

# Einführung Informationsintegration

## Komplexe Informationssysteme

---

Fabian Panse

panse@informatik.uni-hamburg.de

Universität Hamburg

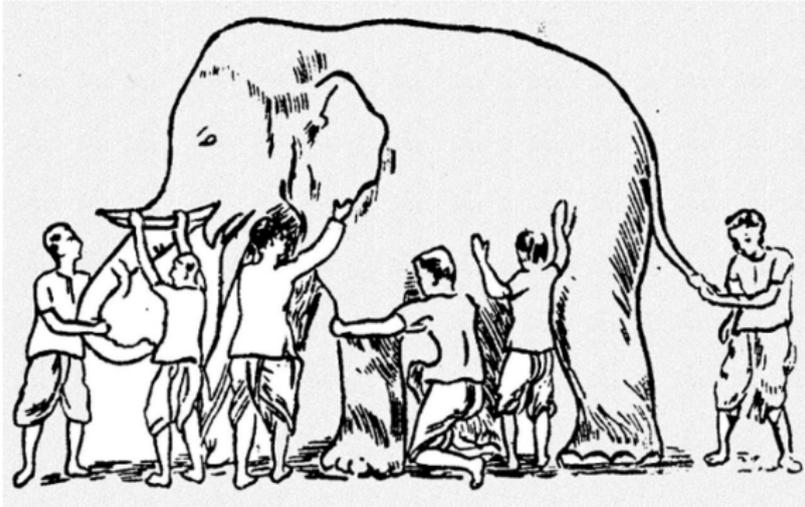


Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG



# Der Elefant und die sechs blinden Männer



Quelle: Laura M. Haas. Beauty and the beast: The theory and practice of information integration. ICDT, 2007.

# Informationsbedarf und potentielle Einsparungen

---

## 1 von 3

Manager treffen häufiger Entscheidungen auf Basis von Informationen, denen sie nicht vertrauen oder die sie gar nicht haben.

## 5700 USD/a

Zeitaufwand je Wissensarbeiter für Umformatierung von Informationen zwischen Anwendungen.

## 1 von 3

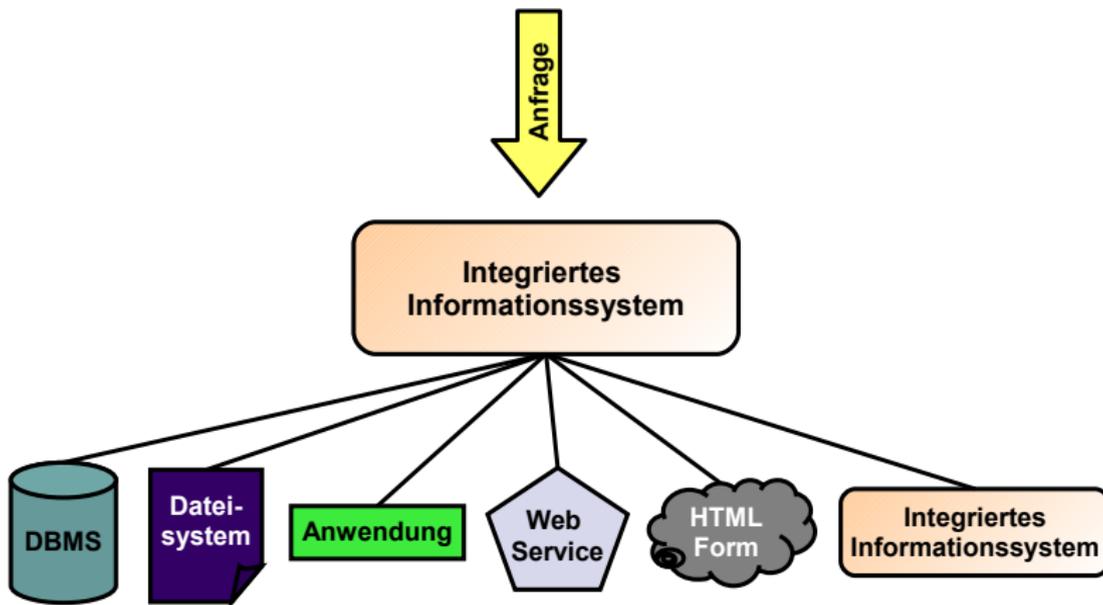
Manager haben nicht Zugriff zu den erforderlichen Informationen.

## 5300 USD/a

Zeitaufwand je Wissensarbeiter für Informationssuche.

Quelle: IBM: Break Away with Business Analytics and Optimization Study, IDC

# Integrierte Informationssysteme



# Organisatorisches

- **Voraussetzungen:**
  - Grundlagen Datenbanken (relationale DBMS, SQL, XML)
  - Interesse an aktuellem Thema
- **Prüfungsinhalt:** ausschließlich Vorlesungsstoff
- **Übungen:** Saalübungen
  
- **Acknowledgements:** Angelehnt an Folien von
  - Dr. Armin Roth (IBM)
  - Prof. Dr. Melanie Herschel (Univ. Stuttgart)
  - Folien zum Buch *Principles of Data Integration*



# Organisatorisches

## 1. Woche (04.09.2017 - 09.09.2017)

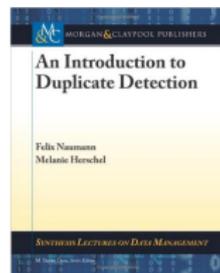
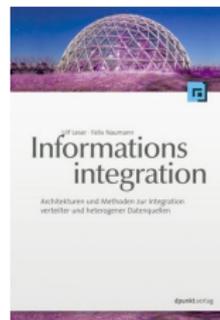
Montag	Dienstag	Mittwoch	Donnerstag	Freitag
Vorlesung	Vorlesung	Vorlesung	Vorlesung	-
9:30	9:30	9:30	9:30	-
13:30	13:30	13:30	13:30	-

## 2. Woche (11.09.2017 - 15.09.2017)

Montag	Dienstag	Mittwoch	Donnerstag	Freitag
Vorlesung	Vorlesung	Vorlesung	Seminar	Seminar
9:30	9:30	9:30	9:30	9:30
13:30	13:30	13:30	16:30	16:30

# Literatur

- Ulf Leser und Felix Naumann. **Informationsintegration**. dpunkt.verlag, 2006 [LN06]
- Anhai Doan, Alon Halevy, Zachary Ives. **Principles of Data Integration**. Morgan Kaufmann, 2012 [DHI12]
- Felix Naumann und Melanie Herschel. **Introduction to Duplicate Detection**. Morgan & Claypool, 2010 [NH10]
- Peter Christen. **Data Matching**. Springer, 2012 [Chr12]



# Was ist Informationsintegration? [LN06]

---

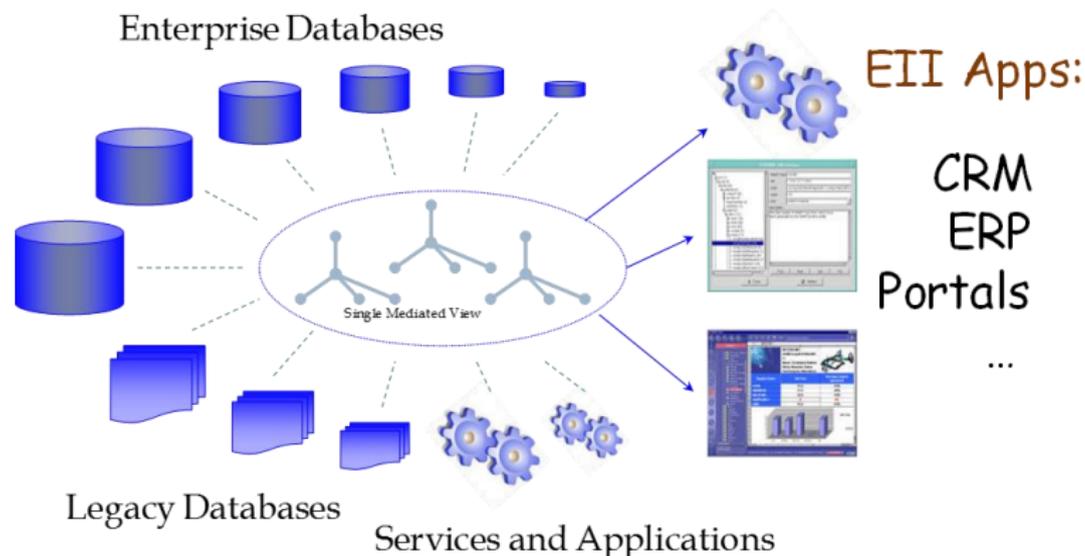
Informationsintegration ist die **korrekte, vollständige** und **effiziente** Zusammenführung von Informationen verschiedener, **heterogener** Quellen zu einer einheitlichen und **strukturierten** Informationsmenge zur **effektiven Interpretation** durch Nutzer und Anwendungen.

# Wofür brauchen wir Informationsintegration? [DHI12]

---

- Informationssysteme in vielen Lebensbereichen.
- In der Realität sind Informationssysteme häufig unabhängig voneinander konzipiert nur um später festzustellen, dass die von ihnen bereitgestellten Informationen kombiniert betrachtet werden müssen.
- Zu diesem Zeitpunkt benutzen die Systeme unterschiedliche Datenmodelle, unterschiedliche Schemata und bieten oft nur einen eingeschränkten Zugriff auf ihre Daten.
- Das Ziel der Informationsintegration ist es verschiedene Informationsquellen unter einer Sicht zu vereinen.

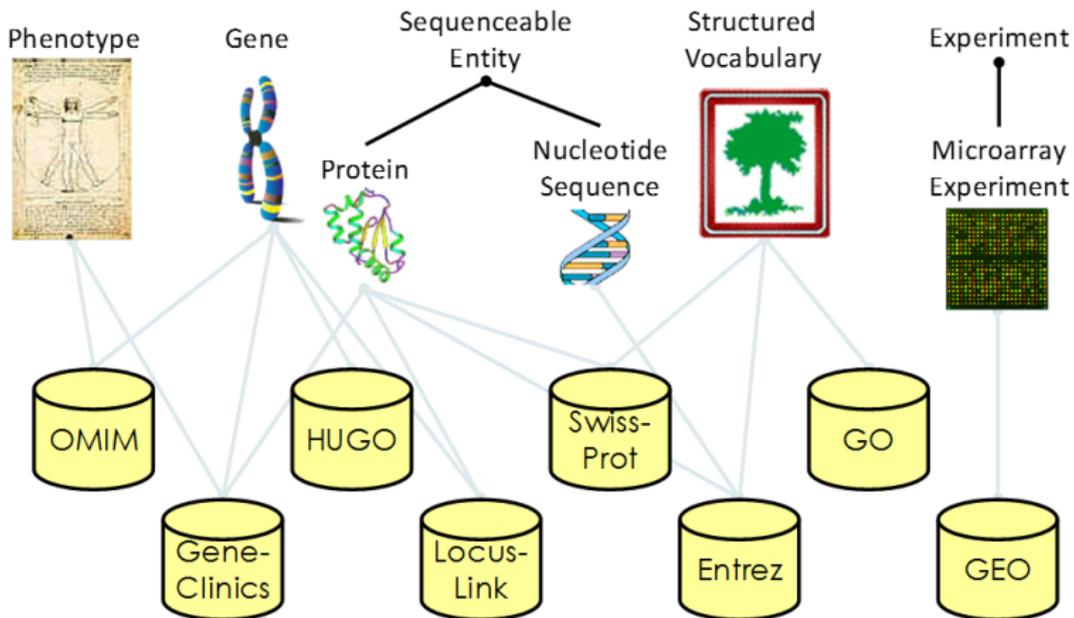
# Anwendungen: Business



**50% of all IT \$\$\$ spent here!**

Quelle: Doan, Halevy and Ives. Principles of data Integration (Slides), 2012 [DHI12]

# Anwendungen: Biowissenschaften



**Hundreds of biomedical data sources available; growing rapidly!**

Quelle: Doan, Halevy and Ives. Principles of data Integration (Slides), 2012 [DHI12]

# Anwendungen: Web Data Integration



ybw.com
   
 books & charts

Quilt-Books.Com



COMPU-BOOKS

KATSUKI BOOKS



Providence-Books  
www.providence-books.com

NGHOSI books



КНИГАРНЯ  
bookshop online



BORDERS.com  
million books, CDs, and videos

CACTUS-BOOKS.COM

metro-Books

BARNES & NOBLE  
www.bn.com

StoneCreekBooks.com

ASTROLOGY-BOOKS.COM

SeeMe4Books

Professional Books Inc.

amazon.com.

HALF PRICE COMPUTER BOOKS  
DANCE BOOKS

acma BOOKS.COM

NORTH 49 BOOKS



Just Great Books

Quelle: Doan, Halevy and Ives. Principles of data Integration (Slides), 2012 [DHI12]

# Anwendungen: Historische Daten

As a thank-you bonus, site members have access to a banner-ad-free version of the site, with print-friendly pages.  
(Already a member? [Click here.](#))

EnchantedLearning.com  
**US History**  
US Geography

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
African-American Artists Explorers of the US Inventors US Presidents US Symbols US States

EnchantedLearning.com  
**The Presidents of the United States of America**  
[In the order in which they served](#) [Alphabetical order](#) [Short table of Data](#)

President's Day Activities Abraham Lincoln

The President and Vice-President are elected every four years. They must be at least 35 years of age, they must be native-born citizens of the United States, and they must have been residents of the U.S. for at least 14 years. (Also, a person cannot be elected to a third term as President.)

President	Party	Term as President	Vice-President
1. <a href="#">George Washington</a> (1732-1799)	None, Federalist	1789-1797	<a href="#">John Adams</a>
2. <a href="#">John Adams</a> (1735-1826)	Federalist	1797-1801	<a href="#">Thomas Jefferson</a>
3. <a href="#">Thomas Jefferson</a> (1743-1826)	Democratic-Republican	1801-1809	Aaron Burr, George Clinton
4. <a href="#">James Madison</a> (1751-1836)	Democratic-Republican	1809-1817	George Clinton, Elbridge Gerry
5. James Monroe (1758-1831)	Democratic-Republican	1817-1825	Daniel Tompkins
6. <a href="#">John Quincy Adams</a> (1767-1848)	Democratic-Republican	1825-1829	John Calhoun
7. Andrew Jackson (1767-1845)	Democrat	1829-1837	John Calhoun, Martin van Buren
8. Martin van Buren			
9. William H. Harrison			
10. John Tyler			
11. James K. Polk			
12. Zachary Taylor			
13. Millard Fillmore			
14. Franklin Pierce (1803-1869)	Democrat	1853-1857	William King
15. James Buchanan (1791-1868)	Democrat	1857-1861	John Breckinridge

**Hundreds of millions of high-quality tables on the Web**

Quelle: Doan, Halevy and Ives. Principles of data Integration (Slides), 2012 [DHI12]

# Warum ist eine solche Integration so schwer? [DHI12]

---

- **System-bedingte Gründe:**
  - Verschiedene Plattformen
  - Anfragebearbeitung über mehrere (autonome) Systeme
- **Soziale Gründe:**
  - Finden relevanter Daten in Unternehmen
  - Beschaffen relevanter Daten in Unternehmen
  - Menschen zur Zusammenarbeit überreden
- **Logik-bedingte Gründe:**
  - Schema- und Datenheterogenität
  - Dies ist unabhängig von der jeweiligen Integrationsarchitektur

# Beispiel: Unternehmen FullServe

---

- Tätigkeitsfeld:
  - Internetprovider
  - Verkauf von Produkten wie Routern, voice-over-IP phones, etc.
- Verschiedene Abteilungen mit eigenen Datenbanken:
  - *Human Resource Department:*
    - Datenbank über Angestellte (Vollzeit und Teilzeit)
    - Datenbank über Bewerbungsverfahren
  - *Training and Development Department:*
    - Datenbank über Trainingskurse
  - *Sales Department:*
    - Datenbank über angebotene Dienste, Kunden und Verträge
    - Datenbank über verkaufte Produkte
  - *Customer Care Department:*
    - Datenbank über Anrufe an das *Help-Line Center*

# Beispiel: Unternehmen FullServe

## Employee Database

FullTimeEmps(ssn, empID, firstName,  
middleName, lastName)

Hire(empID, hireDate, recruiter)

TempEmployees(ssn, hireStart,  
hireEnd, name, hourlyRate)

## Training Database

Courses(courseID, name, instructor)

Enrollments(courseID, empID, date)

## Sales Database

Products(prodName, prodID)

Sales(prodID, custID, custName, address)

## Resume Database

Interviews(interviewDate, pID, recruiter,  
hireDecision, hireDate)

CVs(ID, resume)

## Services Database

Services(packName, textDescription)

Customers(name, ID, zipCode, streedAdr,  
phone)

Contracts(custID, packName, startDate)

## HelpLine Database

Calls(date, agent, custID, text, action)

Quelle: Doan, Halevy and Ives. Principles of data Integration, 2012 [DHI12]

## Beispiel: Unternehmen FullServe

- FullServe will nach Europa expandieren
- Zukauf des Unternehmens **EuroCard**
  - Kreditkartenanbieter
  - Ermöglicht Kunden Zugang zum Internet
- EuroCard hat eigene Datenbanken

### Employee Database

Emp(ID, firstnameMiddleInitial,  
lastName, salary)  
Hire(ID, hireDate, recruiter)

### Credit Card Database

Cards(CustID, cardNum,  
expiration, currentBalance)  
Customers(CustID, name, address)

### Resume Database

Interviews(ID, date, location, recruiter)  
CVs(candID, resume)

### HelpLine Database

Calls(date, agent, custID, description,  
followup)

Quelle: Doan, Halevy and Ives. Principles of data Integration, 2012 [DHI12]

## Beispiel: Unternehmen FullServe

---

Beispiel-Szenarien in denen Daten von mehreren Datenbanken benötigt werden:

- *Human Resource Department* möchte alle Angestellte wissen
- *Help-Line Center* braucht Kundendaten zur Problemlokalisierung und -behebung
- Aufsetzen einer Webseite mit allen angebotenen Produkten und Diensten inkl. Kundenbereich
- Herausfinden von Angestellten die früher bei Konkurrenzunternehmen gearbeitet haben
- Verknüpfung von Help-Line Anrufen mit anderen Daten
  - Beseitigung von Defiziten
    - Bsp. Erhöhte Fehlerhäufigkeit von Diensten/Produkten die von Angestellten installiert wurden, die einen bestimmten Kurs besucht haben
  - Erschliessen neuer Geschäftsideen

# Architekturparadigmen

---

- **Materialisiert**

- A priori-Integration
- Zentrale Datenbasis
- Zentrale Anfragebearbeitung
- Typisches Beispiel: Data Warehouse

- **Virtuell**

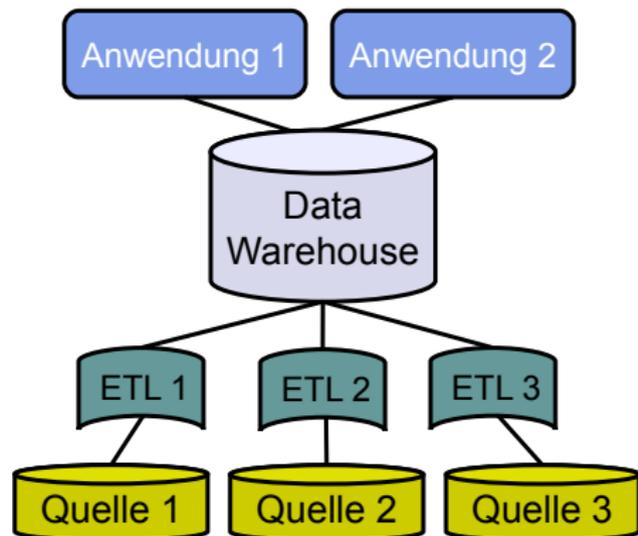
- *On demand*-Integration
- Dezentrale Daten
- Dezentrale Anfragebearbeitung
- Typisches Beispiel: Mediator-basiertes Informationssystem

- Existierende Architekturen befinden sich oft zwischen diesen Extremen

⇒ einige Daten werden materialisiert vorgehalten  
(z.B. durch den Einsatz von Caching)

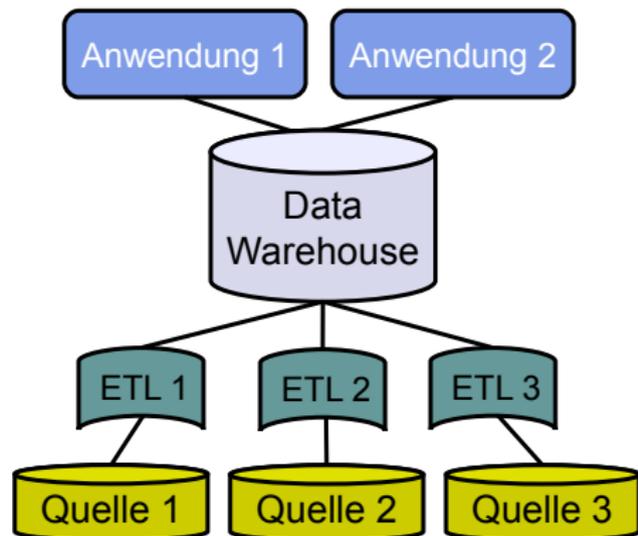
# Materialisierte Integration – Datenfluss

- *Push*
- Erstmaliges Laden (*population*) des DW (inkl. *Data Cleaning*)
- Periodischer Datenimport: *Updating materialized views*
- Redundante Datenhaltung
- Aggregation und Löschung alter Daten



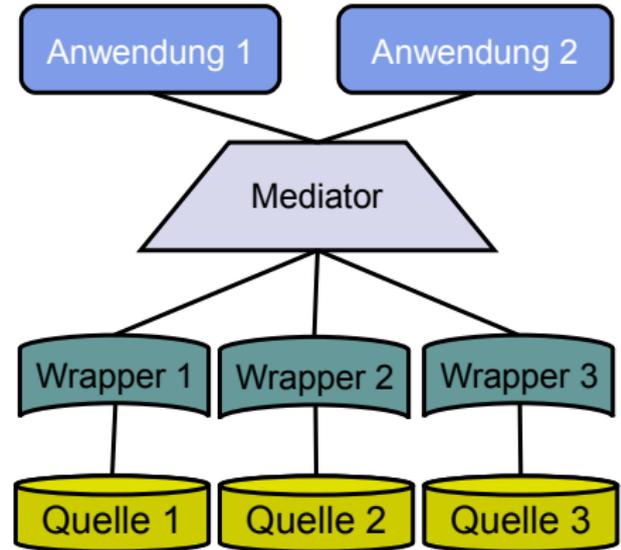
# Materialisierte Integration – Anfragebearbeitung/Schema

- **Anfragebearbeitung:**
  - Wie normale DBMS
  - Oft Aggregationsanfragen
  - *Decision Support*
- **Schema:**
  - Meist *Bottom-Up*-Entwurf
  - Schemaintegration
  - *Star-Schema*
    - *Fact Table*
    - *Dimension Tables*



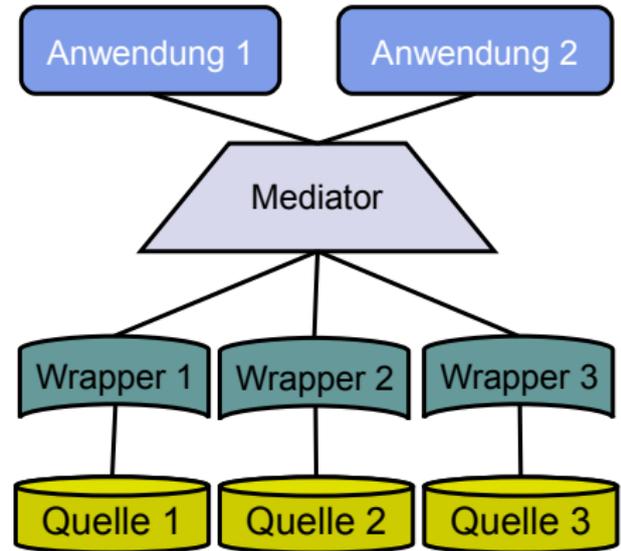
# Virtuelle Integration – Datenfluss

- *Pull*
- Daten sind in Quellen gespeichert
- Nur Anfragen und Ergebnisse werden übertragen (*Query Shipping*)
- *Data Cleaning* nur online möglich



# Virtuelle Integration – Anfragebearbeitung/Schema

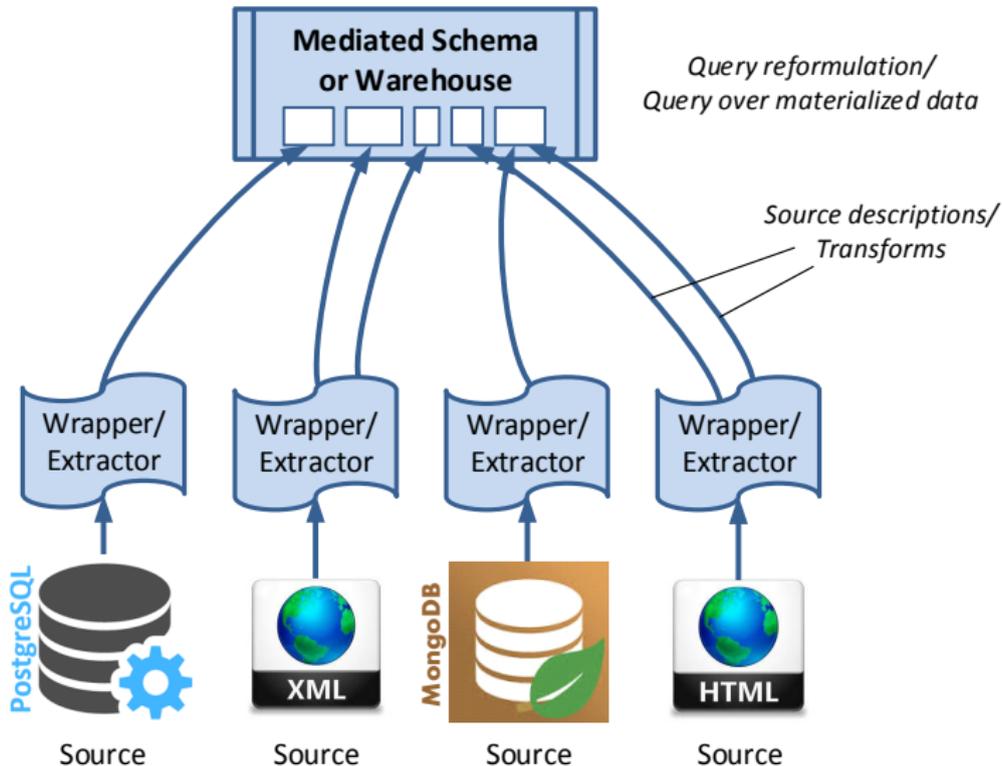
- **Anfragebearbeitung:**
  - Optimierung schwierig (Geschwindigkeiten und Fähigkeiten der Quellen)
  - Viele mögliche Pläne zur Berechnung des Anfrageergebnisses
- **Schema:**
  - Meist *Top-down* Entwurf
  - Leicht erweiterbar
    - neue Quellen
    - neue/geänderte *Mappings*
  - *Schema Mapping* statt Schemaintegration



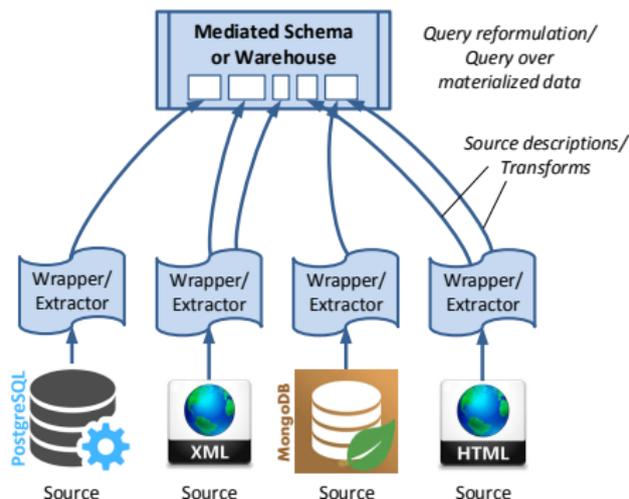
# Materialisiert vs. Virtuelle Integration – Vergleich

	<b>Materialisiert</b>	<b>Virtuell</b>
Aktualität	– (Cache)	+
Antwortzeit	+	–
Flexibilität	– (GaV)	+ (LaV)
Komplexität	+	++
Autonomie	–	+
Anfragemächtigkeit	+	–
Read/Write	+/+	+/-
Ressourcenbedarf	? (workload)	? (workload)
Vollständigkeit	+	? (OWA, CWA)
Datenreinigung	+	–
Informationsqualität	+	–

# Komponenten Virtueller Architekturen



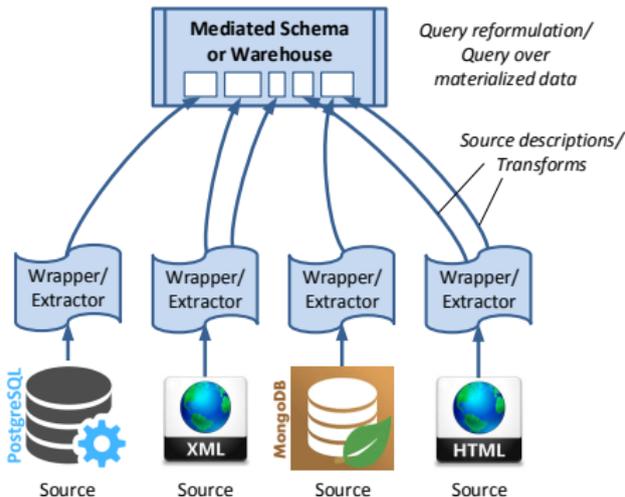
# Komponenten Virtueller Architekturen



## Quellen:

- verschiedene Datenmodelle
- verschiedene Anfragemächtigkeiten
- Quelle kann eine Anwendung sein, die wiederum komplexe Bearbeitungsschritte vollzieht

# Komponenten Virtueller Architekturen



## Wrapper:

- bekommt Anfrage in Sprache des Integrationssystems (z.B. relational oder XML)
- übersetzt Anfrage in Sprache der Quelle (z.B. HTTP Request)
- sendet Anfrage an Quelle
- transformiert Ergebnis (z.B. HTML Datei) in Datenmodell des Integrationssystems (z.B. Tupelmenge oder XML Datei)

# Wrappers

2.  **The Best of the Three Tenors (Audio CD)**  
 ~ by Luciano Pavarotti, Placido Domingo, Jose Carreras  
 Avg. Customer Rating: ★★☆☆☆  
 ([Recommend this item](#))

Usually ships in 24 hours  
 List Price: ~~\$48.98~~ [Used & new](#) from \$8.95  
[Buy new](#): \$14.99

3.  **The Three Tenors In Concert 1994 (Audio CD)**  
 ~ by Jules Massenet, Federico Moreno Torroba, Richard Rodgers  
 Avg. Customer Rating: ★★★★★  
 ([Recommend this item](#))

Usually ships in 24 hours  
 List Price: ~~\$41.98~~ [Used & new](#) from \$1.79  
[Buy new](#): \$10.99 [Club price](#): \$8.49

4.  **Trombonastic (Audio CD)**  
 ~ by Joseph Alessi  
 Avg. Customer Rating: ★★★★★  
 ([Rate this item](#))

Usually ships in 24 hours  
 List Price: ~~\$48.98~~ [Used & new](#) from \$14.23  
[Buy new](#): \$14.99

5.  **The Three Tenors Christmas (Audio CD)**  
 ~ by Carreras, Domingo, Pavarotti  
 Avg. Customer Rating: ★★☆☆☆  
 ([Recommend this item](#))

Usually ships in 3 to 4 days  
 List Price: \$13.98 [Used & new](#) from \$1.89  
[Buy new](#): \$13.98



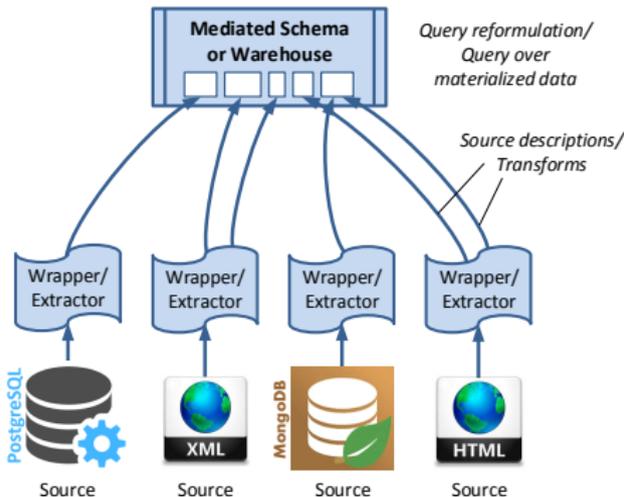
```

<cd>  <title> The best of ... </title>
      <artist> Abiteboul </artist>
      <artist> Pavarotti </artist>
      <artist> Domingo </artist>
      <price> 19.95   </price>

</cd>

...
  
```

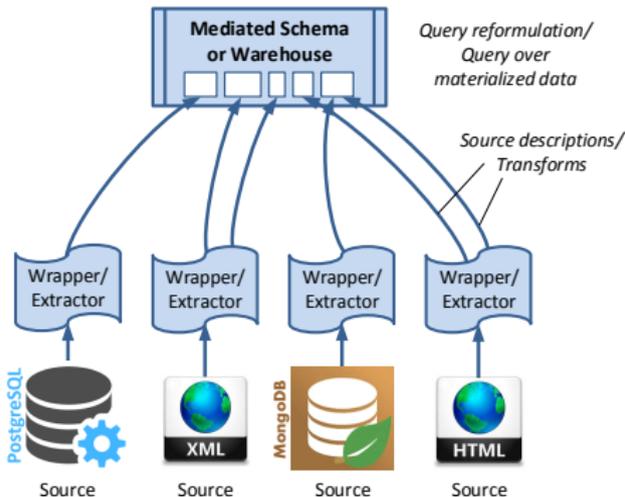
# Komponenten Virtueller Architekturen



## Mediated/Globales Schema:

- dient zur Interaktion mit dem Benutzer
- konzipiert für die Integrationsanwendung (beinhaltet daher nur einen Teil der Aspekte aus den Quellen)
- ist lediglich logisch und dient zur Formulierung von Anfragen

# Komponenten Virtueller Architekturen



## Quellbeschreibungen:

- eine Beschreibung pro Quelle
- enthält alle Informationen die das System braucht um die Quelle zu nutzen
- mappt deklarativ Konzepte zw. globalem Schema und Quellschema
- beschreibt Transformation auf Datenwertebene (z.B. für Konventionen, Einheiten)

# Quellbeschreibungen

## Mediated Schema

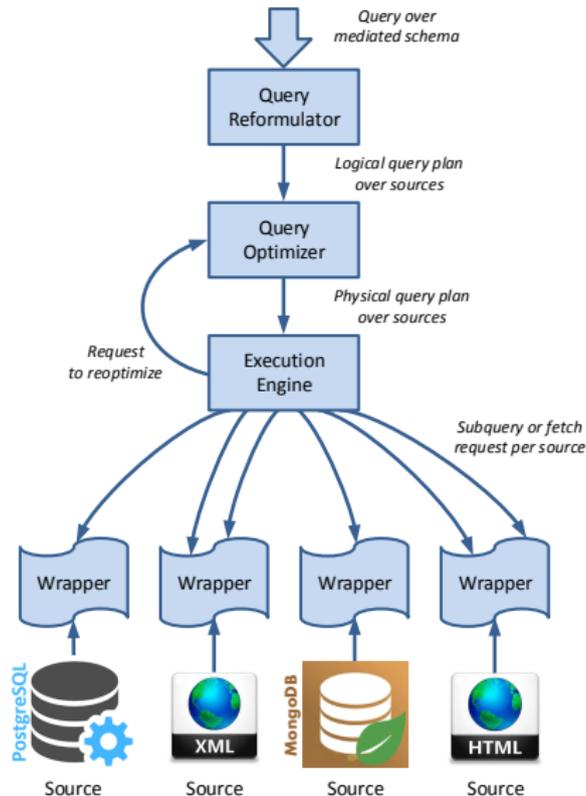
CD: ASIN, Title, Genre,...

Artist: ASIN, name, ...

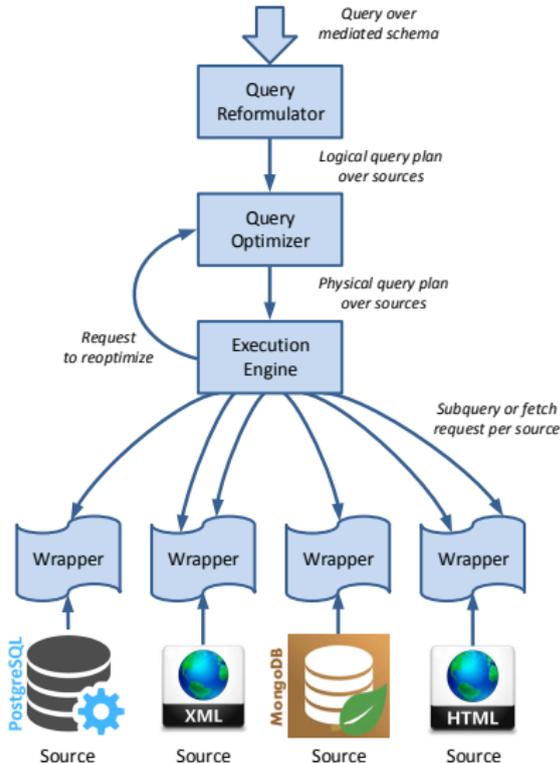
logic



# Anfragebearbeitung (Virtuell)



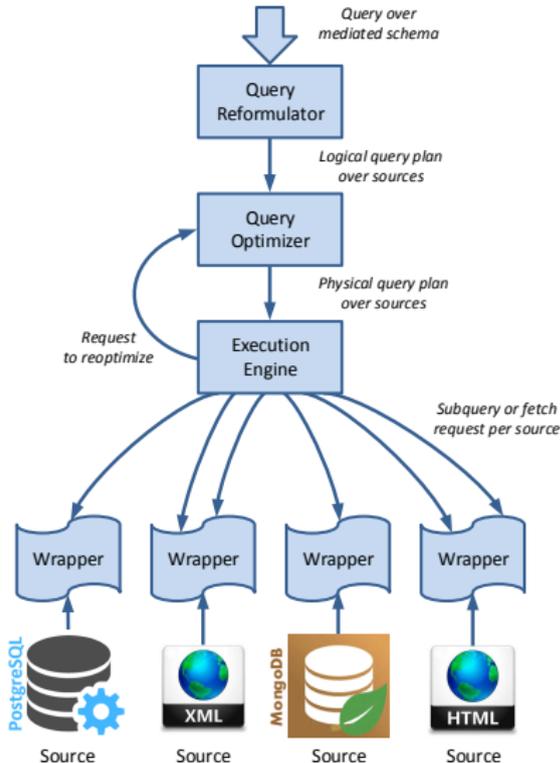
# Anfragebearbeitung (Virtuell)



## Anfrageumschreibung:

- Gegeben: Anfrage auf globales Schema
- Benötigt: Anfragen auf Quellschemata
- Umschreibung mit Hilfe der Quellbeschreibungen
- Ergebnis: Logischer Anfrageplan (inkl. Kombination der Quellenanfragen)
- mehrere Logische Anfragepläne möglich

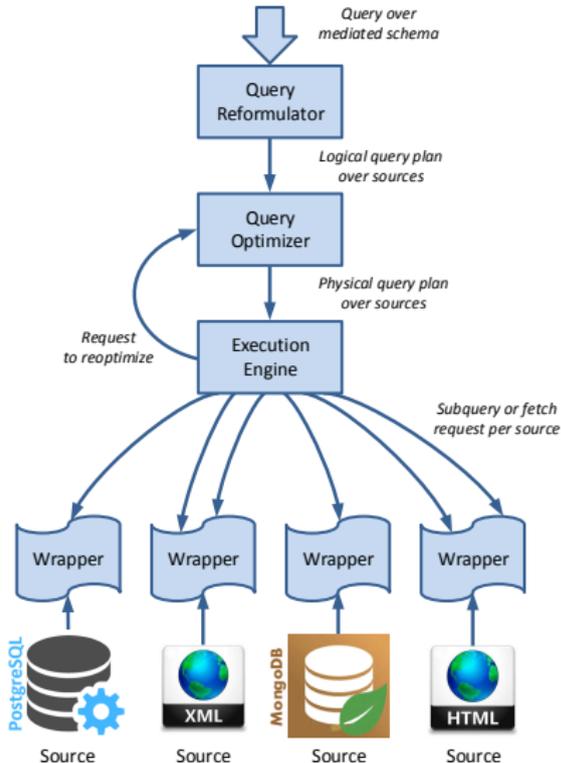
# Anfragebearbeitung (Virtuell)



## Anfrageoptimierung:

- Ergebnis: Physischer Anfrageplan
- bestimmt exakte Reihenfolge in der die Quellen angefragt werden
- bestimmt wann, wie (z.B. Join, Union) und wo (in Quelle oder im Zielsystem) Quellergebnisse kombiniert werden
- bestimmt wann und wo Selektionen durchgeführt werden
- Verteilung von Ressourcen des Zielsystems (Speicher, Prozessor)
- Schnelligkeit vs. Vollständigkeit

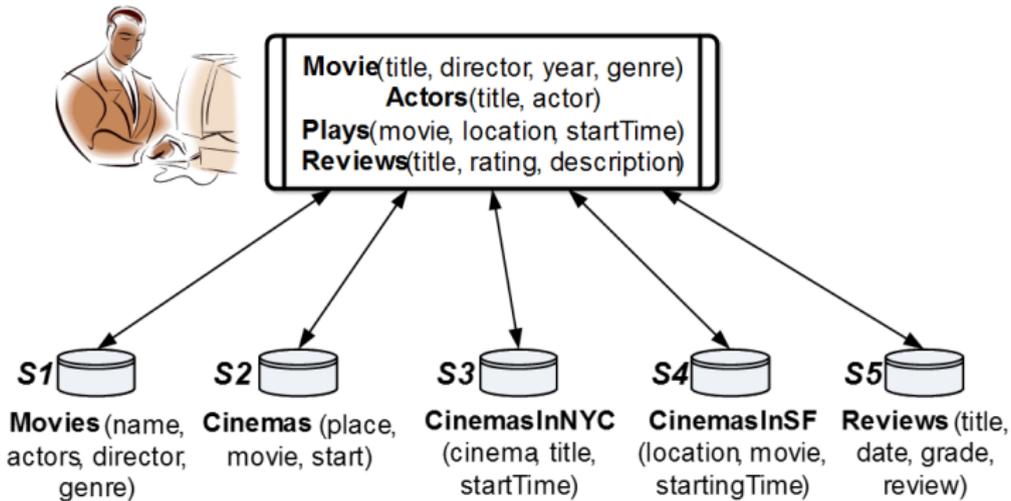
# Anfragebearbeitung (Virtuell)



## Anfrageausführung:

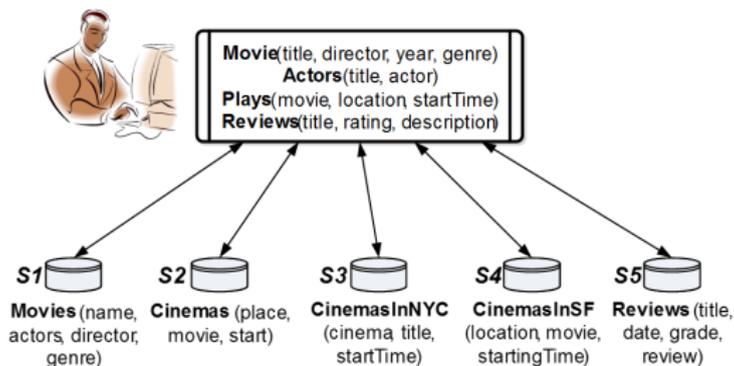
- Ausführung des Physischen Anfrageplans
- verteilt Teilanfragen an die Wrapper
- kombiniert die Ergebnisse der einzelnen Wrapper
- Anfrage beim Optimizers für einen anderen Plan falls Komplikationen auftreten (z.B. Ausfall einer Quelle)

# Anfragebearbeitung am Beispiel



Quelle: Doan, Halevy and Ives. Principles of data Integration (slides), 2012 [DHI12]

# Anfragebearbeitung am Beispiel



- *S1*: Filme
- *S2*: Filmvorstellungen im ganzen Land (unvollständig)
- *S3*: Filmvorstellungen in New York (vollständig)
- *S4*: Filmvorstellungen in San Francisco
- *S5*: Filmreviews
- *S2 - S4* benötigen einen Filmtitel als Eingabe

# Anfragebearbeitung am Beispiel

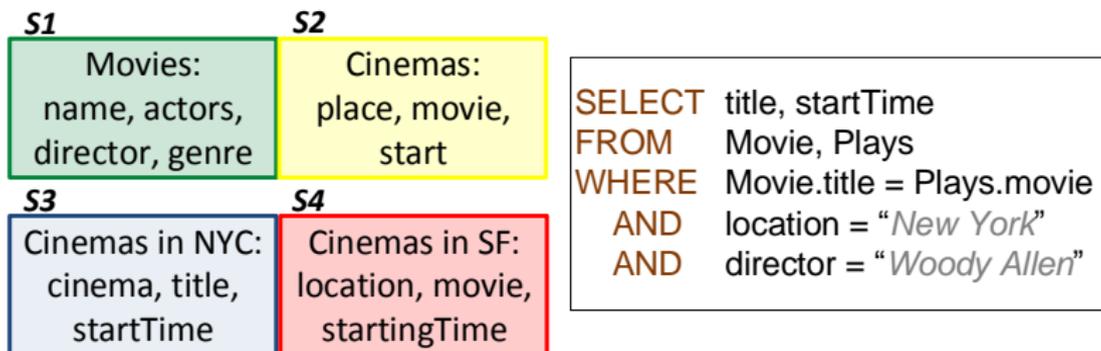
Filmvorstellungen in New York bei denen der Regisseur 'Woody Allen' heißt:

```
Movie: title, director, year, genre  
Actors: title, actor  
Plays: movie, location, startTime  
Reviews: title, rating, description
```

```
SELECT title, startTime  
FROM Movie, Plays  
WHERE Movie.title = Plays.movie  
      AND location = "New York"  
      AND director = "Woody Allen"
```

Quelle: Doan, Halevy and Ives. Principles of data Integration (slides), 2012 [DHI12]

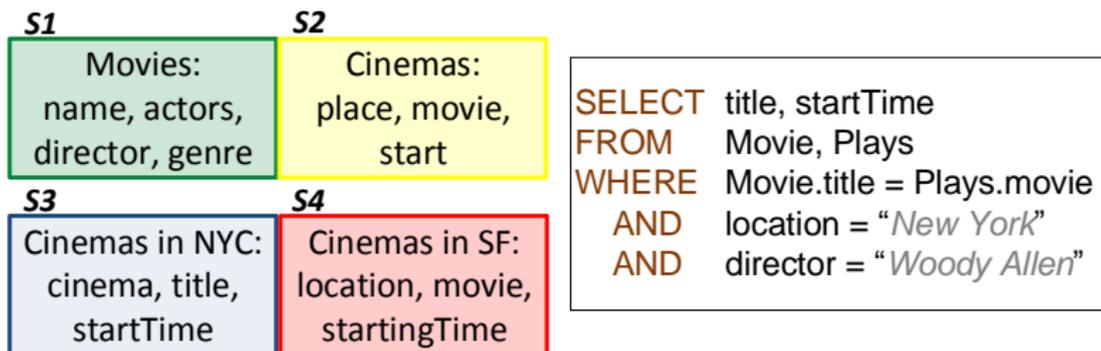
# Anfragebearbeitung am Beispiel



Anfrageumschreibung:

- Tuples für *Movie* können Quelle *S1* entnommen werden
  - Tuples für *Plays* in New York können den Quellen *S2* und *S3* entnommen werden (*S3* ist vollständig für New York)
  - *S2* und *S3* benötigen Filmtitel (nicht Teil der Anfrage)
- ⇒ *S1* muss zuerst angefragt werden
- Zwei Logische Anfragepläne (*S1* und *S2* oder *S1* und *S3*)

# Anfragebearbeitung am Beispiel



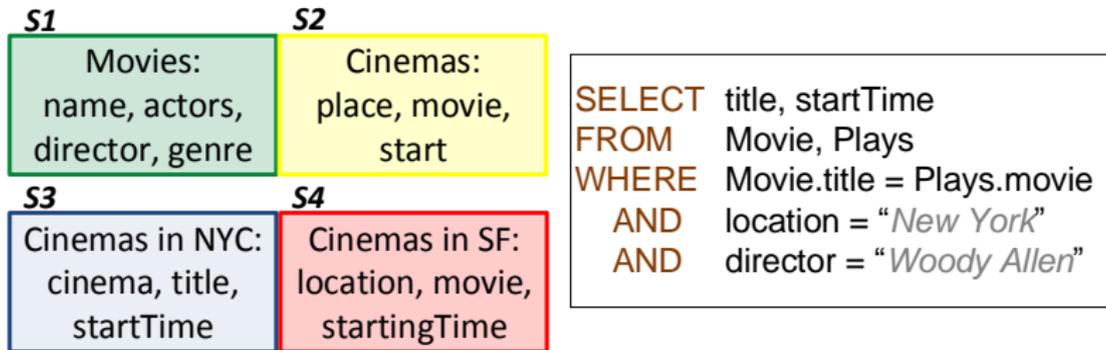
Anfrage für Quelle *S1*:

```

SELECT name AS title
FROM Movies
WHERE director = "Woody Allen"
  
```

- Selektion auf Regisseur kann direkt in Quelle berechnet werden
- Umbenennung des Attributes *name* in *title*

# Anfragebearbeitung am Beispiel



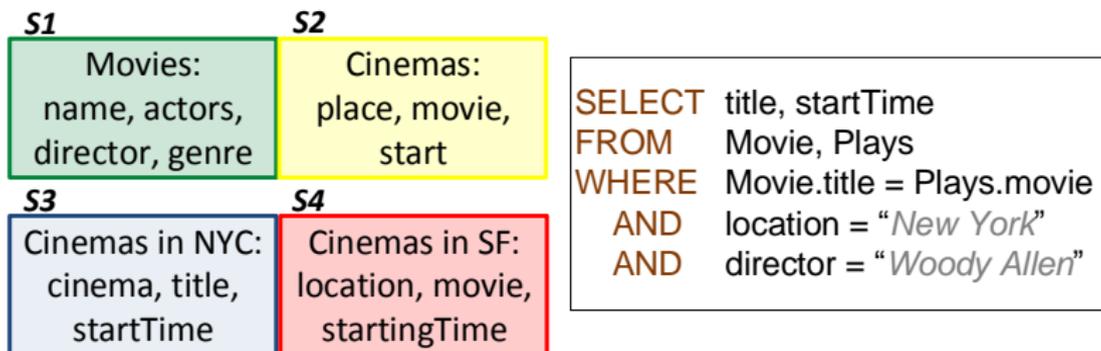
Anfrage für Quelle S2:

```

SELECT movie AS title, start AS startTime
FROM Cinemas
WHERE place = "New York"
      AND movie = @argument
  
```

- Selektion auf Ort kann direkt in Quelle berechnet werden
- Umbenennung der Attribute *movie* und *start*

# Anfragebearbeitung am Beispiel



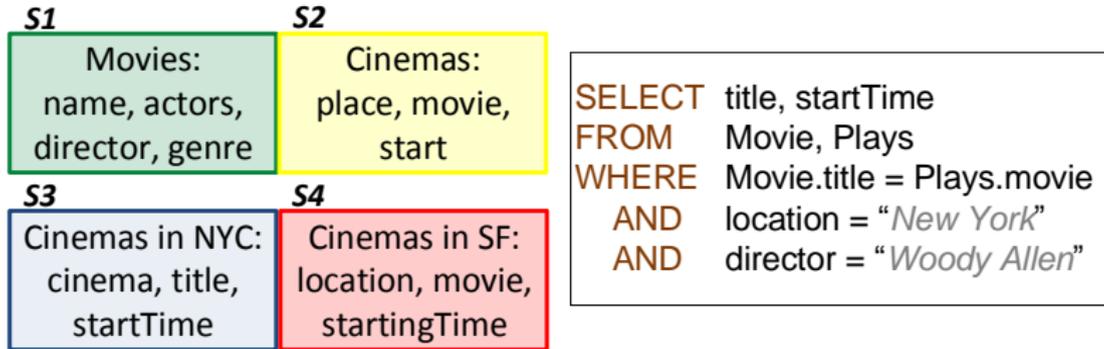
Anfrage für Quelle S3:

```

SELECT title, startTime
FROM CinemasInNYC
WHERE title = @argument
  
```

- Selektion auf Ort ist hier nicht notwendig (S3 enthält nur Filme aus New York)
- Umbenennung der Attribute nicht notwendig

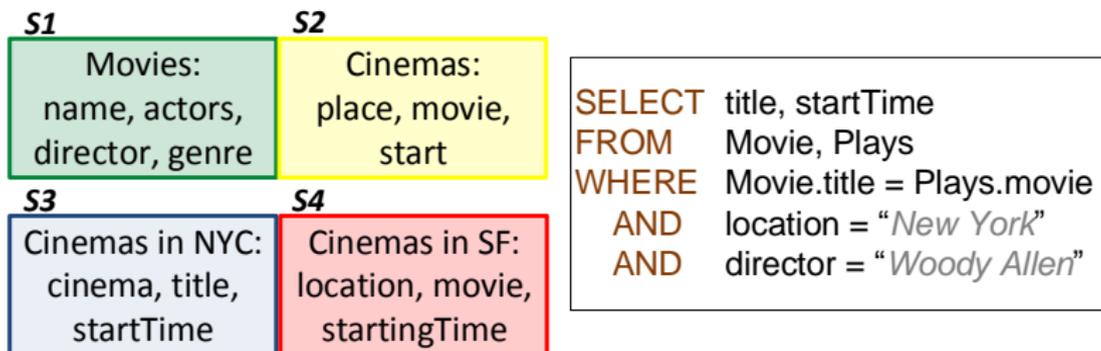
# Anfragebearbeitung am Beispiel



Auswahl eines oder mehrerer Pläne:

- S3 ist vollständig für New York
  - S2 ist evtl. unvollständig für New York
- ⇒ Wenn nur ein Plan ausgeführt werden soll, dann einer mit S3

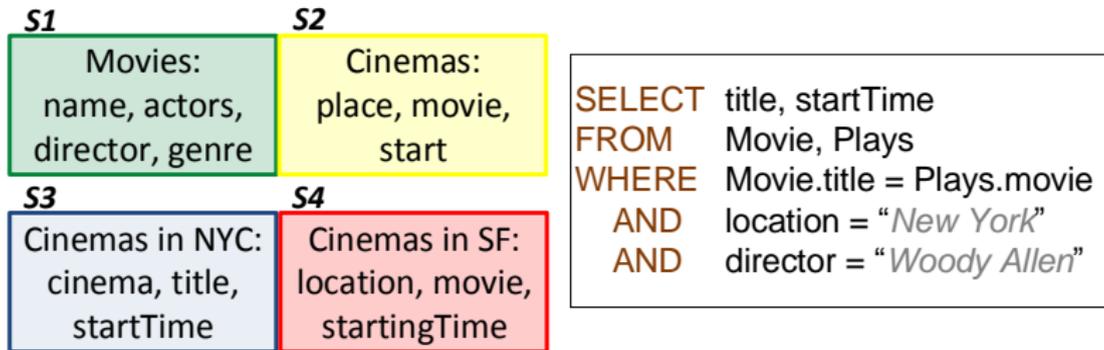
# Anfragebearbeitung am Beispiel



Anfrageoptimierung des Planes mit *S1* und *S3*:

- Auswahl eines Algorithmus um *S1* und *S3* zu joinen (streaming Tuples von *S1* zu *S3* oder komplett *S1* vor *S3*)
- Festlegung wo die Selektion auf den Regisseur durchgeführt wird (in *S1* oder im Zielsystem)

# Anfragebearbeitung am Beispiel



Anfrageausführung:

- Falls *S3* ausfällt oder zu langsam reagiert einen anderen Anfrageplan anfordern (in unserem Fall den mit *S1* und *S2*)

# Schema Matching, Mapping & Datenintegration

---

- **Schema Matching:**
  - Vergleich von Schema-Elementen
  - Zwischen zwei Quellen (Bottom-Up) oder zwischen Quelle und globalem Schema (Top-Down)
- **Schema Mapping:**
  - Ableiten einer Quellbeschreibung (virtuell) oder Transformationsanfrage (materialisiert) basierend auf den Matchingergebnissen
- **Datenintegration:**
  - Zusammenführen der Ergebnisse der einzelnen quellenspezifischen Teil-/Transformationsanfragen
  - Erkennen von semantischen Redundanzen (Duplikaterkennung)
  - Zusammenführen von Duplikaten (Datenfusion)

# Beispiel zu Schema Matching/Datenintegration

- Gegeben: Zwei Webservices *getMov* und *myMov*

The screenshot shows a web service interface titled "GET MOVIES" in a yellow banner. Below the banner, there are two search options:

- Actor Search:** Fields for "FIRST NAME:" and "LAST NAME:" with a "getMovieByActor" button.
- Title Search:** A "TITLE:" field with a "getMovieByTitle" button.

On the right side, there is a "SEARCH BY..." dropdown menu with two options: "ACTOR FIRST & LASTNAME" and "MOVIE TITLE".

The screenshot shows a web service interface titled "MY AWESOME MOVIE WEB SERVICE" in a dark grey banner. Below the banner, there are two search options:

- Year Search:** Fields for "TITLE:" and "YEAR:" with a "MyMovies" button.

On the left side, there is a text block: "Find your favorite movies by simply searching by title and year!"

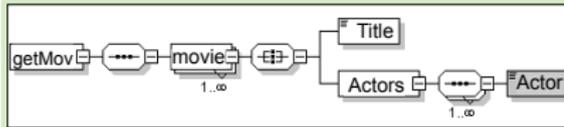
# Beispiel: Quellen

## Web Service getMov



```
<movie>
  <Title> Troy </Title>
  <Actors>
    <Actor> Eric Bana </Actor>
    <Actor> Brad Pitt </Actor>
  </Actors>
</movie>
```

- Operationen:
  - ♦ `getMovieByActor(firstName, lastName)`
  - ♦ `getMovieByTitle(title)`
- Ausgabestruktur:

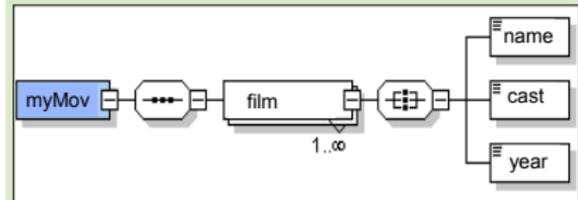


## Web Service myMov



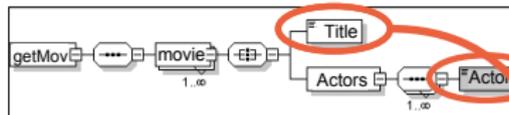
```
<film>
  <name> Troy </name>
  <cast> Pitt & Cox</cast>
  <year> 2003 </year>
</film>
```

- Operation:
  - `myMovies(Actor, Year)`
- Ausgabestruktur:

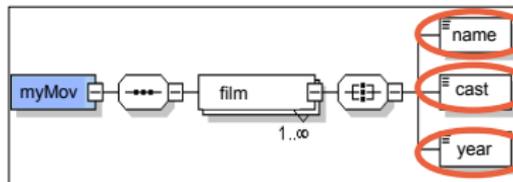


Quelle: Melanie Herschel, Universität Stuttgart

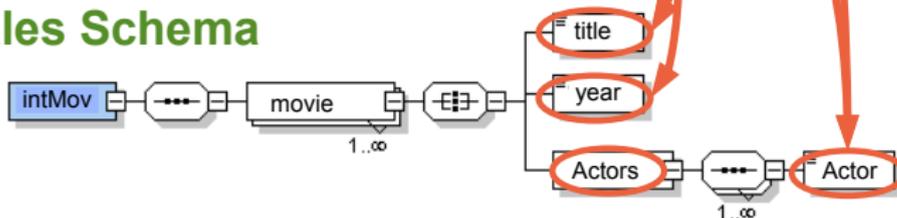
# Beispiel: Schema Matching



Korrespondenzen



## Globales Schema



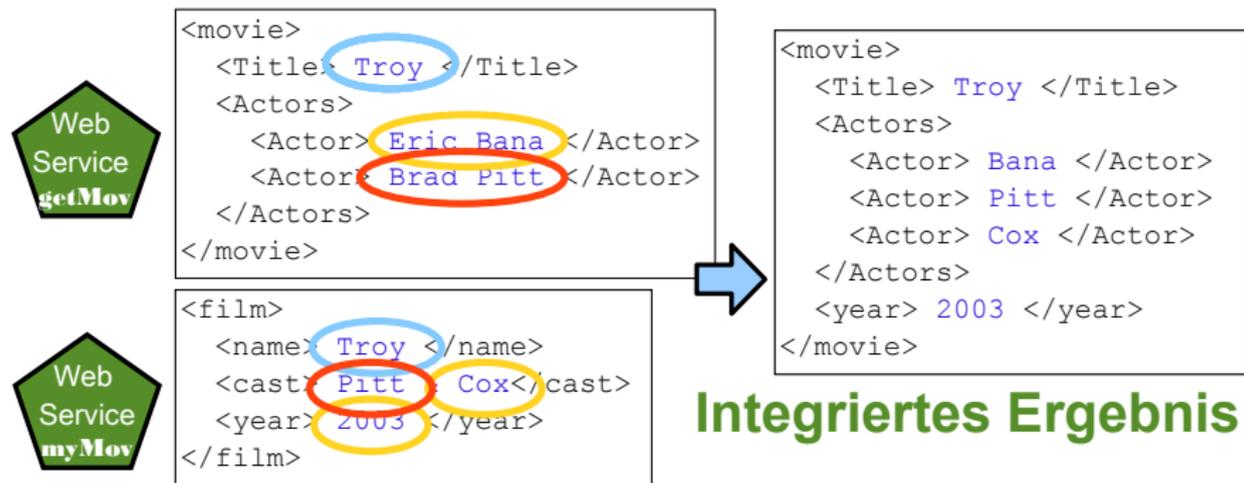
## Beispiel: Duplikaterkennung

- Handelt es sich bei den Filmen, die von Web Services getMov und myMov zurückgegeben werden, um denselben Film?
- Um dies festzustellen, müssen wir
  - (1) semantische Äquivalenzen (Korrespondenzen) beider Strukturen ermitteln und
  - (2) die Daten vergleichen.



## Beispiel: Datenfusion

- Titel stimmt überein  $\Rightarrow$  kein Konflikt
- Eric Bana, Cox & 2003 nur in einer Quelle  $\Rightarrow$  Unsicherheit
- Widersprüchliche Daten  $\Rightarrow$  Konflikt



# Literatur

---

- [Chr12] Peter Christen.  
*Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection.*  
Springer, 2012.
- [DHI12] Anhai Doan, Alon Halevy, and Zachary Ives.  
*Principles of Data Integration.*  
Morgan Kaufmann, 2012.
- [LN06] Ulf Leser and Felix Naumann.  
*Informationsintegration.*  
dpunkt.verlag, 2006.  
In German.
- [NH10] Felix Naumann and Melanie Herschel.  
*An Introduction to Duplicate Detection.*  
Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2010.