Auffinden von und Zugriff auf Datenquellen

Seminar Informationsintegration und Informationsqualität

Dragan Sunjka

TU Kaiserslautern

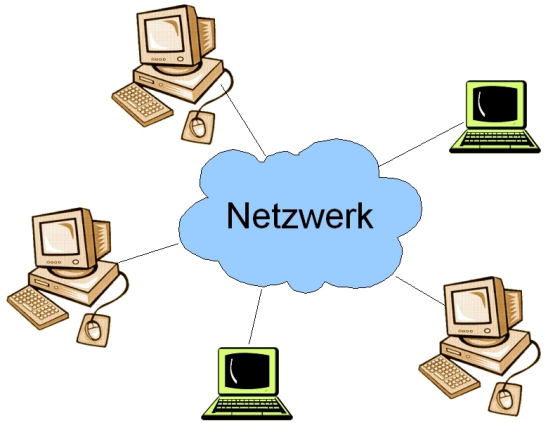
30. Juni 2006

Gliederung

Autonomie

Verteilung führt zu Autonomie...

- Intra-Organisation: historisch
- Inter-Organisation: Internet



Autonomie

- Grad zu dem verschiedene DBMS unabhängig operieren

Autonomieklassen

- Entwurfsautonomie
- Kommunikationsautonomie
- Ausführungsautonomie

Autonomieklassen

Entwurfsautonomie

- Datenmodell, Schema

Kommunikationsautonomie:

- Wahl mit *welchen* Systemen *wann* *was* kommuniziert wird, Anfragesprache

Ausführungsautonomie

- Wahl *wann* und *wie* Anfragen ausgeführt werden
- Wahl der Scheduling- und Optimierungs-Strategie

Autonomie führt zu Heterogenität

Autonomie als Ursache für Heterogenität:

Autonome Systeme

- Gestaltungsfreiheit

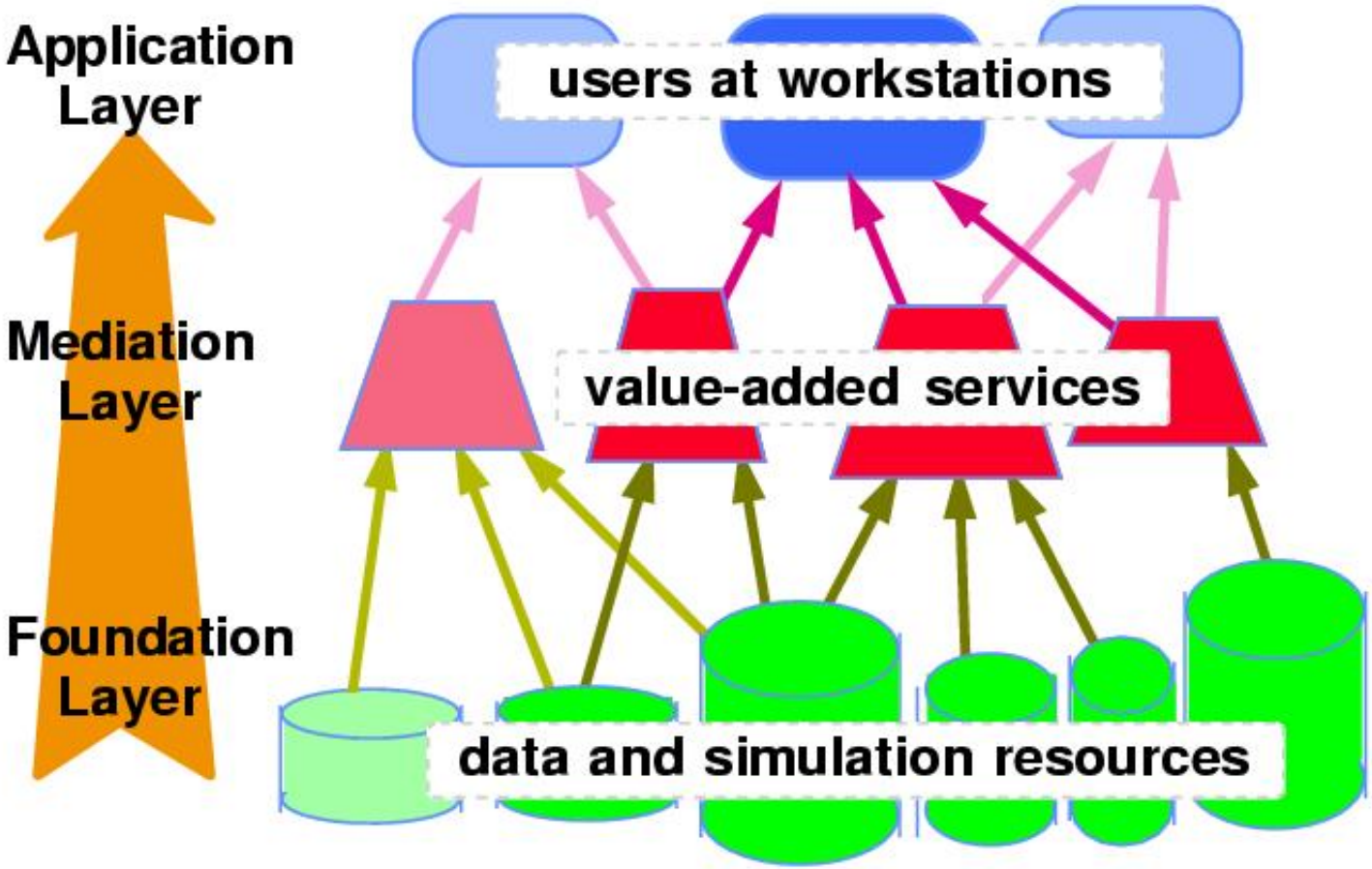
  $\Rightarrow$ unterschiedliche Entscheidungen

  $\Rightarrow$ Heterogenität

  - technisch, logisch, semantisch

Gliederung

Mediation



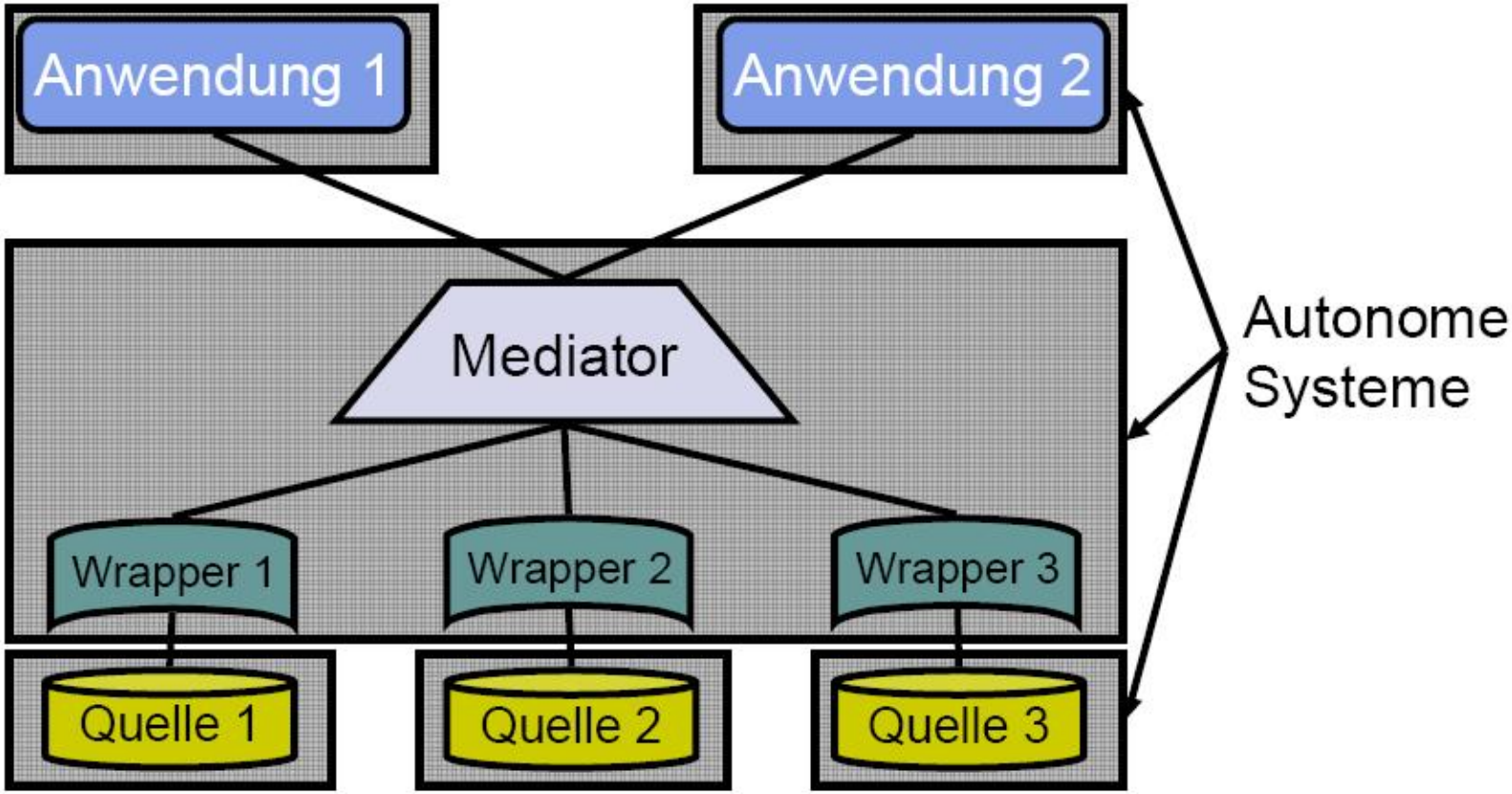
Mediation (2)

Funktionen der Mediation

- Suche und Auswahl von relevanten Datenquellen
- Transformation der Daten anhand von Metadaten
- Integration der transformierten Daten
- Zusammenfassung zur Präsentation

$\Rightarrow$ Transformation von Daten zu Informationen

Wrapper



- Vermittlung zwischen Mediator und Datenquelle
- jeweils spezialisiert auf eine Ausprägung autonomer, heterogener Datenquellen

Wrapper (2)

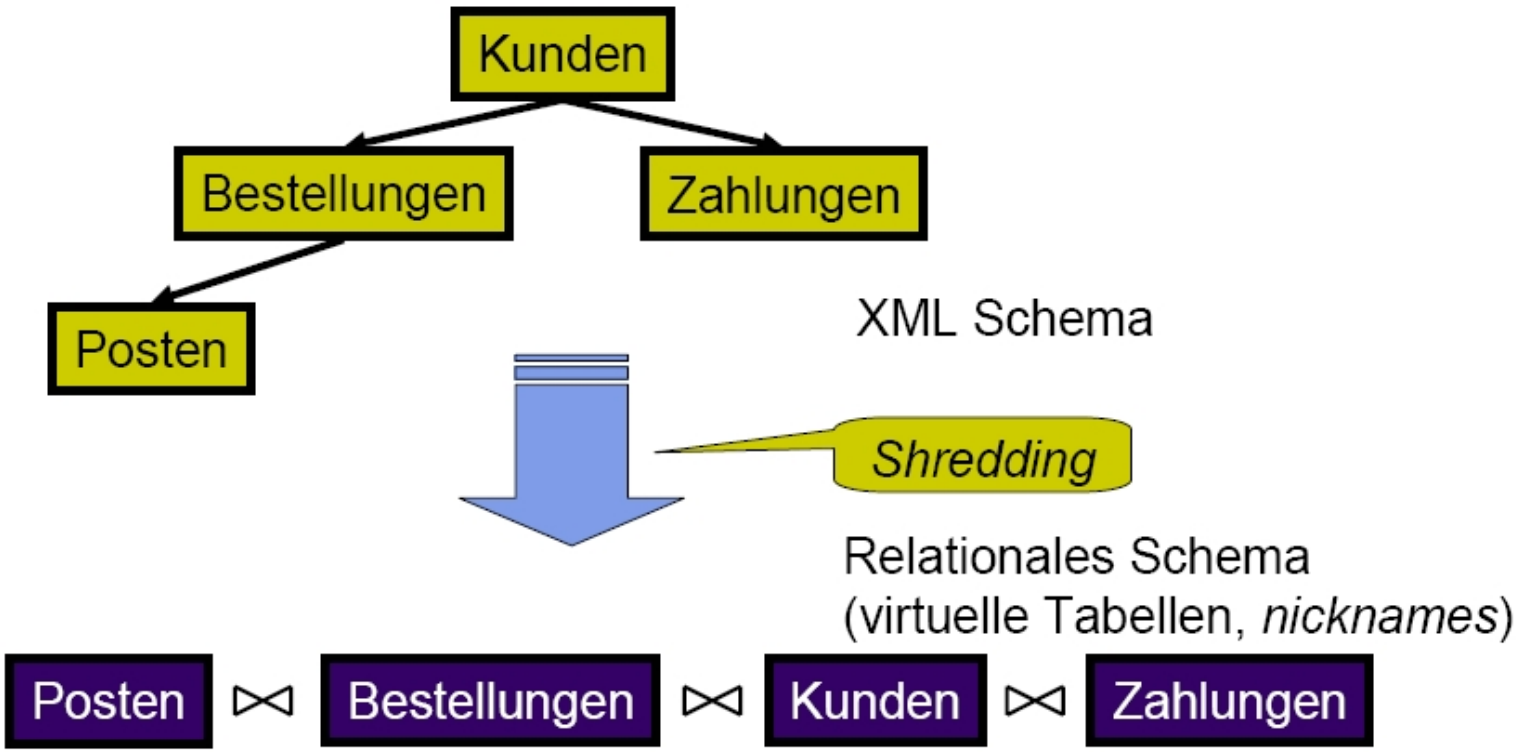
Vorteile des Wrappings

- überwinden Heterogenitäten
- Wiederverwendbarkeit
- Unabhängigkeit der Datenquellen

Nachteile des Wrappings

- i.A. schlechtere Leistung
- Aktualität der Wrapper notwendig

XML Wrapper in IBM DB2 II



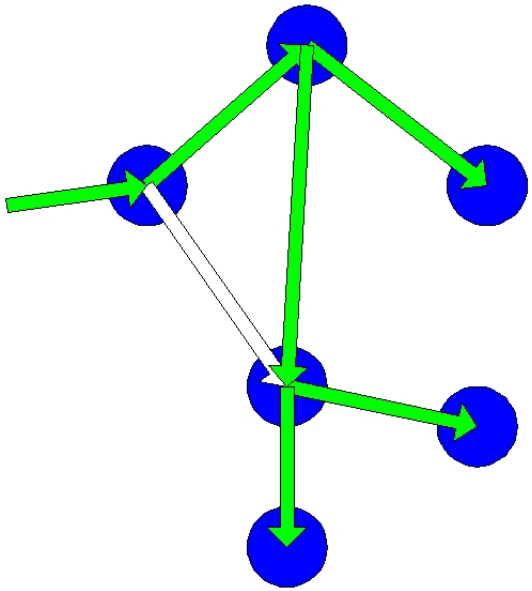
XML Wrapper in IBM DB2 II (2)

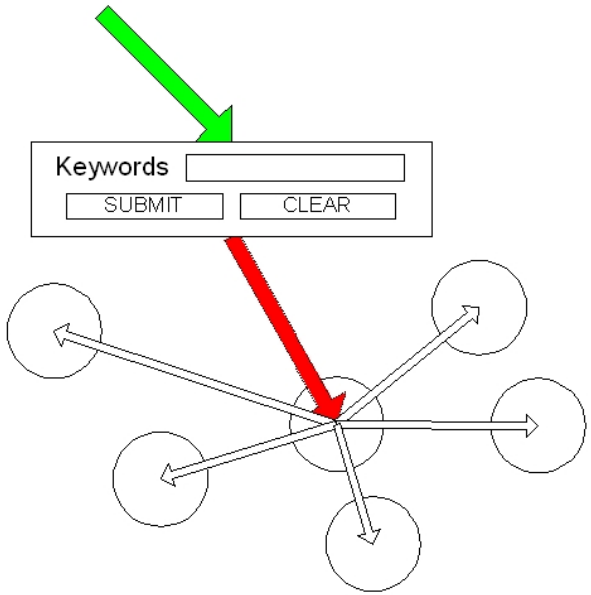
```
CREATE NICKNAME kunden_NN(
name            VARCHAR(48) OPTIONS(XPATH './name/text()'),
adresse         VARCHAR(48) OPTIONS(XPATH './address/text()'),
kunden_NN_ID VARCHAR(48) OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER xml_server
  OPTIONS(XPATH '//customer', FILE_PATH 'customers.xml');
```

```
CREATE NICKNAME order_NN(
amount          DOUBLE      OPTIONS(XPATH './amount/text()'),
date            VARCHAR(48) OPTIONS(XPATH './date/text()'),
order_NN_ID     VARCHAR(48) OPTIONS(PRIMARY_KEY 'YES'),
customer_NN_FID VARCHAR(48) OPTIONS(FOREIGN_KEY 'CUSTOMER_NN'))
FOR SERVER xml_server OPTIONS(XPATH './/order');
```

Gliederung

Surface Web vs. Hidden Web





Surface Web

- Linkstruktur
- zum *Crawlen* geeignet

Hidden Web

- keine Linkstruktur
- Dokumente versteckt in DBMS

Hidden Web

Statistiken zum Hidden Web

- 550 mal größer als das Surface Web
- 7500 Terabyte im Hidden Web
- ca. 84% sind auf Textdokumente spezialisiert
- ca. 95% des Hidden Web ist öffentlich verfügbar
- am schnellsten wachsende Kategorie neuer Informationen im Internet

Herausforderungen

- Auffinden von relevanten Hidden-Web-Quellen
  $\rightarrow$ Klassifikation
- Zugriff auf Hidden-Web-Quellen
  $\rightarrow$ Anfragesprache lernen

Überblick

Klassifikation: Zuordnung zu Kategorien in einer Hierarchie

- Manuell
  - Yahoo!, InvisibleWeb, SearchEngineGuide
- Automatisch

Zwei Arten von Klassifikation

Coverage (Abdeckung)-basierte Klassifikation

- #docs über das Thema

Specifity (Spezifizität)-basierte Klassifikation

- #docs/$|DB|$

Query Probing

Classifier Learning

- Input: Menge von bereits klassifizierten Dokumenten
- Output: Menge von Klassifikationsregeln
  - IF **linux** THEN **Computers**
  - IF **ibm** AND **intel** THEN **Computers**
  - IF **jordan** AND **bulls** THEN **Sports**
  - IF **diabetes** THEN **Health**

Query Probing

Classifier Learning

- Input: Menge von bereits klassifizierten Dokumenten
- Output: Menge von Klassifikationsregeln
  - IF **linux** THEN **Computers** $\rightarrow$ *+linux*
  - IF **ibm** AND **intel** THEN **Computers** $\rightarrow$ *+ibm +intel*
  - IF **jordan** AND **bulls** THEN **Sports** $\rightarrow$ *+jordan +bulls*
  - IF **diabetes** THEN **Health** $\rightarrow$ *+diabetes*

Query Probing

Classifier Learning

- Input: Menge von bereits klassifizierten Dokumenten
- Output: Menge von Klassifikationsregeln
  - IF **linux** THEN **Computers** $\rightarrow$ *+linux*
  - IF **ibm** AND **intel** THEN **Computers** $\rightarrow$ *+ibm +intel*
  - IF **jordan** AND **bulls** THEN **Sports** $\rightarrow$ *+jordan +bulls*
  - IF **diabetes** THEN **Health** $\rightarrow$ *+diabetes*

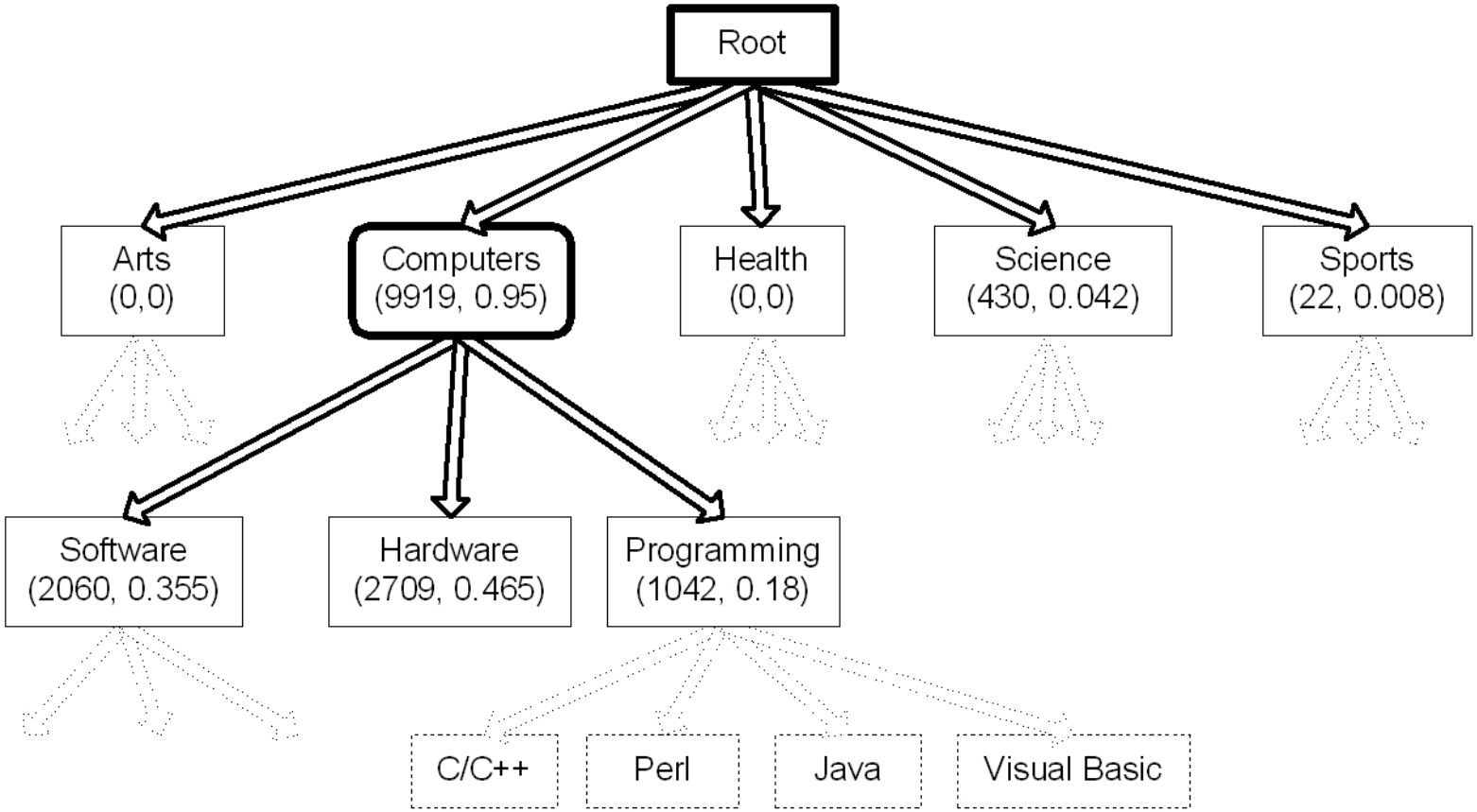
Query Probing

- Schicke Anfragen an die Datenbank
- Hole (parse) Anzahl der Ergebnisse

Klassifikation einer Datenbank

1. Schicke Anfragen für Top-Level-Kategorien
2. Hole **Anzahl der Treffer für jede Anfrage**
3. Berechne geschätzte Spezifizität und Abdeckung für jede Kategorie
4. Bewege die Datenbank in die sich qualifizierenden Kategorien (mit Spezifizität $\geq$ Ts, Abdeckung $\geq$ Tc)
5. Wiederhole für alle sich qualifizierenden Subkategorien
6. Gib alle Kategorien zurück, die sich qualfiziert haben

Beispiel: ACM Digital Library (Tc=100, Ts=0.5)



Gliederung

Begriffsklärung: Grid

Grid-Paradigma: Virtualisierung von Ressourcen

- Ursprüngliche Idee: Rechenleistung aus der Steckdose
- Ziel: High Performance Super-Computing
- CPU-Rechenleisung und Arbeitsspeicher als einzige Ressourcen im Grid

Data Grids

- anfangs: Datentransfer nur dateibasiert, Input/Output-Dateien
- später: Speicherplatz als Ressource
- heute: (semi-)strukturierte Daten als Ressourcen

OGSA-DAI

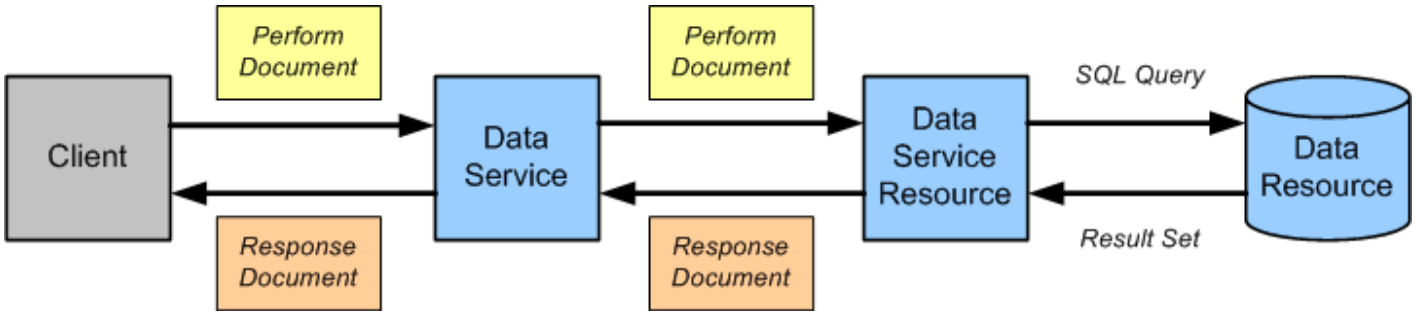
Open Grid Services Architecture - Data Access and Integration

- ein in Java entwickeltes Framework für den Zugriff auf Datenbanken
- Ziel: einheitliche Schnittstelle zum Zugriff auf heterogene Datenbanken
- versucht nicht, die Kluft zwischen SQL und XML zu überbrücken
  $\Rightarrow$ Überwindung technischer Heterogenität

Anfragen an Quellen

- erfolgen mittels XML-basierten Perform-Dokumenten
  - Basisaktivitäten (Anfragen, Transformationen, Übertragung)

Interaktion von Komponenten



Herausforderungen

- Common Data Model?
- SQL vs. XML
- XML WebRowSet ist flexibel aber ineffizient

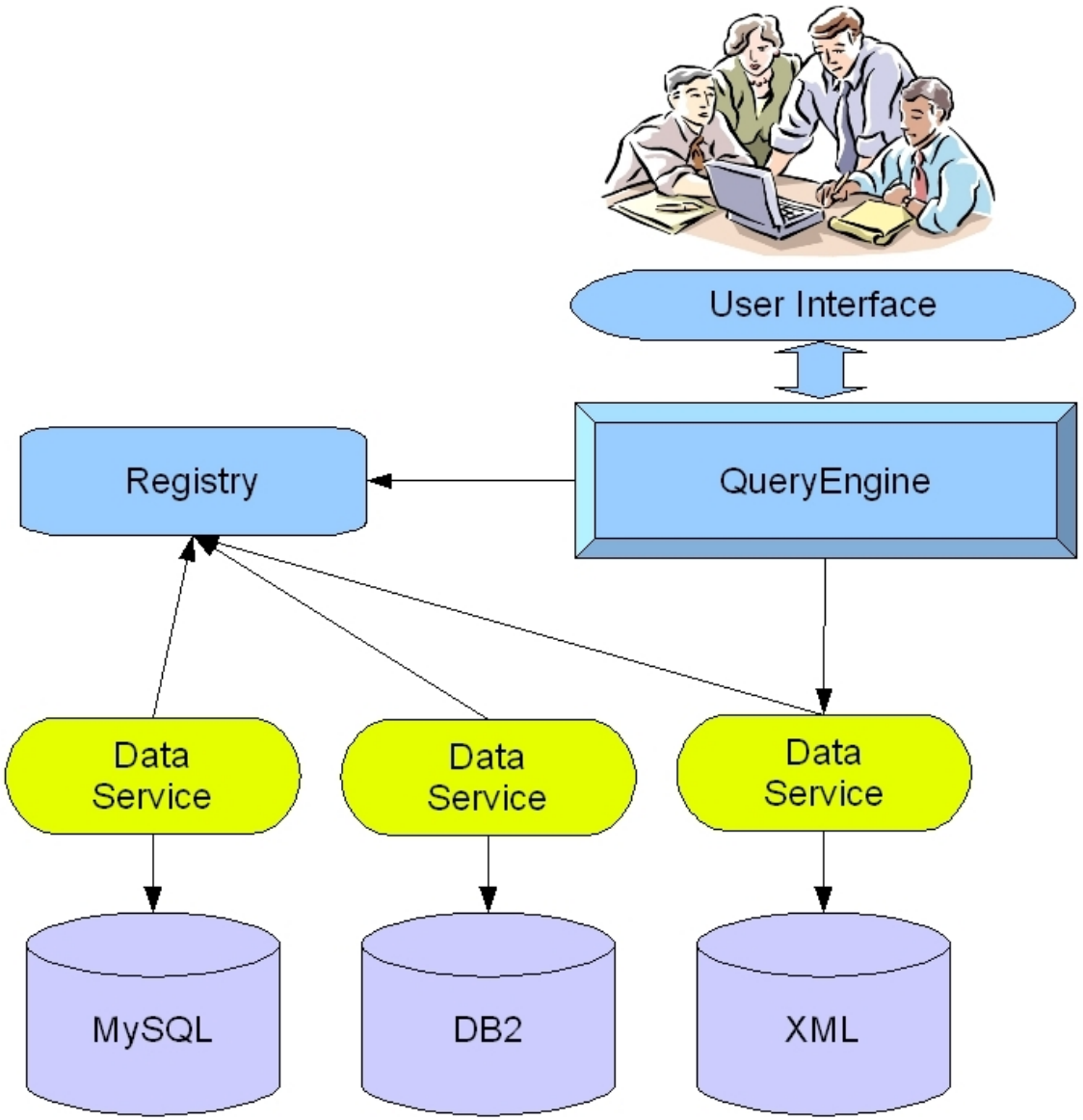
Dynamische Integration in Grid-Umgebungen

Dynamische Integration heterogener, autonomer, verteilter Datenquellen

Merkmale

- Zum Zeitpunkt der Anfrage gibt es kein globales Schema
- automatische Integration von neuen Quellen
- OGSA-DAI als Infrastruktur

DynaGrid Architektur



Zusammenfassung

- viele Varianten klassischer Mediator/Wrapper-Architektur
  - Wrapping zur Überbrückung der Heterogenität
- Hidden Web wächst schnell
  - Wie findet man solche Eintrittspunkte?
  - Semantic Web?
- zahlreiche autonome Datenquellen und schnell wechselnde Benutzeranforderungen
  - dynamische Discovery und flexible Schema-Matching-Verfahren

Vielen Dank für die Aufmerksamkeit

Fragen?