Vienna, Austria - 10.11.2015

Stream Reasoning:
mastering the velocity and the variety dimensions of Big Data at once

Emanuele Della Valle

DEIB - Politecnico di Milano

@manudellavalle

emanuele.dellavalle@polimi.it

http://emanueledellavalle.org

It's a streaming world …

- Off-shore oil operations

- Smart Cities

- Global Contact Center

- Social networks

- Generate data streams!

E. Della Valle, S. Ceri, F. van Harmelen, D. Fensel **It's a Streaming World! Reasoning upon Rapidly Changing Information.** IEEE Intelligent Systems 24(6): 83-89 (2009)

… looking for reactive answers …

- What is the expected time to failure when that turbine's barring starts to vibrate as detected in the last 10 minutes?

- Is public transportation where the people are?

- Who are the best available agents to route all these unexpected contacts about the tariff plan launched yesterday?

- Who is driving the discussion about the top 10 emerging topics ?

- Require continuous processing and reactive answer

… and many more conflicting requirements

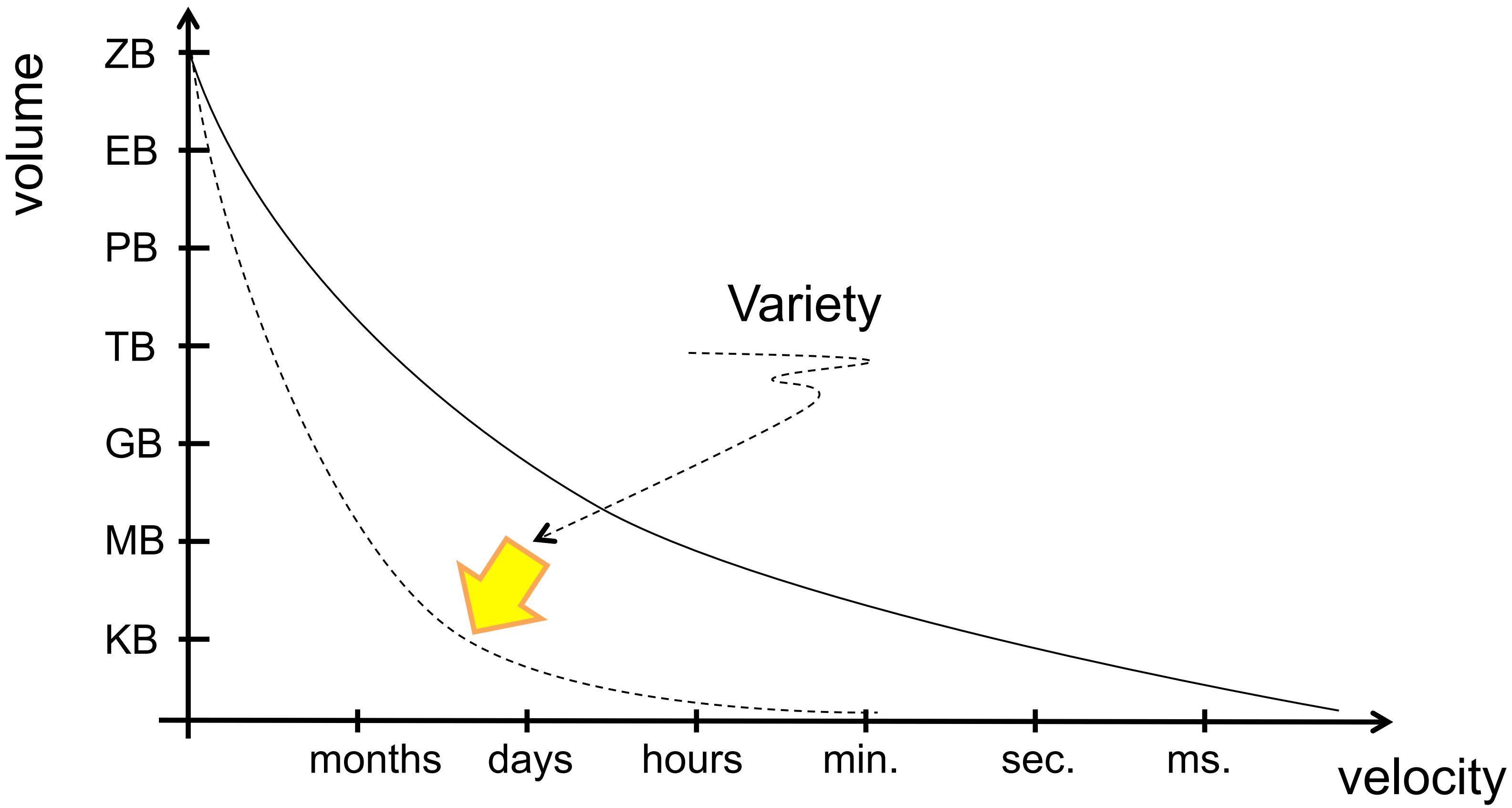
A system able to answer those queries must be able to

	Volume	Velocity	Variety	Veracity
• handle **massive datasets**	x			
• process **data streams** on the fly		x		
• cope with **heterogeneous datasets**			x	
• cope with **incomplete data**			x	x
• cope with **noisy data**				x
• provide **reactive answers**		x		
• support **fine-grained access**	x	x		
• integrate **complex domain models**			x	

In **Big Data** terms → Volume, Velocity, Variety, Veracity

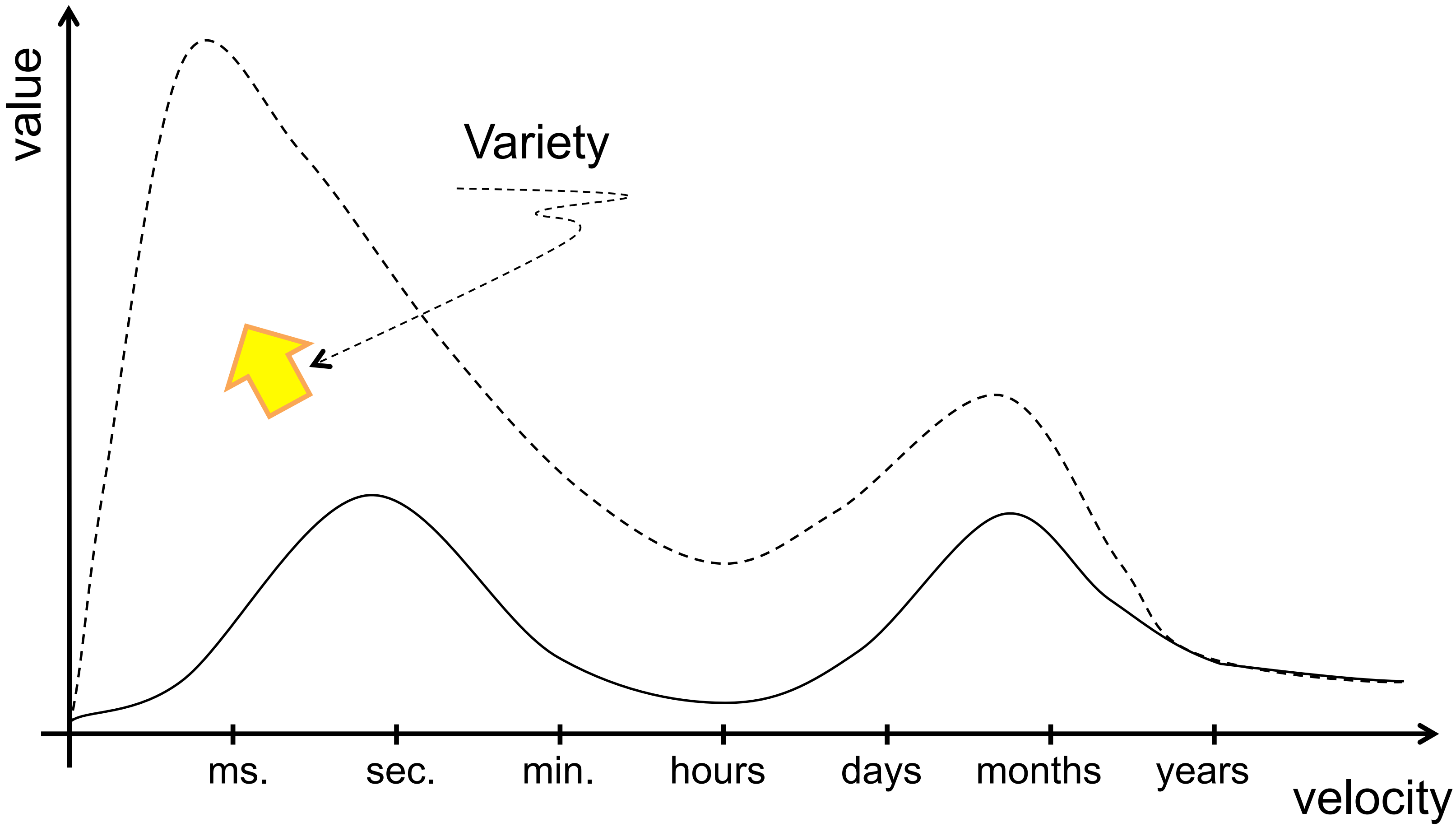
Grand challenge

- Volume + Velocity + Variety = hard deal



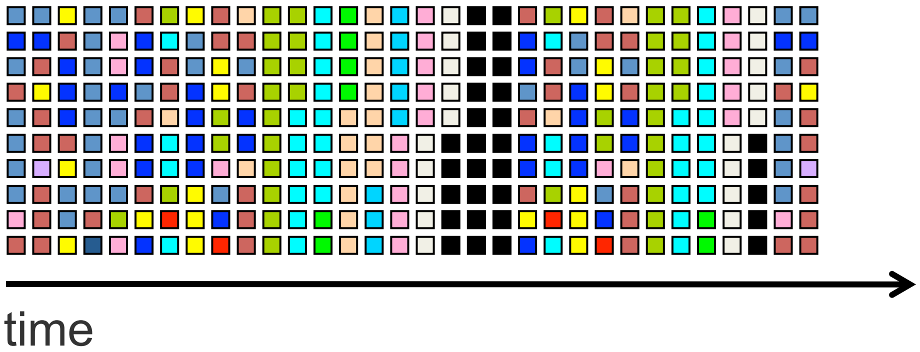
A good reason to embrace it!

- ++ Variety → ++ value



From challenges to opportunities

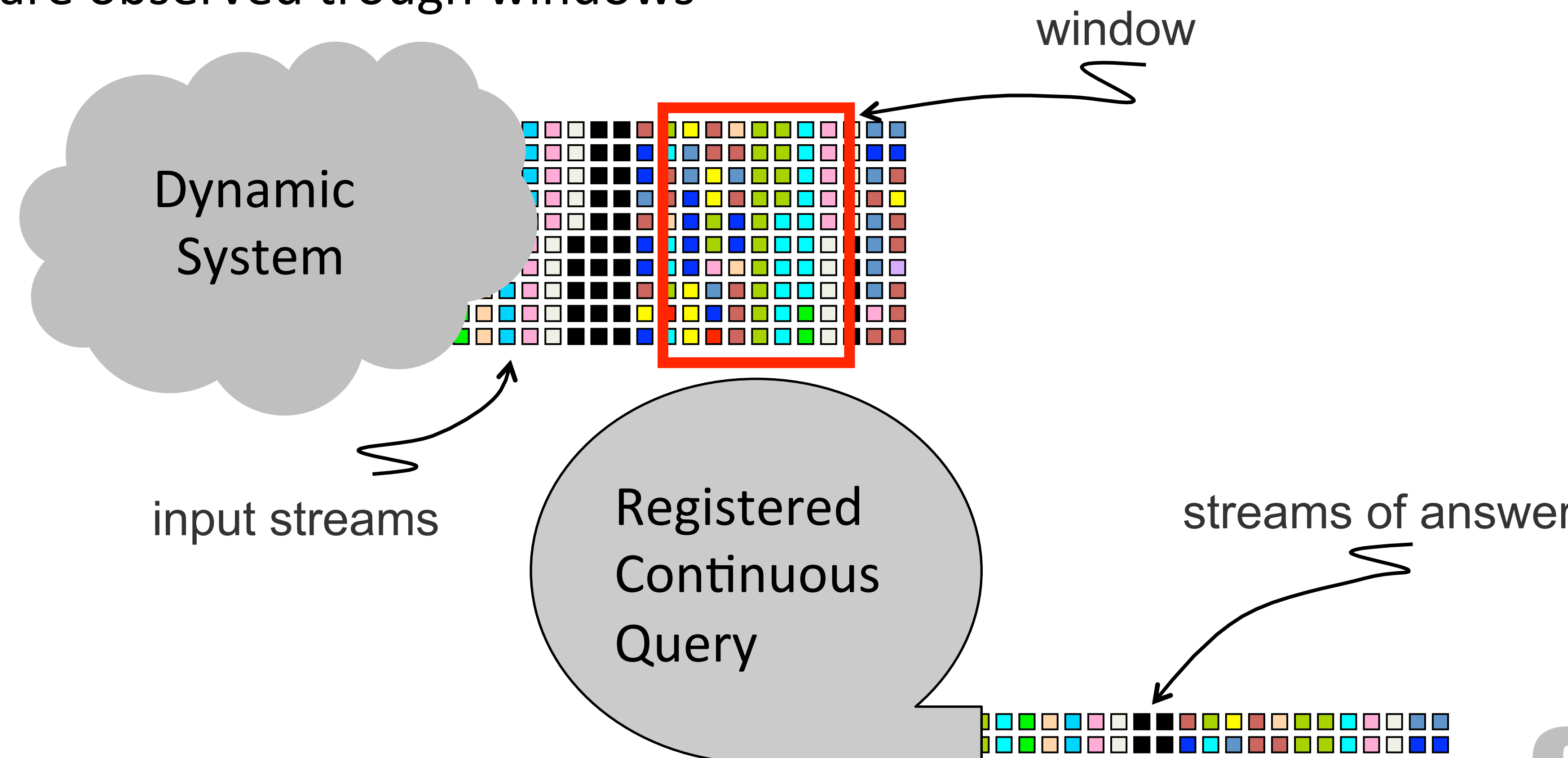
- Formally data streams are :
  - **unbounded** sequences of time-varying data elements



- Less formally, in many application domains, they are:
  - a "continuous" flow of information
  - where **recent information is more relevant** as it describes the current state of a dynamic system
- Opportunities
  - **Forget old enough information**
  - **Exploit** the implicit **ordering** (by recency) in the data

State-of-the-art: DSMS and CEP

- **A paradigmatic change!**
- Continuous queries registered over streams that are observed trough windows



DSMS and CEP vs. requirements

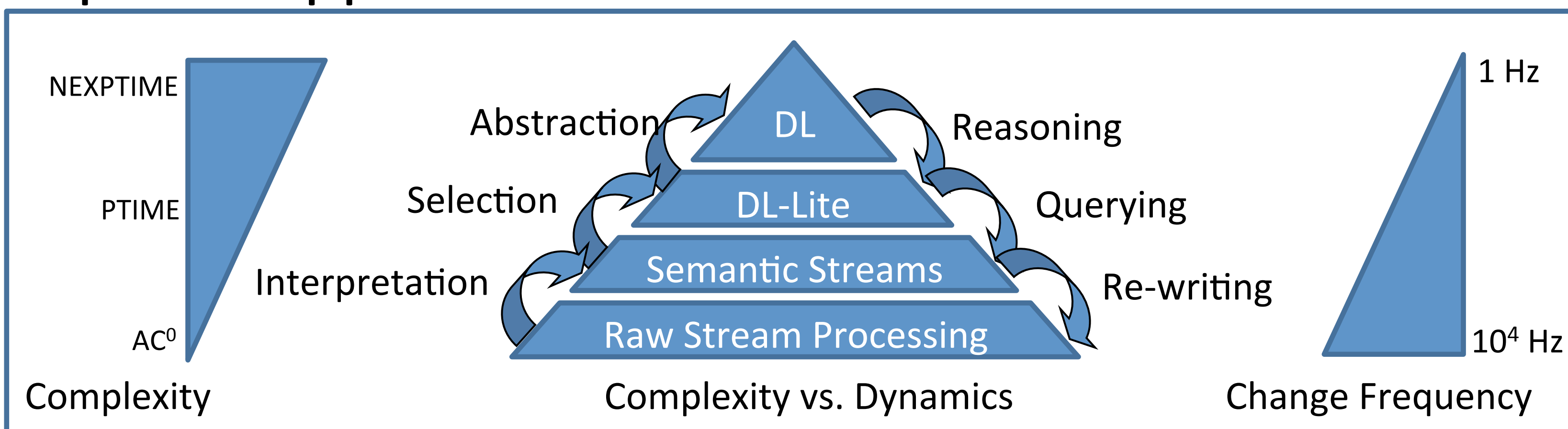
Requirement	DSMS CEP
massive datasets	✓
data streams	✓
heterogeneous dataset	✗
incomplete data	✗
noisy data	✓
reactive answers	✓
fine-grained information access	✓
complex domain models	✗

DSMS/CEP,OBDA vs. requirements

Requirement	DSMS CEP	OBDA
massive datasets	✓	✓
data streams	✓	✗
heterogeneous dataset	✗	✓
incomplete data	✗	✓
noisy data	✓	✗
reactive answers	✓	✗
fine-grained information access	✓	✓
complex domain models	✗	✓

Stream Reasoning

- Research question
  - is it possible to **make sense in real time of multiple**, **heterogeneous**, **gigantic** and inevitably **noisy** and **incomplete data streams** in order **to support** the **decision processes** of extremely large numbers of concurrent users?
- Proposed approach



H. Stuckenschmidt, S. Ceri, E. Della Valle, F. van Harmelen: **Towards Expressive Stream Reasoning.** Proceedings of the Dagstuhl Seminar on Semantic Aspects of Sensor Networks, 2010.

Sub-research questions

1. Is it possible **extend the Semantic Web stack** in order to represent heterogeneous data streams, continuous queries, and continuous reasoning tasks?
2. Does the ordered nature of data streams and the possibility to forget old enough information allow to **optimize continuous querying and continuous reasoning** tasks so **to provide reactive answers** to large number of concurrent users without forsaking correctness or completeness?
3. Can Semantic Web and Machine Learning technologies be jointly employed to **cope with the noisy and incomplete** nature of **data** streams?
4. Are there **practical cases** where processing data stream at semantic level is the best choice?

Sub-research questions

1. Is it possible **extend the Semantic Web stack** in order to represent heterogeneous data streams, continuous queries, and continuous reasoning tasks?
2. Does the ordered nature of data streams and the possibility to forget old enough information allow to **optimize continuous querying and continuous reasoning** tasks so **to provide reactive answers** to large number of concurrent users without forsaking correctness or completeness?
3. Can Semantic Web and Machine Learning technologies be jointly employed to **cope with the noisy and incomplete** nature of **data** streams?
4. Are there **practical cases** where processing data stream at semantic level is the best choice?

Contribution: RDF stream Models

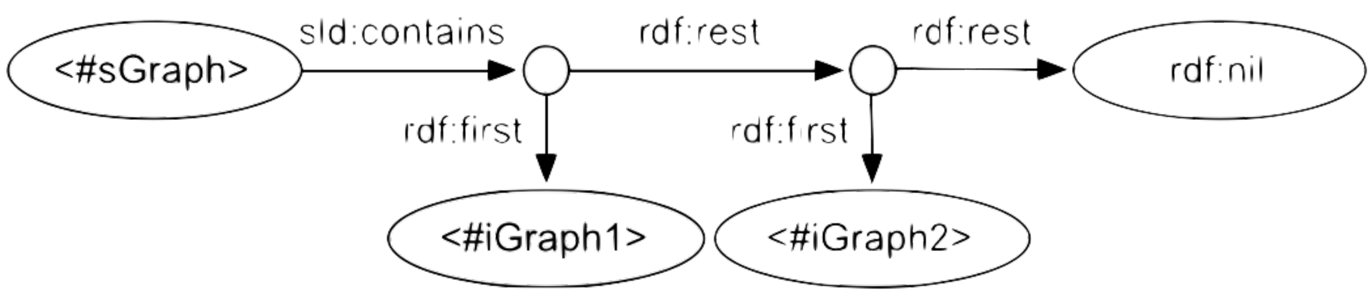
- **RDF Stream** (the C-SPARQL way)
  - **Unbound sequence of time-varying triples**
  - each represented by a pair made of an RDF triple and its timestamp
  - Timestamp are non-decreasing (allowing for simultaneity)

subject	predicate	object	**timestamp**
	…		
@BarakObama	posts	"Four more years",	8:16PM 6 Nov 2012
@Alice	posts	"RT: Four more years",	8:17PM 6 Nov 2012
	…		

D.F. Barbieri, D. Braga, S. Ceri, E. Della Valle, M. Grossniklaus**: Querying RDF streams with C-SPARQL.** SIGMOD Record 39(1): 20-26 (2010)

Contribution: RDF stream Models

- **RDF Stream** (the Streaming Linked Data way)
  - **Unbound sequence of time-varying graphs**
  - each represented by a pair made of an RDF graph and its timestamp
  - Timestamps (if present) are monotonically increasing
  - Graphs act as a form of punctuation (all triples in a graph are simultaneous)



D.F. Barbieri, E. Della Valle: **A Proposal for Publishing Data Streams as Linked Data - A Position Paper**. LDOW (2010)

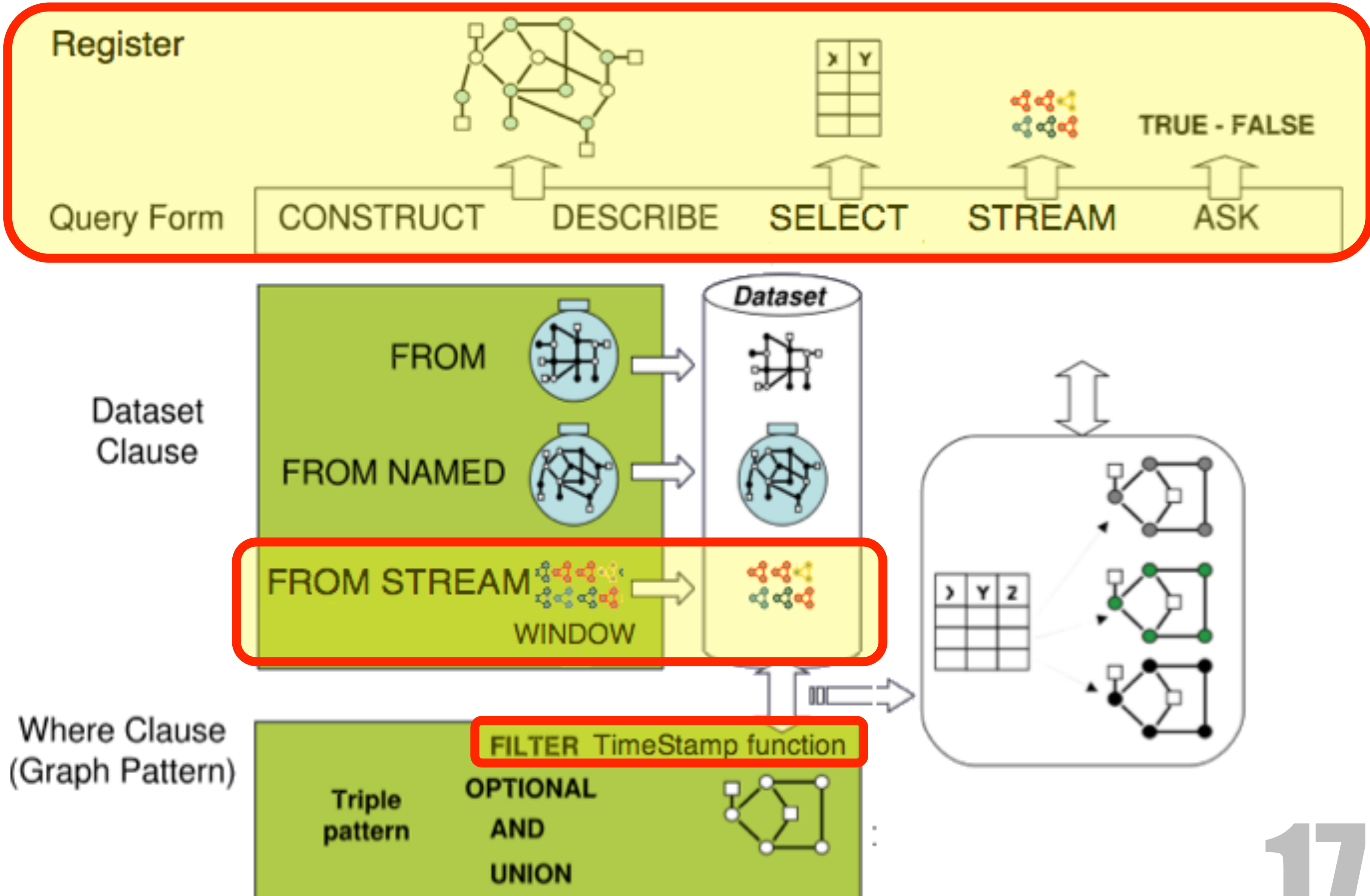
Work in progress

- In 2013, an RDF Stream Processing (RSP) community group was created at W3C

  http://www.w3.org/community/rsp/

- RSP data model and serialization
  – https://github.com/streamreasoning/RSP-QL/blob/master/Serialization.md

Contribution: Continuous-SPARQL



Contribution: Continuous-SPARQL

Who are the opinion makers? i.e., the users who are likely to influence the behavior their followers

```
REGISTER STREAM OpinionMakers COMPUTED EVERY 5m AS
CONSTRUCT { ?opinionMaker sd:about ?resource }
FROM STREAM <http://…>  [RANGE 30m STEP 5m]
WHERE {
     ?opinionMaker ?opinion ?res .
     ?follower sioc:follows ?opinionMaker.
     ?follower ?opinion ?res.
     FILTER (cs:timestamp(?follower ?opinion ?res) >
             cs:timestamp(?opinionMaker ?opinion ?res) )
}
HAVING ( COUNT(DISTINCT ?follower) > 3 )
```

Contribution: Continuous-SPARQL

Who are the opinion makers? i.e., the users who are

l… …havior their fo…

Query registration (for continuous execution)

RDF Stream added as new ouput format

FROM STREAM clause

WINDOW

Builtin to access timestamps

```
REGISTER STREAM OpinionMakers COMPUTED EVERY 5m AS
CONSTRUCT { ?opinionMaker sd:about
FROM STREAM <http://…>  [RANGE 30m STEP 5m]
WHERE {
     ?opinionMaker ?opinion ?res .
     ?follower sioc:follows ?opinionMaker.
     ?follower ?opinion ?res.
     FILTER (cs:timestamp(?follower ?opinion ?res) >
             cs:timestamp(?opinionMaker ?opinion ?res) )
}
HAVING ( COUNT(DISTINCT ?follower) > 3 )
```

D.F. Barbieri, D. Braga, S. Ceri, E. Della Valle, M. Grossniklaus: **Querying RDF streams with C-SPARQL.** SIGMOD Record 39(1): 20-26 (2010)

Alternatives to C-SPARQL

- CQELS
  - What: STREAM clause, focus on new answer
  - Ref: Le-Phuoc, D., Dao-Tran, M., Xavier Parreira, J., & Hauswirth, M. A native and adaptive approach for unified processing of linked streams and linked data. In ISWC 2011, pages 370–388.
- $\text{SPARQL}_{\text{Stream}}$
  - What: window in the past, focus on RDF to Stream operators
  - Ref: Calbimonte, J.-P., Corcho, O., & Gray, A. J. G. Enabling ontology-based access to streaming data sources. In ISWC, 2010, pages 96–111.
- EP-SPARQL
  - What: focus on event specific operators
  - Ref: Anicic, D., Fodor, P., Rudolph, S., & Stojanovic, N. EP-SPARQL: a unified language for event processing and stream reasoning. In WWW 2011, pages 635–644.
- TEF-SPARQL
  - What: adds "facts" as first class elements
  - Ref: https://www.merlin.uzh.ch/publication/show/8467

Alternatives to C-SPARQL

- Comparison between existing approaches

System	S2R	R2R	Time-aware	R2S
C-SPARQL Engine	Logical and triple-based	SPARQL 1.1 query	timestamp function	Batch only
Streaming Linked Data Framework	Logical and graph-based	SPARQL 1.1	no	Batch only
$\text{SPARQL}_{\text{stream}}$	Logical and triple-based	SPARQL 1.1 query	no	Ins, batch, del
CQELS	Logical and triple-based	SPARQL 1.1 query	no	Ins only
TEF-SPARQL	no	SPARQL-like	Temporarily Facts, BEFORE SINCE, UNTIL, DURING,	Batch only
EP-SPARQL	no	SPARQL 1.0	SEQ, PAR, AND, OR, DURING, STARTS, EQUALS, NOT, MEETS, FINISHES	Ins only

Work in progress at RSP@W3C

- RSP-QL
  - Syntax
    - https://github.com/streamreasoning/RSP-QL/blob/master/RSP-QL%20Sample%20Queries.md
  - Proposed semantics
    - D.Dell'Aglio, E.Della Valle, J.-P.Calbimonte, Ó. Corcho: RSP-QL Semantics: **A Unifying Query Model to Explain Heterogeneity of RDF Stream Processing Systems**. Int. J. Semantic Web Inf. Syst. 10(4): 17-44 (2014)
  - Semantics (work in progress)
    - https://github.com/streamreasoning/RSP-QL/blob/master/Semantics.md
  - Quick ref.
    - D. Dell'Aglio, J.-P. Calbimonte, E. Della Valle, Ó. Corcho: **Towards a Unified Language for RDF Stream Query Processing.** ESWC (Satellite Events) 2015: 353-363

Contribution: continuous deductive reasoning

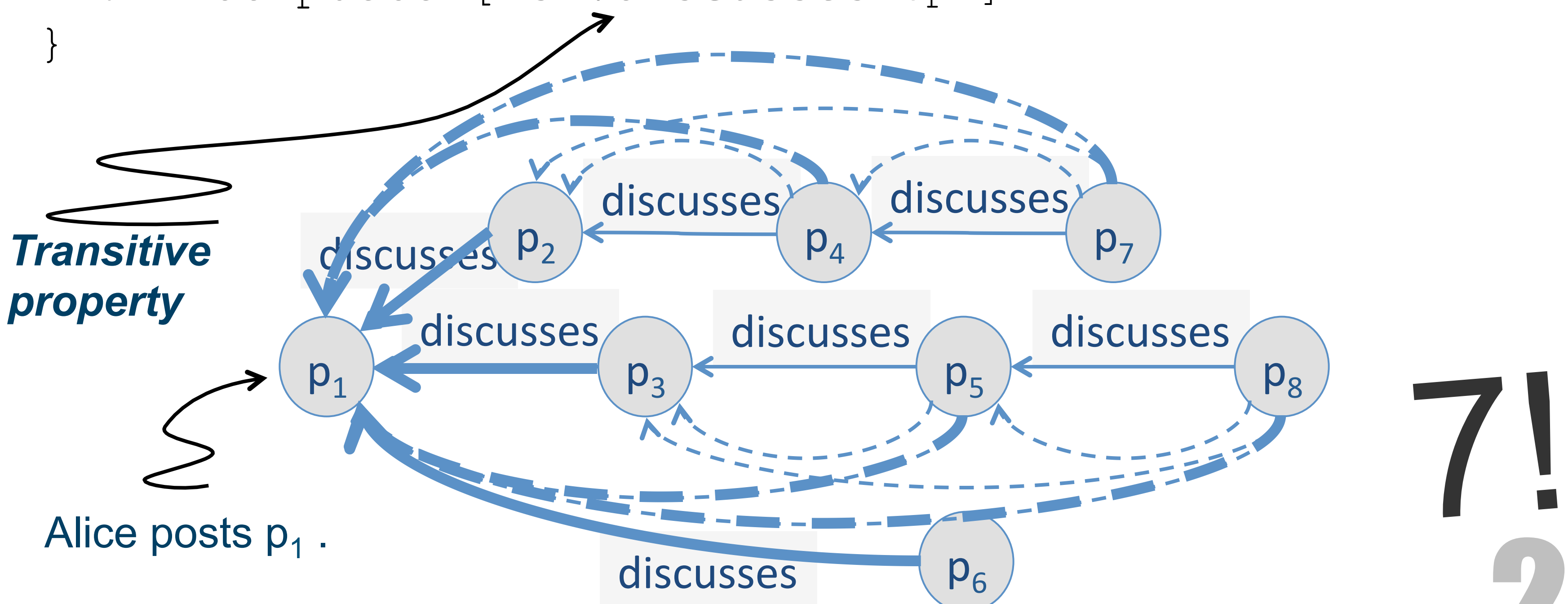
- DL Ontology Stream $\mathbf{S}^{T}$
  - A ontology stream with respect to a static Tbox *T* is a sequence of Abox axioms $\mathbf{S}^{T}(i)$
- A Windowed Ontology Stream $\mathbf{S}^{T}(o,c]$
  - A windowed ontology stream with respect to a static Tbox *T* is the union of the Abox axioms $\mathbf{S}^{T}(i)$ where $o<i\leq c$
- Reasoning on a Windowed Ontology Stream $\mathbf{S}^{T}(o,c]$ is as reasoning on a static DL KB

Emanuele Della Valle, Stefano Ceri, Davide Francesco Barbieri, Daniele Braga, Alessandro Campi: **A First Step Towards Stream Reasoning**. FIS 2008: 72-81

Example of continuous deductive reasoning

What impact has been my micropost p_1 creating in the last hour? Let's count the number of microposts that discuss it …

```
REGISTER STREAM ImpactMeter AS
SELECT (count(?p) AS ?impact)
FROM STREAM <http://…/fb> [RANGE 60m STEP 10m]
WHERE {
  :Alice posts [ sr:discusses ?p ]
}
```



Finding

- **The Semantic Web stack can be extended** so to incorporate streaming data as a first class citizen
  - **RDF stream** data model
  - **Continuous SPARQL** syntax and semantics
  - **Continuous deductive reasoning** semantics

Alternatives to continuous deductive (RDFS++) reasoning

- ETALIS
  - What: RDFS + Allen Algebra
  - Ref: Anicic, D., Rudolph, S., Fodor, P., & Stojanovic, N. Stream reasoning and complex event processing in ETALIS. Semantic Web, 3(4), 2012, 397–407.
- STARQL
  - What:
    - DL-Lite + Conjunctive Query + time-series
    - SHI + Grounded Conjunctive Queries + time-series
  - Ref: ÖL Özçep, R Möller. Ontology Based Data Access on Temporal and Streaming Data. Reasoning Web, 2014
- ASP-based
  - What: time-decaying ASP
  - Ref: http://arxiv.org/abs/1301.1392
- LARS
  - What: high-level unified formal foundation for stream reasoning
  - Ref: H. Beck, M. Dao-Tran, T. Eiter, M. Fink: LARS: A Logic-Based Framework for Analyzing Reasoning over Streams. AAAI 2015: 1431-1438H.

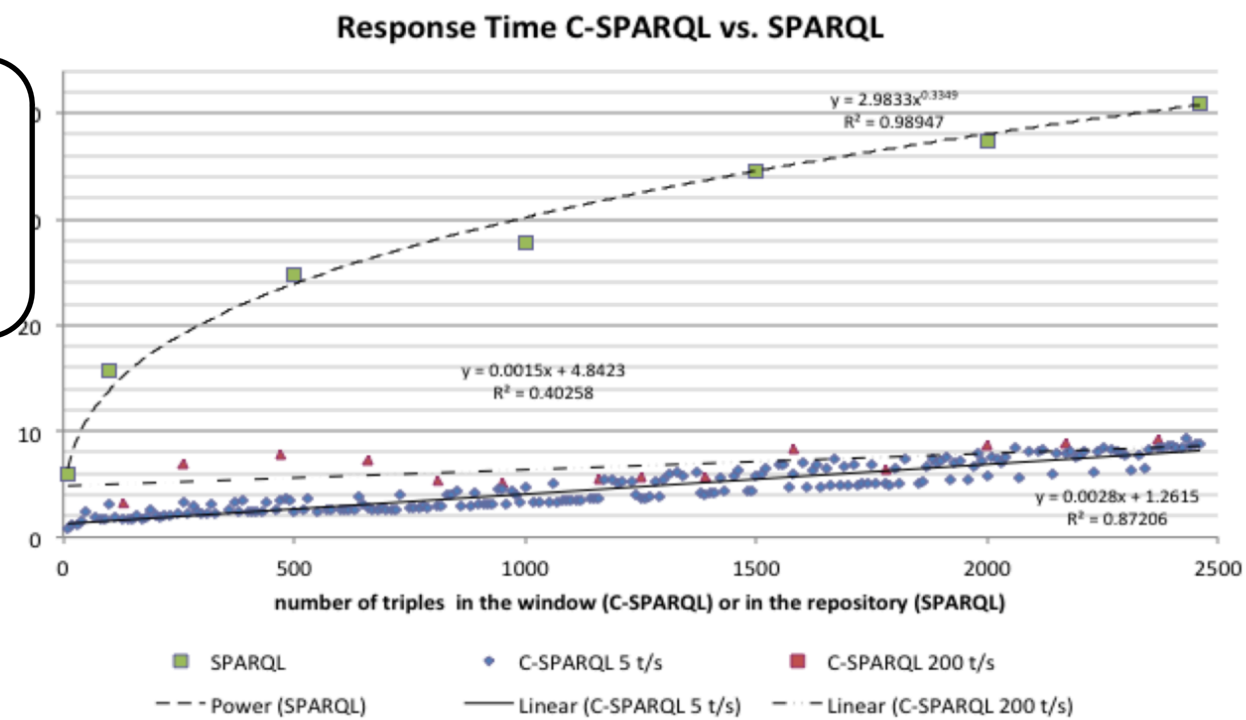
Sub-research questions

1. Is it possible **extend the Semantic Web stack** in order to represent heterogeneous data streams, continuous queries, and continuous reasoning tasks?
2. Does the ordered nature of data streams and the possibility to forget old enough information allow to **optimize continuous querying and continuous reasoning** tasks so **to provide reactive answers** to large number of concurrent users without forsaking correctness or completeness?
3. Can Semantic Web and Machine Learning technologies be jointly employed to **cope with the noisy and incomplete** nature of **data** streams?
4. Are there **practical cases** where processing data stream at semantic level is the best choice?

Contribution: optimize querying for reactive answers

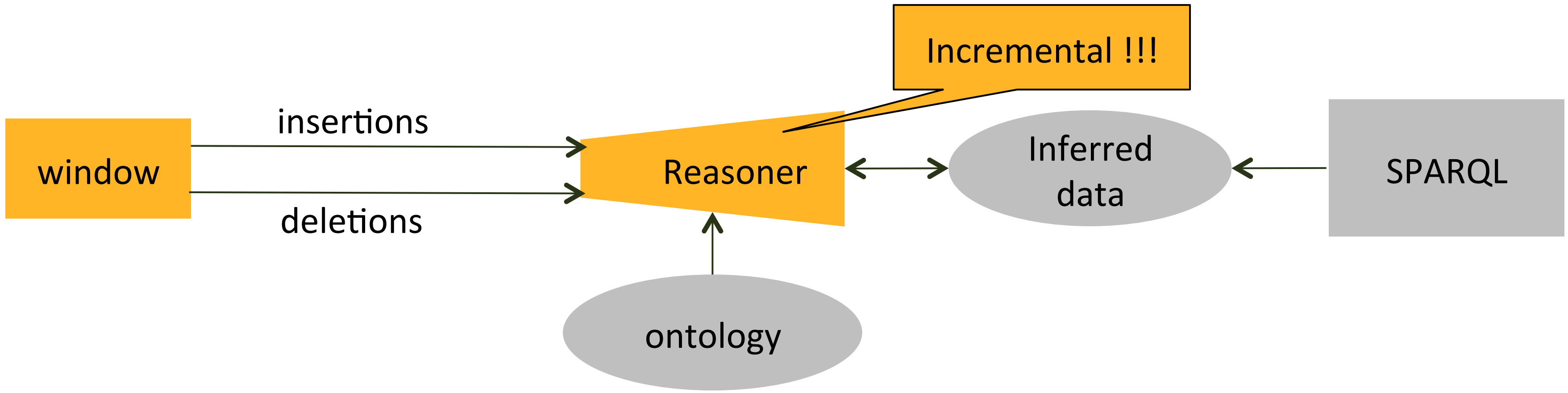
- **C-SPARQL engine** time window-based selection outperforms SPARQL filter-based selection (Jena-ARQ)

Our In-memory RDF stream processing engine



D. Barbieri, D. Braga, S. Ceri, E. Della Valle, Y. Huang, V. Tresp, A.Rettinger, H. Wermser: **Deductive and Inductive Stream Reasoning for Semantic Social Media Analytics** IEEE Intelligent Systems, 30 Aug. 2010.

Not so naïve approach to stream reasoning



- The problem is that materialization (the result of data-driven processing) are very difficult to decrement efficiently.
  - State-of-the-art: DRed algorithm
    - Over delete
    - Re-derive
    - Insert

Y. Ren, J. Z. Pan. Optimising ontology stream reasoning with truth maintenance system. In CIKM (2011)

Is DRed needed?

- **DRed** works with **random insertions** and **deletions**
- **In a streaming setting**, when a triple enters the window, given the size of the window, the reasoner knows already when it will be deleted!
- E.g.,
  - if the window is 40 minutes long, and,
  - it is 10:00, the triple(s) entering now
  - will exit on 10:40.
- Conclusion
  - **deletions are predictable**

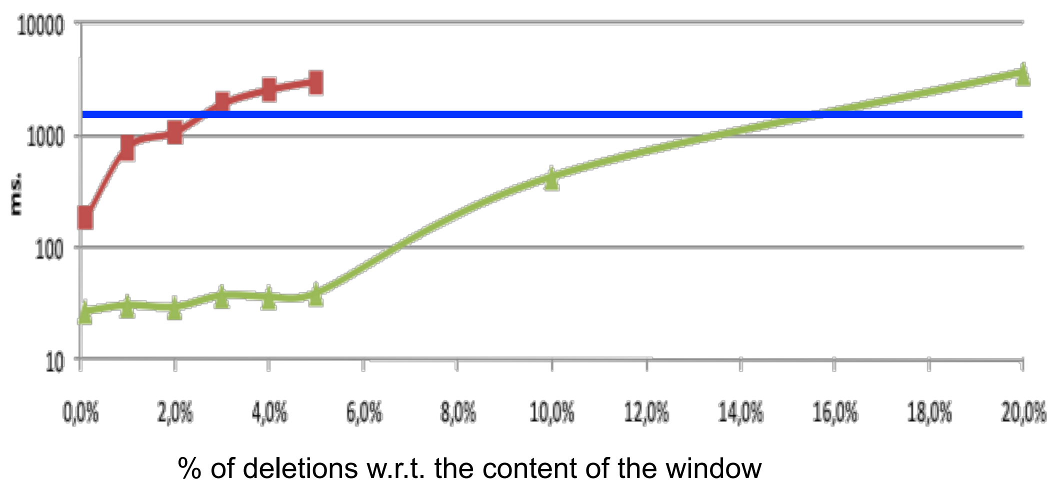
Time	Enter window	Exit window	Explicitly in window
10:00	A←B		A ← B
10:10	B←C		A ← B ← C
10:20	A←E		E A ← B ← C
10:30	E←C		E A ← B ← C
10:40		A←B	E A B ← C
10:50		B←C	E A C
11:00		A←E	E

Contribution: IMaRS algorithm

- Idea:
  - **add an expiration time** to each triple and
  - **use an hash table** to index triples by their expiration time
- The algorithm
  1. **deletes expired triples**
  2. Adds the new derivations that are consequences of insertions **annotating** each inferred triple **with an expiration time** (the **min** of those of the triple it is derived from), and
  3. **when multiple derivations** occur, for each multiple derivation, it keeps the **max** expiration time.

Contribution: IMaRS algorithm

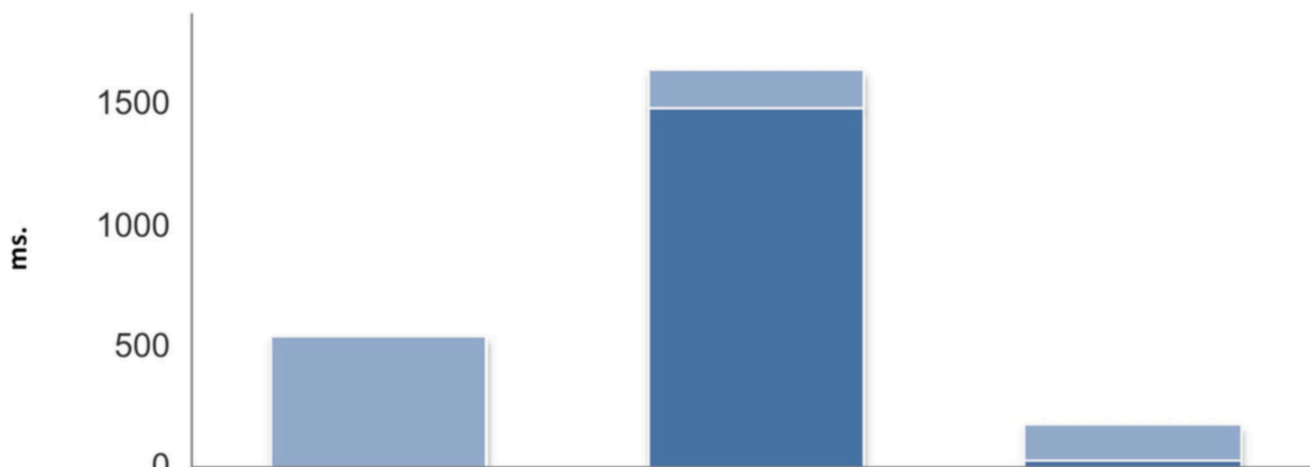
- Incremental Reasoning on RDF streams (**IMaRS**): new reasoning algorithm optimized for reactive query answering
  - Re-materialize after each window slide
  - Use DRed
  - IMaRS



D.F. Barbieri, D. Braga, S.Ceri, E. Della Valle, M. Grossniklaus**: Incremental Reasoning on Streams and Rich Background Knowledge.** ESWC (1) 2010: 1-15
D. Dell'Aglio, E. Della Valle: **Incremental Reasoning on RDF Streams**. In A.Harth, K.Hose, R.Schenkel (Eds.) Linked Data Management, CRC Press 2014, ISBN 9781466582408

Contribution: IMaRS algorithm

- comparison of the average time needed to answer a C-SPARQL query, when 2% of the content exits the window each time it slides, using
  - A backward reasoner on the window content
  - DRed + standard SPARQL on the materialization
  - IMaRS + standard SPARQL on the materialization



	Backward Reasoner	DRed + SPARQL	IMaRS + SPARQL
query	582	161	161
materialization	0	1591	28

Finding

- **Stream Reasoning task is feasible** and the very nature of streaming data offers opportunities to **optimise reasoning tasks** where data is ordered by recency and can be forgotten after a while
  - **C-SPARQL Engine prototype**
  - **IMaRS** continuous incremental reasoning **algorithm**

Optimizing for stream reasoning alternative approaches

- DyKnow
  - How: logical models of an observed dynamic system + metric temporal logics
  - Fredrik Heintz, Jonas Kvarnström, Patrick Doherty: Bridging the sense-reasoning gap: DyKnow - Stream-based middleware for knowledge processing. Advanced Engineering Informatics 24(1): 14-26 (2010)
- MorphStream
  - How: rewriting in DSMS languages (one at a time)
  - Ref: Calbimonte, J.-P., Corcho, O., & Gray, A. J. G. Enabling ontology-based access to streaming data sources. In ISWC, 2010, pages 96–111.
- TR-OWL
  - How: Truth maintenance for EL++ with syntactic approximations
  - Ref: Y. Ren, J. Z. Pan. Optimising ontology stream reasoning with truth maintenance system. In CIKM (2011)
- ETALIS
  - How: rewriting in prolog
  - Ref: Anicic, D., Rudolph, S., Fodor, P., & Stojanovic, N.. Stream reasoning and complex event processing in ETALIS. Semantic Web, 3(4), 2012, 397–407.

(continues in the next slide)

Optimizing for stream reasoning alternative approaches

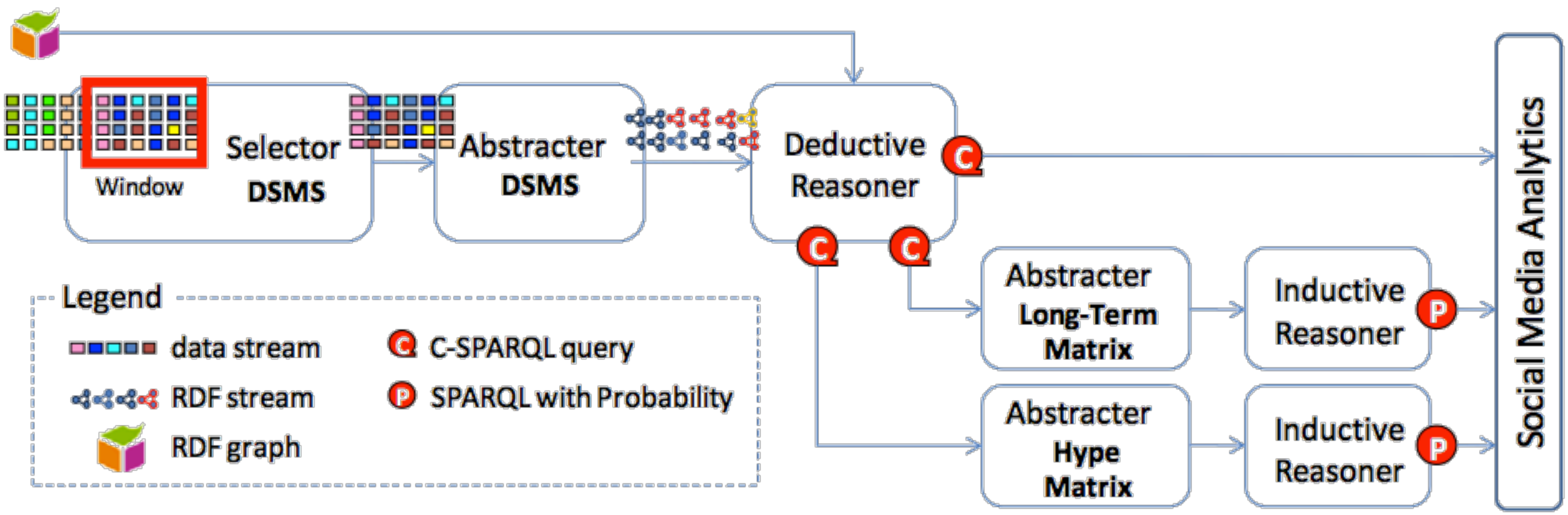
- Sparkwave
  - How: extended RETE algorithm for windows and RDFS
  - Ref: Sparkwave: Continuous Schema-Enhanced Pattern Matching over RDF Data Streams. Komazec S, Cerri D. DEBS 2012
- DynamiTE
  - How: Truth maintenance for ρDF (a fragment of RDFS)
  - J. Urbani, A. Margara, C. J. H. Jacobs, F. van Harmelen, H.E. Bal: DynamiTE: Parallel Materialization of Dynamic RDF Data. ISWC (1) 2013: 657-672
- STARQL
  - How: rewriting on a scalable DSMS with time-series support
  - Ref: ÖL Özçep, R Möller. Ontology Based Data Access on Temporal and Streaming Data. Reasoning Web, 2014
- ASP-based
  - How: optimizing ASP for incremental and time-decaying programs
  - Ref: http://arxiv.org/abs/1301.1392
- The Backward/Forward Algorithm
  - How: optimizing DRed
  - B. Motik, Y. Nenov, R.E.F. Piro, I. Horrocks: Incremental Update of Datalog Materialisation: the Backward/Forward Algorithm. AAAI 2015: 1560-1568

Sub-research questions

1. Is it possible **extend the Semantic Web stack** in order to represent heterogeneous data streams, continuous queries, and continuous reasoning tasks?
2. Does the ordered nature of data streams and the possibility to forget old enough information allow to **optimize continuous querying and continuous reasoning** tasks so **to provide reactive answers** to large number of concurrent users without forsaking correctness or completeness?
3. Can Semantic Web and Machine Learning technologies be jointly employed to **cope with the noisy and incomplete** nature of **data** streams?
4. Are there **practical cases** where processing data stream at semantic level is the best choice?

Cope with the noisy and incomplete data

- **"Noise"** is reduced using **DSMS** techniques
- **Deductive stream reasoning** copes with **incompleteness** deducing implicit facts
- **Inductive stream reasoning** copes with "irrepairable" incompleteness inducing **missing facts**



D.F. Barbieri, D. Braga, S. Ceri, E. Della Valle, Y. Huang, V. Tresp, A. Rettinger, H. Wermser: **Deductive and Inductive Stream Reasoning for Semantic Social Media Analytics.** IEEE Intelligent Systems 25(6): 32-41 (2010)

Findings

- **A combination of deductive and inductive stream reasoning techniques** can cope with incomplete and noisy data

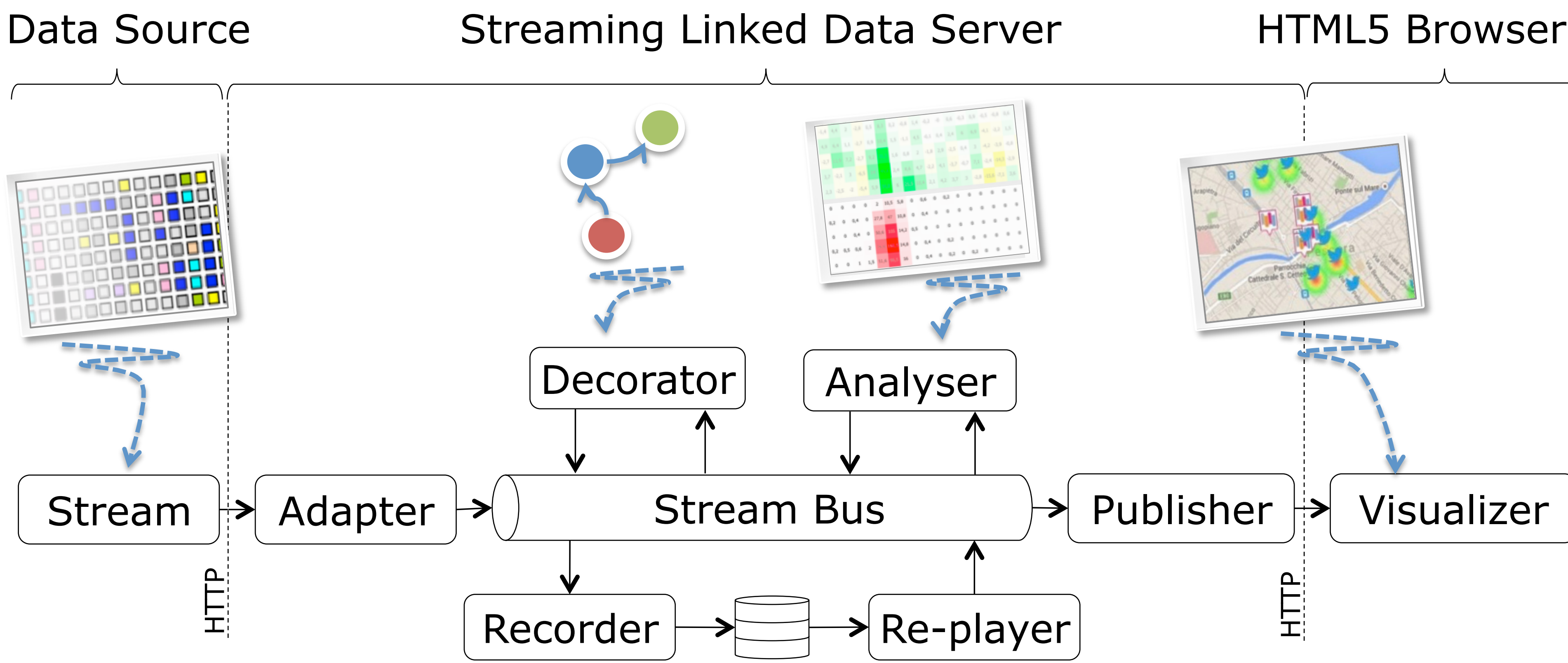
Alternative approaches

- Stream Reasoning with Probabilistic Answer Set Programming
  - Matthias Nickles, Alessandra Mileo: Web Stream Reasoning Using Probabilistic Answer Set Programming. RR 2014: 197-205
  - Anastasios Skarlatidis, Georgios Paliouras, Alexander Artikis, George A. Vouros: Probabilistic Event Calculus for Event Recognition. ACM Trans. Comput. Log. 16(2): 11:1-11:37 (2015)
  - Anni-Yasmin Turhan, Erik Zenker: Towards Temporal Fuzzy Query Answering on Stream-based Data. HiDeSt@KI 2015: 56-69

Sub-research questions

1. Is it possible **extend the Semantic Web stack** in order to represent heterogeneous data streams, continuous queries, and continuous reasoning tasks?
2. Does the ordered nature of data streams and the possibility to forget old enough information allow to **optimize continuous querying and continuous reasoning** tasks so **to provide reactive answers** to large number of concurrent users without forsaking correctness or completeness?
3. Can Semantic Web and Machine Learning technologies be jointly employed to **cope with the noisy and incomplete** nature of **data** streams?
4. Are there **practical cases** where processing data stream at semantic level is the best choice?

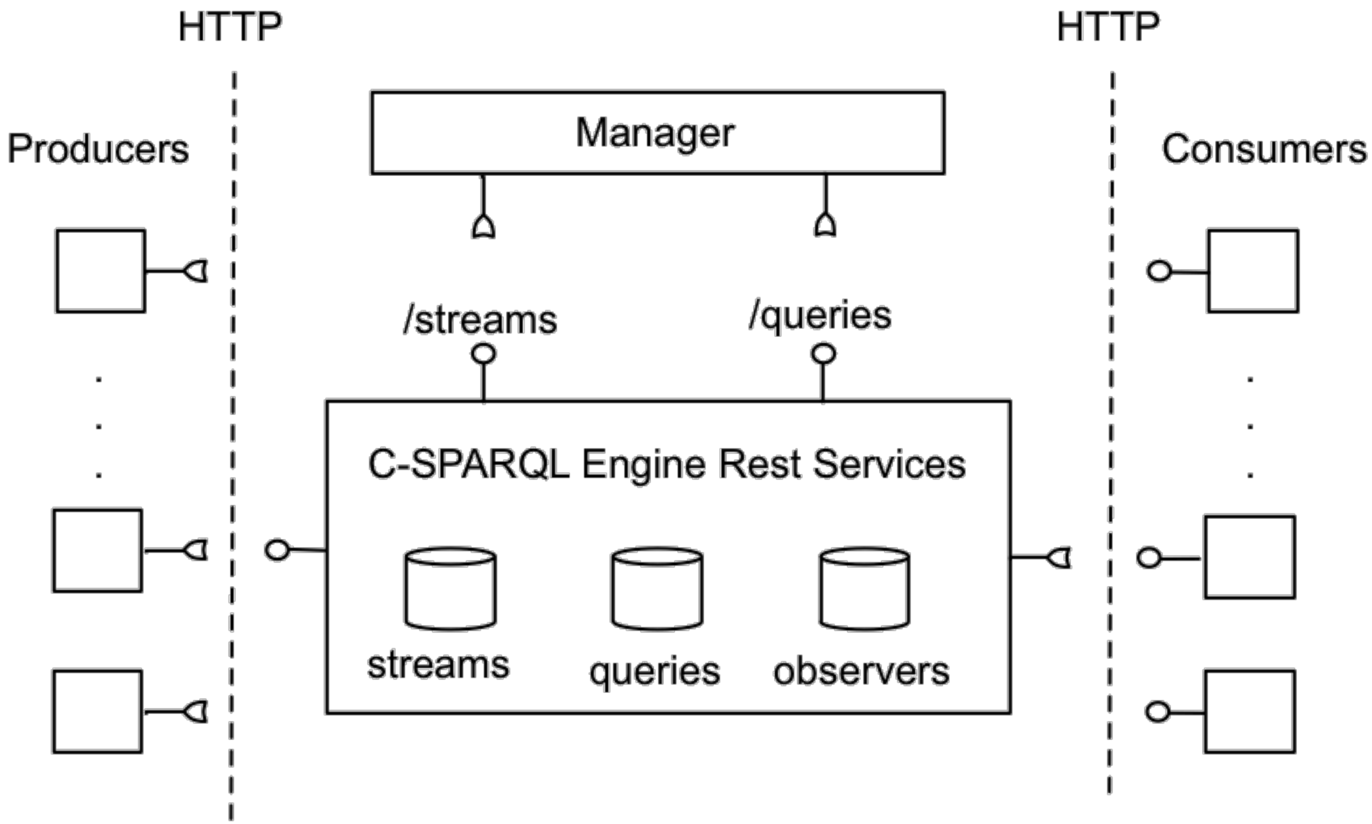
Contribution: Streaming Linked Data Framework



Marco Balduini, Emanuele Della Valle, Daniele Dell'Aglio, Mikalai Tsytsarau, Themis Palpanas, Cristian Confalonieri: **Social Listening of City Scale Events Using the Streaming Linked Data Framework.** International Semantic Web Conference (2) 2013: 1-16

Contribution: RSP services

- RSP services: a RESTful interface for RSP engines



  – http://streamreasoning.org/download/rsp-services

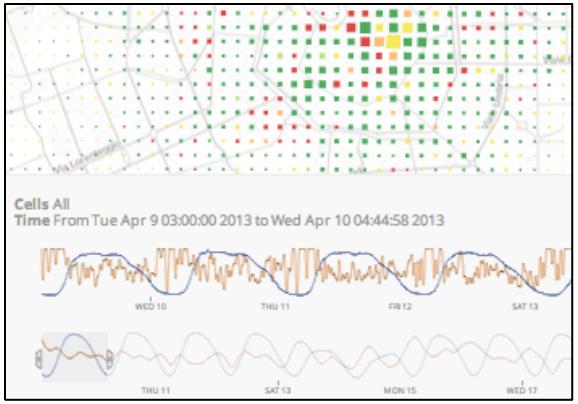
Practical cases

- **10+ deployments** in Sensor Networks & Social media analytics, e.g.

BOTTARI

Winner of Semantic Web Challenge 2011

City Data Fusion



Winner of IBM faculty award 2013

Social Listener

Fluxedo #SimplifyYourLife

M. Balduini, I. Celino, D. Dell'Aglio, E. Della Valle, Y. Huang, T. Lee, S.-H. Kim, V. Tresp: **BOTTARI: An augmented reality mobile application to deliver personalized and location-based recommendations by continuous analysis of social media streams.** J. Web Sem. 16: 33-41 (2012)

M.Balduini, E.Della Valle, M.Azzi, R.Larcher, F.Antonelli, and P.Ciuccarelli: **CitySensing: Fusing City Data for Visual Storytelling.** IEEE MultiMedia 22(3): 44-53 (2015)

Findings

1. **The Semantic Web stack can be extended** so to incorporate streaming data as a first class citizen
   - **RDF stream** data model
   - **Continuous SPARQL** syntax and semantics
   - **Continuous deductive reasoning** semantics
2. **Stream Reasoning task is feasible** and the very nature of streaming data offers opportunities to **optimise reasoning tasks** where data is ordered by recency and can be forgotten after a while
   - **IMaRS** continuous incremental reasoning **algorithm**
   - **C-SPARQL Engine prototype**
3. A combination of **deductive and inductive stream reasoning** techniques **can cope with incomplete and noisy data**
4. There are **application domains** where Stream Reasoning offers an adequate solution

Open issues

1. **The Semantic Web stack can be extended**
   - "Navigating the Chasm between the Scylla of Practical Applications and the Charybdis of Theoretical Approaches"

     A. Bernstein, 2015
2. **Stream Reasoning task is feasible**
   - It's time to start removing assumptions
     - knowledge does not change
     - background data does not change
   - OBDA for SQL ≠ OBDA for continuous querying
3. **Stream reasoning can cope with incomplete and noisy data**
   - Theory is needed!
4. There are **application domains** where Stream Reasoning offers an adequate solution
   - Rigorous quantitative comparative research is needed

Advertisements :-P

- Check out my PhD thesis
  - http://dare.ubvu.vu.nl/handle/1871/53293
  - Chapter 1: Introduction
    - The content of this presentation
  - Chapter 8: conclusions
    - A review of stream reasoning approaches updated in spring 2015
- Put an "I like" to Stream Reasoning on Facebook
  - https://www.facebook.com/streamreasoning

Vienna, Austria - 10.11.2015

Stream Reasoning:
mastering the velocity and the variety dimensions of Big Data at once

Emanuele Della Valle
DEIB - Politecnico di Milano
@manudellavalle
emanuele.dellavalle@polimi.it
http://emanueledellavalle.org