



Informationsintegration

Mediator-basierte Integrationsysteme

Ulf Leser



Informationsveranstaltung
online am **24.11. & 14.12 2021**, 17:30 Uhr
Anmeldung/Fragen: international.mnf@hu-berlin.de
(Zugangsdaten folgen nach Anmeldung + über Webseite)

Erasmus+

Was bietet ERASMUS+ ?

- Große Auswahl an interessanten Universitäten in Europa
- Gute Chancen bei der Platzvergabe
- Finanzielle Unterstützung während d. Auslandsaufenthalts
- Absicherung für die Anerkennung von Kursen
- Betreuung von Anfang bis Ende

über 200
Erasmus-Plätze
an der Fakultät
in 24 Ländern

Wohin soll es gehen? Wie finde ich eine Partneruniversität?

→ online: <https://hu.berlin/partnerunis-mnf>

Ausschreibung 2022/23 und Bewerbung:

→ online: hu.berlin/mnf-erasmus

Kontakt

→ Internationales Büro der Fakultät: Monique Getter, Saskia Bacher
(international.mnf@hu-berlin.de)

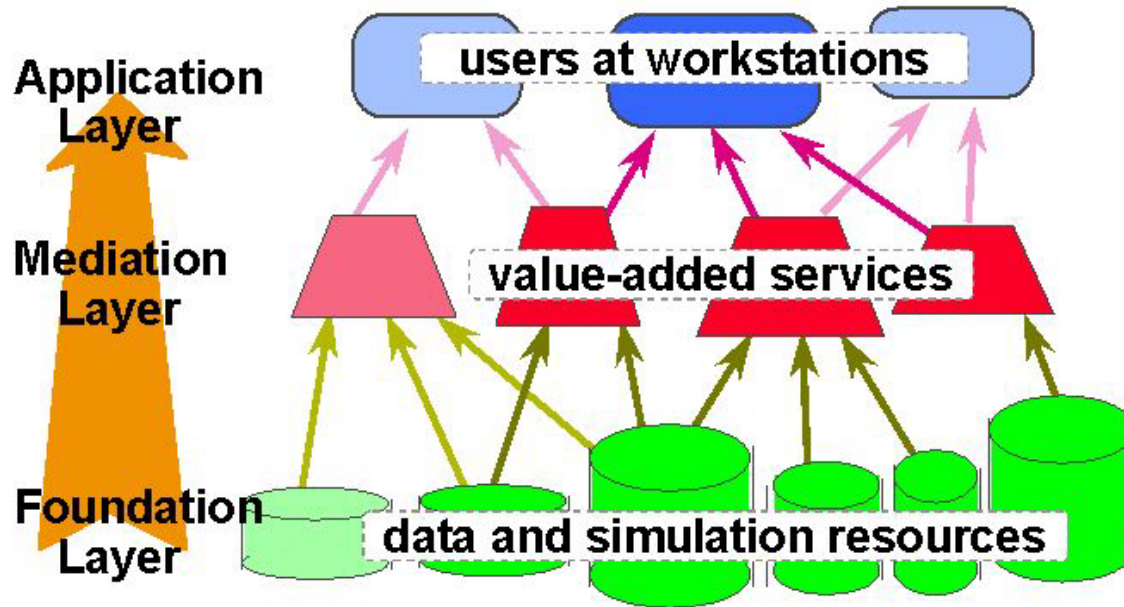
Bewerbungsfrist: 31. Januar 2022

Mathematisch-Naturwissenschaftliche Fakultät

Inhalt dieser Vorlesung

- Mediator-Wrapper Architektur
 - Übersicht
 - Mediatoren
 - Wrapper
- Beispiel

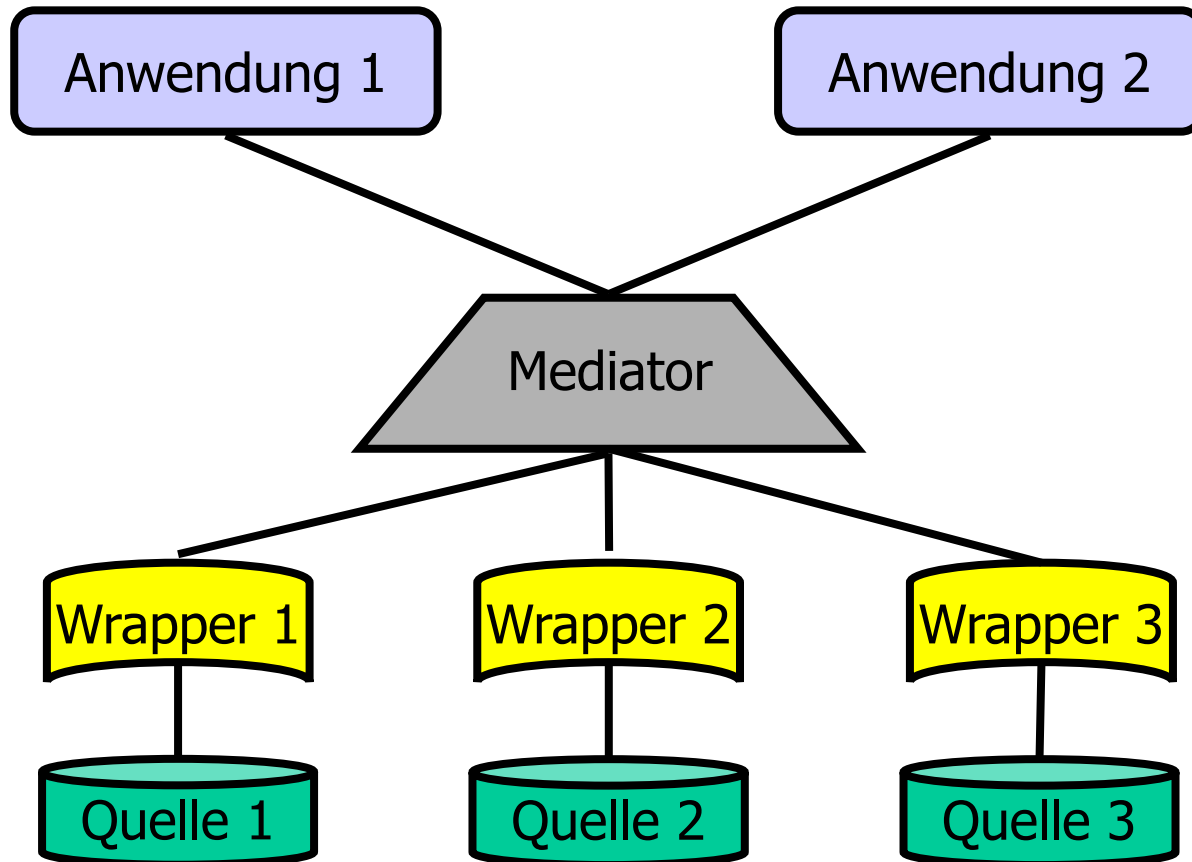
Daten werden zu Informationen



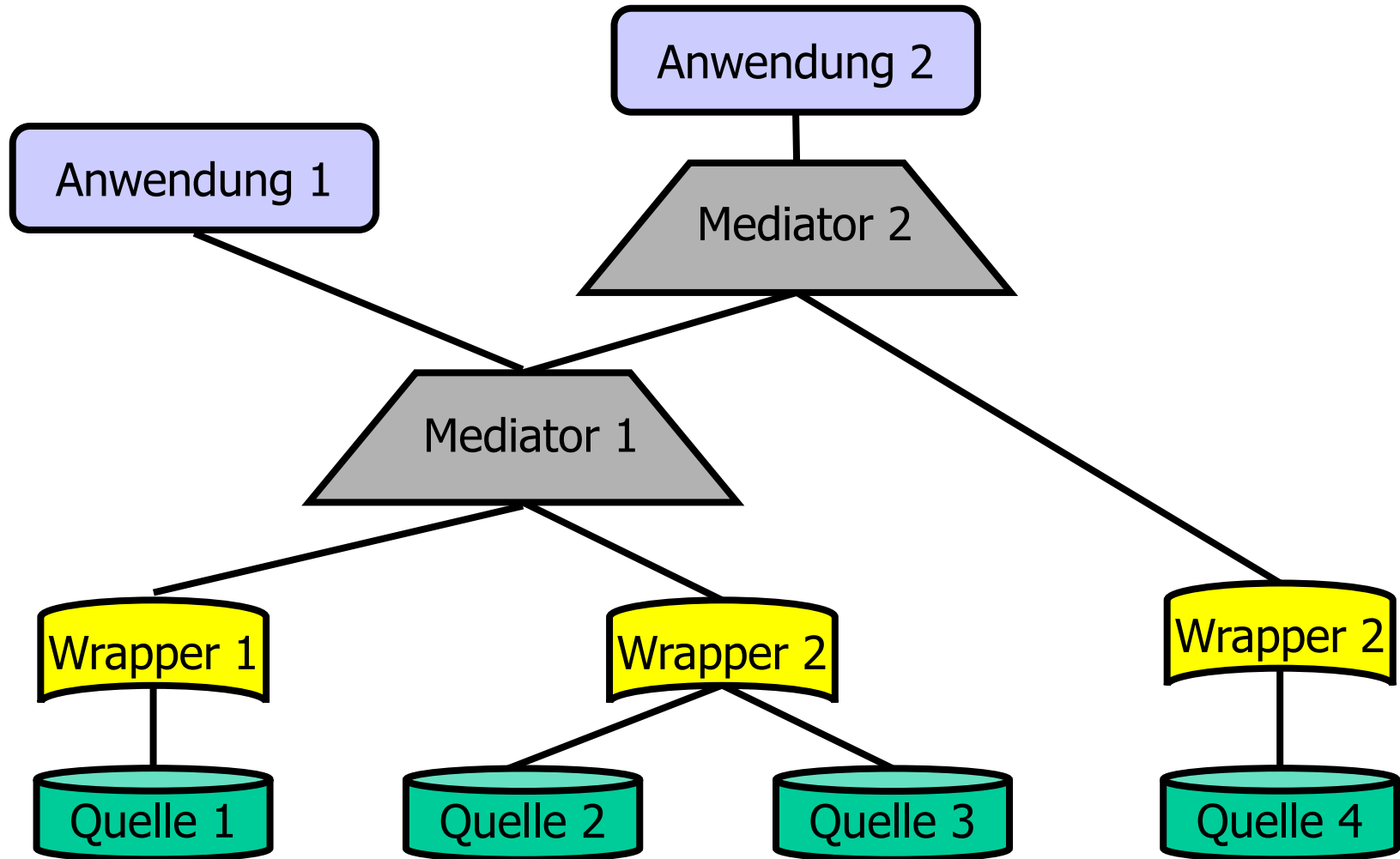
Gio Wiederhold 1999 15

„A mediator is a software module that exploits **encoded knowledge** about certain sets or subsets of data to create information for a **higher layer of applications**“[Wie92]

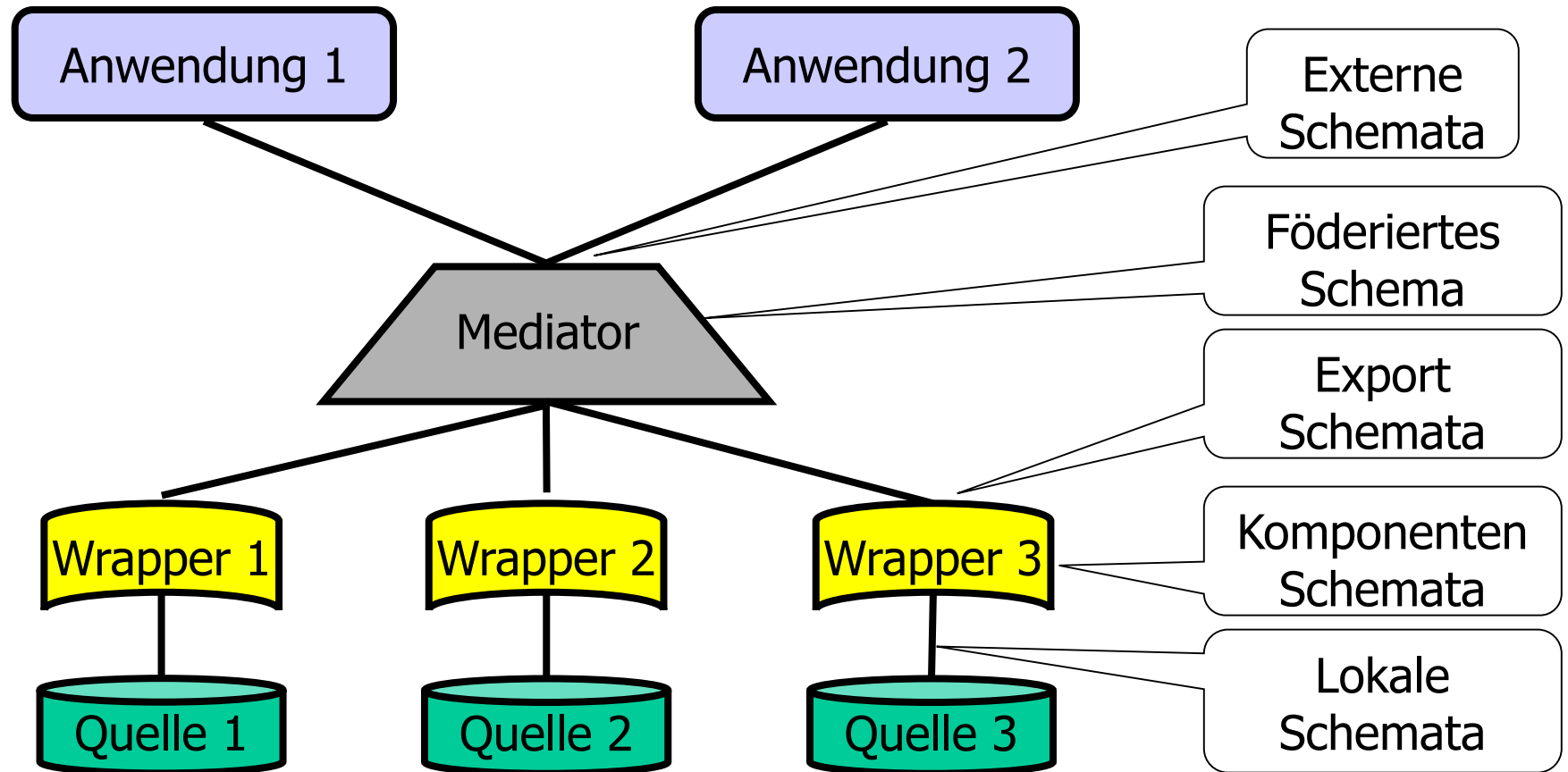
Architektur



Varianten



Vergleich zur 5-Schicht Architektur



Überblick

- Mediator-Wrapper Architektur
 - Übersicht
 - Mediatoren
 - Wrapper
- Beispiel

Funktionen eines Mediators

- Bereitstellung integrierter Daten für ein Gebiet
 - Überwindung semantischer und struktureller Heterogenität
 - Suche und Auswahl relevanter Informationsquellen
 - Auswahl und Ranking gemäß Anforderungen und Qualität
 - Inhärenter Top-Down Ansatz
 - Anreicherung mit Metadaten
 - Autorenschaft, Qualität, Vollständigkeit, Konsistenz, ...
 - Abstraktion (zum besseren Verständnis, als Auswahl)
 - Aggregation, Auswahl
 - Extensionale / intensionale Integration verschiedener Quellen
- Von allgemeinen, breit verfügbaren Daten zu spezifischen Informationen

Ein Ziel: Einfache Mediatoren

- „[A mediator] should be small and simple, so that it can be maintained by one expert or, at most, a small and coherent group of experts.“
 - Einfaches föderiertes Schema, begrenzte Domäne, einfache Schnittstellen
- Separation of concerns
 - Zerlege große Probleme in viele kleine Probleme
 - Klare, begrenzte Aufgabenbereiche
 - Können unabhängig von einander entwickelt und betrieben werden
 - Kann auf extensionaler oder intensionaler Zerlegung beruhen

Arbeitsweise eines Mediators

- Siehe "Anfragebearbeitung in virtuellen Szenarien"

Überblick

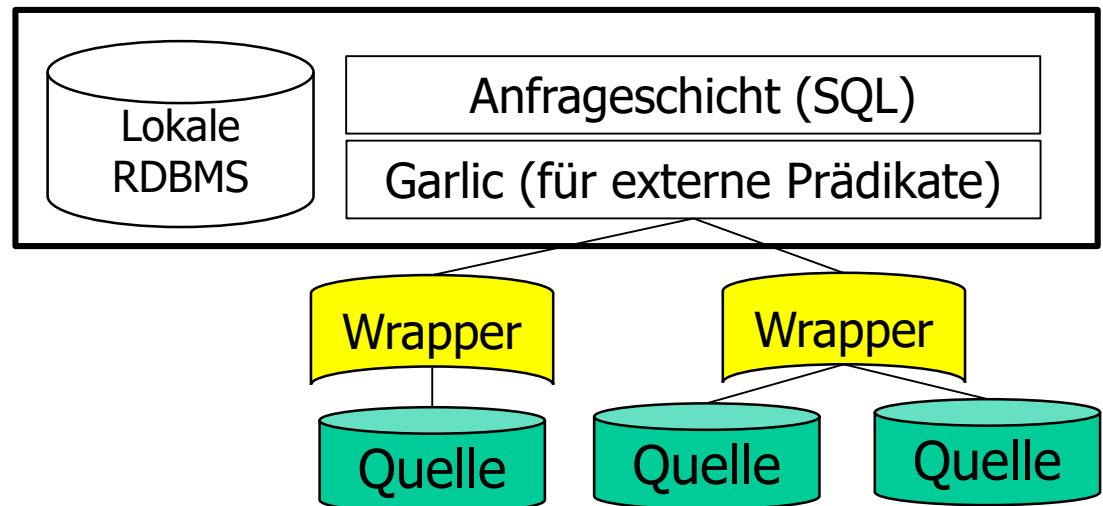
- Mediator-Wrapper Architektur
 - Übersicht
 - Mediatoren
 - Wrapper
 - Beispiel: Garlic
 - Wrappen von Web-Quellen
- Beispiel

Wrapper

- Komponenten, die die Kommunikation **zwischen Mediatoren und Datenquellen** herstellen
 - Spezialisiert auf eine Quelle oder einen Typ von Quelle
 - Haben ein Export-Schema
 - **Übersetzen Anfragen und Ergebnisse** zwischen Mediator (kanonisches Datenmodell, globale Anfragesprache) und Quelle (quellspezifisches Datenmodell und Anfragesprache)
 - Lösen **Schnittstellen-, syntaktische und Datenmodellheterogenität**
- Grenze zw. Mediator und Wrapper i.A. nicht klar definiert
 - Design-Entscheidung – die beste „Separation of Concerns“
 - **Wartbarkeit, Verantwortlichkeit**, klare Schnittstellen, Langlebigkeit

Beispiel: Wrapper im Garlic Projekt [Gar95]

- Fokus: **Optimierung**, Kostenmodell, Wrapper-Toolkit
- Weiterentwickelt zu (WebSphere) Information Integrator
- Aufgaben eines Wrappers in Garlic
 1. Modellierung und Zugriff auf die Daten
 2. Aufruf von **Methoden und Daten aus der Quelle**
 3. Hilfe bei der Anfrageplanung
 4. Anfrageausführung



Definition eines Wrapper

- DDL zur Registrierung / Konfiguration von Wrappern
- Technische Deklaration
 - Definition des entfernten Servers und des zu verwendenden Codes
 - Verschiedene Server können den gleichen Wrapper benutzen

```
CREATE SERVER proteindb
WRAPPER proteinwrapper
OPTIONS (NODE 'www.protein.', PORT '1234', VERSION ...);
```

- Inhaltliche Deklaration
 - Definition eines Schemas für einen Server
 - NICKNAME kann wie eine lokale Tabellen angesprochen werden

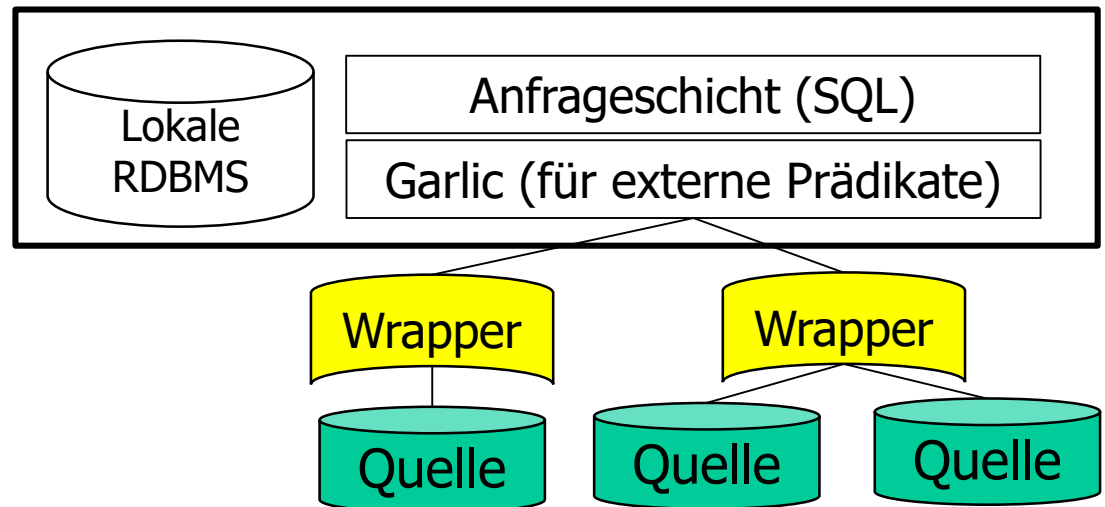
```
CREATE NICKNAME proteins ( att1, att2, ...)
SERVER proteindb;
```

- Orts-/ Schnittstellentransparenz

Syntax standardized as
SQL/MED – widely adopted
(DB2, MySQL, PostGres)

Garlic: Aufgaben eines Wrapper

1. Modellierung und Zugriff auf die Daten
 - Export der Daten in **objektrelationalem Datenmodell**
 - Zugriff über relationale Prädikate (Selektion, Join, UDF, ...)
2. Prädikate nach Möglichkeit **an Quelle delegieren**
 - Um besondere Fähigkeiten von Quellen auszuschöpfen



Garlic: Aufgaben eines Wrapper

3. Hilfe bei der Anfrageplanung

- Typische Queries auf Mediator-Ebene kombinieren Daten aus mehreren Quellen / Wrappern
- **Mediator** erzeugt alternative Pläne und sucht den schnellsten
 - Typisch RDBMS: Keine Unterstützung für semantische Integration
- Dazu: Mediator verschickt **Teilaufgaben** an Wrapper
 - Wrapper kann Pläne ablehnen (je nach Fähigkeiten der Quelle)
 - Mediator führt alles aus, was der Wrapper ablehnt
- Wrapper liefert null oder mehr **Teilpläne** pro Teilaufgabe zurück
 - Mit Angabe von: Kosten, Kardinalität, Sortierung, ...
- Mediator wählt auszuführenden Teilplan aus
 - Abhängig von Teilplänen anderer Wrapper

Beispiel - Selektion

- Quelle
 - Exportschema: `hotels(name, stars, rate, town, street)`
 - Constraint: Kann nur **ein einzelnes Selektionsprädikat** auswerten
 - Z.B: gibt es zwei Suchen in einer Web-API
- Globale Anfrage

```
SELECT name, town
FROM hotels
WHERE stars>3 AND rate<150;
```

 - Selektion auf `stars` **oder** auf `rate` (oder nichts) pushen
 - Wrapper liefert **Kostenschätzungen**, Mediator entscheidet

Export:	<code>name, town, rate</code>
Table:	<code>hotels</code>
Predicate:	<code>stars > 3</code>
Rows:	14.204
Rowsize:	150B
Est. time:	250ms

Export:	<code>name, town, stars</code>
Table:	<code>hotels</code>
Predicate:	<code>rate<150</code>
Rows:	8.900
Rowsize:	152B
Est. time:	455ms

Beispiel - Selektion

- Quelle
 - Exportschema: `hotels(name, stars, rate, town, street)`
 - Constraint: Kann nur **ein einzelnes Selektionsprädikat** auswerten
 - Z.B: gibt es zwei Suchen in einer Web-API
- Globale Anfrage

```
SELECT name, town
FROM hotels
WHERE stars>3 AND rate<150;
```

 - Selektion auf `stars` **oder** auf `rate` (oder nichts) pushen
 - Wrapper liefert **Kostenschätzungen**, Mediator entscheidet

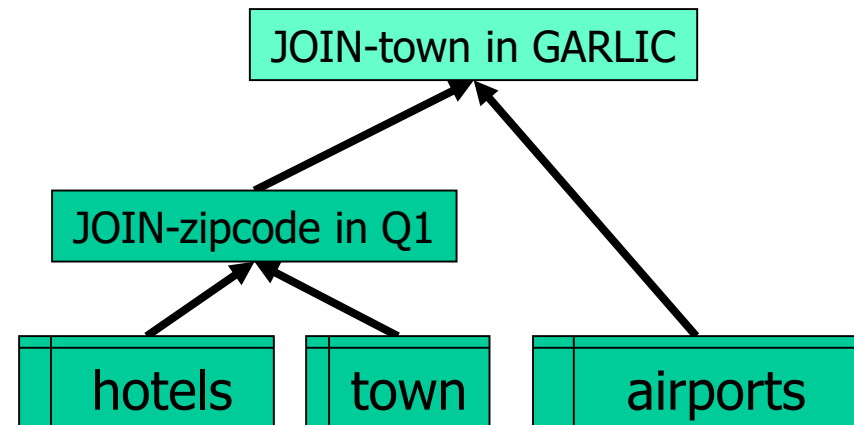
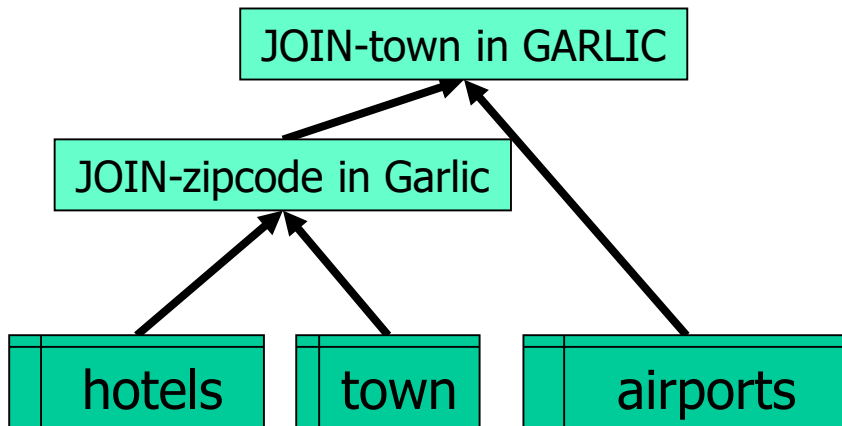
Export:	name, town, rate
Table:	hotels
Predicate:	stars > 3
Rows:	14.204
Rowsize:	150B
Est. time:	250ms

Export:	name, town, stars, rate
Table:	hotels
Predicate:	-
Rows:	123.720
Rowsize:	160B
Est. time:	8754ms

Beispiel – Joins

- 2 Quellen, drei Tabellen, 2 Joins

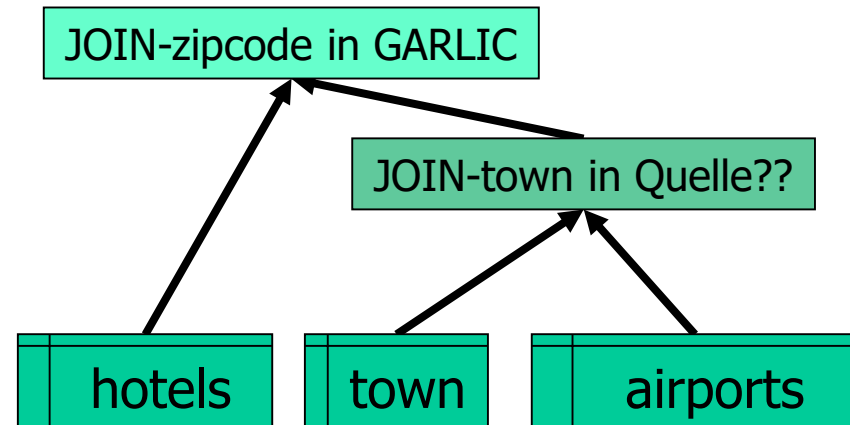
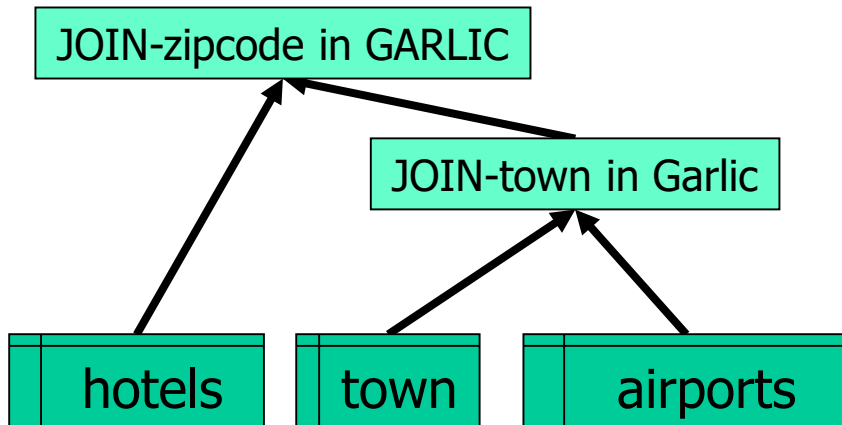
```
SELECT    t1.name, t2.name, t3.name
FROM      q1.hotels t1, q1.town t2, q2.airports t3
WHERE     t1.stars>3
          AND t1.zipcode = t2.zipcode
          AND t2.name = t3.town
```



Beispiel – Joins

- 2 Quellen, drei Tabellen, 2 Joins

```
SELECT    t1.name, t2.name, t3.name
FROM      q1.hotels t1, q1.town t2, q2.airports t3
WHERE     t1.stars>3
          AND t1.zipcode = t2.zipcode
          AND t2.name = t3.town
```

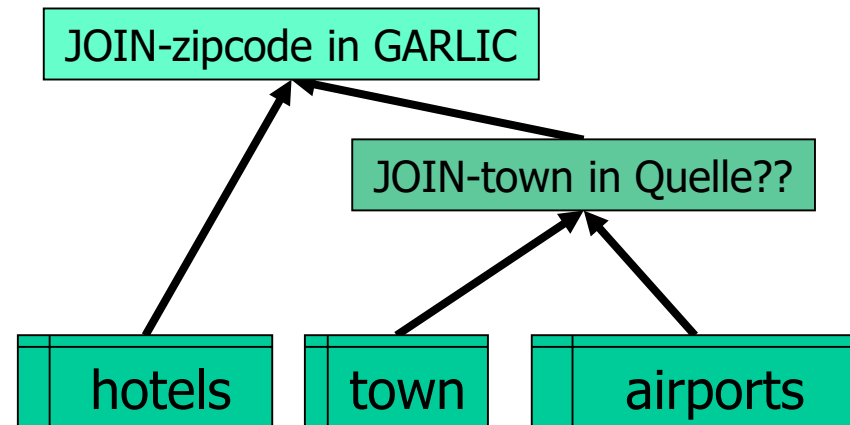
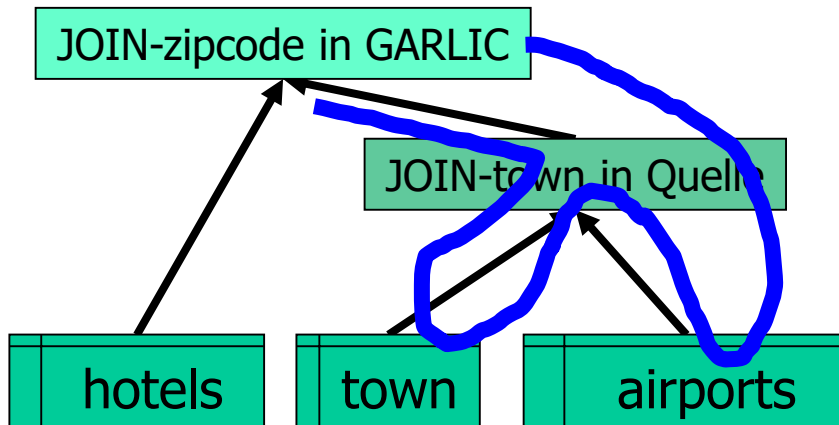


Beispiel – Joins

- 2 Quellen, drei Tabellen, 2 Joins

```
SELECT    t1.name, t2.name, t3.name
FROM      q1.hotels t1, q1.town t2, q2.airports t3
WHERE     t1.stars>3
          AND t1.zipcode = t2.zipcode
          AND t2.name = t3.town
```

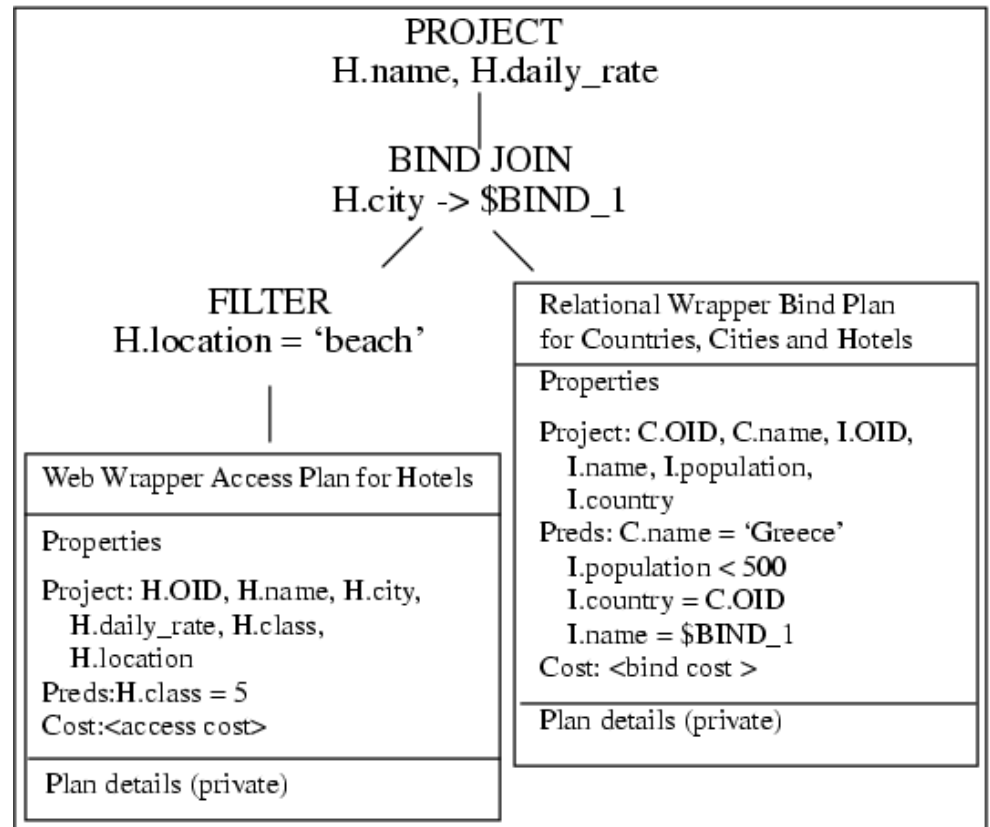
- Wie? Z.B. Bind-Join



Garlic: Aufgaben eines Wrapper

4. Anfrageausführung

- Mediator wählt optimalen Ausführungsplan
- Wrapper-Teilpläne werden von Wrappern ausgeführt
- Alles andere im Mediator



Beurteilung

- Vorteile
 - Flexible Optimierung
 - Schnelle Implementierung „dummer“ Wrapper
 - Sukzessive Verbesserung möglich ohne Anpassung des Optimierungsverfahrens (Mediator bleibt stabil)
 - Optimierung erfolgt automatisch durch Anfrageplanung
 - Berücksichtigung auch „exotischer“ Fähigkeiten von Wrappern möglich
 - Dezentraler Aufbau
 - Mediatoren wissen wenig über Wrapper
 - Weiterentwicklung von Wrappern erfolgt autonom
- Nachteile
 - Erheblicher Kommunikationsaufwand bei der Anfrageplanung
 - „Fat“ Wrapper: Anfragepläne verstehen und bewerten
 - Wichtig: Toolkits & SDK's bereitstellen

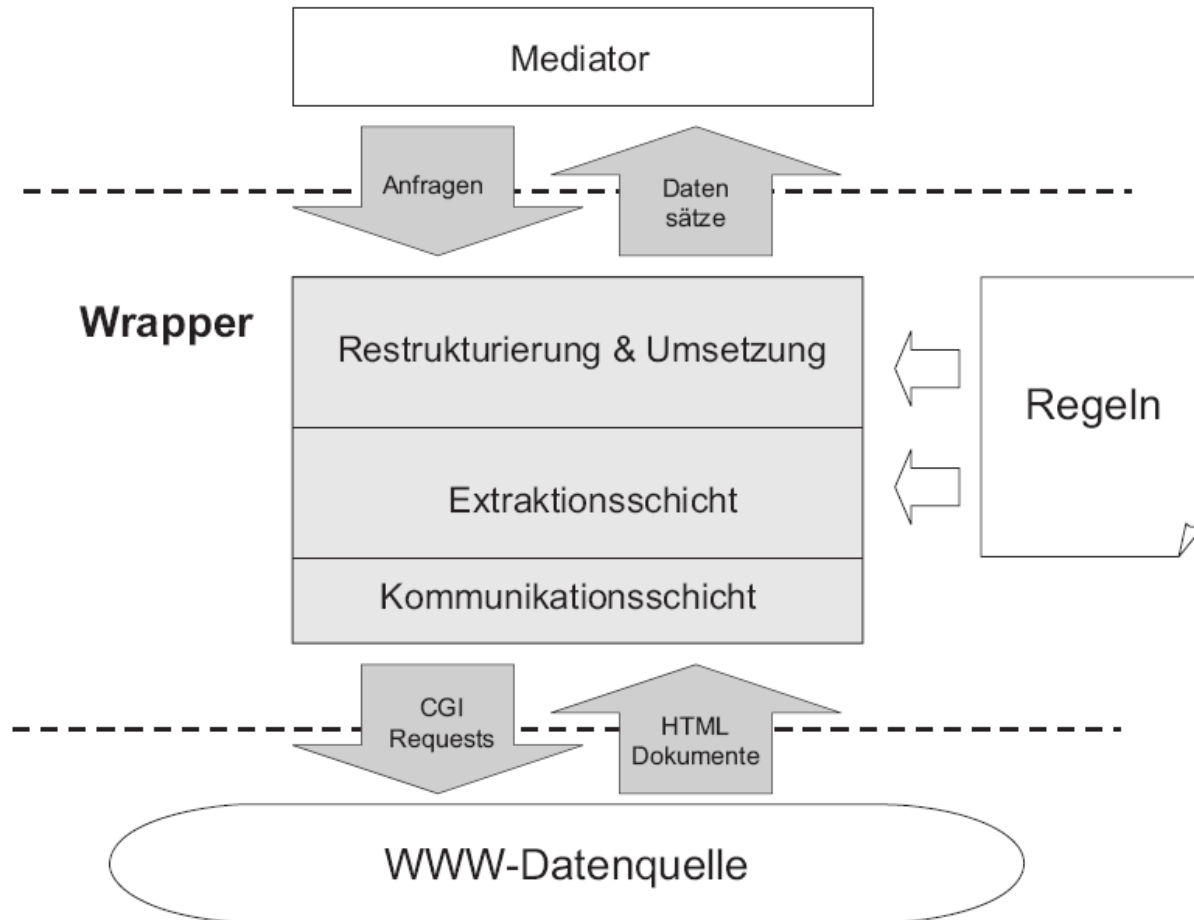
Überblick

- Mediator-Wrapper Architektur
 - Übersicht
 - Mediatoren
 - Wrapper
 - Beispiel: Garlic
 - [Wrappen von Web-Quellen](#)
- Beispiel

Web-Wrapper

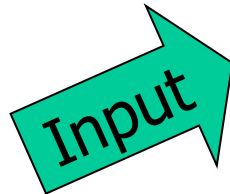
- „Screen Scraping“, „web scraping“
 - Umsetzung von Mediator- aufrufen in **HTTP Requests**
 - Die zu CGI / Servlet / PHP ... Requests werden
 - Die meistens zu (verborgenen) SQL Requests führen
 - **Extraktion** der gewünschten Daten aus HTML Seiten
 - Umwandlung in das Datenmodell des Mediators
- Diverse **eigene DSLs** zur Extraktion: WebL, Xcerpt, ...
- Kommerziell interessant
 - Trendanalysen (Blogs), Beobachtung der Konkurrenz, Business Intelligence, Meta-X (Produktkataloge, Reisen, Paper, ...)
 - Firmen: Lixto, OpenKarpov, UberMetrics, ...
- **Juristische Einschränkungen** beachten

Typischer Aufbau

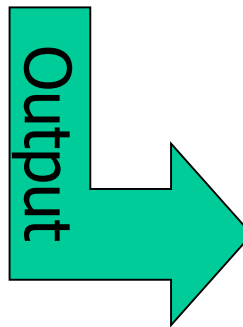


Beispiel: HTML-Tabellen

```
<table>
  <tr>
    <th>Name</th>
    <th>Adresse</th>
  </tr>
  <tr>
    <td>Müller</td>
    <td>Pankow</td>
  </tr>
  <tr>
    <td>Meier</td>
    <td>Rudow</td>
  </tr>
  ...
</table>
```



```
S := getPage( ... );
S := region(S, tableStart, tableEnd);
foreach R=region(S, trStart, trEnd);
  N := region(R, tdStart, tdEnd);
  clip(R, N);
  A := region(R, tdStart, tdEnd);
  createRecord( N, A );
...
```



Personen	
Name	Adresse
Müller	Pankow
Meier	Rudow
...	...

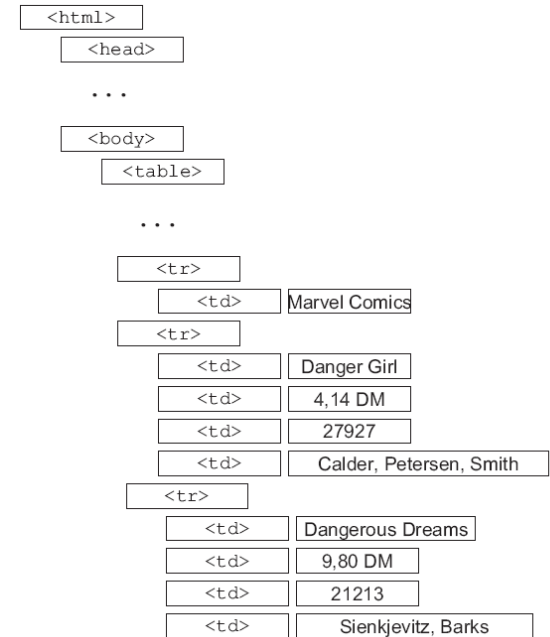
Zwei Herangehensweisen

- Navigation des **DOM Baums**

```
html.body.table[0].tr[1].td[0].txt
```

- Verwendung **regulärer Ausdrücke**

```
["Titel,Preis,BNr,Zeichner","Comic","<td>#</td><td>#</td><td>#</td><td>#</td>*"]
```



Vor- und Nachteile

- DOM Baum
 - Braucht den Baum: Speicherplatzintensiv, langsam
 - Gut bei **regulären Strukturen** (datenbankerzeugte Webseiten)
 - Schwierigkeiten mit irregulärem HTML
 - Anfällig für **Änderungen in Reihenfolge** oder neue Elemente
 - Kann Textfelder nicht weiter unterteilen (also doch RegExp)
- Regulärer Ausdrücke
 - Schnelle Ausführung
 - **Unübersichtlich**
 - Kompliziert, wenn **geschachtelte Elemente** benötigt werden
 - Benötigen stabile Ankerelemente (Überschriften etc.)

Beispiel: Lixto Visual Developer

The screenshot displays the Lixto Visual Developer 4.4.4 interface. The main window shows a Mozilla Web Browser displaying a hotel search page for "Lodging in Vienna (and vicinity)". The page includes a search form, a list of hotels (e.g., Falkensteiner Hotel am Schottenfeld), and a sidebar with "Euro vacation" and "Flight 4 nig from \$".

Three callout boxes highlight key components:

- Navigation Steps:** Points to the left sidebar, which contains a tree view of the application's structure, including folders like "1st customer", "2nd customer", and "3rd customer".
- Mozilla Web Browser:** Points to the central browser window displaying the hotel search results.
- Extraction Configuration:** Points to the bottom panel, which shows the "Filter" configuration for the "hotelist (Pattern)" data extractor. The configuration includes a "Method" of "XPath" and a "Node" of "body[1]/table[1]/tbody[1]/tr[1]/td[3]/table[1]/tbody[1]".

Quelle: http://www.ebizq.net/blogs/nari/visual_developer.jpg

Überblick

- Mediator-Wrapper Architektur
 - Übersicht
 - Mediatoren
 - Wrapper
- Beispiel: Houdini

Interface

Houdini - Mozilla Firefox

HuoDini! Humboldt-Universität zu Berlin Disaster Management Interface

Map Satellite Hybrid

17 Images on Flickr:

San Francisco Earthquake

Search for: San Francisco Earthquake From: 04/12/2006 To: 04/12/2006

Filter (injured < 4) and (dead > 0) change...

5 Posts on... (hide)

Celexa

Indeed, I think that such sist might greatly help towards a cotton-shed s'abuse, a totally new rescript in Non-forgiveness verse ... the Asamanj **Francisco earthquake** and ex-supercargo, for wilness in our swift-hunting land where

Done

Flickr more info

Flickr more info

Flickr more info

Flickr more info

Flickr more info

Flickr more info

Flickr more info

more...

Idea

- **Collection and integration** of relevant information after an earthquake
- Integrated access to various autonomous, distributed and heterogeneous data sources
- Focus on **user-generated content**: High resolution, high diversity
- Motivation
 - Disaster management centers
 - Rescue organization
 - Planning of urgent infrastructure repair
 - ...

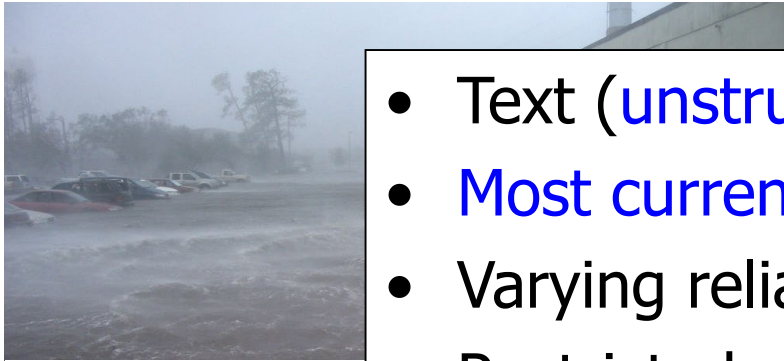
Sources: User-Created Content

URGENT info for New Orleans residents:
[Hwy 90 East at the Pearl River Bridge is closed](#). You cannot take the Highway to Mississippi, Alabama or Florida.

FlickrR, 08/28/2005, 9:48am

Water continues to rise around [our building, as it is throughout the region](#). We want to evacuate our employees and families while we are still able to safely leave our building.

Times-Picayune, 08/30/2005



During Hurricane Katrina, water reached hundreds of yards from the shore.
FlickrR, 08/29/2005

- Text ([unstructured](#))
- [Most current](#) information available
- Varying reliability and [quality](#)
- Restricted queries (full text)
- Contains images, sound, video, ...

We have received reports from NOLA.com readers that the [French Quarter is flooding](#). One reader report came from the corner of Bourbon Street and Royal Street.

nola.com, 08/30/2005

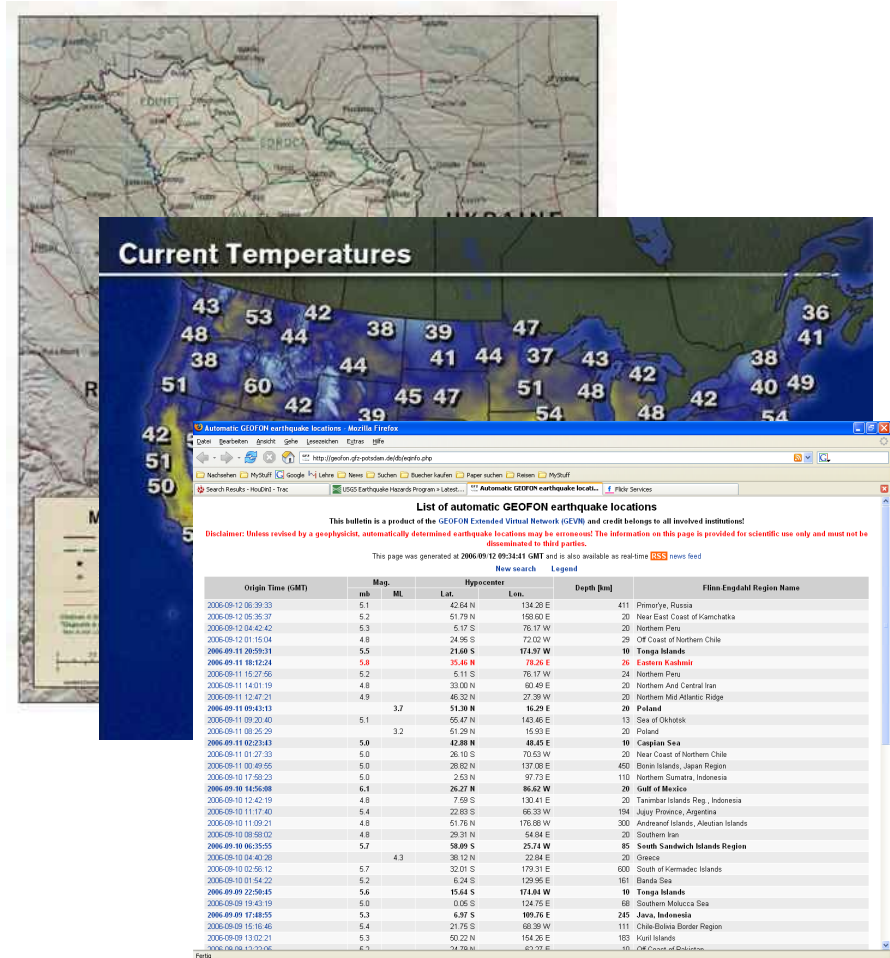


This picture was taken by [Calvin Sylvester in New Orleans](#) of his family's home in the 9th Ward, one of the areas where the pumps failed...

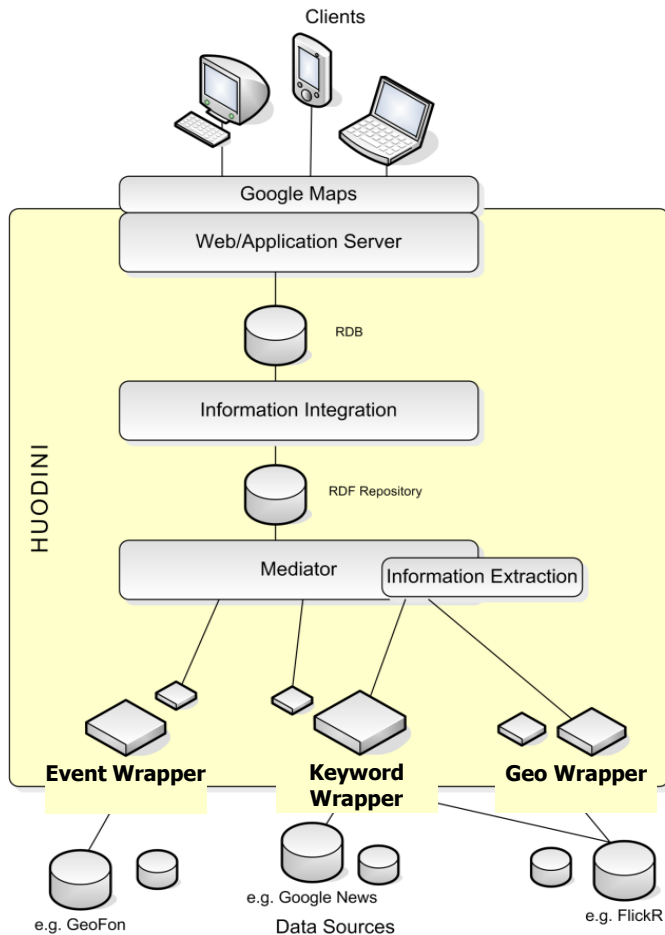
FlickrR, 08/30/2005

Sources: General Information

- Examples: earthquake data, weather, roads, buildings, hospitals, ...
- Usually **highly structured**
- Updated regularly
- Access through **web page**
- Requires web wrapping
- Restricted **query facilities**

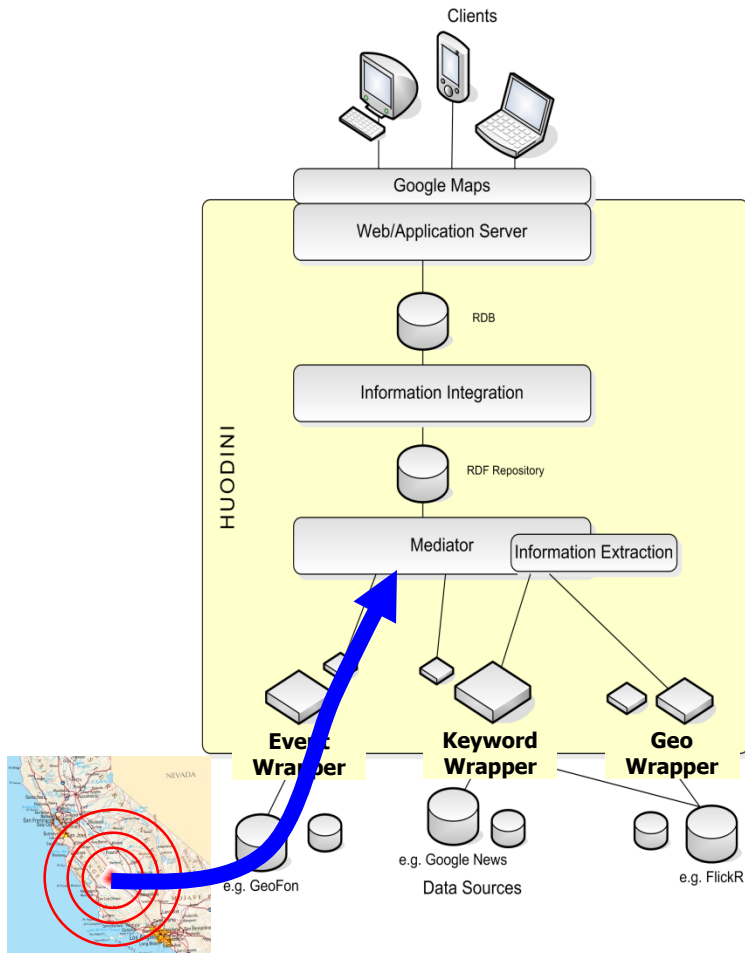


Architecture



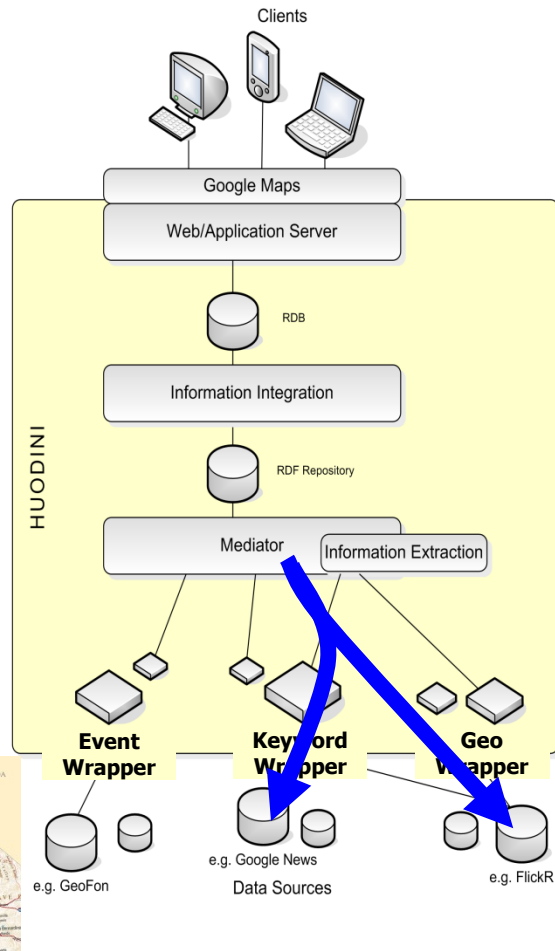
- Mediator-Wrapper architecture
 - One wrapper per **source type**
 - Implements only very simple interface
 - Highly extensible
- Information extraction
 - Parses textual information
 - **Location/time/damage**
- Flexible graph-based data model
 - RDF: Resource description framework

1. Event Detection



- Earthquakes are registered at GeoFon (Location & time)
- Event Wrapper regularly **pulls GeoFon**
- Event wrapper notices event and forwards it to mediator
- Mediator generates **wrapper queries**

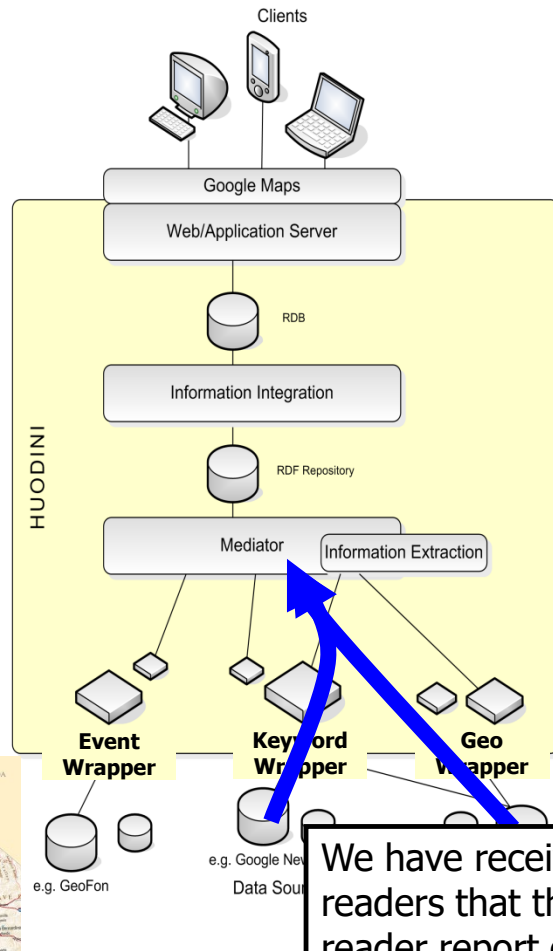
2. Information Collection



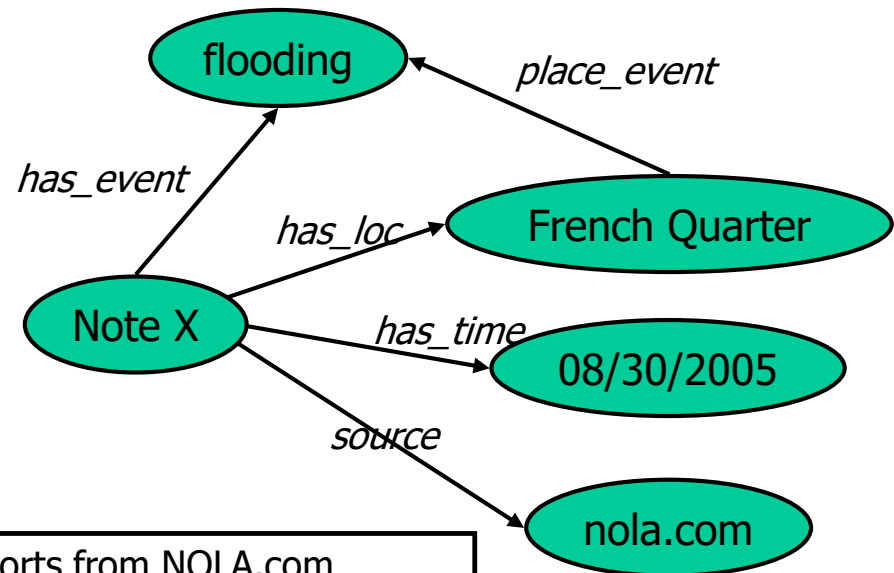
- Some sources allow SQL-like queries for location/time
- Some sources only accessible for **full text queries**
- Query generation
 - **Regions**
(from region ontology)
 - **Keywords**
(from feature ontology)

- *Los Angeles, Golden Gate Bridge, San Francisco,...*
- *Earthquake, destruction, shaking ground, electricity cut, ...*

3. Information Extraction



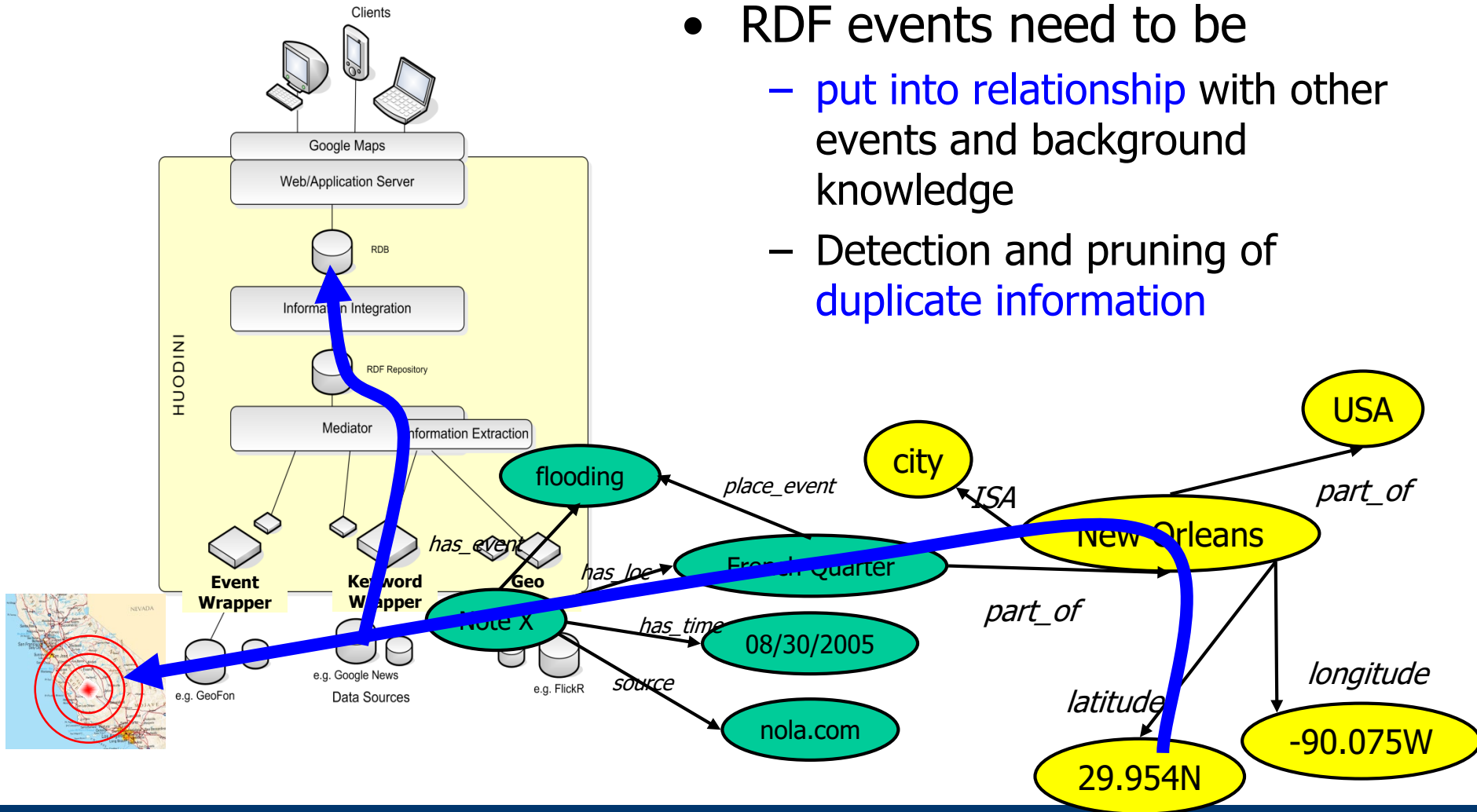
- IE module scans text for **events, location, time**
- Texts are converted into RDF



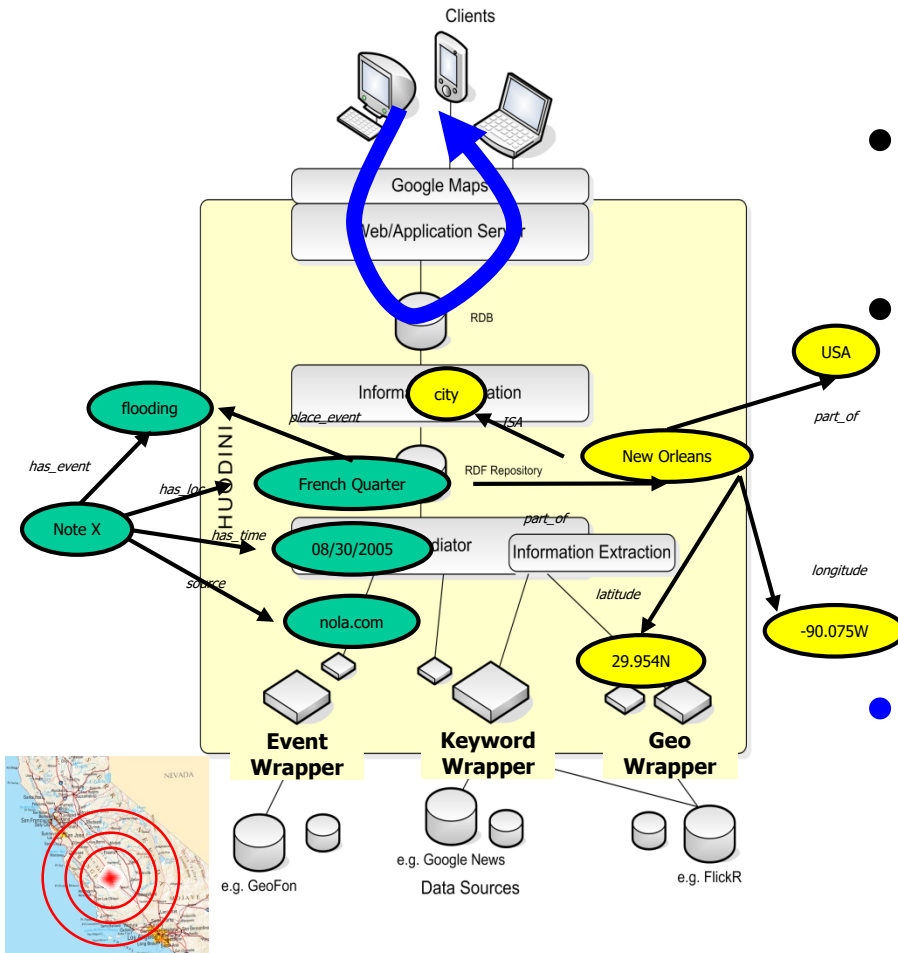
We have received reports from NOLA.com readers that the **French Quarter is flooding**. One reader report came from the corner of Bourbon Street and Royal Street.
nola.com, **08/30/2005, 8:28am**

4. Information Integration

- RDF events need to be
 - put into relationship with other events and background knowledge
 - Detection and pruning of duplicate information



5. Queries and Visualization



- Client based on Google Maps and Web2.0 techniques
- User chooses
 - Location and range
 - Types of events
 - Properties of events
 - Time period
- Detected events are shown and located on map



Approach

- **Hybrid** between materialization / virtual integration
 - Queries are answered from local, integrated repository
 - Repository is fed from queries to external data sources
 - External data sources are **pulled after events**
- **Push / pull**
 - Users pull, sources are pulled, event wrapper push (in DWH)
 - Integrated **sources don't push** – pulling is mandatory
 - Pulling has to be repeated regularly
 - Trade-off: Timeliness / workload
- **Data** integration
 - Events are reported multiple times, at many sites, in various forms
 - **Information fusion and filtering** (space, time, events, ...)