LOD Laundromat

Why the Semantic Web Needs Centralization (Even If We Don't Like It)

Wouter Beek Laurens Rietveld Stefan Schlobach
Frank van Harmelen

According to Linked Open Data principles, data publishers can make their data available online in a machine-processable format. Other publishers can add their own data by linking to existing entities, allowing "everyone to say anything about anything"[1]. Data consumers are then able to process data from different sources, allowing for novel reuse. Since the different data sources are all freely available and linked together, Web agents are able to traverse the interconnected LOD Cloud and perform intelligent tasks[2].

Alas, as is often the case with principles, the factual situation is quite different:

- Today's Semantic Web is generally not machine-processable,
- It cannot be traversed by Web agents or applications,
- Information from different sources cannot be readily accessed, and
- Even though everyone can say anything about anything, very few people are actually heard.

We solve the problems of today's Semantic Web by doing away with some of its fundamental assumptions:

distribution: Instead of a fully distributed approach for both publishing and consuming Linked Open Data, we centralize the gathering, cleaning, querying and (re)publishing of Linked Open Data.

re-use: Contrary to what is advocated by the W3C standards, we do not use the Semantic Web query language SPARQL to disseminate the data.

navigation: Finally, and perhaps most controversially, we drop the dereferenceability requirement for IRIs.

We show that removing these pillars still leaves enough intact to still be considered a "Semantic Web". We also show that what remains is much more usable and is in fact closer

[1] `http://www.w3.org/TR/2002/WD-rdf-concepts-20020829/#xtocid48014`
[2] James A. Hendler: Agents and the Semantic Web. IEEE Intelligent Systems 16(2): 30-37 (2001)

to the original vision of a generally useful Web of Data. This radical departure from Linked Open Data principles is not a mere proposal; it has already been built. Thousands of Semantic Web practitioners are using it on a regular basis. Its name is *LOD Laundromat* (`http://lodlaundromat.org/`).

1 Data cleaning

One problem of today's Semantic Web is that it cannot be easily *read* by computers. In stark contrast with its fundamental motivation, the machine-readability of Linked Open Data is a much bigger obstacle than people realize. For instance, less than 10% of the widely popular and highly curated Freebase dataset (now managed by Wikidata) can be read by a standards-compliant parser, such as Raptor. This percentage is even lower for many of the less commonly used datasets.

The WWW suffers from a similar problem; most HTML pages are not fully conformant either. However, there are significant differences. Firstly, a tremendous development effort has gone into making Web browsers robust against incorrect HTML usage. Another crucial difference is that a traditional Web document is intended for a *human* reader. Even if the layout of a Web page is buggy, a human agent may still be able to process at least some of the page's content. On the Semantic Web however, agents have below-human intelligence. As a result, even a small mistake in syntax breaks their navigation and processing capabilities.

This problem has been widely recognized and various solutions have been attempted. The Semantic Web community has formulated a wide collection of guidelines and best practice documents[3]. There has also been a strong focus on education through courses, handbooks, summer schools and tutorials. The fundamental problem of these approaches is that they all target the human data publisher and consumer. As a result, the success of these approaches crucially depends on the willingness and capability of a large number of humans to *do the right thing*. As we know from other areas of society, there is oftentimes a discrepancy between what most people agree is the right thing to do and what most people actually do, and we don't expect the Semantic Web to be an exception.

Instead, the LOD Laundromat takes on the burden of cleaning HTTP headers, encodings, archive formats, RDF syntax errors, unrecognized literal values and more[4]. This shifts the burden of standards compliance from the (many) data publishers, who – as practice shows – cannot be relied upon, to a single centralized service that can be relied upon.

[3] e.g. `http://www.w3.org/2001/sw/BestPractices/`

[4] see for details: Wouter Beek, Laurens Rietveld, Hamid R. Bazoobandi, Jan Wielemaker, Stefan Schlobach: LOD Laundromat: A Uniform Way of Publishing Other People's Dirty Data. Semantic Web Conference (1) 2014: 213-228

2 Data (re)publishing

In addition to cleaning the data, LOD Laundromat also allows the data to be downloaded, thereby turning LOD Laundromat into a data republishing platform. This is different from previous centralization efforts in Linked Open Data such as Swoogle[5] and Sindice (now defunct) that depend on the availability of the original data. While the LOD Laundromat data collection cannot claim completeness, it is very easy to add new data, either by entering a link to an online document or by uploading an offline document through a Dropbox plugin. Fifteen minutes later a clean version of the data can be downloaded and a query endpoint is added that can be freely used by all.

By carrying the burden of hosting the data, LOD Laundromat is lowering the entry level for Linked Open Data publishing. Instead of countless data providers having to host Web servers (which turn out to be highly unreliable[6], this cost is now offloaded to a reliable centralized service.

3 Consuming data

Besides lowering the costs for data publishers (as described in the previous two sections), LOD Laundromat also lowers the costs for data consumers. The centralization approach of the LOD Laundromat proves to be a big advantage: the RESTful HTTP access, the compression format and the serialization grammar are the same for all 650,000+ data documents. We use a very simple format that is a subset of canonical N-Quads. This allows all RDF data (including named graphs) to be expressed while at the same time facilitating easy processing: all and only newline characters denote ends of statements, statements are sorted lexicographically and do not contain duplicates. This implies that a simple line count gives the number of statements of a data document, and that a straightforward regular expression suffices to parse the data. Since, in addition, no prefix declarations or other header elements are used and blank nodes are globally unique, data documents can be freely split on newlines and/or concatenated, always resulting in a standards-compliant result. For a data consumer, a single simple syntax format without syntax errors makes it both easier and more efficient to process Linked Data.

[5] http://swoogle.umbc.edu/

[6] Carlos Buil Aranda, Aidan Hogan, Jürgen Umbrich, Pierre-Yves Vandenbussche: SPARQL Web-Querying Infrastructure: Ready for Action? International Semantic Web Conference (2) 2013: 277-293

4 Querying

In addition to the rather crude datadump approach, there are two other popular approaches for querying Linked Open Data. One of them is dereferencing, i.e., performing an HTTP GET operation on an IRI in order to retrieve information about that IRI. It has not been standardized which statements are part of a dereference result. While it is common practice to return triples that have the dereferenced IRI in the subject or, optionally, object position, this practice does not guarantee that a result set is complete. It is also not possible for a Web agent to traverse a dataset by using the dereferencing approach. Since blank nodes are very common (7% of all terms) but are not dereferenceable, only relatively small subcomponents of a graph can be traversed. Also, by relating the data that is requested about an IRI to the authority of that IRI, dereferencing only returns information about a resource that is asserted by the owner of that resource. This means that while everyone is theoretically able to make assertions about any IRI, only the assertions of the owner of the IRI's authority can be easily retrieved. As a result, dereferenceability does not support the findability of multiple opinions or perspectives on the same topic, and does not contribute to the democratic potential of the Semantic Web.

The other approach towards querying Linked Data is SPARQL, the standardized Semantic Web query language. Since SPARQL locates the computationally expensive task of query evaluation at the server side exclusively, most SPARQL endpoints enforce restrictions to prevent the endpoint from collapsing under multiple simultaneous requests. The most often enforced stricture is a limit on the size of the result set, implying that in practice SPARQL results are incomplete. Although rudimentary in comparison to SPARQL, data dumps at least ensure completeness.

The SPARQL observatory SPARQLes[7] reports a surprisingly low number of (known) SPARQL endpoints: 545. Moreover, SPARQLes shows that only 181 SPARQL endpoints have high ($\leq$99%) availability. In addition, the growth of the number of SPARQL endpoints has been linear over the last 5 years. With at least millions of data documents out there but only hundreds of SPARQL endpoints with reasonable availability, existing deployment techniques are simply unable to close the gap: data is growing faster than SPARQL deployment uptake. This is also bad news for the democratic potential of the Semantic Web: the number of voices that are heard on the Semantic Web is surprisingly low when compared to the WWW which contains millions of sites.

An explanation for the slow adoption of SPARQL can be found in economics. As the SPARQL paradigm of querying puts the computational burden of answering a query on the server side, the cost of publishing Linked Data is proportional to the usefulness of the data to others. This negatively incentivizes publishing large amounts of valuable data. At the same time, the client's cost of posing a query is zero. A healthy market exhibits *allocative efficiency*, i.e., the price a consumer pays should equal the marginal cost of producing the consumed service. Since a client pays nothing and the marginal cost of production is relatively high, the SPARQL paradigm is

[7] `http://sparqles.ai.wu.ac.at/`

inherently far removed from allocative efficiency.

LOD Laundromat is able to make its entire Linked Open Data collection available for others to utilize since it does not use the SPARQL paradigm of server-side query evaluation. Instead it uses the underlying data storage format Header Dictionary Triples[8] (HDT) and the Linked Data Fragments[9] (LDF) API. This shifts the computational burden of query evaluation from the server onto the clients, resulting in a significantly reduced hardware footprint. Since the only hardware resource that is consumed by the HDT+LDF approach is disk space and disk space is relatively cheap, the LOD Laundromat is not unduly penalized for exposing large volumes of valuable data.

4.1 Web-scale querying

So far we have only been concerned with a client querying a single server. Ideally, we want to ask a question to and receive an answer from the whole Semantic Web. The current SPARQL-based deployment is discouraging federated, let alone Web-scale querying. When querying multiple endpoints the least capable server decides which language constructs can be used, and the slowest SPARQL endpoint decides the speed at which a federated query is serviced. More fundamentally, a federated query in SPARQL requires each queried endpoint to be explicitly qualified. This makes Web-wide queries impossible.

While the LOD Laundromat Web Services can be used to upload, download and query Linked Data on a per-document basis, large-scale cross-document processing can be further simplified. For this we have created the Federated Resource Architecture for Networked Knowledge or *Frank*. Its purpose is to allow large-scale Linked Data consumption from the command line, using one or two lines of code.

Frank is able to retrieve data documents, filtered by metadata properties, namespaces and/or IRIs that appear in those documents. In addition, it allows single triple patterns to be evaluated across all 650,000+ data documents. *Frank* provides many benefits over the traditional approach of IRI dereferencing. Firstly, it always returns all matches for the given single triple pattern (as per SPARQL 1.1 triple matching). This includes the statements that are commonly included in a dereference result set. Secondly, it retrieves authoritative statements about a resource as well as non-authoritative ones, allowing alternative views about a resource to be findable as well. For each returned statement the document can be included, so that the authority can be verified. Thirdly, since LOD Laundromat makes blank nodes globally unique (in line with the RDF 1.1 specification), *Frank* is also able to query for blank nodes and traverse the LOD Laundromat data collection in a way in which the original LOD Cloud cannot be traversed. Finally, and most importantly, LOD Laundromat does not set any limits to the size of the retrieved result

[8] `http://www.rdfhdt.org/`
[9] `http://linkeddatafragments.org/`

set. Results obtained through the Web Services and *Frank* are guaranteed to be complete with respect to the LOD Laundromat data collection.

5 The future of Semantic Web deployments

The LOD Laundromat is the first Linked Open Data API that provides uniform access to a large and ever increasing subcollection of the LOD Cloud. The LOD Laundromat has been hosting over 650,000 query endpoints as of February 2015. It has been used by thousands of unique users that have posed tens of millions of queries and have downloaded millions of documents. These numbers show that the LOD Laundromat approach is scalable and robust and that its Web Services and APIs are easy (enough) for others to understand and use.

As we have seen, dereferenceability cannot be used to traverse or otherwise process large portions of the LOD Cloud in a reliable way. SPARQL endpoints offer the ability to express very powerful queries but will not work for Web-wide querying. SPARQL is still very important, but only in certain use cases and with a limited user base in mind. Other query paradigms will have to be attempted, HDT and LDF are currently exploring this space of potential Web query languages.

A new development we observed is the building of custom API's on top of a SPARQL endpoint. A custom API ensures that only a small number of SPARQL patterns can be queried for. This significantly simplifies endpoint optimization. We do *not* think this is a good development. The deficiencies of the existing deployment paradigm should not result in the altogether abandonment of the idea of a machine-processable Web. Doing away with dereferenceability and SPARQL as the only or even main ways of disseminating Linked Open Data may be necessary to save the machine-processable Web.

Even though centralization has a negative ring to it, LOD Laundromat is an inclusive environment: all data is accepted and is treated equally by our indexes and interfaces. Indeed, we saw that it is much easier to find alternative or non-authoritative assertions about an entity using the centralized LOD Laundromat. Maybe the Semantic Web requires a certain level of centralization for it to reach its full democratic potential.