RDF Stream Processing on Big Data

Oscar Orlando CEBALLOS ARGOTE, M.Sc.
oscar.ceballos@correounivalle.edu.co

John SANABRIA, Ph.D.
Universidad del Valle
Director

Oscar CORCHO, Ph.D.
Universidad Politécnica de Madrid
Codirector

María-Constanza PABÓN, Ph.D.
Pontificia Universidad Javeriana
Asesora

Doctorado en Ingeniería

Énfasis en Ciencias de la Computación

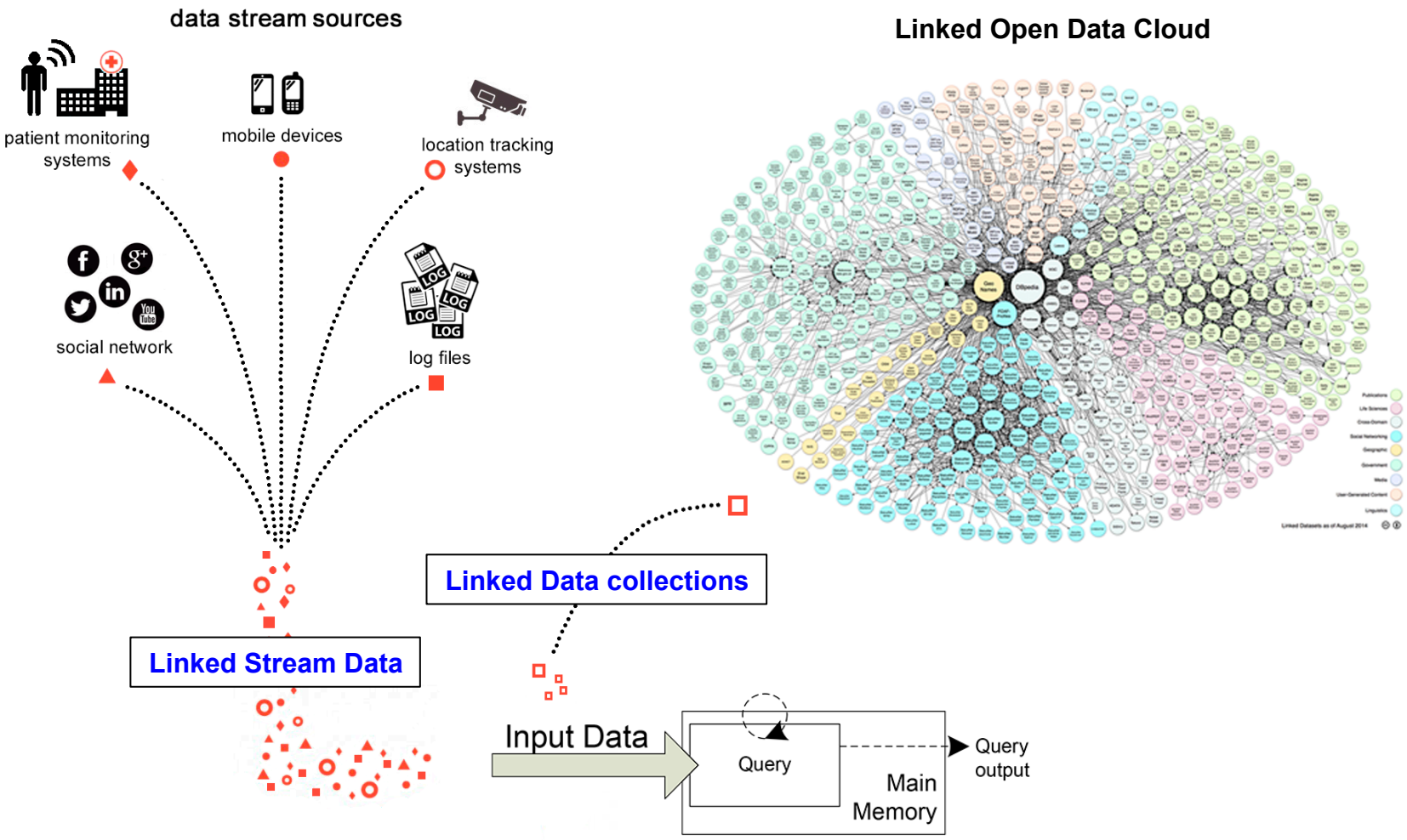
Escuela de Ingeniería de Sistemas y Computación

Universidad del Valle, Cali - Colombia

Contents

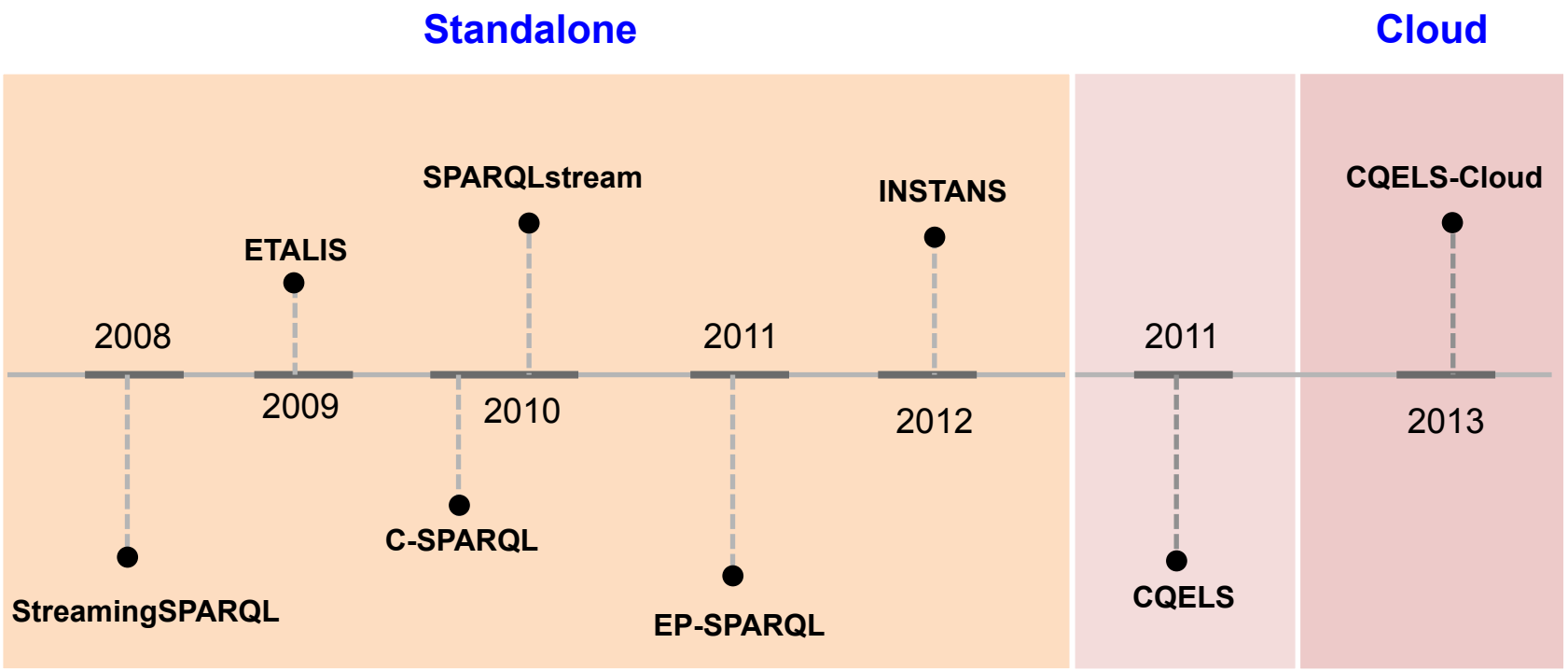
Contents

1. RDF Stream Processing



1. RDF Stream Processing

Some **RDF Stream Processing Systems (RSP Systems)** support unified query processing over **Linked Stream Data** (**RDF Stream**) and **Linked Data collections** (**static RDF datasets**)

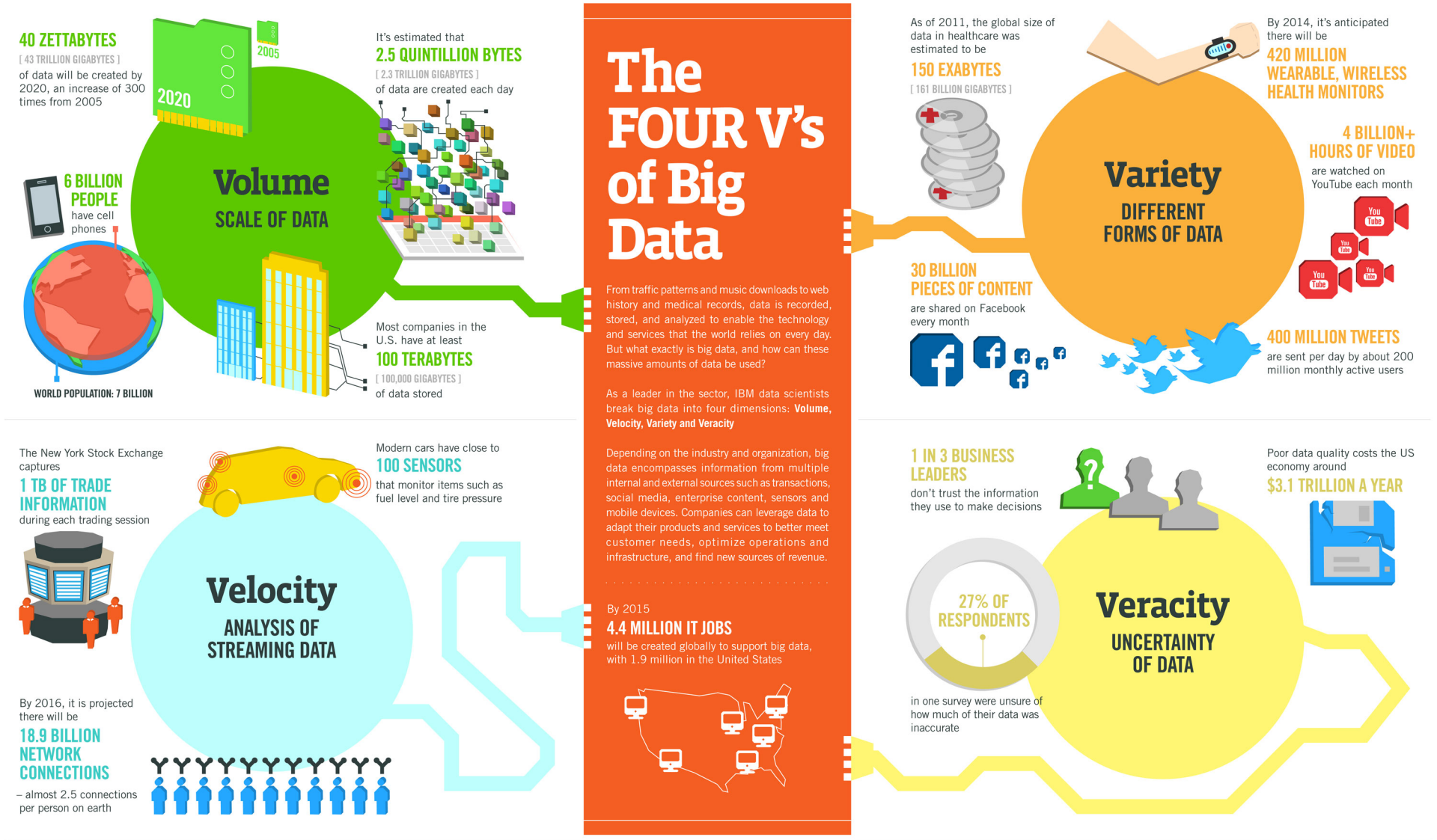


Problem Statement

RSP Systems **performance** and **scalability** still need to be improved in terms of **query execution time**, **data size (dynamic** and **static)**, **number of cuncurrent queries**, and **number of streams**

Contents

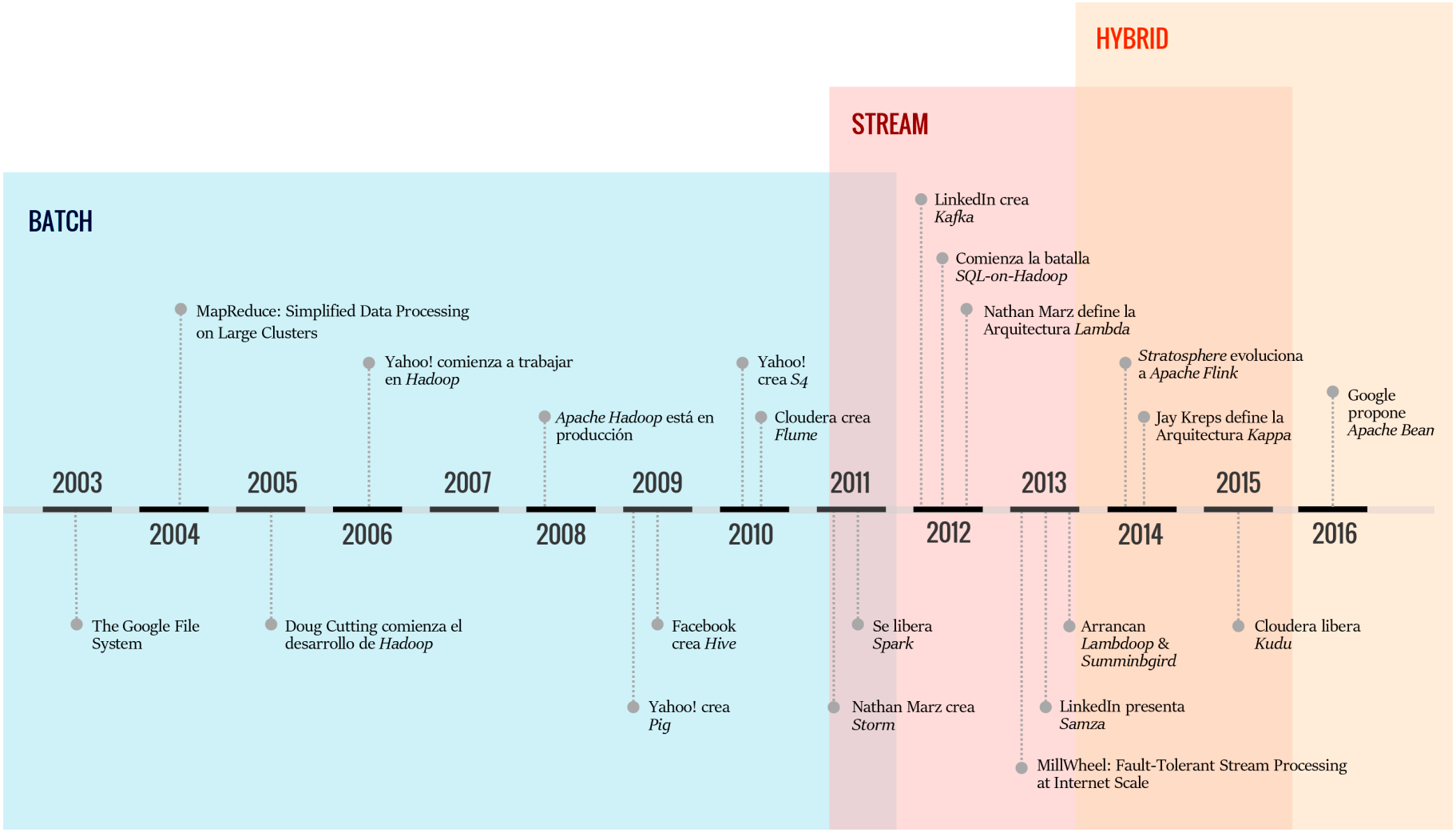
2. Big Data Technologies



Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, QAS

IBM

2. Big Data Technologies



Hypothesis

When extending **Big Data Technologies** to process real time **unified queries** over **RDF streams** and **static RDF datasets**, it is possible to **improve perfomance and scalability** limits of current **RSP Systems**

Research Questions

1. Which of the existing technologies in Big Data are appropiate?

2. Based on such techonology:
   Is it possible to process continuous queries on RDF streams?
   Is the set of operations available (e.g., map, reduce, filter, join) sufficient?
   What other operators are required and necessary to be implemented?
   What query optimization techniques are necessary to be adapted?

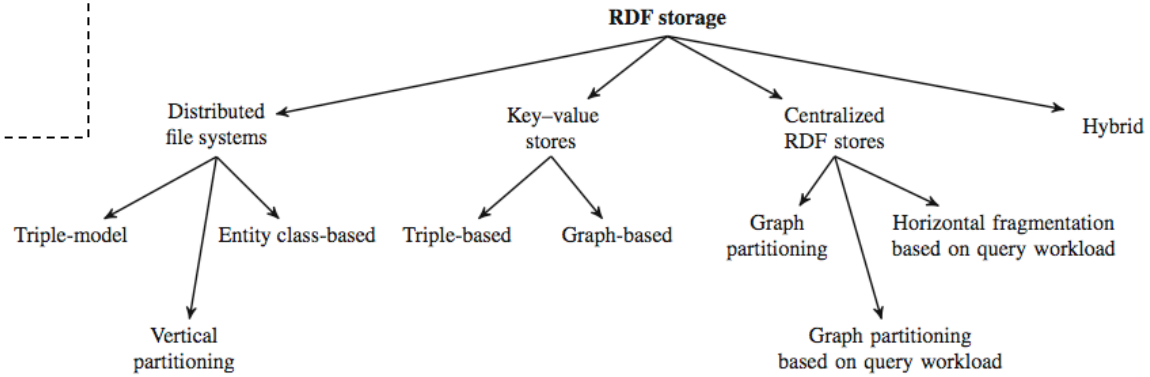
3. What tests can be used to evaluate performance, scalability, completeness and correctness properties on the proposed extensions?

State of Art

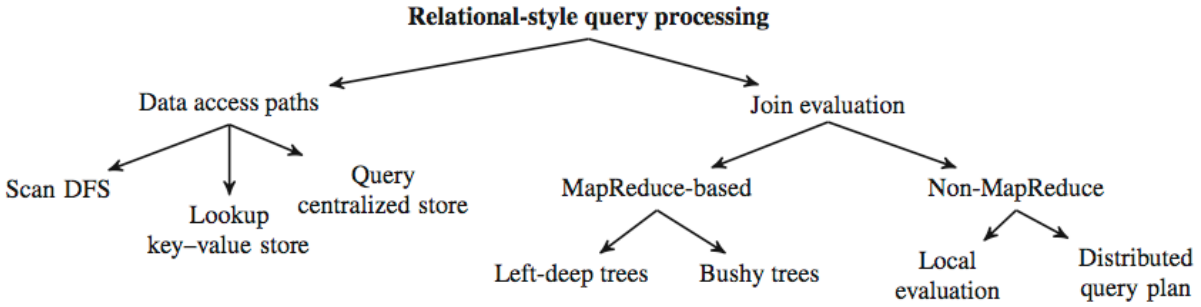
The VLDB Journal (2015) 24:67–91
DOI 10.1007/s00778-014-0364-z

RDF in the clouds: a survey

Zoi Kaoudi · Ioana Manolescu



Taxonomy of storage back-ends and partitioning schemes used by the systems

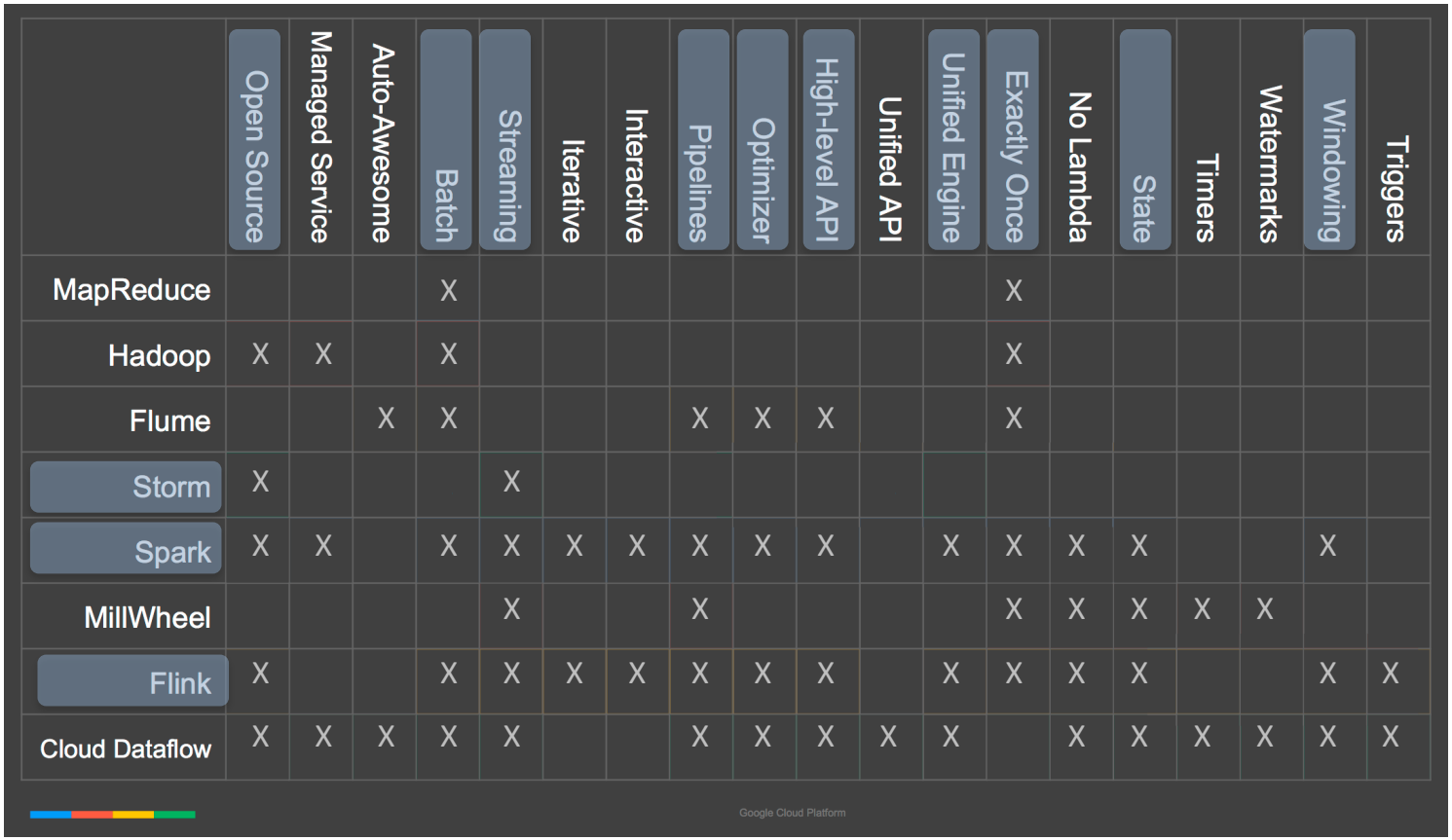


Taxonomy of relational-style query processing strategies

State of Art

The evolution of massive-scale data processing

(https://goo.gl/5k0xal - 2016)

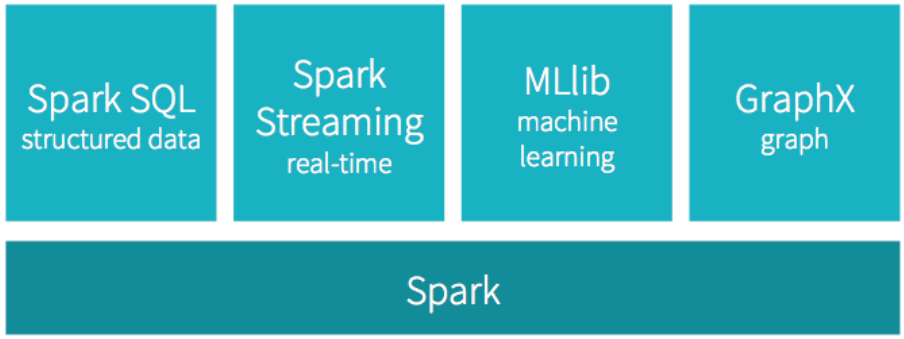


Apache Spark

Resilient Distributed Datasets: A fault-tolerant abstraction for in-memory cluster computing
(Zaharia et al., 2012)

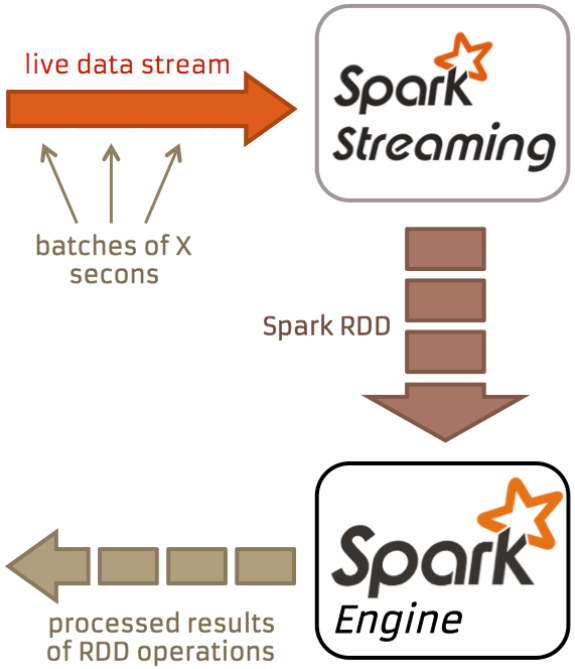
Resilient Distributed Datasets (RDDs)

- Write programs in terms of operations on distributed datasets
- Partitioned collections of objects spread across a cluster, stored in memory or on disk
- RDDs built and manipulated through a diverse set of parallel transformations (map, filter, join) and actions (count, collect, save)
- RDDs automatically rebuilt on machine failure

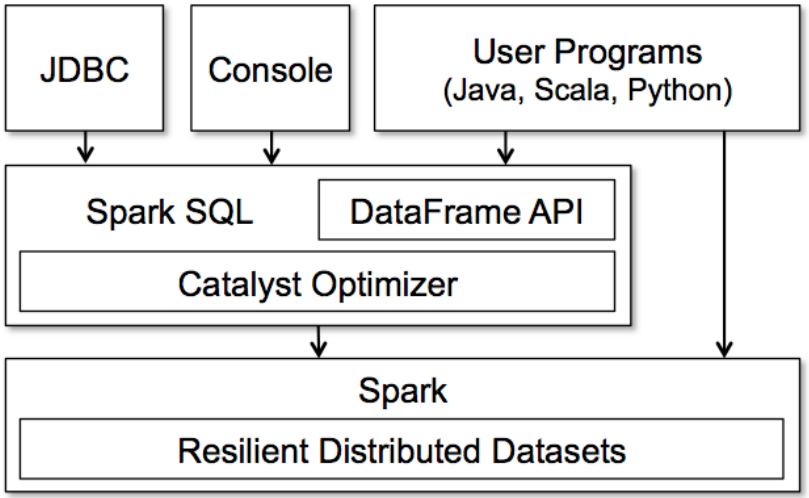


Apache Spark

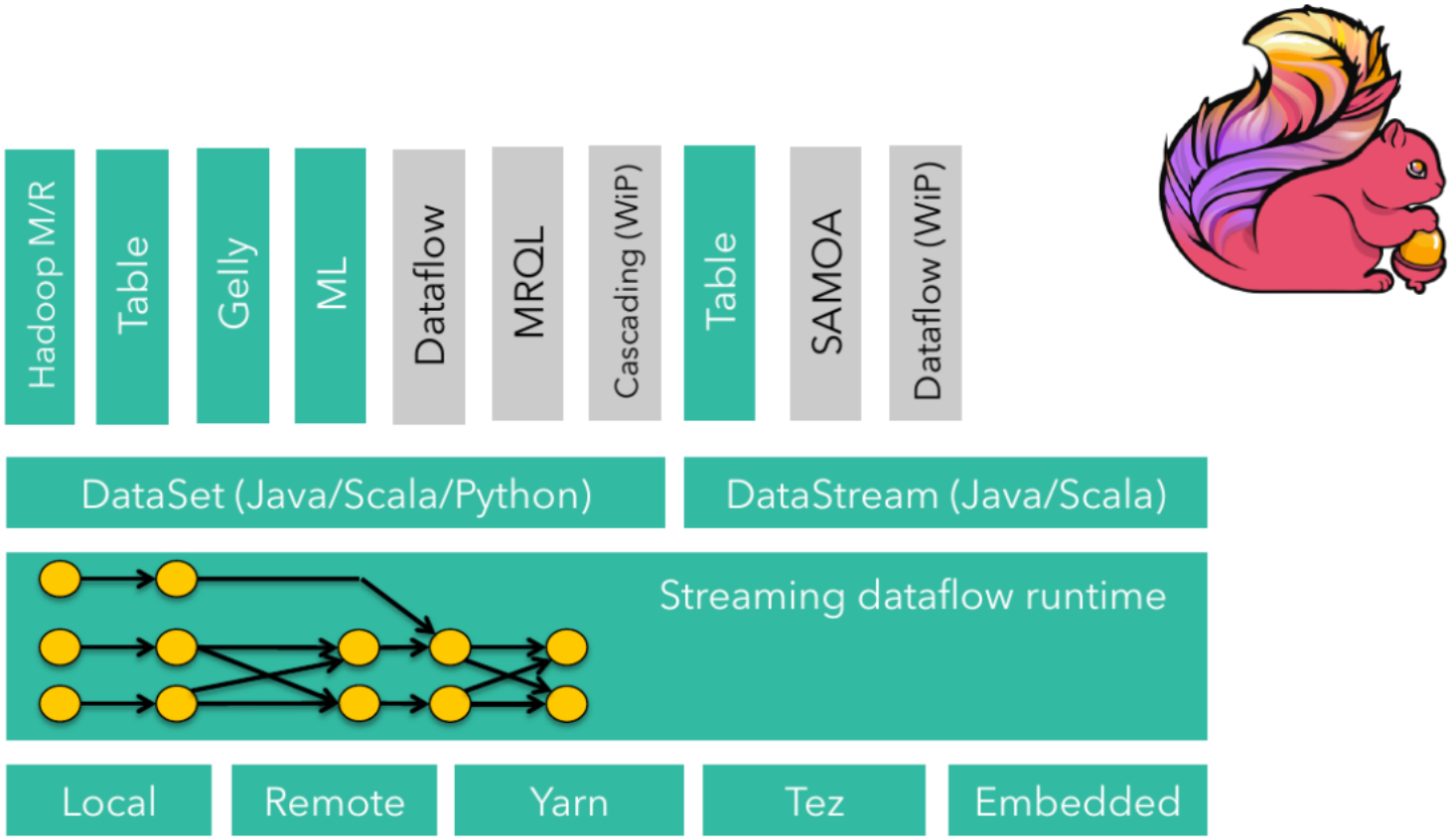
Discretized Streams: Fault-tolerant streaming computation at scale
(Zaharia et al., 2013)



Spark SQL: Relational Data Processing in Spark
(Armbrust et al., 2015)



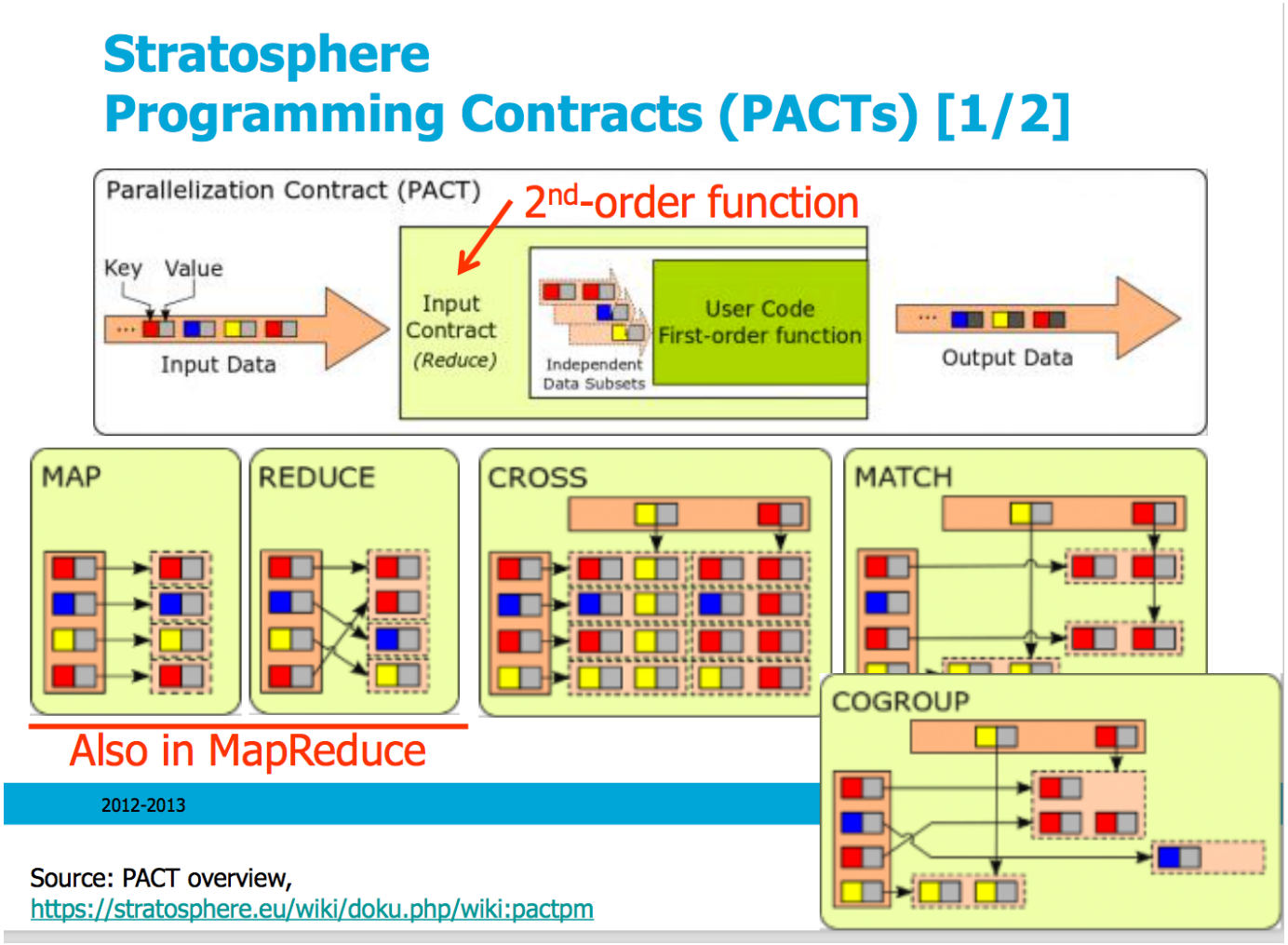
Apache Flink



Apache Flink

The Stratosphere platform for big data analytics *(Alexandrov et al., 2014)*

Apache Flink



Apache Flink

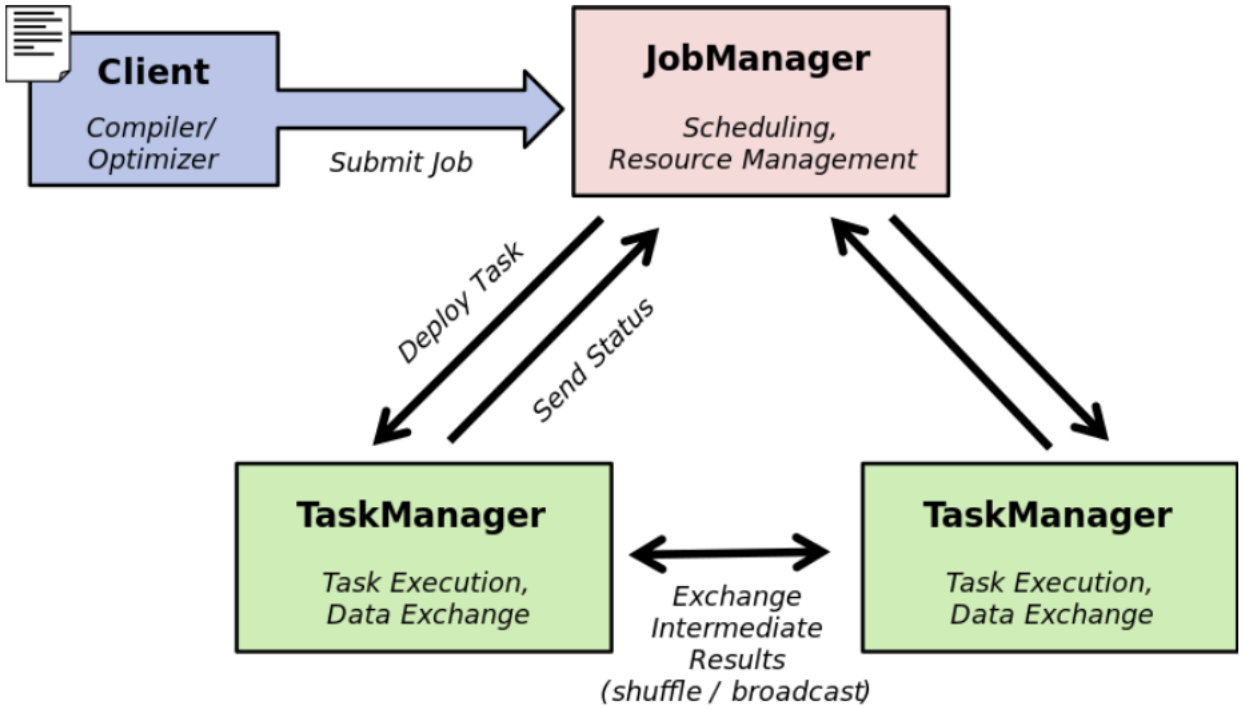
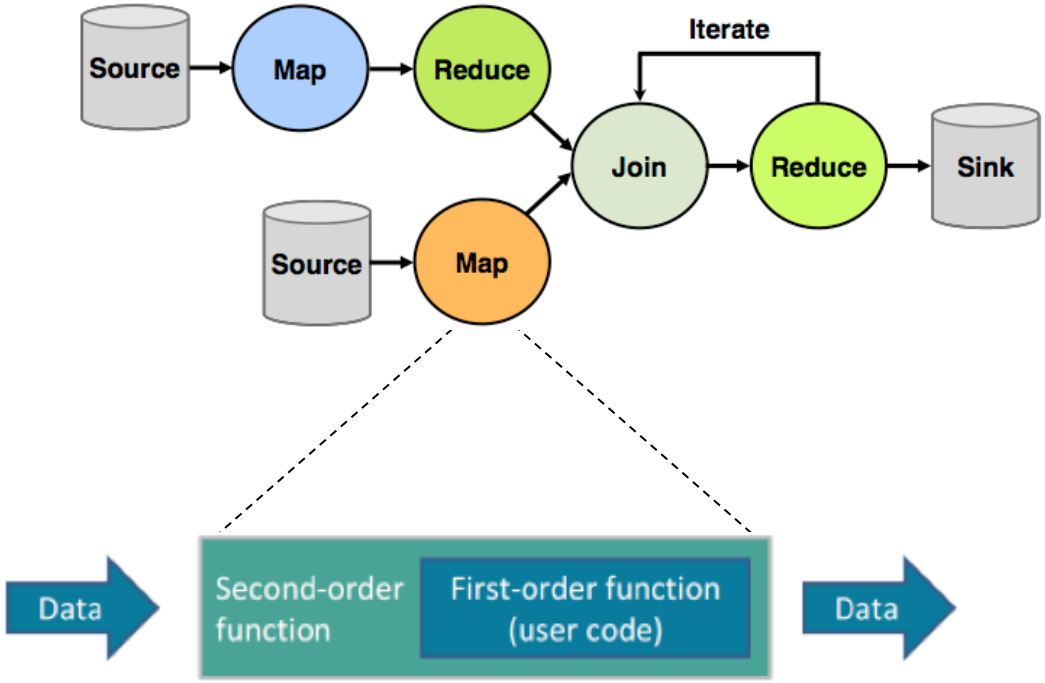


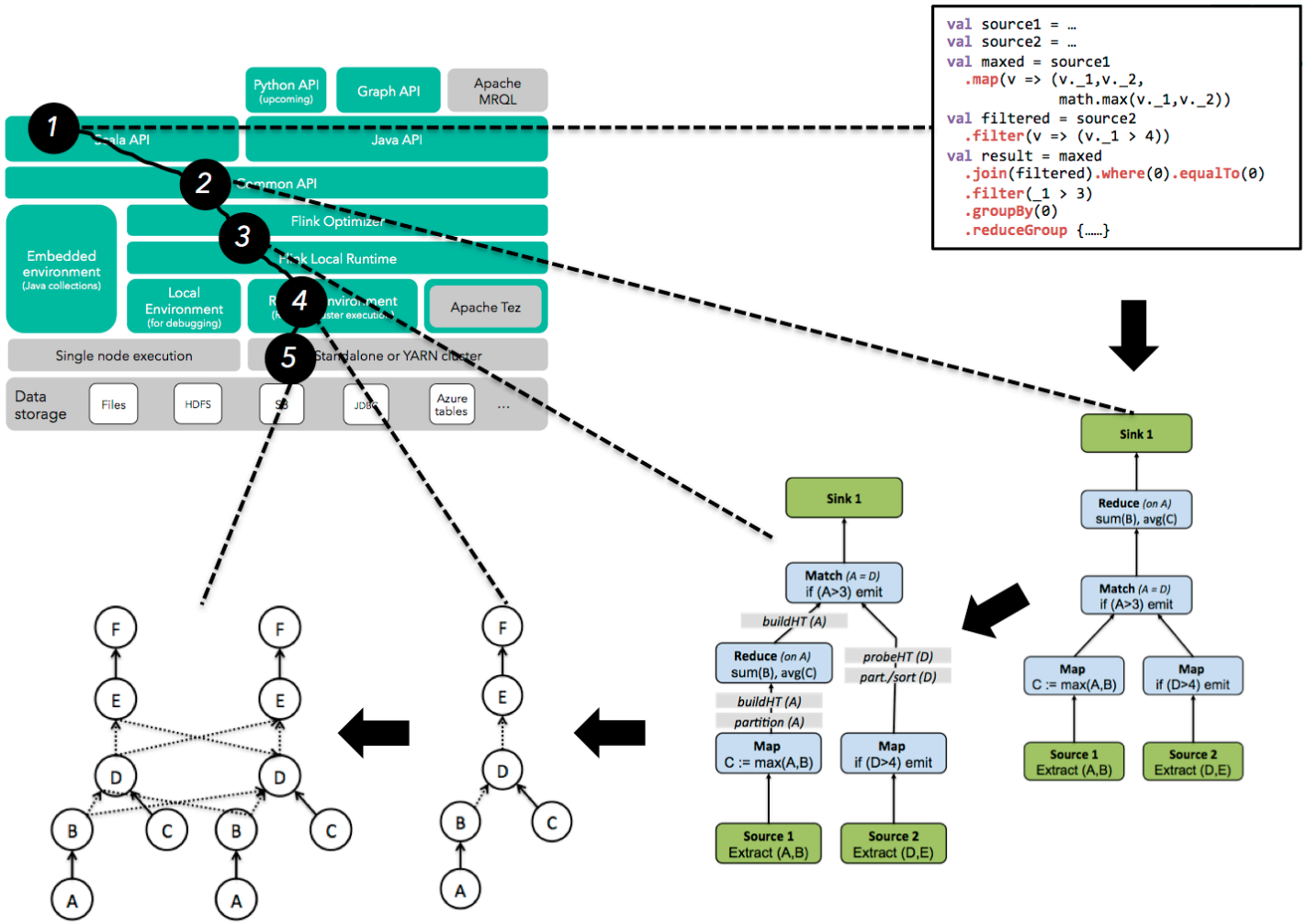
Figure: The JobManager is the coordinator of the Flink system
TaskManagers are the workers that execute parts of the parallel programs.

Apache Flink

A program is expressed as an arbitrary data flow consisting of transformations, sources and sinks.



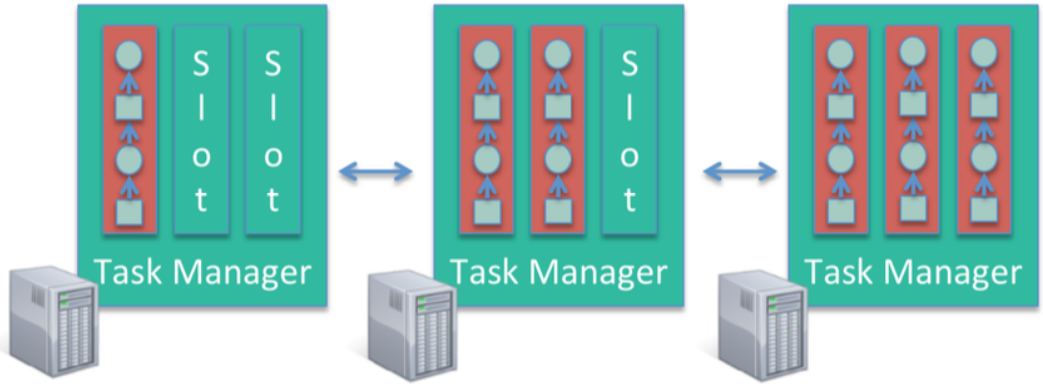
Apache Flink



Apache Flink

Task Manager

1. Operations are split up into **tasks** depending on the specified parallelism
2. Each parallel instance of an operation runs in a separate **task slot**
3. The scheduler may run several tasks from different operators in one task slot



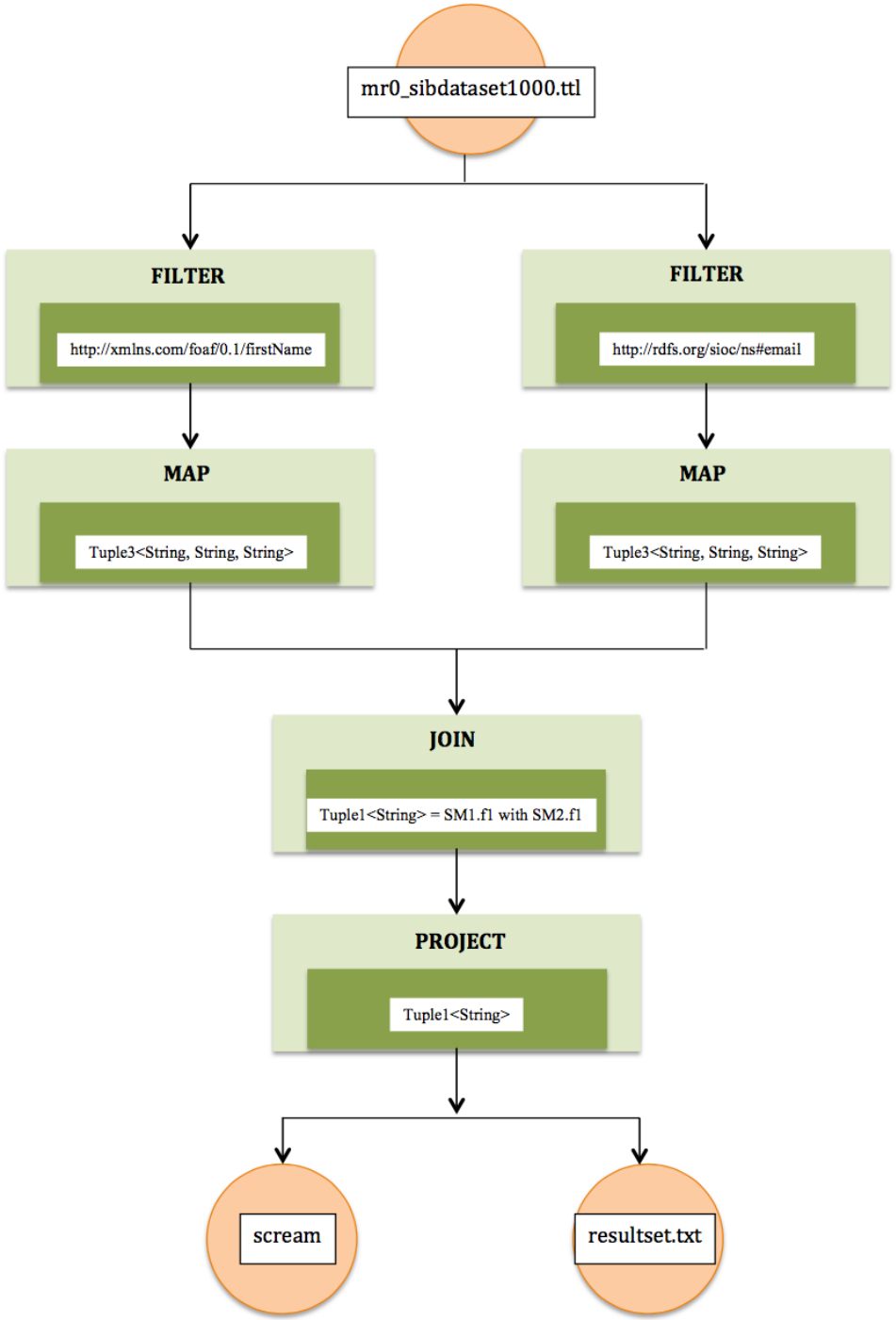
Contents

Contents

3. Current Work - batch

```
PREFIX  sibp: <http://www.ins.cwi.nl/sib/person/>
PREFIX  foaf: <http://xmlns.com/foaf/0.1>
PREFIX  sioc: <http://rdfs.org/sioc/ns#>

SELECT  ?name
FROM <mr0_sibdataset1000.ttl>
WHERE
  { ?person  foaf:firstName  ?name ;
             sioc:email      ?email
  }
```

```
(project (?name)
  (bgp
    (triple ?person <http://xmlns.com/foaf/0.1firstName> ?name)
    (triple ?person <http://rdfs.org/sioc/ns#email> ?email)
  ))
```


mr0_sibdataset1000.ttl
FILTER
http://xmlns.com/foaf/0.1/firstName
FILTER
http://rdfs.org/sioc/ns#email
MAP
Tuple3<String, String, String>
MAP
Tuple3<String, String, String>
JOIN
Tuple1<String> = SM1.f1 with SM2.f1
PROJECT
Tuple1<String>
scream
resultset.txt

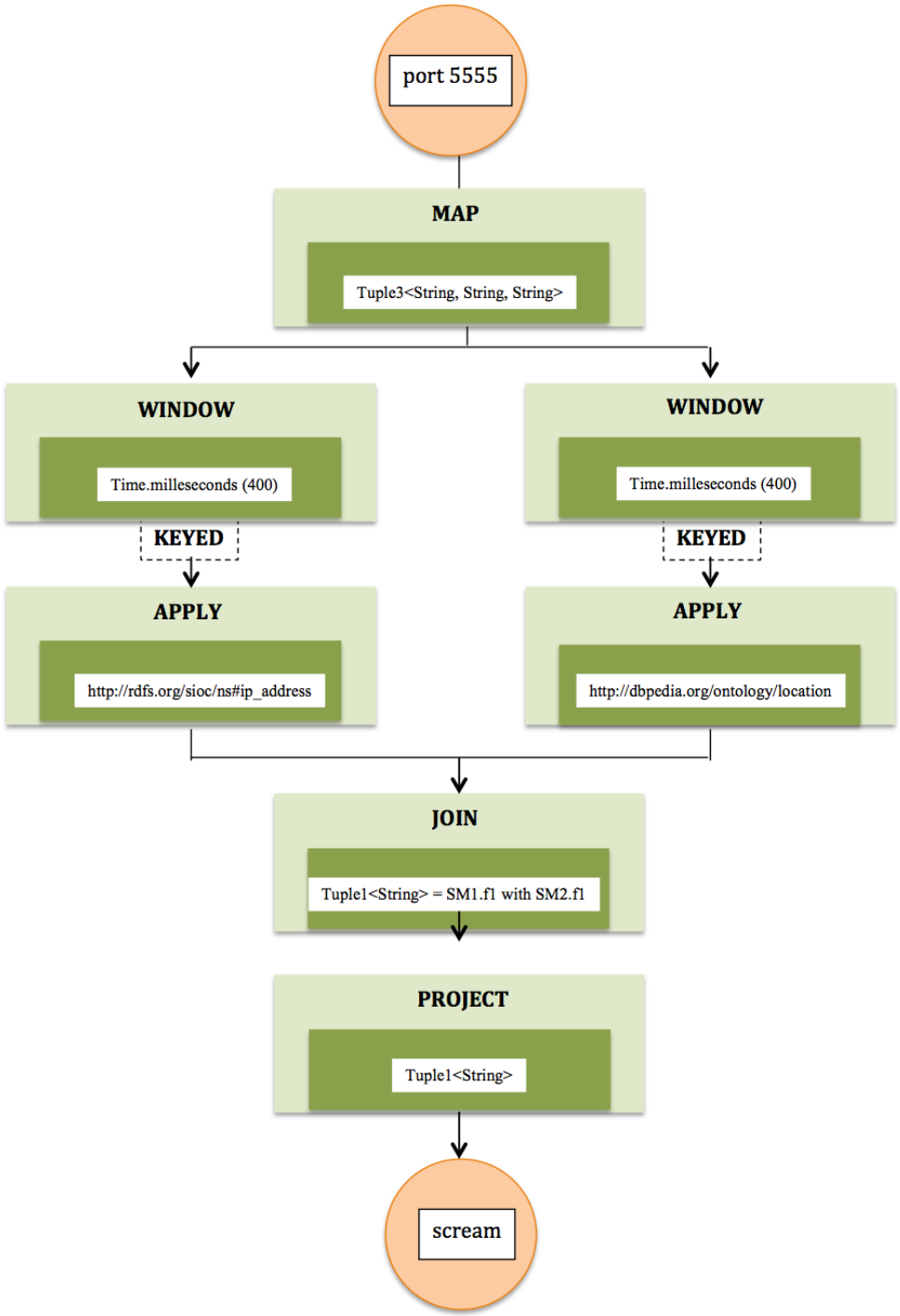
Contents

3. Current Work - streaming

```
String cqels_query = "PREFIX sibp: <http://www.ins.cwi.nl/sib/person/>" +
        "PREFIX foaf: <http://xmlns.com/foaf/0.1>" +
        "PREFIX sioc: <http://rdfs.org/sioc/ns#>" +
        "SELECT ?name " +
        "WHERE {" +
        "  STREAM <http://univalle.org/streams/rfid> [RANGE 400ms] " +
        "    { ?person foaf:firstName ?name ." +
        "    ?person sioc:email ?email }" +
        "}";
```

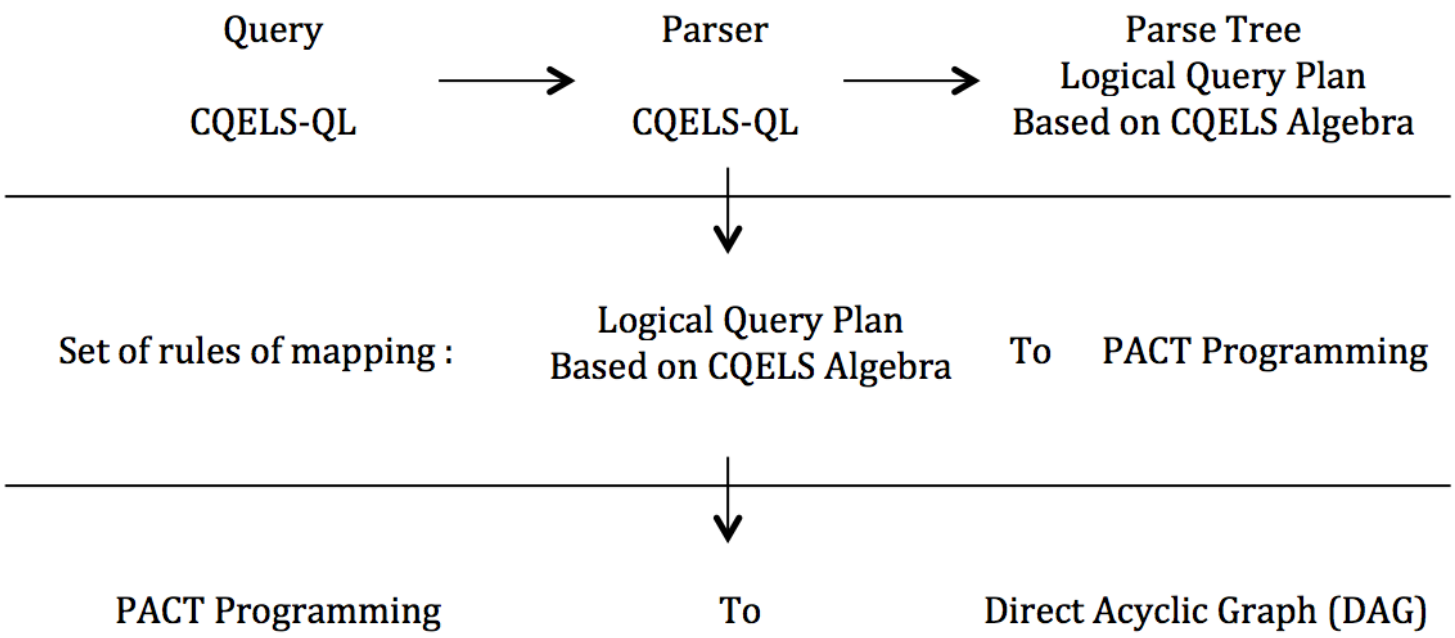
```
PREFIX  sibp: <http://
PREFIX  foaf: <http://
PREFIX  sioc: <http://

SELECT  ?name
WHERE
  { GRAPH <http://univalle.org/streams/rfid>
      { ?person foaf:firstName ?name .
        ?person sioc:email ?email
      }
  }
```

```
(project (?name)
  (graph <http://univalle.org/streams/rfid>
    (bgp
      (triple ?person <http://xmlns.com/foaf/0.1firstName> ?name)
      (triple ?person <http://rdfs.org/sioc/ns#email> ?email)
    )))
```


port 5555
MAP
Tuple3<String, String, String>
WINDOW
Time.milleseconds (400)
KEYED
APPLY
http://rdfs.org/sioc/ns#ip_address
WINDOW
Time.milleseconds (400)
KEYED
APPLY
http://dbpedia.org/ontology/location
JOIN
Tuple1<String> = SM1.f1 with SM2.f1
PROJECT
Tuple1<String>
scream

Contents

4. To do


Query
CQELS-QL
Parser
CQELS-QL
Parse Tree
Logical Query Plan
Based on CQELS Algebra
Set of rules of mapping :
Logical Query Plan
Based on CQELS Algebra
To
PACT Programming
PACT Programming
To
Direct Acyclic Graph (DAG)

4. To do

```
String cqels_query="PREFIX lv: <http://deri.org/floorplan/>" +
        "PREFIX dc: <http://purl.org/dc/elements/1.1/>" +
        "PREFIX foaf: <http://xmlns.com/foaf/0.1/>" +
        "SELECT ?loc2 ?locName (count(distinct ?coAuth) as ?noCoAuths)" +
        "FROM NAMED <http://deri.org/floorplan/>" +
        "WHERE {" +
        "   GRAPH <http://deri.org/floorplan/>" +
        "       {?loc2 lv:name ?locName." +
        "       ?loc2 lv:connected ?loc1}" +
        "   STREAM <http://deri.org/streams/rfid> [TRIPLES 1]" +
        "       {?auth lv:detectedAt ?loc1}" +
        "   STREAM <http://deri.org/streams/rfid> [RANGE 30s]" +
        "       {?coAuth lv:detectedAt ?loc2}" +
        "   {?paper dc:creator ?auth." +
        "   ?paper dc:creator ?coAuth." +
        "   ?auth foaf:name \"AUTHORNAME\"^^<http://www.w3.org/2001/XMLSchema#string> }" +
        "   FILTER(?auth!=?coAuth)" +
        "}" +
        "GROUP BY ?loc2 ?locName";
```

4. To do

```
PREFIX  lv:   <http://deri.org/floorplan/>
PREFIX  foaf: <http://xmlns.com/foaf/0.1/>
PREFIX  dc:   <http://purl.org/dc/elements/1.1/>

SELECT  ?loc2 ?locName (count(distinct ?coAuth) AS ?noCoAuths)
FROM NAMED lv:
WHERE
  { GRAPH lv:
      { ?loc2 lv:name ?locName .
        ?loc2 lv:connected ?loc1
      }
    GRAPH <http://deri.org/streams/rfid>
      { ?auth lv:detectedAt ?loc1 }
    GRAPH <http://deri.org/streams/rfid>
      { ?coAuth lv:detectedAt ?loc2 }
    { ?paper dc:creator ?auth .
      ?paper dc:creator ?coAuth .
      ?auth foaf:name "AUTHORNAME"^^<http://www.w3.org/2001/XMLSchema#string>
    }
    FILTER ( ?auth != ?coAuth )
  }
GROUP BY ?loc2 ?locName
```

4. To do

```
(project (?loc2 ?locName ?noCoAuths)
  (extend ((?noCoAuths ?.0))
    (group (?loc2 ?locName) ((?.0 (count distinct ?coAuth)))
      (filter (!= ?auth ?coAuth)
        (join
          (join
            (join
              (graph <http://deri.org/floorplan/>
                (bgp
                  (triple ?loc2 <http://deri.org/floorplan/name> ?locName)
                  (triple ?loc2 <http://deri.org/floorplan/connected> ?loc1)
                ))
              (graph <http://deri.org/streams/rfid>
                (bgp (triple ?auth <http://deri.org/floorplan/detectedAt> ?loc1))))
            (graph <http://deri.org/streams/rfid>
              (bgp (triple ?coAuth <http://deri.org/floorplan/detectedAt> ?loc2))))
          (bgp
            (triple ?paper <http://purl.org/dc/elements/1.1/creator> ?auth)
            (triple ?paper <http://purl.org/dc/elements/1.1/creator> ?coAuth)
            (triple ?auth <http://xmlns.com/foaf/0.1/name> "AUTHORNAME"^^<http://www.w3.org/2001/XMLSchema#string>)
          ))))))
```

References

[1] Vincenzo Gulisano, Ricardo Jimenez-Peris, Marta Patino-Martinez, Claudio Sorien- te, and Patrick Valduriez. Streamcloud: An elastic and scalable data streaming system. IEEE Trans. Parallel Distrib. Syst., 23(12):2351–2365, December 2012.

[2] Jean-Paul Calbimonte, Oscar Corcho, and AlasdairJ.G. Gray. Enabling ontology- based access to streaming data sources. In ISWC 2010, volume 6496, pages 96–111. Springer Berlin Heidelberg, 2010.

[3] Juan Sequeda and Oscar Corcho. Linked stream data: A position paper. In SSN09, pages 148–157, 2009.

[4] Danh Le-Phuoc, Minh Dao-Tran, Josiane Xavier Parreira, and Manfred Hauswirth. A native and adaptive approach for unified processing of linked streams and linked data. In Proceedings of the 10th International Conference on The Semantic Web - Volume Part I, ISWC'11, pages 370–388, Berlin, Heidelberg, 2011

[5] Danh Le-Phuoc, Hoan Nguyen Mau Quoc, Chan Le Van, and Manfred Hauswirth. Elastic and scalable processing of linked stream data in the cloud. In The Semantic Web – ISWC 2013, volume 8218 of Lecture Notes in Computer Science, pages 280–297. Springer Berlin Heidelberg, 2013.

[6] IBM, Paul Zikopoulos, and Chris Eaton. Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. McGraw-Hill Osborne Media, 1st edition, 2011.