausgewertet wird
- Änderungen an den Datenquellen führen zu Problemen
  - da keine wohldefinierte Schnittstelle

Problembereiche

Auf folgende Problembereiche soll im folgenden näher eingegangen werden:

- Facilitator
  - Quellenauswahl
- Mediator
  - einheitliches Informationsmodell
  - Anfragezerlegung, Anfrageübersetzung
  - semantische Integration (s. Vorlesung am 15.12.2014)
  - Objektverschmelzung
- Wrapper
  - Informationsextraktion

Quellenauswahl

- Das pauschale Durchreichen aller Anfragen an alle Quellen ist nicht besonders effizient.
  - Ziel: a priori Abschätzung der Relevanz einer Quelle für eine Anfrage
- Wie beschreibt man die Inhalte der Quelle?
  - gelieferte Entitäten und Attribute
  - bei Stichwortsuche: Index der Stichwörter
  - Beschreibung durch eine Anfragebedingung
    - Auswertung: Kann eine Benutzeranfrage überhaupt von einer Quelle beantwortet werden?
- Wie werden Inhaltscharakterisierungen gewonnen?
  - ohne weitergehenden Zugriff auf die Quellen schwierig
  - statistische Methoden

Einheitliches Informationsmodell
Globales Schema

- Den Nutzern bzw. der Portalanwendung soll eine einheitliche Sicht der Informationsobjekte präsentiert werden.

Vorgehensweisen:

- *1. Möglichkeit:* ***Globales Schema***
  Integration der Schemata vorhandener Quellen in ein globales Schema *(wie bei materialisiertem Ansatz)*
  - globales Schema ist als Sicht auf die Quellen definiert (global as view)
  - Ausgehend von Quellenschemata wird globales Schema gebildet
  - Vgl. Techniken zur Schemaintegration
    - One-shot, iterativ etc.
    - semi-automatische Unterstützung möglich
  - ändert sich mit der Integration neuer Quellen

Einheitliches Informationsmodell
Domänenmodell

- *2. Möglichkeit:* ***Domänenmodell***
  Modellierung der Anwendungsdomäne in einem Domänenmodell
  - Quellen werden als Sicht des Domänenmodells definiert (local as view)
  - Modellierungsaufgabe: zunächst Modellierung des Problembereichs, dann Einbinden der Quellen in dieses Modell
  - (im wesentlichen) quellenunabhängig
  - Dieselbe Aufgabe ist auch beim materialisierten Ansatz zu lösen
    - vergleichbar zu Datenaustauschformat

Anfragezerlegung

- Allgemeine Anfragen können aus Verknüpfungen (Joins) mehrerer Quelleninhalte bestehen
- Effiziente Zerlegung erforderlich

- Spezielle Klasse von Anfragen: Verschmelzungsanfragen *(fusion queries)*
  - nur ein Typ von Informationsobjekten, keine Verknüpfungen
  - Ziel: Zusammenführung der Informationen über ein Informationsobjekt
  - dann geht die identische Anfrage an alle (relevanten) Quellen

Anfrageübersetzung – Probleme

- Ziel: möglichst viel an die Quelle zur Ausführung übergeben *(query shipping)*

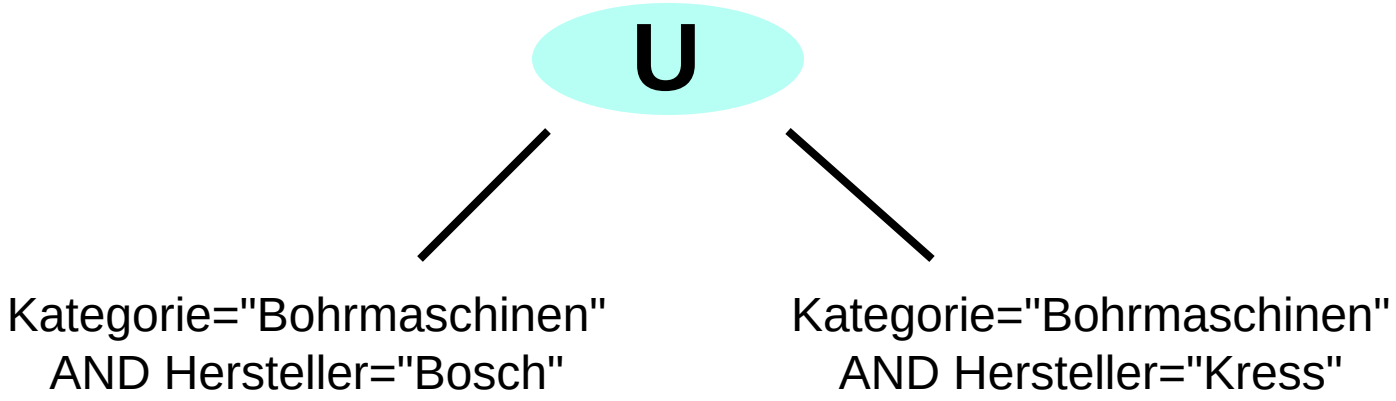
- Problem: Wie bildet man fehlende Mächtigkeit der Anfragesprache der Quellen nach? (**Anfragefähigkeitsanpassung**)
  - boolesche Operatoren (AND, OR, NOT)
  - Beschränkungen der Anfragestruktur
  - Beschränkungen der anfragbaren Attribute
  - Fehlende Anfrageprädikate (Phrasen, ...)

Anfrageübersetzung – Subsumierende Anfragen und Filter

- Lösung: Konzept der »subsumierenden Anfragen«
  - Finde eine Anfrage, die von der Quelle unterstützt wird und die eine minimale Obermenge zum gewünschten Ergebnis liefert
  - Anschließend werden Filteroperationen angewendet.
  - Nicht immer möglich

Anfrageübersetzung – Bsp.: Fehlender boolescher Operator

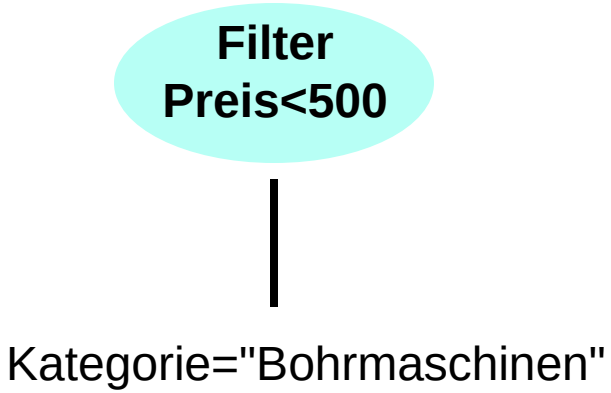
- *Kategorie="Bohrmaschinen" AND (Hersteller="Bosch" **OR** Hersteller="Kress")*



- ersetze durch äquivalenten Mengenoperator
- vorher evtl. Transformation in DNF/KNF o.ä. erforderlich

Anfrageübersetzung – Beispiel: Nicht suchbares Attribut

- *Kategorie="Bohrmaschinen"* ***AND Preis<500***



- konjunktiv verknüpfte Anfrageprädikate können als Filter nachgeschaltet werden.

Anfrageübersetzung – Unvollständige Information

- Was ist, wenn eine Quelle ein Attribut in der Anfrage-bedingung nicht besitzt?
- Zwei Lösungsstrategien
  - **Vollständigkeit**: ignoriere die entsprechenden Anfrageteile
  - **Qualität**: werte die entsprechenden Anfrageteile zu falsch aus

- Beispiel: Gesucht sind Bohrmaschinen mit einer Garantiezeit von mindestens 3 Jahren
  - Vollständigkeit: Wenn Garantiezeit nicht verfügbar, dann trotzdem zurückgeben.
  - Qualität: Auf jeden Fall keine Bohrmaschine mit einer Garantiezeit unter 3 Jahren zurückliefern.

Anfrageübersetzung – Lokalisierung

- Wo findet die Anfrageübersetzung statt?
  - hängt von der Schnittstellendefinition des Wrappers ab
- in den Wrappern
  - alle Wrapper haben eine einheitliche Schnittstelle im Hinblick auf die Anfragesprache
  - quellenspezifische Anfrageübersetzungstechniken leicht möglich
- in den Mediatoren
  - Wrapper beschreiben ihre Anfragefähigkeiten
  - Generische Anfragebearbeitungsfunktionalität
  - Wrapperübergreifende Optimierung (Alternativquellen o.ä.)
  - reduziert die Wrapper-Komplexität
- Fazit: meistens Aufteilung der Aufgabe zwischen Wrappern und Mediatoren

Informationsextraktion

- Einfacher Fall:
  - Quelle liefert Information bereits in sehr strukturierter Form (z.B. relationale DBMS)
  - nur Transformation ins gemeinsame Datenmodell erforderlich

- Schwieriger:
  - Informationselemente müssen aus unstrukturierter Information extrahiert werden (z.B. HTML-Seiten) und sind über unterschiedliche Ergebnisseiten hinweg fragmentiert
  - z.B. Ausnutzen der HTML-Struktur
    - html.body.table[2].td
  - reguläre Ausdrücke
    - Preis: (\d)+ €
  - Nutzung von Hilfswerkzeuge zur Generierung

Integration

- Transformation ins Domänenmodell
  - Strukturtransformationen
  - Werttransformationen
  - berechnete Attribute
  - Aggregation

Integration auf Schemaebene

- Objektverschmelzung
  - über semantische Schlüssel
    - evtl. Normalisierung erforderlich
  - paarweiser Ähnlichkeitsvergleich
  - Noch viele offene Probleme (insbesondere Objektidentität)

Integration auf Instanzebene

- Mehr Details später

Virtuelle Integration: Nachbemerkung

- Die hier vorgestellte Aufteilung der Funktionalität auf unterschiedliche Schichten/Komponenten ist idealisiert.
- In realen System ist die Aufgabenteilung zwischen Wrappern und Mediatoren uneinheitlich
  - **Fette Wrapper.** Hier werden z.B. große Teile der Anfrageübersetzung vom Wrapper übernommen.
  - **Dünne Wrapper.** Hier ist der Wrapper nur für die Informationsextraktion und die Transformation der Daten in das gemeinsame Datenmodell zuständig.
- Auch die Aufgaben der Facilitator-Schicht werden oft entweder in die Mediatoren oder die Anwendung direkt verlagert.

VIRTUELL VS. MATERIALISIERT

Virtuell vs. Materialisiert – Vergleich

- Aktualität

Virtuell	Materialisiert
• Sehr gut • Manchmal: Caching	• Je nach Update-Frequenz • Meist täglich (z.B. über Nacht)

- Antwortzeit

Virtuell	Materialisiert

• Nicht gut
• Daten sind entfernt, d.h. Übertragung durch das Netz
• Abhängig von Antwortzeit der Quellen
• Optimierung schwierig
• Komplexe Operatoren müssen naiv ausgeführt werden
• Data Cleansing Operationen müssen nachgeholt werden.
	• Sehr gut
	• Lokale Bearbeitung
	• Wie DBMS
	  • Optimierung
	  • Materialisierte Sichten
	  • Indices etc.
	• Allerdings: typische Anfragen sind komplex

Quelle: Felix Naumann, Informationsintegration: Materialisierte vs.Virtuelle Integration, Hasso Plattner Institut, 2008

Virtuell vs. Materialisiert – Vergleich

- Flexibilität/Wartbarkeit

Virtuell	Materialisiert
• Einfacher • Entfernen/Ändern/Hinzufügen einer Quelle wirkt sich nur auf das Mapping dieser Quelle aus • Quellen müssen Daten selbst warten	• Schwierig • Entfernen/Ändern/Hinzufügen einer Quelle kann gesamte Integration verändern • Lokale Wartung eines großen und wachsenden Datenbestandes (Indices etc.) • Tägliche Integration nötig

Quelle: Felix Naumann, Informationsintegration: Materialisierte vs.Virtuelle Integration, Hasso Plattner Institut, 2008

Virtuell vs. Materialisiert – Vergleich

- Komplexität

Virtuell	Materialisiert

• Modellierung der Quellen wichtig
  • Fähigkeiten der Quellen
• Oft verschiedenste Quellen
  • Web Services
  • HTML Formulare
  • Bilder etc.
	• Wie DBMS
	• Komplexe Anfragen möglich
	• Quellen sind oft untereinander ähnlich (oft selbst DBMS).

Quelle: Felix Naumann, Informationsintegration: Materialisierte vs.Virtuelle Integration, Hasso Plattner Institut, 2008

Virtuell vs. Materialisiert – Vergleich

- Autonomie

Virtuell	Materialisiert

• Quellen können autonom sein
• Volle Design-Autonomie
• Fast volle Kommunikations-Autonomie
  • Gewisse Kommunikation ist nötig, sonst nicht Teilnehmer der Integration
• Fast volle Ausführungs-Autonomie
  • Nur: Anfragen müssen irgendwann beantwortet werden
	• Quellen wenig autonom
	  • Geringe Design-Autonomie
	  • Keine Kommunikationsautonomie
	  • Geringe Ausführungsautonomie

Quelle: Felix Naumann, Informationsintegration: Materialisierte vs.Virtuelle Integration, Hasso Plattner Institut, 2008

Virtuell vs. Materialisiert – Vergleich

- Anfragebearbeitung/Mächtigkeit

Virtuell	Materialisiert

• Anfragebearbeitung komplex
  • Verteilung
  • Autonomie
  • Heterogenität
• Mangelnde Fähigkeiten der Quellen können global eventuell ausgeglichen werden
• Aber auch: Spezialfähigkeiten der Quellen können genutzt werden:
  • Image Retrieval etc.
	• Anfragebearbeitung wie DBMS
	• Anfragemächtigkeit wie globales System, z.B. volle SQL Mächtigkeit

Quelle: Felix Naumann, Informationsintegration: Materialisierte vs.Virtuelle Integration, Hasso Plattner Institut, 2008

Virtuell vs. Materialisiert – Vergleich

- Größe/Speicherbedarf

Virtuell	Materialisiert

• Gering
  • Metadaten
  • Cache
  • Zwischenergebnisse
	• Hoch
	  • Redundante Datenhaltung
	  • Historische Daten bei Data Warehouse
	• Wachstum
	  • Stetig wachsend
	  • Ggf. konstant durch zunehmende Aggregation

- Ressourcenbedarf

Virtuell	Materialisiert

• Potentiell hohe Netzwerklast
• Daten werden mehrfach übertragen
  • Ggf. Cache
• Nur jeweils nötige Daten werden übertragen
	• Planbare Netzwerklast
	• Evt. unnötig übertragene Daten
	• übertragen
	  • Abhängig von Anfrage
	  • Aggregation
	  • Pre-Aggregation

Quelle: Felix Naumann, Informationsintegration: Materialisierte vs.Virtuelle Integration, Hasso Plattner Institut, 2008

Virtuell vs. Materialisiert – Vergleich

- Vollständigkeit

Virtuell	Materialisiert
• Nur bei Verfügbarkeit aller nötigen Quellen • Ggf. Anfrage unbeantwortbar oder nur unvollständig beantwortbar	• Gut • Annahme: Materialisierung ist vollständig

- Data Cleansing/Informationsqualität

Virtuell	Materialisiert

• Online cleansing schwierig
  • Aufwändig
  • Keine Interaktion mit Experten möglich
• Informationsqualität abhängig von Quellen
  • z.T. zweifelhaft
	• Viele Cleansing-Methoden
	  • Offline möglich (über Nacht)
	• Hohe Informationsqualität
	  • Kontrolliert
	  • Kann bei Bedarf verbessert werden

Quelle: Felix Naumann, Informationsintegration: Materialisierte vs.Virtuelle Integration, Hasso Plattner Institut, 2008

Virtuell vs. Materialisiert – Fazit

- Virtueller Ansatz ist zu bevorzugen bei Datenquellen
  - mit hoher Änderungsrate
  - mit großem Datenvolumen
  - mit geringer Anfragehäufigkeit/geringem Anfragevolumen
- Um den Implementierungsaufwand für ein virtuelles Integrationssystem in Grenzen zu halten:
  - Wie heterogen sind die anzubindenden Systeme?
  - Welche Anfragefunktionalität wird benötigt? (Joins?, boolesche Verknüpfungen, ...)
- Oft bietet sich auch ein hybrider Ansatz an
  - kleinere, weniger mächtige Quellen werden repliziert
  - die restlichen werden virtuell angebunden

Virtuell vs. Materialisiert – Fazit (2)

- Was die Erfahrung zeigt:
  - virtuelle Integrationsansätze ohne wohldefinierte Schnittstellen an der Grenze des eigenen Einflussbereiches taugen nur für Ad-hoc-Integration oder "feindliche Nutzung"
  - Ist eine Kooperation mit dem Anbieter möglich, so erhöht dies deutlich die Stabilität
  - Zur Erhöhung der Dienstgüte setzt sich bei überschaubaren Datenvolumina der materialisierte Ansatz durch
    - Preissuchmaschine.de: pull-Ansatz
    - froogle.de: push-Ansatz
- Fazit
  - materialisierten Ansatz sollte man als Ausgangspunkt sehen
  - wenn das nicht geht (organisatorisch/technisch), dann virtuell

Literatur

- Arens Y., Hull R., King R. (Hrsg.): Reference Architecture for the Intelligent Integration of Information, Program on Intelligent Integration of Information, ARPA, Draft Version 2.0, 22. August 1995
- Leser U., Naumann F. (2007): Informationsintegration: Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen; dpunkt Verlag 2007
- Tamer Özsu M., Valduriez P.: Principles of Distributed Database Systems, Prentice Hall, (1991/)1999
- Wiederhold G.: Mediators in the architecture of future information systems, *IEEE Computer Magazine* **25** (3): 38–49, 1992

SONSTIGES

Studenten-Nachmittag

am FZI Forschungszentrum Informatik
im FZI House of Living Labs

11. Dezember 2014 ab 12.00 Uhr

Komm vorbei und erfahre bei einem Getränk mehr über deine Möglichkeiten am FZI!

Dein Weg zu Führungsaufgaben
beginnt hier: www.fzi.de/karriere

CAS IT Lounge

04.12.2014 ab 18 Uhr

„Event-driven Architecture and Semantic Web“

Artur Felic

CAS Software AG, CAS-Web 1-5