ICT Support for Adaptiveness and (Cyber)Security in the Smart Grid DAT300

An overview of Data Streaming

Vincenzo Gulisano

vinmas@chalmers.se (room 5119)

Chalmers University of technology

Distributed Computing and Systems
Chalmers university of technology

Agenda

- Motivation
- The data streaming philosophy
- System Model
- Sample Data Streaming application
- Evolution of Stream Processing Engines
- Challenges in the context of Smart Grids

Agenda

- Motivation
- The data streaming philosophy
- System Model
- Sample Data Streaming application
- Evolution of Stream Processing Engines
- Challenges in the context of Smart Grids

Motivation

- Applications such as:
  - Sensor networks
  - Network Traffic Analysis
  - Financial tickers
  - Transaction Log Analysis
  - Fraud Detection

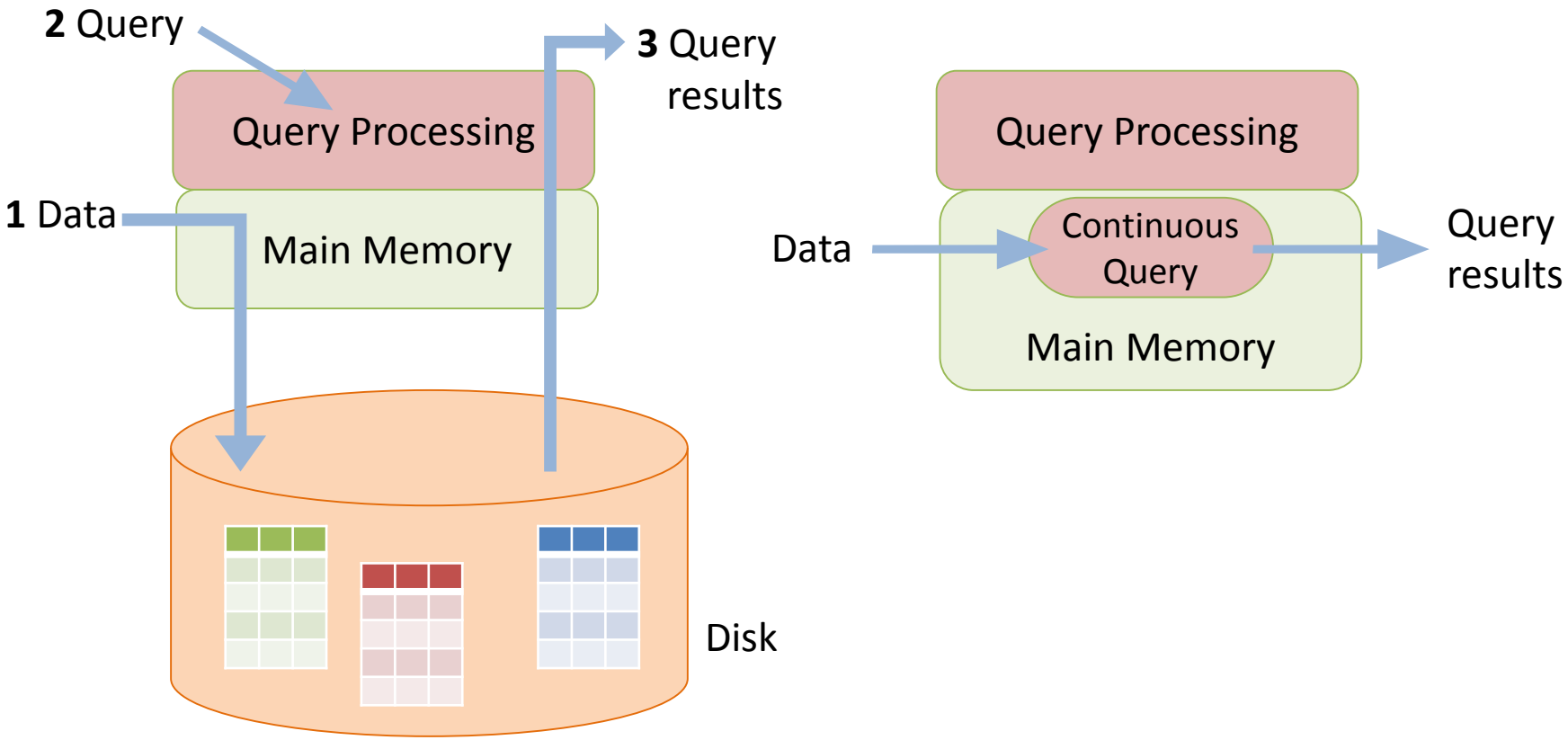
- Require:
  - Continuous processing of data streams
  - Real Time Fashion

Motivation

- Store and process is not feasible
  - high-speed networks, nanoseconds to handle a packet
  - ISP router: gigabytes of headers every hour,…
- Data Streaming:
  - In memory
  - Bounded resources
  - Efficient one-pass analysis

Motivation

- DBMS vs. DSMS



Agenda

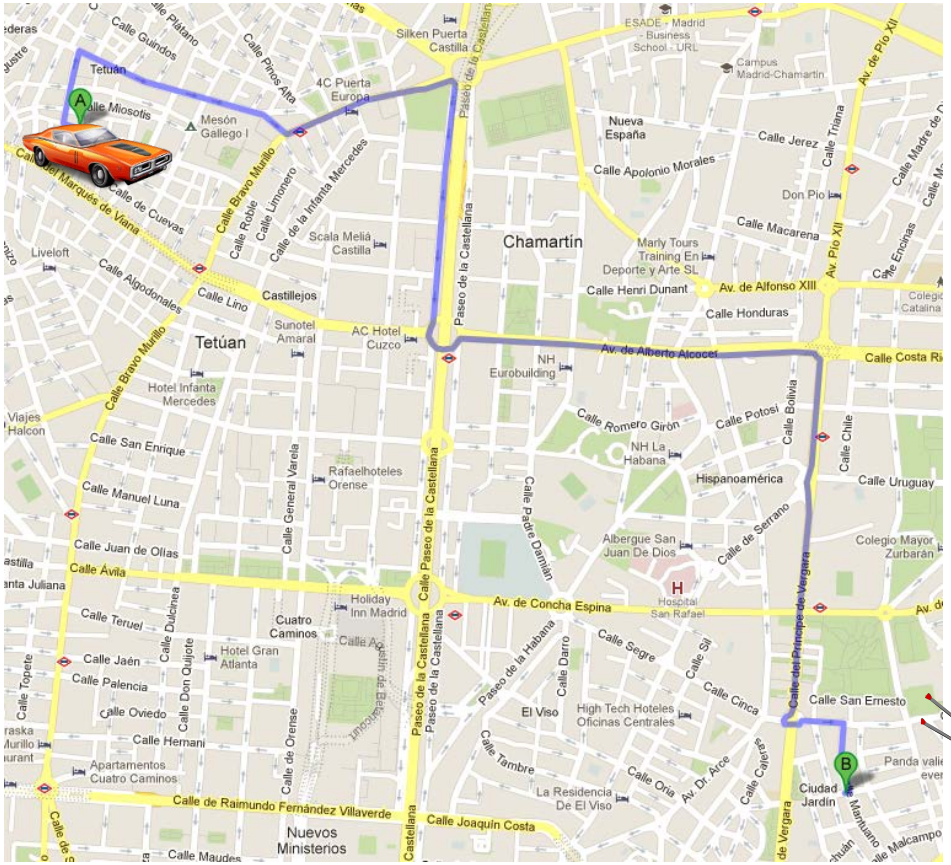
- Motivation
- The data streaming philosophy
- System Model
- Sample Data Streaming application
- Evolution of Stream Processing Engines
- Challenges in the context of Smart Grids

Database vs. Data Streaming

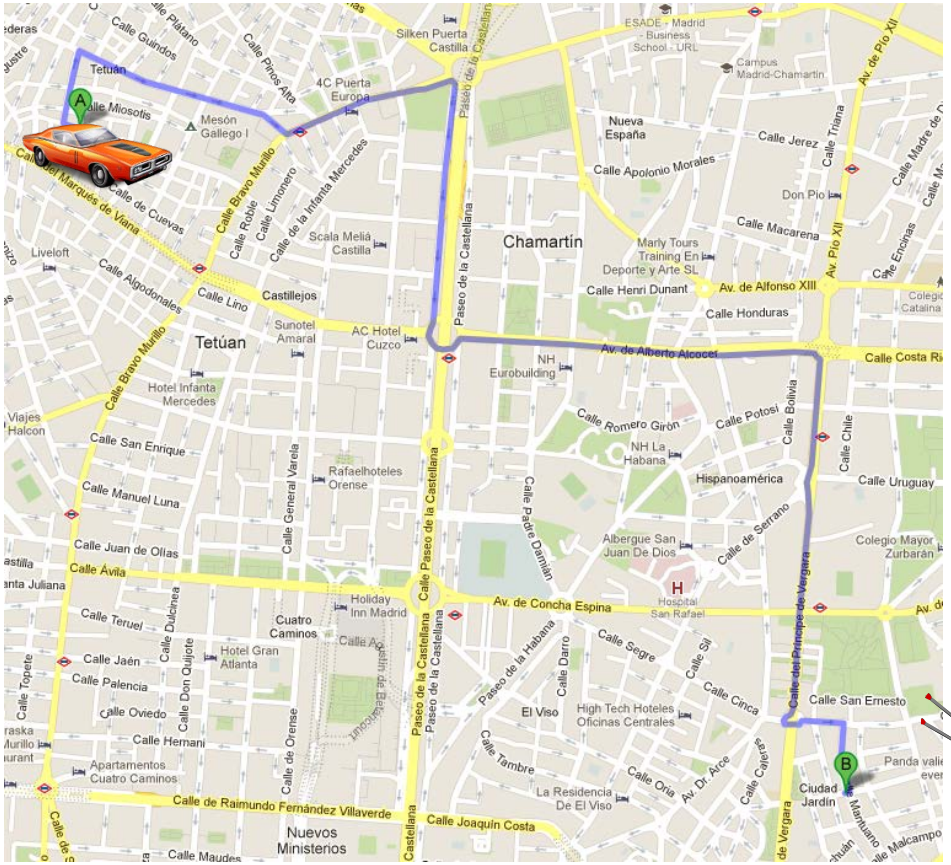
- Problem:
  - James travels by car from A to B
  - His grandmother is worried, she wants to know if he exceeds the speed limit

- How will the “database” and the “data streaming” grandmothers do this?

Database vs. Data Streaming



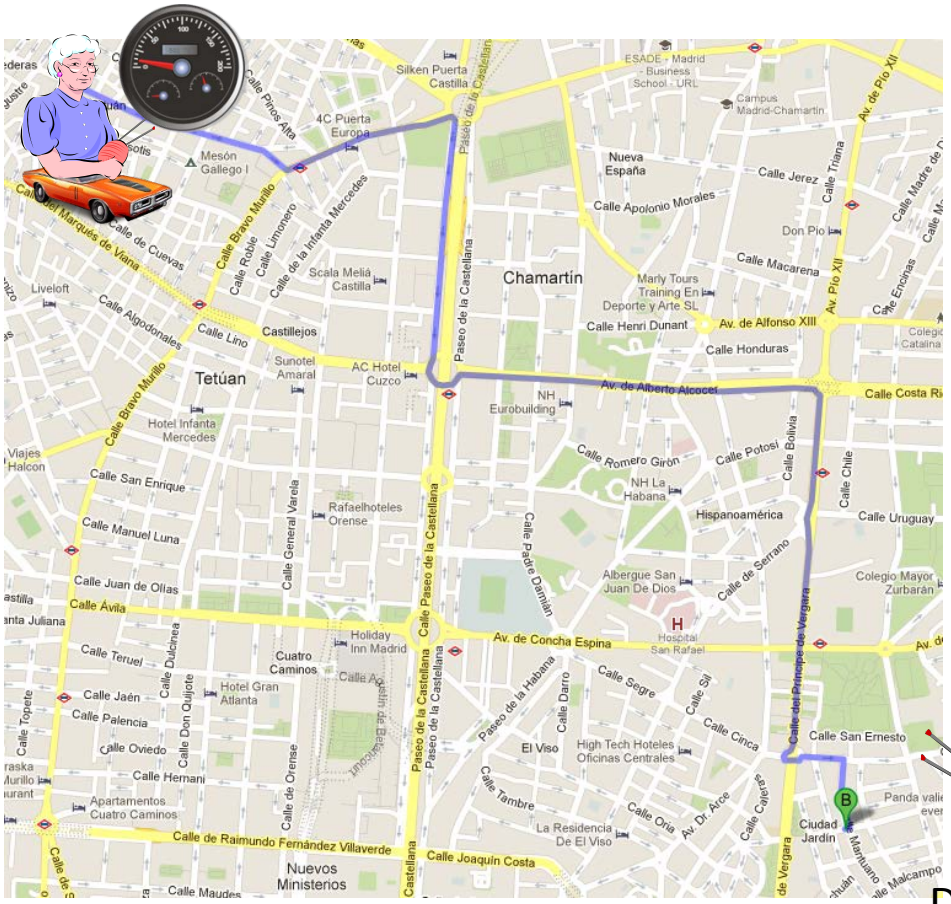
Database vs. Data Streaming



1. First the data, then the query
2. Precise result
3. Need to store information

Database grandmother

Database vs. Data Streaming



1. First the query, then the data
2. “Continuous” result
3. No need to store information

Data streaming grandmother

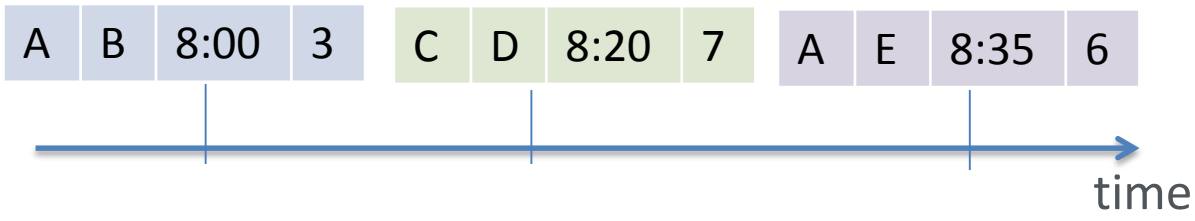
Agenda

- Motivation
- The data streaming philosophy
- System Model
- Sample Data Streaming application
- Evolution of Stream Processing Engines
- Challenges in the context of Smart Grids

System Model

- Data Stream: unbounded sequence of tuples
  - Example: Call Description Record (CDR)

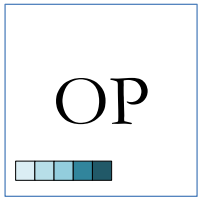
Field	Field
Caller	text
Callee	text
Time (secs)	int
Price (€)	double



System Model

- Operators:

Stateless
1 input tuple
1 output tuple

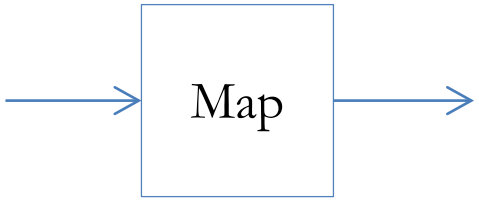


Stateful
1+ input tuple(s)
1 output tuple

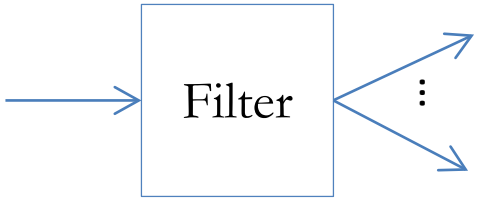
System Model

Stateless Operators

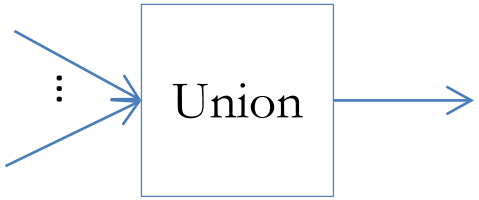
Map: transform tuples schema
Example: convert price € → $



Filter: discard / route tuples
Example: route depending on price



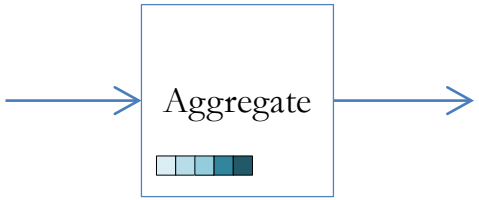
Union: merge multiple streams
(sharing the same schema)
Example: merge CDRs from
different sources



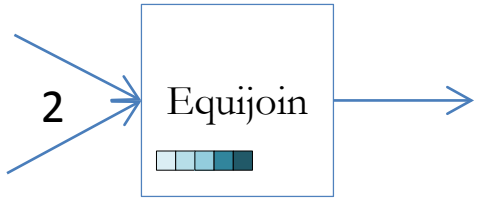
System Model

Stateful Operators

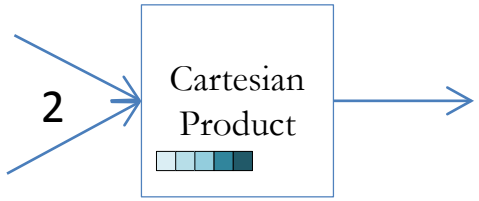
Aggregate: compute aggregate functions (group-by)
Example: compute avg. call duration



Equijoin: match tuples from 2 streams (equality predicate)
Example: match CDRs with same price



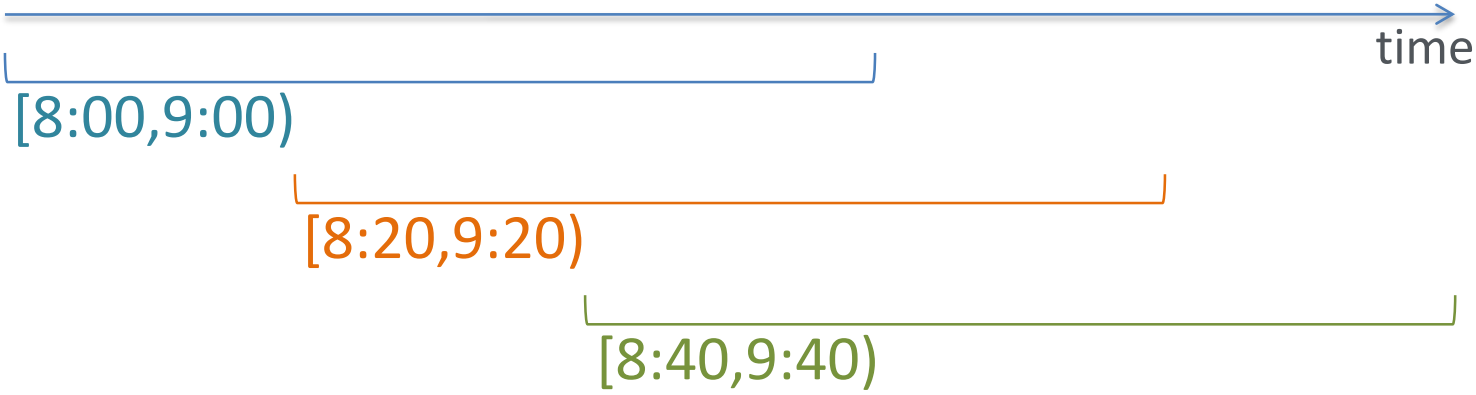
Cartesian Product: merge tuples from 2 streams (arbitrary predicate)
Example: match CDRs with prices in the same range



System Model

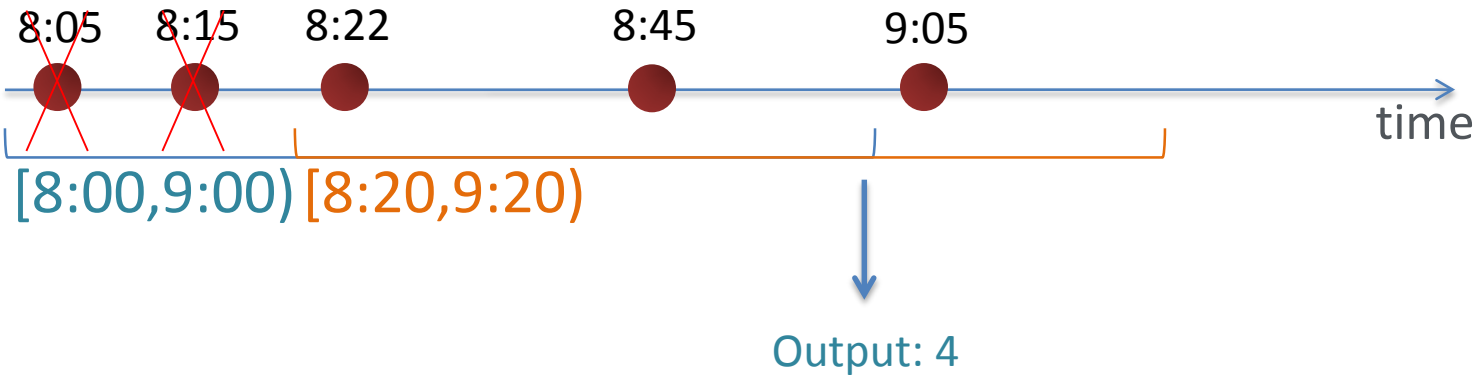
- Infinite sequence of tuples / bounded memory

  → windows
- Example: 1 hour windows



System Model

- Infinite sequence of tuples / bounded memory
  → windows
- Example: count tuples - 1 hour windows



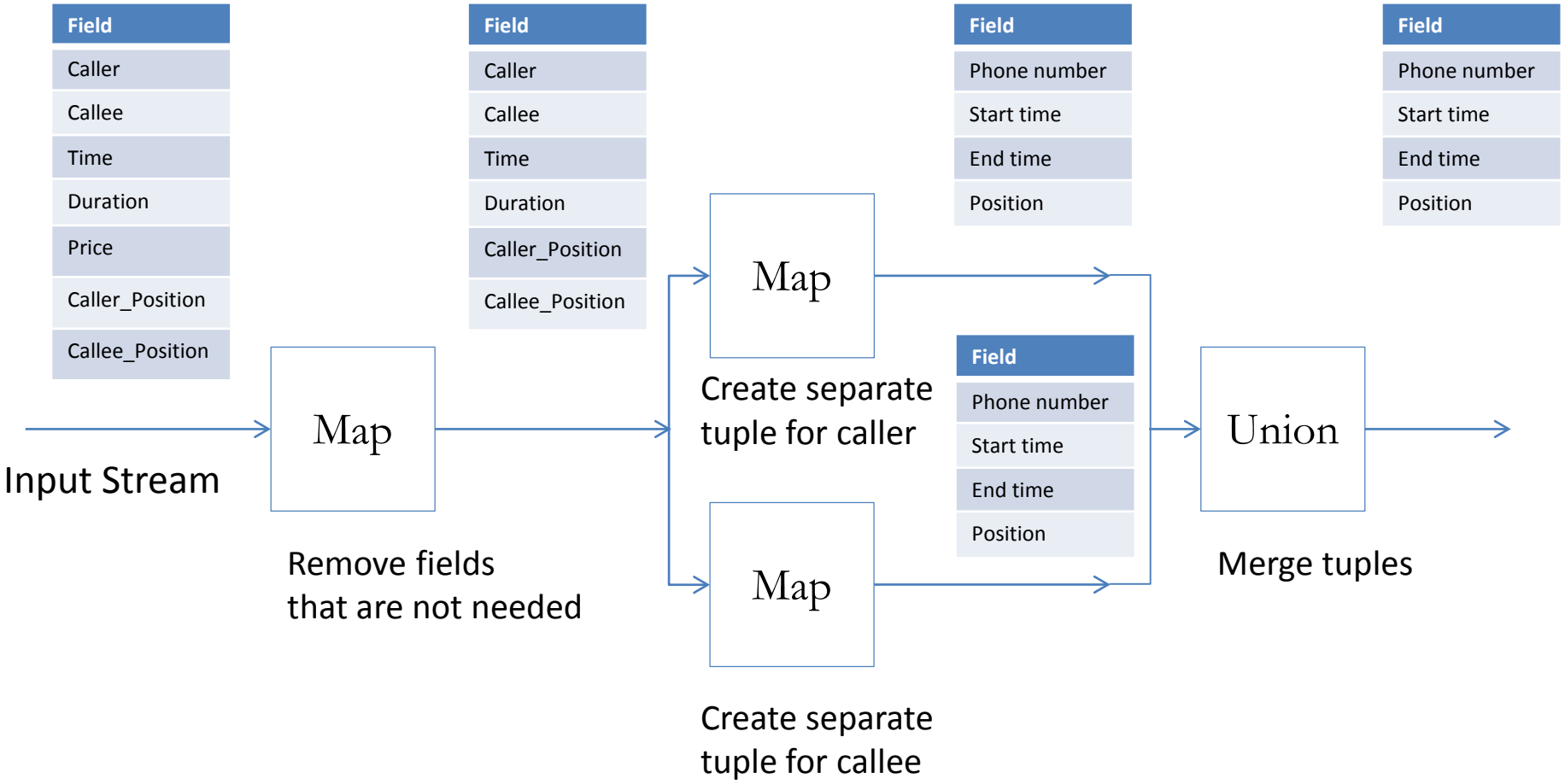
Agenda

- Motivation
- The data streaming philosophy
- System Model
- Sample Data Streaming application
- Evolution of Stream Processing Engines
- Challenges in the context of Smart Grids

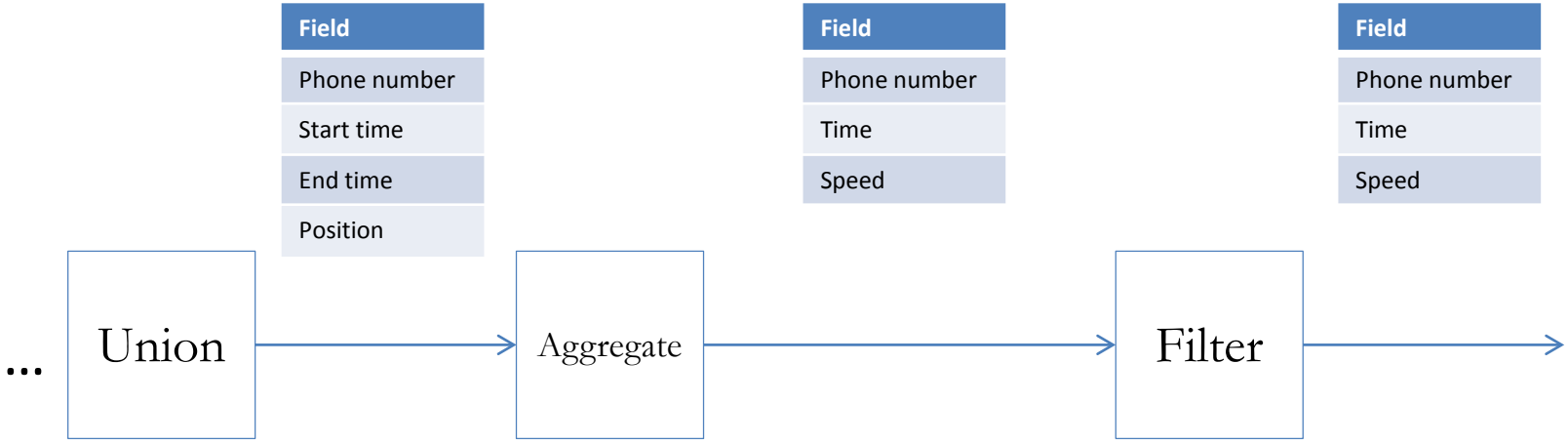
Continuous Query Example

- Fraud detection, High Mobility
  - Spot mobile phone whose space and time distance between two consecutive calls is suspicious

High Mobility Continuous Query (1/2)



High Mobility Continuous Query (2/2)



Merge tuples

For each consecutive pair of calls referring to the same number compute speed

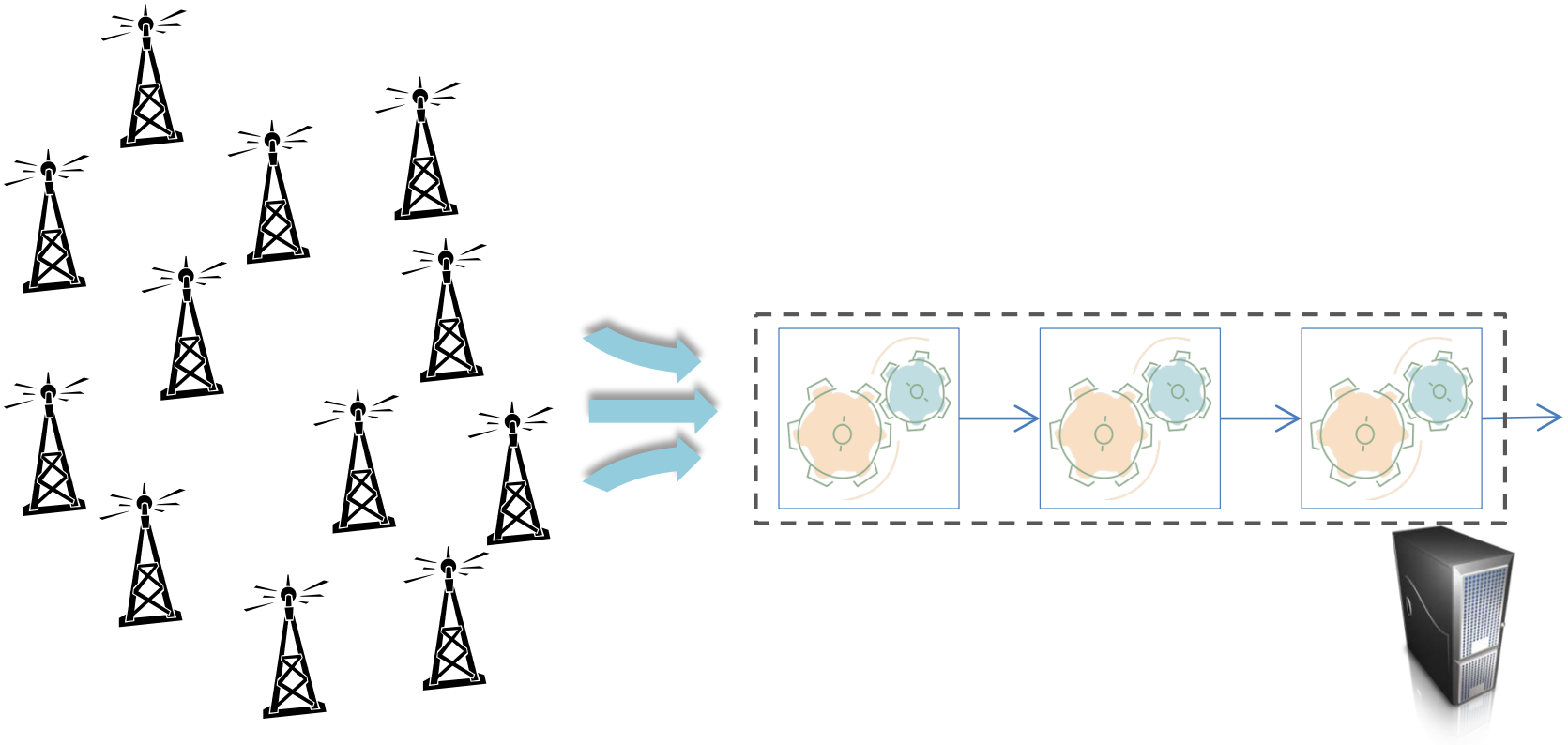
Window type: tuple based
Window size: 2
Window Advance: 1

Forward tuples with speed exceeding a given threshold

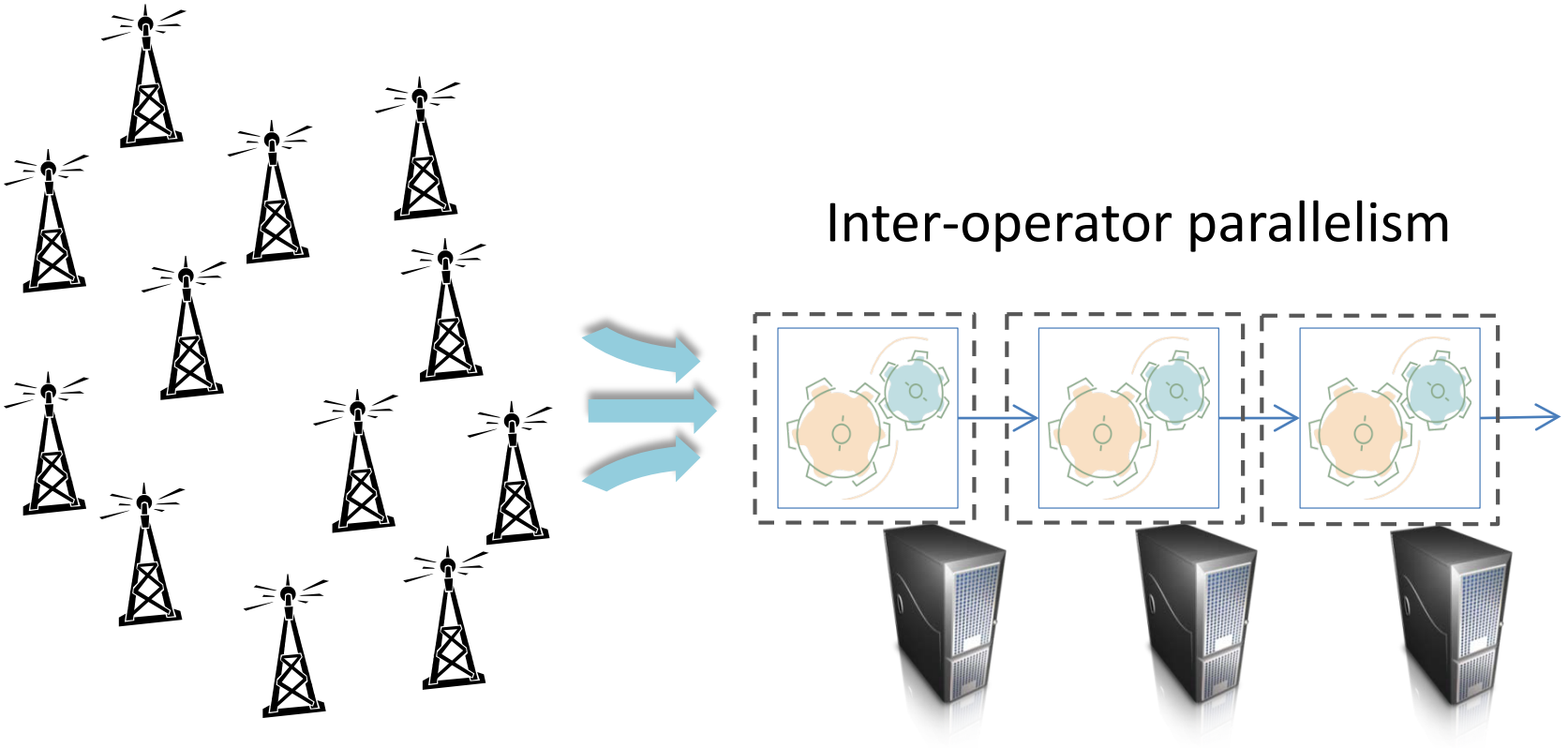
Agenda

- Motivation
- The data streaming philosophy
- System Model
- Sample Data Streaming application
- Evolution of Stream Processing Engines
- Challenges in the context of Smart Grids

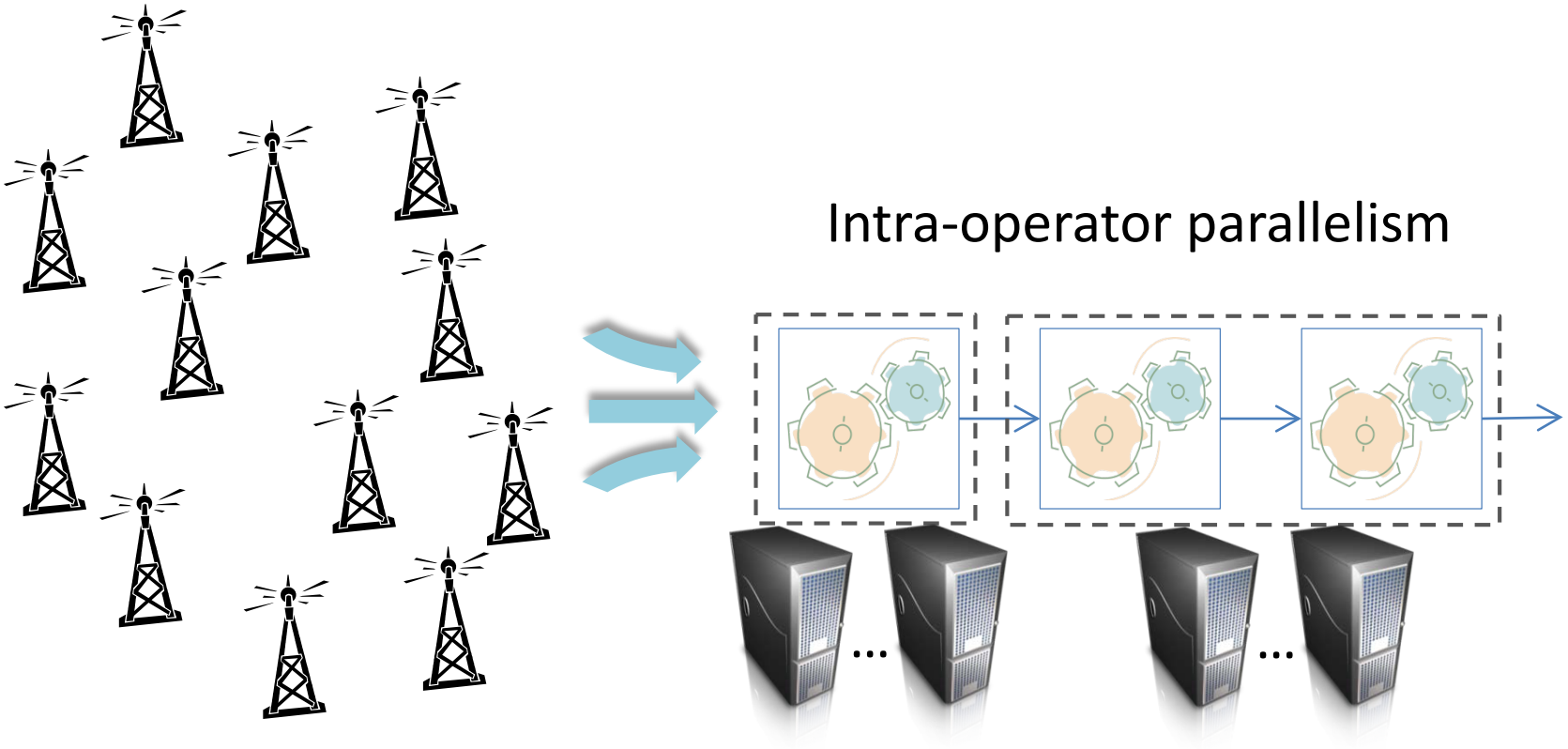
Centralized SPEs



Distributed SPEs

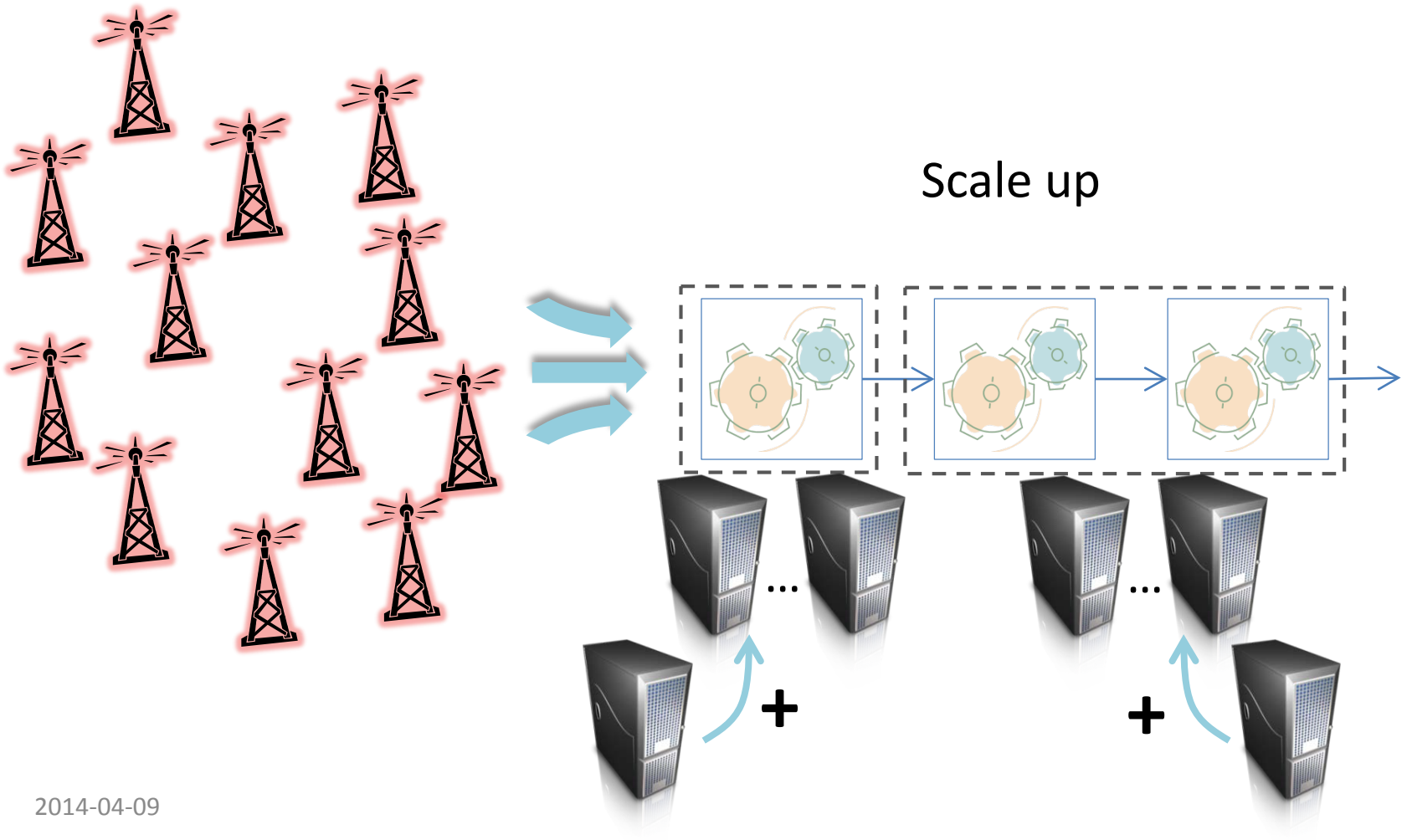


Parallel SPEs

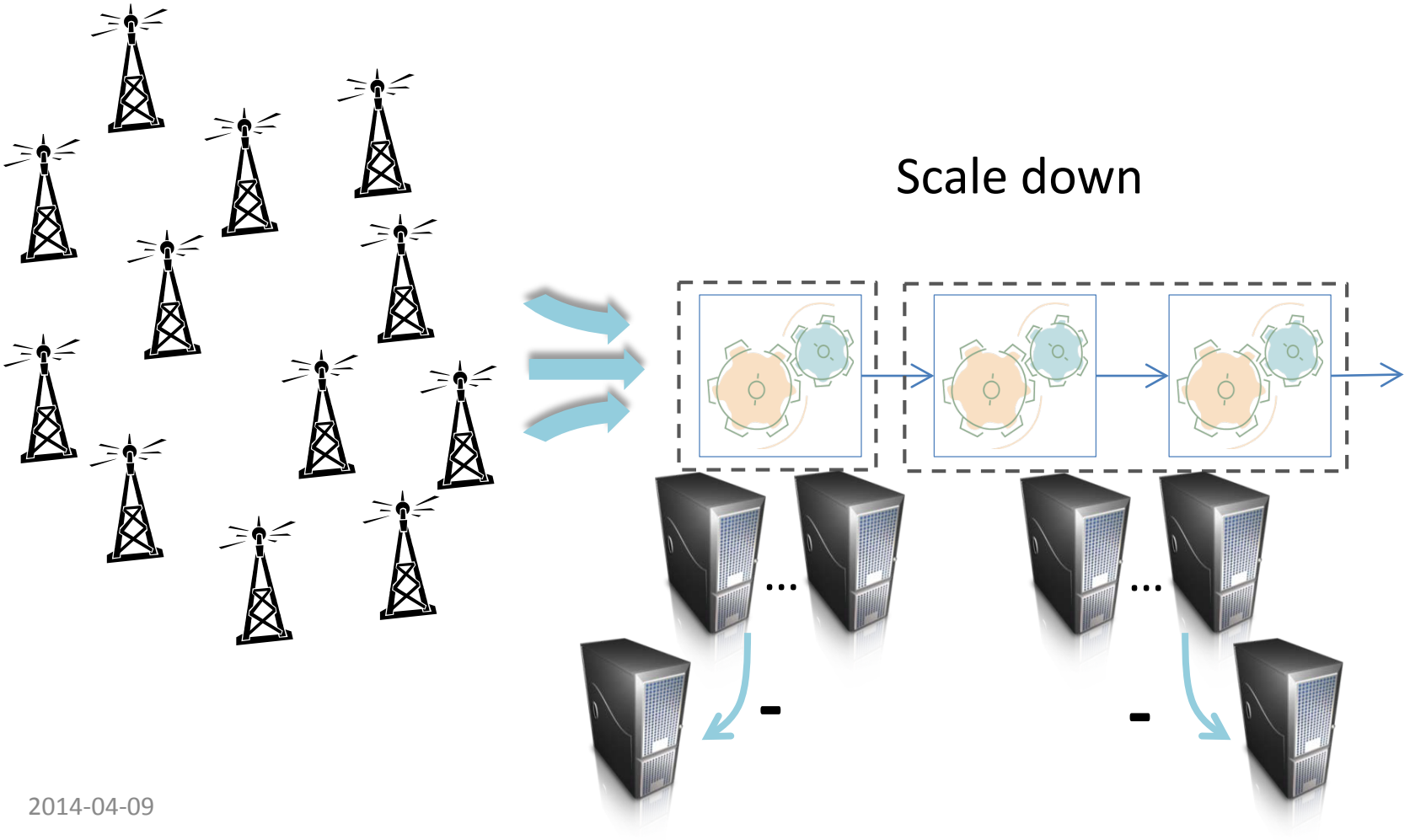


Over-provisioning or under-provisioning?

Elastic SPEs



Elastic SPEs



Agenda

- Motivation
- The data streaming philosophy
- System Model
- Sample Data Streaming application
- Evolution of Stream Processing Engines
- Challenges in the context of Smart Grids

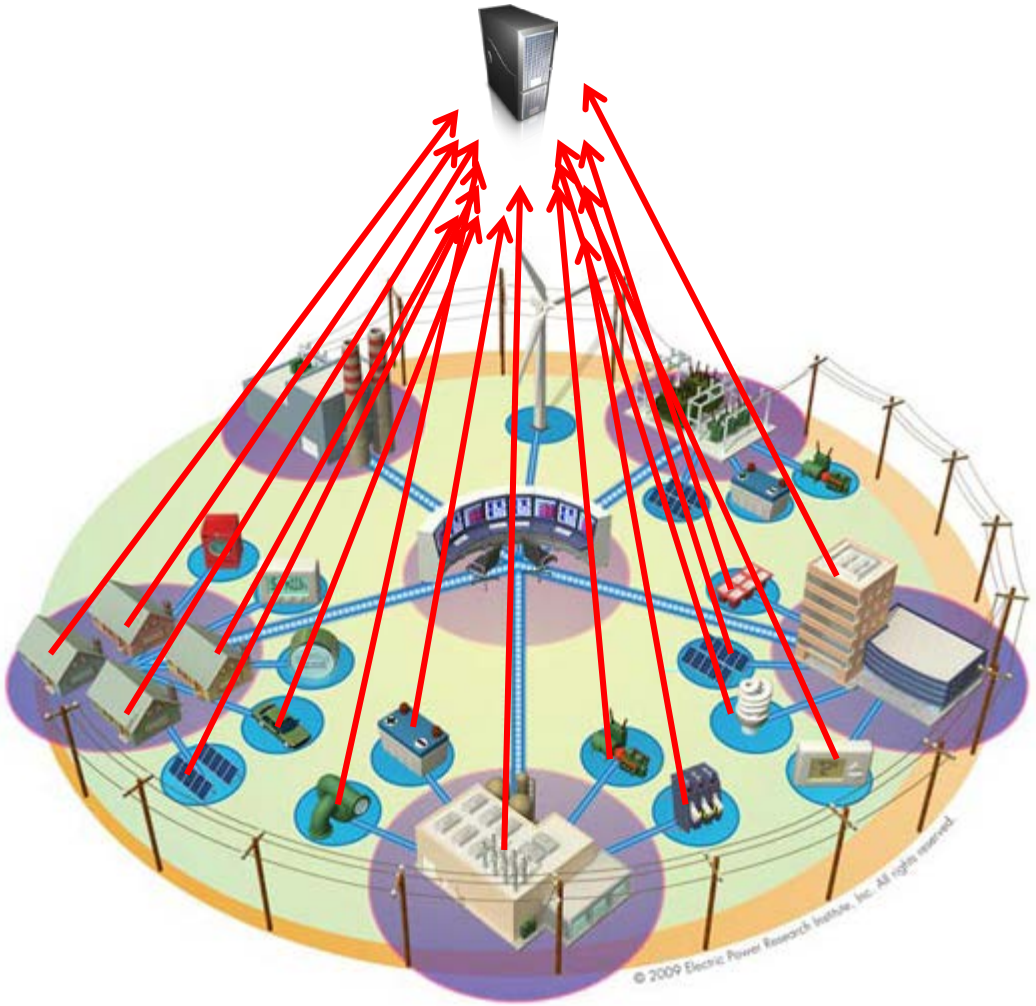
Challenges in the context of Smart Grids

- Process energy consumption data
  - Build profiles and spot deviations
  - Predictions / forecasts about consumption

Challenges in the context of Smart Grids

- Process control events
  - Spot possible threats
  - Monitor the devices status

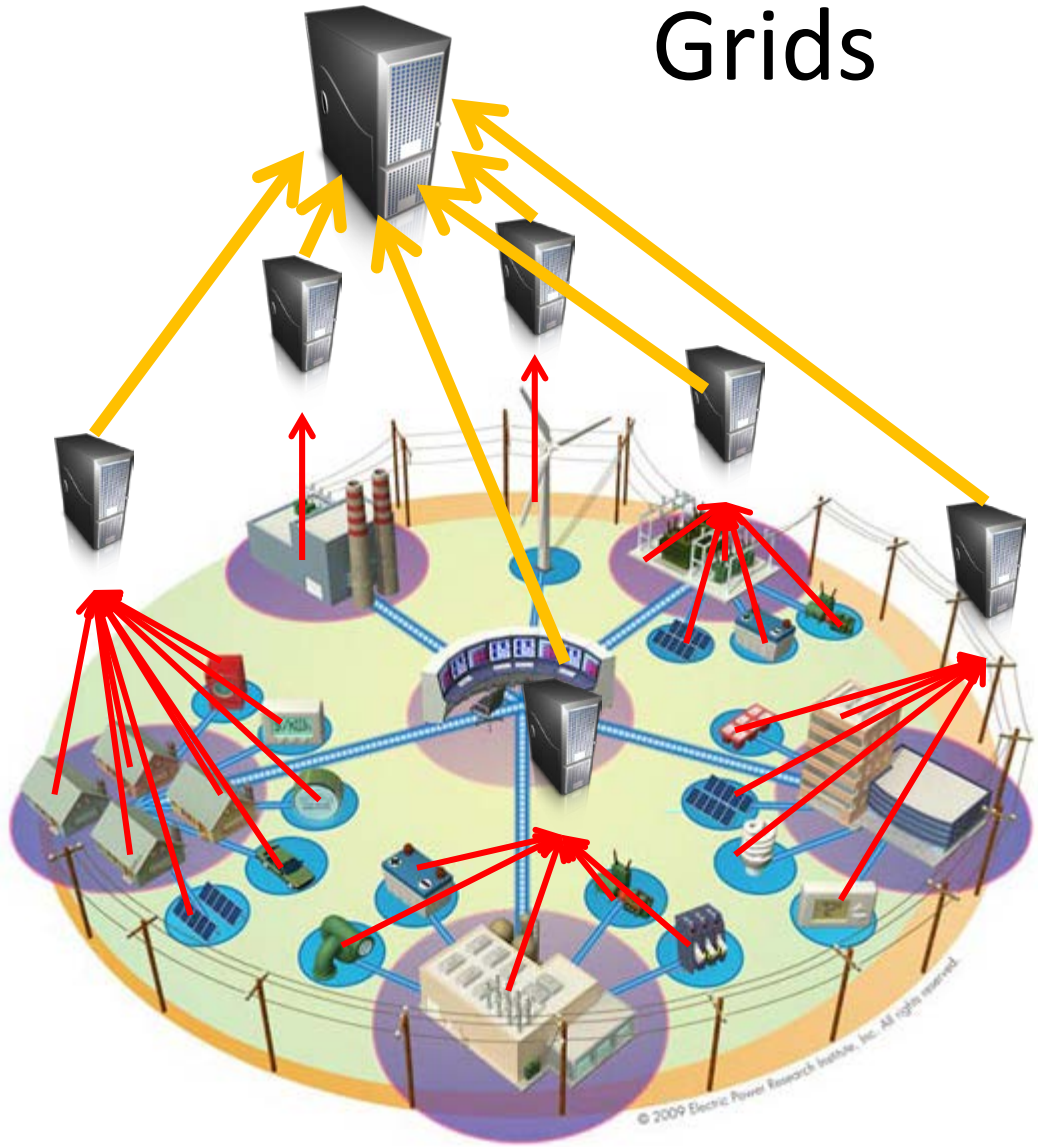
Challenges in the context of Smart Grids



How to process the information?

Centralized

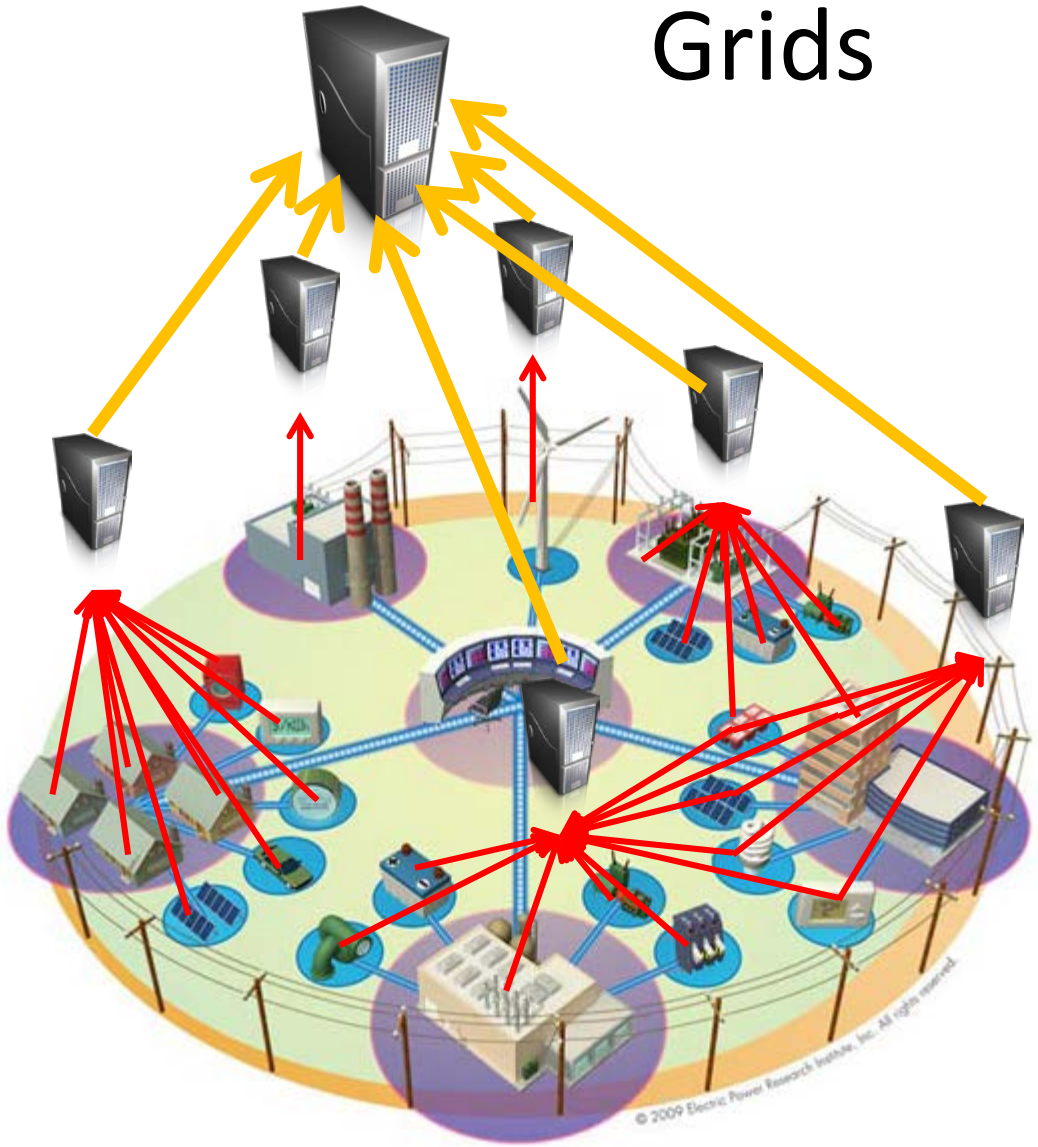
Challenges in the context of Smart Grids



How to process the information?

Distributed
(In-network aggregation)

Challenges in the context of Smart Grids



How to deal with constrained/limited resources?

What if this device is running out of battery?

An overview of Data Streaming

Questions?

Bibliography

1. Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, PODS ’02, New York, NY, USA, 2002. ACM.
2. Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, PODS ’02, New York, NY, USA, 2002. ACM.
3. Michael Stonebraker, U˘gur Çetintemel, and Stan Zdonik. The 8 requirements of realtime stream processing. SIGMOD Rec., 34(4), December 2005.
4. Nesime Tatbul. QoS-Driven load shedding on data streams. In Proceedings of the Worshops XMLDM, MDDE, and YRWS on XML-Based Data Management and Multimedia Engineering-Revised Papers, EDBT ’02, London, UK, UK, 2002. Springer-Verlag.
5. Arvind Arasu, Shivnath Babu, and Jennifer Widom. The CQL continuous query language: semantic foundations and query execution. The VLDB Journal, 15(2), June 2006.
6. Daniel J. Abadi, Don Carney, Ugur Cetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Michael Stonebraker, Nesime Tatbul, and Stan Zdonik. Aurora: a new model and architecture for data stream management. The VLDB Journal, 12(2), August 2003.
7. Arvind Arasu, Shivnath Babu, and Jennifer Widom. The CQL continuous query language: semantic foundations and query execution. The VLDB Journal, 15(2), June 2006.
8. Daniel J. Abadi, Don Carney, Ugur Cetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Michael Stonebraker, Nesime Tatbul, and Stan Zdonik. Aurora: a new model and architecture for data stream management. The VLDB Journal, 12(2), August 2003.

Bibliography

9. Vincenzo Gulisano, Ricardo Jiménez-Peris, Marta Patiño-Martínez, and Patrick Valduriez. Streamcloud: A large scale data streaming system. In ICDCS 2010: International Conference on Distributed Computing Systems, pages 126–137, June 2010.
10. Mehul Shah Joseph, Joseph M. Hellerstein, Sirish Ch, and Michael J. Franklin. Flux: An adaptive partitioning operator for continuous query systems. In In ICDE, 2002.
11. Vincenzo Gulisano, Ricardo Jimenez-Peris, Marta Patino-Martinez, Claudio Soriente, and Patrick Valduriez. Streamcloud: An elastic and scalable data streaming system. IEEE Transactions on Parallel and Distributed Systems, 99(PrePrints), 2012.
12. Thomas Heinze. Elastic complex event processing. In Proceedings of the 8th Middleware Doctoral Symposium, MDS ’11, New York, NY, USA, 2011. ACM.