Linked Data and Live Querying for Enabling Support Platforms for Web Dataspaces

Jürgen Umbrich[1], Marcel Karnstedt[1], Josiane Xavier Parreira[1], Axel Polleres[2], Manfred Hauswirth[1]

[1]*Digital Enterprise Research Institute, National University of Ireland, Galway, Ireland*
[2]*Siemens AG Österreich, Siemensstraße 90, 1210 Vienna, Austria*
{firstname.lastname}@[1]deri.org/[2]siemens.com

***Abstract*— Enabling the "Web of Data" has recently gained increased attention, particularly driven by the success of Linked Data. The agreed need for technologies from the database domain is therein often referred to as the "Web as a Database", a concept that is still more a vision than a reality. Meanwhile, the database community proposed the notion of dataspaces managed by support platforms, as an alternative view on the data management problem for small-scale, loosely connected environments of heterogenous data sources.**

The Web of Data can actually be seen as a collection of inter-connected dataspaces. In this work, we propose a combination of Linked Data and database technologies to provide support platforms for these Web dataspaces. We argue that while separated, Linked Data still lacks database technology and the dataspace idea lacks openness and scale. We put particular focus on the challenge of how to index, search and query structured data on the Web in a way that is appropriate for its dynamic, heterogeneous, loosely connected, and open character. Based on an empirical study, we argue that none of the two extremes on its own – centralised repositories vs. on-demand distributed querying – can meet all requirements. We propose and discuss an alternative hybrid approach combining the best of both sides to find a better tradeoff between result freshness and fast query response times.

I. Introduction

The World Wide Web is the largest collection of information created by mankind and bears potential for unthought-of applications. As of today, its potential still cannot be fully explored because current Web technologies mostly rely on manual efforts for integrating the huge amount of heterogenous, loosely interrelated, and highly dynamic Web data. To address this problem, the Semantic Web community proposes a transition from the Web of documents to the "Web of Data", combining raw data with data describing its semantics [23]. This enables an integrated view on the information of the World Wide Web for machines and humans alike. The required annotation and linkage approach is supported by the linked data initiative, a core building block of the Semantic Web [2], [3]. While this movement recently gained a wealth of attention and clearly marked its success, the often demanded idea of the Web as the largest existing "heterogeneous distributed database" [12] is far from reality, mainly due to the lack of a wide range of technology and functionality established by the database community.

Meanwhile, the notion of dataspaces was coined by Franklin, Halevy, and Maier [10] as a new research agenda for the database community. This notion argues for the need of a different view on data management problems in the context of enterprise data from a range of heterogenous and loosely connected sources. They propose the idea of a dataspace support platform (DSSP) as a key element of a dataspace, hiding all the data management complexity behind a range of services for search, monitoring, integration, etc.

Up to now, much of the work from both directions has been carried out independently from each other, though following very similar objectives and applying similar approaches. Recently, Heath and Bizer [13] compared the Web of Data to a global dataspace. The Linked Data guidelines indeed provide abstractions and technologies to publish, access and process linked data – which approaches some of the challenges discussed in [10]. Thanks to the Linked Data initiative, data from different domains is already accessible in (to some extent) integrated dataspaces, such as government data [8] and the different domains covered by the Linking Open Data cloud [4]. Moreover, links between these different dataspaces already exist and new links are continuously added. However, [13] misses to address how a support platform for the Web of Data can be accomplished. We extend their view by understanding the global dataspace as a combination of smaller interlinked Web dataspaces. We point out that Linked Data still lacks in the support of important components like cost-based query processing supporting efficiency and update guarantees alike. On the other hand, the characteristics of the Web introduce a new range of challenges that are not present in the controlled, small-scale enterprise environments, for which dataspaces were originally designed.

We propose our vision of enabling support platforms for Web dataspaces via a combination of Linked Data and database technologies. In Section II, we show that Linked Data already provides an initial set of required services, including a unified data model for pay-as-you-go integration, approaches for the discovery of relationships, standardised access methods, and a common query language that can be mapped to other languages and data models. Section III discusses why the Web of Data should be aligned to the principles of dataspaces and DSSPs and what requirements and modifications of the original concepts this requires. We provide an overview of the challenges we identified, before we focus particularly on the question of how to implement efficient query processing functionality that can cope with the dynamic and open character of the Web in Section IV. We present empirical results that show that we cannot rely on only central repositories materialising Web data, as in this case we are not able to cope with the observed dynamics. Neither can we rely on only querying Web data sources live, as this does

not provide nearly sufficient answer times. Instead, we propose to design and develop a hybrid query processing mechanism, which combines offline data repositories and online data sources based on an appropriate cost model that incorporates expected answer times as well as freshness of results. Last but not least, we also argue for the feasibility of the proposed approach by overviewing our recent work on the first building blocks of the resulting architecture.

II. Linked Data and Dataspaces

The Linked Data initiative aims at enabling the Web of Data – a single global network of human and machine readable information, based on the core Linked Data principles [3]. Linked Data standards and technologies resemble some of the services proposed for a DSSP. In fact, the Web of Data can be seen as a collection of interlinked Web dataspaces, in which support platforms can be achieved via combining Linked Data and database technologies. To illustrate this, Figure 1 gives an example of such a Web dataspace and the components of a support platform, aligned to the original dataspace concept in [10].

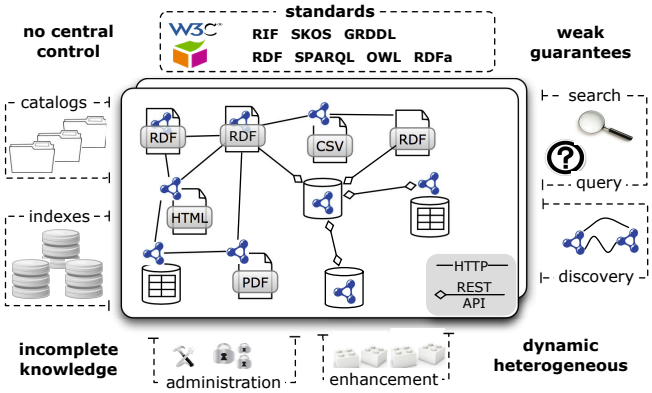

Fig. 1. Example of a Web dataspace and components of a support platform.

The figure shows some of the originally proposed dataspace features, such as interlinked heterogenous data sources as dataspace participants, surrounded by a set of services that hide the complexity of data integration and management from the data consumers and publishers. In contrast, data integration and access is driven and eased by standardisations (W3C), there is a lack of central control, and available knowledge is usually incomplete with respect to data dynamics.

We base our argument that Linked Data partially implements some of the concepts of support platforms for Web dataspaces on the following observations:

1) At the core of dataspaces are participants (data sources with different formats and different processing capabilities) and relationships. Linked Data is built around the same concepts, although they are called resources (documents, database endpoints, Web services, etc.) and links.
2) While [10] proposes the usage of XML for interchanging data, Linked Data relies on RDF [22] for this purpose, which offers even more advantages. RDF is not tied to a specific schema, it is self-describing, and allows a mix of structured and semi-structured data. It acts as the least common denominator between different data models, formats, and views, i.e., supporting the DSSP idea of being able to "query everything". This is further supported by a wide range of tools and wrappers for transforming and accessing various common data models into RDF (cf. Figure 1).
3) Describing the relationships between participants as explicit RDF triples serves the integration aspect of dataspaces by materialising "hard-wired" join structures, but also enables on-the-fly integration, e.g., by reasoning techniques that rely on query rewriting [15].
4) As required for a support platform, a standardised access method (based on HTTP) and a common query language (SPARQL [18]) exist for Linked Data. As for the actual data model, mappings between alternative query languages and SPARQL (e.g., W3Cs RDB2RDF [5]) exist.
5) URIs serve as global keys and support the integration of different sources and the identification of inconsistencies.
6) The discovery component of a support platform offers services to locate participants and data and to create and refine relationships among them. For Linked Data, resources can be discovered by Web crawlers and sources can acknowledge their existence and updates with so called "ping services", e.g. as used by major search engines. Link traversal and reasoning support the automatic identification and inference of links (i.e. relations) [14].
7) Integration of different Web dataspaces is further encouraged and eased by techniques and tools for entity (a real-world object identified by a URI) recognition and by the provision of (optional) vocabularies and ontologies [16].
8) As required for dataspaces Linked Data can be streamed [17] or stored, enabling, for instance, the integration of sensor data.

III. Open Challenges

This list above illustrates that the tools, techniques, guidelines and principles of Linked Data represent a major step towards enabling support platforms for Web dataspaces. However, while the Linked Data community successfully approaches the scale and openness of the Web, several, particularly database oriented, services of support platforms are still missing. In this section, we give a short overview of the main open challenges in this context and why the underlying concepts are essential to make the Web of Data a real success.

a) Graph-Based Data Model: RDF marks its impact for solving the problems of data integration and exchange of information, a fundamental requirement for overcoming disconnected data silos. The RDF data model is a form of a highly-normalised relational model with binary relations between global unique identifiers. However, without a deeper understanding of its theoretical foundations we will not be able to exploit the full potential of the Web of Data. For instance, without the identification of reoccurring structures and motifs, as well as the support of efficient graph traversals, indexing and query processing approaches will hardly achieve the scalability and efficiency required for the scale of the Web. Several research fields are approaching these issues. The logics and reasoning communities focus on the theoretical aspects,

several recent data-mining efforts focus on graph analysis, and novel graph databases provide a good starting point for efficient graph management and processing. These directions, traditionally related to data management, have to be enriched by other research directions, such as behavioural analysis to help coping with the openness of the Web.

b) Search and Query: A fundamental requirement that has to be adopted from dataspaces is the urgent need for combining structured queries with unstructured (e.g., keyword-based) search. Without search functionality users will not be able to efficiently explore and discover the knowledge available. Beyond that, only structured querying will allow users to perform complex data analytics. By today, none of the established search engines for Linked Data offers efficient structured and meta-data queries, while SPARQL endpoints (HTTP interfaces for remotely posting SPARQL queries to RDF repositories) often lack in the support of efficient keyword-based search. Luckily, first approaches to support combined search and query are being discussed, designed and expectedly advanced and further developed.

Apart from that, in a data collection as huge as the Web, users will usually be overwhelmed by the vast amount of somewhat relevant results. Modern Web technologies show that ranking is mandatory in such a situation. Some of the most prominent ranking algorithms for the Web have been adapted for the Web of Data [6], but they only partially explore the richness of relationships embedded in RDF links. Further, there is the need for supporting ranking at different levels (domains, resources, etc) and for incorporating trust and personalisation. All these requirements are already researched by the information retrieval, Web, and database communities, but to an extent that has to be leveraged. This also holds for the crucial requirement of supporting additional "meta" information with query results. For instance, provenance (or lineage) and trustworthiness are essential for assessing quality of data and query results, a feature that is mandatory for being able to handle amounts of data as large and diverse as produced in the Web. Similarly, approaches for identifying inconsistencies, object consolidation and entity resolution represent a good starting point, but have to be advanced regarding the scalability of actually fixing encountered problems. These areas can strongly benefit from fundamental database research.

c) Guarantees: Guarantees are essential for dataspaces, for enterprise data and Web data alike. If there is no assurance that the recent data is received, that updates do not get lost, that results are complete (with respect to the currently available data), no meaningful interchange and interaction between the participants in the Web will ever get into place. However, considering the character of the Web, we cannot achieve full guarantees but have to aim for loosened features like eventual consistency [9]. Issues like access control & policies and the data sovereignty of the publishers are already intensely discussed, but a full assessment of available guarantees is mostly missing.

IV. The Need for Fundamental Changes in Query Processing and Indexing

So far we have discussed the advantages of seeing the Web of Data as a Web dataspace, and have shown how Linked Data already provides some of the required services for a support platform. We have highlighted what is still missing and what are the main difference compared to the original dataspace concept. One core difference concerns the data catalog, a key component for the success of a support platform. In [10] the catalog contains all information about participants and the relationships among them, which are required for the query planning and processing. While efficient query processing in a Web dataspace will also depend on catalogs, the size, the dynamics, and the open character of the Web, makes it impossible to have a complete source of knowledge at one central point.

Currently, Linked Data addresses the catalog problem with vocabularies to describe the attributes of a resource or a collection of resources. Web repositories (e.g., The data hub, formally know as CKAN [1]) are used to collect these meta-data descriptions. While these catalogs support browsing of meta-data, they still lack completeness and freshness guarantees. Moreover, efficient and scalable means to combine and interlink the available data catalogs are still missing.

A data catalog relies on other services, like storing and indexing facilities, to support tasks such as data caching for query optimisation and to guarantee data availability and recovery. Traditional indexing approaches that materialise data cannot keep up with the growth rate and dynamics of the Web resources. Current indexes and search engines, even those designed for the Web of Data cannot offer efficient execution of more complex structured queries. Moreover, as we will demonstrate shortly, they fail to provide fresh and up-to-date results in most of the situations, mainly due to the low update rate of their indexes. Distributed query processing techniques, initially designed to handle a small number of large and stable data repositories, can also not cope with the millions of rather small and dynamic data sources on the Web of Data.

In an attempt to address these issues, recent work has addressed the possibilities and limitations of executing a query "live", i.e. directly over the Web data, by applying crawling techniques and explorative querying. While these live query approaches guarantee up-to-date results, their execution times are usually in the range of seconds, even for simple queries. Moreover, they consider only documents, while a combination of documents and available SPARQL endpoints is necessary.

We envision a system that offers a hybrid query processing mechanism, which combines offline (static) and online (dynamic) query processing over documents and endpoints, in order to achieve the best trade-off between performance, completeness, and freshness. The query engine will consider the combined information from several available repositories (comprising materialised data copies, caches and indexes), centralised and decentralised ones. While they will be useful to execute the more stable parts of a query, they cannot fully guarantee freshness and completeness of the more dynamic parts. For those, the results will be fetched directly from the data sources. Next we show the need of such hybrid query model, by giving experimental evidence that results from data repositories are often incomplete and outdated, and that "pure live" query processing is very expensive. Later we present our contribution in designing such a hybrid query model.

A. Index Consistency Study

Centralised indexes that materialise (or cache) Linked Data provide efficient data access, but to guarantee a consistent and fresh view of the indexed data it would require constant updates, a very challenging task due to the size and dynamic nature of the Web data. Therefore, indexes for Linked Data in the Web are often incomplete and outdated.

To verify this assumption we performed an experiment which examine the content provided by two major public Linked Data Web indexes, namely the SPARQL endpoint of "the Semantic Web Index" Sindice[1] and the endpoint powered by OpenLink[2]. The former contains over hundreds of millions Web pages with structured data, the latter is the major Linked Open Data (LOD) SPARQL endpoint hosting all officially registered LOD datasets, as well as other data sources.

Our experiment compares the available information for a set of entities returned by one of the indexes against the available information returned by performing a `HTTP Get` operation on that particular entity URI.

For each entity URI in the set we have performed two operations: i) the execution of an atomic query against one of the two Web indexes to collect all the information about that particular entity returned by the index, and ii) the execution of a `HTTP Get` operation for that entity URI to retrieve all of the direct accessible entity information from the Web. Both operations return a set of RDF triples, denoted as *SPARQL endpoint results* ($\mathcal{S}$) and *Web results* ($\mathcal{W}$). Figure 2 depicts both result sets and the possible relations between them.

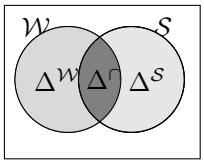

Fig. 2. Result set (Venn) diagram.

The set $\Delta^{\mathcal{W}} := \mathcal{W} \setminus \mathcal{S}$ contains only information about entity URI's available on the Web but not returned by the endpoint. $\Delta^{\cap} := \mathcal{W} \cap \mathcal{S}$ contains the information about URI's found both through the endpoint and in the Web. Note that the information returned only by an index but not found in the retrieved Web content is in the set $\Delta^{\mathcal{S}} := \mathcal{S} \setminus \mathcal{W}$, but we do not consider it in this study. There are many possible reasons why a an entity appears in $\Delta^{\mathcal{S}}$. It can be due to some temporary unavailability (e.g. server downtime or connection error), due to changes in the data, or additional inferred data that was materialised in the index [7]. The exact classification is non-trivial and requires a theoretical assessment that is out of the scope of this study.

1) Experimental Results: For the entities set, we randomly sampled 144,094 URIs from the billion triple challenge dataset of 2011[3]. This dataset contains 2 billion RDF statements from 8 million Web documents, crawled during May and June 2011, and contains a significant amount of the Linked Data available in the Web. All selected URIs return a valid RDF document from the live look up. We then counted how many of these entities are also returned by the endpoints. We also measure the average query time taken to retrieve the available information for an entity. Table I shows the results for the three different cases (Web, Sindice and OpenLink).

	Web	**Web ∩ Sindice**	**Web ∩ Openlink**
entities found	144094	128856	43096
avg. query time	2506 ms	154 ms	75 ms

TABLE I
ENTITIES' STATISTICS.

At the time of our experiment, in September 2011, the Sindice Web index returned information for 128,856 of our selected entities, approximately 89% of the number found on the Web, whereas the OpenLink index gave information for about 43,096 entities of our set (approximately 30%). However, in terms of query execution, both endpoints are considerably faster than fetching the data directly from the Web.

In addition, for each of the entities contained in both sets, we compute its *Web result recall* in a SPARQL endpoint as $R_{\mathcal{W}} := \frac{|\Delta^{\cap}|}{|\mathcal{W}|}$. A $R_{\mathcal{W}}$ value of 1 means that all the information found online about the entity is also returned by the endpoint.

Figure 3 shows the distribution of the entities over the different intervals of $R_{\mathcal{W}}$, together with the cumulative distribution functions, for both endpoints.

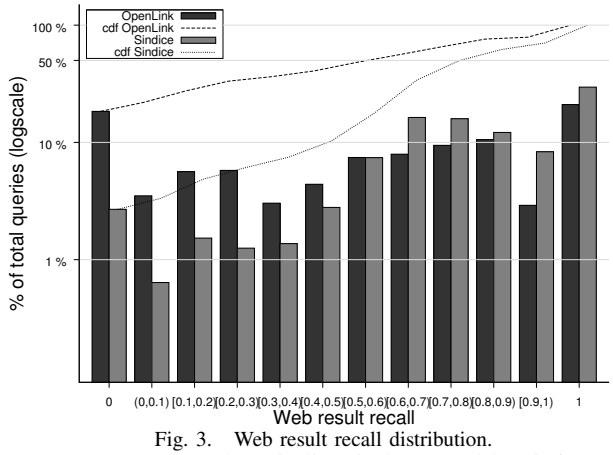

Fig. 3. Web result recall distribution.

We can see that the Sindice index provides information which is consistent with the Web for only approximately 30% of the entities. For the OpenLink index this value is even lower – only approximately 20%. In addition, Sindice returned only 50% of the available Web information for approximately 50% of the entity (cf. cdf curve). The OpenLink index returned only 50% of the Web information for around 80% of the entities.

Our experiments show that the available indexes for the Web of Data currently do not reflect the available information on the Web of Data. Not all entities found in the Web could be retrieved from the endpoints, and for the entities that were returned by the endpoints, the information about them was often incomplete. However, they still provide considerably faster query response times, in comparison to fetching the data directly from the Web. All this should be taken into account by the query processing engines.

Motivated by the results above, we have also studied the Web result recall per RDF predicate across entities. The

[1] `http://sparql.sindice.com`
[2] `http://lod.openlinksw.com/sparql`
[3] `http://challenge.semanticweb.org/`

rationale behind this is that in RDF, links are typed and explicitly represent a particular relationship between two entities. Intuitively, we expect some relationships (consequently some links) to be more stable, for instance, that *Galway is-a City*, and others to be rather more dynamic/temporal, like *Jürgen lives-in Galway*. We followed the same procedure as above, but now we partitioned the result sets based on the predicate.

Figure 4 shows the distribution of the average over all entities for the resulting complementary Web result recall per predicate over the different intervals, together with the cumulative distribution functions for both indexes. We show the complementary recall to gain more detailed insights.

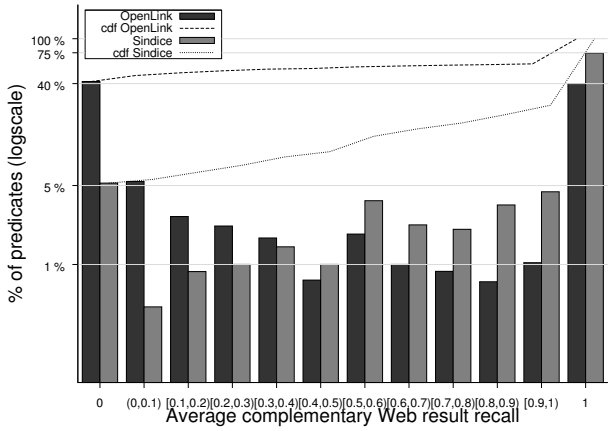

Fig. 4. Distribution of average complementary Web result recall per predicate.

Here we can observe two different characteristics between the two endpoints. The OpenLink's result set missed roughly 40% of the information for predicates contained in the Web results for the entity queries (cf. right part of Figure 4). A possible explanation is that the index is missing major updates of datasets (e.g. several datasets changed their knowledge representation). The Sindice index missed all the available Web information for nearly 70% of the predicates. We can also see that there exists a significant amount of predicates for which the index contains all the information available on the Web (cf. left part of Figure 4). This indicates that indeed some predicates are more stable than other. Exploring this knowledge would greatly improve not only the query processing, but also the index updating and caching techniques.

B. A Hybrid Query Processing Model

Our study revealed that while popular SPARQL endpoints provide very fast query results, these results are often outdated. On the other hand, retrieving results live from the Web guarantees up-to-date results (with respect to availability), but is very time consuming. Thus, we propose to combine the best of both worlds resulting in a hybrid query architecture as depicted in Figure 5. In the following, we briefly discuss the components of this architecture, their interplay, and the resulting technical and conceptual challenges.

For retrieving results, we assume the availability of at least one public SPARQL index interface (endpoint) for an RDF repository and further a query engine that can execute SPARQL queries directly over the Linked Data in the Web. The former could be one of the endpoints used in our study, the latter could be based on, for instance, linked-traversal [11] or on a data summary approach as proposed in [20]. Further, we consider the currently available Web data as the most accurate and up-to-date. In future work we will extend this to cover index data not available in the Web, which is, as mentioned before, a particularly challenging problem. Thus, we actually propose to always rely on the actual Web results, but to utilise centralised indexes to significantly speed up query processing. The intuition behind our approach is that we split each query in two parts: one that we assume to be rather static and for which the results from the repository are reliable, and one that we assume to refer to rather dynamic data and for which we retrieve results live from the Web. Note that this supports different concrete approaches:

1) fix the query split and execute the static part only on the index and the dynamic part only live,
2) follow a confirmation approach, i.e., run the whole query or only the static part on the index, and the whole query or only the dynamic part live,
3) meet users requirements in terms of freshness of results, maximal answer time, or both.

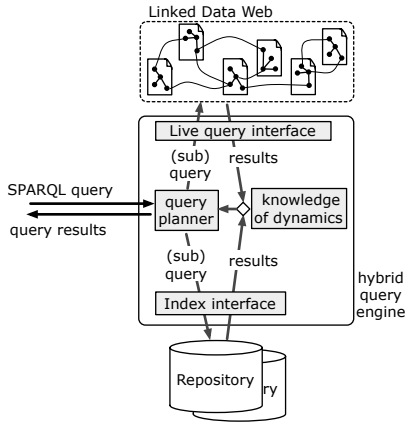

Fig. 5. Architecture overview of a hybrid query engine.

In either case, we require the following functionality from the single components:

Knowledge of Dynamics: This component provides the knowledge required to split a query in static and dynamic parts. By today, there exists no work that provides an appropriate approach for Linked Data. However, first steps have been made. One option is to base such knowledge on empirical studies [19]. Alternatively, this can be based on mining the dynamics of Linked Data in order to identify patterns [21]. For both directions, a major challenge is the size and openness of the Web, which makes it hard to predict the underlying dynamics. Once we overcome this challenge, we have to face the problem of mapping the patterns we found to concrete query patterns. Eventually, this will provide the knowledge about the change frequency of Web sources and the consistency of the SPARQL index required for query planning.

Query Planner: The dynamic knowledge directly feeds into a cost-based query planning. Such a query planner shall be based on the cost-based approaches known from traditional database systems, combining expected answer times and freshness as cost factors. This results in a range of major challenges. First, query answer times are usually almost impossible to predict in distributed environments, due to unknown delays and

unpredictable load of Web sources. However, this functionality is a main requirement to find the right trade-off between freshness and answer times. Second, we have to be able to express and adhere to freshness guarantees and completeness of answers, every time in the particularly context of current user requirements. Third, the eventual combination of these cost factors has not been approached before. The query planner uses the resulting cost model to minimise the number of HTTP lookups necessary to guarantee certain freshness requirements.

Index Interface: We aim at exploring any Web index available which offers a SPARQL endpoint, that means we will rely on their internal query optimiser and executor for the results. In this situation, estimating the query execution costs becomes more challenging. In simple cases, the static and the dynamic parts of a query would be disjoint. However, the complexity increases when both parts depend on each other (e.g. a join involving static and dynamic data). For all the cases we should aim at ways of influencing the query planning at the endpoint (in the line of optimiser hints, which might not be supported by all endpoints). This should come along with the option of interleaving index and live processing, e.g., getting some first static results, enriching them by dynamic content, and finally retrieving the last missing parts from the index again. There exists no proposal for such a requirement on top of arbitrary endpoints yet. Currently, we are analysing general classes of queries and assessing how endpoints would actually have to be influenced.

Live Querying: While the link-traversal approach is one option, in prior work we showed that it can be significantly outperformed by data summary approaches [20]. The idea behind it is that we use (approximate) index information (from the endpoint or an additional summary structure) to identify relevant Web sources before actually executing a live query. However, to not miss results of unknown sources, this has to be combined with a sort of link-traversal technique. While there exist approaches proposing both directions, there exists no proposal on how to actually combine them. Achieving the best trade-off here is another crucial requirement to speed up live query processing itself, and poses another main challenge.

Our proposed hybrid query engine architecture does not fully rely on any sort of index, we can still execute certain types of queries directly over the Web. However the third party repositories are still required for "fuzzy" and "incomplete" guidance. Thus, it is nevertheless crucial to maintain the used indexes. A mix between push based (e.g. ping services) and pull based (e.g. continuos crawling) can result in efficient strategies to learn, discover and update content changes. One can even think of a tight integration between the live query processor and the used repository. But also experiences from research about (Web) caching and replication, in conjunction with the results from mining Web data, will have major impact on the actually chosen combination of these methods.

V. Conclusion

In this work, we propose our view on the Web of Data as a Web of dataspaces and particularly focus on the missing alignment to the requirements and components of a support platform. We argue that the combination of Linked Data and database technologies qualify to establish missing services and to overcome the challenges we identified. We discuss that particularly the question of how to index, search and query structured data on the Web poses a crucial challenge in the context of dynamic data – which is truly existing, as we could show empirically. While we believe in the proposed alternative approach that combines live Web queries with the access to centralised repositories, we understand the wide range of challenges this bears. However, first steps that were taken in order to overcome these challenges are indicating the feasibility of the proposed architecture. Together with other research directions, we are confident that our work on the missing pieces and concepts will successfully enable the Web of Data on the basis of powerful and flexible support platforms.

Acknowledgement This research has been supported by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-II), by the Irish Research Council for Science, Engineering and Technology (IRCSET)

References

[1] The data hub. http://ckan.net/.
[2] T. Berners-Lee. Linked data - design issues. http://www.w3.org/DesignIssues/LinkedData.html, 2006.
[3] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.
[4] R. Cyganiak and A. Jentzsch. The linking open data cloud diagram. http://richard.cyganiak.de/2007/10/lod/.
[5] S. Das, S. Sundara, and R. Cyganiak. R2rml: Rdb to rdf mapping language. http://www.w3.org/TR/r2rml/.
[6] R. Delbru, N. Toupikov, M. Catasta, G. Tummarello, and S. Decker. Hierarchical link analysis for ranking web data. In *ESWC*, 2010.
[7] R. Delbru, G. Tummarello, and A. Polleres. Context-dependent owl reasoning in sindice - experiences and lessons learnt. In *RR*, 2011.
[8] L. Ding, D. DiFranzo, A. Graves, J. R. Michaelis, X. Li, D. L. McGuinness, and J. Hendler. Data-gov wiki: Towards linking government data. In *AAAI Spring Symposium on Linked Data Meets AI.*, 2010.
[9] A. Fekete, D. Gupta, V. Luchangco, N. Lynch, and A. Shvartsman. Eventually-serializable data services. In *PODC*, 1996.
[10] M. Franklin, A. Halevy, and D. Maier. From databases to dataspaces: a new abstraction for information management. *SIGMOD Rec.*, 2005.
[11] O. Hartig, C. Bizer, and J.-C. Freytag. Executing sparql queries over the web of linked data. In *ISWC*, 2009.
[12] M. Hausenblas and M. Karnstedt. Understanding Linked Open Data as a Web-Scale Database. In *DBKDA*, 2010.
[13] T. Heath and C. Bizer. *Linked data: evolving the web into a global data space*. Morgan & Claypool, [San Rafael, Calif.], 2011.
[14] A. Hogan, A. Harth, and A. Polleres. Scalable authoritative owl reasoning for the web. *Int. J. Semantic Web Inf. Syst.*, 5(2):49–90, 2009.
[15] Y. Li and J. Heflin. Using reformulation trees to optimize queries over distributed heterogeneous sources. In *ISWC*, 2010.
[16] D. L. McGuinness and F. van Harmelen. Owl web ontology language overview. http://www.w3.org/TR/owl-features/.
[17] D. L. Phuoc, M. Dao-Tran, J. X. Parreira, and M. Hauswirth. A native and adaptive approach for unified processing of linked streams and linked data. In *ISWC*, 2011.
[18] E. Prud'hommeaux and A. S. (eds.). SPARQL Query Language for RDF. W3C Recommendation, W3C, Jan. 2008. available at http://www.w3.org/TR/rdf-sparql-query/.
[19] J. Umbrich, M. Hausenblas, A. Hogan, A. Polleres, and S. Decker. Towards dataset dynamics: Change frequency of linked open data sources. In *LDOW2010 at WWW*, 2010.
[20] J. Umbrich, K. Hose, M. Karnstedt, A. Harth, and A. Polleres. Comparing data summaries for processing live queries over linked data. *WWW Journal (Springer US), Special Issue "Querying the Data Web"*, 2011.
[21] J. Umbrich, M. Karnstedt, and S. Land. Towards understanding the changing web: Mining the dynamics of linked-data sources and entities. In *KDML*, 2010.
[22] W3C. Resource description framework. http://www.w3.org/RDF/.
[23] W3C. Semantic web activity. http://www.w3.org/2001/sw/.