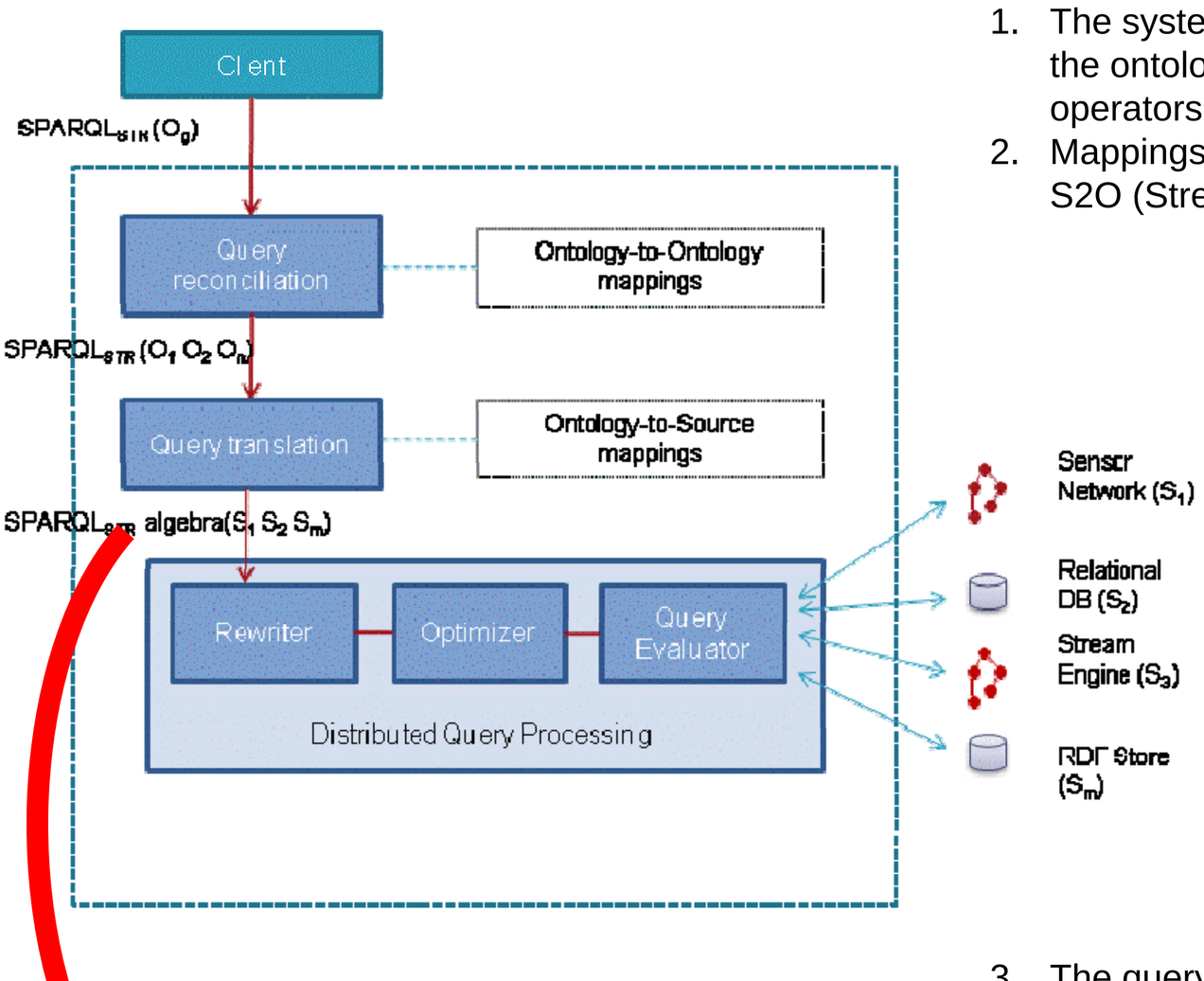
Enabling Ontology-based Access to Streaming Data Sources

Jean Paul Calbimonte[1], Oscar Corcho[1], Alasdair J G Gray[2]
[1] Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
[2] Information Management Group, University of Manchester, United Kingdom

Summary: The availability of streaming data sources is progressively increasing thanks to the development of ubiquitous data capturing technologies such as sensor networks. The heterogeneity of these sources introduces the requirement of providing data access in a unified and coherent manner, whilst allowing users to express their needs at an ontological level. We propose an ontology-based streaming data access service, where sources link their data content to ontologies through S2O mappings, and then users can query these sources using $SPARQL_{Stream}$, an extension of SPARQL for data streams.

Architecture and Query Processing Steps



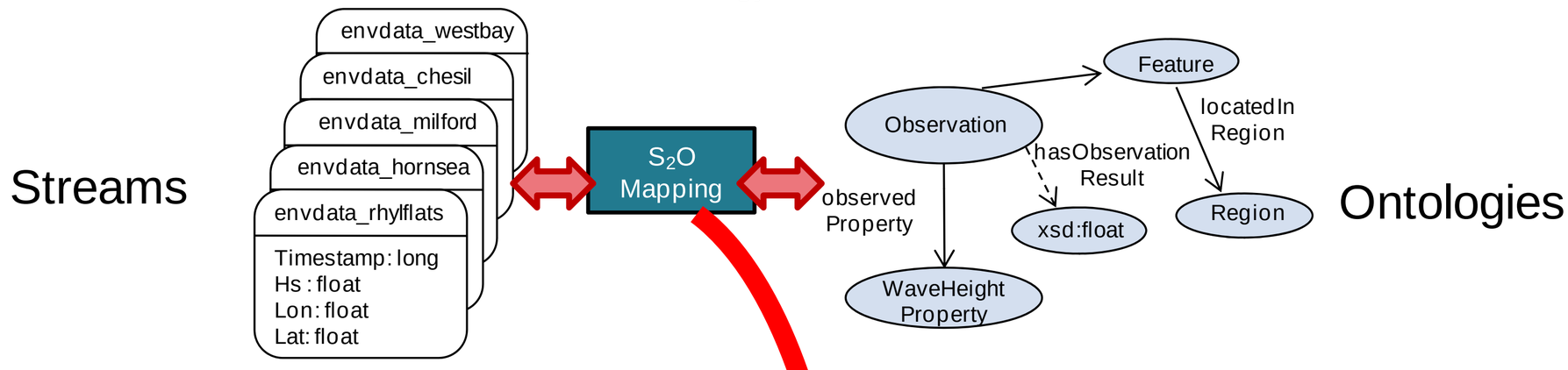
1. The system receives queries specified in terms of the classes and properties of the ontology using $SPARQL_{Stream}$, an extension of SPARQL that supports operators over RDF streams
2. Mappings between the data sources and the ontologies are expressed in the S2O (Stream to Ontology) mapping language.

```
streamschema-desc
    name WaveSensors
    has-stream envdata_rhylflats
        streamType pulled
        documentation "Wave measurements"
        keycol-desc Timestamp
            columnType integer
        timestamp-desc  Timestamp
            columnType datetime
        nonkeycol-desc  Hs
            columnType float
        nonkeycol-desc Lon
            columnType float
...

conceptmap-def Observation
    virtualStream <http://semsorgrid4env.eu/ccometeo.srdf>
    uri-as
        concat(rhylflats.Timestamp)
    applies-if
        <cond-expr>
    described-by
        attributemap-def observationResult
            virtualStream http://semsorgrid4env.eu/ccometeo.srdf>
            operation constant
                has-column envdata.rhylflats.Hs
```

3. The query translation process transforms from $SPARQL_{Stream}$ to SNEEql
4. Query processing starts, and the evaluator uses distributed query processing techniques to extract the relevant data from the sources and to perform the required query processing, e.g. selections, projections and joins.
5. The result of the query processing is a set of tuples that are transformed into ontology instances.

Execution Example




```
PREFIX cd: <http://www.semsorgrid4env.eu/ontologies/CoastalDefences.owl#>
PREFIX sb: <http://www.w3.org/2009/SSN-XG/Ontologies/SensorBasis.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT  ?waveheight ?wavets  ?lat ?lon
FROM STREAM <http://www.semsorgrid4env/ccometeo.srdf>
WHERE
{
 ?WaveObs a cd:Observation;
    cd:observationResult ?waveheight;
    cd:observationResultTime ?wavets;
    cd:observationResultLatitude ?lat;
    cd:observationResultLongitude ?lon;
    cd:observedProperty ?waveProperty;
    cd:featureOfInterest ?waveFeature.
 ?waveFeature  a cd:Feature;
    cd:locatedInRegion cd:SouthEastEnglandCCO.
 ?waveProperty a cd:WaveHeight.
 }
```

```
(SELECT Lon,timestamp,Hs,Lat FROM envdata_rhylflats) UNION
(SELECT Lon,timestamp,Hs,Lat FROM envdata_hornsea)  UNION
(SELECT Lon,timestamp,Hs,Lat FROM envdata_milford) UNION
(SELECT Lon,timestamp,Hs,Lat FROM envdata_chesil) UNION
(SELECT Lon,timestamp,Hs,Lat FROM envdata_perranporth) UNION
(SELECT Lon,timestamp,Hs,Lat FROM envdata_westbay) UNION
(SELECT Lon,timestamp,Hs,Lat FROM envdata_pevenseybay)
```

References

[1] JP Calbimonte, O Corcho, AJG Gray. Enabling Ontology-based Access to Streaming Data Sources. Proceedings of the 9th International Semantic Web Conference (ISWC-2010). Shanghai, China. Nov 2010.