SemWebbers, LODers: What PubSubHubbub can do for you

Alexandre Passant
DERI, NUI Galway
A *real-time* Web

- **Information is no longer static**
  - The Social Web as an information stream (Twitter, 4sq, ...)
  - New trends in ubiquitous computing
  - Sensor data and the IoT
    - Even the @towerbridge is Tweeting!

- **A new area for Citizen Sensing**
  - Earthquake detection (WWW2010 paper)
  - Emergency management, reporting and monitoring
    (Mumbai attacks on Flickr, Twitter, etc.)
  - Opinion / trends mining (Box office prediction by HP labs)
Semantics and the real-time Web

- Combining social stream, sensors and social data
  - What’s happening right now, sport-wise, 25km around here? – Geonames, Twitter, Dbpedia, etc.
  - Who in my social network, whatever the website we’re connected through, is currently in my hometown – FOAF, Geonames, etc.
  - Any humidity issues in our server rooms in the US – Sensors, Geonames, Legacy Enterprise Data, etc.

- New architectures are required
  - Enabling pro-active notification (PuSH) based on triggers
  - Defining semantic triggers? Yes, with SPARQL!
Pull vs Push

- **Pull**
  - Constant queries to various sources to get what’s new
  - Useless HTTP calls (API/RSS), risks of being banned (TOS)

- **Push**
  - Websites let me know when they have something relevant
  - **Wait. Receive. Consume**
PubSubHubbub (PuSH) at a glance

- **Google’s Push approach**
  - [http://code.google.com/p/pubsubhubbub](http://code.google.com/p/pubsubhubbub)
  - Using Atom / RSS
    - `link rel="hub"` header for identifying Hub from a feed
  - Simple registration / notification approach
    - API available in various languages
  - Broadcasting data through public hubs
    - Use Google’s one, Superfeedr, or host your own
PubSubHubbub, SemWeb and LOD?

- Registrations and notifications about structured data
  - Broadcasting SPARQL-defined events using PuSH

- sparqlPuSH
  - Pro-active notifications of changes in RDF stores
  - Monitoring of real-time changes using SPARQL and PuSH

- SMOB
  - Distributed microblogging
  - Synchronising hubs using SPARQL 1.1 Update and PuSH

- Twarql
  - Twitter stream analysis
sparqlPuSH at a glance

- **sparqlPuSH**
  - Combining SPARQL, SPARQL Update and PubSubHubbub for proactive notifications of changes in RDF stores
  - An interface that can be plugged on the top of any RDF store - [http://code.google.com/p/sparqlpush/](http://code.google.com/p/sparqlpush/)

- **Based on**
  - **SPARQL** to register feeds based on query patterns
  - **SPARQL 1.1 Update** for triggers registration
  - **Atom and RSS** to get feeds of related changes
  - **PubSubHubbub** for broadcasting changes
A 2-steps approach

- A two-steps approach
  - Query registration
  - Change notification

- A sparqlPuSH endpoint
  - Plugged on top of any RDF store (implementation-agnostic)
  - Registration can be done remotely, through an HTTP request sent to the sparqlPuSH endpoint
  - Notification is triggered as soon as relevant data appears in the store, loaded with SPARQL Update through sparqlPuSH
  - Clients must understand the rel="hub" link in the feed header, and interpret notification from PuSH hubs
Query registration

1. Client sends a query to the PuSH Hub.
2. The PuSH Hub forwards the query to the RDF Store.
3. The RDF Store processes the query.
4. The PuSH Hub receives the results from the RDF Store.
5. The PuSH Hub notifies the client of the results.
Example of query registration

- Identifying changes on a particular object
  - Be notified when something happens to ex:FooBar
  - Using the Changeset vocabulary by Talis

```
PREFIX cs: <http://purl.org/vocab/changeset/schema#>

WHERE {
  ?uri a cs:ChangeSet ;
  cs:creatorName ?author ;
  cs:changeReason ?label ;
  cs:createdDate ?date ;
  cs:subjectOfChange <http://example.org/FooBar> .
}
ORDER BY ASC(?date)
```
Query registration

<title type="text">sparqlPuSH</title>
<link rel="self" href="http://rdfs.org/tmp/server/feed/4bcb13c699e89" title="sparqlPuSH" type="application/atom+xml"/>
<link rel="hub" href="http://pubsubhubub.appspot.com"/>

<entry>
  <id>http://twitter.com/cshirky/status/12399443075</id>
  <title type="text">Update - 2010-04-18T15:19:27+01:00</title>
  <link rel="alternate" href="http://twitter.com/cshirky/status/12399443075" title="Update - 2010-04-18T15:19:27+01:00" type="text/html"/>
  <content type="text">The big gulf between TwpParty &amp; others-Repubs &amp; gen'l public are social: gay marriage, abortion, immigration, cl</content>
  <published>2010-04-18T15:19:27+01:00</published>
  <updated>2010-04-18T15:19:27+01:00</updated>
  <author>
    <name>http://twitter.com/cshirky</name>
  </author>
</entry>

<entry>
  <id>http://twitter.com/jstan/status/12399270077</id>
  <title type="text">Update - 2010-04-18T15:17:23+01:00</title>
  <link rel="alternate" href="http://twitter.com/jstan/status/12399270077" title="Update - 2010-04-18T15:17:23+01:00" type="text/html"/>
  <content type="text">RT @Enderscience: Call or papers: Technology, Operations and Strategy in Entrepreneurship: A special issue of Inte... ht</content>
  <published>2010-04-18T15:17:23+01:00</published>
  <updated>2010-04-18T15:17:23+01:00</updated>
  <author>
    <name>http://twitter.com/jstan</name>
  </author>
</entry>

<entry>
  <id>http://twitter.com/webr3/status/12398025115</id>
  <title type="text">Update - 2010-04-18T14:50:02+01:00</title>
  <link rel="alternate" href="http://twitter.com/webr3/status/12398025115" title="Update - 2010-04-18T14:50:02+01:00" type="text/html"/>
  <content type="text">RT @jahendler: RT @cherimossey: WOW. RT @gabbycat: Amazing picture of lightning at Iceland volcano: http://bit.ly/bH3nhN</content>
  <published>2010-04-18T14:50:02+01:00</published>
  <updated>2010-04-18T14:50:02+01:00</updated>
  <author>
    <name>http://twitter.com/webr3</name>
  </author>
</entry>
Conventions in query registration

- Using conventions to get a well-formatted Atom / RSS feed
  - Easier to read in standard aggregators

- Mandatory elements
  - ?uri - their URI of the element(s) to be retrieved
  - ?date - their creation / modification date
  - Can be used to retrieve named graphs if content itself is not dated

- Optional elements
  - ?label - their label
  - ?author - their author
Browsing available feeds

- The sparqlPuSH UI
  - Lists available feeds, including timestamp of last update
  - Ability to create feeds from the interface

sparqlPuSH home

Type-in a SPARQL query in the following form and the results will be provided as a RSS feed, including a pubsubhubbub hub address for broadcasting updates. To make the RSS feed more friendly, use the following conventions regarding the items you query:

- ?uri: their URI
- ?label: their label
- 7date: their creation / modification date
- ?author: their author

As an example, the default query on the right will provide you with an RSS feeds describing the last SIoC items that have been loaded in the RDF store. In addition, you can check the list of available feeds that you can subscribe to.

Available feeds

Feed: http://localhost/~alex/pshh/feeds/46ba06f4dcb51

```
SELECT ?uri ?author ?title
WHERE {
  ?uri a sioc:Post;
  sioc:has_creator ?author;
  dcterms:creator ?author;
  dc:created ?date .
}
```

This is the sparqlPuSH interface for http://localhost/~alex/pshh

Feed: http://localhost/~alex/pshh/feeds/46ba4d0be79850

```
SELECT ?uri ?author ?title
WHERE {
  ?uri a sioc:MicroblogPost;
  sioc:has_creator ?author;
  dcterms:topic dbpedia:Healthcare_reform;
  sioc:has_creator ?author;
  dcterms:creator ?author;
  dc:created ?date .
}
```

This is the sparqlPuSH interface for http://localhost/~alex/pshh
Notification

1. RDF Data
2. RDF Store
3. PuSH Hub
4. Client
**SPARQL Update support**
- HTTP-Posting data to the sparqlPuSH endpoint
  - Then loaded in the underlying RDF store
  - Allows *real-time* identification (as opposed to cron-job)

**Identifying relevant changes**
- Applying all registered queries to the updated dataset

**Broadcasting changes**
- Using PubSubHubbub! (Scalability and RT delivery)
Implementation

- **Source code (PHP)**

- **Server**
  - Connection to any SPARQL endpoint
  - Additional connector for ARC2 using the ARC2 API
  - Generating RSS or Atom feeds

- **Demo client**
  - Registering / unregistering queries to remote sparqlPuSH interfaces
  - Receiving updates from registered feeds
SMOB

- **Distributed and Semantic Microblogging**
  - **Setup** your own hub, **own** your data, **broadcast** it
  - Fully SemWeb based (RDF backend, RDFa, LOD, etc.)

- **SMOB and PuSH?**
  - Broadcasting SPARQL 1.1 Update queries
  - Instant notification for your followers
  - Embed rich-data in the query, encapsulated in a RSS feed
  - Ability to broadcast content deletion
SPARQL 1.1 Update + RSS + PuSH

- **SPARQL 1.1 Update**
  - Enabling write access to RDF Store

- **SMOB and SPARQL 1.1 Update?**
  - Used to send messages to followers
  - Query pattern encapsulated in a RSS feed
  - RSS feed sent to followers using PuSH
  - **Efficiency. Scalability. Rich content.**
Twarql

- Twitter feeds through SPARQL
  - Entity extraction from Twitter feeds
  - Representation as Linked Data (SMOB Stack)
  - SPARQL queries + PuSH for notifications

- Code and demos
Acknowledgements

- Google Research Awards
- Julie Anaya (SMOB + PuSH)
- Pablo Mendes / Pavan Kapanipathi (Twarql)

Contact

- alexandre.passant@deri.org
- http://apassant.net
- @terraces