TRIPLEWAVE: SPREADING RDF STREAMS ON THE WEB

Andrea Mauri, Jean-Paul Calbimonte, Daniele Dell'Aglio, Marco Balduini, Marco Brambilla, Emanuele Della Valle, Karl Aberer

Universität Zürich[UZH]

TripleWave is an open-source framework for **creating** and **publishing** RDF streams over the Web.

Why

- Streams are getting a momentum
- **Standard protocols** and **exchanging mechanisms** for RDF stream are **missing**.
- We need **generic** and **flexible** solutions for making RDF streams available on the Web.

What: Input

TripleWave *should* support a variety of data sources.

- Streams available on the Web
- RDF dumps with temporal information
- RDF with temporal information exposed through SPARQL endpoints

What: Output

TripleWave *should* exploit and be compatible to existing standards and recommendations

- The output format compatible with RDF

What: Output

TripleWave *should* exploit and be compatible to existing standards and recommendations

- The output format compatible with RDF

TripleWave *must* be able to provide the stream to processing engine through the Web.

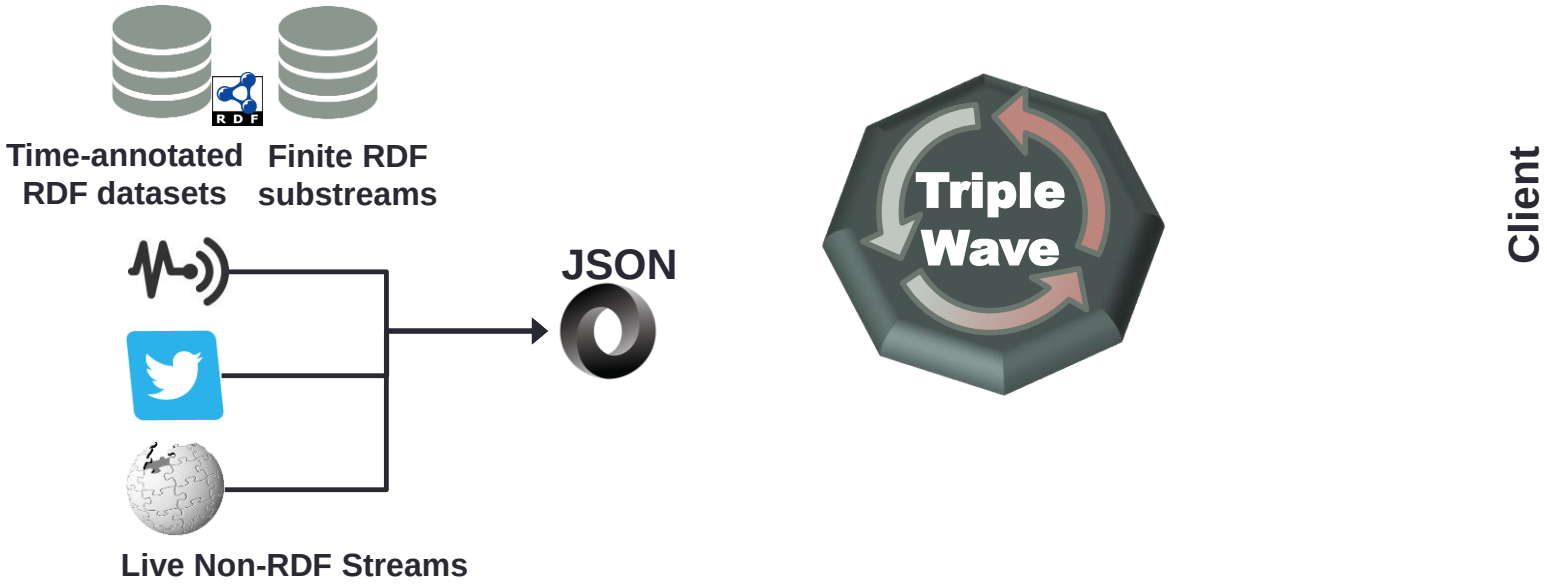
- HTTP
- HTTP chunk
- Web sockets

What: Interaction with RSP

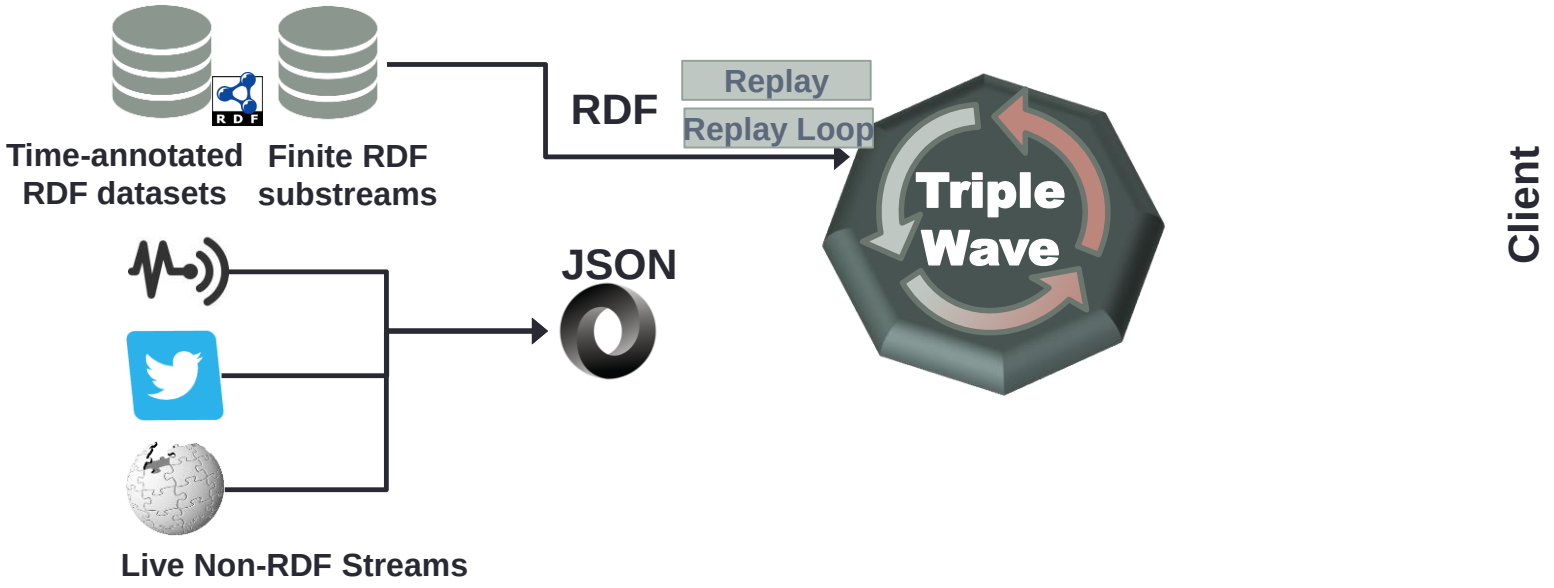
TripleWave *should* be able to interact with **RSP Services**, that use continuous execution model

- The metadata and the schema of the data should be separated from the stream itself.

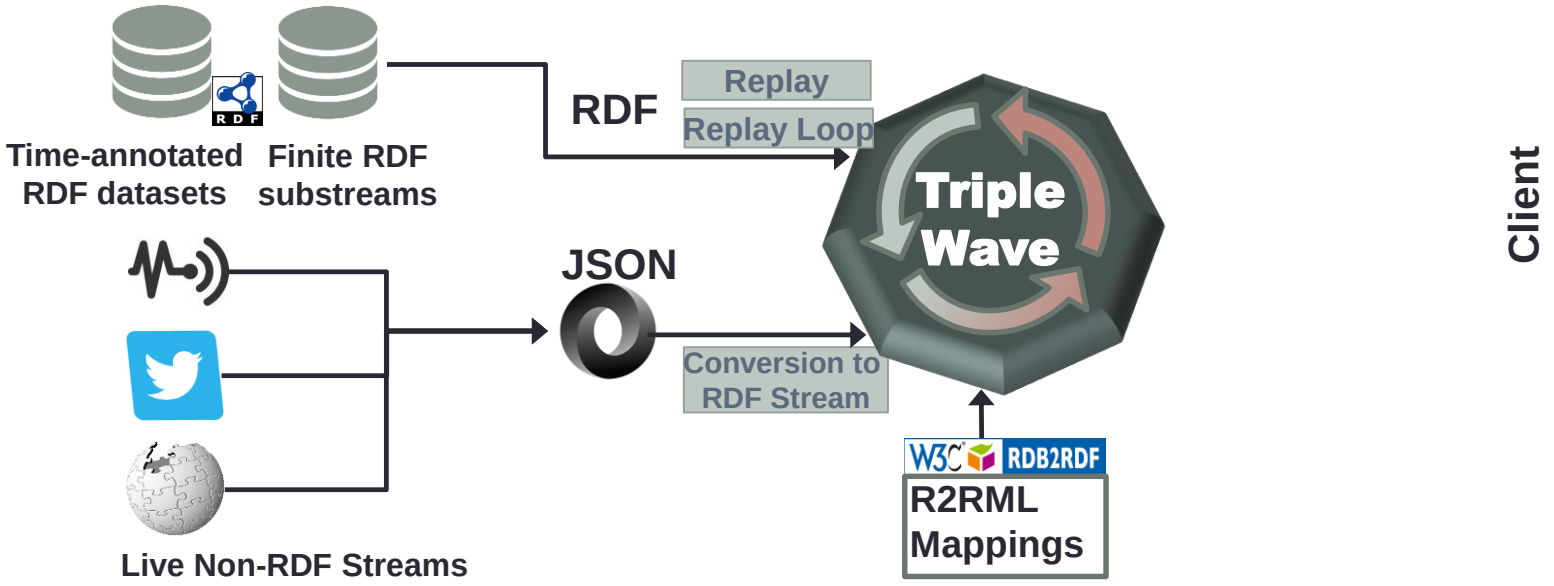
How: Architecture



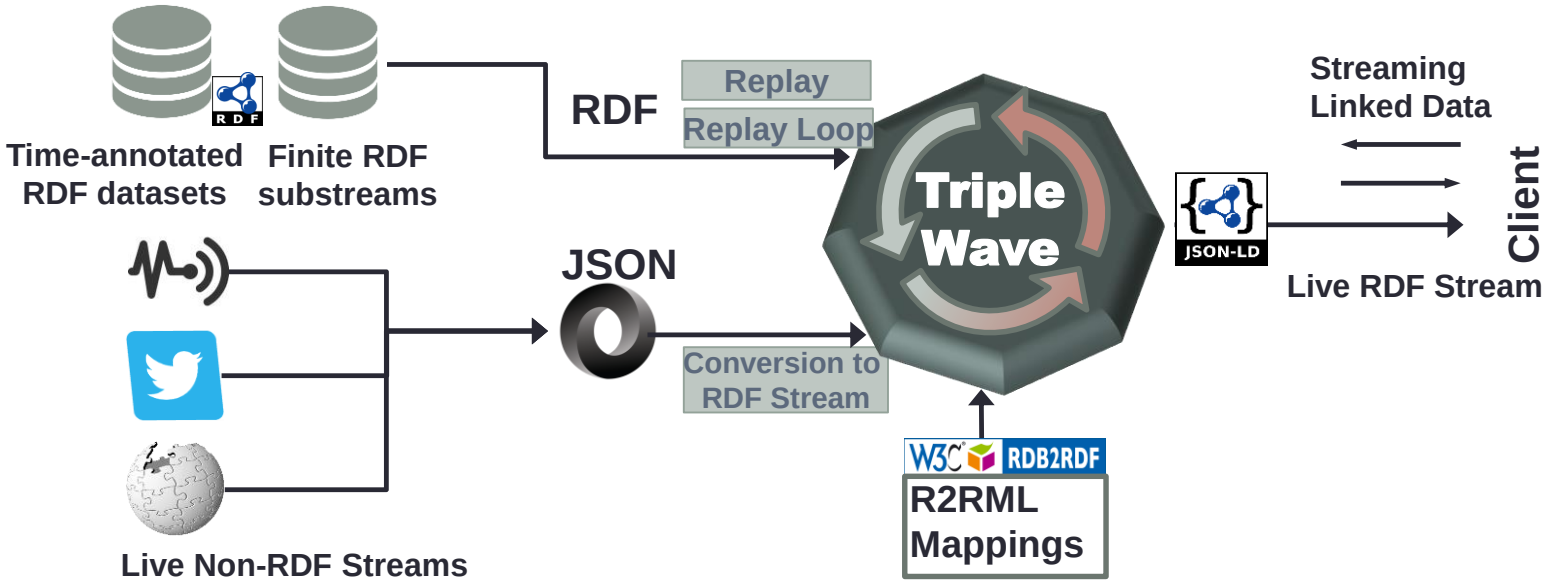
How: Architecture



How: Architecture



How: Architecture



How: From RDF to RDF streams

- Converts …
  - RDF stored in files/SPARQL endpoints
  - Containing some time information

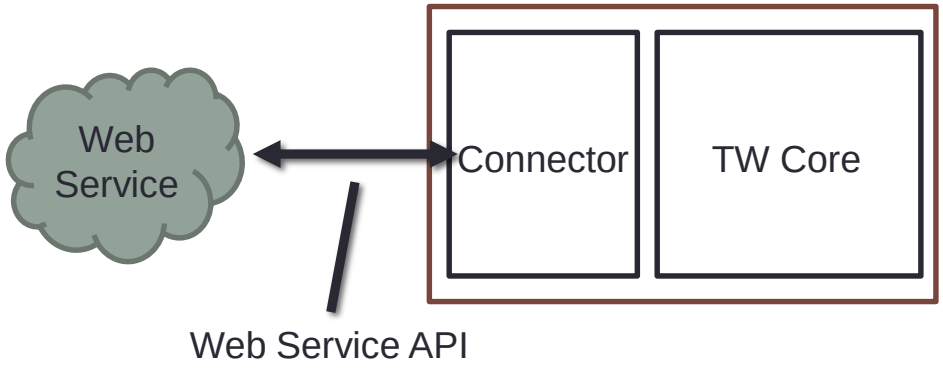
How: From RDF to RDF streams

- Converts …
  - RDF stored in files/SPARQL endpoints
  - Containing some time information
- … into an RDF stream
  - **continuous flow** of RDF data
  - **ordered** according the original timestamps
  - the time between two items is preserved

How: From RDF to RDF streams

- Converts …
  - RDF stored in files/SPARQL endpoints
  - Containing some time information
- … into an RDF stream
  - **continuous flow** of RDF data
  - **ordered** according the original timestamps
  - the time between two items is preserved
- **Use Cases**
  - Evaluation, testing and benchmarking
  - Simulation systems

How: From Web stream to RDF stream



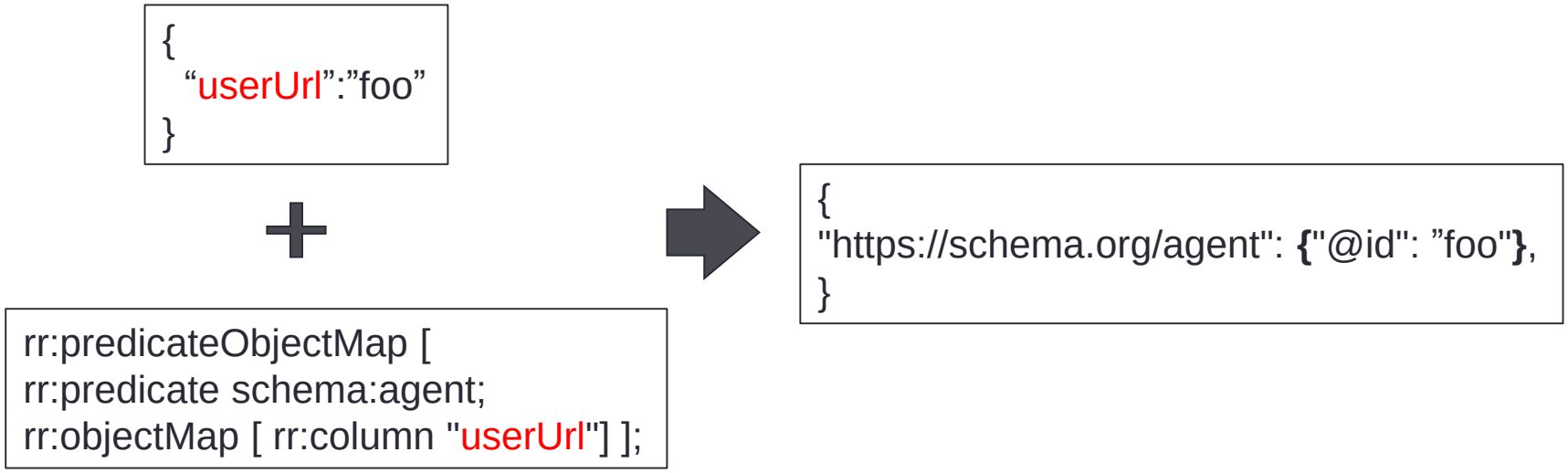
- Consumes ...
  - An existing Web stream through connectors
- And converts it into an RDF Stream
  - Each data item is lifted to RDF
- **Use Cases**
  - Querying and reasoning
  - Data integration

How: From Web stream to RDF stream

Convertions is made through **R2RML**

- Mappings to convert each data item in RDF

Example: map a field

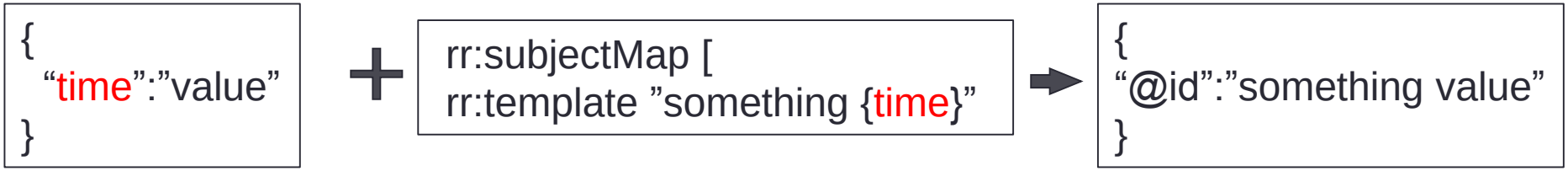


How: From Web stream to RDF stream

Convertions is made through **R2RML**

- Mappings to convert each data item in RDF

Example: map a field with template



How: From Web stream to RDF stream

Convertions is made through **R2RML**

- Mappings to convert each data item in RDF

Example:

- Add a new constant field

```
  rr:predicateObjectMap
[    rr:predicate rdf:type;  rr:objectMap
[ rr:constant schema:UpdateAction]];
```

```
{
"http://www.w3.org/1999/02/22-rdf-syntax-ns#type":
                    {"@id": "https://schema.org/UpdateAction"}
}
```

Implementation

TripleWave is a **NodeJS** Web Application

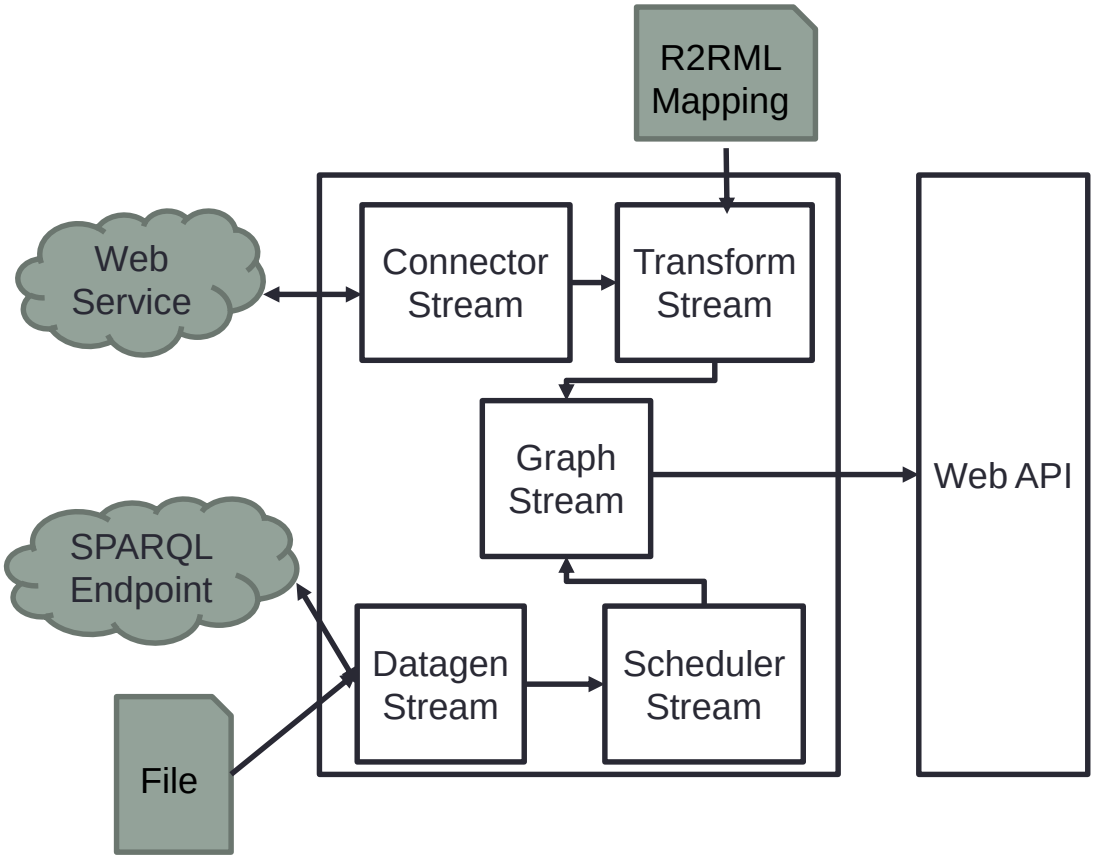
NodeJS is a JavaScript runtime built on Chrome's V8 JavaScript engine.

TripleWave is released with a Apache 2.0 License and the source code is hosted on github at:

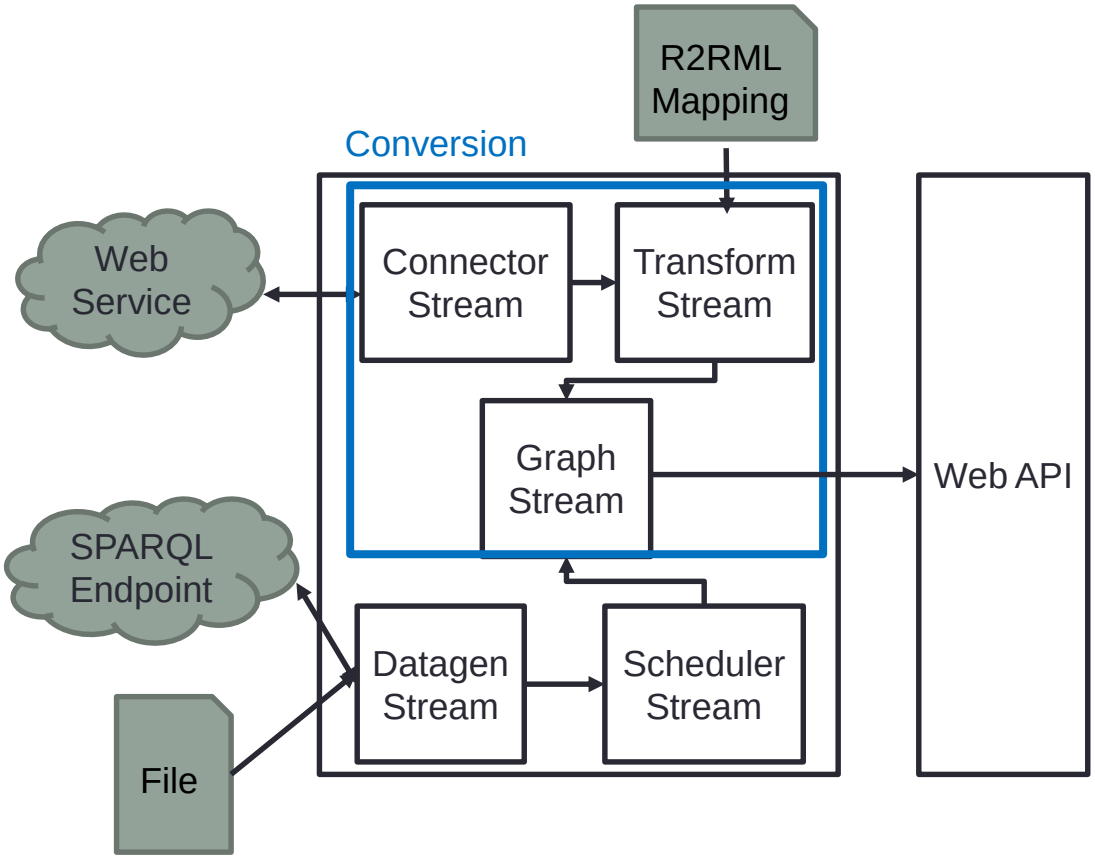
- https://github.com/streamreasoning/TripleWave

- It’s maintained by the Stream Reasoning Initiative

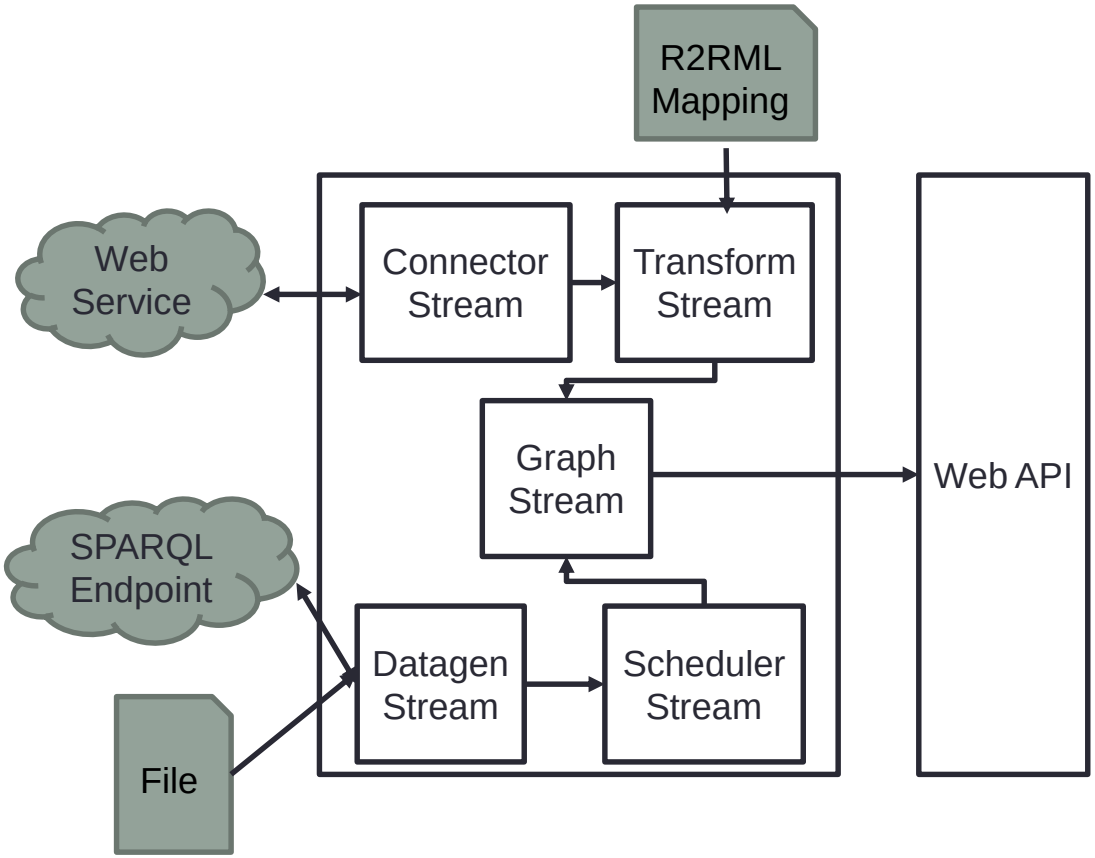
TripleWave Real* Architecture



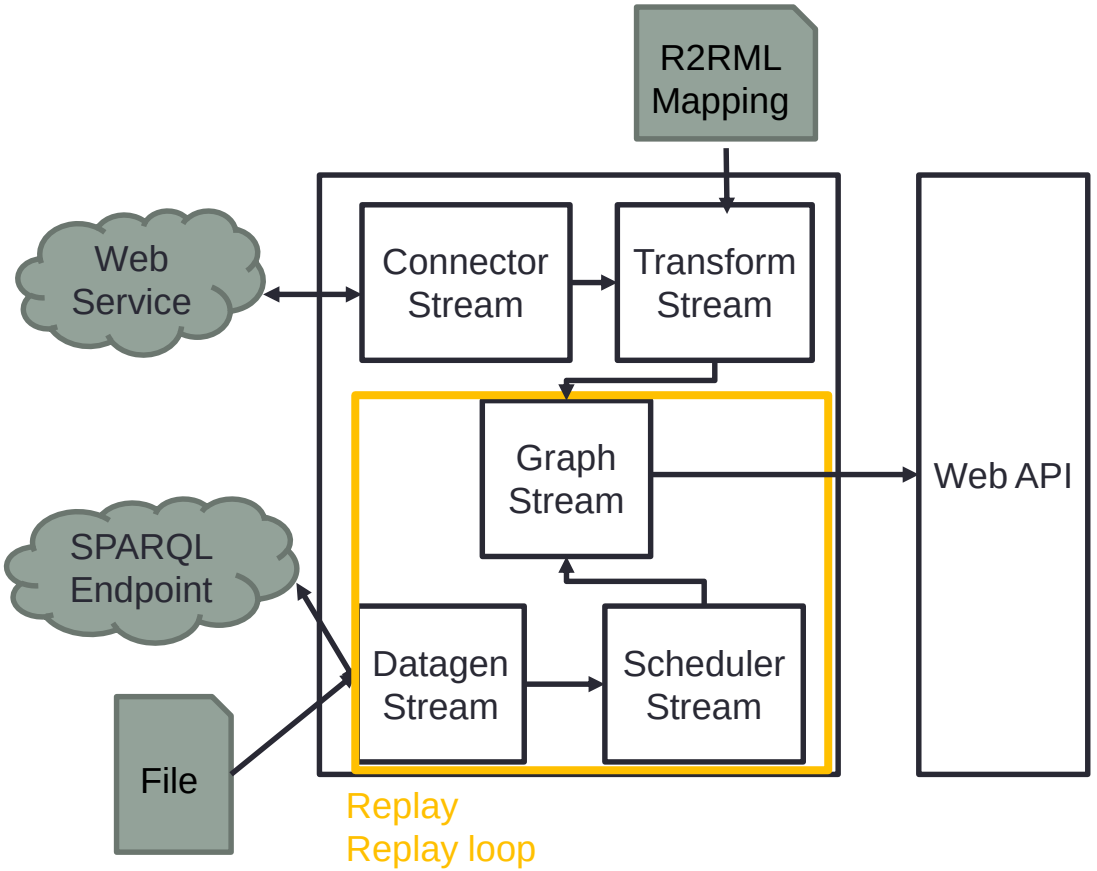
TripleWave Real* Architecture



TripleWave Real* Architecture



TripleWave Real* Architecture



Consuming TripleWave RDF Stream

TripleWave publish the stream with *both* pull and *push* mechanism.

- The **pull** method follows the Linked Data principles
- The **push** method exploit a RSP service

Consuming TripleWave RDF Stream - Pull

TripleWave distinguish two type of named graph:

- Stream Graph (**sGraph**): describes the stream
- Instantaneous Graph (**iGraph**): represents the single stream element

Furthermore is possible to access **directly** to the stream

Consuming TripleWave RDF Stream - sGraph

The sGraph contains metadata about the graph, e.g.,

- The identifier of the stream
- Data item samples (see next slide)
- A description of the schema
- The location of the stream endpoint

Consuming TripleWave RDF Stream - iGraph

The iGraph

- embeds an item of an RDF stream
- is annotated with temporal information
  - Creation time
  - Extendible to other time annotations

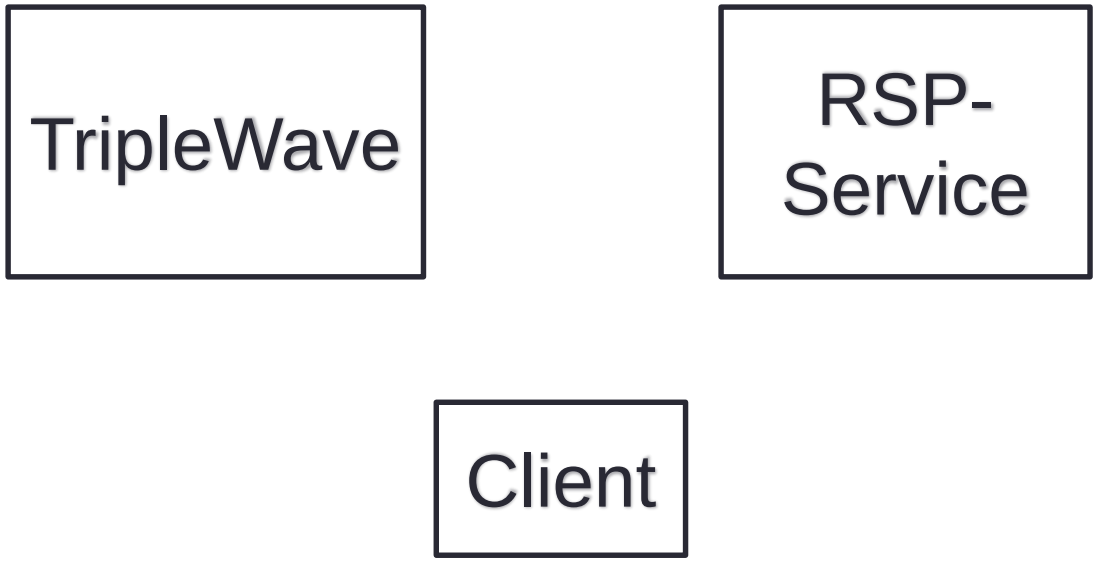
Consuming TripleWave RDF Stream - Example

```
{
  "@context":{
    "sld":"http://streamreasoning.org/ontologies/SLD4TripleWave#",
    "generatedAt":{
      "@id":"http://www.w3.org/ns/prov#generatedAtTime",
      "@type":"http://www.w3.org/2001/XMLSchema#dateTime"
    }
  },
  "@id":"tr:sGraph",
  "sld:contains":{
    "@list":[
      {
        "generatedAt":"2016-10-18T03:35:32.425Z",
        "@id":"http://131.175.141.249/TripleWave-endless/Observation_AirTemperatur
      },
      {
        "generatedAt":"2016-10-18T03:35:32.496Z",
        "@id":"http://131.175.141.249/TripleWave-endless/Observation_AirTemperatur
      },
      {
        "generatedAt":"2016-10-18T03:35:32.546Z",
        "@id":"http://131.175.141.249/TripleWave-endless/Observation_AirTemperatur
      }
    ]
  },
  "sld:streamLocation":"ws://131.175.141.249/TripleWave-endless/ws/stream",
  "sld:tBoxLocation":{
    "@id":"http://purl.oclc.org/NET/ssnx/ssn"
  },
  "sld:lastUpdated":"2016-10-18T03:35:32.546Z"
}
```


```
{
  "@graph":{
    "@id":"http://knoesis.wright.edu/ssw/Observation_RelativeHumidity_C1390_2004_08_08_06_05_00",
    "@type":"http://knoesis.wright.edu/ssw/ont/weather.owl#RelativeHumidityObservation",
    "observedProperty":"http://knoesis.wright.edu/ssw/ont/weather.owl#_RelativeHumidity",
    "procedure":"http://knoesis.wright.edu/ssw/System_C1390",
    "result":"http://knoesis.wright.edu/ssw/MeasureData_RelativeHumidity_C1390_2004_08_08_06_05_00",
    "samplingTime":"http://knoesis.wright.edu/ssw/Instant_2004_08_08_06_05_00",
    "@context":{
      "procedure":{
        "@id":"http://knoesis.wright.edu/ssw/ont/sensor-observation.owl#procedure",
        "@type":"@id"
      },
      "samplingTime":{
        "@id":"http://knoesis.wright.edu/ssw/ont/sensor-observation.owl#samplingTime",
        "@type":"@id"
      },
      "observedProperty":{
        "@id":"http://knoesis.wright.edu/ssw/ont/sensor-observation.owl#observedProperty",
        "@type":"@id"
      },
      "result":{
        "@id":"http://knoesis.wright.edu/ssw/ont/sensor-observation.owl#result",
        "@type":"@id"
      }
    }
  }
}
```

Consuming TripleWave RDF Stream - Push

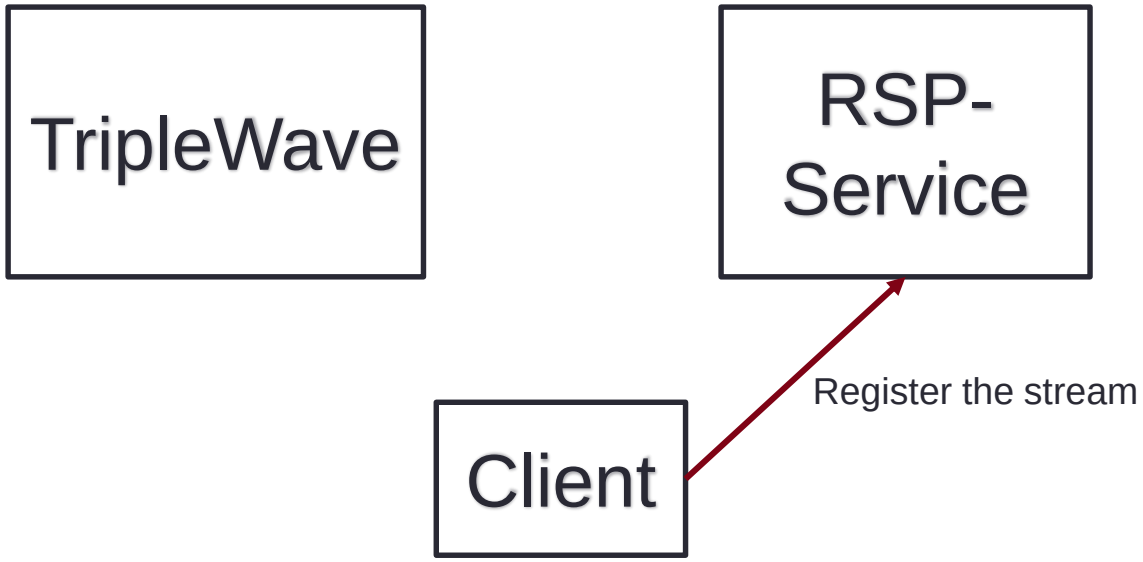
The TripleWave stream can be consumed via push by extending the RSP service framework[1]



[1]https://github.com/streamreasoning/rsp-services

Consuming TripleWave RDF Stream - Push

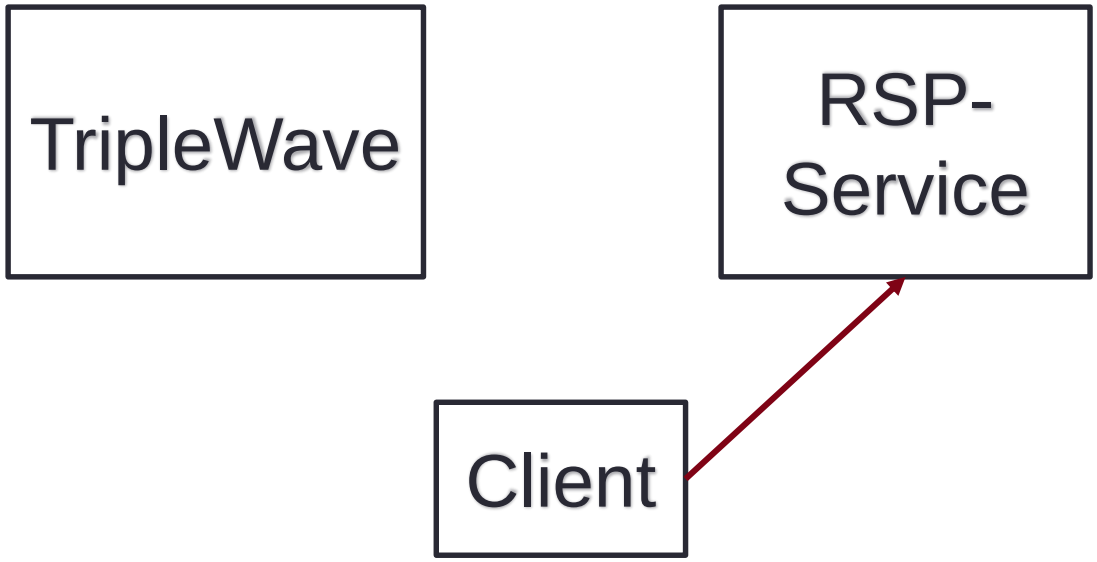
The TripleWave stream can be consumed via push by extending the RSP service framework[1]



[1]https://github.com/streamreasoning/rsp-services

Consuming TripleWave RDF Stream - Push

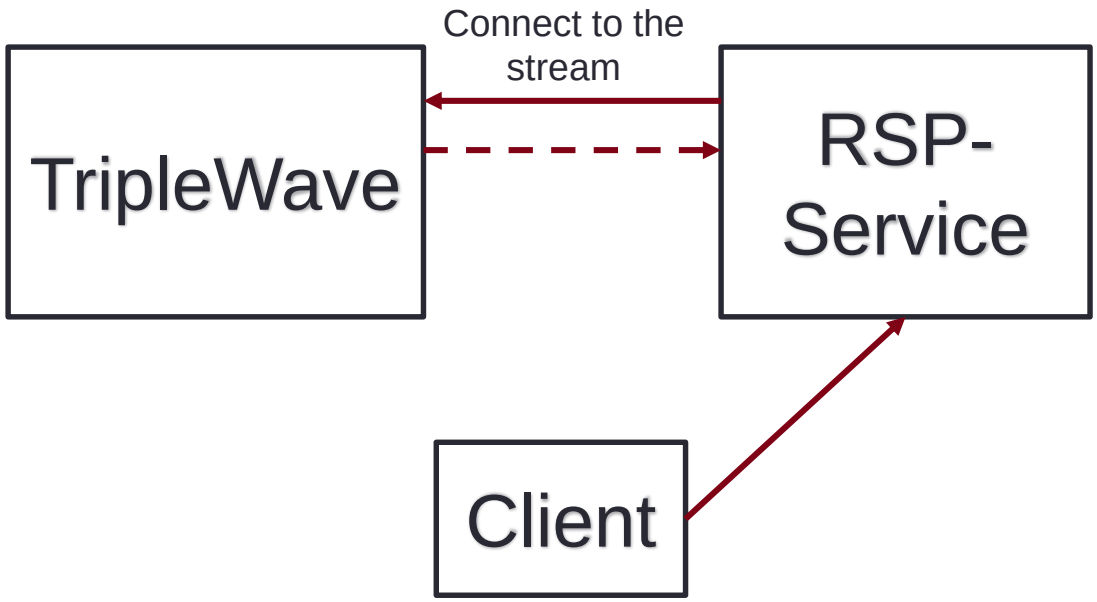
The TripleWave stream can be consumed via push by extending the RSP service framework[1]



[1]https://github.com/streamreasoning/rsp-services

Consuming TripleWave RDF Stream - Push

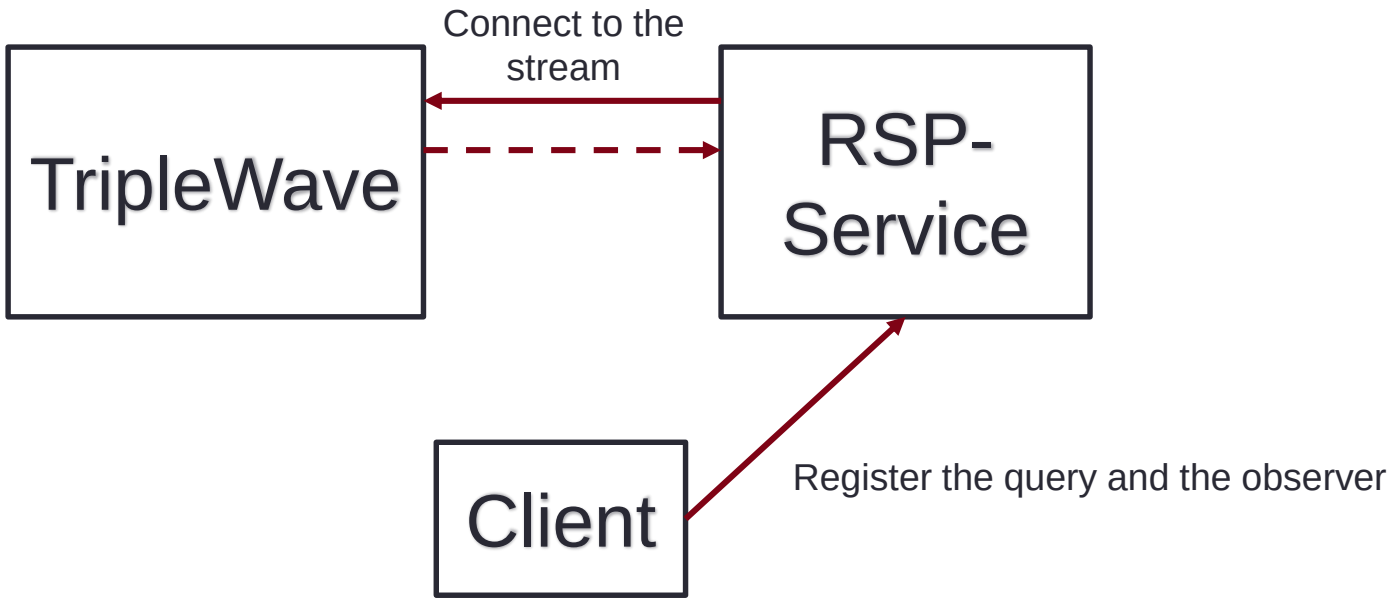
The TripleWave stream can be consumed via push by extending the RSP service framework[1]



[1]https://github.com/streamreasoning/rsp-services

Consuming TripleWave RDF Stream - Push

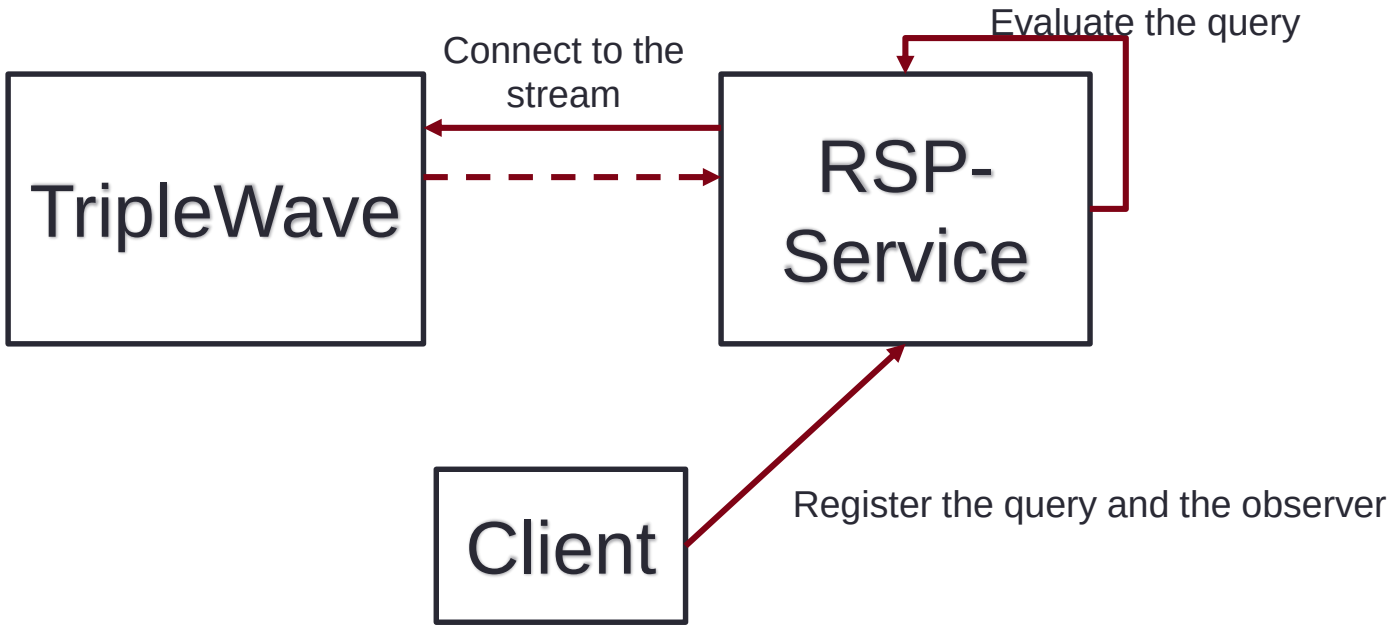
The TripleWave stream can be consumed via push by extending the RSP service framework[1]



[1]https://github.com/streamreasoning/rsp-services

Consuming TripleWave RDF Stream - Push

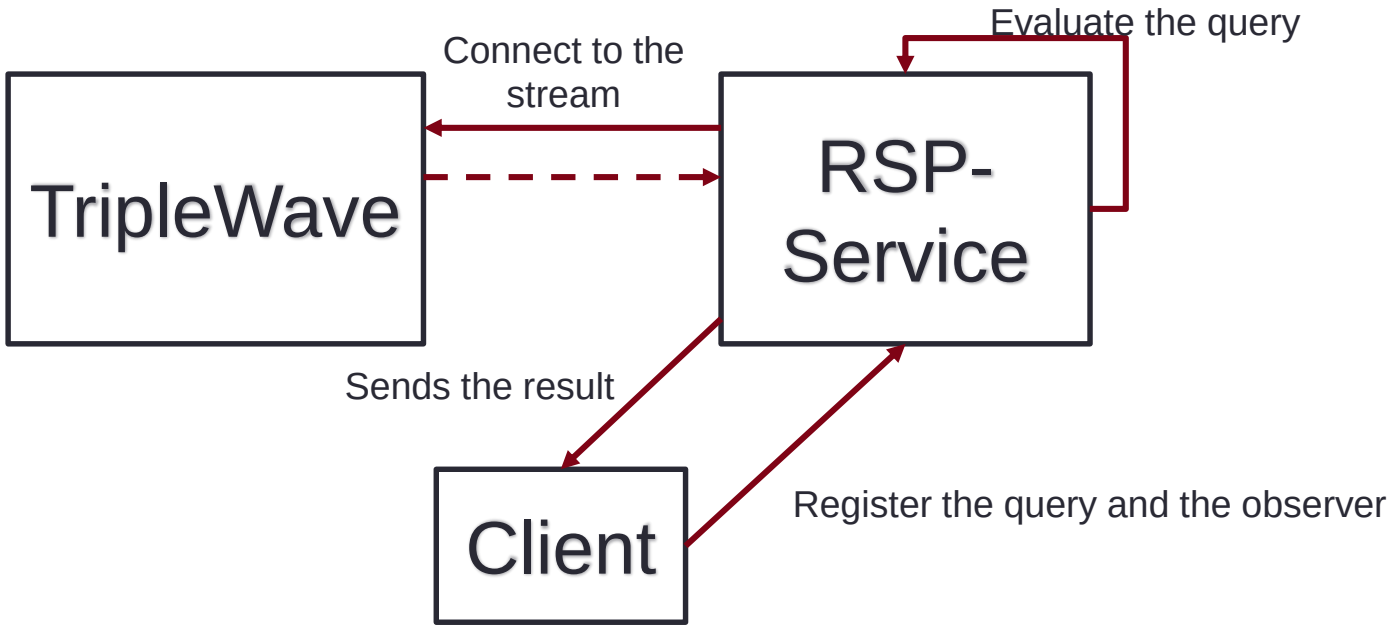
The TripleWave stream can be consumed via push by extending the RSP service framework[1]



[1]https://github.com/streamreasoning/rsp-services

Consuming TripleWave RDF Stream - Push

The TripleWave stream can be consumed via push by extending the RSP service framework[1]



[1]https://github.com/streamreasoning/rsp-services

Show cases

Three **demos** have been developed to show the capabilities of the system.

- Wikipedia changes stream conversion.
  - http://131.175.141.249/TripleWave-transform/sgraph
- Endlessly replay as a stream the Linked Sensor Data dataset.
  - http://131.175.141.249/TripleWave-endless/sgraph
- Endlessly replay as a stream the LDBC social graph dataset.
  - http://131.175.141.249/TripleWave-ldbc/sgraph

Show cases (2)

A fourth demo was shown at the **Tutorial on RDF-Stream Processing** Tutorial.

http://tinyurl.com/csparqlgui

- This demo allows you to:
  - Register the TripleWave stream to C-SPARQL (you need to change the url of the stream in the my.js file)
  - Register a query
  - Look at results

Conclusion

- TripleWave is an open-source framework for publishing and sharing RDF streams on the Web.
  - It fills an important gap in the RDF stream reasoning.
  - It covers a set of crucial **requirements** for both the stream reasoning and the semantic Web community.
  - We think it will help spreading the **adoption** of RDF for streaming data scenarios and applications

Thank you for the attention

Andrea Mauri
andrea.mauri@polimi.it

https://github.com/streamreasoning/TripleWave