

Ontology-Based Event Processing

Stream Reasoning Workshop - 17.04.2019

Linköping, Sweden, Europe, Earth.....42

R. Tommasini, Pieter Bonte, Emanuele Della Valle,
Femke Ongenaes, Filip De Turck



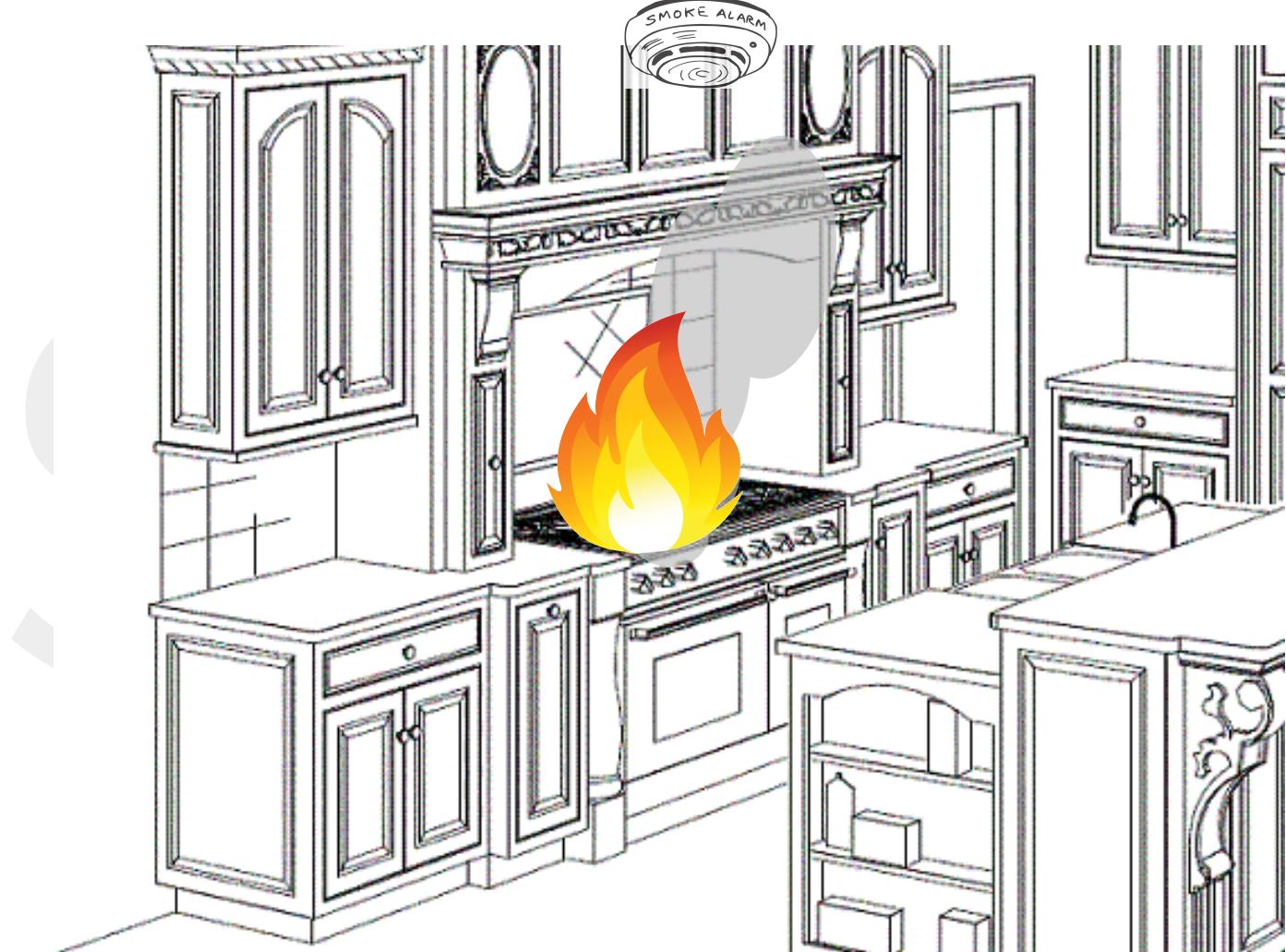
Can you detect fire?

*Expected Answer: **YES**

Can you (actually) detect fire?

Expected Reaction: **Perplexed Audience**















Summary

Workarounds

Smoke Detection

Humidity Variations (decreases)

Temperature Variations (increases)

Ontology-Based Event Processing

An **RDF Stream** is an **partially** ordered sequence of pairs (G_i, t_i) where

- **G_i** , is a [named] **RDF graph** and
- **t_i** is a timestamp.

- an RDF Stream is **Well-Founded** when it does **not** contain **terminological** triples, e.g.,

```
<:c rdf:type rdfs:Class>,  
<:p rdf:type rdfs:Property> etc..
```

An Example

```
( { :s1 :observes :o1 ; :o1 :value 20C }, 1)
( { :s1 :observes :o2 ; :o2 :value 20C }, 2)
( { :s1 :observes :o3 ; :o3 :value 30C }, 3)
( { :s1 :observes :o4 ; :o4 :value 50C }, 4)
```

An **Ontology Stream** is an **partially** ordered sequence of pairs (A_i, t_i) where

- **A_i** , is a set of a **ABox** axioms w.r.t. a static **TBox** T .
- **t_i** is a timestamp.

An Ontology Stream S is an **Event Streams** when

- the **static TBox** E contains some axioms of the form $E \sqsubseteq B$ where B is a **basic concept**,
- E_i denotes an **event** type
- for some $(A_i, t_i) \exists E_i \in E$,

A Language Proposal

Logical Events

Logical Event

EVENT OfficeTemperatureEvent **AS**
TemperatureEvent **and** (hasLocation **some**
Office)

Logical Events

EVENT OfficeTemperatureEvent **AS**
TemperatureEvent **and** (hasLocation **some**
Office)

DL Specifications

Composite Events

Composite Event

EVENT FireEvent

MATCH TemperatureEvent

SEQ SmokeEvent **WITHIN** (5m)

Composite Events

EVENT FireEvent

MATCH TemperatureEvent

SEQ SmokeEvent **WITHIN** (5m)

Logical Event

Composite Events

EVENT FireEvent

MATCH TemperatureEvent

SEQ SmokeEvent **WITHIN** (5m)

Logical Event

Composite Events

EVENT FireEvent

CEP Specification

MATCH TemperatureEvent

SEQ SmokeEvent **WITHIN** (5m)

Composite Events (Filters)

```
EVENT FireEvent MATCH TemperatureEvent SEQ SmokeEvent  
WITHIN (5m)  
IF { EVENT TemperatureEvent {?loc0 hasValue ?v} }
```

Composite Events (Filters)

```
EVENT FireEvent MATCH TemperatureEvent SEQ SmokeEvent
WITHIN (5m) JOINS
IF {
  EVENT TemperatureEvent {?loc0 hasValue ?v}
  EVENT SmokeEvent {?loc1 hasValue ?v}
  FILTER (?smokeLevel == 3) }
```

Composite Events (Filters)

```
EVENT FireEvent MATCH TemperatureEvent SEQ SmokeEvent
WITHIN (5m)
IF {
  EVENT TemperatureEvent {?loc0 hasValue ?v}
  EVENT SmokeEvent {?loc1 hasValue ?v }
  FILTER (?smokeLevel == 3) }
```

Filters

Processing Model

Processing Model

Ingestion

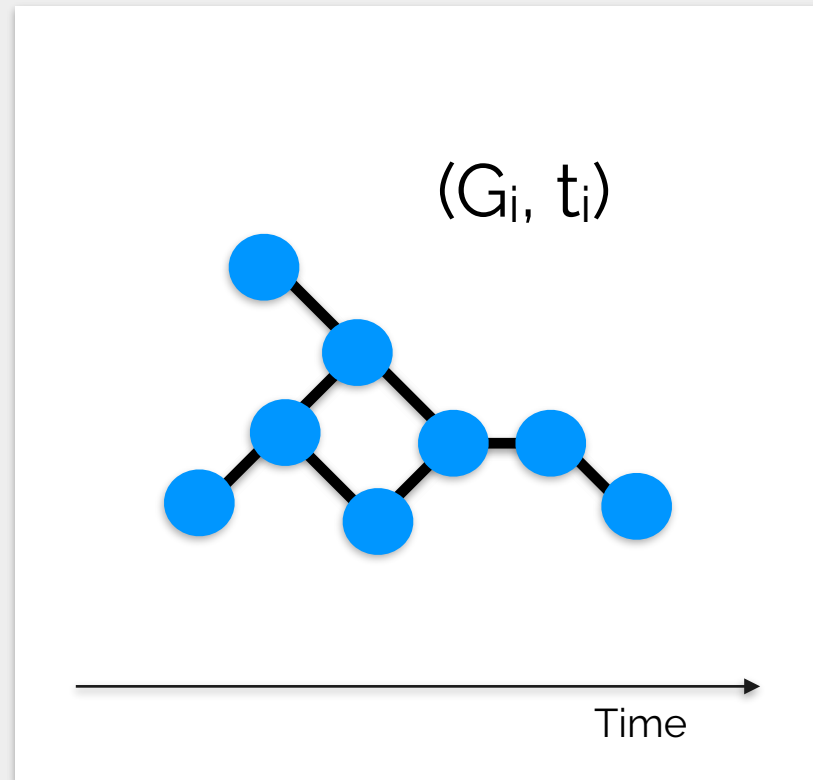


[Icon by GraphicsBay](#)

normalization by shashank singh from the
Noun Project

Data Integrations

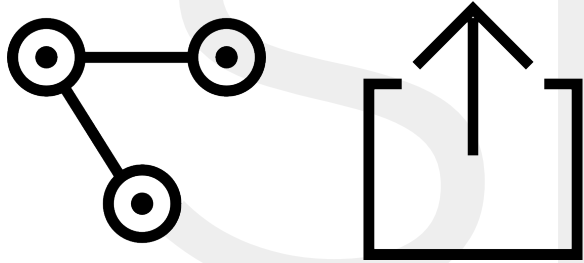
We assume Well-Grounded
RDF Stream as common
data model



Processing Model

Ingestion

Abstraction



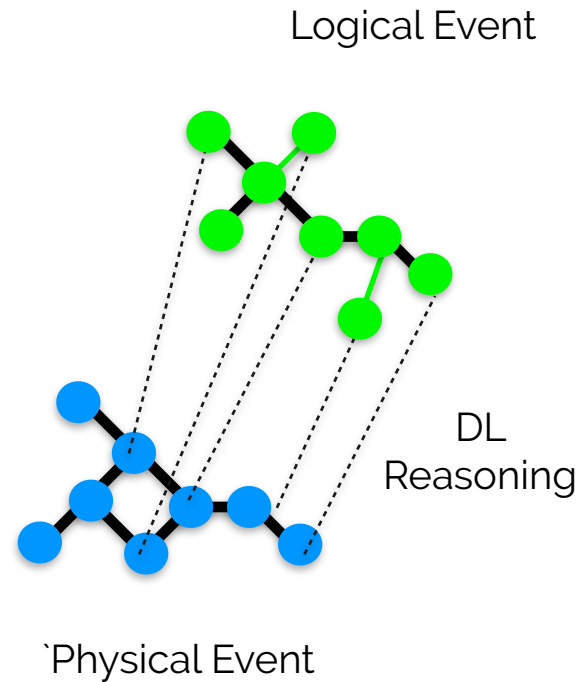
Explanation

Icon by GraphicsBay

normalization by shashank singh from the
Noun Project

Events!

first-class objects in
the language

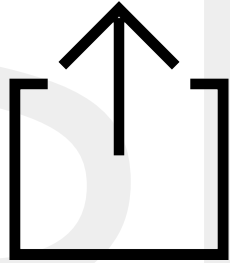
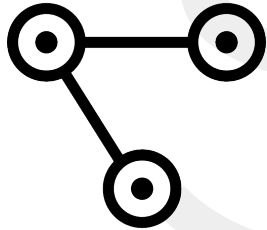


Processing Model

Ingestion

Normalisation

Abstraction



Explanation

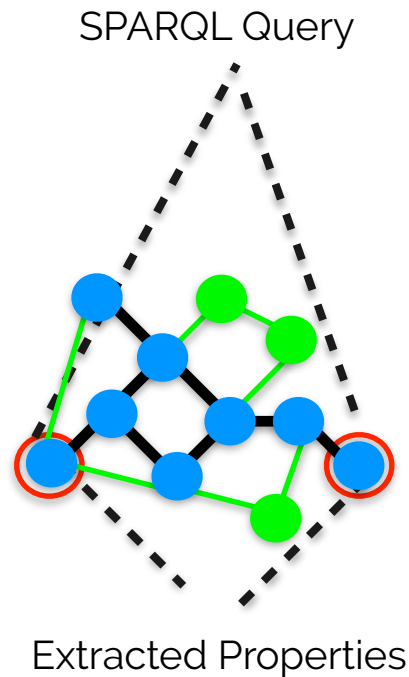
Filtering

[Icon by GraphicsBay](#)

normalization by shashank singh from the
Noun Project

Property Normalisation

To enable **filtering** and
joining



Processing Model

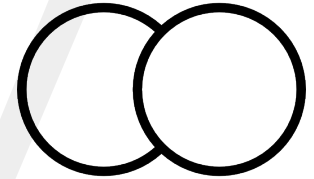
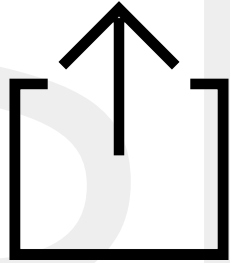
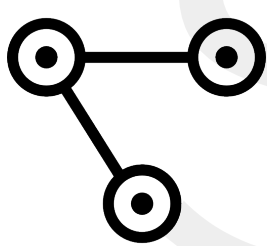
Ingestion

Normalisation

Composition

Abstraction

Correlation



Explanation

Temporal

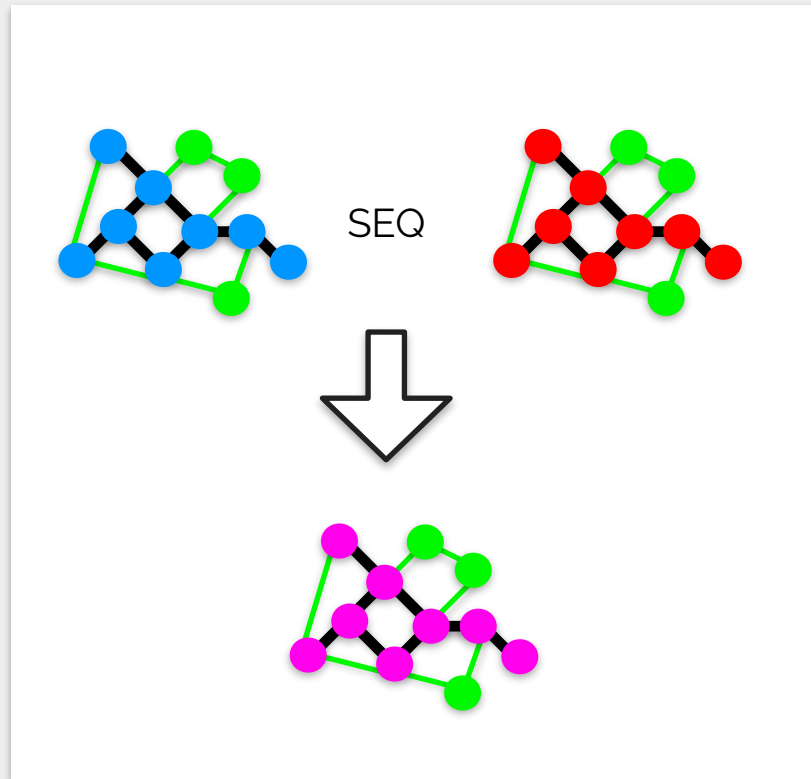
Filtering

Icon by GraphicsBay

normalization by shashank singh from the
Noun Project

Event Detection & Composition

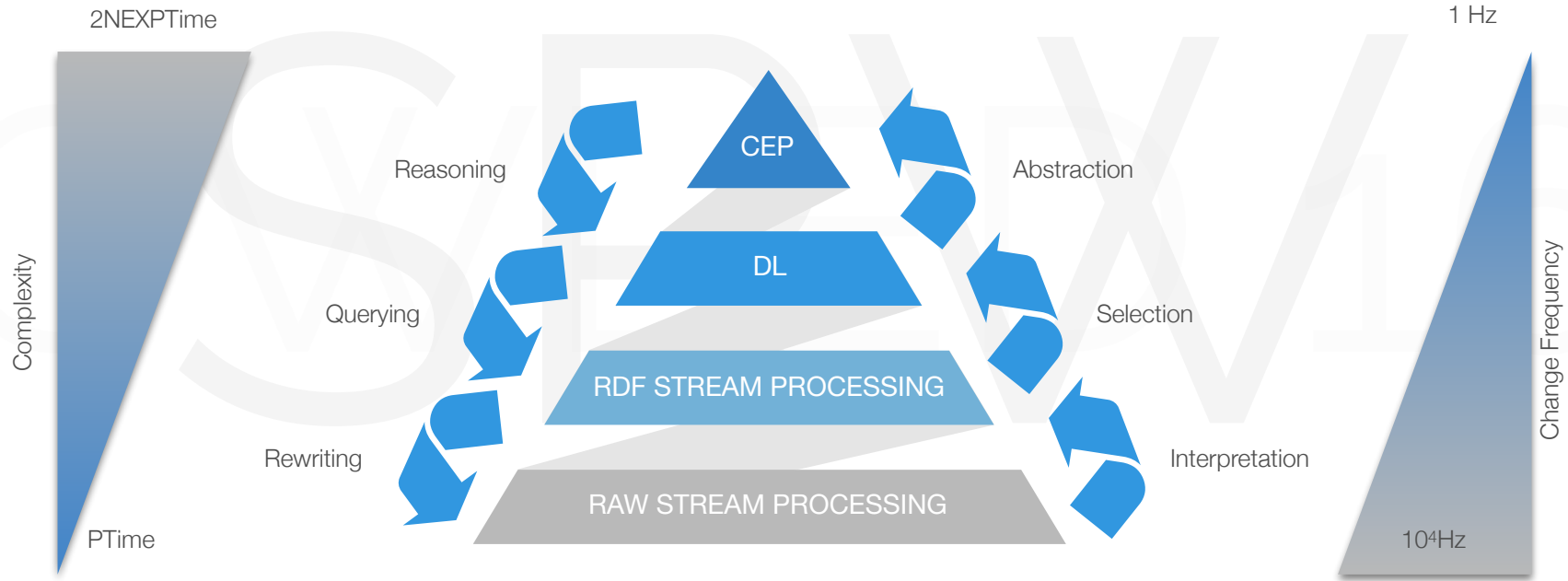
To enable **temporal
correlation** and
reactions



Stream Reasoning

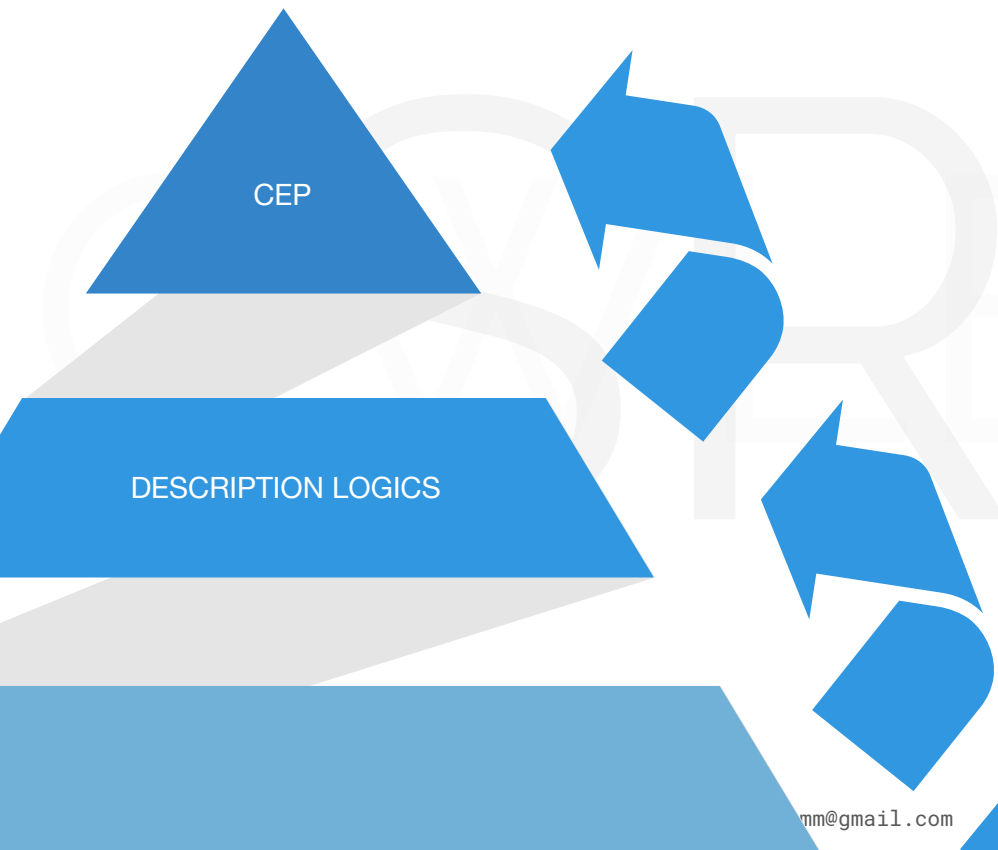
Can we **make sense** in **real-time**
of **heterogeneous**, **vast**,
incomplete, and **noisy** and **data**
streams coming from **complex**
domains?

Cascading Reasoning Revised



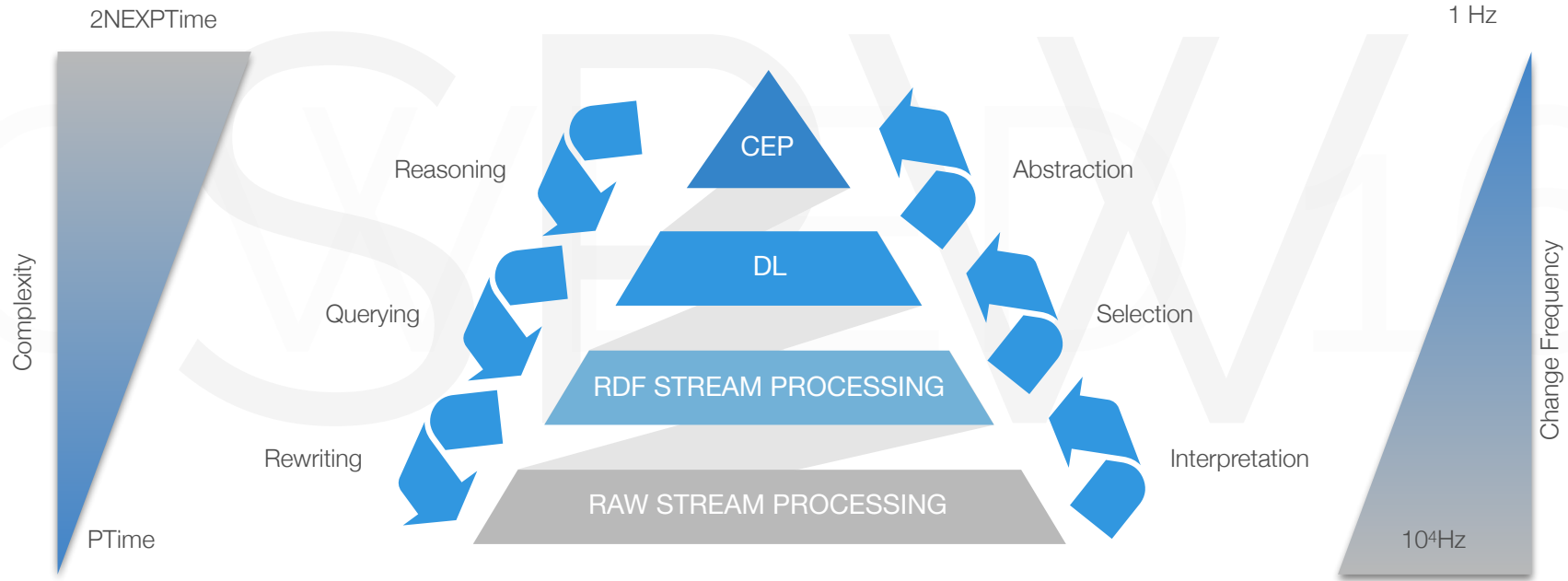
Stuckenschmidt, H., Ceri, S., Della Valle, E., & Van Harmelen, F. (2010). Towards expressive stream reasoning

Ontology-Based Event Processing



- Abstracting **events** from **RDF Streams*** by means of **DL reasoning**
- Computing **temporal relation** across high-level event **abstractions** using **Complex Event Processing**

Cascading Reasoning Revised



Stuckenschmidt, H., Ceri, S., Della Valle, E., & Van Harmelen, F. (2010). Towards expressive stream reasoning

Rewriting and Interpreting

RDF
STREAM PROCESSING

RAW
STREAM PROCESSING

- including **continuous semantics** will **enable** continuous querying over **virtual streaming sources**;
- including **time** operators like windows will **enable query rewriting** into **continuous query languages**

**CURRENT
GROUPS****REPORTS****ABOUT**[Home](#) / [RDF Stream Processing...](#)

RDF STREAM PROCESSING COMMUNITY GROUP

The mission of the RDF Stream Processing Community Group (RSP) is to define a common model for producing, transmitting and continuously querying RDF Streams. This includes extensions to both RDF and SPARQL for representing streaming data, as well as their semantics. Moreover this work envisions an ecosystem of streaming and static RDF data sources whose data can be combined through standard models, languages and protocols. Complementary to related work in the area of databases, this Community Group looks at the dynamic properties of graph-based data, i.e., graphs that are produced over time and which may change their shape and data over time.

Note: Community Groups are proposed and run by the community. Although W3C hosts these conversations, the groups do not necessarily represent the views of the W3C Membership or staff.

[drafts / licensing info](#)

Tools for this group

-  [Mailing List](#)
-  [Wiki](#)
-  [IRC](#)
-  [GitHub repositories](#)
-  [RSS](#)
-  [Contact This Group](#)

Pages

[RDF Stream Processing Workshop at ESWC 2017](#)
[RDF Stream Processing Workshop at ESWC2015](#)
[Sample Page](#)

Questions?

Email: riccardo.tommasini@polimi.it

Twitter: @rictomm

Github: [riccardotommasini](https://github.com/riccardotommasini)

Web1: riccardotommasini.com

Web2: streamreasoning.org

