# Towards sustainable publishing and querying of distributed Linked Data archives

Miel Vander Sande[†*], Ruben Verborgh[†], Patrick Hochstenbach[þ], Herbert Van de Sompel[‡]

[†] Ghent University – imec
Sint-Pietersnieuwstraat 41
9000 Ghent, Belgium
*{miel.vandersande,ruben.verborgh}@ugent.be*

[‡] Los Alamos National Laboratory
Los Alamos, NM, USA
*herbertv@lanl.gov*

[þ] Ghent University Library
Rozier 9
9000 Ghent, Belgium
*patrick.hochstenbach@ugent.be*

**Abstract**

**Purpose** This paper details a low-cost, low-maintenance publishing strategy aimed at unlocking the value of Linked Data collections held by libraries, archives and museums.

**Design/methodology/approach** The shortcomings of commonly used Linked Data publishing approaches are identified, and the current lack of substantial collections of Linked Data exposed by libraries, archives and museums is considered. To improve on the discussed status quo, a novel approach for publishing Linked Data is proposed and demonstrated by means of an archive of DBpedia versions, which is queried in combination with other Linked Data sources.

**Findings** We show that our approach makes publishing Linked Data archives easy and affordable, and supports distributed querying without causing untenable load on the Linked Data sources.

**Practical implications** The proposed approach significantly lowers the barrier for publishing, maintaining, and making Linked Data collections queryable. As such, it offers the potential to substantially grow the distributed network of queryable Linked Data sources. Because the approach supports querying without causing unacceptable load on the sources, the queryable interfaces are expected to be more reliable, allowing them to become integral building blocks of robust applications that leverage distributed Linked Data sources.

**Originality/value** The novel publishing strategy significantly lowers the technical and financial barriers that libraries, archives and museums face when attempting to publish Linked Data collections. The proposed approach yields Linked Data sources that can reliably be queried, paving the way for applications that leverage distributed Linked Data sources through federated querying.

[*]Corresponding author

# 1 Introduction

## 1.1 Demolishing Metadata Silos

Libraries, Archives and Museums (LAMs) are long-term custodians of substantial structured metadata collections and active information curators. Over the past decades, digitizing records and making them available online has become irresistible and unavoidable. A digital agenda has allowed LAMs to further democratize knowledge by making collections more broadly available to audiences and applications alike (Clough, 2013). This knowledge only reaches its highest potential when we are able to query *across* different data sources, but unfortunately, too much data remains confined to the basements of individual institutions. In order to break through these *silos of the LAMs* (Zorich *et al.*, 2008), a strong engagement in metadata sharing was put on top of the digital agenda. On the bright side, this has already resulted in the establishment of standards and best practices for expressing and sharing metadata aimed at achieving cross-institution interactions (Waibel and Erway, 2009). Efforts toward the eventual goal – offering an *"integrated, seamless level of service that tech-savvy users are increasingly coming to expect"* (Leddy, 2012) – resulted in two types of investments made by LAMs: *a*) *semantic integration*, aligning metadata descriptions across institutional boundaries, and *b*) *Web publishing*, leveraging the Web to share metadata in a manner that supports effective reuse beyond institutional boundaries. On a more critical note, however, these efforts have not fully paid off yet, since actual integration of the cross-institutional data for end-user queries has so far remained a technical challenge. The resulting decisions are layed out in Figure 1 and discussed next.

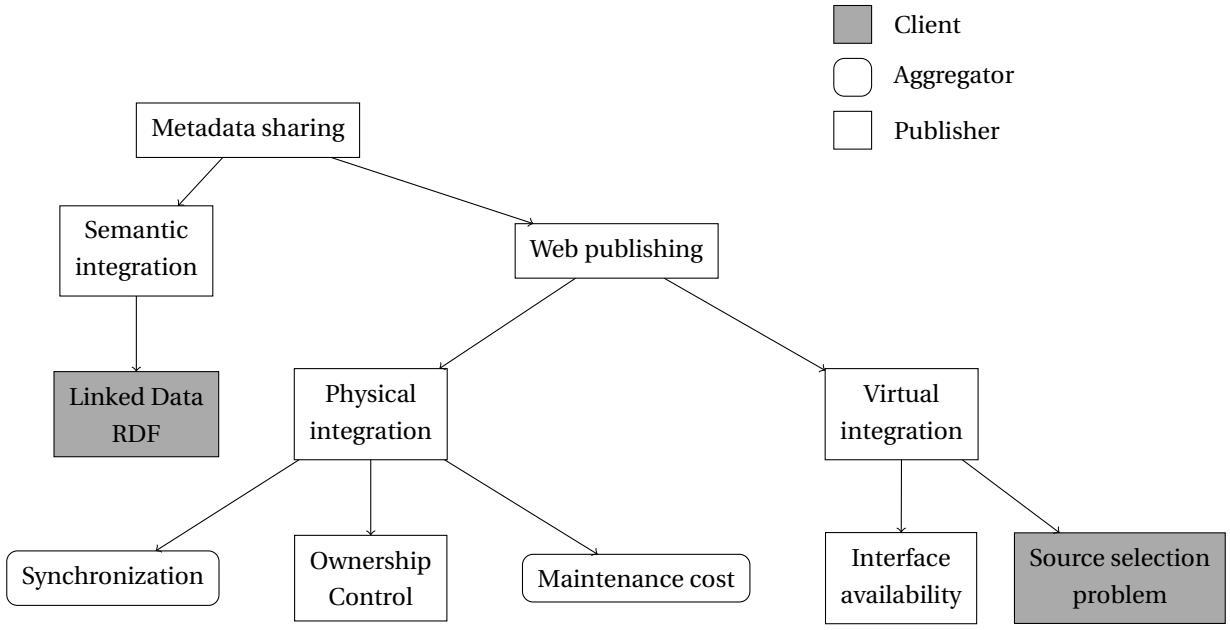



**Figure 1:** LAM institutions choose between *physical* data integration and *virtual* data integration strategies to publish their metadata on the Web.

### 1.1.1 Semantic integration with Linked Data

To integrate collections semantically, institutions have started to adopt a Linked Data approach, which Bizer *et al.* (2009) define as "a set of best practices for publishing and connecting structured data on the Web". Linked Data is most commonly materialized using the Resource Description Framework (RDF), which entails the use of basic machine-actionable relationship statements – called triples – composed of three components: a subject, a predicate and an

object. RDF leverages the the global HTTP URI scheme to identify resources. Thus, semantic integration is achieved by reusing the same URI to refer to the same resource, or by expressing equivalence between different URIs that identify the same resource (Hausenblas, 2009). As institutions use this approach, their respective RDF descriptions pertaining to a given resource are complemented by those of other institutions. Typically, when descriptions are merged, the provenance of each RDF statement is maintained. When these descriptions share URIs, they become automatically interconnected, resulting in a distributed *Web of Data* (Heath and Bizer, 2011).

### 1.1.2 Web publishing through physical integration

Currently, the most common approach by which institutions expose collections of RDF statements for reuse is to make them available as Linked Datasets for batch download. In this approach, one or more aggregators step in and collect the distributed datasets and publish a merged dataset either – again – for batch download or as a machine-queryable endpoint. This *physical integration* approach is cost-effective for institutions that expose Linked Datasets and aggregators often add value, for example, by performing data cleansing and mapping equivalent URIs. But the approach also has a some important drawbacks.

First, data in different institutions evolves at a different pace. Keeping an aggregated dataset continuously synchronized with the evolving distributed datasets is a non-trivial technical challenge (Klein *et al.*, 2014); tackling it in a realistic manner would necessarily involve additional infrastructure (and hence investment) at the end of the institutions that expose them. Lacking this, at any moment in time, it is uncertain whether or not an aggregated dataset is in sync with the state of the datasets it merges.

Second, institutions often fear loosened control when making their Linked Datasets available for reuse. Considering their own datasets as highly curated, they might be reluctant to allow a merge with datasets perceived to be of lower quality. In addition, descriptions originating from different institutions are commonly made available under different licenses, making it difficult to understand what the terms of use for aggregated descriptions are, especially when licenses are in conflict. Although maintaining the data provenance for the entire aggregation could ensure a basic sense of control, this burdens institutions to share this information as well.

Third, when an aggregated dataset is exposed as a centralized query endpoint, scaling its infrastructure can become expensive and unpredictable, as the server needs to be prepared to respond to potentially complex queries from an unknown amount of client applications over an extensive merged dataset.

### 1.1.3 Web publishing through virtual integration

Another approach that is currently used by institutions to make Linked Datasets available for reuse is to expose their own query endpoints. In this case, client applications query distributed datasets, benefiting from a uniform query interface. This *virtual integration* approach is more expensive for institutions because it requires maintaining these endpoints. Also, it yields the non-trivial *source selection problem* (Saleem *et al.*, 2016) as client applications need to limit the distributed datasets they consult for any given query in order to balance completeness of results with acceptable latency.

Despite these issues, virtual integration does not have the significant drawbacks of physical integration described above. If technological advances can be made that ameliorate problems related to data source selection, uniform access for clients, and maintenance costs, it is an attractive alternative to avoids conflicts on policy level.

## 1.2 Linked Data adoption in libraries, archives and museums

Linked Data's main premise is that multiple datasets can be uniformly accessed and interpreted, facilitating knowledge integration and sharing. The uniform interface enables answering more complex questions by combining multiple Linked Datasets. This potential has inspired the LAM community, which has an inherent focus on information sharing, to adopt Linked Data principles and to start implementing them at scale (van Hooland and Verborgh,

2014). In a recent survey conducted by OCLC Research (Smith-Yoshimura, 2014), 96 participants identified 172 Linked Data projects or services being implemented. Of the 76 that were actually described, 67% published Linked Data. These project mostly aimed at bibliographic metadata enrichment, data interlinking, source referencing, unifying data from various sources, and enhancing existing applications.

Meanwhile, the size of some available datasets already ranges between tens of millions and billions of triples. Prominent examples include WorldCat.org (15 billion), Europeana (4 billion), The European Library (2 billion), Library of Congress (500 million) and the British Library (100 million). Efforts are currently ongoing in a wide range of domains (Mitchell, 2015), including electronic thesis and dissertations (ETD) (Mak *et al.*, 2014), image collections (Getty http://www.getty.edu/research/tools/vocabularies/lod/index.html), digital humanities (DARIAH http://www.dariah.nl/, Pleiades https://pleiades.stoa.org/), cultural heritage (Nationale Coalitie Cultureel Erfgoed http://www.ncdd.nl/).

A main focus has also been on establishing basic building blocks that are essential for a LAM Linked Data infrastructure such as expressing bibliographic metadata, authority files, subject classification schemes, etc. as Linked Data. Examples include BIBFRAME (Kroeger, 2013), BIBFRAME Lite[1], VIAF [2], LCSH [3], Medical Subject Headings (MeSH)[4] as RDF, and the International Standard Bibliographic Description (ISBD) in Semantic Web (Bianchini and Willer, 2014).

As major motivations to adopt Linked Data, publishers indicate *a*) exposure to larger audiences, and *b*) broadly demonstrating the added value of their datasets (Smith-Yoshimura, 2014). The community's wish for more *visibility* is confirmed by Mitchell (2015). On the consumer side, Digital Humanities and Data Science stand to benefit, as the availability of an ever increasing number of datasets exposed through a uniform interface lowers the barrier for inclusion in research threads. But, obviously, the general public will benefit from the integration of currently silo-ed datasets, and from multi-lingual, multi-faceted user interfaces.

Despite this increase in uptake, Mitchell (2015) states it is unsure *"whether or not Linked Data has reached critical mass in the LAM community to ensure further adoption and transformation"*. While tight collaboration between LAM institutions has already resulted in some interesting merged Linked Datasets (e.g. Europeana), the question needs to be asked as to why, thus far, very few applications that leverage multiple datasets have emerged. An anecdotal illustration of this consideration was provided at a recent cultural heritage Liked Data hack-a-ton in The Netherlands [5]. Although the charge was explicitly to build demonstrators that integrated heterogeneous Linked Data sets, only 1 out of 8 projects actually did.

The slow adoption is usually attributed to the *vocabulary chaos* (Dunsire *et al.*, 2012), resulting from poor practices in vocabulary management, development, and re-use. But, (Miller and Ogbuji, 2015) argues that the details about vocabularies and modeling can gradually be refined as we go along, and that the first goal should be to increase the visibility of library collections on the Web, to establish the *Visible Library*. Libraries, archives and museums will naturally become more visible Web sources of credible information as they increasingly collaborate and interlink their collections.

Erik *et al.* (2015) indicates that LAM institutions consider selecting infrastructure required to surface functional Linked Data (e.g. triplestores, SPARQL engines, indexing platforms) a high threshold for their projects. Easily deployable, comprehensive publishing solutions are not available. In general, Linked Data *publishing costs* are higher than publishing those of well established traditional publishing solutions (e.g., Integrated Library Systems, Digital Asset Management systems) and as such present a proposition that is hardly feasible or sustainable (Erik *et al.*, 2015). Also, adoption of a Linked Data publication approach does not mean giving up on the traditional approach. Indeed, the traditional systems typically provide a wide variety of services beyond description and discovery. Hence, early Linked Data adopters have no other option than to invest in both.

Clearly, bringing down the *cost* of Linked Data publishing would be welcome as a means to increase adoption

[1] http://www.bibfra.me/
[2] http://viaf.org/viaf/data/
[3] http://id.loc.gov/authorities/subjects.html
[4] https://id.nlm.nih.gov/mesh/
[5] http://hackalod.com/

by LAMs. If the Linked Data could also be published in a manner that lowers the barrier for the development of cross-institutional applications, the increased *visibility* that LAMs strive for would also be in reach.

## 1.3 The need for Linked Data archiving

Thus far, publishing and consuming Linked Data has largely focused on collections of current RDF statements. This focus on leveraging the current Web of Data to a large extent parallels the typical focus on current content on the Web. As a matter of fact, this *eternal now* perspective is hardwired in the very architecture of the Web as, at any moment in time, the representation that is available from an HTTP URI is the then-current one. But Web content is ephemeral; content disappears or changes over time.

The consideration that our shared digital footprint was vanishing at an alarming rate (Koehler, 2002; Ntoulas *et al.*, 2004)led to the emergence of Web archives that crawl the Web and capture content to preserve it for prosperity. It also led to the Memento protocol (Van de Sompel *et al.*, 2013), a lightweight extension of HTTP that introduces datetime negotiation as a means to bridge between the current Web and remnants of the past Web held in Web archives and resource versioning systems. Since their inception, Web archives have been used as a means to revisit old Web pages, one by one. Increasingly, they are used as Big Data sets of historical significance.

Thus far, comparable systemic efforts aimed at collecting and preserving Linked Datasets have been rare. However, as Linked Data publishing becomes more mainstream, it is to be expected that archiving efforts will follow suit. If this happens, the ability to access and query archived Linked Datasets – possibly residing in distributed archives – will become essential. The aforementioned source selection problem will then have both a content-based and a time-based dimension, as queries will need to be issued against temporally matching datasets in order to yield results that are historically accurate.

LAMs have an inherent role as curating and preserving institutions, and have played a pioneering role in long-term digital preservation efforts. As such, it would only be natural for LAMs to also play a leadership role when it comes to archiving Linked Datasets, preserving them, and making them accessible. Admittedly, this seems like yet another daunting task to be achieved with the limited resources available. Then again, as this paper argues, an appropriate choice of technologies may help to significantly constrain the investment required to establish an ecology of distributed Linked Data Archives that provides significant value to downstream applications and users. Meanwhile, it enhances the combined agenda of long-term digital preservation and access (Corrado and Moulaison, 2014; Hedstrom and Montgomery, 1998), as ongoing Linked Data access can not be reliable over time *without* preservation (Rinehart *et al.*, 2014).

## 1.4 A sustainable strategy for publishing queryable Linked Data archives

This article takes a practical perspective on cross-institutional data integration utilizing Linked Data. As *virtual data integration* conceptually addresses pressing concerns in the LAM community, we aim to revive it as a viable, concrete alternative for sharing metadata collections over the Web by introducing approaches that improve on the technical status quo. We focus on a sustainable publishing strategy for institutions that consists of exposing Linked Data interfaces that achieve a sweet spot regarding *interface availability* by balancing maintenance costs and query functionality. Furthermore, this strategy complements the Extract-Transform-Load (ETL) workflow commonly used for Linked Data generation, and combines the preservation and access facilities expected in a digital preservation setting.

Concretely, we propose a Linked Data publishing tool chain to support querying, with the following four-fold contribution:

1. We introduce a strategy of recurrently publishing a static snapshot of an evolving Linked Dataset, while the evolving Linked Dataset itself is not published. For a static snapshot, we suggest a low-maintenance, easily archiveable binary storage format that supports basic queries.

2. We show how these recurrently taken snapshots of a Linked Dataset, published using the binary storage format, can be used as building blocks of a Linked Data Archive that stores temporal versions of Linked Datasets.
3. We describe a lightweight wrapper interface to the Linked Data Archive that transparently supports basic temporal queries into contained Linked Dataset versions.
4. We demonstrate how complex queries can be executed over distributed Linked Data Archives using the lightweight wrapper interface.

The remainder of this article is structured as follows. First, Section 2 provides an overview of our alternative tool chain and supplies a validating use case. Next, Section 3 introduces Linked Data and its related concepts used in our approach. Then, the proposed toolchain is discussed layer by layer in Section 4 (Storage), Section 5 (Publishing), and Section 6 (Querying) in context of our use case. Finally, we identify some remaining challenges to further improve the visibility of Linked Data of LAM institutions in Section 7, and end with some general conclusions in Section 8.

# 2 Approach and Methodology

Next, we introduce a layered Linked Data publishing tool chain, which is more maintainable and unlocks more data potential. In this section, we first provide a high level overview and illustrate the overall strategy shift compared to the current approach. In order to validate our claims, we then introduce a use case that states a set of queries to represent a real-world scenario. Later sections provide more details about each layer and reprise the use case.

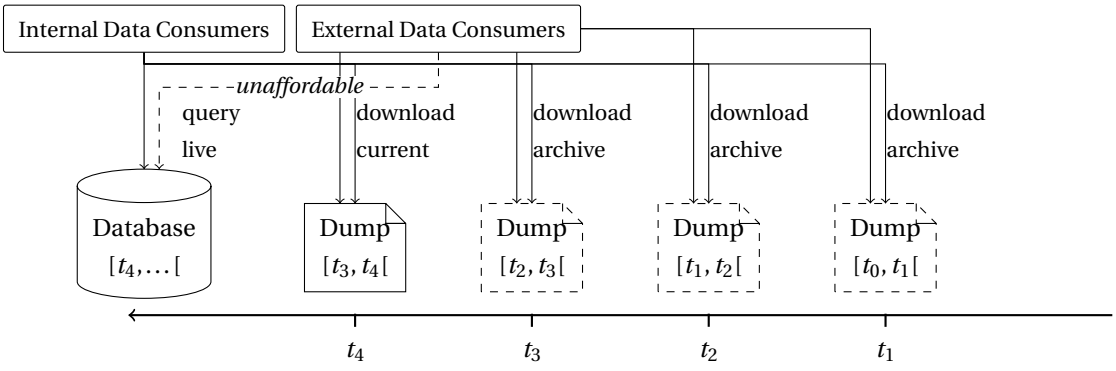
## 2.1 Linked Data publishing tool chain

Often, an institution's RDF metadata collection is not continuous, but is generated periodically from the current collection with an extract-transform-load (ETL) process. Metadata can reside in heterogeneous data sources, which need more complex transformations first in order to be unified (Binding *et al.*, 2015). Consequently, with attention to digital preservation, a Web publishing workflow always applies to an RDF *version* of the dataset, for which three conceptually different types exist:

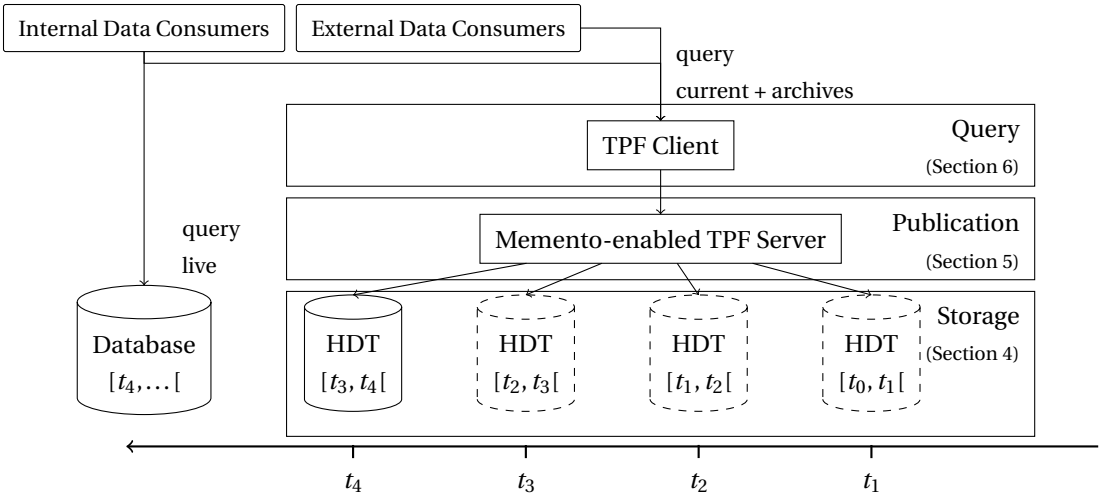
- the **live** version, representing the last possible state of the dataset, i.e., the collection currently known to the publishing institution.
- the **current** version, representing the latest *published* state of the dataset, i.e., the most recent version accessible to consumers.
- an **archive** version, representing a state of the dataset bounded by a fixed temporal range for which it is valid, i.e., preserved versions for future reference.

Typically, the *live* version is loaded into an RDF database, which offers powerful functionalities to clients, such as fine-grained querying. Yet, underresourced organizations struggle to enhance this workflow to provide effective external access to their digital preservation efforts — the *archive* versions. They resort to the Linked Data publishing approach shown in Figure 2a. Despite its usefulness, the *live* RDF database rarely serves as *current* version as well, i.e., exposing the database to external consumers, but is restricted to internal consumption; and understandably so. As mentioned before, there is little economical justification to share complex, maintenance-intensive SPARQL systems with unpredictable load. Thus, such institutions release timely *archive* versions as data dumps and make them available for download (Yoshimura, 2016). The most recent dump serves as *current* version, and is therefore never entirely up-to-date. In the best case, multiple *archive* dumps serve as preservation strategy, making the history available for internal and/or external use; however, aforementioned issues, like the fear of loosened control, might prevent publishers to keep all archives available forever. Despite its inexpensive character, this approach strongly limits functionality for the client, as the data is not directly queryable.

Figure 2b displays our suggested enhancements that continues consideration for the under-resourced institutions. A low-cost virtual integration strategy can democratize Linked Data publishing for many small and mid-sized institutions, but digital preservation cannot be considered an afterthought here as well; in this regard, Schumacher




**(a)** Common strategy




**(b)** Proposed strategy

**Figure 2:** Enhancements of the proposed Linked Data publishing strategy to the common approach.

*et al.* (2014) calls for "Good Enough" Digital Preservation Solutions. Therefore, we argue to keep the database, holding the *live* version, unexposed, but to recurrently publish *archive* versions of a dataset in the more expressive Header-Dictionary-Triples (HDT). The *current* and *live* version still do not coincide, but compared to data dumps, HDT files are of lower maintenance. Although HDT requires an extra processing step, generating such files is still relatively cheap for publishers, while the extended lookup interface highly increases functionality for external and internal consumers (see Section 4). The true potential, though, lies in exploiting this modest shift to do digital preservation in combination with access through a low-cost Web publishing wrapper stack. Complex queries can be executed in a public setting on both a *current* version, i.e., the most recently extracted HDT, and its *archives*. This unlocked functionality is facilitated by the following three layers, each described in a separate section:

**Storage.** Section 4 describes how HDT snapshots can be organized to create a pragmatic RDF archive.

**Publication.** Section 5 discusses a wrapper interface to publish Linked Data with support for temporal queries.

**Query.** Section 6 explains how multiple of such interfaces can be queried together to answer complex queries over archives.

In its entirety, this toolchain yields three main benefits for the publisher, its internal consumers, and external consumers. First, infrastructure requirements become *lower cost* , while query capabilities remain. By restricting the expressiveness of the interface, the required computational resources become more predictable. This lowers

the barrier for institutions to publish their metadata collections online. Second, *uniform access* to the semantically integrated data across institutions is achieved by merely relying on the HTTP protocol and a self-describing interface. Hence, any Web application can interoperate with the datasets and clients can autonomously discover how to interact. Third, it provides *synchronization* in the sense that querying dataset versions that have matching temporality and that reside in distributed RDF Archives is possible using the same interface.

## 2.2 Use case: reconstructing institutional history from archives

To support and validate the approach above, we select a Digital Humanities use case that consists of *reconstructing memory from distributed knowledge sources*, for example, recreating the status of scientific collaboration networks or reassembling the virtual presence of institutions as they existed at some point in the past. In technical terms, in order to tackle this type of challenge, a client's need to be able to fulfill two actions: *a*) Selecting distributed data sources in a manner that ensures that their contained knowledge temporally coincides, that is, represents the state of affairs at a same point in time; *b*) Querying the selected distributed data sources to retrieve the data required to reconstruct the desired memory.

As a concrete example, we selected three semantically integrated Linked Datasets (Figure 3) as distributed data sources. These complementary datasets are created by different authorities, and, as is typically the case with Linked Datasets, they are interlinked through the mutual reuse of URIs. This selection of datasets is heterogeneous regarding the type of knowledge that is represented, the typical purpose the knowledge serves, and the size of the knowledge database:

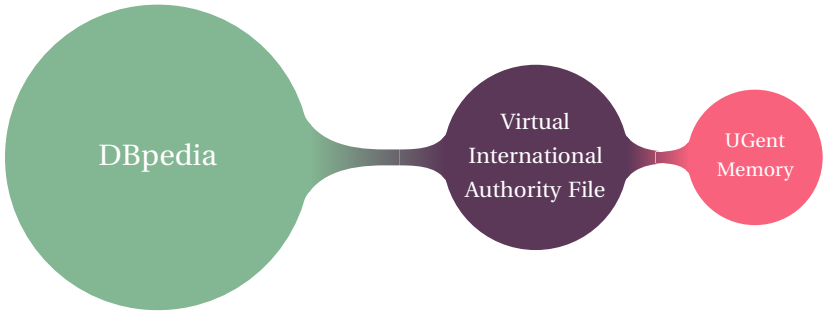
**DBpedia** DBpedia is a *large* (> 1B triples) RDF dataset derived from Wikipedia. It contains an abundance of *common knowledge* facts and acts as a *general-purpose* linking hub between numerous domain-specific datasets that reuse DBpedia URIs. DBpedia also reuses external URIs including those from the VIAF authority file. DBpedia is versioned in bi-annual static releases.

**Virtual International Authority File (VIAF)** VIAF (Bourdon and Boulet, 2013) is a reputable authority file that is jointly compiled by LAMs worldwide. An authority file is an independent index of authority records to relate and govern the headings used in a bibliographic catalog, thus enforcing authority control. VIAF is a *medium-sized* (10M – 1B triples) dataset available in various formats, including as a Linked Dataset. It contains entries pertaining to organizations and authors, and as such qualifies as a *thematic* dataset. As a global authority file, it is a *general-purpose* dataset that is used in a wide variety of settings, including as a linking hub in the LAM community.

**UGentMemorialis** UGentMemorialis (Verbruggen and Deneckere, n.d.) is a *small* (< 10M triples), *thematic*, *organization-specific* Linked Dataset that contains information about professors that worked at the University of Ghent, Belgium. It is maintained by the University Library, and serves as a prime example of a LAM institution using Linked Data to increase institutional visibility. UGentMemorialis uses VIAF URIs to uniquely identify the professors about which it holds information.

In accordance to the aforementioned client actions, we formulate four queries over DBpedia, VIAF and/or UGentMemorialis, that each represent a common query format. Besides the sources' *current* version, we also target *archive* versions to enable memory reconstruction. Thereby, we tackle the problem of *reproducibility* caused by evolving RDF statements. Factual changes, i.e., demographics or artwork portfolios, or the schematic changes of the dataset itself, i.e., use of categories, types or relations, prevent re-executing a query in its valid context. Querying drifting sources simultaneously can produce invalid results. Prominent examples are schema changes, broken links, or combining facts that are no longer true.

Therefore, our use case targets both *single source* and *multiple sources*, to access the contained knowledge of distributed data sources. Such queries exploit the links between multiple datasets i.e., UGentMemorialis is linked to DBpedia via VIAF through the resources that represent authors. By combining multiple sources that support access to *archive* versions, we can ensure temporal synchronization. The resulting queries are presented below:




| | | | | |
|---|---|---|---|---|
| **Knowledge:** | Common | | Thematic | Thematic |
| **Purpose:** | General | | General | Organization-specific |
| **Size:** | Large | | Medium | Small |

**Figure 3:** Our Use Case exists of a illustrative semantically integrated network of three organizations publishing the datasets DBpedia, VIAF and UGentMemorialis.

**Query to a Single Source (DBpedia)**

- Current: *What is the number of awards won by Belgian academics?*
- Archive: *How did the number of awards won by Belgian academics evolve between 2008 to 2016?*

**Query to Multiple Sources (DBpedia, VIAF & UGentMemorialis)**

- Current: *What is the number of DBpedia triples describing all professors of Ghent University?*
- Archive: *How did the number of DBpedia triples describing all professors of Ghent University evolve between 2008 and 2016?*

# 3 Preliminaries: technologies, concepts and languages

In this section, we first cover the technical aspects of Linked Data that are essential for an understanding of this paper. Also, we introduce the existing technologies Header-Dictionary-Triples (HDT), Linked Data Fragments and Memento that are used in our approach.

## 3.1 Linked Data in the Resource Description Framework

Linked Data is data created and published in a way that makes it interoperable and interconnected with other data. The Resource Description Framework standard (RDF) provides a model and syntaxes to implement and represent Linked Data (Cyganiak *et al.*, 2014).

RDF represents data as *triples*. A triple consists of a *subject*, *predicate*, and *object* and can hence be considered a simple sentence. For example, the statement "'The Name of the Rose' was written by Umberto Eco" can be represented with an RDF triple as follows:

```
<http://dbpedia.org/resource/The_Name_of_the_Rose>
      <http://dbpedia.org/ontology/author>
            <http://dbpedia.org/resource/Umberto_Eco>.
```

We recognize the subject (*The Name of the Rose*), predicate (*has author*), and object (*Umberto Eco*). Together, they form a triple that constitutes a statement.

RDF achieves interoperability and interconnection by using Uniform Resource Identifiers (URIs): each of the components of the above triple is in fact a URI. This facilitates unambiguous referencing to concepts such as book, author, and predicates. The syntax we use to write triples in this paper is Turtle, which uses angular brackets (<>) to indicate URIs. For brevity, Turtle allows to abbreviate URIs with prefixes. This article uses the following prefixes:

```
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

More complex data can be expressed by constructing an RDF *graph*, i.e., a collection of triples:

```
dbr:The_Name_of_the_Rose  dbo:author dbr:Umberto_Eco.
dbr:The_Name_of_the_Rose  rdf:type   dbo:Book.
dbr:Umberto_Eco  dbo:birthDate  "1932-01-05"^^xsd:date.
```

Note the use of prefixes to shorten URIs; the triple on the first line is identical to the triple we discussed earlier. The above triples state that *something* identified by `dbr:The_Name_of_the_Rose` is of type `dbo:Book` and has a relationship identified by `dbo:author` with *something* identified by `dbr:Umberto_Eco`. In turn, the thing identified by `dbr:Umberto_Eco` is related to 5 January 1932 with the predicate `dbo:birthDate`. However, by visiting the URIs and applying the machine-readable semantics captured by the schema – in this case the DBpedia ontology prefixed by `dbo:` – a software application can interpret that a *book The Name of the Rose* is authored by *Umberto Eco*, who was born on 5 January 1932. Note that the triple on the last line has a literal value as object, which is common for basic values like strings, numbers, and, in this case, dates. But, for reusable concepts, the use of URIs is preferred to be referenceable.

## 3.2 Header-Dictionary-Triples: a self-indexed and compressed RDF format

Header-Dictionary-Triples (HDT) is a compressed self-indexed binary RDF format introduced by Fernández *et al.* (2013). Designed with exchange in mind, it targets the ever increasing data volumes by dealing with the redundancy and verbosity custom to RDF representations, and the expensive parsing it bears.

An HDT file encapsulates an RDF dataset into a single file, consisting of three components:

- a *Header* with metadata about the dataset for discovery and as entry point
- a *Dictionary* to encode all URIs and Literals to avoid redundancies
- a *Triples* encoding scheme which both compresses and indexes for search operations.

The result is a highly compressed read-only binary archive, with reduces the original dataset size up to 15 times (Fernández *et al.*, 2013). For big semantic data management systems, HDT offers 25% storage space compared to state-of-the-art RDF indexes, while still competing in query performance (Martínez-Prieto *et al.*, 2015).

The Triples component enables very efficient search and browse operations. Triple pattern lookups can be performed fast without having to decompress any data, keeping storage and memory usage within acceptable bounds. In addition, it can estimate the cardinality of such patterns efficiently, which is useful for optimizing query planning over HDT files.

## 3.3 Querying with SPARQL

We can extract information from the collection of available Linked Data by querying. The SPARQL Query Language and Protocol is a W3C specification (Harris *et al.*, 2013) that describes a uniform query interface for RDF datasets. The specification defines two distinct concepts, which both are referred to by the acronym "SPARQL":

1. **the SPARQL language**: a query syntax and algebra to formalize questions, and;
2. **the SPARQL protocol**: a HTTP interface for clients to request the execution of SPARQL queries by a server.

The next subsections elaborate on each of these concepts.

### 3.3.1 The SPARQL language

The SPARQL query language has an SQL-like syntax that supports expressing graph patterns that can match a number of triples in a dataset. A graph pattern is composed of one or multiple triple patterns. A *triple pattern* is similar to an RDF triple, but it can have a wildcard as subject, predicate and/or object. For instance, a triple pattern aimed at selecting all works by Umberto Eco is expressed as follows:

```
?work dbo:author dbr:Umberto_Eco.
```

Every triple in the dataset that has `dbo:author` as predicate and `dbr:Umberto_Eco` as object will match the above query. By combining several triple patterns, more complex graph patterns can be constructed that match a collection of connected triples. For instance, the following query is aimed at selecting all works by Umberto Eco, and returning their label (prefix declarations are omitted for brevity):

```
SELECT ?name
WHERE {
  ?work dbo:author dbr:Umberto_Eco.
  ?work dbo:name ?name.
}
```

The first triple pattern yields a set of URIs, each identifying a *work* authored by Umberto Eco. The second triple pattern yields the label associated with each of those URIs. That is, from the description of the work, the object of the triple that has the work's URI as subject and the `rdfs:label` as predicate is chosen. The query as such returns all the resulting labels.

### 3.3.2 The SPARQL protocol

The SPARQL protocol defines the notion of an *endpoint*, which is a standardized interface on top of HTTP that accepts queries formulated in the SPARQL language and returns the results. The interface simply consists of a single `/sparql` resource with the query encoded in the parameter `query`. In the HTTP response, the query results are formatted according to a predefined structure and can be serialized as CSV, JSON or XML.

For example, the query above can be executed on the DBpedia SPARQL endpoint by performing a GET request to the following resource:

```
http://dbpedia.org/sparql?query=SELECT%20%3Fname%20WHERE%20%7B%20%3Fwork%20%3Chttp%3A%2F%
    2Fdbpedia.org%2Fproperty%2Fauthor%3E%20%3Chttp%3A%2F%2Fdbpedia.org%2Fresource%2FUmberto_Eco%
    3E.%20%3Fwork%20%3Chttp%3A%2F%2Fdbpedia.org%2Fproperty%2Fname%3E%20%3Fname%20%7D
```

The SPARQL protocol has become the de-facto approach for clients to query RDF datasets. Like for SQL databases, a protocol with such extensive query expressiveness can work adequately in controlled, private environments. However, SPARQL is also used by publishers to offer an open query interface to RDF data on the *public* Web. Since SPARQL allows for arbitrarily complex queries, which, on the open Web, can originate from an unlimited and unpredictable number of simultaneous clients, hosting a public SPARQL endpoint yields significant scalability problems, as assumptions that hold in a private environment (predictable number of users, predictable query types, ...) do no longer hold in public environments.

As a result, few open SPARQL endpoints exist compared to the number of available datasets, and research has shown that the ones that do exist are frequently unavailable (Buil-Aranda *et al.*, 2013).

## 3.4 Linked Data Fragments

To share Linked Open Data online, Libraries and cultural heritage institutions publish their collections through a Web interface. For interoperability's sake, a LOD environment that supports SPARQL querying is recommended (Mitchell, 2013). Hence, hosting a public *endpoint* supporting the SPARQL protocol is an obvious choice (Marden *et al.*, 2013).

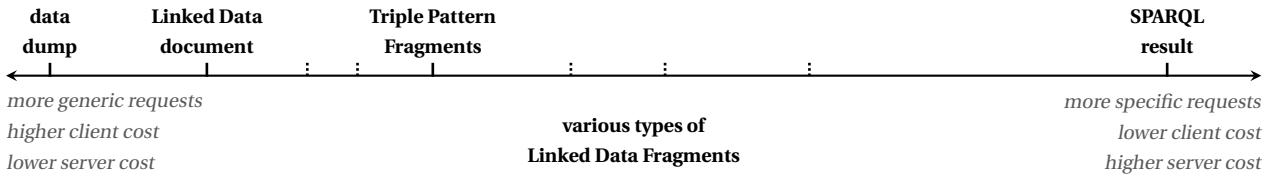



**Figure 4:** All Web APIs to RDF triples offer Linked Data Fragments of a knowledge graph. These fragments differ in the specificity of the data they contain, and thus the cost to create them.

Successful examples are Europeana (Isaac and Haslhofer, 2013) serving metadata about 2.4M objects, the British Museum[6] publishing over 750,0000 records, or U.S. National Library of Medicine[7] offering access to their biomedical information from the Medical Subject Headings (MeSH).

However, not all institutions enjoy similar financial resources, and consider the maintenance cost of a reliable public SPARQL endpoint too high. This resulted in a clear online dominance of the downloadable *data dumps* or Linked Data documents. Of 9960 known datasets, 2838 are dumps opposed to 151 endpoints[8]. Despite that the majority of triples (98%) is served by the endpoints, this number is dominated by a minority of datasets. A recent RDF dump crawl collected over 38 billion triples[9]), indicating that, along with the 900,129 crawled Linked Data documents (Schmachtenberg *et al.*, 2014), many small datasets out there fail to be published in a queryable way. Due to their low expressivity, these significantly more economical Web interfaces remain popular. Indirectly, SPARQL querying is still possible client-side, but is no longer live or fast. Even then, the complete dataset needs first to be downloaded, and, then, indexed in a local RDF database. Clearly, both interfaces can not be the final answer, as Web APIs always have thrived in diversity.

The Linked Data Fragments conceptual framework (Verborgh *et al.*, 2014) was defined to explore more client-server trade-offs in Linked Data publishing. This framework enables the analysis and comparison of Web APIs by abstracting each API by its access method to subsets of a certain dataset. As illustrated in Figure 4, each of these APIs can be plotted on a horizontal axis according to specificity, their computational cost, or bandwidth needs. Such subset is a *Linked Data Fragment* (LDF), consisting of data, metadata, and controls.

- **data** is a set of those triples of the dataset that match a given interface-dependent selector.
- **metadata** is the set consists of triples that describe the dataset and/or the current fragment or related fragments.
- **controls** are hypermedia links and/or forms that allow clients to retrieve other fragments of the same or other datasets.

Data dumps, Linked Data documents, and SPARQL endpoint responses can be described as LDFs as well. A data dump exploits a dataset's entry triple set as the data. Information such as filename, publication date or license composes the metadata. Controls are missing, since the entire dataset is already contained.

Linked Data documents are a HTML response when dereferencing a URI, for instance, the DBpedia page http://dbpedia.org/page/Linked_data. In this case, the data are all triples having this URI as subject or object. The metadata is often empty, but the control set contains all the dereferencable URIs in the response.

The SPARQL protocol responds to queries in the SPARQL language. Hence, each response from a `CONSTRUCT` query, which returns RDF, is considered an LDF. The data is the set of RDF triples that matches the query, while control and metadata sets are empty.

Unfortunately, neither of them is ideal in an archiving scenario. Linked Data Documents might be easy to serve, but is still very inefficient for complex query execution. For SPARQL endpoints, the complexity of the underlying

[6] http://collection.britishmuseum.org/sparql
[7] https://id.nlm.nih.gov/mesh/query
[8] http://stats.lod2.eu/
[9] http://lodlaundromat.org

index or storage structure increases greatly.

### 3.5 Memento

Memento is a straightforward extension of the HTTP protocol that adds a time dimension to the web. It adheres to the REST and "follow your nose" patterns of web architecture and consists of two components: TimeGates and TimeMaps. A TimeGate is a resource associated with an Original Resource that supports datetime negotiation, a variant on HTTP content negotiation. A user agent interested in retrieving a past representation of an Original Resource follows a link - provided in the Link response header of the Original Resource - to the TimeGate. Next, the user agent requests a past representation of the Original Resource by providing a preferred datetime - expressed using the accept-datetime header - in its request to the TimeGate. The TimeGate then redirects the user agent to a past representation of the Original Resource that is temporally the best match for the requested datetime. Since it is unlikely that prior representations - named Mementos - exist for all possible requested datetimes, there may be a temporal discrepancy between the user agent's preferred datetime and the version/archival datetime of the Memento to which the TimeGate redirects. Therefor, the Memento expresses its datetime using the memento-datetime response header. A Memento also provides a link - again conveyed in the Link header - to the original resource of which it is a prior version. As such, the TimeGate component of the Memento protocol provides a bridge from the current Web to the Web of the Past and back. A TimeMap is a resource that provides a version history of the Original Resource it is associated with, including the URI and version/archival datetime of each known Memento. A user agent finds its way to a TimeMap by following a link - expressed in the Link header - provided by a TimeGate, a Memento, or an Original Resource.

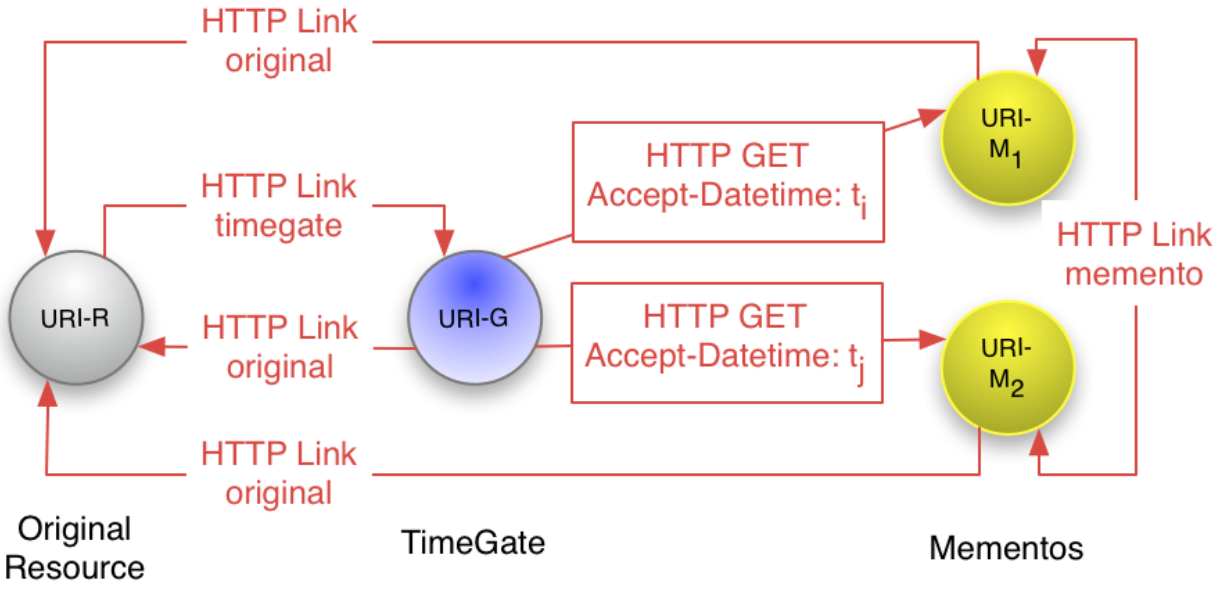



**Figure 5:** Memento adds a time dimension to an Original (Web) Resource by defining "follow your nose" patterns between TimeGate and Memento resources.

The Memento protocol is currently supported by most public web archives around the world[10], including by the massive Internet Archive. As exemplified by the recent adoption by the W3C[11], there is a growing interest in supporting Memento for resource versioning systems such as wikis, document management systems, and software version control systems. The applicability and power of the Memento protocol for Linked Data has been pointed out

[10] http://mementoweb.org/depot/
[11] https://www.w3.org/blog/2016/08/memento-at-the-w3c/

as soon as the protocol was initially introduced (de Sompel *et al.*, 2010) and since then various efforts have leveraged it (Fernández *et al.*, 2015; Meinhardt *et al.*, 2015; Vander Sande *et al.*, 2013).

# 4 Storing Linked Datasets using the archive-ready HDT format

Before RDF data can be published on the Web and subsequently queried, it first needs to be *stored* in a sustainable way. The kind of queries clients want to execute typically influences how data is published, and the publication interface in turn sets the requirements for storage. However, for the purpose of cost-effective data archiving, we reverse this chain and instead start from the storage technology, as it is the constraining factor for the remainder of the pipeline. The goal is a static and easy to maintain storage solution for current and archive versions, striking a balance between storage space and accessibility.

## 4.1 Requirements for long-term metadata preservation

Access and preservation are preferably a combined effort (Corrado and Moulaison, 2014). This is particularly important when striving for *reproducability* of queries. As mentioned before, distributed sources drift and require time-based synchronization.

For the current move to Linked Data, Papastefanatos and Imis (2014) identified the two challenges in Linked Open Data evolution management: *a*) *quality control and maintenance*, semantic integration issues such as schema or URI changes, and *b*) *data exploitation*, ensuring valuable insights can be retrieved from the evolution itself. This article engages in the latter and offers so-called LOD long term accessibility (Papastefanatos and Imis, 2014), where "datasets with different time and schema constraints coexist and must be uniformly accessed, retrieved and combined".

Therefore, an archive providing a “Good Enough” Linked Data preservation solution (Schumacher *et al.*, 2014) should adhere to the following long-standing characteristics of data archives. Houghton (2016) puts the **large data volume** forward as major challenge. Although the overall cost is decreasing, it is still challenging to store, maintain, and process the vast amounts of metadata today. Thus, both infrastructure and software need to efficiently cope with resource constraints. Next, Ross (2012) states data in a digital library should be read-only in order to "accept it as authentic". Metadata collections need to be represented as **immutable snapshots** versions to protect them from change and corruption. Finally, to ensure synchronization, such past versions should be **immediately accessible**, i.e., browse and lookup operations on current and past snapshots need to be equally fast.

## 4.2 Pragmatic Linked Data archives with HDT

Figure 6 presents a pragmatic RDF archive founded upon the file-system, where a collection of HDT files is organized in a matrix structure. The vertical axis represents the different RDF collections that are currently archived. The horizontal axis represents versions of those collections generated at prior points in time. Each version is valid for the interval between the time it was generated and when the succeeding version was generated. These generation times can be retrieved from the metadata in the HDT Headers to automatically construct a simple lookup index. Thereby, we can retrieve the valid HDT file with a given dataset name (e.g., RDF Dataset B) and timestamp (e.g., $t_2$). Every newly extracted HDT file is added to the first column, which contains the *current* version. All affected datasets shift right in the matrix, and the index is updated to reflect this change.

For validation, we re-architected the existing DBpedia archive initially released by the Los Alamos National Laboratory in 2010. By applying the pragmatic archiving method, we decreased storage space and time-to-publish by orders of magnitude, as shown in Table 1. Simultaneously, the expressiveness of a search operation evolved from DBpedia URI dereferencing only to support for triple pattern queries, including dereferencing.

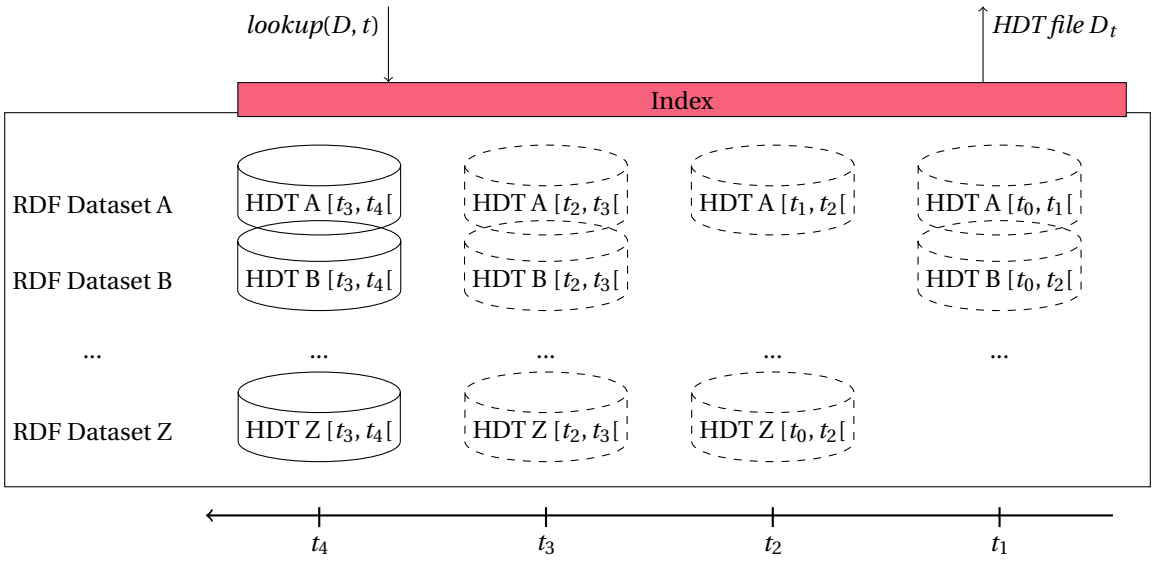



**Figure 6:** By applying HDT horizontally and vertically, we can create a simple, but efficient RDF archive.

| | **First generation** | **Second generation** |
|---|---|---|
| **indexing** | custom | HDT-CPP |
| **indexing time** | ~ 24 hours per version | ~ 4 hours per version |
| **storage** | MongoDB | HDT binary files |
| **space** | 383 Gb | 179 Gb |
| **# versions** | 10 versions: 2.0 through 3.9 | 14 versions: 2.0 through 2015-10 |
| **# triples** | ~ 3 billion | ~ 6.2 billion |
| **search expressiveness** | subject-based lookup | subject-based lookup<br>triple pattern |

**Table 1:** An DBpedia archive based on HDT files decreases storage space and the time-to-publish significantly.

The *first generation* used an archiving approach (de Sompel *et al.*, 2010) where DBpedia pages are reconstructed from a custom index, holding the ten DBpedia versions 2.0 to 3.9. Per DBpedia URI, this index stored the RDF description as a blob in MongoDB. Adding a new version triggered a re-indexing of the entire database, resulting in scalability issues that prevented further expansion. With Wikipedia being a constant-changing system, new DBpedia releases are frequent. Hence, a maximum of ten versions is unacceptable for an archive.

We replaced this custom solution with fourteen HDT files for version 2.0 to 2015. The compression ratios are shown in Table 2. On average, the size of each HDT file is only 13% of its N-triples counterpart. This has a huge positive impact on the amount of RDF triples that can be archived. In total, the *second generation* archive contains around 6.2 billion triples, which doubles the original amount with less storage use (Table 1). Storage space is decreased with 53%, allowing adding more snapshots in the future. The ETL process benefits as well, as it takes 20 hours less on average to a new DBpedia version into the archive.

| version | release date | # triples (millions) | N-Triples size (GB) | HDT size (GB) | compression ratio (%) |
|---|---|---|---|---|---|
| 2.0 | July 2007 | 8.5 | 2.6 | 1.4 | 53.85 |
| 3.0 | January 2008 | 120.3 | 18 | 2.3 | 12.78 |
| 3.1 | July 2008 | 137 | 21 | 2.7 | 12.86 |
| 3.2 | October 2008 | 150 | 22 | 2.6 | 11.82 |
| 3.3 | May 2009 | 170 | 25 | 2.9 | 11.60 |
| 3.4 | September 2009 | 190 | 28 | 3.1 | 11.07 |
| 3.5 | March 2010 | 257 | 37 | 4.4 | 11.89 |
| 3.6 | October 2010 | 288 | 42 | 4.6 | 10.95 |
| 3.7 | July 2011 | 386 | 55 | 5.5 | 10.00 |
| 3.8 | July 2012 | 736 | 103 | 6.6 | 6.41 |
| 3.9 | September 2013 | 812 | 115 | 7.2 | 6.26 |
| 2014 | September 2014 | 866 | 123 | 8.2 | 6.67 |
| 2015-04 | April 2015 | 1,030 | 142 | 8.8 | 6.20 |
| 2015-10 | October 2015 | 1,087 | 149 | 9.2 | 6.17 |

**Table 2:** Fourteen DBpedia versions 2.0 to 2015-10 can be stored with a high average compression rate of 13%.

As mentioned before, the second generation archive is not restricted to DBpedia pages. It supports efficient look-ups by triple pattern, which facilitates more complex query execution. In the next section, we explain how this capability is leveraged to create a sustainable infrastructure for publishing Linked Data archives on the Web.

# 5 Publishing versioned Linked Data

External consumption unlocks the real potential of archive versions. Thus, the archive discussed in the previous section needs to be exposed on the public Web. Downloadable RDF data dumps hardly qualify, as they imply high bandwidth usage, high client costs and weakened control for publishers; a queryable interface using SPARQL is preferred instead. Nevertheless, given the current problems with public SPARQL endpoints, special care is needed to make such a publication mechanism scalable. Hence, sustainable online access to an RDF archive requires a different LDF interface.

A Web API that publishes data generally has two main functions: *a)* to abstract any database-specific aspects, such as schema or query language, from the client; and *b)* to restrict the type of queries clients can execute. Interestingly, both are means for *scalability*, i.e. maximizing the number of data consumers that can be sustained over a long period of time. Databases are abstracted through the uniform HTTP interface, which scales the interoperability with clients. Restricting the types of queries balances the load on the server, by protecting the necessary computational resources or increased caching; it scales the cost or infrastructure.

It is vital to acknowledge these objectives. In fact, negligence in this regard causes SPARQL endpoints to fail on the public Web (i.e., direct SPARQL access). Therefore, this section suggest to combine the low-cost Triple Pattern Fragments (TPF) (Verborgh *et al.*, 2016) interface with the Web resource versioning protocol Memento (Van de Sompel *et al.*, 2013).

## 5.1 Triple Pattern Fragments

The Triple Patterns Fragments interface was defined as a trade-off between the low server-side cost of data dumps and the live queryable aspect of SPARQL endpoints, as displayed in Figure 4. This API only accepts triple patterns, and responds with a TPF. It is a specific LDF interface with the following data, metadata, and controls:

- **data** are all the triples in the dataset that match a given triple pattern. To keep the fragment size small, these fragments are often paged.
- **metadata** is the estimated total number of matching triples.
- **controls** are hypermedia to retrieve other TPFs of the same dataset.

Complex SPARQL queries can be evaluated on the client-side, by splitting the query into triple patterns and using the fragment's metadata to optimize the order of execution. As clients handle some of the complexity, the server-side requires lower processing power, which makes hosting RDF with high availability less expensive than a SPARQL endpoint (Verborgh *et al.*, 2016). In turn, executing a SPARQL query will take notably longer.

Practical examples have shown that this approach can publish RDF datasets with high availability. Since October 2014, snapshots of DBpedia have been published at an official Triple Pattern Fragments interface fragments.dbpedia.org. This service provided access to 1B triples by triple patterns. It received 19M requests so far with 99.999% (Verborgh *et al.*, 2015) availability up till now. For an average of 178K queries a day (Möller *et al.*, 2010), the SPARQL endpoint only reaches 98.86% availability[12], measured over the same timespan.

Another example is the LODLaundromat (Beek *et al.*, 2014), a republishing service for RDF datadumps. The Web is crawled for datasets, which are then cleaned and stored as HDT files and republished with a TPF interface. More than 650,000 datasets containing over 38B triples are now available for query. The possibility of having this amount of triples live accessible can put many new use cases into practice. For instance, Rietveld *et al.* (2015) proposed LODlab, an architecture to re-evaluate recent work in Semantic Web at Web-scale.

The Triple Pattern Fragments interface can be implemented with the HTTP protocol, by applying the REST architectural style. All possible fragments, and their pages, are assigned an IRI identifying them as a Web resource. For instance, an IRI for the pattern `?work dbo:author dbr:Umberto_Eco.` is given below.

```
http://fragments.dbpedia.org/en?subject=&predicate=dbpedia-owl%3Aauthor&object=dbpedia%3AUmberto_
    Eco
```

All IRIs are constructed from a template, which can be chosen freely by the server implementation. Despite this flexibility, the interface ensures compatibility through *self-descriptiveness*. Metadata and hypermedia controls are embedded in every response, formalized in the Hydra vocabulary[13] (Lanthaler and Gütl, 2013). After requesting any TPF, clients can construct correct IRIs from the hypermedia description.

## 5.2 Access to archived Triple Pattern Fragments with Memento

The archive in Figure 6 adds a temporal dimension to each Linked Dataset. The current version, and each of its archive versions, can be published in a sustainable way through a Triple Pattern Fragments API. Although each individual version is thereby queryable, the relation between them is not. Without prior knowledge, clients cannot automatically navigate from one version to another using the temporal dimension.

The Memento (Van de Sompel *et al.*, 2013) framework can enable such by adding time-based versioning to the HTTP resources a Linked Data interface exposes. However, adding Memento to downloadable data dumps, Linked Data documents or SPARQL endpoints is hindered by their granularity or specification. Downloadable data dumps do not allow modifications to the HTTP layer, preventing adding extra headers. Linked data documents can support Memento, but, as mentioned before, are inadequately expressive for complex query execution. A SPARQL endpoint can answer such queries, but makes versioning hard due to the fine granularity of Web resources it exposes. Triple indexes that combine SPARQL with versioning are extremely complex and increase server cost even more.

Again, the Triple Pattern Fragments trade-off in interface granularity is beneficial. The amount of Web resources induced by triple patterns is finite and can be generated with minimal cost. This facilitates a smooth integration of the Memento protocol, in addition to low server-side cost and being sufficiently expressive for complex query execution on the client-side.

[12] http://sparqles.ai.wu.ac.at/endpoint?uri=http%3A%2F%2Fdbpedia.org%2Fsparql
[13] http://www.hydra-cg.com/spec/latest/core

Supporting Memento implies implementing the three resource types: Original Resource, TimeGate, and Memento. As shown in Figure 7, we extend the API to expose the following specific resource types:

- the **TPF** resource, which is exposes the *current* version (e.g., the most recent HDT file) and is appointed as Original Resource.
- the **LDF TimeGate** resource, which is independent of the interface (i.e., can be any LDF).
- the **TPF Memento** resource, which encapsulates an *archive* version, i.e., exposing a former representation of the TPF resource.

When requesting a TPF, a `Link` header to its corresponding LDF TimeGate is now included in the response. An LDF TimeGate accepts the `Accept-Datetime:` $t \in [t_i, t_j[$ header, which redirects the client to a TPF Memento valid at datetime $t$. This memento corresponds with a HDT file $H_{[}t_i, t_j[$ retrieved from the RDF Archive's index based on $t$. Once redirected, a client can access $H_{[}t_i, t_j[$ through the TPF Memento $URI_{[}t_i, t_j[$ it was navigated to.

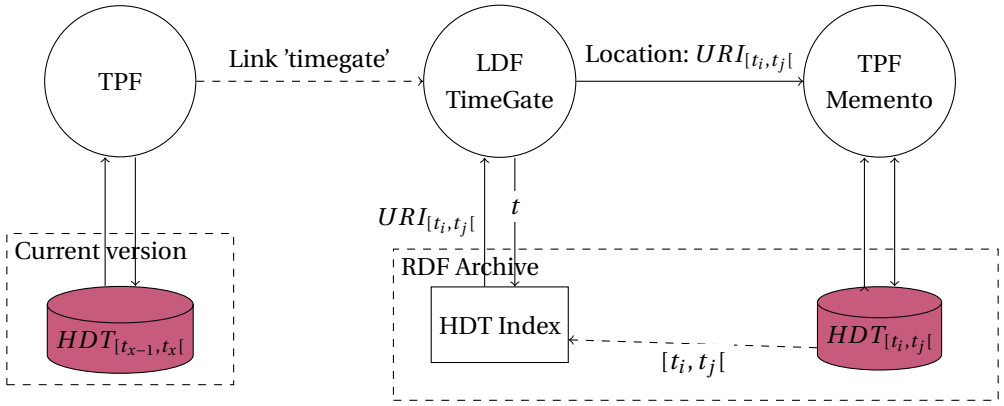



**Figure 7:** A client can transparently navigate from a TPF Resource to a TPF Memento at a specific datetime $t$.

## 5.3 Transparent time-based Web access to the DBpedia archive

We applied this Memento-enabled TPF interface to the DBpedia archive described in Section 4.2 to enable live querying historical versions of dbpedia on the public Web. This is established by two physically separate instances. The public DBpedia fragments interface on http://fragments.dbpedia.org provides access to the current DBpedia version and is the official entry point for querying clients. The archived versions of DBpedia reside on http://fragments.mementodepot.org, exposing each of the fourteen HDT files as TPF Memento resources. A `Link` header in each response from http://fragments.dbpedia.org to the DBpedia LDF TimeGate hosted at http://fragments.mementodepot.org/timegate/dbpedia connects both interfaces in a way transparent to clients.

The HTTP communication to negotiate the TPF for `?work dbo:author dbr:Umberto_Eco.` on December 1st, 2013 is illustrated in Listing 1.

```
(1)
HEAD http://fragments.dbpedia.org/en?subject=&predicate=dbpedia-owl%3Aauthor&object=dbpedia%3
    AUmberto_Eco HTTP/1.1
-----------------------------------------------------------------------
HTTP/1.1 200 OK
Link: <http://fragments.mementodepot.org/timegate/dbpedia?subject=&predicate=dbpedia-owl%3Aauthor&
    object=dbpedia%3AUmberto_Eco>;rel=timegate

(2)
HEAD http://fragments.mementodepot.org/timegate/dbpedia?subject=&predicate=dbpedia-owl%3Aauthor&
    object=dbpedia%3AUmberto_Eco HTTP/1.1
Accept-Datetime: Sun, 01 Dec 2013 22:30:00 GMT
```

```
-----------------------------------------------------------------
HTTP/1.1 302 Found
Location: http://fragments.mementodepot.org/dbpedia_3_9?subject=&predicate=dbpedia-owl%3Aauthor&
    object=dbpedia%3AUmberto_Eco

(3)
GET http://fragments.mementodepot.org/timegate/dbpedia?subject=&predicate=dbpedia-owl%3Aauthor&
    object=dbpedia%3AUmberto_Eco HTTP/1.1
Accept-Datetime: Sun, 01 Dec 2013 22:30:00 GMT
-----------------------------------------------------------------
HTTP/1.1 200 OK
Memento-Datetime: Sat, 15 Jun 2013 00:00:00 GMT # = Best matching memento
Link: <http://fragments.dbpedia.org/en?subject=&predicate=dbpedia-owl%3Aauthor&object=dbpedia%3
    AUmberto_Eco>;rel=original,
<http://fragments.mementodepot.org/timegate/dbpedia?subject=&predicate=dbpedia-owl%3Aauthor&object
    =dbpedia%3AUmberto_Eco>;rel=timegate

# Payload
...
```

**Listing 1:** Selects the Memento valid on December 1st, 2013 at 22:30:00 GMT, which is the Memento from June 15, 2013

A combined implementation of the TPF interface with Memento support and the HDT-based archived is present in the NodeJS server of Triple Pattern Fragments, available at https://github.com/LinkedDataFragments/Server.js.

# 6 Querying versioned and distributed Linked Data

In a virtual integration scenario, the consumer is responsible for physically integrating data from different publishers. Having discussed the technologies and architecture for archiving and versioning above, we now present a use case in which a query is evaluated over a federation of multiple data interfaces on the Web at several points of time in the past. The next section discusses the use case and aim, while the other sections describe how it is executed on the Web.

## 6.1 Use case and query results

As a use case, we focus our attention on the queries formulated in Section 2.2. First, we need to translate each query into SPARQL. Note that we need to include a `UNION` statement to deal with the changing schema, an issue discussed in the next section. The single-source query for the number of awards by Belgian academics becomes:

```
SELECT DISTINCT ?award
WHERE {
  ?person dcterms:subject <http://dbpedia.org/resource/Category:Belgian_academics>.
  {
    ?person <http://dbpedia.org/ontology/award> ?award.
  } UNION {
    ?person <http://dbpedia.org/property/awards> ?award.
  }
}
```

Running it for different times in the past results in the progression discussed in Figure 8.

The multi-source query for details about professors Jacques-Joseph Haus becomes:

```
SELECT ?professor ?property ?value
WHERE {
  ?professor dbpedia-owl:viafId ?viafId.
  ?professor foaf:name "Haus, Jacques-Joseph".
  ?viafId schema:sameAs ?dbpediaId.
```

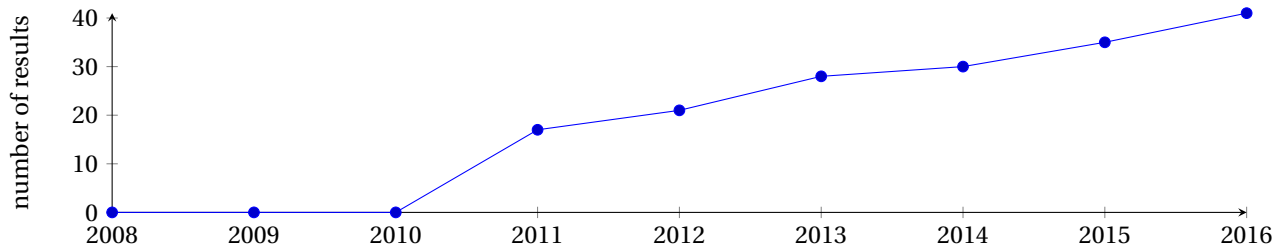



**Figure 8:** By running the same query over DBpedia from 2008 to 2016, we can detect an increase in awards won by Belgian academics.

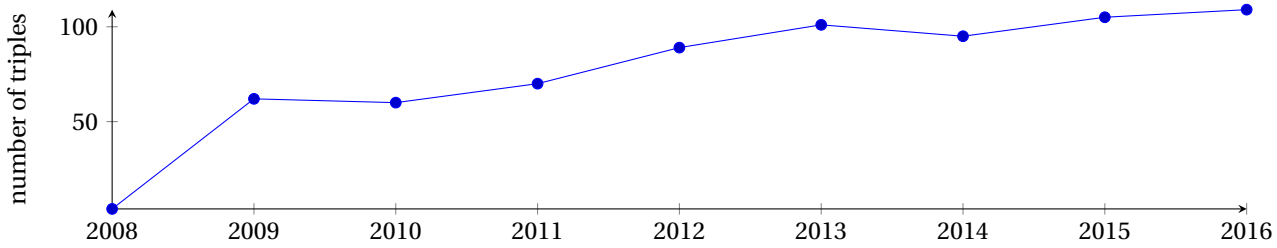



**Figure 9:** By running the same query over UGentMemorialis, VIAF and DBpedia for every year, we gain insight in the evolution of Wikipedia activity for professor Jacques-Joseph Haus.

```
  ?dbpediaId ?property ?value.
}
```

We evaluated this query at 9 timestamps in the past over UGentMemorialis, VIAF, and DBpedia. Note how every year has a different number of results.

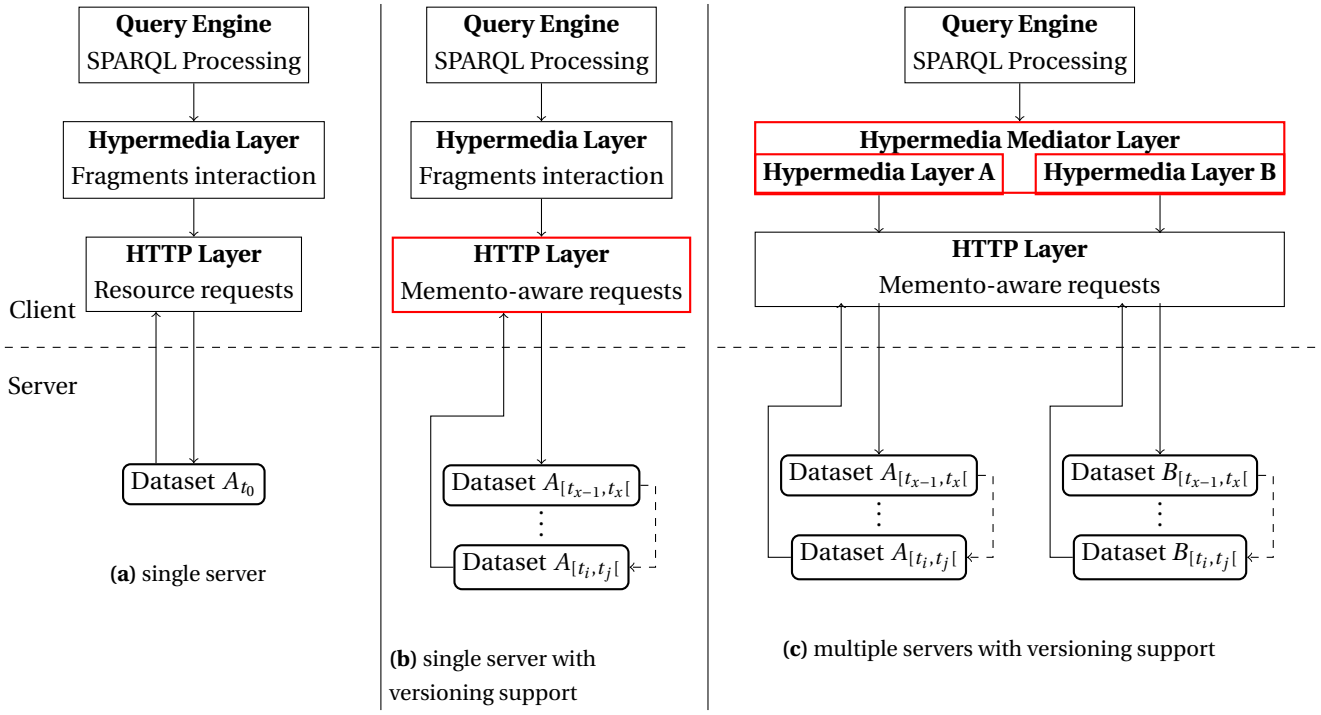
The next sections explain how these and other queries can be evaluated over live TPF interfaces on the Web, using the archive infrastructure explained in this article.

## 6.2 Querying multiple Triple Pattern Fragments interfaces over time

To bring actual *virtual integration* to consumers, a client should seamlessly query a network of distributed interfaces. Then a user can "query using a classical query language against the federated schema with an illusion that he or she is accessing a single system" (Sheth and Larson, 1990). Any SPARQL query exceeding the expressiveness of a single triple pattern forces clients of the TPF interface to break down that query into multiple requests. The client has to split the query into one or more TPF requests and produce results by combining the TPF responses locally.

In order to add support for versioning and multiple data sources to the client *without* modifying the query algorithm or its implementation, we can employ a three-tier architecture consisting of *a*) the **Query Engine**; *b*) the **Hypermedia Layer**; and *c*) the **HTTP Layer**.

The Query Engine uses an existing query algorithm for triple patterns (Verborgh *et al.*, 2016), which results in several requests to the server (Figure 10a). These results are ultimately performed by the HTTP layer, which we have modified to use the Memento protocol (Figure 10b) such that the client can choose different points in the past to evaluate the query. In between, the Hypermedia Layer is responsible for translating every request for a triple pattern into a concrete HTTP request for a TPF. By modifying this layer to consult *multiple* TPF interfaces instead (Figure 10c), the client can evaluate federated queries. This shows how we can support multiple sources over time without increasing the complexity of the query engine itself.




**(a)** single server

**(b)** single server with versioning support

**(c)** multiple servers with versioning support

**Figure 10:** The client was extended to support versioning and multiple servers, without changing the core query engine.

# 7 Challenges

This article provides the practical foundations for reviving virtual integration in LAM institutions, without synchronization issues. Thereby, they can share semantically integrated metadata at lower cost, enhancing consumption within and outside of the organization. While the proposed tool chain aids institutions to participate in the Web of Data, a number of critical challenges still need to be addressed. We discuss the most major ones to be taken up in future work below.

**Improving performance.** One of the main challenges for virtual integration is to obtain a sufficiently fast result arrivals. We stress that it is inherently difficult to impossible to achieve the same performance as with physical integration; however, the question is rather whether the achieved performance is sufficient to support the envisaged use cases, and if not, how it can be improved. As every compromise sacrifices a benefit on either the consumer side or the publisher side, finding sweet spots requires thorough research and evaluation in context of their application domain. For instance, the Triple Pattern Fragments interface balances between the expressivity of a SPARQL endpoint and the low computational cost of data dumps, at the cost of increased bandwidth and query execution time. However, in scope of this article, we observe three directions for further improvement.

First, the current client-side query algorithm can be improved to reduced the number of necessary requests, both for a single interface and multiple interfaces setup. Verborgh *et al.* (2016) introduced the first greedy approach, based on the count metadata. The client always selects the triple pattern with lowest cardinality, downloads it, and binds each resulting triple to the remaining triple patterns. This process is repeated until all variables are bound. An improvement was proposed by Van Herwegen, Verborgh, Mannens and Van de Walle (2015), by migrating decisions from local to global, thereby minimizing the number of HTTP calls. Based on multiple heuristics, the client estimates from the intermediate results whether the greedy approach is suboptimal. If so, the triple patterns are downloaded separately instead, resulting in fewer requests. However, because the join process is more complex, it requires more computational work from the client. Lastly, Acosta and Vidal (2015) introduced a query engine

that better adapts to unexpected data source conditions. A bushy tree query plan reduces intermediate results and, thus, the amount of HTTP requests. Furthermore, the query execution schedule is adaptive to live conditions, such as interface response times.

In addition to these existing methods, many optimization techniques from relational databases (Halevy, 2001), distributed databases (Galárraga *et al.*, 2014), or RDF databases (Schmidt *et al.*, 2010) are directly applicable. However, their effectiveness is determined by the live Web environment and the limited expressiveness of the interface. Literature closest to those circumstances are in the field of federated SPARQL querying (Rakhmawati *et al.*, 2013; Saleem *et al.*, 2016)

Next, the server-side interface can be equipped with additional interface features beyond TPF. These can vary per interface, and the client-side query algorithm can be adjusted to use them independently or in conjunction. This amounts to an exploration of the LDF axis, by influencing: *a*) the possible selectors, i.e., allowing more complex questions, or *b*) the offered metadata, i.e., including more information useful to the client. Previous work has explored both dimensions, such as facilitating textual searches (Van Herwegen, De Vocht, Verborgh, Mannens and Van de Walle, 2015) or moving certain types of joins from clients to servers (Hartig and Buil-Aranda, 2016). Extra metadata that benefits the final stage of query processing was shown to reduce bandwidth (Vander Sande *et al.*, 2015).

Finally, for federated queries, a more sophisticated mechanism for source selection can eliminate several of them beforehand for (parts of) a given query, reducing the number of needed HTTP requests. Source selection is known to be an important performance factor (Saleem *et al.*, 2016). Applying this process involves (client-side or server-side) pre-processing, which in the case of multiple versions per dataset means that extra processing per version would be required. It remains to be investigated to what extent this extra effort, if performed by the server, could reduce query times on the client.

**Preserving data semantics between archives.** In this article, we have demonstrated how a single SPARQL query can be executed at two different points in time. From a technical perspective, we can deduce the evolution of the results by comparing resultsets. However, it is important to note that this is insufficient to correctly interpret the semantics of this change. If an RDF description evolves between two versions of the same dataset, its basis is unknown. For instance, several scenarios could be in effect to explain the sudden presence of a fact: it was added, because it was not known before; it was known before, but it was missing; or it is replaces a previously present, but incorrect fact. In limited cases, it might be safe to make a few assumptions (e.g., when a date changes for a stable subject and predicate), but the semantics can never be deduced with full certainty without additional knowledge. Thus, this indicates the importance of *accessible* provenance information explaining made changes and providing the missing link between prior versions. Coppens *et al.* (2011) proposed a Memento-compatible approach for this issue.

**Live updating the archive.** Commonly, the *current* version of a dataset approximates the *live* version. As a result, adjusting the generation frequency of new shapshots to the live versions update speed, is an open challenge. Many archiving scenarios are served by the presented HDT-based storage layer, but its update frequency is limited. For realistic datasets sizes, the average HDT generation time is expressed in matter of hours. Therefore, scenarios with live updates, i.e., an update frequency below a couple of seconds, require new storage layers that handle small incremental changes to RDF data better. Research to combine such fine-grained RDF history with performant triple pattern access is still in its infancy.

**Improving usability for deployers and data consumers.** Finally, the challenge of *usability* involves packaging the technical solution into an easily accessible unit. While the current solution meets all technical goals outlined in this article, the more accessible it becomes, the higher the chances of adoption. On the one hand, usability improvements should focus on reducing the time and effort for data publishers to bring their data online, as well as version management. On the other hand, query execution should be facilitated for data consumers. This includes support for writing queries, which can become complicated—especially when multiple sources with different vocabularies

are consulted. For added ease of use, more simple query languages should be supported by the client, including visual languages that allow users to create queries through drag and drop.

Another issue related to usability, is the lack of clear schema evolution management for Linked Open Datasets. For instance, common knowledge bases DBpedia and VIAF have both introduced significant changes to the schema over time, aggravating the reproducability problem and contributing to the *vocabulary chaos* mentioned in the introduction. In the continuing effort to expose LAM information as Linked Data to increase visiblity (Miller and Ogbuji, 2015), Dunsire *et al.* (2012) urge to double efforts on supporting infrastructure, along with "guiding principles and best practices around reuse, extension of existing vocabularies, as well as development of new vocabularies". Of course, semantic changes are inevitable; however, it is the deprecation processes — ensuring backwards compatibility and communicating version changes — that badly require clear planning and interoperability. Furthermore, in line with the infrastructure presented in this article, application developers indicate the desire for "machine-readable, API-based access to version history" in vocabulary registries as well (Dunsire *et al.*, 2012).

# 8 Conclusion

This article addressed the issue of *reproducibility* when querying distributed Linked Data sources. As Web content drifts over time, query results need to remain valid when they are regenerated later. In that respect, Libraries, Archives and Museums (LAMs), in particular the increasing number of underresourced institutions, struggle to provide a single queryable point of access to their digital preservation efforts. Hence, this article proposed an alternative publishing strategy that lowers maintenance cost for publishers, and covers drifting sources. These claims are supported by a memory reconstruction Use Case based on three complimentary, but distributed Linked datasets (DBpedia, VIAF, and UGentMemorialis).

We proposed a shift to a *virtual integration* approach to consolidate these *silos of the LAMs*, i.e., composing a consumer view over distributed datasets that remain in control of the institutions, in favor of *physical integration*, i.e., metadata aggregation by a central party. The tendency to conflict on a policy level (e.g., challenging metadata custody and control), synchronization problems, and high infrastructural costs make physical integration particularly troublesome. Virtual integration is an attractive alternative if technological advances can be made that ameliorate problems related to data source selection, uniform access for clients, and maintenance costs. Therefore, we argued that meaningful steps can be taken that lead to a better *practical* solution for LAM institutions even though at this point not all drawbacks can be addressed.

Considering that most institutions update a "live" RDF database continuously through an Extract-Tranform-Load process, publishing recurrently extracted snapshots through a low-cost wrapper stack, avoids the high maintenance cost of exposing the "live" database directly. For cases where snapshots are created at a medium pace (~ daily), the Header-Dictionary-Triples (HDT) format is an excellent storage candidate, as it creates very compact immutable files that are queryable. For DBpedia, this implies an average size gain of 87% compared to the Ntriples format. In combination with the file-system, HDT can serve as pragmatic, but extremely useful RDF archive, which serves basic digital preservation needs with little resources. Cases with higher update frequencies (< hourly) are momentarily dependent on the ongoing research in RDF archiving systems.

Publishing these snapshots with the Triple Pattern Fragments (TPF) interface in combination with Memento, the HTTP datetime negotiation framework, enables complex queries over current and archived versions of a dataset. Potentially, we can gain insight on the evolution of facts in Linked Data by executing the same query at different comparing time points. Without essential provenance metadata about the modifications between versions, however, such insights are not yet reliable. Yet, the same functionality can be used to synchronize distributed sources on the Web, when querying multiple interfaces at once.

## References


Acosta, M. and Vidal, M.-E. (2015), Networks of linked data eddies: An adaptive web query processing engine for rdf data, in *International Semantic Web Conference*, Springer International Publishing, pp. 111–127.

Beek, W., Rietveld, L., Bazoobandi, H. R., Wielemaker, J. and Schlobach, S. (2014), Lod laundromat: a uniform way of publishing other people's dirty data, in *International Semantic Web Conference*, Springer, pp. 213–228.

Bianchini, C. and Willer, M. (2014), "Isbd resource and its description in the context of the semantic web", *Cataloging & Classification Quarterly*, Vol. 52, Taylor & Francis, pp. 869–887.

Binding, C., Charno, M., Jeffrey, S., May, K. and Tudhope, D. (2015), "Template Based Semantic Integration:", *International Journal on Semantic Web and Information Systems*, Vol. 11, IGI Global, pp. 1–29.
**URL:** *http://www.igi-global.com/article/template-based-semantic-integration/135560*

Bizer, C., Heath, T. and Berners-Lee, T. (2009), "Linked data-the story so far", *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pp. 205–227.

Bourdon, F. and Boulet, V. (2013), Viaf: A hub for a multilingual access to varied collections, in *World library and information Congres: 78th IFLA general Conference and Assembly*.

Buil-Aranda, C., Hogan, A., Umbrich, J. and Vandenbussche, P.-Y. (2013), Sparql web-querying infrastructure: Ready for action?, in *International Semantic Web Conference*, Springer, pp. 277–293.

Clough, G. W. (2013), *Best of Both Worlds: Museums, Libraries, and Archives in the Digital Age*, Smithsonian Institution, chapter 1, pp. 9–10.

Coppens, S., Mannens, E., Van Deursen, D., Hochstenbach, P., Janssens, B. and Van de Walle, R. (2011), Publishing provenance information on the web using the memento datetime content negotiation, in *WWW2011 workshop on Linked Data on the Web (LDOW 2011)*, Vol. 813, pp. 6–15.

Corrado, E. M. and Moulaison, H. L. (2014), *Digital preservation for libraries, archives, and museums*, Rowman & Littlefield.

Cyganiak, R., Wood, D. and Lanthaler, M. (2014), RDF 1.1 concepts and abstract syntax, Recommendation, World Wide Web Consortium.
**URL:** *https://www.w3.org/TR/rdf11-concepts/*

de Sompel, H. V., Sanderson, R., Nelson, M. L., Balakireva, L., Shankar, H. and Ainsworth, S. (2010), "An http-based versioning mechanism for linked data", *CoRR*, Vol. abs/1003.3661.
**URL:** *http://arxiv.org/abs/1003.3661*

Dunsire, G., Harper, C., Hillmann, D. and Phipps, J. (2012), "Linked data vocabulary management: infrastructure support, data integration, and interoperability", *Information Standards Quarterly*, Vol. 24, pp. 4–13.

Erik, T. *et al.* (2015), "The evolving direction of LD research and practice", *Library Technology Reports*, Vol. 52, pp. 29–33.

Fernández, J. D., Martínez-Prieto, M. A., Gutiérrez, C., Polleres, A. and Arias, M. (2013), "Binary RDF representation for publication and exchange (HDT)", *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 19, Elsevier, pp. 22–41.

Fernández, J. D., Schneider, P. and Umbrich, J. (2015), The DBpedia wayback machine, in *Proceedings of the 11th International Conference on Semantic Systems*, SEMANTICS '15, ACM, New York, NY, USA, pp. 192–195.
**URL:** *http://doi.acm.org/10.1145/2814864.2814889*

Galárraga, L., Hose, K. and Schenkel, R. (2014), Partout: a distributed engine for efficient RDF processing, in *Proceedings of the 23rd International Conference on World Wide Web*, ACM, pp. 267–268.

Halevy, A. Y. (2001), "Answering queries using views: A survey", *The VLDB Journal*, Vol. 10, Springer, pp. 270–294.

Harris, S., Seaborne, A. and Prud'hommeaux, E. (2013), "SPARQL 1.1 query language", *W3C Recommendation*, Vol. 21.

Hartig, O. and Buil-Aranda, C. (2016), *Bindings-Restricted Triple Pattern Fragments*, Springer International Publishing, Cham, pp. 762–779.

Hausenblas, M. (2009), "Exploiting linked data to build web applications", *IEEE Internet Computing*, Vol. 13, IEEE Computer Society, p. 68.

Heath, T. and Bizer, C. (2011), "Linked data: Evolving the web into a global data space", *Synthesis lectures on the semantic web: theory and technology*, Vol. 1, Morgan & Claypool Publishers, pp. 1–136.

Hedstrom, M. L. and Montgomery, S. (1998), *Digital preservation needs and requirements in RLG member institutions*, Research Libraries Group Mountain View, Calif.

Houghton, B. (2016), "Preservation Challenges in the Digital Age", *D-Lib Magazine*, Vol. 22, Corporation for National Research Initiatives, p. 1.
**URL:** *http://dlib.org/dlib/july16/houghton/07houghton.html*

Isaac, A. and Haslhofer, B. (2013), "Europeana linked open data – data.europeana.eu", *Semantic Web*, Vol. 4, IOS Press, pp. 291–297.

Klein, M., Sanderson, R., de Sompel, H. V. and Nelson, M. L. (2014), "Real-time notification for resource synchronization", *CoRR*, Vol. abs/1402.3305.
**URL:** *http://arxiv.org/abs/1402.3305*

Koehler, W. (2002), "Web page change and persistence—a four-year longitudinal study", *Journal of the American Society for Information Science and Technology*, Vol. 53, Wiley Online Library, pp. 162–171.

Kroeger, A. (2013), "The road to BIBFRAME: the evolution of the idea of bibliographic transition into a post-marc future", *Cataloging & classification quarterly*, Vol. 51, Taylor & Francis, pp. 873–890.

Lanthaler, M. and Gütl, C. (2013), "Hydra: A vocabulary for hypermedia-driven web apis.", *LDOW*, Vol. 996.

Leddy, C. (2012), "Linking libraries, museums, archives", *Harvard gazette*, pp. 2–3.

Mak, L., Higgins, D., Collie, A. and Nicholson, S. (2014), "Enabling and integrating ETD repositories through linked data", *Library Management*, Vol. 35, pp. 284–292.
**URL:** *http://dx.doi.org/10.1108/LM-08-2013-0075*

Marden, J., Li-Madeo, C., Whysel, N. and Edelstein, J. (2013), Linked open data for cultural heritage: Evolution of an information technology, in *Proceedings of the 31st ACM International Conference on Design of Communication*, SIGDOC '13, ACM, New York, NY, USA, pp. 107–112.
**URL:** *http://doi.acm.org/10.1145/2507065.2507103*

Martínez-Prieto, M. A., Cuesta, C. E., Arias, M. and Fernández, J. D. (2015), "The solid architecture for real-time management of big semantic data", *Future Generation Computer Systems*, Vol. 47, Elsevier, pp. 62–79.

Meinhardt, P., Knuth, M. and Sack, H. (2015), TailR: a platform for preserving history on the web of data, in *Proceedings of the 11th International Conference on Semantic Systems*, SEMANTICS '15, ACM, New York, NY, USA, pp. 57–64.
**URL:** *http://doi.acm.org/10.1145/2814864.2814875*

Miller, E. and Ogbuji, U. (2015), "Linked data design for the visible library", *Bulletin of the Association for Information Science and Technology*, Vol. 41, pp. 23–29.
**URL:** *http://dx.doi.org/10.1002/bult.2015.1720410409*

Mitchell, E. T. (2013), *Library linked data: Research and adoption*, American Library Association.

Mitchell, E. T. (2015), "The current state of linked data in libraries, archives, and museums", *Library Technology Reports*, Vol. 52, pp. 5–13.

Möller, K., Hausenblas, M., Cyganiak, R., Grimnes, G. and Handschuh, S. (2010), Learning from linked open data usage: Patterns & metrics, in *WebSci10: Extending the Frontiers of Society On-Line*.

Ntoulas, A., Cho, J. and Olston, C. (2004), What's new on the web?: The evolution of the web from a search engine perspective, in *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, ACM, New York, NY, USA, pp. 1–12.
**URL:** *http://doi.acm.org/10.1145/988672.988674*

Papastefanatos, G. and Imis, R. C. A. (2014), "Challenges and Opportunities in the Evolving Data Web", pp. 23–28.

Rakhmawati, N. A., Umbrich, J., Karnstedt, M., Hasnain, A. and Hausenblas, M. (2013), "Querying over federated sparql endpoints—a state of the art survey", *arXiv preprint arXiv:1306.1723*.

Rietveld, L., Beek, W. and Schlobach, S. (2015), LOD lab: Experiments at LOD scale, in *International Semantic Web Conference*, Springer, pp. 339–355.

Rinehart, A. K., Prud'homme, P.-A. and Huot, A. (2014), "Overwhelmed to action: digital preservation challenges at the under-resourced institution", *OCLC Systems & Services: International Digital Library Perspectives*, Vol. 30, pp. 28–42.
**URL:** *http://search.proquest.com/docview/1531921571?accountid=12339%5Cnhttp://mcgill.on.worldcat.org/atoztitles/link?sid=ProQ:&*
*01-01&at*

Ross, S. (2012), "Digital preservation, archival science and methodological foundations for digital libraries", *New Review of Information Networking*, Vol. 17, Taylor & Francis, pp. 43–68.

Saleem, M., Khan, Y., Hasnain, A., Ermilov, I. and Ngonga Ngomo, A.-C. (2016), "A fine-grained evaluation of sparql endpoint federation systems", *Semantic Web Journal*, Vol. 7, IOS Press, pp. 493–518.

Schmachtenberg, M., Bizer, C. and Paulheim, H. (2014), Adoption of the linked data best practices in different topical domains, in *International Semantic Web Conference*, Springer, pp. 245–260.

Schmidt, M., Meier, M. and Lausen, G. (2010), Foundations of sparql query optimization, in *Proceedings of the 13th International Conference on Database Theory*, ACM, pp. 4–33.

Schumacher, J., Thomas, L. M., Vandecreek, D., Erdman, S., Hancks, J., Haykal, A., Miner, M., Prud 'homme, P.-A. and Spalenka, D. (2014), "From Theory to Action: " Good Enough " Digital Preservation Solutions for Under-Resourced Cultural Heritage Institutions A Digital POWRR White Paper for the Institute of Museum and Library Services", *Institute of Museum and Library Services*.
**URL:** *http://powrr-wiki.lib.niu.edu/images/a/a5/FromTheoryToAction_POWRR_WhitePaper.pdf%5Cnhttp://commons.lib.niu.edu/han*

Sheth, A. P. and Larson, J. A. (1990), "Federated database systems for managing distributed, heterogeneous, and autonomous databases", *ACM Computing Surveys (CSUR)*, Vol. 22, ACM, pp. 183–236.

Smith-Yoshimura, K. (2014), 'Linked Data Survey results 1 – Who's doing it'.
**URL:** *http://hangingtogether.org/?p=4137*

Van de Sompel, H., Nelson, M. and Sanderson, R. (2013), HTTP framework for time-based access to resource states – Memento, Request For Comments 7089, Internet Engineering Task Force.
**URL:** *https://tools.ietf.org/rfc/rfc7089*

Van Herwegen, J., De Vocht, L., Verborgh, R., Mannens, E. and Van de Walle, R. (2015), Substring filtering for low-cost Linked Data interfaces, in Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayan, K. and Staab, S. (Eds.), *The Semantic Web – ISWC 2015*, Vol. 9366 of *Lecture Notes in Computer Science*, Springer, pp. 128–143.
**URL:** *http://linkeddatafragments.org/publications/iswc2015-substring.pdf*

Van Herwegen, J., Verborgh, R., Mannens, E. and Van de Walle, R. (2015), Query execution optimization for clients of triple pattern fragments, in *European Semantic Web Conference*, Springer International Publishing, pp. 302–318.

van Hooland, S. and Verborgh, R. (2014), *Linked Data for Libraries, Archives and Museums*, Facet Publishing.

Vander Sande, M., Coppens, S., Verborgh, R., Mannens, E. and Van de Walle, R. (2013), Adding time to Linked Data: a generic Memento proxy through PROV, in *Proceedings of the 2013th International Conference on Posters & Demonstrations Track-Volume 1035*, CEUR-WS.org, pp. 217–220.

Vander Sande, M., Verborgh, R., Van Herwegen, J., Mannens, E. and Van de Walle, R. (2015), Opportunistic Linked Data querying through approximate membership metadata, in Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayan, K. and Staab, S. (Eds.), *The Semantic Web – ISWC 2015*, Vol. 9366 of *Lecture Notes in Computer Science*, Springer, pp. 92–110.
**URL:** *http://linkeddatafragments.org/publications/iswc2015-amf.pdf*

Verborgh, R., Hartig, O., De Meester, B., Haesendonck, G., De Vocht, L., Vander Sande, M., Cyganiak, R., Colpaert, P., Mannens, E. and Van de Walle, R. (2014), Querying datasets on the Web with high availability, in Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C., Vrandečić, D., Groth, P., Noy, N., Janowicz, K. and Goble, C. (Eds.), *Proceedings of the 13th International Semantic Web Conference*, Vol. 8796 of *Lecture Notes in Computer Science*, Springer, pp. 180–196.
**URL:** *http://linkeddatafragments.org/publications/iswc2014.pdf*

Verborgh, R., Mannens, E. and Van de Walle, R. (2015), Initial usage analysis of DBpedia's Triple Pattern Fragments, in *Proceedings of the 5th USEWOD Workshop on Usage Analysis and the Web of Data*.
**URL:** *http://linkeddatafragments.org/publications/usewod2015.pdf*

Verborgh, R., Vander Sande, M., Hartig, O., Van Herwegen, J., De Vocht, L., De Meester, B., Haesendonck, G. and Colpaert, P. (2016), "Triple pattern fragments: A low-cost knowledge graph interface for the web", *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 37, Elsevier, pp. 184–206.

Verbruggen, C. and Deneckere, G. (n.d.), 'UGentMemorialis. biographical data of UGent professors between 1817 and 2012 [dataset]'.
**URL:** *www.UGentMemorialis.be*

Waibel, G. and Erway, R. (2009), "Think globally, act locally: Library, archive, and museum collaboration", *Museum Management and Curatorship*, Vol. 24, Taylor & Francis, pp. 323–335.

Yoshimura, K. S. (2016), "Analysis of international linked data survey for implementers", *D-Lib Magazine*, Vol. 22, Corporation for National Research Initiatives, p. 6.

Zorich, D. M., Waibel, G., Erway, R., Zorich, D. M., Waibel, G., Erway, R., Programs, O. and Waibel, G. (2008), 'Beyond the silos of the LAMs: collaboration among libraries, archives and museums'.