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

An overview of Data Streaming

Vincenzo Gulisano

vinmas@chalmers.se (room 5124A)

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 (some examples)

Agenda

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


The privacy debate
Geekonomics
BusinessTechnology
BIG DATA
Big Data
FOR
DUMMIES
A Wiley Brand
Making Everything Easier!
BIG DATA
COMPUTERWORLD
WHY BIG DATA IS A BIG DEAL
The Economist
The data deluge
AND HOW TO HANDLE IT: A 14-PAGE SPECIAL REPORT

Motivation

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

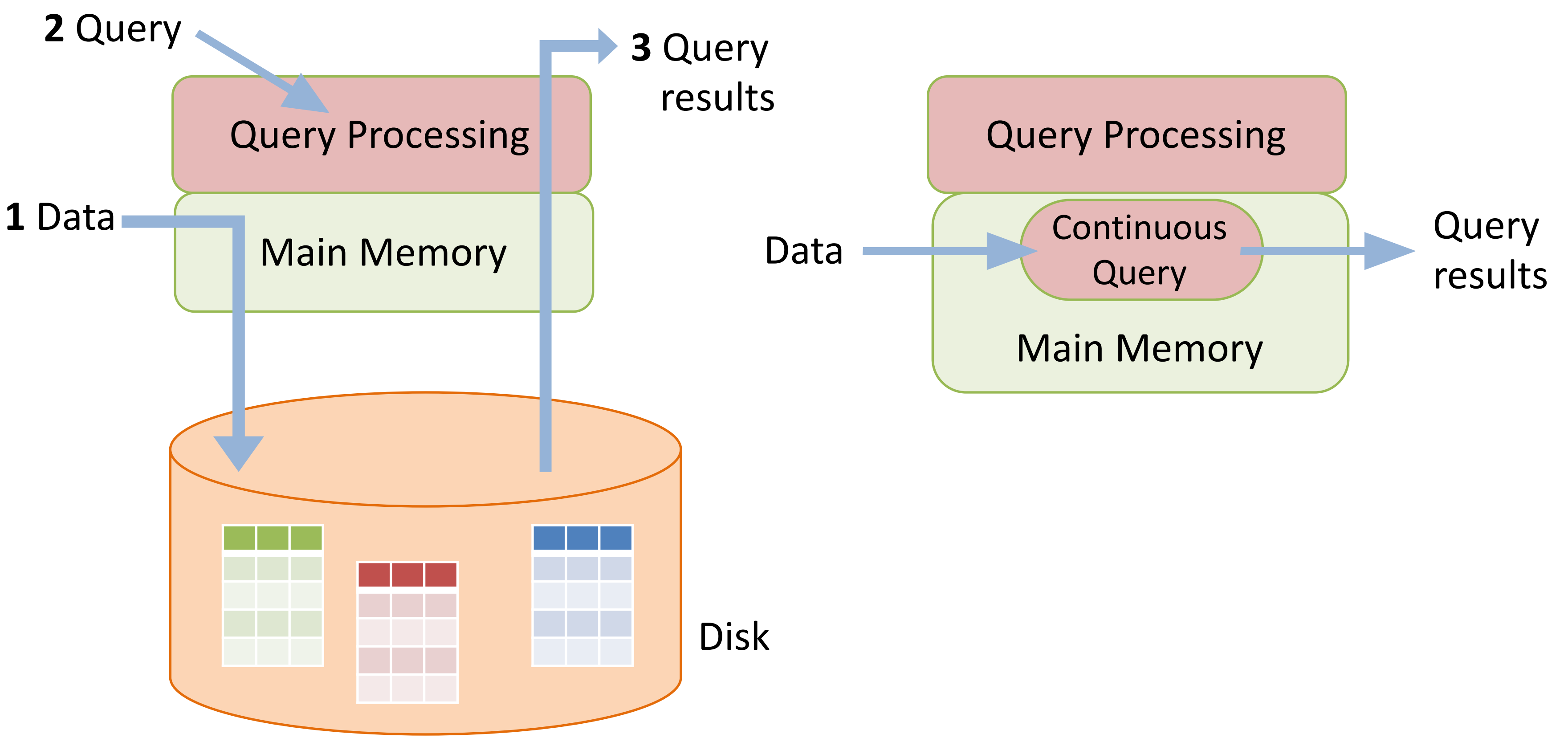
- Require:
  - Continuous processing of data streams
  - Real Time Fashion

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

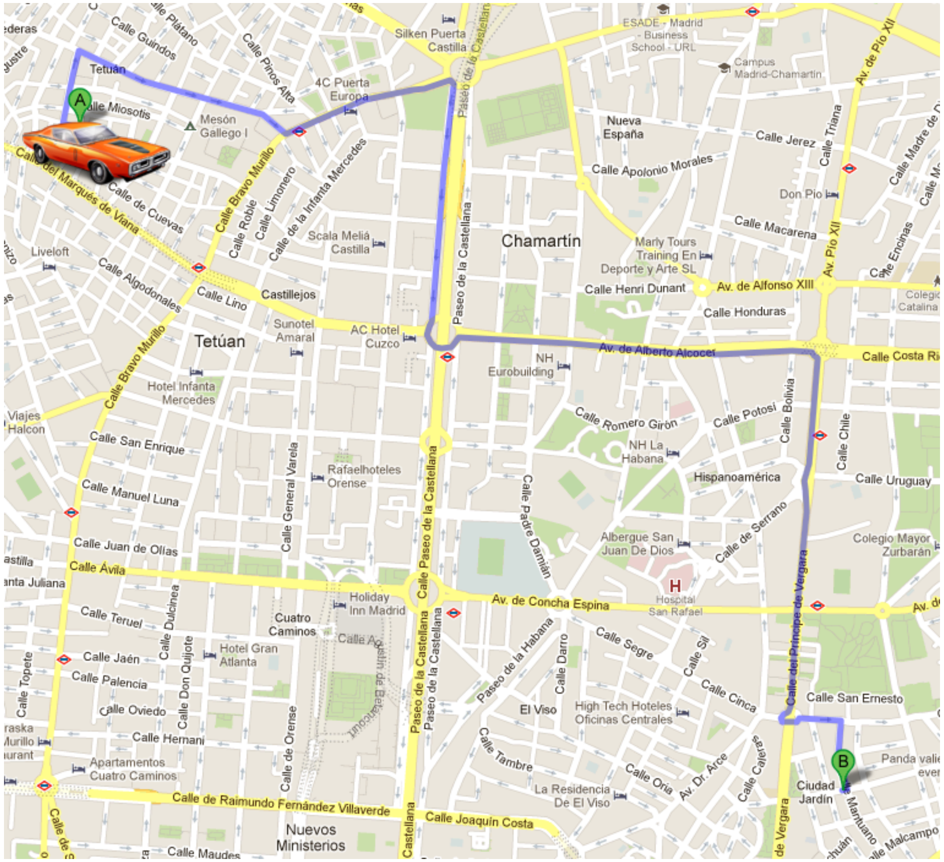
- Motivation
- The data streaming philosophy
- System Model
- Sample Data Streaming application
- Evolution of Stream Processing Engines
- Challenges in the context of Smart Grids (some examples)

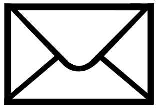
Database vs. Data Streaming

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

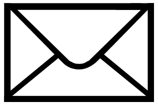
- How will a “database” and “data streaming” approach this?

Database vs. Data Streaming





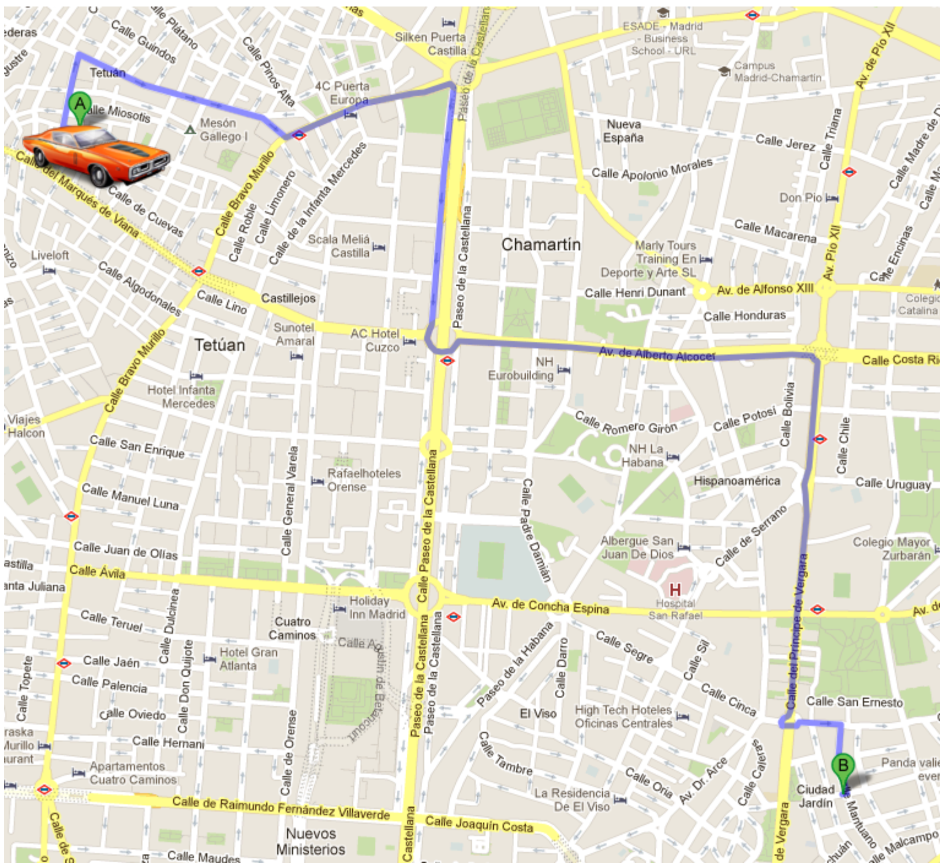
Start time
Position A



End time
Position B

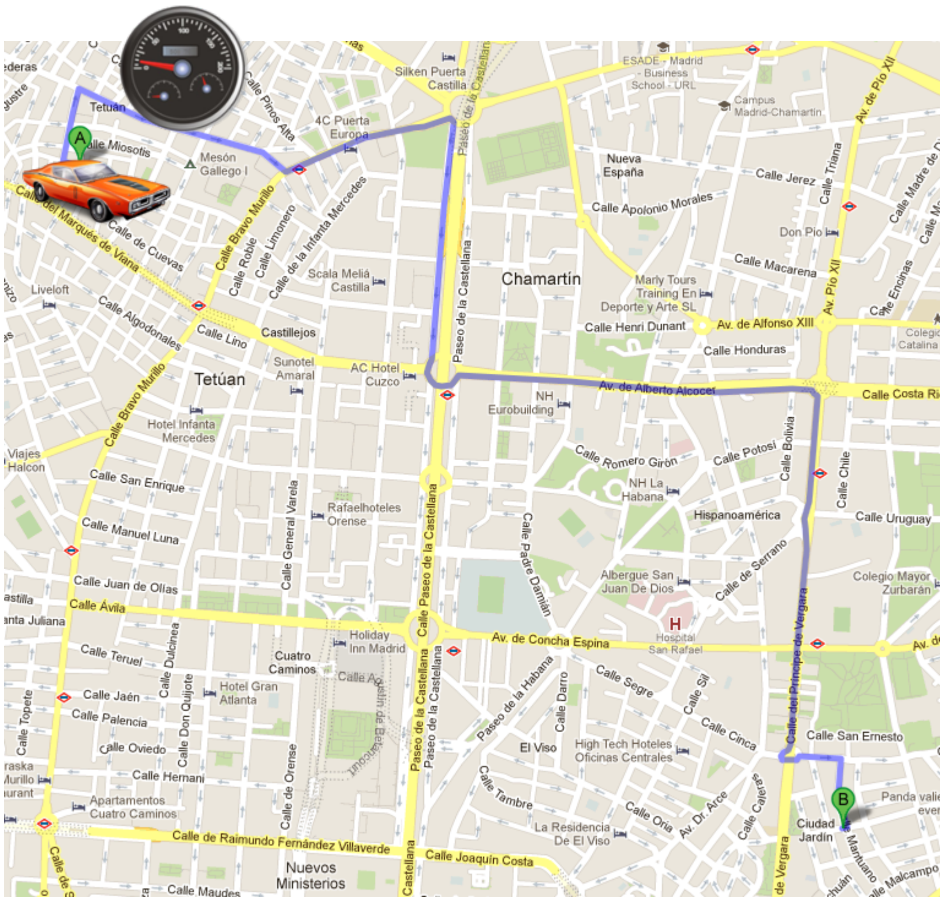
$$\frac{distance(A, B)}{End\ time\ - Start\ Time}$$

<u>Database</u> vs. Data Streaming



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

Database vs. Data Streaming



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

Agenda

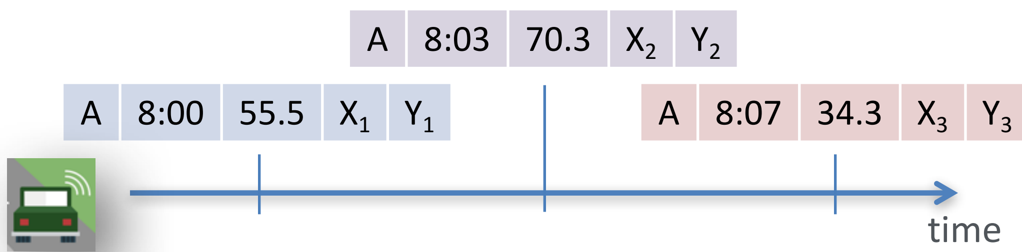
- Motivation
- The data streaming philosophy
- System Model
- Sample Data Streaming application
- Evolution of Stream Processing Engines
- Challenges in the context of Smart Grids (some examples)

data stream: unbounded sequence of tuples sharing the same schema

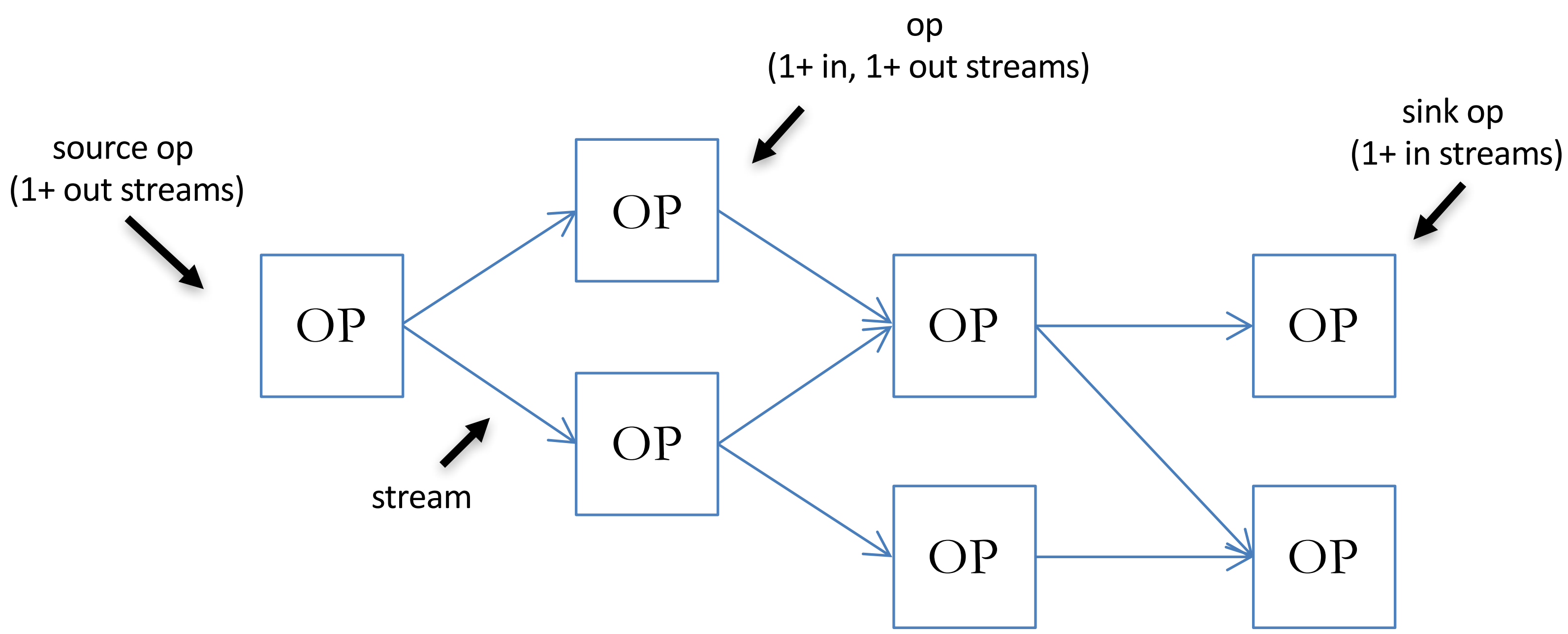
Example: vehicles' speed reports

Field	Field
vehicle id	text
time (secs)	text
speed (Km/h)	double
X coordinate	double
Y coordinate	double

Let's assume each source (e.g., vehicle) produces and delivers a timestamp sorted stream



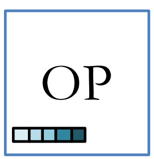
continuous query (or simply **query**): Directed Acyclic Graph (DAG) of streams and operators



data streaming **operators**

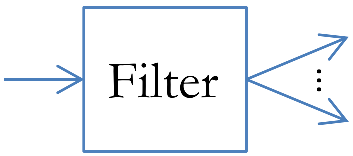
Two main types:

- Stateless operators
  - do not maintain any state
  - one-by-one processing
  - if they maintain some state, such state does not evolve depending on the tuples being processed

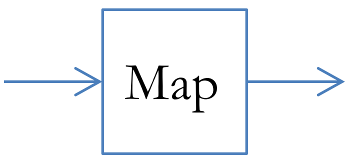


- Stateful operators
  - maintain a state that evolves depending on the tuples being processed
  - produce output tuples that depend on multiple input tuples

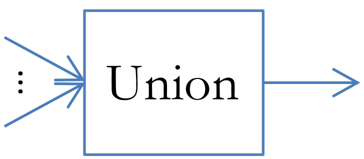
stateless operators



Filter / route tuples based on one (or more) conditions



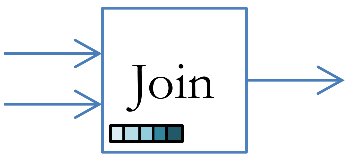
Transform each tuple



Merge multiple streams (with the same schema) into one

stateful operators

Aggregate information from multiple tuples
(e.g., max, min, sum, ...)

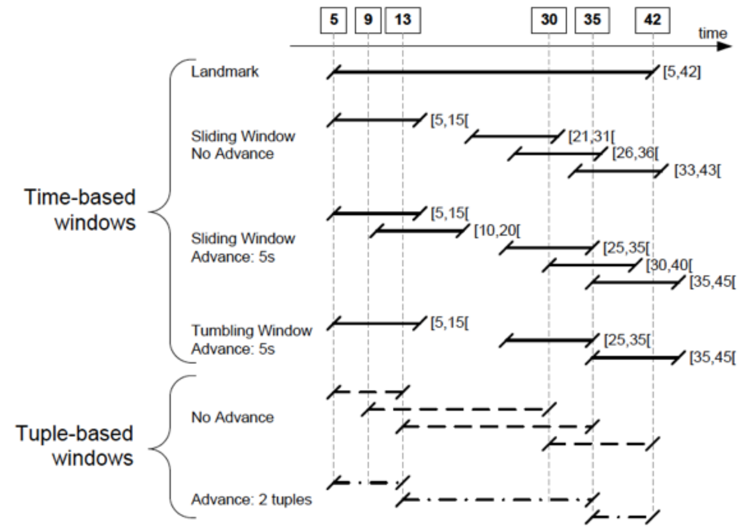


Join tuples coming from 2 streams given a certain predicate

Wait a moment!

if streams are unbounded, how can we aggregate or join?

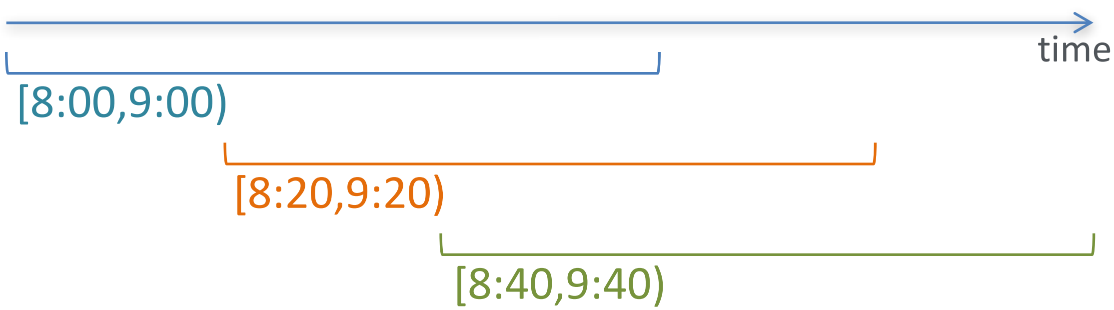
windows and **stateful analysis [18]**



Stateful operations are done over windows:

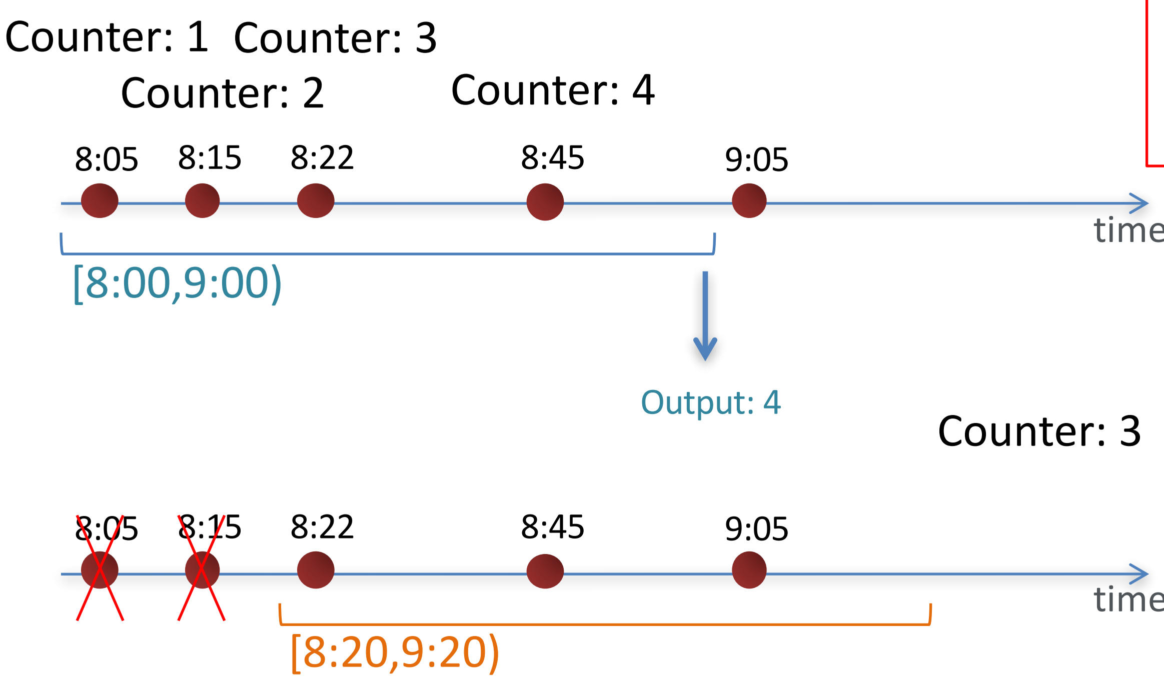
- Time-based (e.g., tuples in the last 10 minutes)
- Tuple-based (e.g., given the last 50 tuples)

Usually applications rely on time-based sliding windows

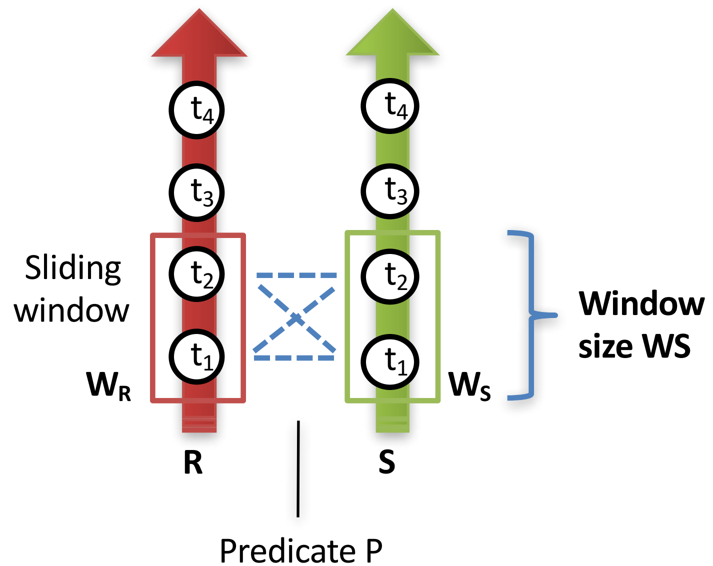


time-based sliding window aggregation (count)

we assumed each source produces and delivers a timestamp sorted stream! What happens if this is not the case?



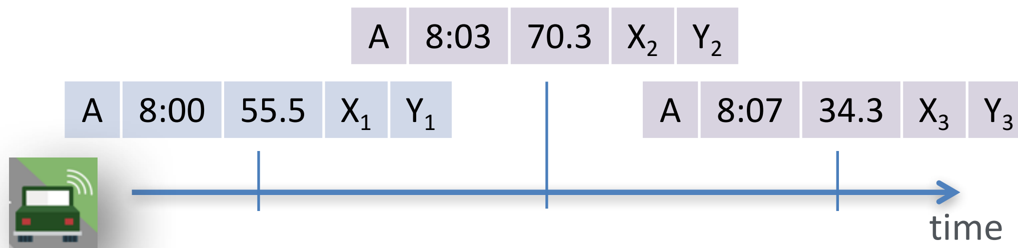
time-based sliding window joining



Agenda

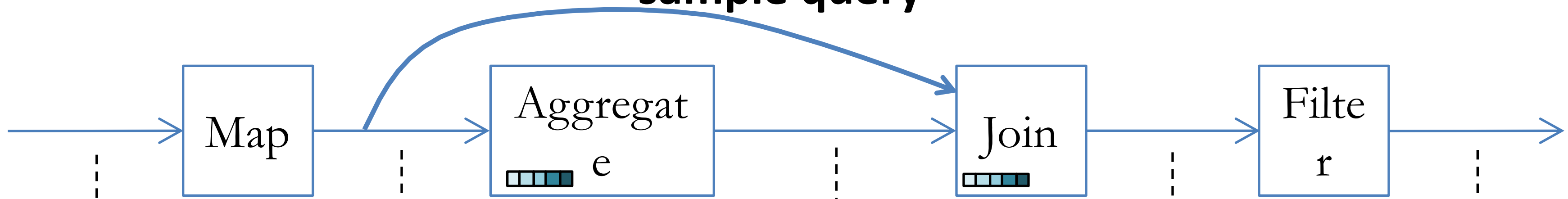
- Motivation
- The data streaming philosophy
- System Model
- Sample Data Streaming application
- Evolution of Stream Processing Engines
- Challenges in the context of Smart Grids (some examples)

sample query



For each vehicle, raise an alert if the speed of the latest report is more than 2 times higher than its average speed in the last 30 days.

sample query



Remove unused fields

Compute average speed for each vehicle during the last 30 days

Join on vehicle id

Check condition

Field
vehicle id
time (secs)
speed (Km/h)
X coordinate
Y coordinate

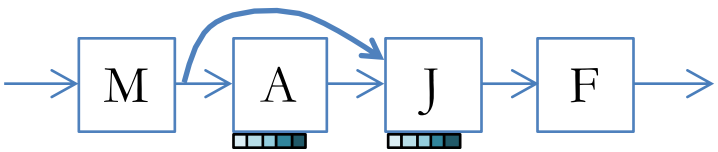
Field
vehicle id
time (secs)
speed (Km/h)

Field
vehicle id
time (secs)
avg speed (Km/h)

Field
vehicle id
time (secs)
avg speed (Km/h)
speed (Km/h)

Field
vehicle id
time (secs)
speed (Km/h)

sample query



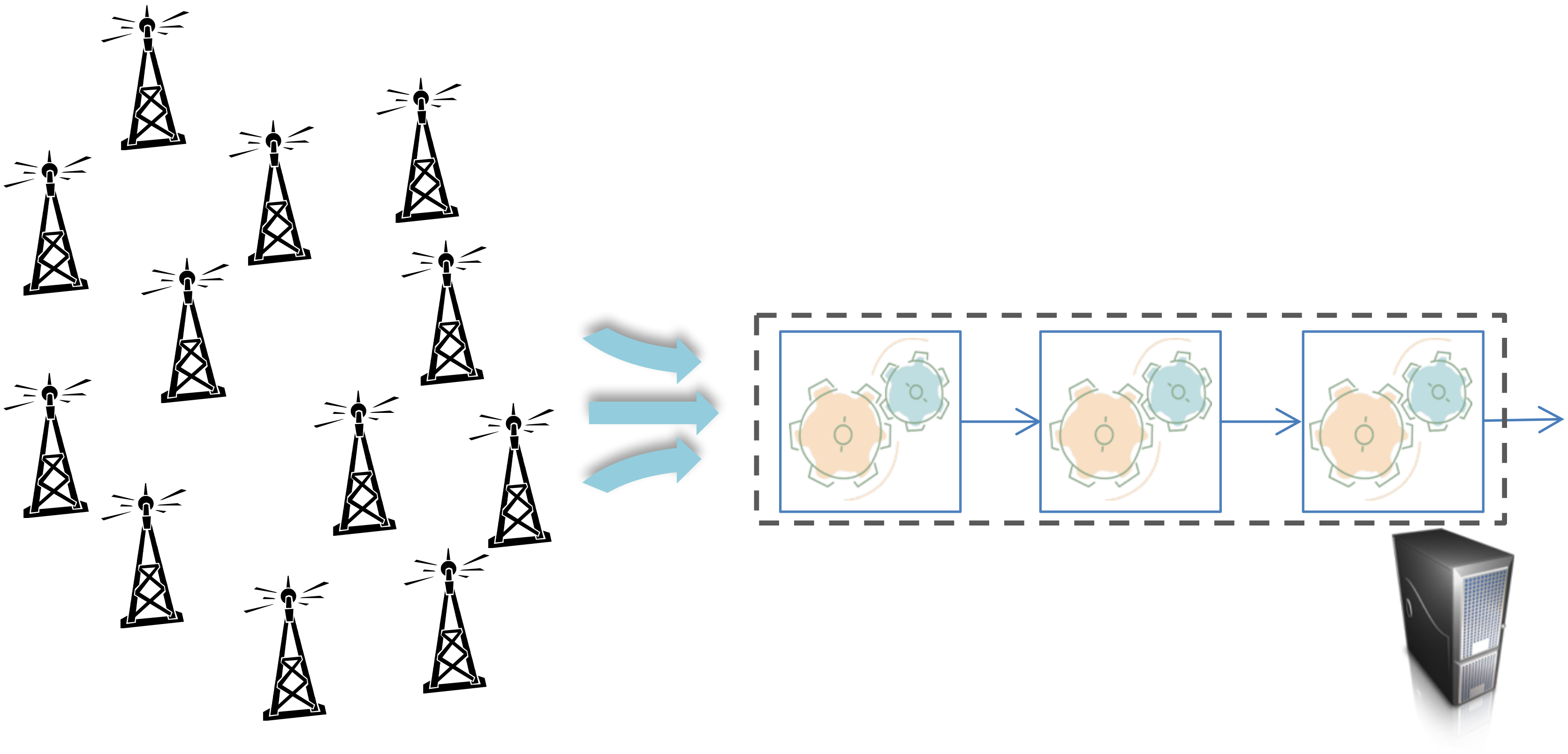
Notice:

- the same semantics can be defined in several ways (using different operators and composing them in different ways)
- Using many basic building blocks can ease the task of distributing and parallelizing the analysis (more in the following...)

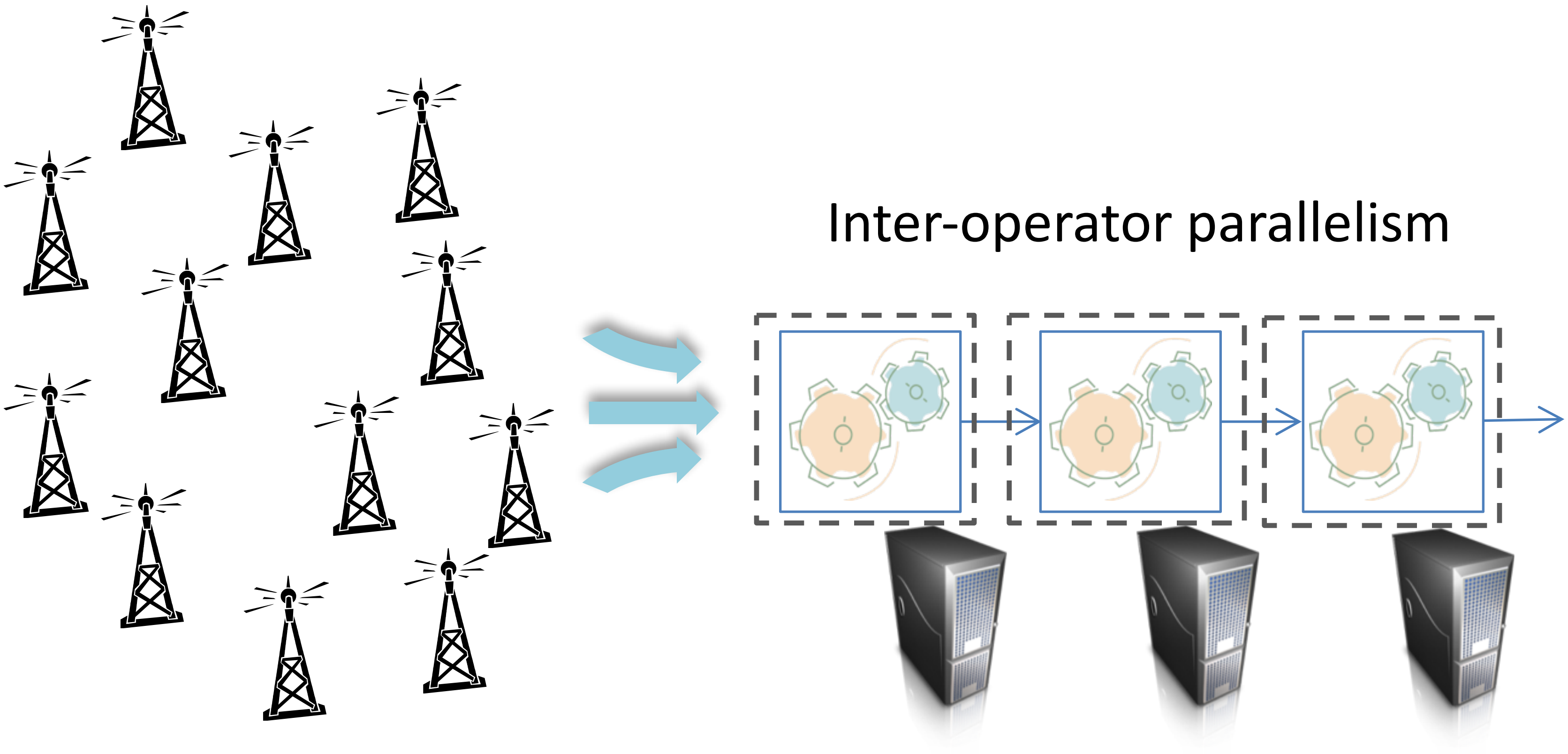
Agenda

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

Centralized SPEs

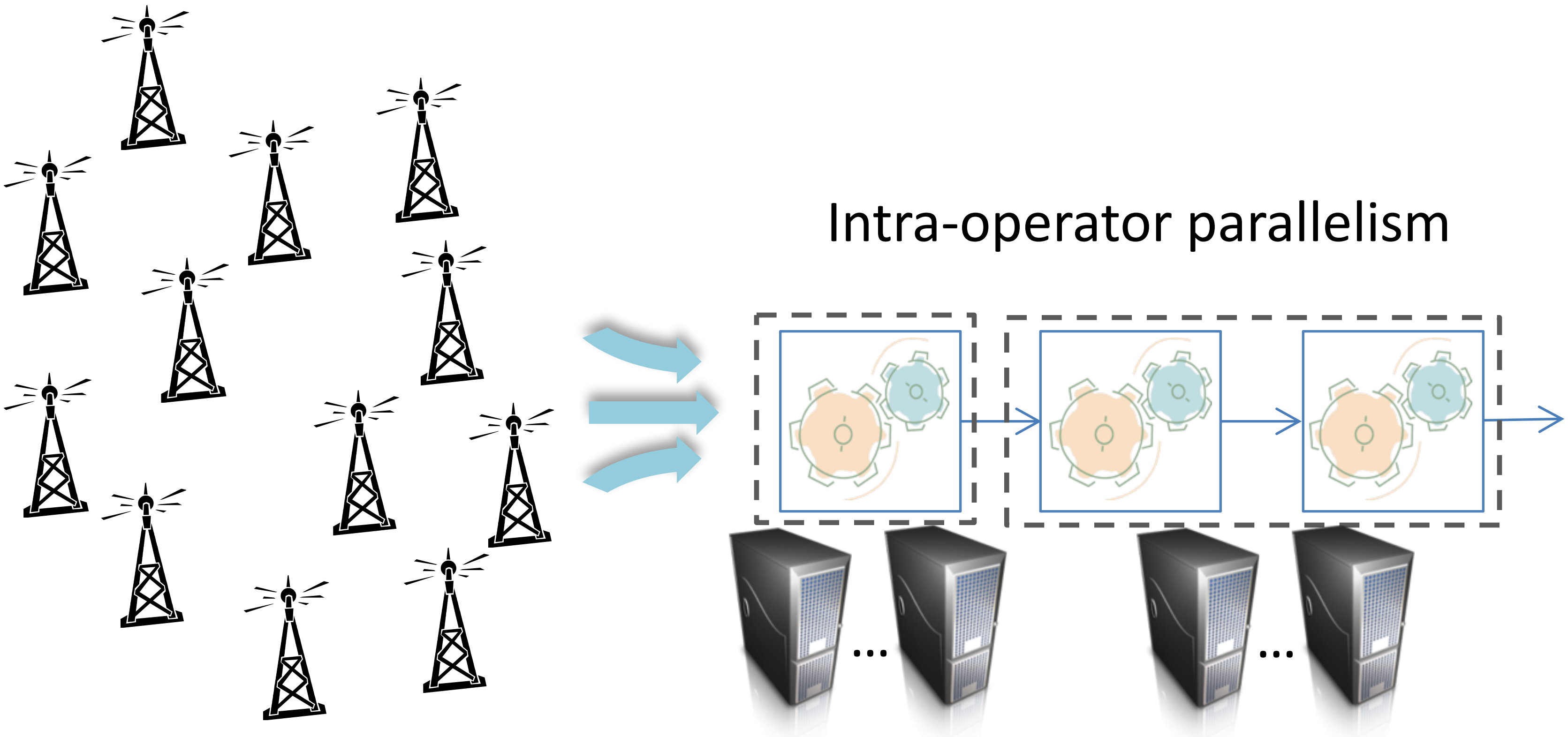


Distributed SPEs



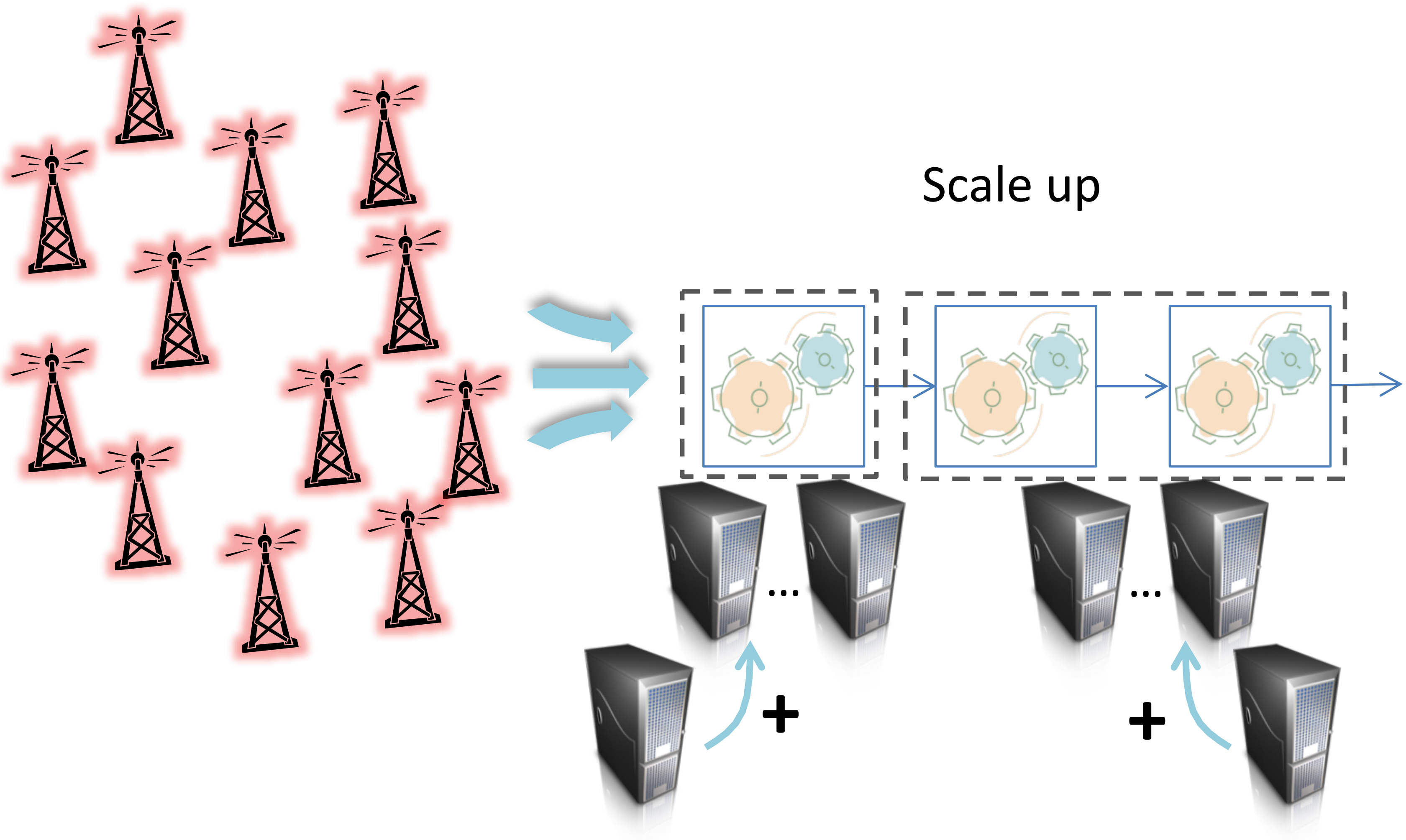
Parallel SPEs

Intra-operator parallelism

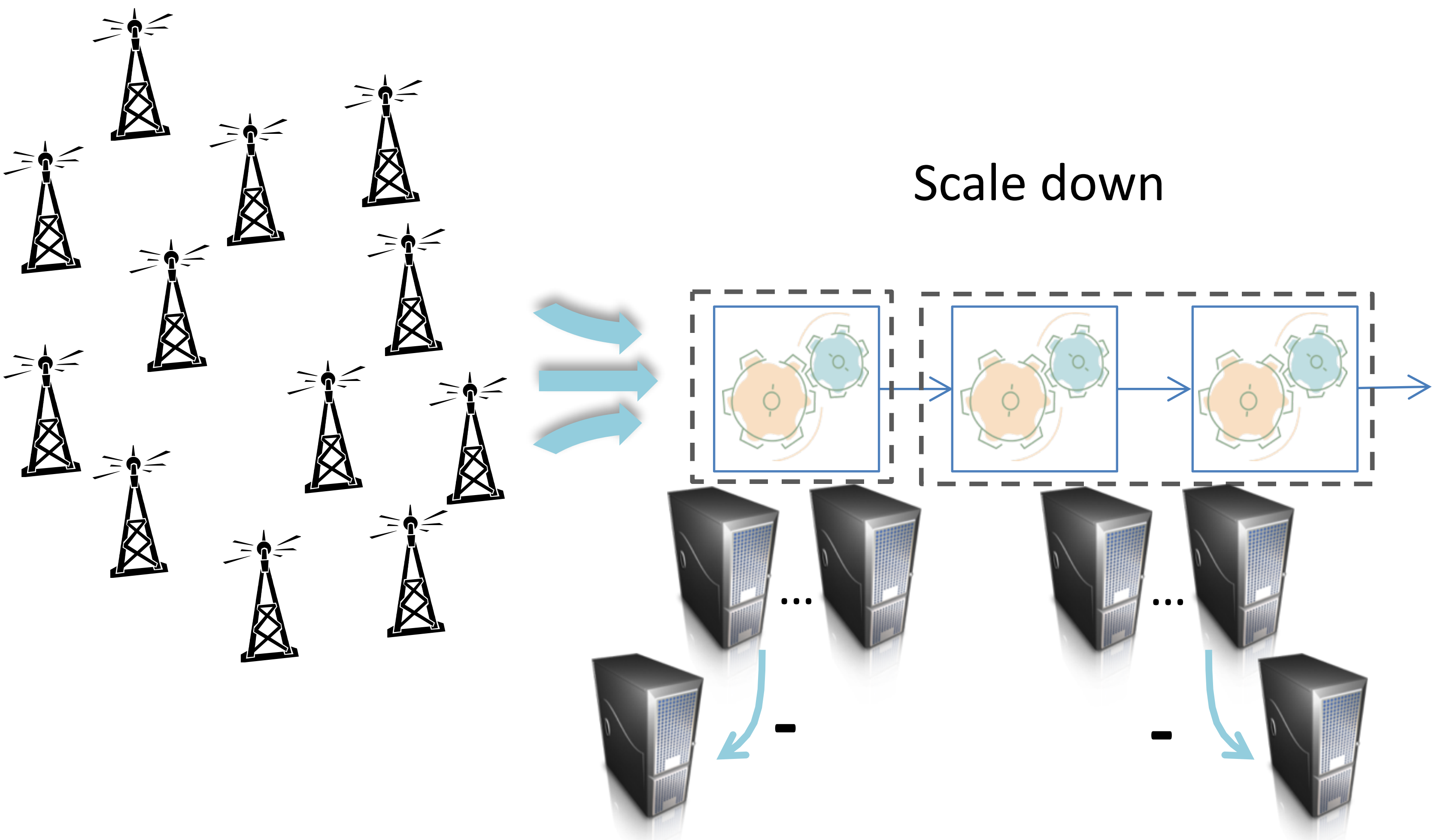


Over-provisioning or under-provisioning?

Elastic SPEs



Elastic SPEs



Agenda

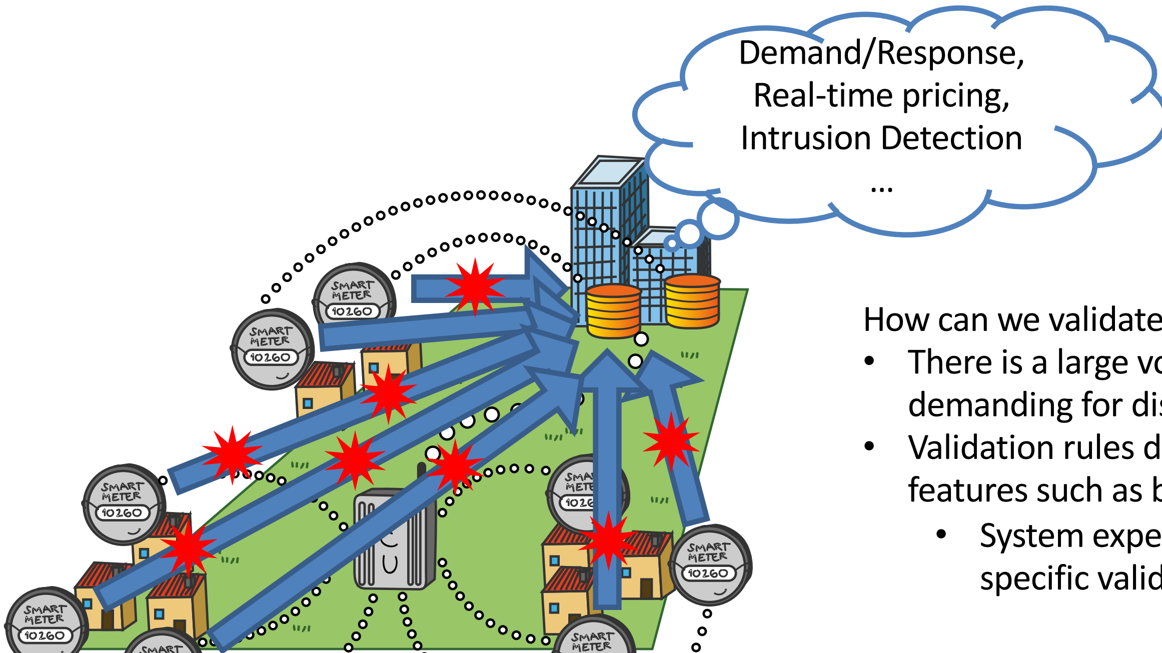
- Motivation
- The data streaming philosophy
- System Model
- Sample Data Streaming application
- Evolution of Stream Processing Engines
- Challenges in the context of Smart Grids (some examples)

Online and Scalable Data Validation in Advanced Metering Infrastructures

Vincenzo Gulisano, Magnus Almgren
and Marina Papatriantafilou

Chalmers University
of technology

Online and Scalable Data Validation in Advanced Metering Infrastructures



Noisy and Lossy data: bad-calibrated / faulty devices, lossy communication, malicious users, …

How can we validate data given that…

- There is a large volume of *continuous* data demanding for distributed and parallel analysis
- Validation rules depend on installation-specific features such as brands, devices, protocols, …
    - System experts should define installation-specific validation rules?

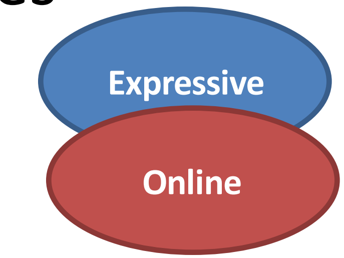
Why Streaming-based data Validation?

Expressive

Online

Parallel &
Distributed

Sample Streaming-based Data Validation: Interpolate missing consumption values



Field
Device
Time
Consumption

Aggregate: Match consecutive readings from each meter

Field
Device
Time$_1$
Consumption$_1$
Time$_2$
Consumption$_2$

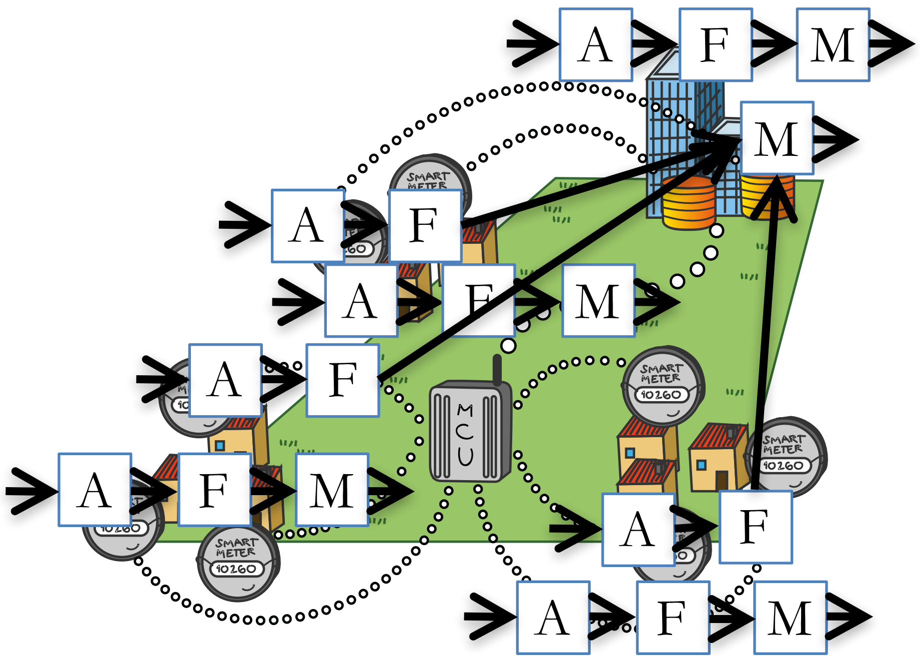
Filter: Forward if time distance exceeds a certain threshold

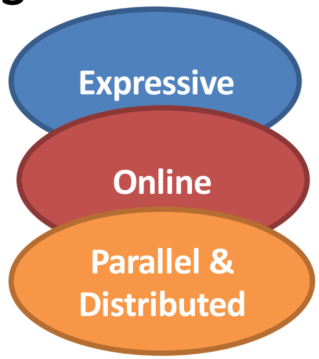
Field
Device
Time$_1$
Consumption$_1$
Time$_2$
Consumption$_2$

Map: Interpolate missing values

Field
Device
Time
Consumption

Sample Streaming-based Data Validation: Interpolate missing consumption values





METIS: a Two-Tier Intrusion Detection System for Advanced Metering Infrastructures

Vincenzo Gulisano, Magnus Almgren and Marina Papatriantafilou

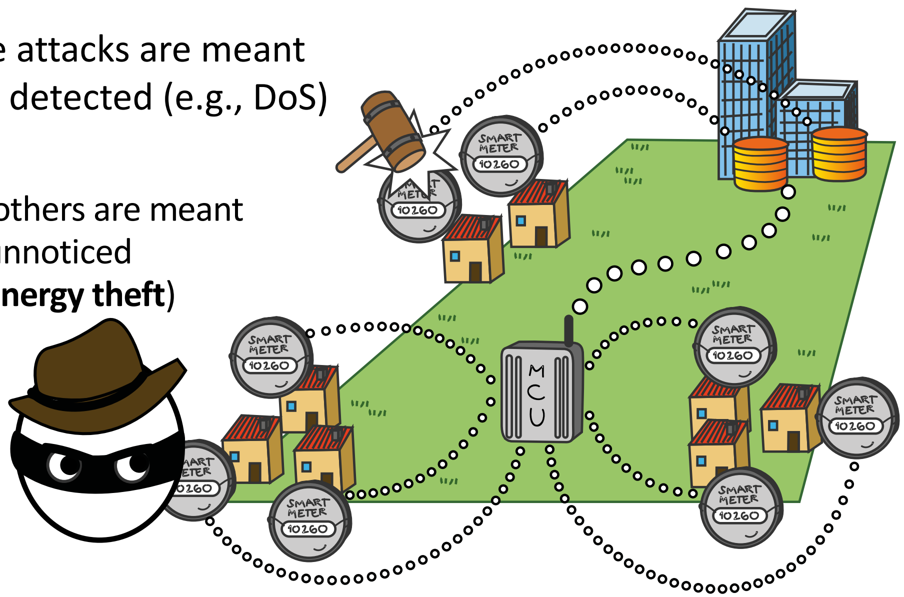
Chalmers University
of technology

Why METIS?

Advanced Metering Infrastructures (AMIs)

Some attacks are meant to be detected (e.g., DoS)

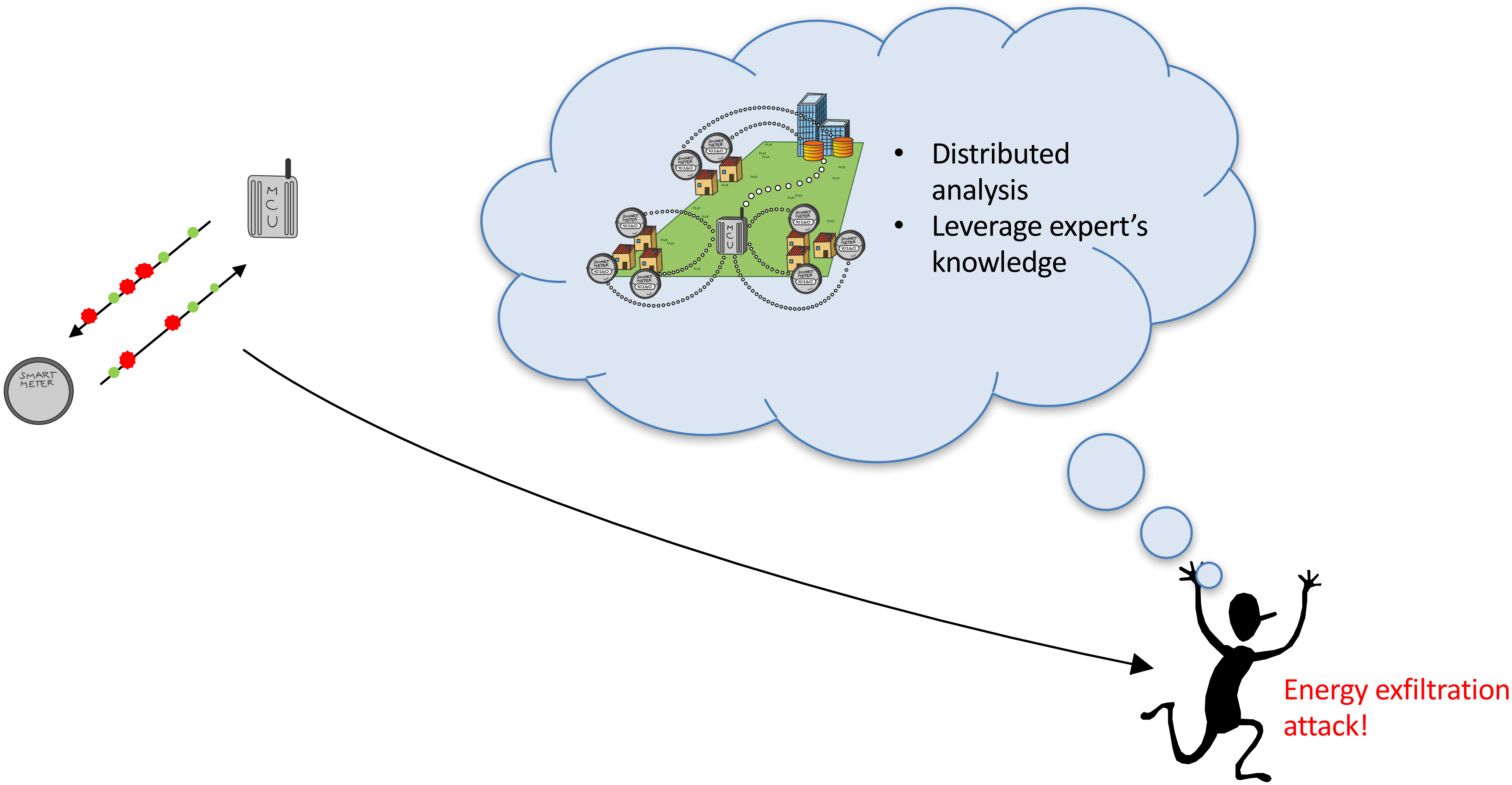
Some others are meant to go unnoticed (e.g., **energy theft**)



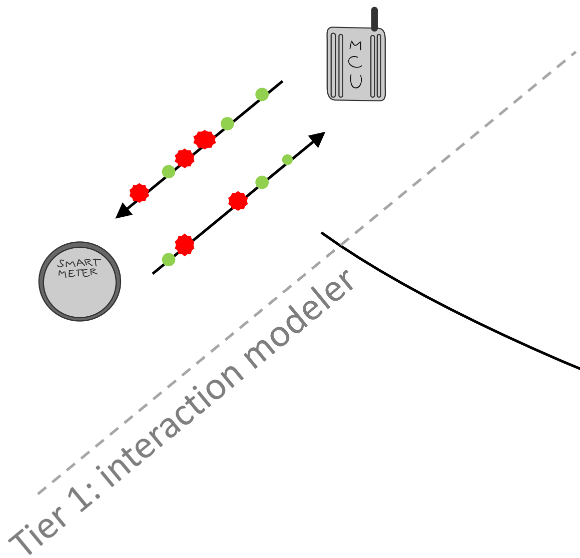
How can we detect them given that…

- … there is a large volume of *continuous* data demanding for distributed and parallel analysis
- … Most data is local to the devices
- … Such attacks are not documented
- … Each AMI relies on its brands, devices, protocols (i.e., system expert’s knowledge plays a key role)

METIS overview



METIS overview



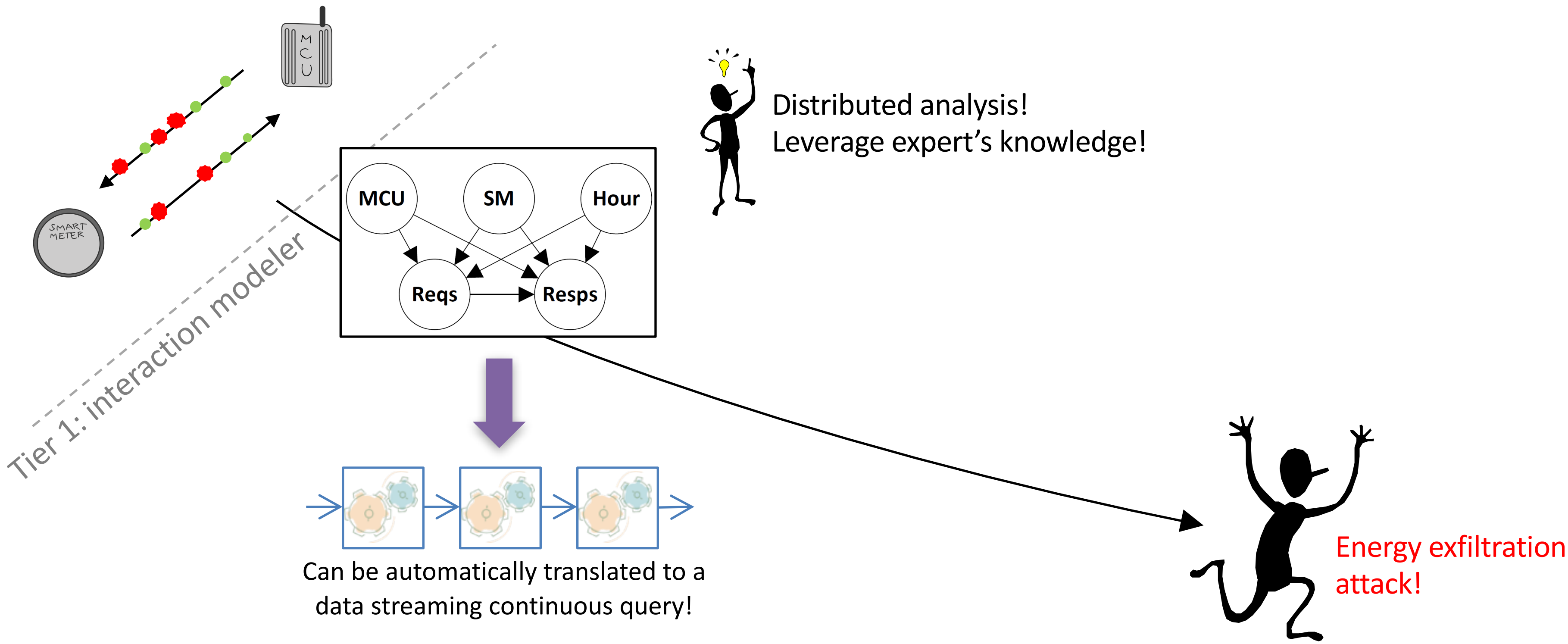
AMIs evolve slowly and have limited heterogeneity.
Let's learn which messages are expected and which are not!

What influences an expected message?

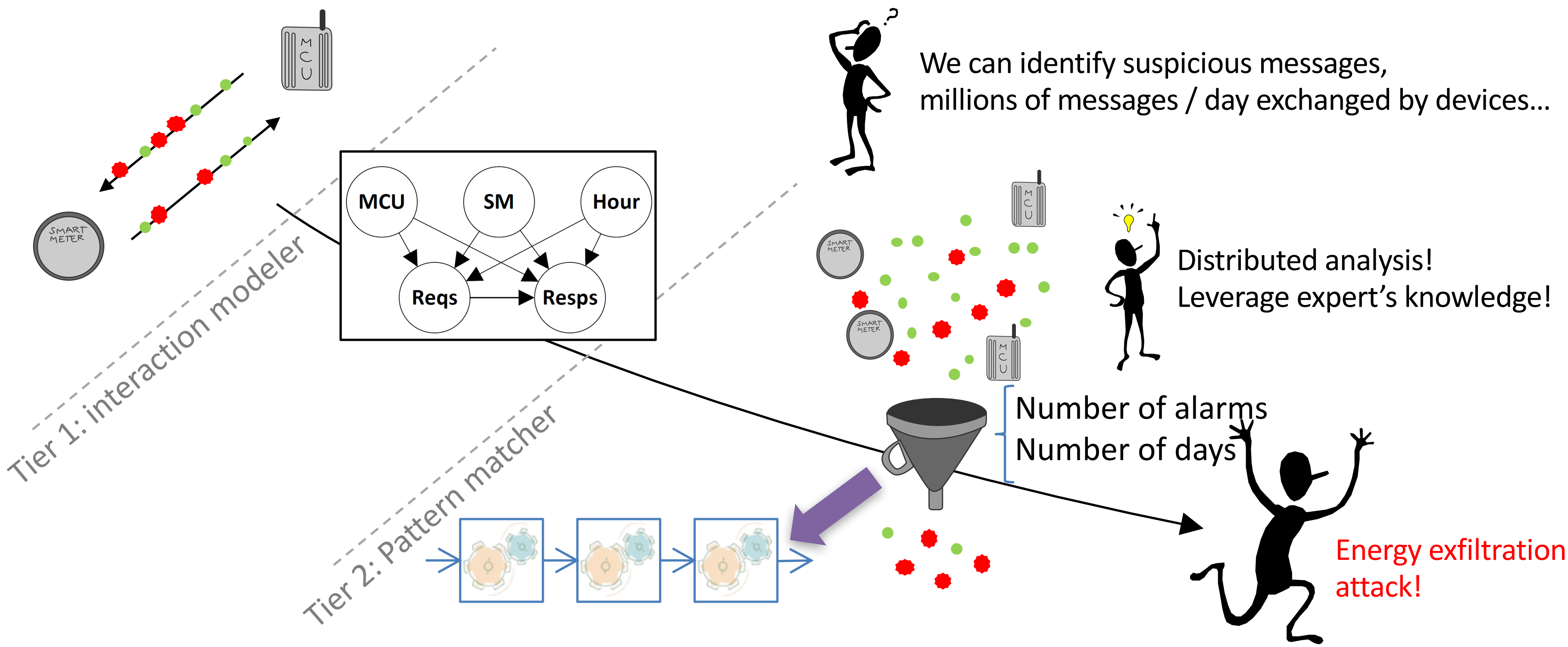
Let's use a Bayesian Network!

Energy exfiltration attack!

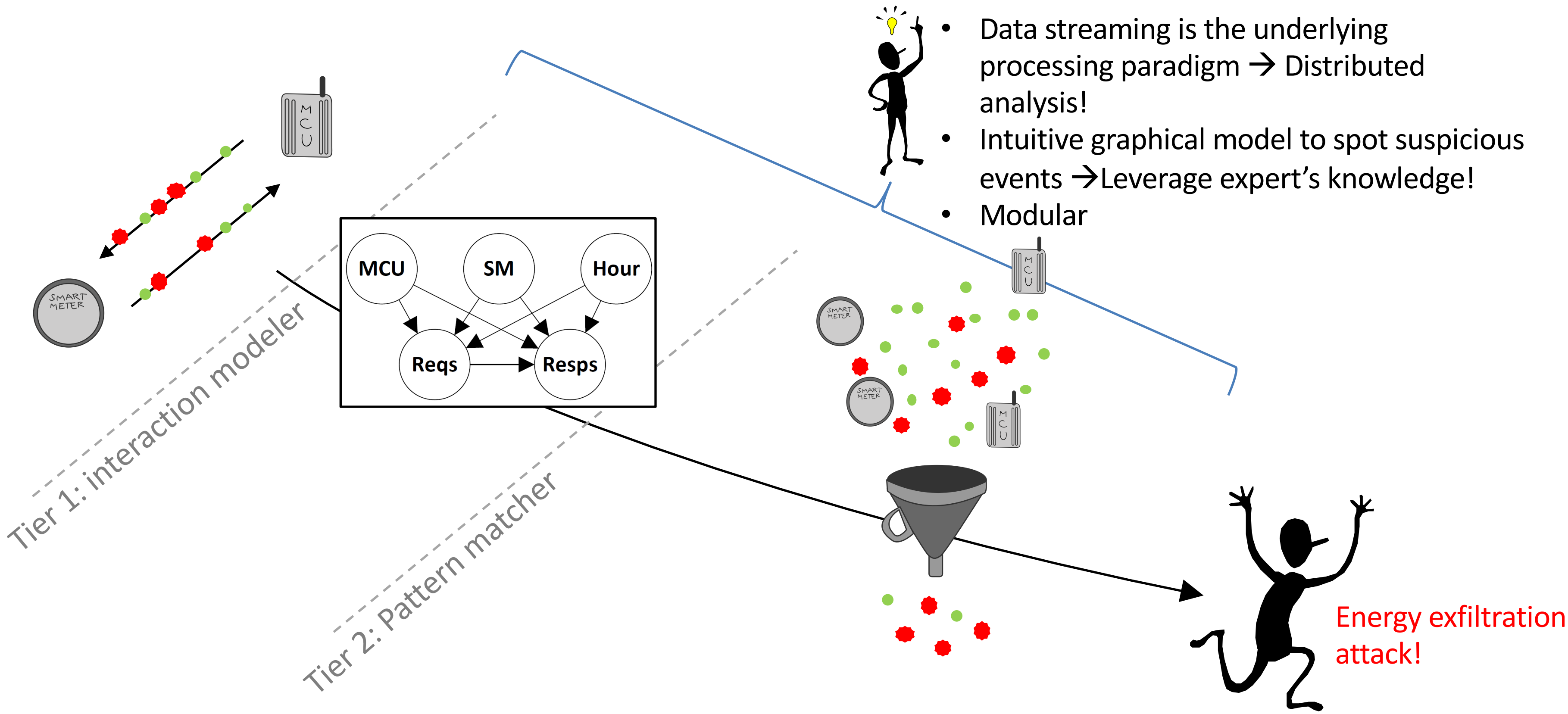
METIS overview



METIS overview



METIS overview



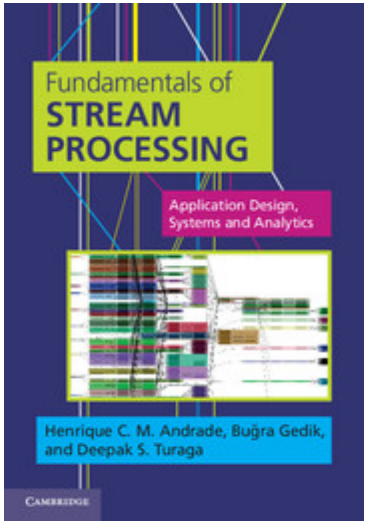
- Data streaming is the underlying processing paradigm → Distributed analysis!
- Intuitive graphical model to spot suspicious events →Leverage expert's knowledge!
- Modular

An overview of Data Streaming

Questions?

An overview of Data Streaming

- Something to read:



http://www.cambridge.org/se/academic/subjects/engineering/communications-and-signal-processing/fundamentals-stream-processing-application-design-systems-and-analytics

Bibliography

1. Online and Scalable Data Validation in Advanced Metering Infrastructures. Vincenzo Gulisano, Magnus Almgren, Marina Papatriantafilou. The 5th IEEE PES Innovative Smart Grid Technologies (ISGT) European 2014 Conference
2. METIS: a Two-Tier Intrusion Detection System for Advanced Metering Infrastructures. Vincenzo Gulisano, Magnus Almgren, Marina Papatriantafilou. 10th International Conference on Security and Privacy in Communication Networks (SecureComm) 2014
3. 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.
4. 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.
5. Michael Stonebraker, Uˇgur Çetintemel, and Stan Zdonik. The 8 requirements of realtime stream processing. SIGMOD Rec., 34(4), December 2005.
6. 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.
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.
9. Arvind Arasu, Shivnath Babu, and Jennifer Widom. The CQL continuous query language: semantic foundations and query execution. The VLDB Journal, 15(2), June 2006.
10. 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.