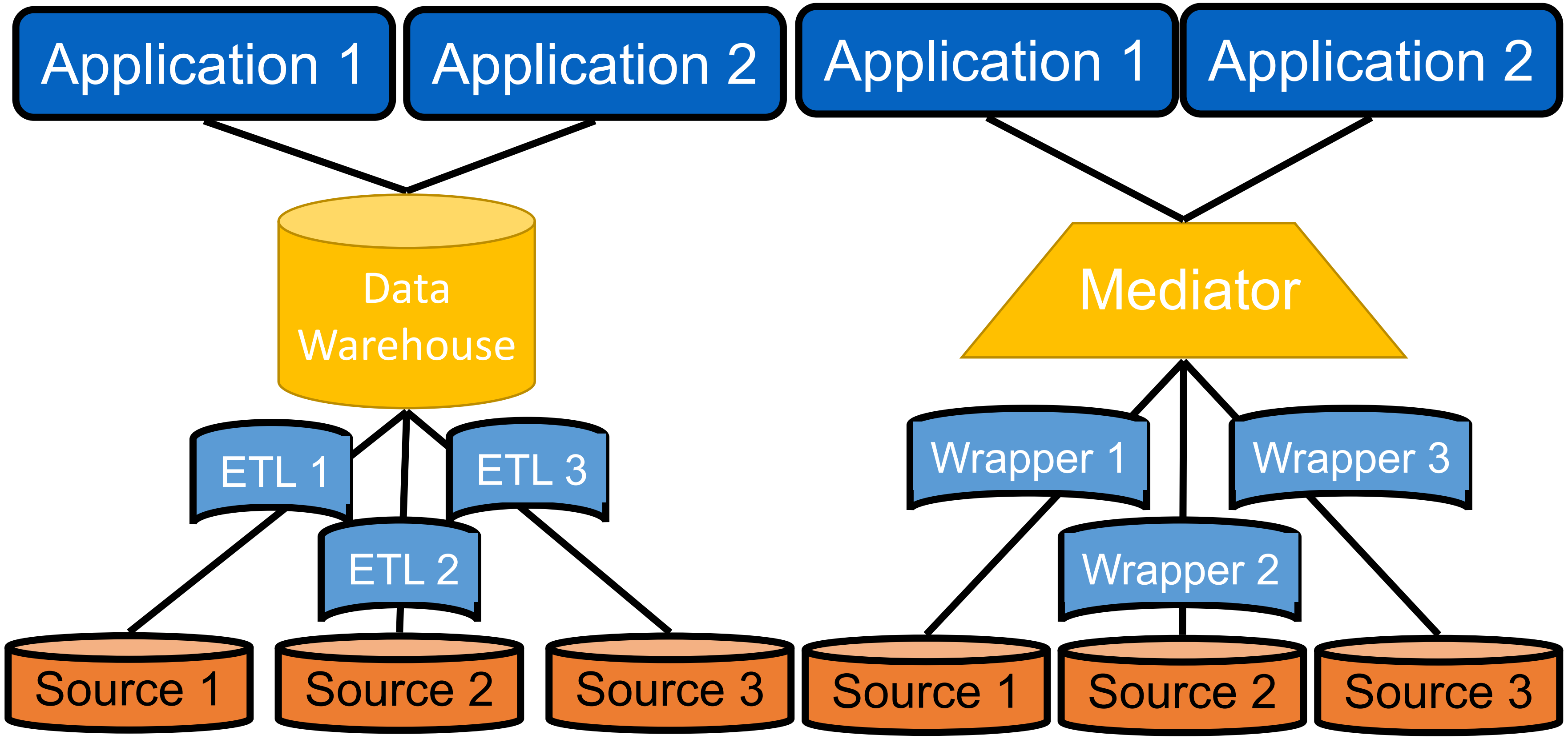
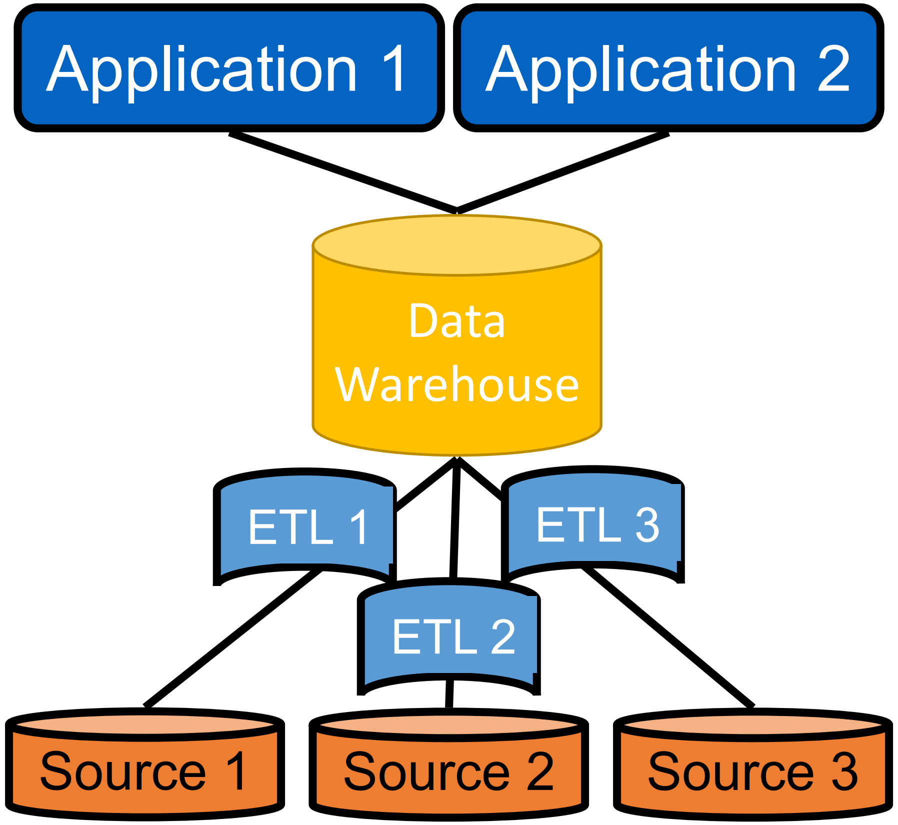
Overview

1. Data Integration Scenarios
   - Data Warehouse
   - Federated Databases
2. Materialized
   - Data Warehouse
3. Virtual
   - Mediator Wrapper System
4. **Comparison**
   - Flexibility
   - Response time
   - Currency
   - etc.

Data Warehouse vs. Mediator

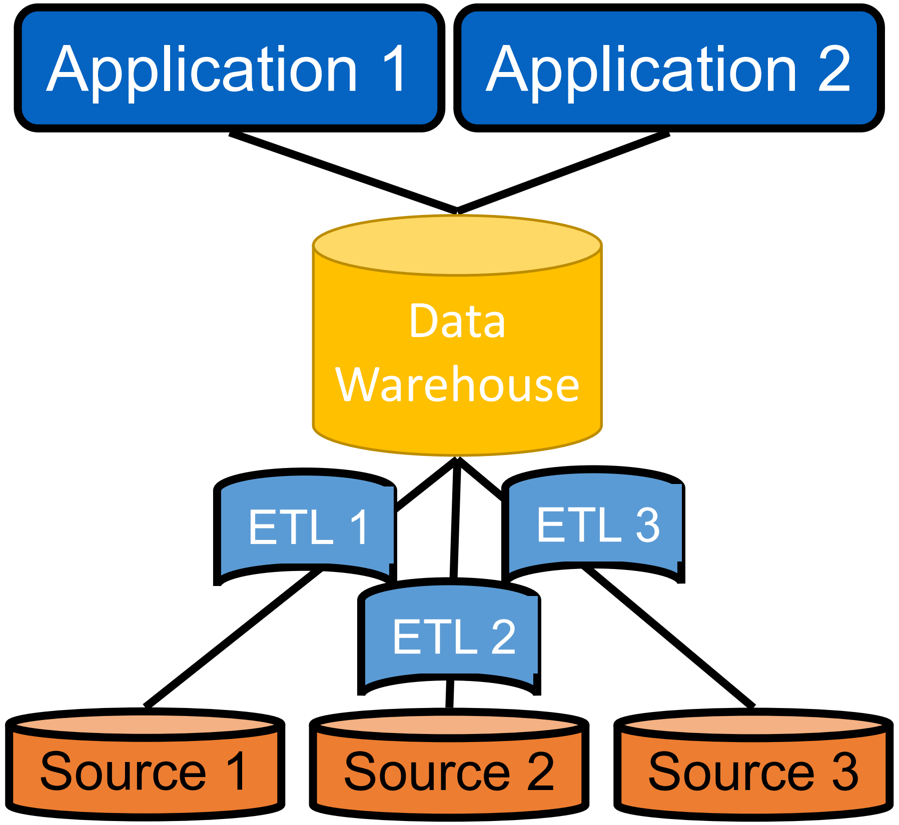


Materialized Integration – Data Flow



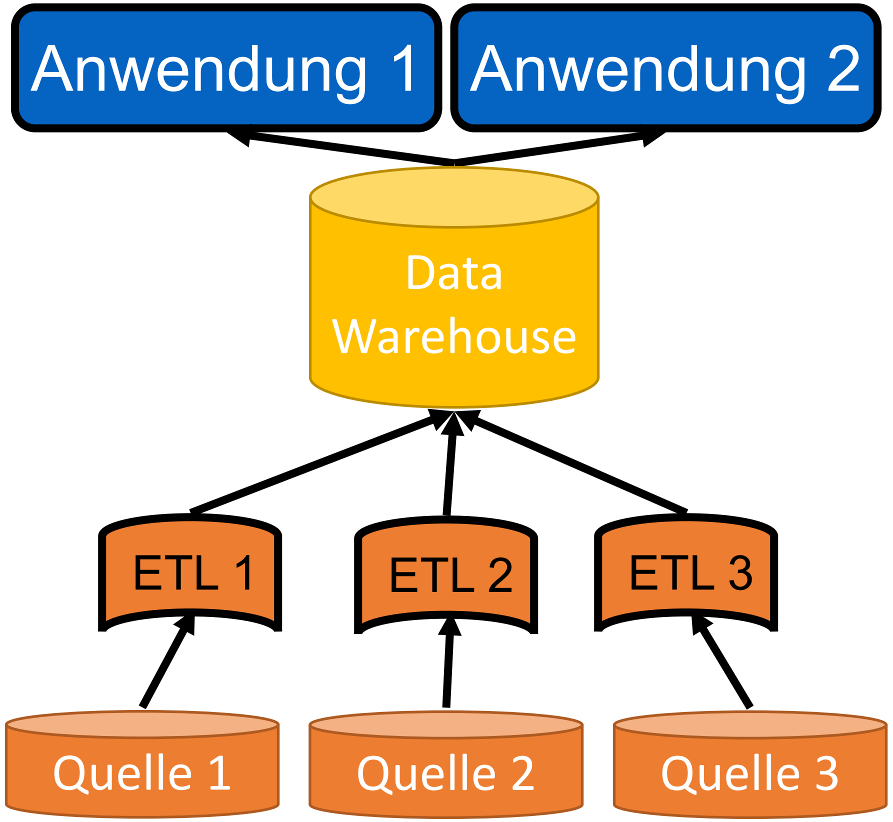
- Push
- Initial population
  - Data cleansing
- Periodical Import
  - Hourly/ daily/ weekly
  - Materialized views/ View updates
- Redundant data storage
- Aggregation und deletion of old data
  - The older the more aggregated

Materialized View – Query processing



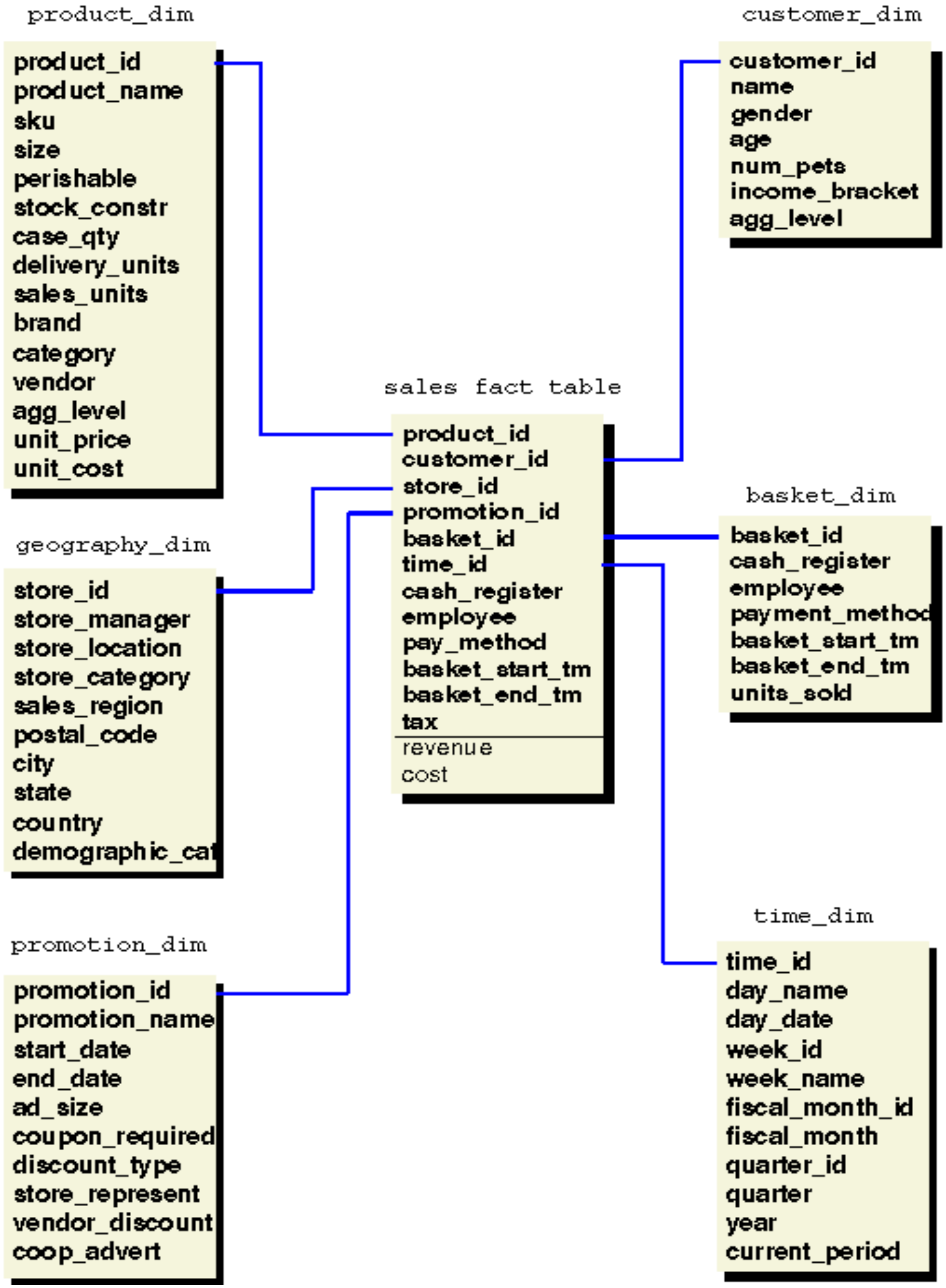
- Like „normal“ DBMS
    - Oblivious to apriori ETL
- Specials
    - Star schema
    - Aggregation
    - Decision Support

Materialized Integration – Schema



- Bottom-Up design
- Schema integration
- Star-Schema
  - *Fact-Table*
  - *Dimension Tables*

gration - Schema



product_dim
product_id
product_name
sku
size
perishable
stock_constr
case_qty
delivery_units
sales_units
brand
category
vendor
agg_level
unit_price
unit_cost

geography_dim
store_id
store_manager
store_location
store_category
sales_region
postal_code
city
state
country
demographic_cat

promotion_dim
promotion_id
promotion_name
start_date
end_date
ad_size
coupon_required
discount_type
store_represent
vendor_discount
coop_advert

sales fact table
product_id
customer_id
store_id
promotion_id
basket_id
time_id
cash_register
employee
pay_method
basket_start_tm
basket_end_tm
tax
revenue
cost

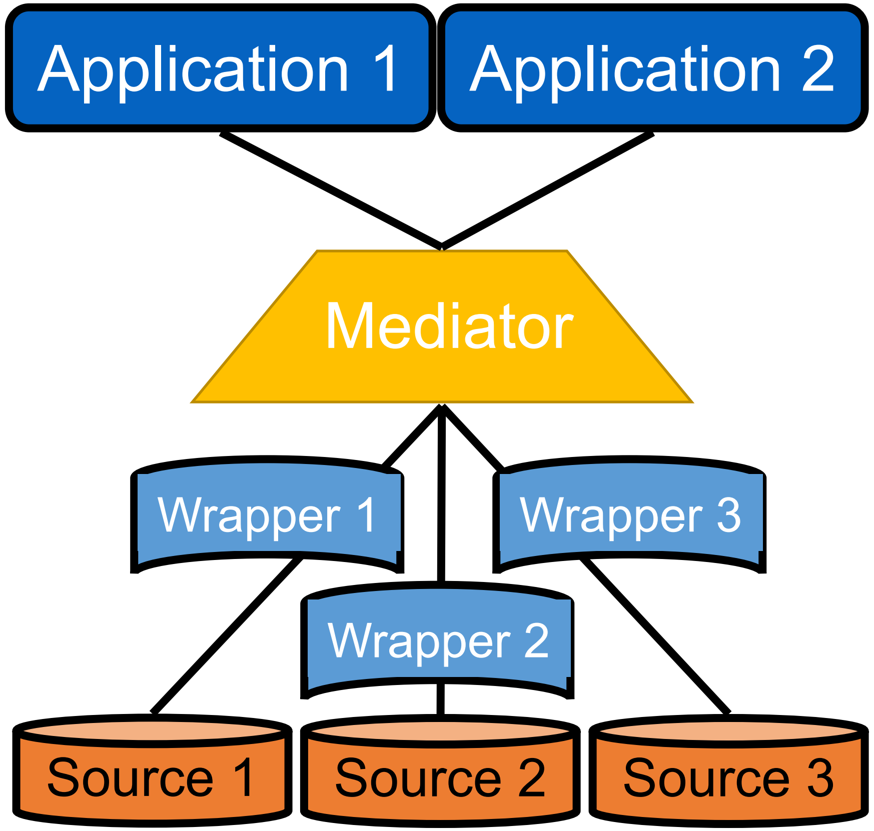
customer_dim
customer_id
name
gender
age
num_pets
income_bracket
agg_level

basket_dim
basket_id
cash_register
employee
payment_method
basket_start_tm
basket_end_tm
units_sold

time_dim
time_id
day_name
day_date
week_id
week_name
fiscal_month_id
fiscal_month
quarter_id
quarter
year
current_period

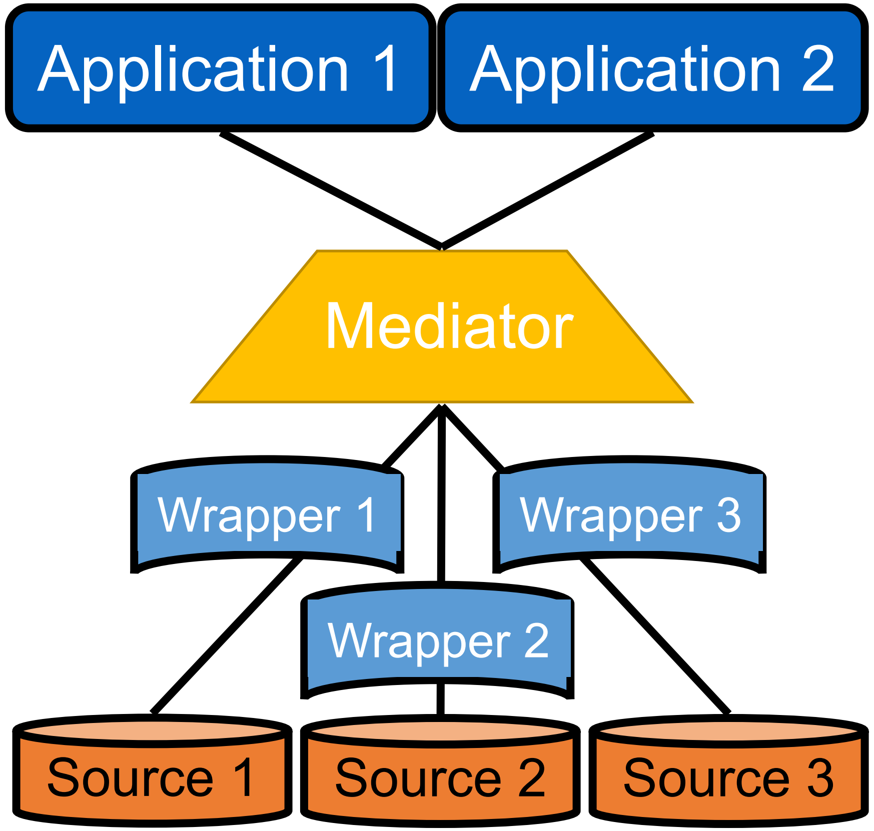
- Bottom-up design
- Schema integration
- Star schema
  - *Fact table*
  - *Dimension tables*

Virtual Integration – Data flow



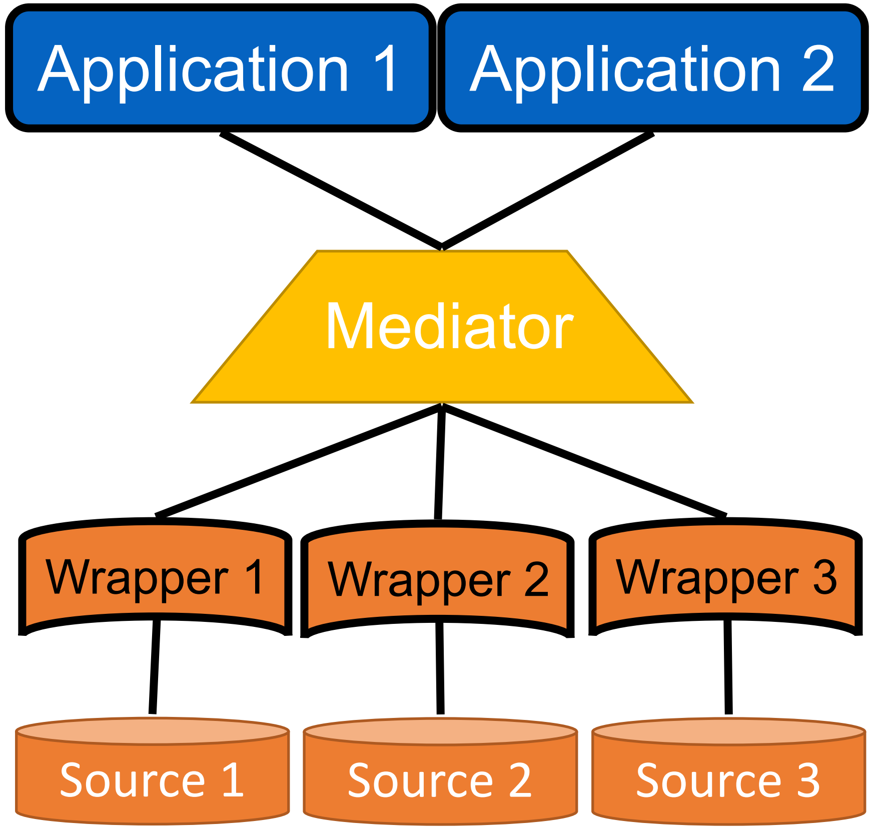
- Pull
- Data remains in sources
- Only query related data is transmitted
- Data cleansing only online (on demand)

Virtual Integration – Query Processing



- Optimization is difficult
  - Depends on sources
- Many possible plans
  - Redundant sources
  - Redundant plans
- Dynamically adapt to missing sources

Virtual Integration - Schema



- Top-Down design
- Easy to extend
  - Global: find new sources
  - Local: Only change one mapping.
- Schema mapping instead of integration (later)

Dimensions of the comparison

- Currency
- Response time
- Flexibility/ maintenance
- Complexity
- Autonomy
- Query processing / Expressiveness
- Read / Write
- Size / Storage requirements
- Resources
- Completeness
- Data cleansing
- Information quality

Currency (up-to-date-ness)

- **Materialized integration**
  - Depends on update frequency
  - In companies usually daily (over night)
  - Example SwissProt
    - Daily updates
    - But releases are monthly

- **Virtual integration**
  - Always up-to-date
  - Solely depends on currency of autonomous systems
  - Sometimes: caching

Response Time

- **Materialized integration**
  - Pretty good
  - Local access
  - Similar to DBMS
    - Optimization
    - Materialized views
    - Indices
    - ...
  - Usually queries are very complex

- **Virtual integration**
  - Problematic
  - Data is in remote autonomous DBMS
    - Transmission through network
  - Source response time
  - Hard to optimize
  - Complex operations have to be carried out in a naïve manner
  - Data cleansing has to be applied during query time or afterwards

Flexibility / Maintenance

- **Materialized integration**
  - Hard
  - Removing/ Updating/ Adding of sources can affect the whole integration
  - Local maintenance of a growing huge databases
    - With Indices etc.
  - Daily integration is needed

- **Virtual integration**
  - Easier
  - Removing/ Updating/ Adding of sources can affect only the specific source
  - Sources have to perform maintenance on their own
    - Backups, DBMS maintenance etc.

Complexity

- **Materialized integration**
  - Like DBMS
  - Complex queries
  - Query planning is easy (global as view)
  - Sources are often similar to each other
    - Often they are DBMS

- **Virtual integration**
  - Modelling sources is important
    - Expressiveness of sources
  - Query planning is hard (local as view)
  - Often very different sources
    - Web services
    - HTML forms
    - Flat files
    - ...

Autonomy

- **Materialized integration**
  - Sources less autonomous
    - No communication autonomy
    - Low execution autonomy
    - Low design autonomy
  - Must allow bulk-read
  - Update notifications

- **Virtual integration**
  - Sources very autonomous
  - Full design autonomy
  - Nearly full communication autonomy
    - Some communication is necessary otherwise system cannot be part of IIS
  - Nearly full execution autonomy
    - Only: Queries have to be answered at some point

Query planning / Expressiveness

- **Materialized integration**
  - Query planning similar to a DBMS
  - Expressiveness like a global system
    - E.g., Full SQL expressiveness

- **Virtual integration**
  - Query planning is complex
    - Distribution
    - Autonomy
    - Heterogeneity
  - Limited expressiveness has to be compensated on global level
  - But also: Special expressiveness of sources can be exploited:
    - Image retrieval
    - Text index

Read / Write

- **Materialized integration**
  - Read is always possible
  - DW: Write often not allowed but possible
    - Can lead to inconsistencies with sources

- **Virtual integration**
  - Read is often possible
  - Availability!
  - Write often not possible
    - In terms of redundancy: Where to write
    - Transactions are hard
    - Autonomy

Size / Memory consumption

- **Materialized integration**
  - High
    - Redundant data storage
    - DW: Historical data
  - Growth
    - Continuous
  - Footprint: like DBMS

- **Virtual integration**
  - Low
    - Meta data
    - Cache
    - Intermediate results
  - Footprint: like DBMS

Resource Consumption

- **Materialized integration**
  - Network load can be predicted
  - All data is being transmitted
    - Depending on query
    - Aggregation
    - Pre-Aggregation

- **Virtual integration**
  - Potentially high network load
  - Data is transmitted multiple times
    - Cache can help.
  - Only needed data is transmitted

Je nach *Workload*.
Spannendes Optimierungsproblem!

Completeness

- **Materialized integration**
  - Good
  - Assumption: Materialization is complete

- **Virtual integration**
  - Only when all sources are available
  - Query can be left unanswered or partly answered
    - Fuzzy query semantic:
      - All tuples?
      - All attributes?
  - Definition of completeness
    - Open World Assumption
    - Closed World Assumption

Data Cleansing

- **Materialized integration**
  - Many methods
    - Still tedious
  - Offline (during night)

- **Virtual integration**
  - Online cleansing is hard
    - tedious
    - No expert sourcing is possible

Information Quality

- **Materialized integration**
  - High
  - Verified
  - Can be improved over time

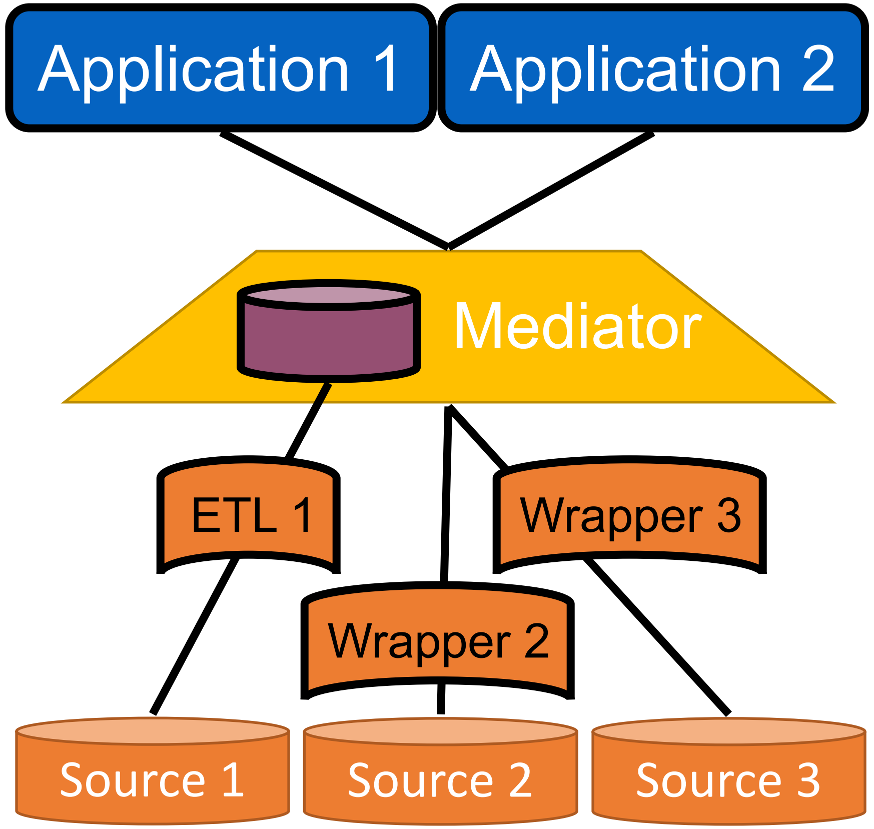
- **Virtual integration**
  - Depends on sources
  - Often problematic
    - Autonomy

Summary

	Materialized	Virtual
Currency	- (Cache)	+
Response time	+	-
Flexibility	- (GaV)	+ (LaV)
Complex query planning	-	--
Source autonomy	-	+
Expressiveness	+	-
Read/Write	+/+	+/-
Size	-	+
Resource consumption	? (workload)	? (workload)
Completeness	+	? (OWA, CWA)
Data cleaning	+	-
Information quality	+	-

Hybrid Solution

- Subset of the data can be materialized
  - □ Popular subsets (cache)
  - □ Data that is available as bulks
    - – Dump Files
    - – SQL access
    - – ...
- Subset has to stay in the sources
  - □ Often updated data
  - □ Data with limited access
    - – At least one bound variable
    - – Limited licenses
- Optimization prefers local data
  - □ Checking whether data up-to-date



Overview

1. **Data Integration Scenarios**
   - Data Warehouse
   - Federated Databases
2. **Materialized**
   - Data Warehouse
3. Virtual
   - Mediator Wrapper System
4. **Comparison**
   - Flexibility
   - Response time
   - Currency
   - etc.

Literature

- [BKLW99] Busse, Kutsche, Leser, Weber, Federated Information Systems: Concepts, Terminology and Architectures. Forschungsbericht 99-9 des FB Informatik der TU Berlin, 1999.
  Online: http://www.informatik.hu-berlin.de/~leser/publications/tr_terminology.ps
- [DD99] Ruxandra Domenig, Klaus R. Dittrich: An Overview and Classification of Mediated Query Systems. SIGMOD Record 28(3): 63-72 (1999)