LOD Laundromat offers a centralized solution for today's Semantic Web problems. This solution adheres more closely to the original vision of a Web of Data, providing uniform access to a large and ever-increasing subcollection of the LOD Cloud.

According to Linked Open Data (LOD) 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.” Data consumers are then able to process data from different sources, allowing for novel reuse. Because 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.

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

- Today’s Semantic Web generally isn’t machine-processable;
- It can’t be traversed by Web agents or applications;
- Information from different sources can’t 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 LOD, we centralize the gathering, cleaning, querying, and republishing of LOD.
- Reuse. Contrary to what’s advocated by W3C standards, we don’t use the Semantic Web query language SPARQL to disseminate the data.
- Navigation. Finally, and perhaps most controversially, we drop the dereferencibility requirement for internationalized resource identifiers (IRIs).

We show that removing these pillars still leaves enough intact to be considered a “Semantic Web.” We also show that what remains is much more usable and is in fact closer to the original vision of a genuinely useful Web of Data. This radical departure from LOD principles isn’t 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 (see Figure 1 and http://lodlaundromat.org/).

Data Cleaning

One problem of today’s Semantic Web is that it can’t be read easily by computers. In stark contrast with its fundamental motivation, the machine-readability of LOD is a much bigger obstacle than people realize. For instance, less than 10 percent 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 aren’t fully conformant, either. However, there are significant efforts underway. First, a tremendous development effort has gone into making Web browsers more 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 webpage is buggy, a
human agent might 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, with various solutions attempted. The Semantic Web community has formulated a wide collection of guidelines and best practice documents (see www.w3.org/2001/sw/BPPractices). There also has been a strong focus on education through courses, workshops, 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, often there’s 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. This shifts the burden of standards compliance from the (many) data publishers, who – as practice shows – can’t be relied upon, to a single centralized service that can be relied upon.

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 LOD, such as Swoogle [http://swoogle.umbc.edu] and Sindice [now defunct], that depend on the original data’s availability. While the LOD Laundromat data collection can’t claim completeness, it’s 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 freely can be used by all.

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

Figure 1. The Linked Open Data (LOD) Laundromat layer cake. The red and brown layers are for crawling the Web and extracting the standards-compliant subset of the LOD cloud. The yellow layer shows the database technologies that allow LOD Laundromat to serve tens of billions of statements. The green layer shows the APIs that case data consumption. All crawling and cleaning information is stored as metadata. This also includes the data's structural properties that are used in the database and data consumption layers. The blue layer provides a unified programmable interface over the entire LOD Laundromat ecosystem. This is the API that is used to scale Semantic Web research in our LOD Lab (http://lod.unic.edu).

Consuming Data

Besides lowering the costs for data publishers (as described previously), LOD Laundromat also lowers the costs for data consumers. The centralization approach of LOD Laundromat proves to be a big advantage: the RESTful HTTP access, the compression format, and the serialization grammar are the same for the more than 650,000 data documents. We use a simple format that’s a subset of canonical N-Triples. This allows all RDF data (including named graphs) to be expressed, while at the same time facilitating easy processing: all and only new-line characters denote ends of statements; statements are sorted lexicographically and don’t 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. Because, 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 simple single syntax format without syntax errors makes it both easier and more efficient to process Linked Data.

In addition to the rather crude data dump approach, there are two other popular approaches for querying LOD. One of them is dereferencing – that is, performing an HTTP GET operation on
Linked Data

Linked Data is able to make its entire LOD collection available for others to utilize.

The other approach towards querying Linked Data is SPARQL, the standardized Semantic Web query language. Because SPARQL locates the computationally expensive task of query evaluation at the server-side exclusively, most SPARQL endpoints enforce restrictions to prevent the end-point from collapsing under multiple simultaneous requests. The most often enforced restriction 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 SPARQL.es (http://sparql.es.w3.ac.uk) reports a den 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—that is, the price a consumer pays should equal the marginal cost of producing the consumed service. Because a client pays nothing and the marginal cost of production is relatively high, the SPARQL paradigm is inherently far removed from allocative efficiency.

LOD Laundromat is able to make its entire LOD collection available for others to utilize, because it doesn't use the SPARQL paradigm of server-side query evaluation. Instead, it uses the underlying data storage format Header Dictionary Triples (HDT; www.rdfhdt.org) and the Linked Data Fragments (LDF; http://linkeddatafragments.org) API. This shifts the computational burden of query evaluation from the server onto the clients, resulting in a significantly reduced hardware footprint. Because the only hardware resource that's consumed by the HDT/LDF approach is disk space and disk space is relatively cheap, LOD Laundromat isn't unduly penalized for exposing large volumes of valuable data.

Web-Scale Querying

So far we've 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 served. 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've 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 the more than 650,000 data documents. Frank provides many benefits over the traditional approach of IRI dereferencing. First, 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. Second, it retrieves authoritative
statements about a resource as well as
non-authoritative ones, allowing alter-
native views about a resource to be
findable as well. For each returned state-
ment the document can be included, so
that the authority can be verified. Third,
because 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
can’t be traversed. Finally, and most
importantly, LOD Laundromat doesn’t
limit the size of the retrieved result set.
Results obtained through the Web ser-
vices and Frank are guaranteed to be
complete with respect to the LOD Laun-
dromat data collection.

The LOD Laundromat is the first LOD
API that provides uniform access to
a large and ever-increasing subcollection
of the LOD cloud. As of February
2015, LOD Laundromat has been host-
ing more than 650,000 query endpoints.
It has been used by thousands of unique
users who have posed tens of millions
of queries and downloaded millions
of documents. These numbers show that
the LOD Laundromat approach is scal-
able and robust, in that its Web Services
and APIs are easy (enough) for others to
understand and use.

As we’ve seen, dereferenceability
can’t be used to traverse or otherwise
process large portions of the LOD
cloud in a reliable way. SPARQL en-
dpoints offer the ability to express very
powerful queries, but they won’t work
for Web-wide querying. SPARQL is
still important, but only in certain use
cases and with a limited user base in
mind. Other query paradigms should be
attempted; HDT and LDP are cur-
rently exploring this space of potential
Web query languages.

A new development we observed
is the building of custom APIs 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 don’t think this is a
good development. The deficiencies
of the existing deployment paradigms
shouldn’t result in the altogether aban-
donment of the idea of a machine-
processable Web. Doing away with
dereferenceability and SPARQL as the
only or even main ways of disseminat-
ing LOD might 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
are accepted and treated equally by
our indexes and interfaces. Indeed, we
saw that it’s much easier to find alter-
avtive or non-authoritative assertions
about an entity using the central-
ized LOD Laundromat. Perhaps the
Semantic Web requires a certain level
of centralization for it to reach its full
democratic potential.

References
1. Resource Description Framework (RDF): Con-
ccepts and Abstract Data Model, W3C working
WD-rdf-concepts-20020830 .
3. W. Beek et al., "LOD Laundromat: A Uni-
form Way of Publishing Other People’s Data
pp. 211–230.
4. C.B. Aranda et al., "SPARQL Web-Querying
Infrastructure: Ready for Action?" Proc.

Wouter Beek is a PhD researcher in the Knowl-
edge Representation and Reasoning (KRMH)
group at VU University Amsterdam. His research
focuses on the Semantic Web as a platform
for knowledge-intensive applications, the deploy-
ment of large-scale knowledge bases for innov-
ative reuse, and the interaction between Web
semantics and pragmatics. Beek has an MSc in
logic from VU University Amsterdam. Contact him
at w.gj.beek@vu.nl.

Laurens Rietveld is a PhD candidate in the
Knowledge Representation and Reasoning
(KRMH) group at VU University Amsterdam.
His research focuses on practical solutions
for Linked Data publishing and consumption,
using concepts and solutions from outside
the Linked Data community (such as network
analysis and columns stored to bridge the
gap between research and practical scalable
solutions. Rietveld has an MSc in artificial
intelligence from VU University Amsterdam.
Contact him at l.rietveld@vu.nl.