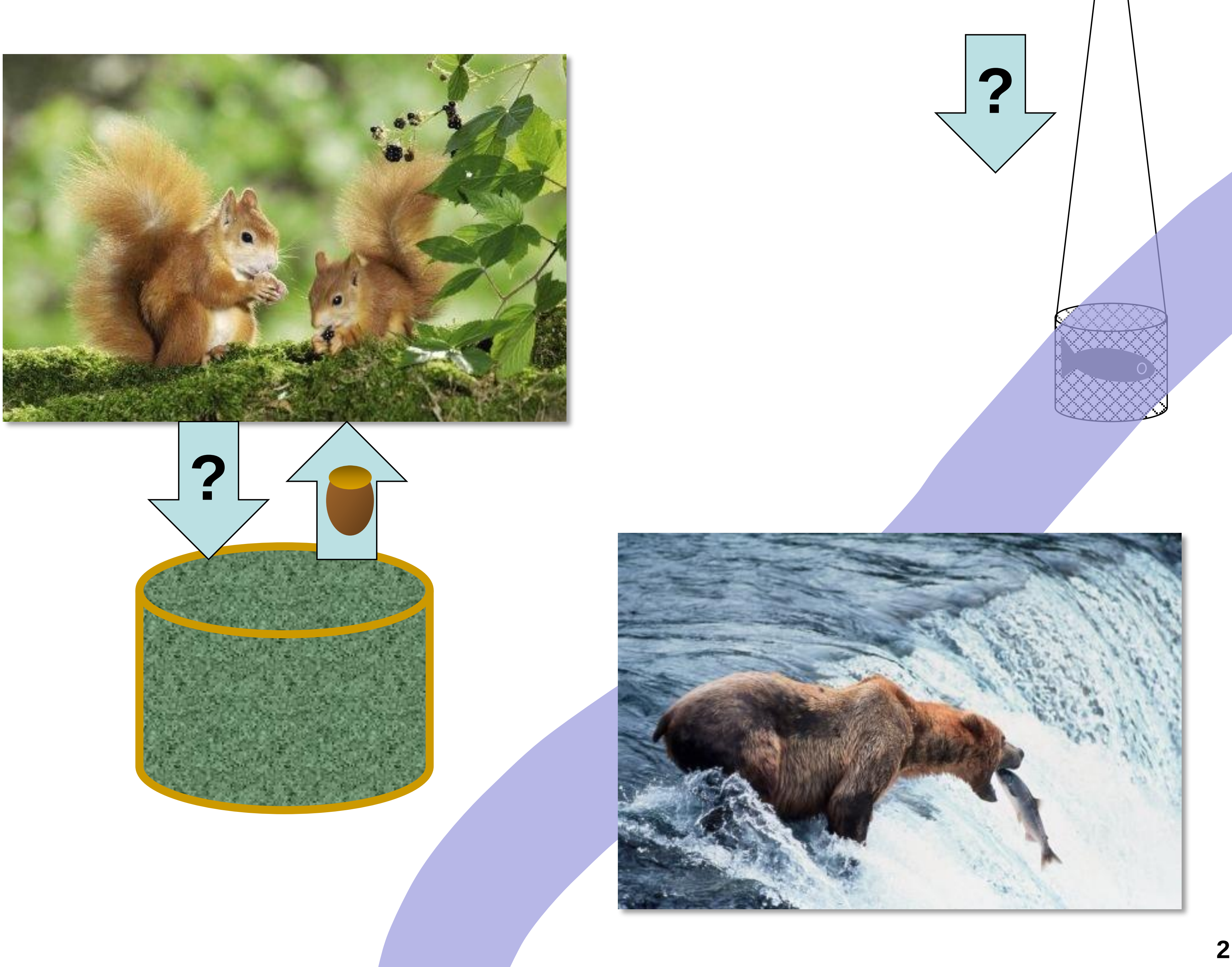
Issues in combined static and dynamic data management

DATA conference, 25-27 July, 2012

Daniela Nicklas

Department für Informatik
Database and Internet Technologies
Carl von Ossietzky Unversität Oldenburg
Germany

dnicklas@acm.org



Overview

- Introduction
- Some existing approaches
  - for dynamic data management
  - for static data management
- Federated Architecture
- Issues
- Outlook

Issues in combined static and dynamic data management

DATA conference, 25-27 July, 2012

Application Areas (... of my current interest)

Crew

Congestion

Snow

Pedestrians

Wind

Goods

Cargo

Trucks

Processes

Safe Off-shore Operations

Traffic monitoring

Logistics

Parcels

Sea

Ships

Obstacles

Application Areas

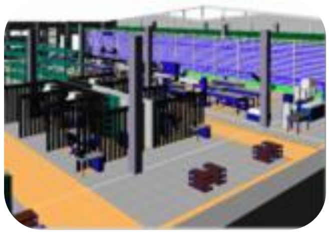
Vessel Traffic Management

Ground

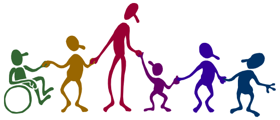
Driver assistant systems

Lanes

Products



Smart Factory



...

... and many more!

Cars

Workers

Machines

Roads

Example Application Domain: Safe Offshore Operations

- Goal: Improve the safety of offshore operation for people, machines, and processes (4 year project, more @ http://soop.offis.de)

Source: Bard Offshore

Static and Dynamic Data Management …

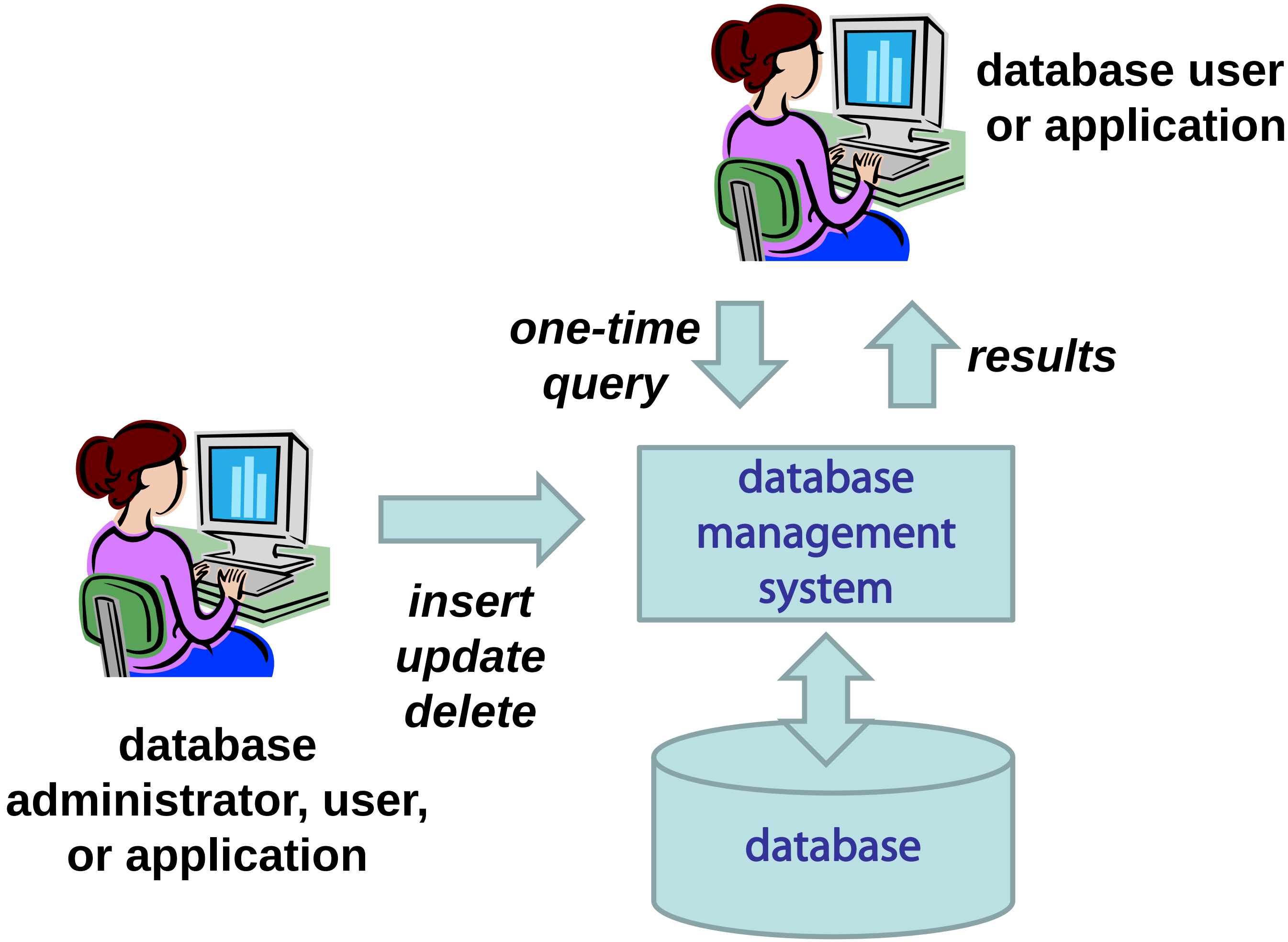
Data Base Management Systems

- „store, then process“
- analytical queries
- one-time queries
- well-known technology
- user/application is active, data management passive

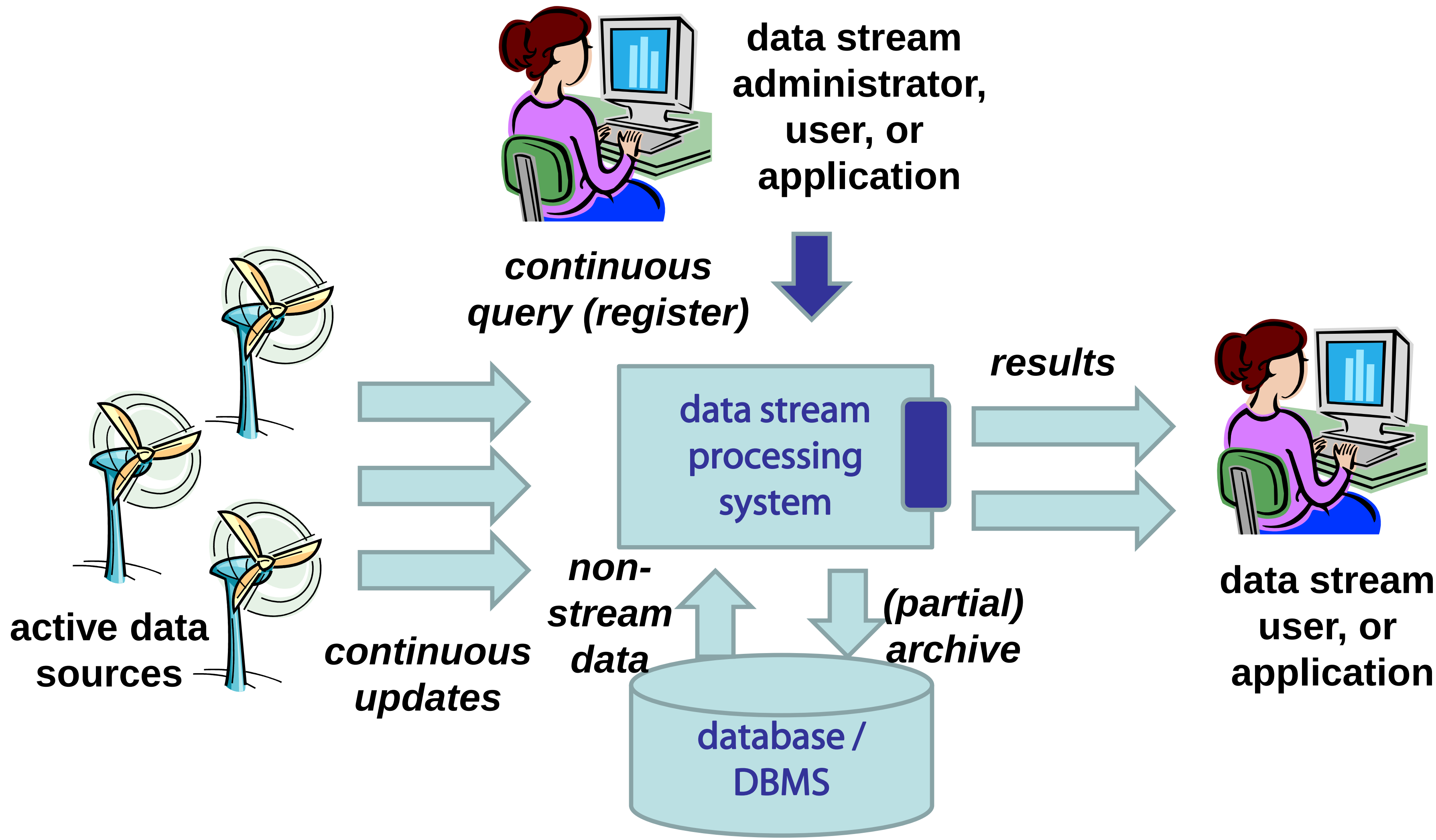
Data Stream Management Systems

- „on-the-fly“ data processing
- real-time reaction on changes
- store only what needs to be kept
- register long-running queries
- user/application is passive, data management pushes data
- Complex event processing for temporal patterns

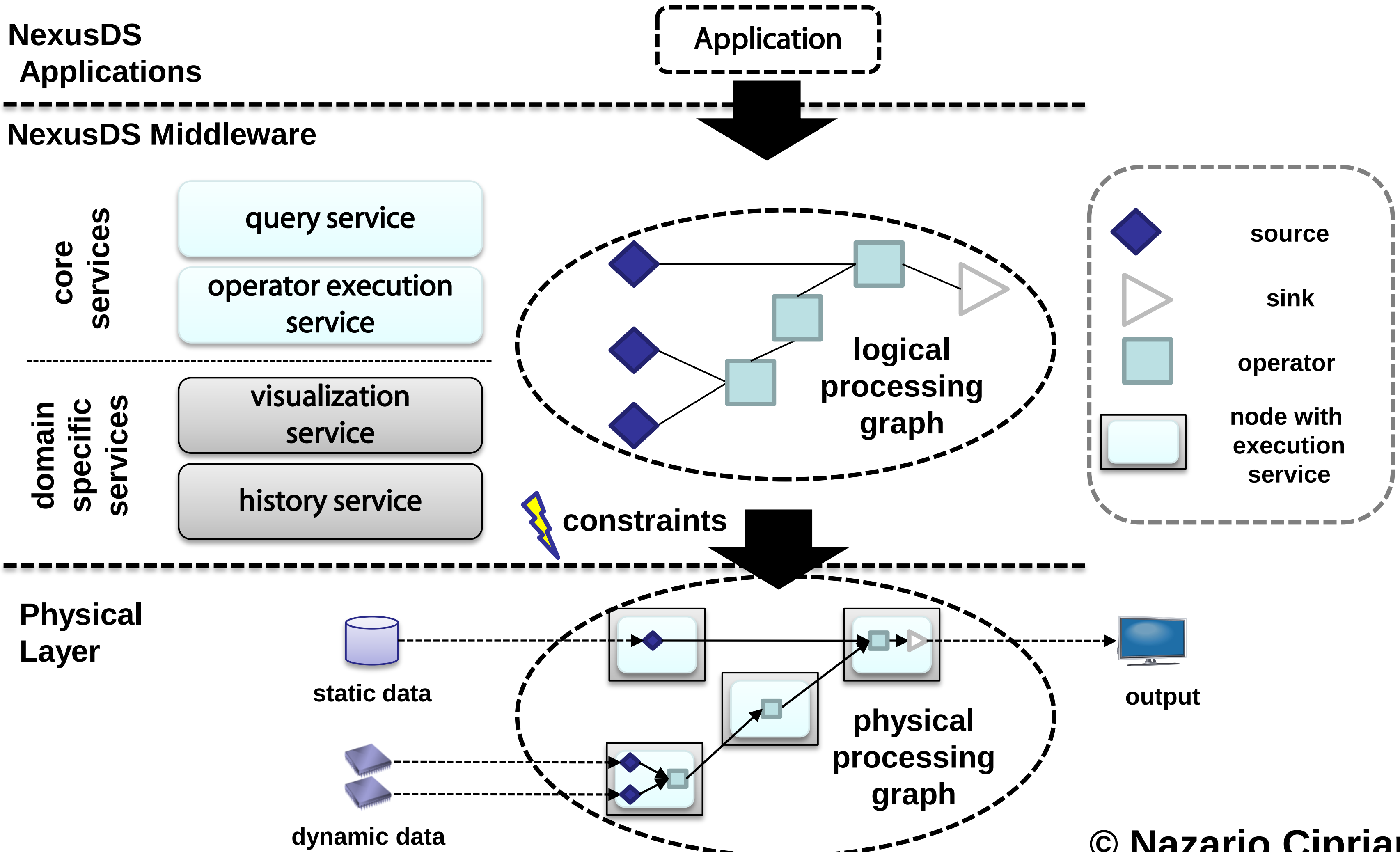
Typical queries in database management systems



Typical queries in data stream processing systems



Example: NexusDS



see Friday 27, Session 4, 11:30 - 13:00 Trust, Privacy and Security.

NexusDSS: A System for Security Compliant Processing of Data Streams

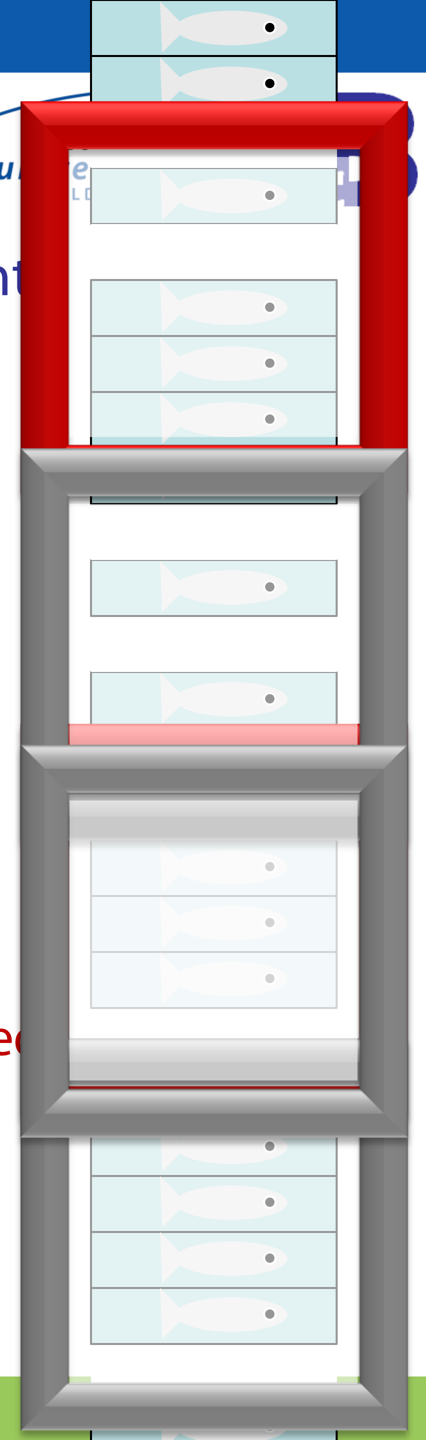
DSM – Challenges

- Data stream
  - Continuously arriving data
  - Potential infinite
- Challenges
  - Blocking behaviour of query operators not adequate for stream processing
    - E.g., how to compute the average?

    → unblocking by windows
  - Temporal relationship of streaming data

Window definitions

- Monotone window operator to split stream into segment
- Window size can be based on:
  - number of elements (e.g., last 100 elements)
  - time (e.g., last 5 seconds)
  - predicates of elements (e.g., value between two threashholds)
- Window stride: how it moves
  - jumping or tumbling (no overlaps, data processed once)
  - sliding (continously, overlaps, data processed more than once)
  - sampling (no overlaps but gaps, some data is not processe
- Window realization (next slide)
  - interval appraoch
  - positive/negative approach



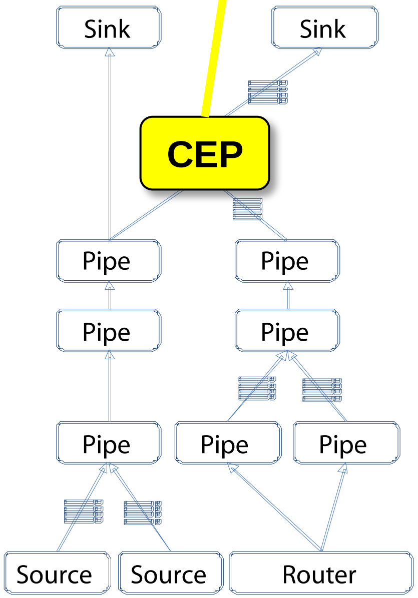
dnicklas@acm.org

Features of Data Stream Management Systems

- Programming Abstraction
  - declarative: query
  - functional: flow graph
  - → enables optimizations
  - → better maintanance of systems
  - → using a DSMS on data streams is like using a DBMS instead of files
- Data flow vs. event bus (as in many CEP engines)
  - execution of data flow processes only data items that are needed by a query
- Parallel execution of operators in graph
  → no shared memory
- Data streams can be unbounded:
  - issues with sorting, joins, aggregation
  - → approximate answers
  - → window semantics

```
SELECT ego.pos
RANGE 10 secon
radar RANGE 15
WHERE ego.speed > 30 AND
radar.speed > 30
AND s2.pos - ego.pos < 15
```

Some DSMS provide CEP operators



However …

Data Base Management Systems

- ACID properties, persistant storage
- well-established semantics (rel. algebra)
- well-established optimizers
- often deeply integrated into enterprise IT infrastructure
- can also cope with dynamic management (triggers, repeated optimized queries)

Data Stream Management Systems

- main memory processing
- often non-deterministic results (depend on timing of streams)
- no standard query semantics (often relational algebra with extras, or many user-defined operators)
- new system in IT infrastructure
- has to be combined with static data management

Can‘t we do that with a DBMS?

- Active Data Management in DBMS (Triggers, CQN)
- Magic Updates: The Bonn Approach

Dynamic Data in DBMS

- Database triggers
  - Are designed to intercept database change requests and transaction requests
  - Are part of a transaction – synchronous
  - Allow to check validity of changes
  - Widely used in applications
  - But: since triggers can trigger triggers, complex logic gets hard to maintain
- Database Log Miner (e.g., Oracle)
  - The log miner is designed to select specific journal entries in batch or real time mode
    - Works asynchronously
    - Shows 'dirty' data
    - Works closely with dissemination services such as Oracle Streams and Oracle GoldenGate
    - Used by many infrastructure services such as replication, logical standby, real time warehousing, security auditing, real time analytics through TimesTen (In Memory Database), CEP

Continuous Query Notification - CQN

- CQN is designed to notify users/applications if the result set of a query has changed
  - Requests are defined as registered queries – normal query plus a notification destination
  - Works asynchronously
  - Shows only ‘visible’ data

- The following events generate notifications:
  - Committed DML Transactions
  - Committed DDL Statements
  - Deregistration
  - Global Events

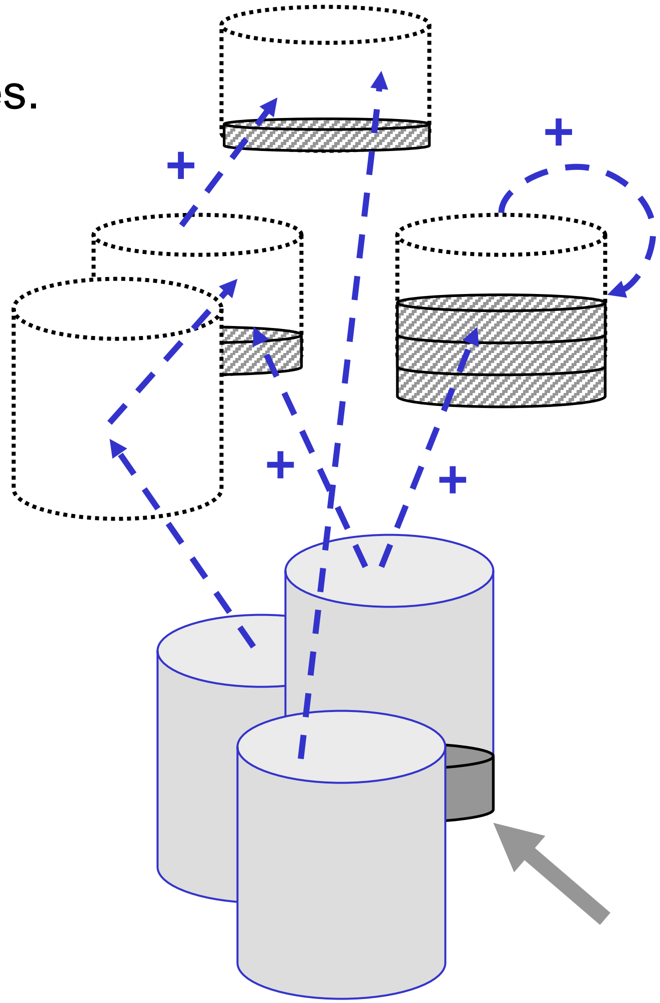
Continuous queries 'made' in Bonn:

- We‘re experts for update propagation techniques.
- A <u>considerable</u> amount of stream applications can be realized using conventional DBS.
- Continuous queries are implemented using materialized views.
- Incremental techniques are used for their maintenance.
- In order to process

  <u>static</u> domain knowledge with

  <u>dynamic</u> stream data,

  we employ Magic Updates!



universitätbonn

Data Stream research today:

- It is widely believed that conventional DBS are <u>not</u> suited for processing continuous queries.

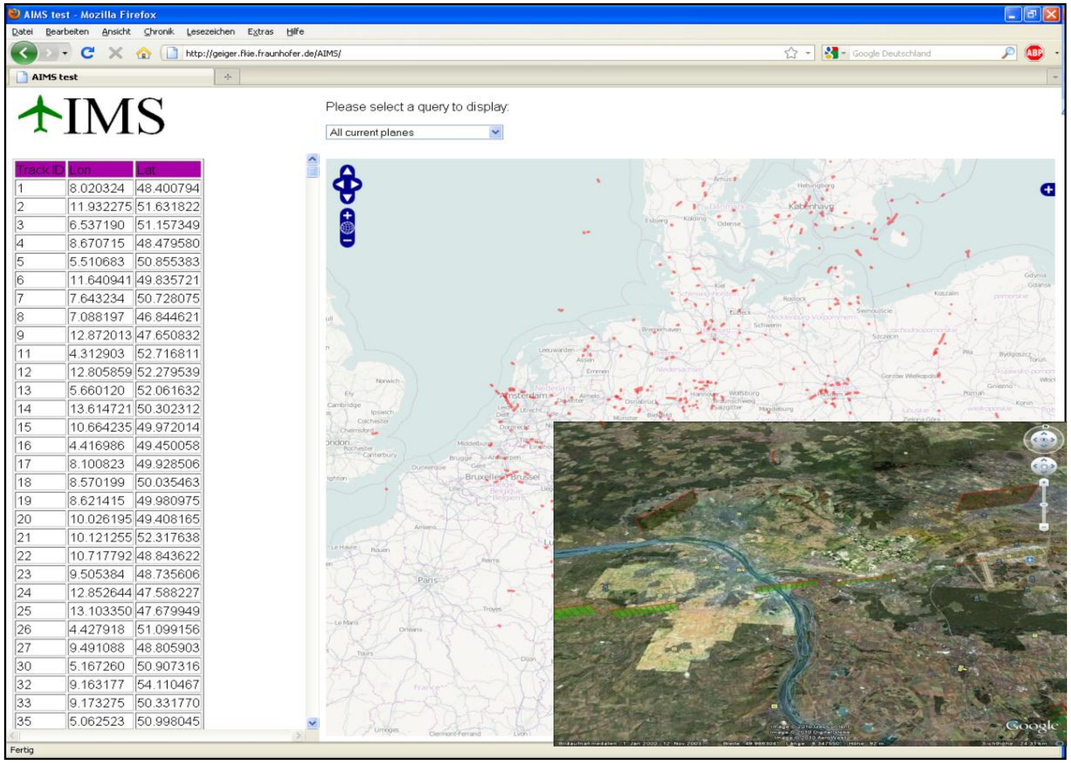
- Instead, main memory-based DSMS are employed:
  - no transactions (no recovery, no rollback,…)
  - Limited multi-user access
  - no history recorded
  - etc.

    $\Rightarrow$ Basically, only SQL survived

- DSMS can process fast stream data but static domain knowledge only in a limited way.

universitätbonn

Case study: View-based analysis of aircraft movements using the Airspace Monitoring System (AIMS)

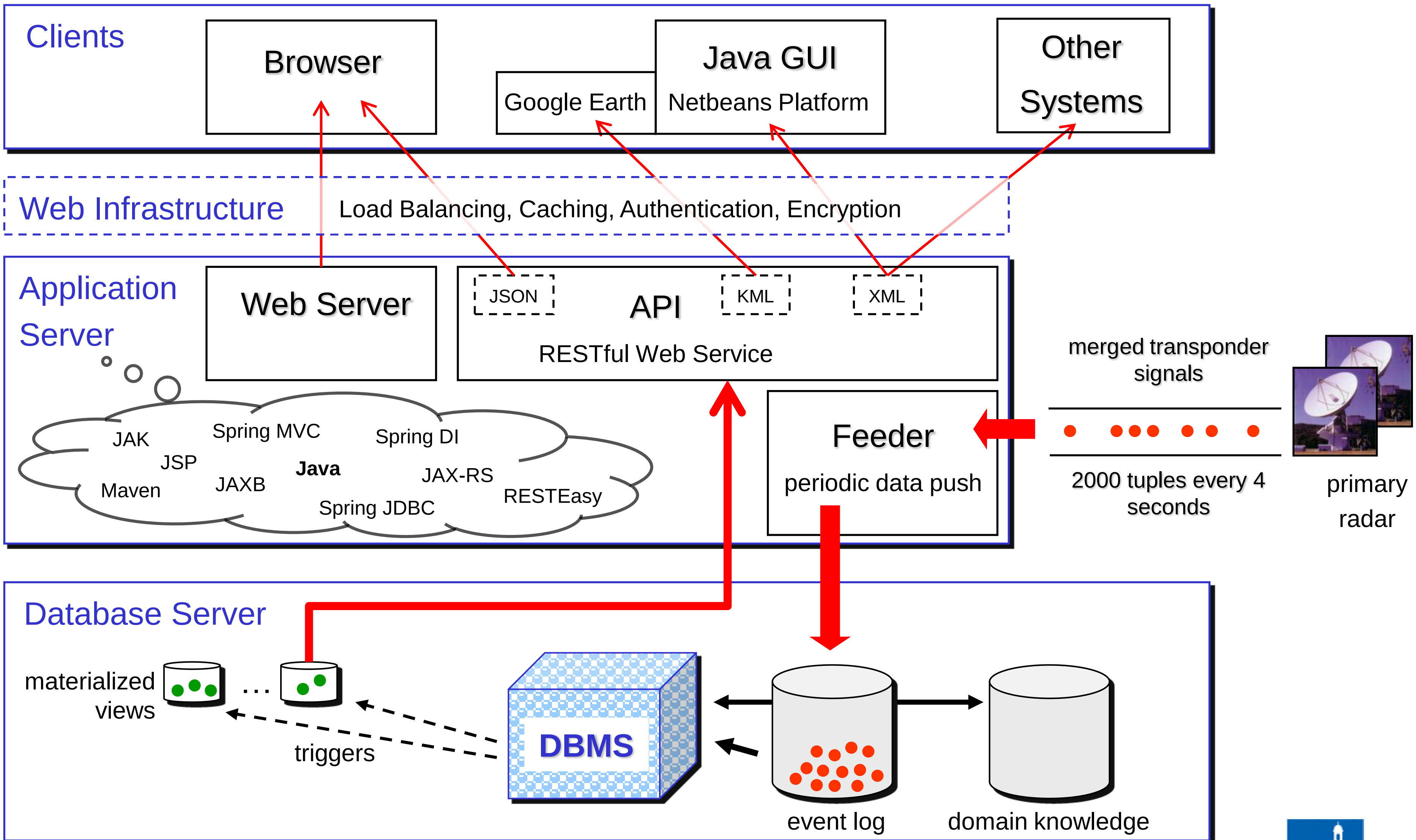


- Which aircrafts approach each other critically?
- Which aircrafts are currently landing?
- Which planes are late?
- Which flight are entering areas with bad weather conditions?
- Which planes are changing their course unexpectedly?
- What is the average number of flights for an arbitrarily chosen area?

AIMS prototype in cooperation with CASSIDIAN AN EADS COMPANY

universitätbonn

Architecture of AIMS



dynamic track data

static domain data

Flights

- Call sign of a flight
- Aircraft type
- Registration
- Transponder code
- Position by latitude, longitude, altitude
- Vertical speed
- Horizontal speed
- Heading of the aircraft in degrees
- Time stamp
- …

Airports

- Name of airport, city, country
- IATA / ICAO-Code
- Position by latitude, longitude, altitude
- Region code like “E” for Europe
- …

Flight plan

- Waypoints: beacon-code, position, speed, time
- Plan ID, source, country code
- Callsign, aircraft type, destination airport
- …

An aircraft is considered to be landing if

- its vertical velocity is negative
- its flight level is below zero
- it is currently approaching an airport less than 20 miles away.

```
CREATE VIEW landeAnflug AS
 SELECT f.*
 FROM  flight AS f
 WHERE f.Incline < 0 AND
       f.altitude < 'FL0' AND
       EXISTS( SELECT *
               FROM  airport AS a
               WHERE getDist(f.Lon,f.Lat,a.Lon,a.Lat) < 20000)
```

Aircraft positions frequently change:

CallSign	Type	Longitude	Latitude	…	Date
ACA876	B763	-4,27333	52,50833	…	23-03-2007 **14:17:03**
ACA876	B763	-4,82512	51,33	…	23-03-2007 **14:17:07**
ACA876	B763	-5,92113	51,1211	…	23-03-2007 **14:17:11**
…	…	…			…

t

Time stamped data stream

<u>CQ</u>: Which aircrafts are currently landing?

landed

Currently landing

Starting to land

Landing state remains relatively stable!

Motivation for

⇒ incremental evaluation

universitätbonn

CS	Time
…	**…**
ACA873	**14:17:03**
ACA876	**14:17:05**

+

CS	Time
BAW069	**14:17:08**

new derivation

```
CREATE MATERIALIZED VIEW landeAnflug AS
  SELECT f.CallSign, f.Time
  FROM flight AS f
  WHERE f.Incline < 0 AND
        …
```

```
INSERT INTO landeAnflug
 SELECT nf.CallSign, nf.Time
  FROM  new_flight_data AS nf
  WHERE nf.Incline < 0 AND
              …
```

CS	Longitude	Latitude	…	Time
…	…	…	**…**	**…**
ACA873	-4,556667	52,578333	**…**	**14:17:03**
ACA876	-4,273333	52,508333	**…**	**14:17:05**
AFR1341	-4,803333	52,18333	**…**	**14:17:08**

Flight data

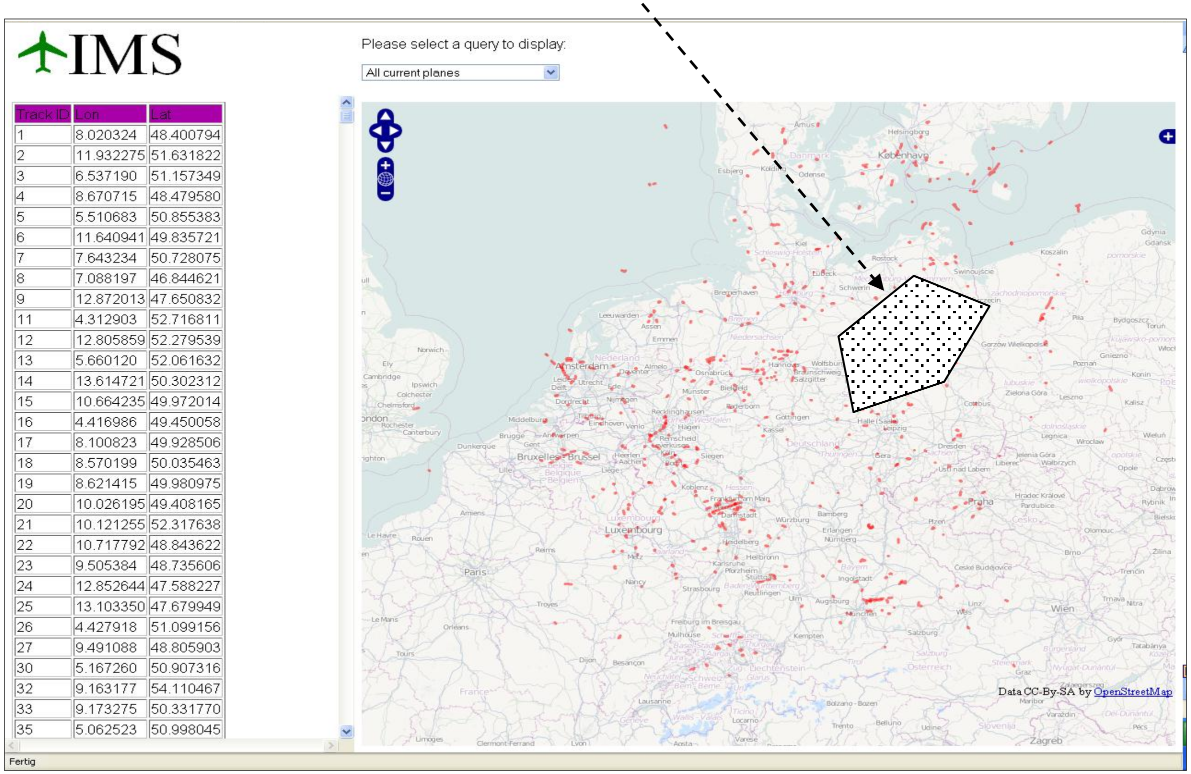
CS	Longitude	Latitude	…	Time
BAW069	**1,521667**	**51,35254**	**…**	**14:17:08**

new data

universitätbonn

Performance achieved so far with AIMS:

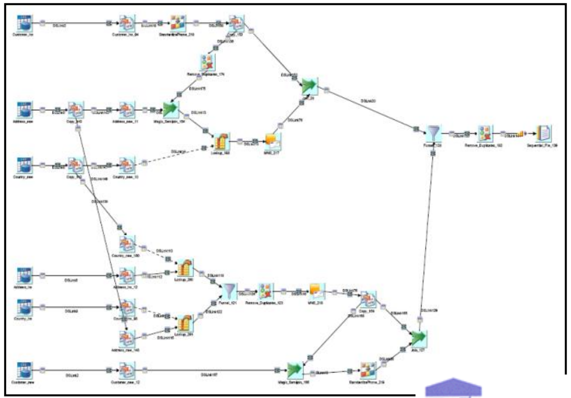
- AIMS typically processes 2000 tupels (20 attributes) every 3-4 seconds
- Determination of landing/starting planes <1 second
- Determination of critical approximations <1 second
- Counting of flights within a user-defined region <2 seconds



universitätbonn

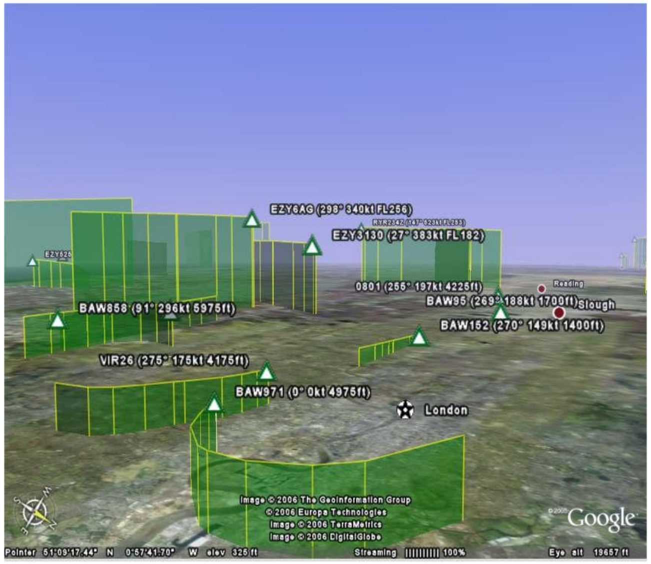
Update propagation is suited for
analyzing a wide spectrum
of stream applications.

Optimizing ETL jobs



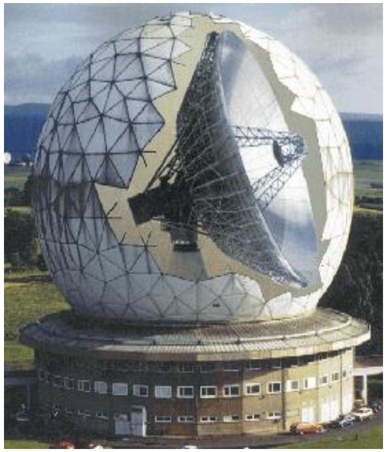
In cooperation with
FACHBEREICH INFORMATIK

Air Traffic Control



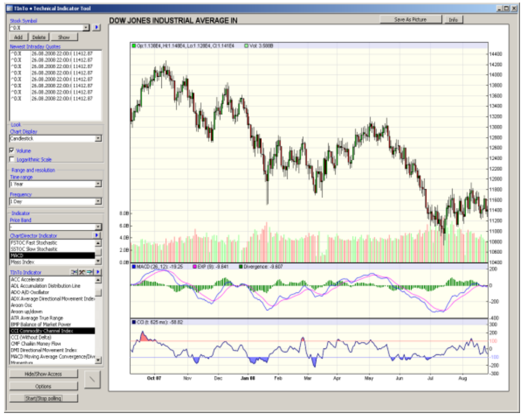
In cooperation with
CASSIDIAN
AN EADS COMPANY

Stochastic Analysis of
Radar Data



In cooperation with
Fraunhofer
FKIE

Analyzing Stock Market Data



universitätbonn

DBMS approaches assessment

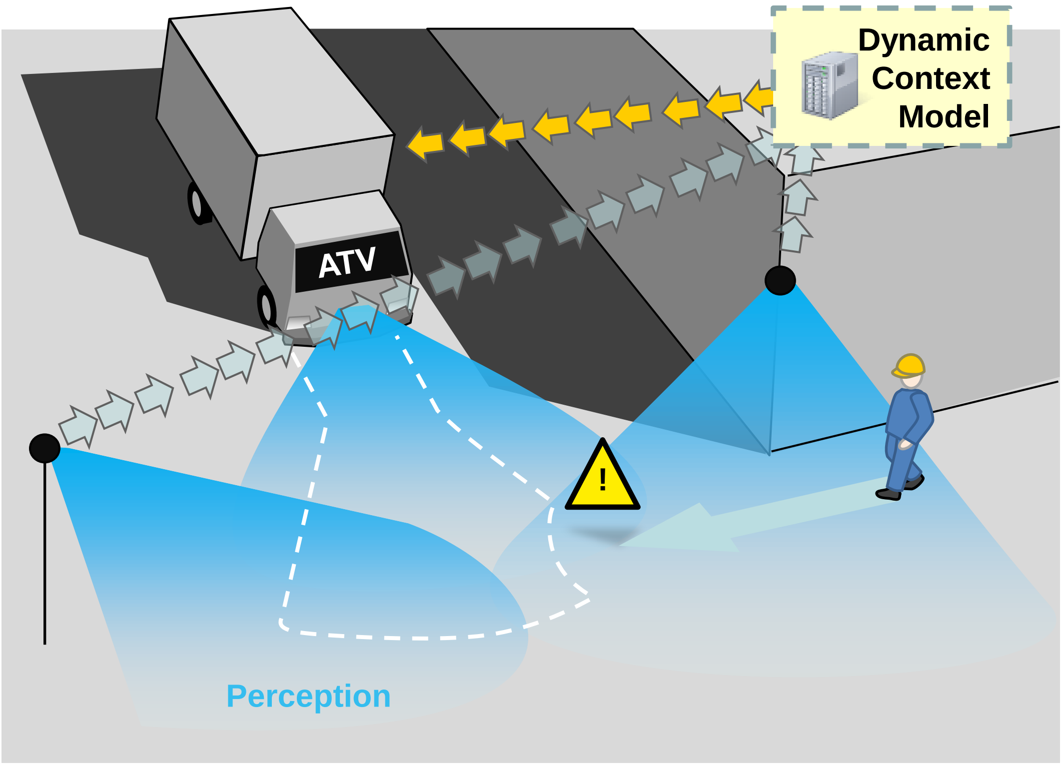
- Consider updates and the change of queries
- Are good up to a couple thousand of tuples/second

- However, there are application that need much higher throughput
  - real-time multi-sensor fusion
  - dynamic context models

Sensor Fusion Example (project SaLsA)

- Context-Model Generation for Safe Autonomous Transport Vehicles
  - Autonomic vehicles perform mobility operations with walking speed to ensure safety
  - Faster mobility requires specific knowledge of the environment
  - Realization via a dynamic context model based on sensor data with uncertainties
  - Using Data Stream Management Technology for Sensor Fusion

Project SaLsA
http://www.offis.de



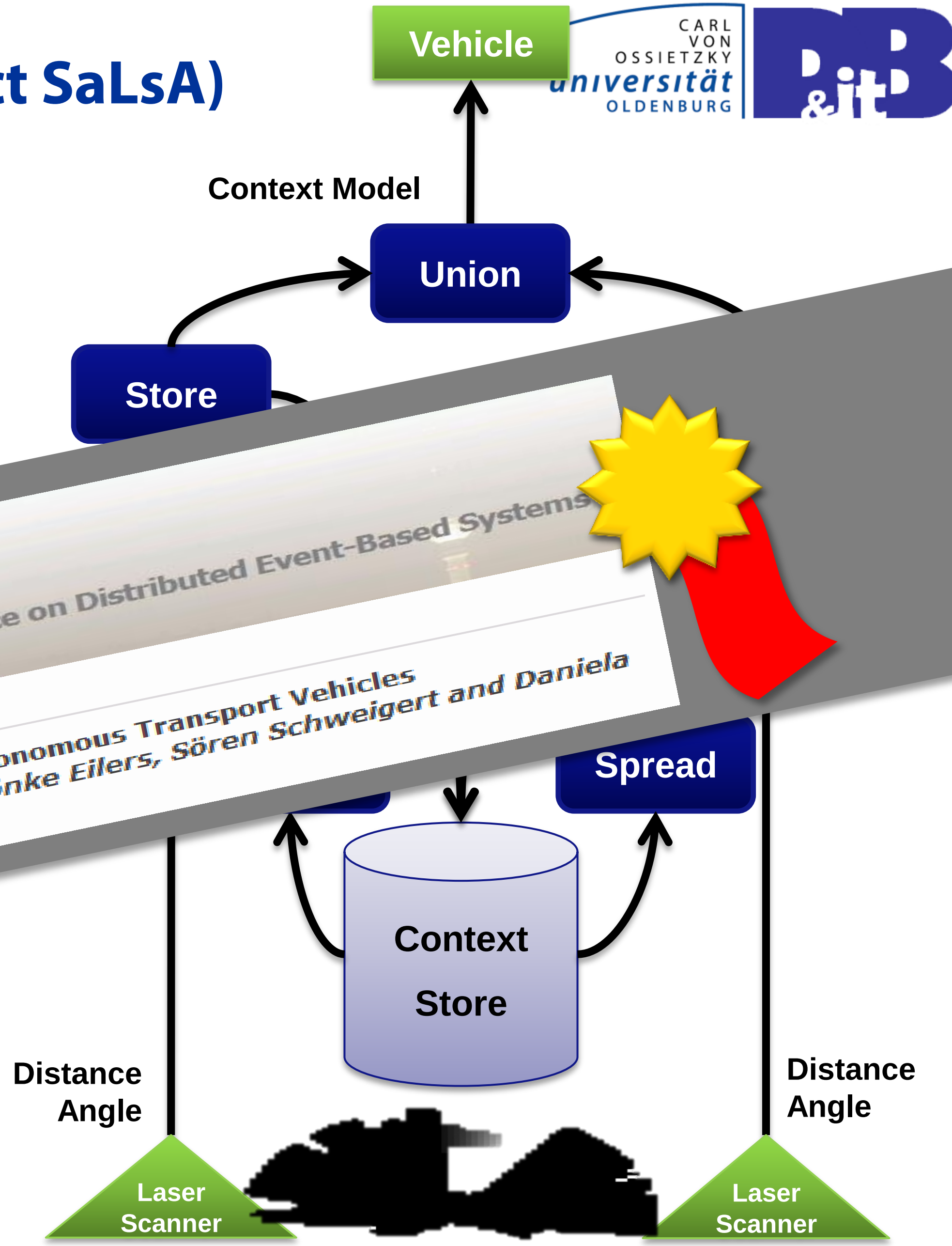
Sensor Fusion Example (project SaLsA)

Operators:

- Union Operator: Union and transmission of updated context models according to their timestamps
- Store Operator: Storing of new occupancy grid into the Context Store
- Merge Op…

…ple …th their time …of validity

- 2 Laserscanner with up to 1080 beams (@0.25°), resulting in **54,000 points per second per sensors**




DEBS 2012
The 6th ACM International Conference on Distributed Event-Based Systems
July 16-20, 2012

Best Demonstration Award

Context-Model Generation for Safe Autonomous Transport Vehicles
Christian Kuka, Sebastian Gerwinn, Sönke Eilers, Sören Schweigert and Daniela Nicklas

DBMS approaches assessment

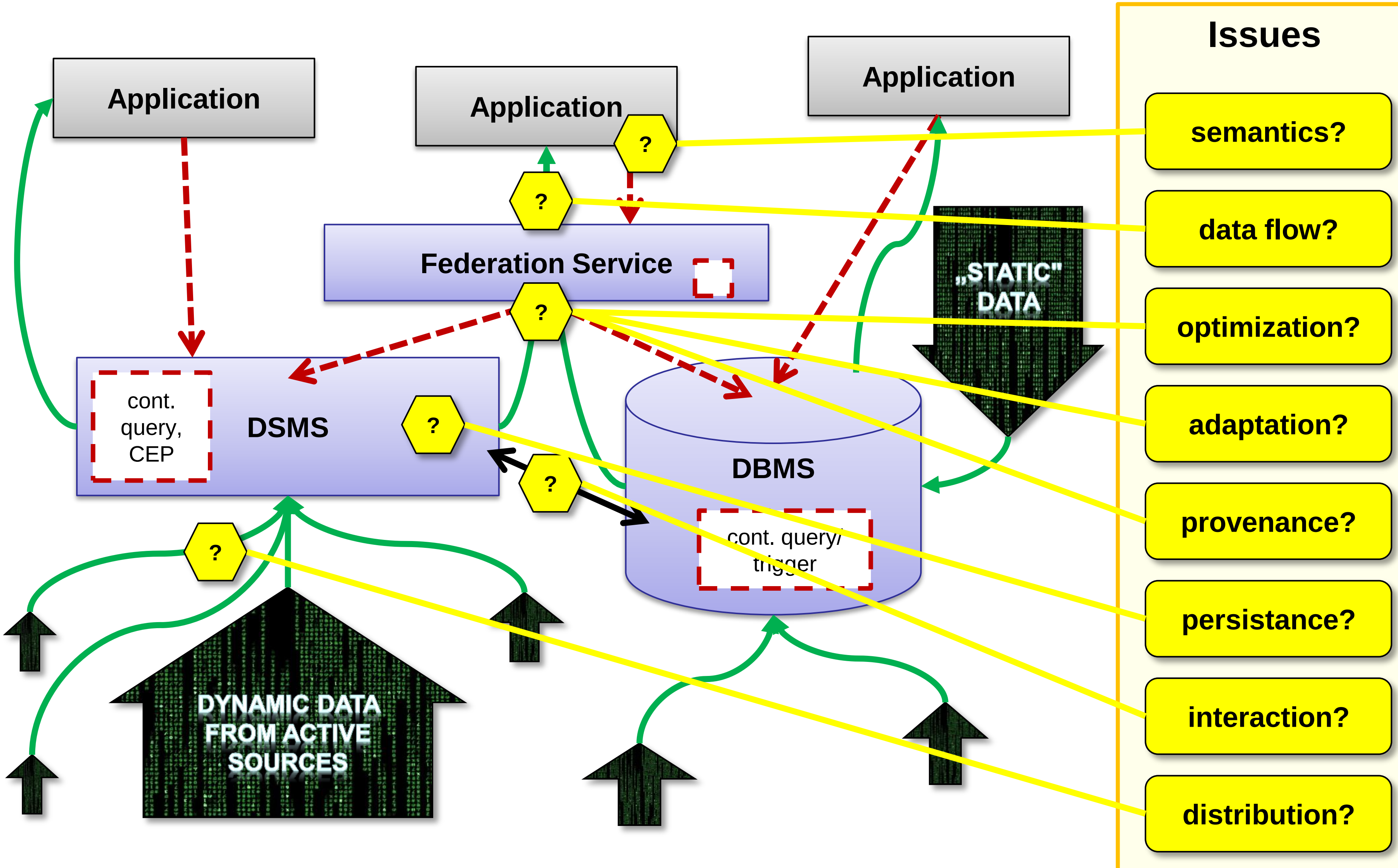
- Consider updates and the change of queries
- Are good up to a couple thousand of tuples/second

- However, there are application that need much higher throughput
  - real-time multi-sensor fusion (54000 points/second/ sensor)
  - dynamic context models
  - distributed data processing
  - specialized operators
  - not all data needs to be stored

→ **a federated approach might be a good idea!**

To boldly go for a Federation

Federation: the Big Picture



Query Semantics

- How do we formulate federated queries?
- DBMS:
  - SQL (but no standardized window semantics)
  - ECA rules, stored procedures, CQN
- DSMS:
  - CQL (Arasu, Babu, and Widom 2006)
  - StreamSQL (StreamBase Systems 2012)
  - Sase (Gyllstrom et al. 2007)
  - Aurora's boxes and arrows (Abadi et al. 2003)
  - IBM Infosphere Stream's Stream Programming Language SPL (Biem et al. 2010)

→ for federated queries, we need a common query language and semantics!

Issues

- **semantics?**
- data flow?
- optimization?
- adaptation?
- provenance?
- persistance?
- interaction?
- distribution?

Optimized Data Flow

- How do we decide about the data flow?

- Data from active resources is often produced decentrilized
- In classical federation approaches, data is sent to the federation software
  - might not be the best idea here
- Decide:
  - where to process which operator?
  - send data from DBMS to application or from DSMS?

… data flow must be part of the processing optimization.

Issues

- semantics?
- **data flow?**
- optimization?
- adaptation?
- provenance?
- persistance?
- interaction?
- distribution?

Optimized Cont. Query Distribution

- How do we decide about the optimal distribution of the continuous queries?

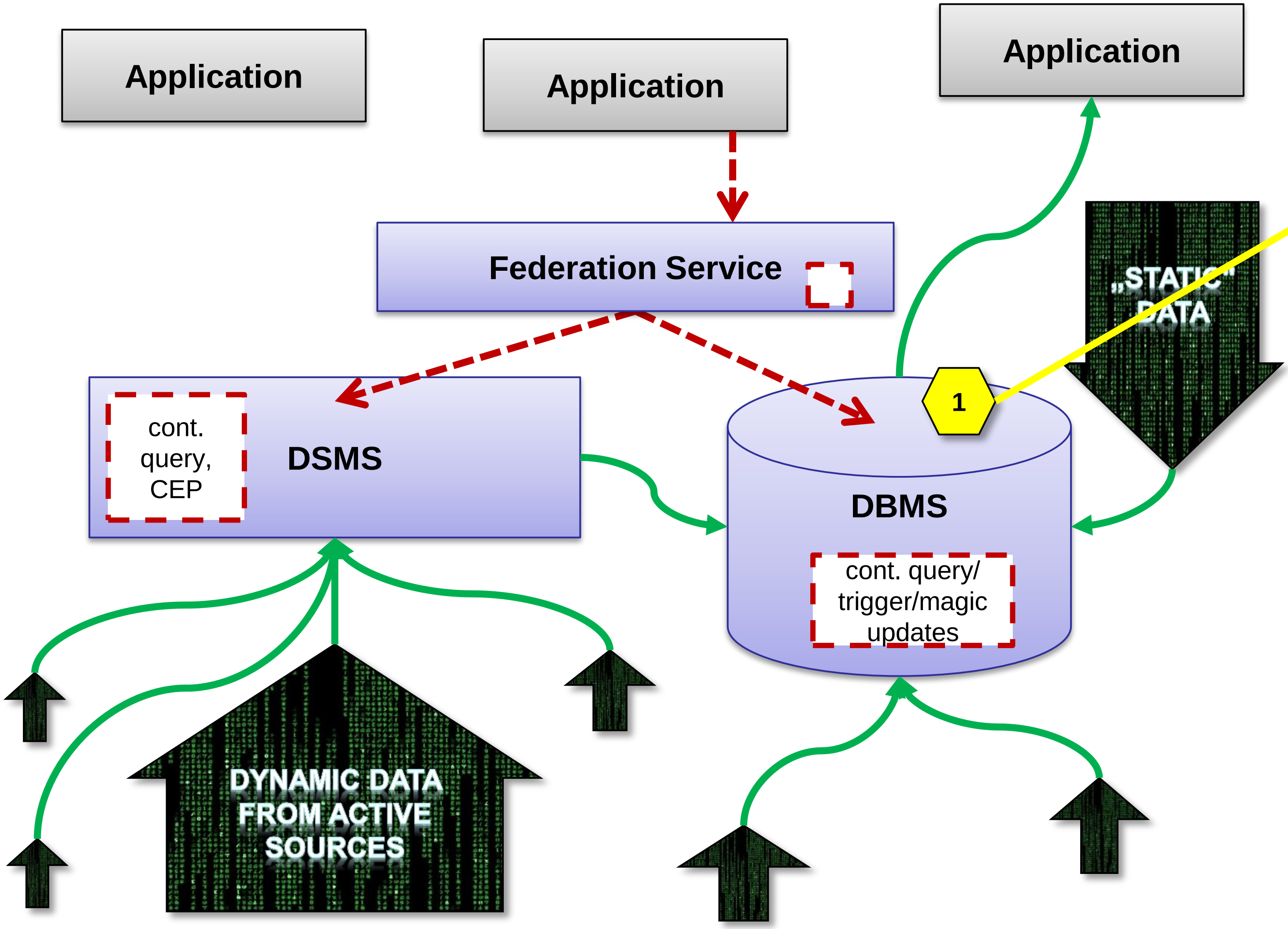
- Depends on
  - data location
  - update rate of data
  - required report rate of query result
  - required persistance of data
    - if system crashes and recovers, is the application interested in old (recovered) results?

Options range from CQN/triggers to pure DSMS queries to mixed executions

Issues

- semantics?
- data flow?
- **optimization?**
- adaptation?
- provenance?
- persistance?
- interaction?
- distribution?

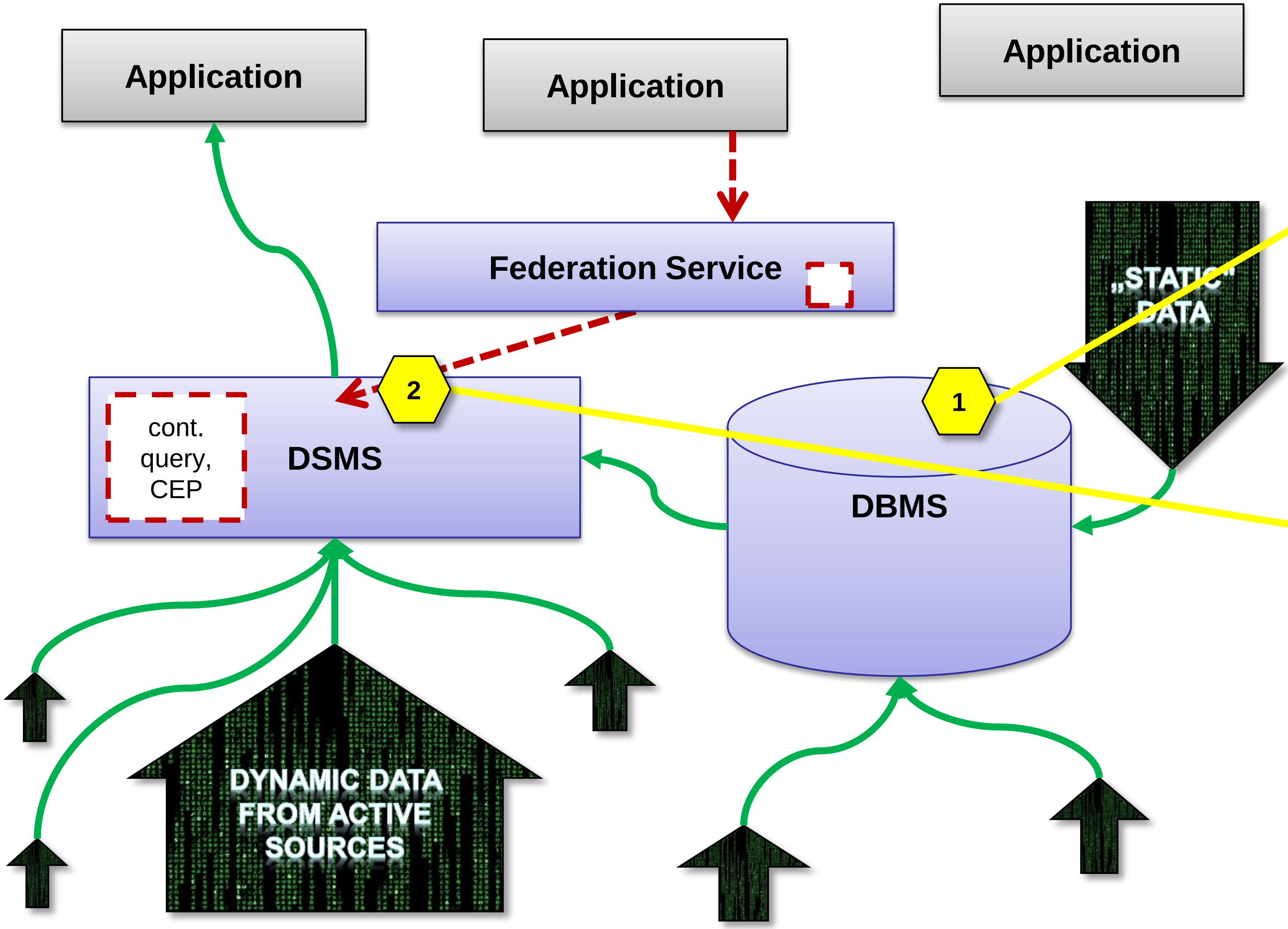
Query execution options



Options

pre-process in DSMS, register CQN in DBMS

Query execution options



Options

pre-process in DSMS, register CQN in DBMS

process in DSMS, get data from DBMS

Adaptation and Migration

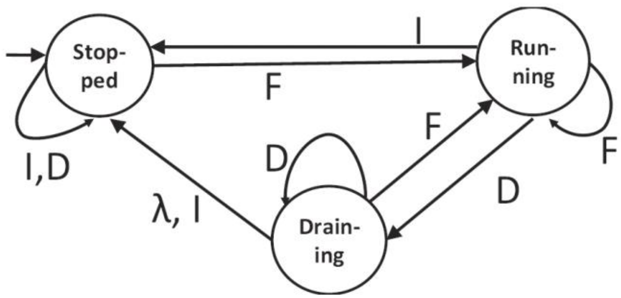
- If the parameters used for the query distribution change, how do we adapt the continous query to the new situation?
  - monitoring of
    - data location
    - update rate of data
    - required report rate of query result
    - required persistance of data
- If we decide to adapt – how do we migrate from old state to new state?
  - have to take care of stateful operators

Issues

- semantics?
- data flow?
- optimization?
- **adaptation?**
- provenance?
- persistance?
- interaction?
- distribution?

Adaptation and Migration

- Some works has been done for migrating pure DSMS queries:
  - F: Fresh start: start a new continuous query
  - I: Immediate stop: the query stops processing immediately
  - D: Drain stop: the query will be graduately stopped, finishing partially results
- Resulting stateflow diagram for continuous queries:



Kyumars Sheykh Esmaili, Tahmineh Sanamrad, Peter M. Fischer, Nesime Tatbul: Changing flights in mid-air: a model for safely modifying continuous queries. SIGMOD 2011:613-624

Issues

- semantics?
- data flow?
- optimization?
- **adaptation?**
- provenance?
- persistance?
- interaction?
- distribution?

Data and Query Provenance

- How can we know why a notification was produced (or not?)
- Many applications would code complex decision trees or situation recognition in federated DSMS/DBMS query plans
- Data provenance:
  - for a given federated result, which data tuples contributed to it?
- Query provenance:
  - when was / is which query plan active?
    - can be used to re-construct data provenance

Issues

- semantics?
- data flow?
- optimization?
- adaptation?
- **provenance?**
- persistance?
- interaction?
- distribution?

Data and Query Provenance

- Use Cases
  - Ad-hoc human inspection
  - Continuous query debugging
  - Indicator-based assurance
  - Event warehousing
- Challenges
  - Infinity, Performance, and Aggregation
  - Non-determinism (of some DSMS)
  - Order (of DSMS processing)
- Some work for DSMS:

Boris Glavic, Kyumars Sheykh Esmaili, Peter M. Fischer, Nesime Tatbul: The Case for Fine-Grained Stream Provenance. BTW Workshops 2011:58-61

Issues

- semantics?
- data flow?
- optimization?
- adaptation?
- **provenance?**
- persistance?
- interaction?
- distribution?

Persistance and Intelligent Archiving

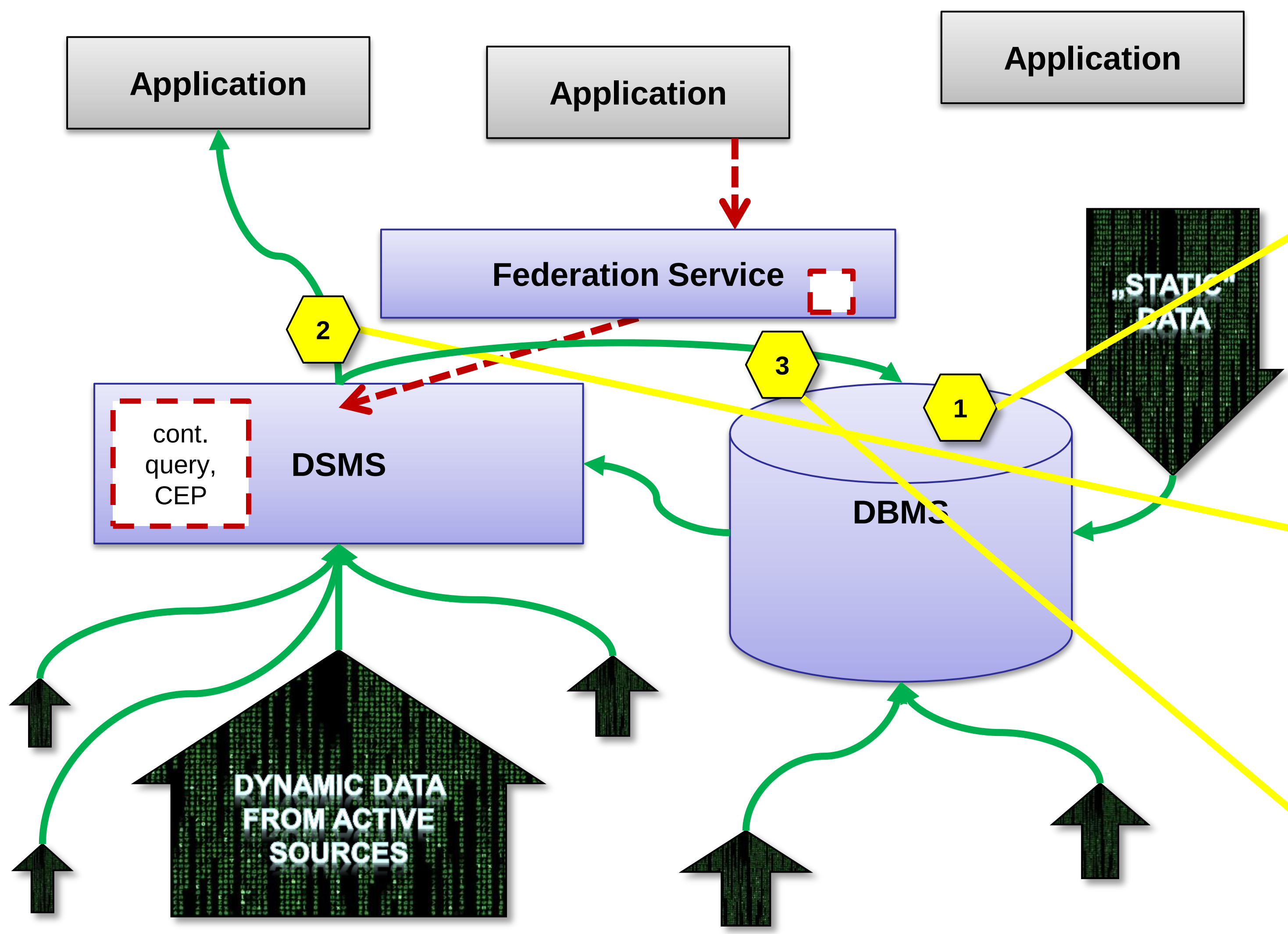
- How can we realize persistance in a world of high volume data streams?
  - not everything has to be persistant
  - but how to decide?

1. manually
   - by registering archiving queries
2. randomly
   - by sampling parts of the stream
3. automatically intelligent = model-based & adaptive
   - by learning patterns / models in the stream, e.g., by stream data mining
     - archive only new patterns in hig h resolution
     - count the other (= store aggregated information)

Issues

- semantics?
- data flow?
- optimization?
- adaptation?
- provenance?
- **persistance?**
- interaction?
- distribution?

Query execution options



Options

1. pre-process in DSMS, register CQN in DBMS
2. process in DSMS, get data from DBMS
3. if persistance or provenance is needed, store results (and queries) in DBMS

DSMS and DBMS interaction

- How do the DSMS and the DBMS interact?
  - who controls whom?
  - how tightly are they coupled?

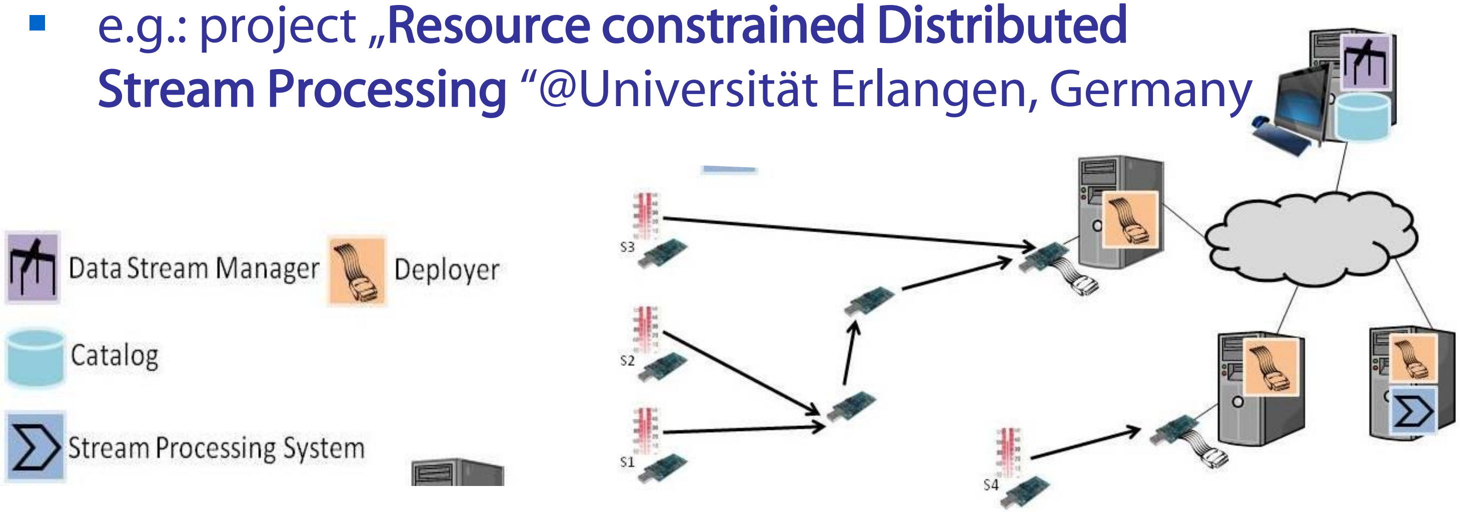
1. The federation controls both system
   - low coupling between the systems
   - only running queries are affected if a system changes (interface, schema, …)
2. The DSMS controls the DBMS
   - it issues queries to the DBMS to access static data
   - this function is integrated in most DSMS
3. The DBMS controls the DSMS
   - would require concepts to include DSMS queries in, e.g., trigger definitions

Issues

- semantics?
- data flow?
- optimization?
- adaptation?
- provenance?
- persistance?
- **interaction?**
- distribution?

Distributed Continuous Query Processing

- How to optimize the distribution of the query processing?
- Many active data sources are physically distributed or even mobile (sensor networks, mobile sensors in smart phones or trucks)
- Many sensing devices have (some) processing power
- Should be considered in query optimization
- e.g.: project „**Resource constrained Distributed Stream Processing** "@Universität Erlangen, Germany



Michael Daum, Frank Lauterwald, Philipp Baumgärtel, Niko Pollner, Klaus Meyer-Wegener: Efficient and cost-aware operator placement in heterogeneous stream-processing environments. DEBS 2011:393-394

Issues

- semantics?
- data flow?
- optimization?
- adaptation?
- provenance?
- persistance?
- interaction?
- **distribution?**

Conclusion

- To combine static and dynamic data management, many issues have to be addressed
  - DBMS has to be extended for temporal semantics
  - DSMS have to mature in processing semantics, determinism and persistance concepts
  - Together, a data management biotop has to have a
- In the long run, DSMS functionalities will be part of a DBMS system
- Outlook: consideration of NoSQL-concepts

Thank's for the fish!

… also to Andreas Behrend (Uni Bonn), Nazario Cipriani (Uni Stuttgart), Dieter Gawlick (Oracle), Marco Grawunder (Uni Oldenburg) and Christian Kuka (OFFIS) for donating slides and ideas