Tutorial on RDF Stream Processing 2016

M.I. Ali, J-P Calbimonte, D. Dell'Aglio, E. Della Valle, and A. Mauri

http://streamreasoning.org/events/rsp2016

ISWC 2016 Kobe, Japan The 15th International Semantic Web Conference

Insight

Hes·so VALAIS WALLIS

Universität Zürich[UZH]

POLITECNICO MILANO 1863

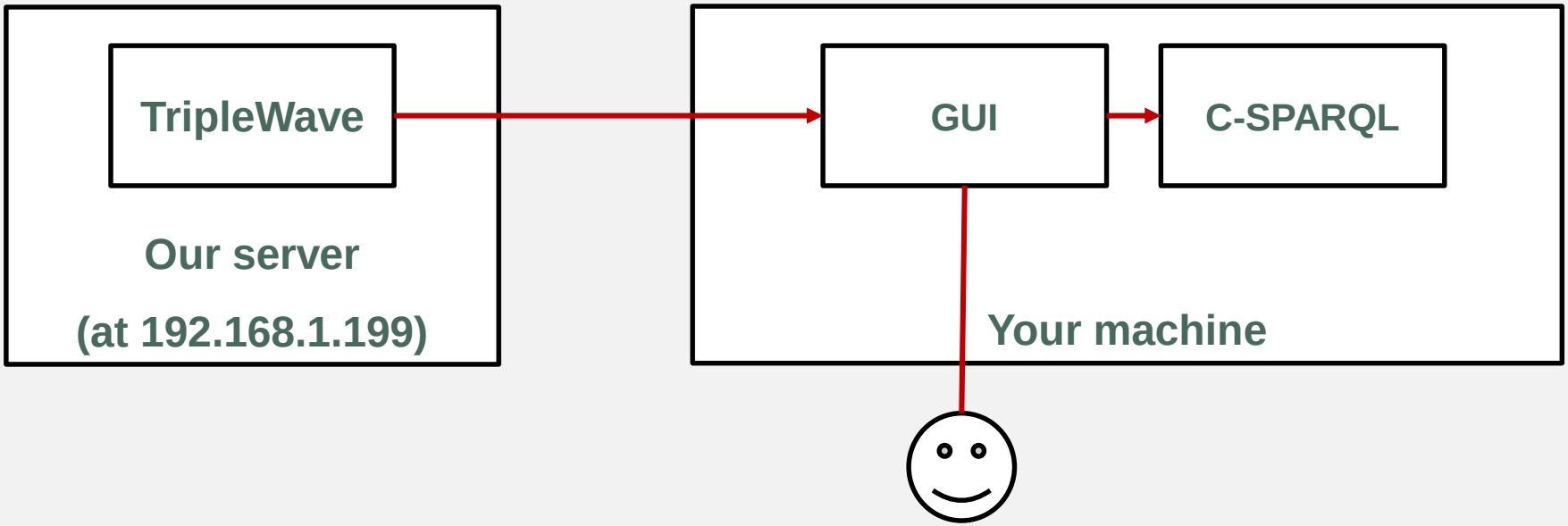
Hands-on session

Daniele Dell'Aglio

dellaglio@ifi.uzh.ch http://dellaglio.org @dandellaglio

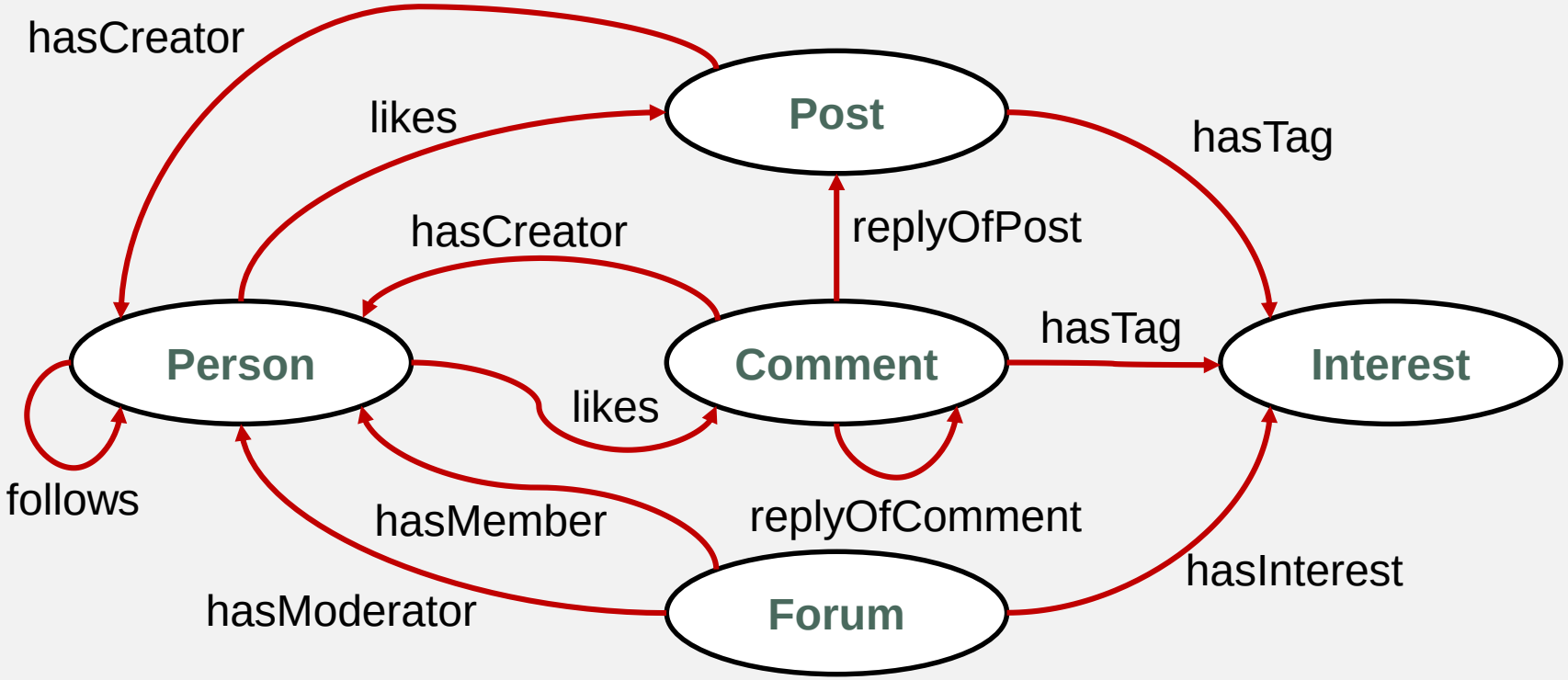
It's time to process some streams!

- Let's write together your (first?) continuous query
- We will work in the following setting

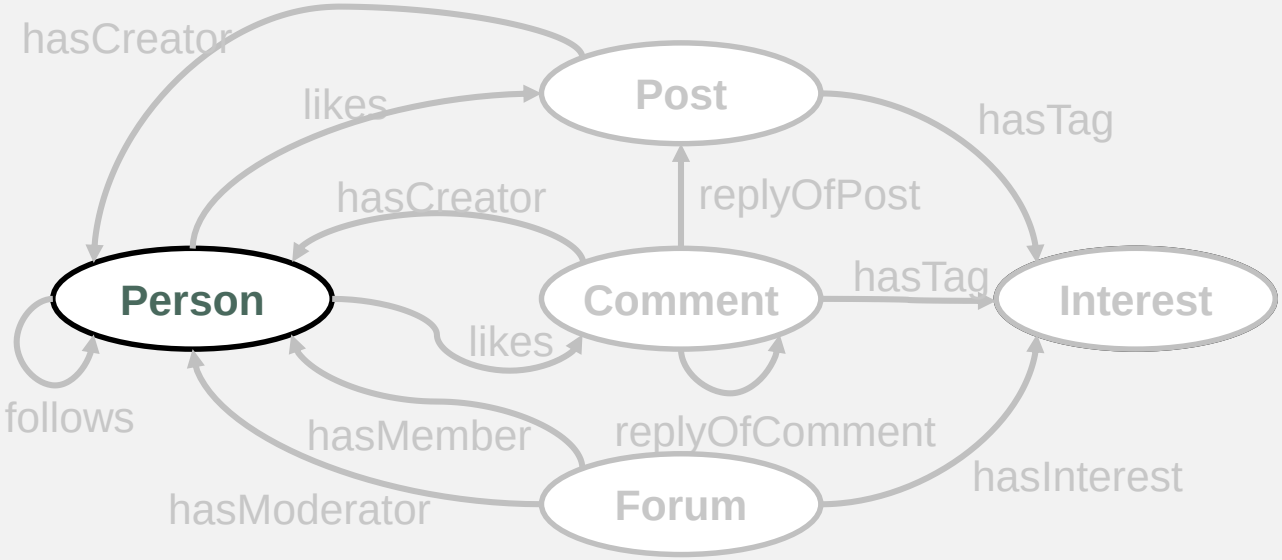


What is TripleWave streaming?

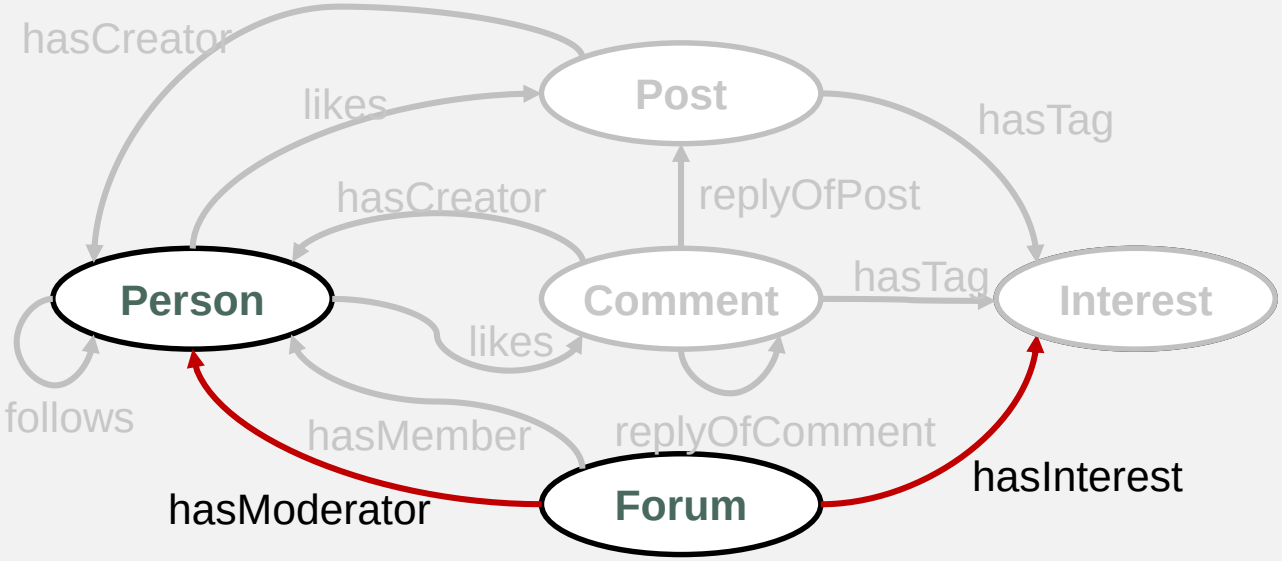
- The stream we consider in this session is a syntetic stream on social networks
  - We used the LDBC social network benchmark generator:
  - http://ldbcouncil.org/developer/snb
  - We converted it in an RDF stream (2GBs of data)
- Social network data



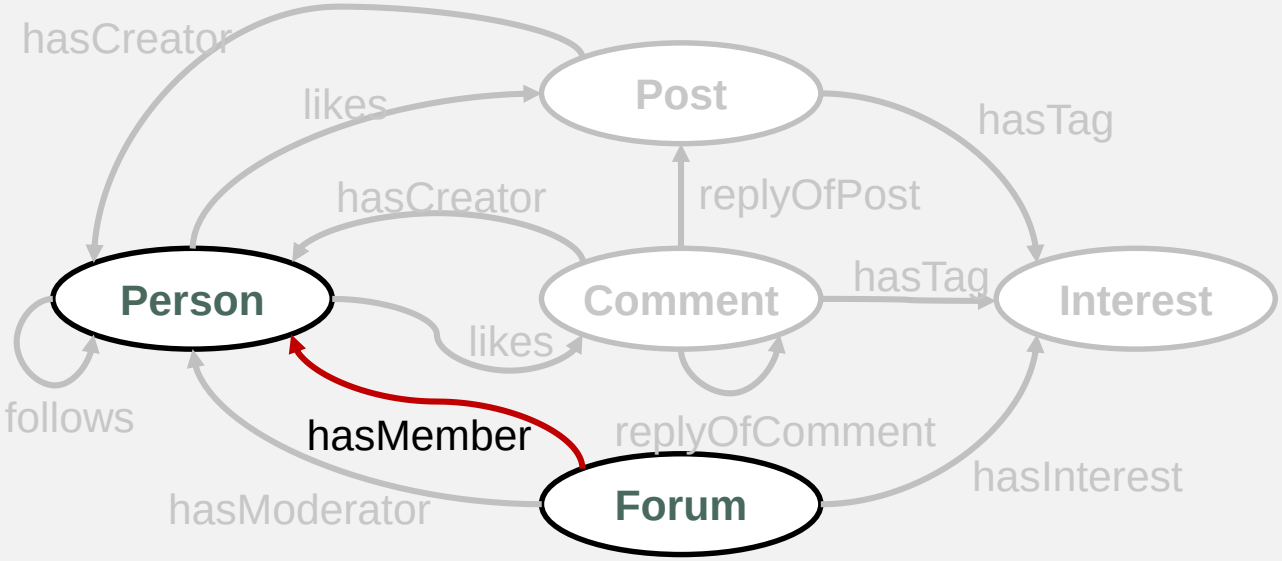
- The stream brings eight different kinds of event
  - Each of them brings a new update in the social network
- The possible events are:
  - AddPerson



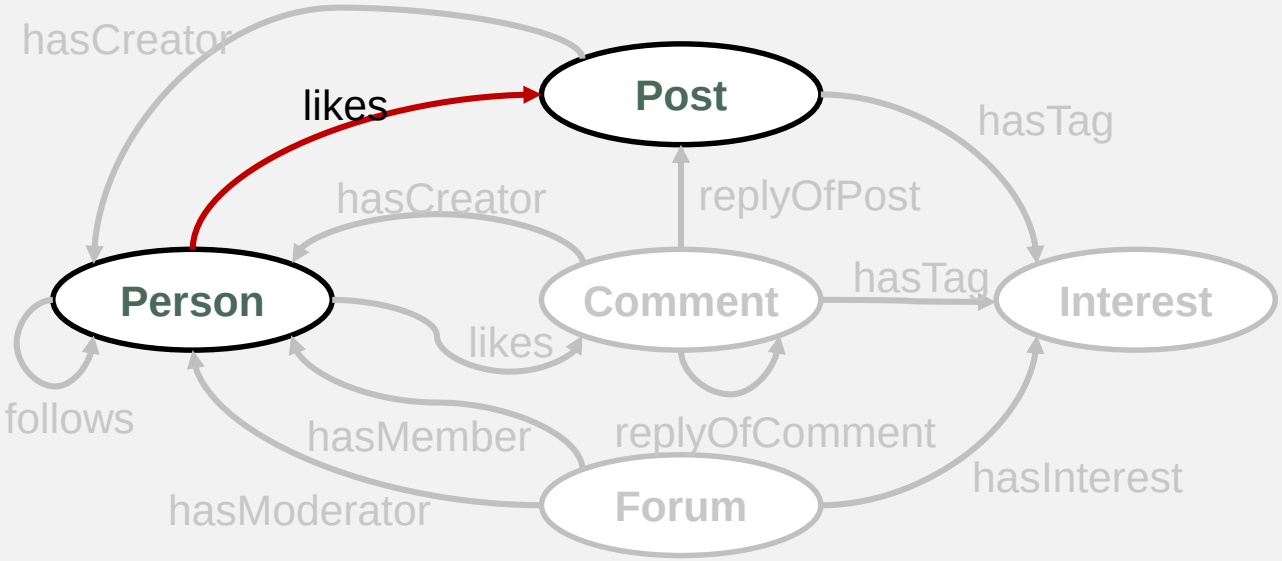
- The stream brings eight different kinds of event
  - Each of them brings a new update in the social network
- The possible events are:
  - AddPerson
  - AddForum



- The stream brings eight different kinds of event
  - Each of them brings a new update in the social network
- The possible events are:
  - AddPerson
  - AddForum
  - AddMembership

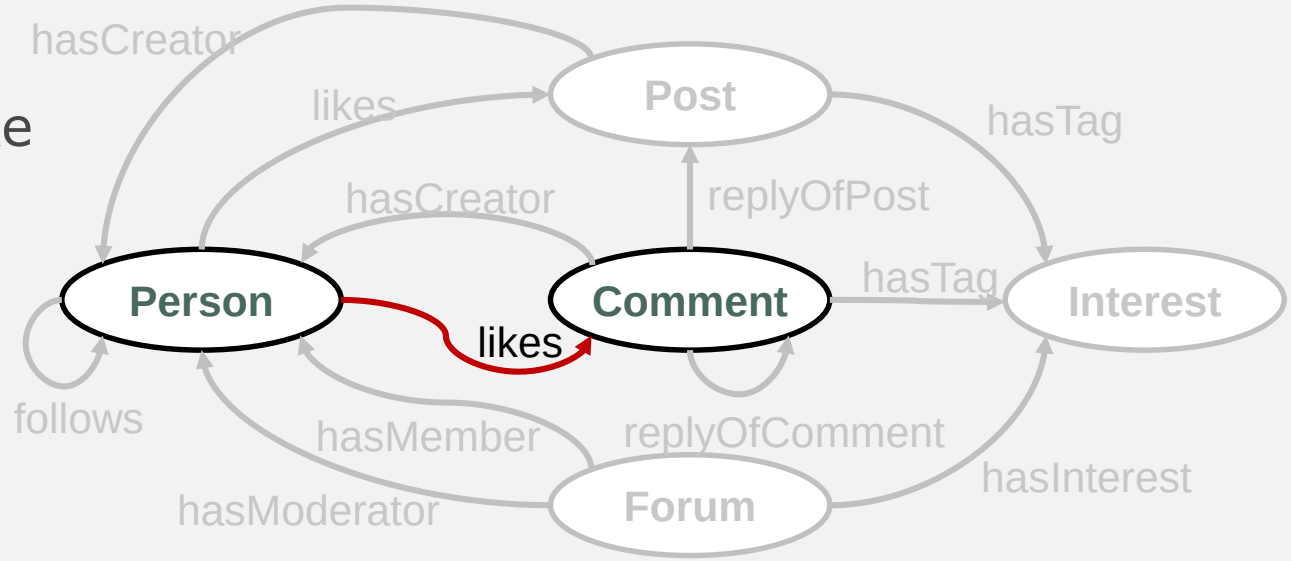


- The stream brings eight different kinds of event
  - Each of them brings a new update in the social network
- The possible events are:
  - AddPerson
  - AddForum
  - AddMembership
  - AddPostLike



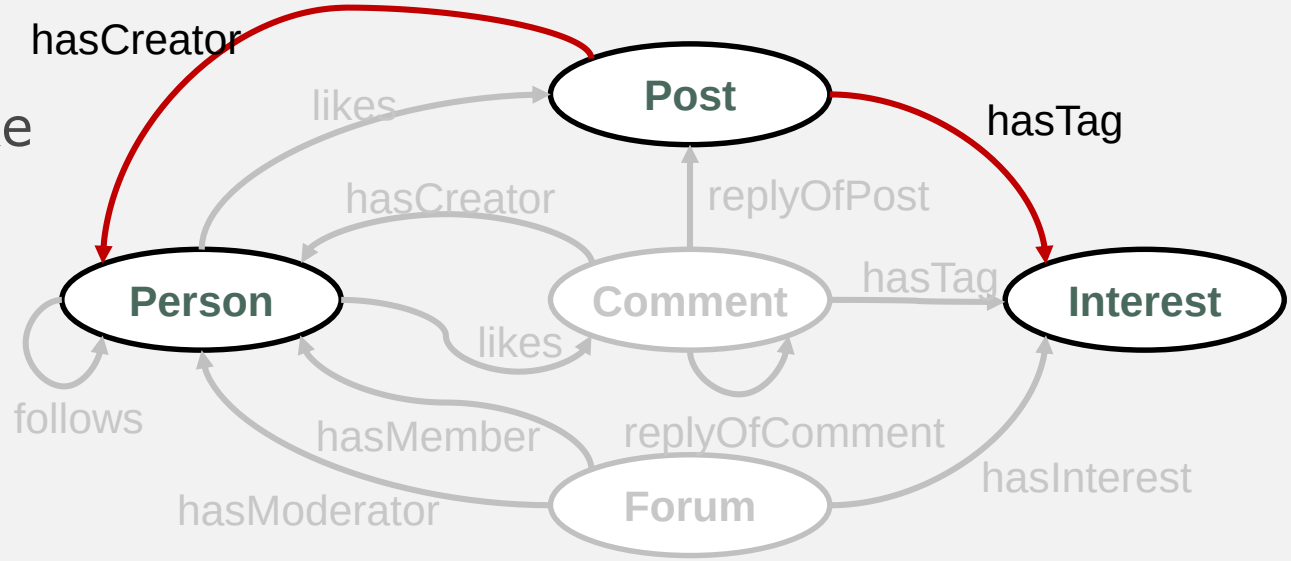
The stream

- The stream brings eight different kinds of event
  - Each of them brings a new update in the social network
- The possible events are:
  - AddPerson
  - AddForum
  - AddMembership
  - AddPostLike
  - AddCommentLike

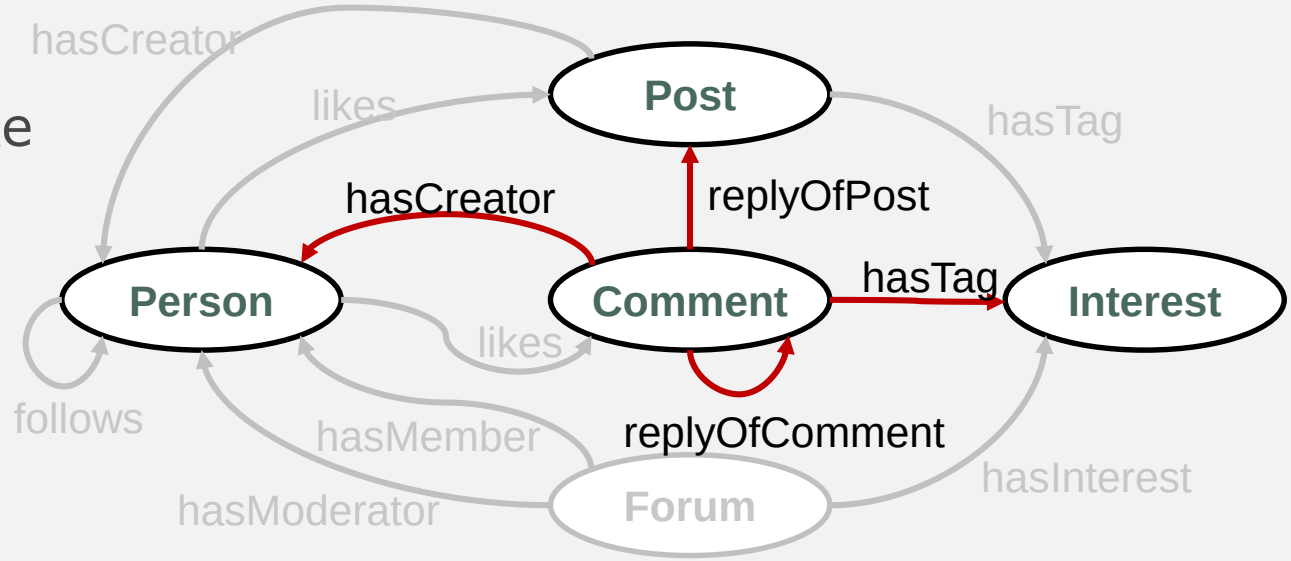


The stream

- The stream brings eight different kinds of event
  - Each of them brings a new update in the social network
- The possible events are:
  - AddPerson
  - AddForum
  - AddMembership
  - AddPostLike
  - AddCommentLike
  - AddPost

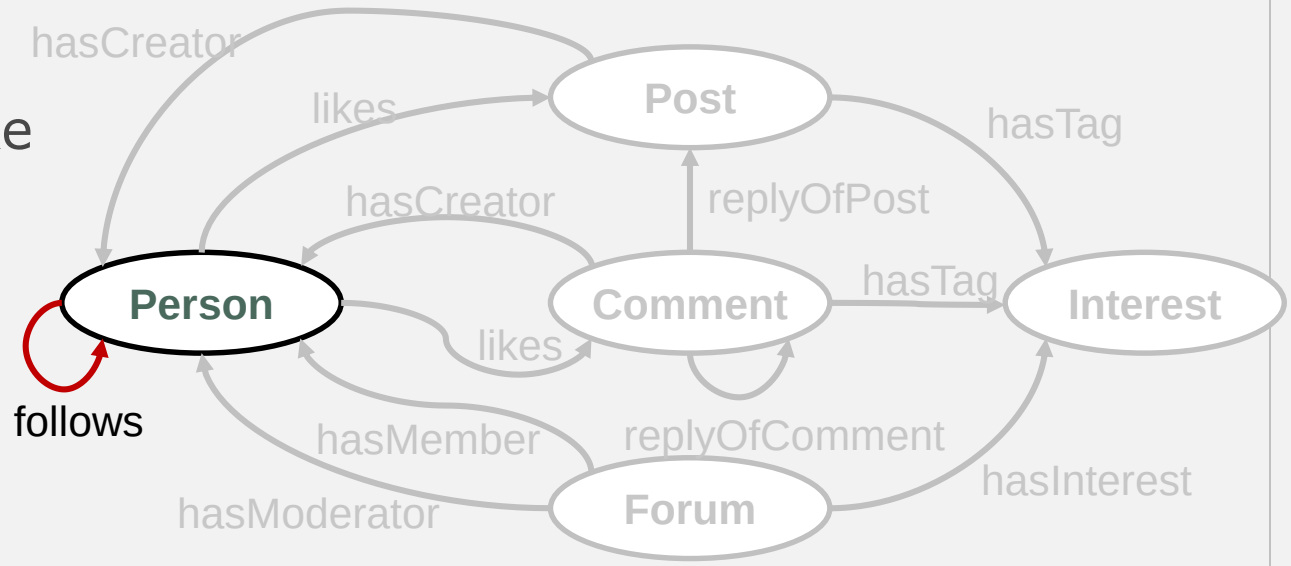


- The stream brings eight different kinds of event
  - Each of them brings a new update in the social network
- The possible events are:
  - AddPerson
  - AddForum
  - AddMembership
  - AddPostLike
  - AddCommentLike
  - AddPost
  - AddComment



The stream

- The stream brings eight different kinds of event
  - Each of them brings a new update in the social network
- The possible events are:
  - AddPerson
  - AddForum
  - AddMembership
  - AddPostLike
  - AddCommentLike
  - AddPost
  - AddComment
  - AddFriendship



Setting up the environment

- Your machine should have:
  - Java (7 or newer)
  - A browser
  - C-SPARQL + GUI
    - C-SPARQL
    - Jetty
    - Web App
- You shoul be connected to the RSP local network

  Name: rsp2016 Password: iswc2016wifi
- You can find C-SPARQL + GUI at:

  http://tinyurl.com/csparql

  http://192.168.1.198:8080/rsp2016/csparqlgui.zip

Running C-SPARQL and the GUI

- Unzip the zip fine in one folder
- Run C-SPARQL
  - On Linux/Mac: ./start_rsp_server.sh
  - On Win: Start_Rsp_Server.bat
- Run Jetty
  - On Linux/Mac: ./jetty.sh run
  - On Win:
    java -jar ..\jetty\start.jar --add-to-start=deploy,http,logging
- Open a browser and go to:

  http://localhost:8080/triplewave-wgui-socialgraph

- Now please do not press any button and wait for instructions ☺

Who are the people appearing in the stream in the last 30 seconds?

Query 1

- We want to select all the person that appear on the stream in the last 30 seconds and update the result every 30 seconds.
- In order to solve the task, we need
  1. Feed C-SPARQL with the input stream
  2. Register a C-SPARQL query
  3. Register an observer to get the results

Query 1: Feed C-SPARQL with the input stream

- Press the connect to stream button
- If everything worked fine, now you can see the stream in your browser

Query 1: Register a C-SPARQL query

- Press the query 1 button
- The following query is shown in the query text area:

```
SELECT DISTINCT ?p
FROM STREAM <http://.../sgraph> [RANGE 30s STEP 30s]
WHERE {?p a ldbc:Person }
```

- Submit the query to C-SPARQL
- Oh no! An error! What is the problem?
- Let's fix the query:

```
REGISTER QUERY query1 AS
SELECT DISTINCT ?p
FROM STREAM <http://.../sgraph> [RANGE 30s STEP 30s]
WHERE {?p a ldbc:Person }
```

- Submit the query to C-SPARQL

Who are the trending users in the last minute, updating the result every 10 seconds?

Query 2

- A trending user is a user that is increasing the number of followers in the last time period
- In order to solve the task, we need
  1. Feed C-SPARQL with the input stream
     - already done in the previous query!
  2. Register a C-SPARQL query
  3. Register an observer to get the results

Query 2: Register a C-SPARQL query

- Press the query 2 button

```
REGISTER QUERY query2 AS
SELECT DISTINCT ?p (COUNT(*) AS ?f)
FROM STREAM <http://.../sgraph>
WHERE { ?a ldbc:follows ?p}
GROUP BY ?p ORDER BY DESC(?f)  LIMIT 5
```

- Submit the query to C-SPARQL
- Let's fix the query:

```
REGISTER QUERY query2 AS
SELECT DISTINCT ?p (COUNT(*) AS ?f)
FROM STREAM <http://.../sgraph> [RANGE 60s STEP 10s]
WHERE { ?a ldbc:follows ?p}
GROUP BY ?p ORDER BY DESC(?f)  LIMIT 5
```

- Submit the query to C-SPARQL

Who are the trending topics in the last minute, updating the result every 10 seconds?

Query 3

- A trending user is a user that is increasing the number of followers in the last time period
- In order to solve the task, we need
  1. Feed C-SPARQL with the input stream
     - already done in the previous query!
  2. Register a C-SPARQL query
  3. Register an observer to get the results

Query 3: Register a C-SPARQL query

- Press the query 3 button

```
REGISTER QUERY query3 AS
SELECT ?t (COUNT(*) as ?total)
FROM STREAM <http://.../sgraph> [RANGE 60s STEP 10s]
WHERE { { ?c ldbc:hasTag ?t } UNION { ?f ldbc:hasInterest ?t}}
ORDER BY DESC(?total) LIMIT 5
```

- Submit the query to C-SPARQL
- Let's fix the query:

```
REGISTER QUERY query3 AS
SELECT ?t (COUNT(*) as ?total)
FROM STREAM <http://.../sgraph> [RANGE 60s STEP 10s]
WHERE { { ?c ldbc:hasTag ?t } UNION { ?f ldbc:hasInterest ?t}}
GROUP BY ?t ORDER BY DESC(?total) LIMIT 5
```

- Submit the query to C-SPARQL

Tutorial on RDF Stream Processing 2016

M.I. Ali, J-P Calbimonte, D. Dell'Aglio, E. Della Valle, and A. Mauri

http://streamreasoning.org/events/rsp2016

ISWC 2016 Kobe, Japan The 15th International Semantic Web Conference

Insight

Hes·so VALAIS WALLIS

Universität Zürich[UZH]

POLITECNICO MILANO 1863

Hands-on session

Daniele Dell'Aglio

dellaglio@ifi.uzh.ch http://dellaglio.org @dandellaglio