

# A Restful Interface for RDF Stream Processors

Marco Balduini and Emanuele Della Valle  
Politecnico di Milano

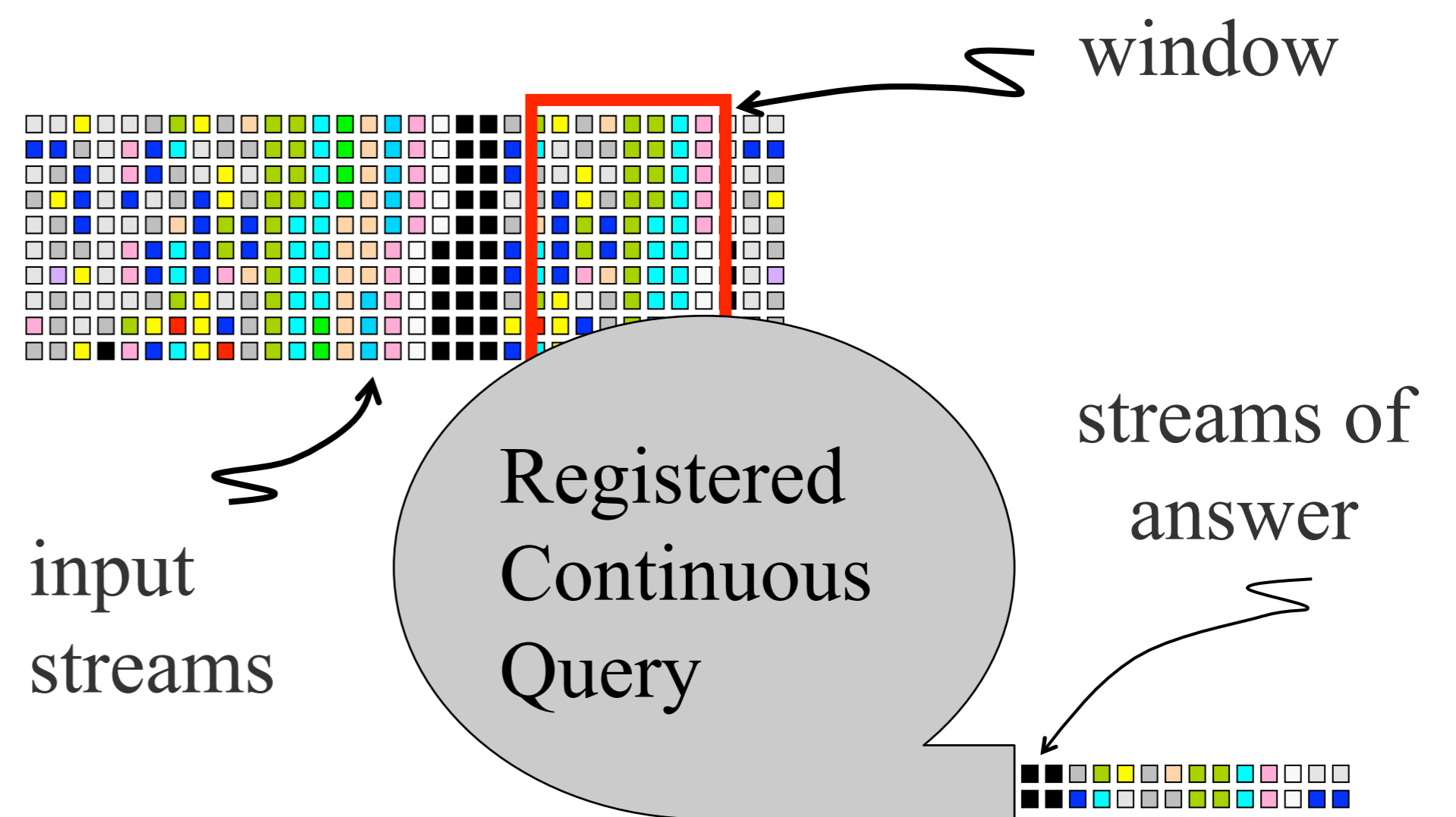
## The Problem

RDF Stream Processors do not talk each other due to a lack of a protocol for:

- **transmitting RDF stream**
- **registering a continuous query** on one or more streams
- **observing** the continuously evolving **results** of a query
- stop/pause/delete a query
- stop observing the results of a query

## Background

What is an RDF Stream Processors?



## Our Solution

### RDF Streams

Method	Address	Body	Description
PUT	/streams/<id>		Register new stream
DELETE	/streams/<id>		Delete specified stream
POST	/streams/<id>	RDF model	Stream new information
GET	/streams		Get the list of streams

### Continuous SPARQL queries

Method	Address	Body	Description
PUT	/queries/<id>	query	Register new query
DELETE	/queries/<id>		Delete specified query
POST	/queries/<id>	callback URL	Adds an observer
POST	/queries/<id>	Action [pause, restart]	Change query status
GET	/queries		Get the list of queries

### Observers

Method	Address	Body	Description
DELETE	/queries/<id>/observers/<id>		Delete specified observer
GET	/queries/<id>/observers		Get observers list

## Advantages

### 1. Minimal

- no timestamps
- no transmission optimization

### 2. backward compatible

- information is stream in RDF
- SPARQL 1.1 result formats adopted

### 3. Combinable

- output of a processor can serve as input to a processor
- self-consumption is also possible

## Learn More

- **Download** a C-SPARQL implementation of the RDF stream Processors RESTful Interfaces  
<http://streamreasoning.org/download/rsp-service4csparql>
- Check out the hands-on sessions of the Stream Reasoning tutorial  
<http://streamreasoning.org/sr4ld2013>
- Have a look to our **demos** to grasp the potential of RDF stream Processing  
<http://streamreasoning.org/demos/>