Building Enterprise Ready Applications Using Linked Open Data

Amar-Djalil Mezaour[1](✉), Bert Van Nuffelen[2], and Christian Blaschke[3]

[1] Dassault Systemes, Vélizy-Villacoublay, France
amardjalil.mezaour@3ds.com
[2] Tenforce, Leuven, Belgium
bert.van.nuffelen@tenforce.com
[3] SWCG, Vienna, Austria
blaschkec@semantic-web.at

Abstract. Exploiting open data in the web community is an established movement that is growing these recent years. Government public data is probably the most common and visible part of the later phenomena. What about companies and business data? Even if the kickoff was slow, forward-thinking companies and businesses are embracing semantic technologies to manage their corporate information. The availability of various sources, be they internal or external, the maturity of semantic standards and frameworks, the emergence of big data technologies for managing huge volumes of data have fostered the companies to migrate their internal information systems from traditional silos of corporate data into semantic business data hubs. In other words, the shift from conventional enterprise information management into Linked Opened Data compliant paradigm is a strong trend in enterprise roadmaps. This chapter discusses a set of guidelines and best practices that eases this migration within the context of a corporate application.

1 Introduction

Linked Data, Open Data and Linked Open Data (LOD) are three concepts that are very popular nowadays in the semantic community. Various initiatives, like openspending.org, are gaining ground to promote the openness of data for more transparency of institutions. But what is the difference between these three concepts?

Linked Data refers to the way of structuring the data and creates relationships between them. Open Data similar to open-source, opens content to make it available to citizens, developers, etc. for use with as limited restrictions as possible (legal, technological, financial, license). Linked Open Data, that we refer to as LOD, is the combination of both: to structure data and to make it available for others to be reused.

The LOD paradigm democratized the approach of opening data sources and interlinking content from various locations to express semantic connections like similarity or equivalence relationships for example. In business environment,

S. Auer et al. (Eds.): Linked Open Data, LNCS 8661, pp. 155–174, 2014.
DOI: 10.1007/978-3-319-09846-3_8

data interlinking practice is highly recommended for lowering technological and cost barriers of data aggregation processes. In fact, semantic links between data nuggets from separate corporate sources, be they internal or external, facilitate the reconciliation processes between data references, enhance semantic enrichment procedures of data like for example propagating annotations from similar references to incomplete data, etc.

In the context of enterprises, the LOD paradigm opens new scientific and technical challenges to answer emerging semantic requirements in business data integration. The impact of LOD in enterprises can be measured by the deep change that such an approach brings in strategic enterprise processes like domain data workflows. In fact, semantic enrichment and data interlinking contribute to optimize business data lifecycle as they shorten the data integration time and cost. Moreover, when data is semantically managed from its source, i.e. from its acquisition or creation, less time and efforts are required to process and integrate it in business applications. This semantic management implies a set of procedures and techniques like data identification as resources using URIs, metadata annotations using W3C standards, interlink with other data preferably from authority sources or domain taxonomies, etc.

On the other hand, LOD techniques foster the creation of advanced data applications and services by mashing up various heterogeneous content and data:

- from internal sources like CRM, ERP, DBMS, filesystems;
- from external sources like emails, web sources, social networks, forums.

As a consequence, new perspectives are open to offer innovative channels to consume, exploit and monetize business data and assets. To understand the rationale behind this new perspectives, Fig. 1 depicts a generic enterprise semantic data lifecycle from the acquisition to the final consumption.

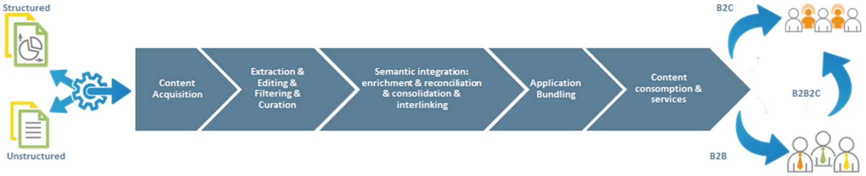

Structured
Unstructured
Content Acquisition
Extraction & Editing & Filtering & Curation
Semantic integration: enrichment & reconciliation & consolidation & interlinking
Application Bundling
Content consumption & services
B2C
B2B2C
B2B

Fig. 1. Data workflow in enterprise application

2 The Landscape of Enterprise and Corporate Data Today

Data integration and the efficient use of the available information in a business context are major challenges. A typical enterprise has critical applications from different vendors running on various technologies, platforms and communicating via different routes and protocols within and outside an Enterprise. These applications create disparate data sources, data silos and introduce enormous costs. To manage this complexity, an enterprise IT eco-system is viewed as a set of interconnected (or partially connected) applications managing different processes of the enterprise, where separation of the applications often means replication of the same data in different forms. Each process manipulates different kinds of data and produces new data in a structured or unstructured fashion as it is depicted in Fig. 2.

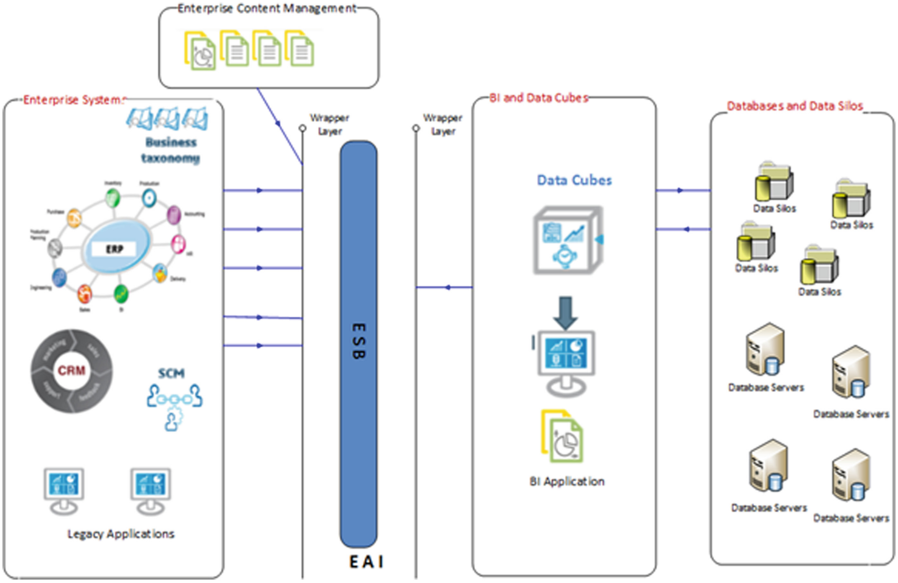


Fig. 2. Classical Enterprise Information System

Existing technological approaches such as Enterprise Application Integration (EAI) create middleware between all these diverse information sources making use of several architectural models with examples being Event Driven Architecture (EDA) or Service Oriented Architecture (SOA) which are usually implemented with web services and SOAP. Common approaches in the enterprises typically include Data Warehousing and Master Data Management.

Simple XML messages and other B2B standards ensure the "flow" of information across internal systems in an easy-to-use and efficient manner. In some cases this is enough but, for example, with a portfolio of over 30,000 offered products and services it is not possible to describe complex components with a handful of simple XML elements. There is a clear need for providing clear definitions or semantics to the data to facilitate integration at the data layer.

However, integration in the data layer is far from being a straightforward task and the Linked Data paradigm provides a solution to some of the common problems in data integration. The two technological approaches, i.e. EAI and LOD, are not contradictory but rather complementary. SOA architecture deployed in an EAI approach works with service oriented whereas LOD works with hyperlinked resources (data, data sets, documents, ...). Note that SOA architecture needs many custom services where LOD uses only a few services (SPARQL, REST) and hyperlinking with the referenced resources. Both approaches have complementary standardization efforts (on metadata vs. services) which makes them better suited for different tasks. EAI-SOA approach is well suited for well-defined tasks on well-defined service data whereas LOD is more targeted for innovative tasks involving semantics (integrations, mappings, reporting, etc.).

3 Why Should My Company Assets Go Linked Open Data?

The benefit of the adoption of Linked Data technologies in enterprises is multidimensional:

- address the problem of data heterogeneity and integration within the business;
- create value chains inside and across companies;
- meaning on data enables search for relevant information;
- increase value of existing data and create new insights using BI and predictive analytics techniques;
- Linked Data is an add-on technology which means no need to change the existing infrastructure and models;
- get a competitive advantage by being an earlier adaptor of LOD technologies.

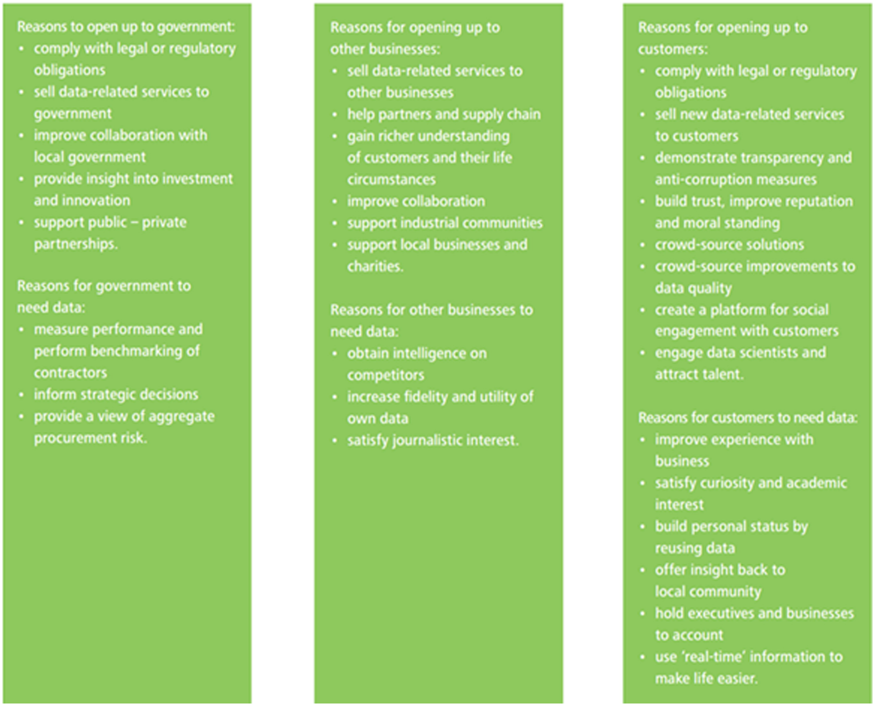
These benefits are better detailed in Fig. 3 taken from Deloitte report "Open data: Driving growth, ingenuity and innovation"[1].

4 LOD Enterprise Architectures

When adopting LOD principles, the Classical Enterprise IT Architecture (Fig. 2) is enhanced for working over the Internet with means to overcome the technical barriers of the format and semantic differences of exchanged and manipulated data. This generates a data processing workflow that is described in the following three figures:

1. Figure 4 evolves the legacy or classic architecture by replacing the Enterprise Software Bus (ESB) with Linked Open Data protocols for data published on an external server.

[1] http://www.deloitte.com/assets/Dcom-UnitedKingdom/Local%20Assets/Documents/Market%20insights/Deloitte%20Analytics/uk-insights-deloitte-analytics-open-data-june-2012.pdf



Reasons to open up to government:

- comply with legal or regulatory obligations
- sell data-related services to government
- improve collaboration with local government
- provide insight into investment and innovation
- support public – private partnerships.

Reasons for government to need data:

- measure performance and perform benchmarking of contractors
- inform strategic decisions
- provide a view of aggregate procurement risk.

Reasons for opening up to other businesses:

- sell data-related services to other businesses
- help partners and supply chain
- gain richer understanding of customers and their life circumstances
- improve collaboration
- support industrial communities
- support local businesses and charities.

Reasons for other businesses to need data:

- obtain intelligence on competitors
- increase fidelity and utility of own data
- satisfy journalistic interest.

Reasons for opening up to customers:

- comply with legal or regulatory obligations
- sell new data-related services to customers
- demonstrate transparency and anti-corruption measures
- build trust, improve reputation and moral standing
- crowd-source solutions
- crowd-source improvements to data quality
- create a platform for social engagement with customers
- engage data scientists and attract talent.

Reasons for customers to need data:

- improve experience with business
- satisfy curiosity and academic interest
- build personal status by reusing data
- offer insight back to local community
- hold executives and businesses to account
- use 'real-time' information to make life easier.

Fig. 3. Benefits for businesses to go LOD

2. Figure 5 evolves the legacy or classic architecture by replacing the Enterprise Software Bus with Linked Open Data protocols among the enterprise LOD publishing servers.
3. Figure 6 zooms-in on a publishing workflow, a transformation pipeline that is added on top of the legacy enterprise services (CRM, ERP, ...). Some legacy systems may evolve and upgrade to include LOD publishing or they may provide feeds into the LOD publishing workflow.

4.1 LOD Enterprise Architecture with a Publishing Workflow

Figure 4 illustrates the LOD Enterprise architecture where the middleware framework (ESB) of the Classical IT architecture (Fig. 2) is replaced with the LOD cloud. This architecture shows two types of data publishing, with the enterprise putting their RDF data on an external LOD server (server 5 in Fig. 4) according to one of two scenarios:

1. An RDF data set is produced from various data sources and subsystems (box 1 in Fig. 4) and is transferred to an external central LOD server.
2. Metadata is added to a classic web site using semantic metadata annotations (*e.g.* RDFa, Schema.org) on the HTML pages (box 3 in Fig. 4). An LOD server

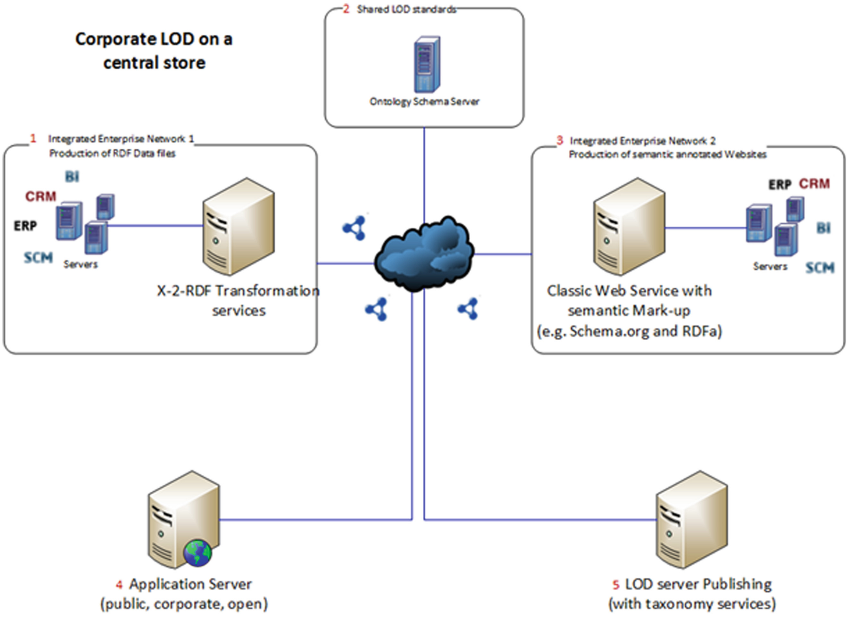

Corporate LOD on a central store
2 Shared LOD standards
Ontology Schema Server
1 Integrated Enterprise Network 1
Production of RDF Data files
BI
CRM
ERP
SCM
Servers
X-2-RDF Transformation services
3 Integrated Enterprise Network 2
Production of semantic annotated Websites
ERP
CRM
BI
SCM
Servers
Classic Web Service with semantic Mark-up (e.g. Schema.org and RDFa)
4 Application Server (public, corporate, open)
5 LOD server Publishing (with taxonomy services)

Fig. 4. LOD Enterprise Architecture

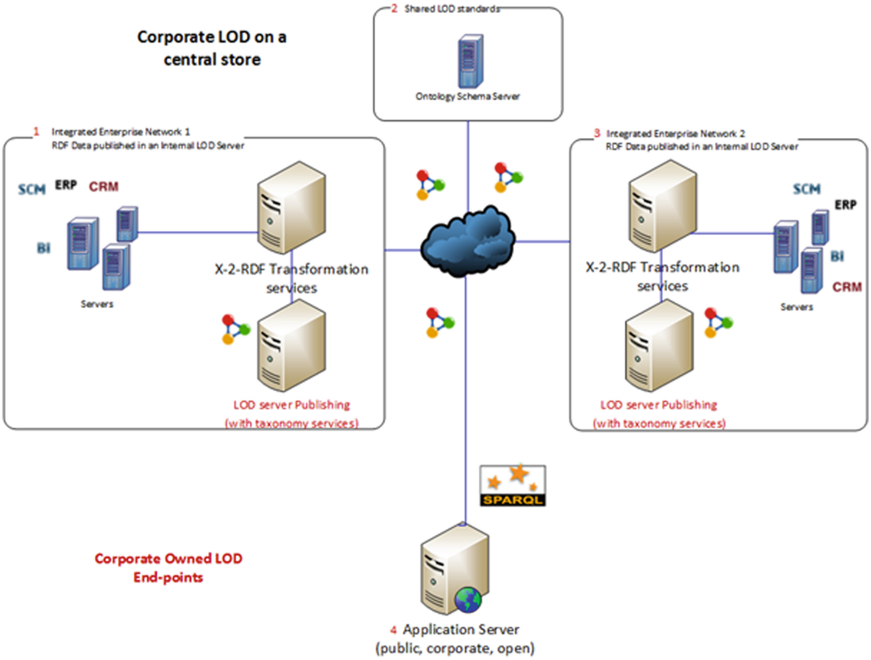

Corporate LOD on a central store
2 Shared LOD standards
Ontology Schema Server
1 Integrated Enterprise Network 1
RDF Data published in an Internal LOD Server
SCM
ERP
CRM
BI
Servers
X-2-RDF Transformation services
LOD server Publishing (with taxonomy services)
3 Integrated Enterprise Network 2
RDF Data published in an Internal LOD Server
SCM
ERP
BI
CRM
Servers
X-2-RDF Transformation services
LOD server Publishing (with taxonomy services)
SPARQL
Corporate Owned LOD End-points
4 Application Server (public, corporate, open)

Fig. 5. LOD Enterprise Integration Architecture

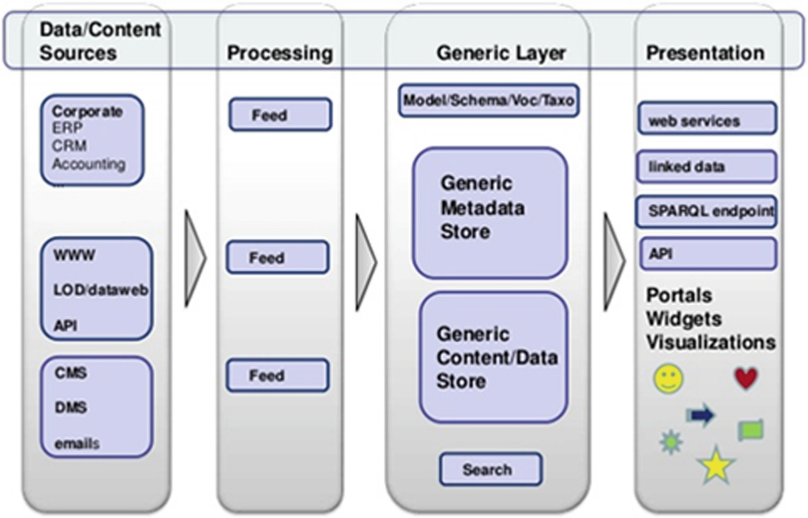


Fig. 6. Transformation pipeline

extracts this metadata, organizes it and makes it accessible as a central service (box 4 in Fig. 4).

The Ontology Schema Server (box 2 in Fig. 4) hosts the used ontologies capturing the semantics of the Linked Open Data. It may be standard (preferred) or custom designed. Other application services or platforms (server 4 in Fig. 4) may use the central LOD services to build specific calculations and reports. Business niches or completely new business opportunities can be created with visualizations and aggregations of data.

Example
A head-hunter can crawl job postings and match with CVs. Aggregations of offered vacancies in real-estates can create new insights. Search engines may use the data for advanced searching while portals[2] can harvest data sets from other portals and publish them from a single point of access in a LOD server (server 5 in Fig. 4).

4.2 LOD Enterprise Architecture Integration

In the previous LOD Enterprise Architecture (Fig. 4 on p. 160) the business operations are described where Linked Data were produced or semantic annotations were made to the corporate content. The data produced (or extracted from crawling through websites) was not published by the enterprise but was made available to the community. Any external LOD server could be used for the publishing of the data, depending on the needs and requirements of the re-users.

[2] Such as ODP: http://open-data.europa.eu/en/data/

In Fig. 5 the highlight is put on the operation of an enterprise that publishes its own data on a LOD server. Furthermore, the enabled integration is illustrated between various networks whether they belong to different branches of the same enterprise or entirely different companies. Figure 5 on p. 160 shows two company owned LOD publishing services (box 1 and 3 in Fig. 5). The published RDF is put on the company owned (corporate or enterprise) server platform. Other application services or platforms (server 4 in Fig. 5) may use the owned LOD services to build specific calculations and reports. Such application services may be on a dedicated external platform or they may be on one or more of the corporate owned LOD platforms/end-points. The Ontology Schema Server (box 2 in Fig. 5) hosts the used ontologies capturing the semantics of the Linked Open Data. It may be standard (preferred) or custom designed.

4.3 Transformation Pipeline to LOD Enterprise Architecture

The implementation of the previously described types of LOD architectures (shown in Figs. 4 and 5) is based on a transformation pipeline that is added on top of the legacy enterprise services (e.g. CRM, ERP, etc.). The pipeline includes:

1. Identification of the types of data which are available, i.e. separate data into public and private and define access security strategy, identify data sources, design retrieval procedures, setting data versions, provide data provenance;
2. Modelling with domain-specific vocabularies;
3. Designing the URI Strategy for accessing the information, i.e. how the model and associated data should be accessed;
4. Publishing the data which includes extraction as RDF, storage and querying;
5. Interlinking with other data.

5 Best Practices

5.1 Data Sources Identification

Corporate information can be defined as the data that is used and shared by the different employees, departments, processes (IT or not) of a company. Depending on the information security policy, corporate data can be accessed, processed and published via different business applications of the enterprise IT system. Note that it may be spread across different locations (internal departments and entities, regional or cross countries subsidiaries, etc.).

When integrating LOD technologies into an existing enterprise IT system or application, the first recommendation is to perform an audit on the different business data sources used by the company. This audit should include the following elements:

- Classification of business data sources according to their importance to the operation of strategic business processes.

- Cartography of data workflow between the identified data sources to discover missing, redundant or incomplete information exchanged, the type of data (structured, unstructured), etc.
- Mapping table between native business data formats and the corresponding standard formats (preferably W3C RDF like formats) and the impact from shifting from the native to the standard format.

This audit allows the data architects to better understand the corporate applications' functioning and help them evaluating the cost of integrating LOD technology. According to the required effort and cost, the first best practice consists on migrating as much as possible native formats to standards, preferably RDF-like W3C standards when possible. This considerably eases the publishing, annotation and interlinking of business data.

To comply with the openness criterion of LOD paradigm, publishing data is a major recommendation in the "LODification" process of corporate data. To do so, a licensing scheme must be released to define how the opened data can be reused and exploited by third-party users, applications and services. Considering the company interest, a compromise must be found to open as much data as possible and maintaining a good balance between keeping strategic enterprise data confidential, like the know-how for example, and the rest of data open. Lot of reusable licensing schemes can be considered.

Last but not least, the opened data license scheme must guarantee the reuse principle of data by third-party applications with as few technical, financial and legal restrictions as possible. One way of achieving these goals is to provide rich metadata descriptions of the opened data with appropriate vocabularies, like DCAT[3], VoID[4], DublinCore[5], etc. To make the opened and published data understandable and retrievable, the metadata description must provide key elements like the copyright and associated license, update frequency of data, publication formats, data provenance, data version, textual description of the data set, contact point when necessary to report inconsistencies or errors for example, etc.

5.2 Modelling for the Specific Domain

In order to transform the existing model of an enterprise to a more interoperable schema, best practices focus on the use of common vocabularies. Using terms of existing vocabularies is easier for the publisher and contributes a lot in the re-use and the seamless information exchange of enterprise data.

As a first step, the inherent structure of the legacy data has to be analysed. If no specified hierarchy exists, it can often be created based on expert knowledge of the data. If such an organization of the data is not possible, then only a list of concepts, basically a glossary, can be constructed. Depending on the complexity

[3] http://www.w3.org/TR/vocab-dcat/

[4] http://www.w3.org/TR/void/

[5] http://dublincore.org/

of the data and how the entities are related, different data schemas can be used to express them.

5.3 Migration of Legacy Vocabularies

The migration of an existing vocabulary to an RDF scheme varies in complexity from case to case, but there are some steps that are common in most situations. Transforming enterprise data to RDF requires:

- Translating between the source model and the RDF model is a complex task with many alternative mappings. To reduce problems, the simplest solution that preserves the intended semantics should be used.
- The basic entity of RDF is a resource and all resources have to have a unique identifier, a URI in this case. If the data itself does not provide identifiers that can be converted to URIs, then a strategy has to be developed for creating URI for all the resources that are to be generated (see Sect. 5.4).
- Preserve original naming as much as possible. Preserving the original naming of entities results in clearer and traceable conversions. Prefix duplicate property names with the name of the source entity to make them unique.
- Use XML support for data-typing. Simple built-in XML Schema datatypes such as xsd:date and xsd:integer are useful to supply schemas with information on property ranges.
- The meaning of a class or property can be explicated by adding an "`rdfs:comment`", preferably containing a definition from the original documentation. If documentation is available online, "`rdfs:seeAlso`" or "`rdfs:isDefinedBy`" statements can be used to link to the original documentation and/or definition.

Domain specific data, can be modelled with vocabularies like Org[6] or GoodRelations[7]. Only when existing vocabularies do not cover ones needs new schemas should be developed. Data sets that will be published on the web should be described with metadata vocabularies such as VoiD, so that people can learn what the data is about from just looking at its content.

Where suitable vocabularies to describe the business data do not exist, one possibility is to develop a SKOS thesaurus instead of an RDFS model (e.g. taxonomies, organizations, document types). This approach is easier to follow for organisations new to RDF. Tools such as PoolParty[8] exist and support users in such a task. The most recent international standard regarding thesaurus development is the ISO 25964[9]. This standard provides detailed guidelines and best practices that interested readers should consider.

Once the data is in this format it can be loaded in a triple store like Virtuoso and published internally or on the web.

[6] http://www.w3.org/TR/2014/REC-vocab-org-20140116/

[7] http://www.heppnetz.de/projects/goodrelations/

[8] http://www.poolparty.biz

[9] http://www.niso.org/schemas/iso25964/

5.4 Definition of the URI Strategy

To meet high quality standards for managing business data, a company must define a persistent data representation policy for identifying each data item from the enterprise data universe. Such a policy must include the addressing schemes for locating data resources within the enterprise space. An URI[10] is a relevant mechanism for defining a global representation scheme of the enterprise business data space.

Identification of Business Items as Resources Referenced by URIs
The first recommendation in building a coherent and persistent representation policy is to identify business data items as resources, which can be individually referenced. To conform to the LOD principles, URIs should be used as the identification mechanism for referencing the business information resources.

Use HTTP/DNS Based URIs
A URI is a mechanism that can be used for identifying different objects and concepts. Some of these objects and concepts may have a physical existence like books for example with ISBN, web page with URL locations. Other concepts are abstract and represent conceptual things like ontology concepts or data items. Different schemes of URIs exist for representing a resource: URIs based on DNS (Domain Name Server) names, ARK (Archival Resource Key) and URIs based on names and IDs like ISBN (International Standard Book Number), DOI (Digital Object Identifiers), Barcodes, etc. Some of the schemes described above can be inadequate to implement basic Linked Open Data features like publishing, referencing and interlinking. Therefore, it is strongly recommended to use URIs based on HTTP protocol and DNS names (like URLs and ARK) to ensure visibility, accessibility and reuse of business items in external applications and to third party users.

Use De-referenceable URIs
Human users associate mechanically HTTP based URIs to URLs and expect to have a web page when pasting a URI into a browser address bar. Unfortunately, the association of a URI to a web page is not always true and automatic. For some businesses, such a situation may generate confusion and frustration. To avoid such misunderstanding, it is highly recommended to provide means to have "dereferenceable" and resolvable URIs, *i.e.* URIs that return meaningful responses when pasted into a browser's address bar. A typical meaningful response could be an HTML page containing a complete or partial description, including the properties of the corresponding resource.

Separate Resource and Resource Representation
Making business items accessible and dereferenceable through the HTTP protocol may generate a conceptual confusion between the resource itself and the document describing it (the HTML answer for example when requesting the resource over HTTP). The resource itself as a business data item should be identified by

[10] Uniform Resource Identifier: RFC3986 http://tools.ietf.org/html/rfc3986

a URI that is different from the possible representations that one could generate to describe the resource (an HTML, RDF, XML or JSON description document, a document in a given language, a document using a given technology: PHP, HTML, etc.). W3C proposes two technical solutions to avoid the previous confusion: use of hash URIs and use of 303 URIs:

- Hash URIs - This solution consists in using fragment URIs to reference a non-document business resource. A fragment URI is a URI that separates the resource identifier part from the DNS server path location part using the hash symbol '#'. For example, a book reference 2-253-09634-2 in a library business application could be dissociated from its description using a hash URI as follows: http://www.mylibrary.com/books/about#2-253-09634-2. With this example, the library can manage a repository of books in one single big RDF file containing all the books references and their properties. When accessing 2-253-09634-2 book, a selection query can be applied on that RDF document to extract the RDF fragment corresponding to 2-253-09634-2 triples. The HTTP server managing the de-referencement of URIs will apply business specific rules to render the RDF fragment in the desired technology (as JSON, HTML, XML, etc.).
- 303 URIs - This solution consists in implementing a redirection mechanism represented by the HTTP response code 303 to indicate that the resource has been identified and the server is redirecting the request to the appropriate description option. In the example of the library, the URI could be http://www.mylibrary.com/books/2-253-09634-2. The HTTP server will answer to the request of that URI by a redirect (`HTTP code 303`) to a new location; let's say http://www.library.com/books/2-253-09634-2.about.html, to provide the description of the requested resource.

Both techniques have advantages and drawbacks as discussed by Sir Tim Berners Lee here: http://www.w3.org/Provider/Style/URI. Whereas hash URI technique may look restrictive due to the same root part URI (before the hash) for different resources, the 303 URI technique introduces latency in requests due to the redirection mechanism.

Design Cool URIs

On the conceptual design side of URIs, Sir Tim Berners Lee proposes the notion of Cool URIs to guarantee that URIs are maintainable, persistent and simple (see http://www.w3.org/TR/cooluris/). To ensure sustainable URIs, it is important to design a "cool" URI scheme that doesn't change over time. To do so one has to follow these basic rules, resumed in Fig. 7:

- Make the URI independent from the underlying technology used to generate or describe the resource. This means avoid extensions such as `.php`, `.cgi` and `.html` in the URI path. To know what to return when a resource is requested (without any extension), it is recommended to implement a content negotiation mechanism in the HTTP server that is able to outstream the appropriate content.

- Make also the URI independent of the physical location of the file describing the resource. Never forget that physical locations are subject to change.
- Make sure that resource metadata are not included in the URI because of their evolution over time. In other words, one has to not encode the following in the URI the authorship, the status of the resource (final, old, latest, etc.), the access rights (public, private, etc.) since this may change over time.

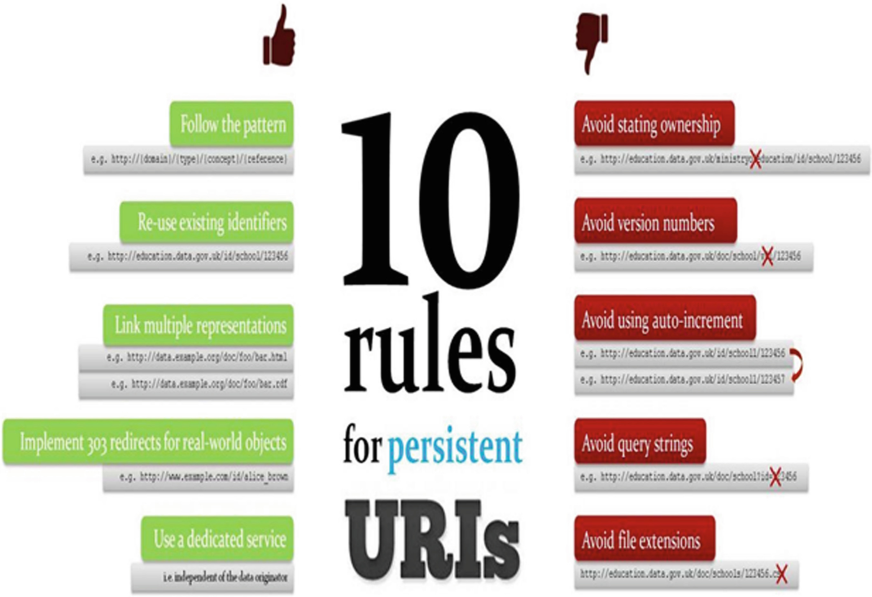

Follow the pattern
e.g. http://{domain}/{type}/{concept}/{reference}
Re-use existing identifiers
e.g. http://education.data.gov.uk/id/school/123456
Link multiple representations
e.g. http://data.example.org/doc/foo/bar.html
e.g. http://data.example.org/doc/foo/bar.rdf
Implement 303 redirects for real-world objects
e.g. http://www.example.com/id/alice_brown
Use a dedicated service
i.e. independent of the data originator
10 rules for persistent URIs
Avoid stating ownership
e.g. http://education.data.gov.uk/ministryofeducation/id/school/123456
Avoid version numbers
e.g. http://education.data.gov.uk/doc/school/v1/123456
Avoid using auto-increment
e.g. http://education.data.gov.uk/id/school1/123456
e.g. http://education.data.gov.uk/id/school1/123457
Avoid query strings
e.g. http://education.data.gov.uk/doc/school?id=123456
Avoid file extensions
http://education.data.gov.uk/doc/schools/123456.csv

Fig. 7. URI Design rules

Opaque vs. Non opaque URIs

Designing mnemonic and readable URIs for identifying business resources can help human users to get preliminary knowledge on the targeted item. However, from a business point of view, this readability may have side effects if it also reveals an internal organisation system structure. Non Opaque URI may reveal conceptual structure but never should reveal physical or logical data structures. In fact, third party users or external applications can attempt to hack the URI scheme, reverse engineer all business resources and abuse the access to some strategic assets. When there are security risks, it is recommended to use opaque URIs instead of readable ones. An opaque URI is a URI conforming to a scheme that satisfies the following conditions:

- Addressable and accessible resources should be referenced by identifiers instead of human readable labels.

- The resource URI should not contain explicit path hierarchy that can be hacked to retrieve sibling resources for example.
- The URI content should not provide means to gain information on the referenced resource, i.e., a third party client cannot analyse the URI string to extract useful knowledge on the resource.

For non-opaque URI, only the first constraint is not followed.

5.5 Publishing

Publishing procedures in Linked Data follows the identification of data sources and the modelling phase and actually refers to the description of the data as RDF and the storing and serving of the data. A variety of tools have been created to assist the different aspects of this phase from different vendors and include a variety of features. According to the needs of each specific business case and the nature of the original enterprise data, shorter publishing patterns can be created.

5.5.1 Publishing Pattern for Relational Data

Relational databases (RDB) are the core asset in the existing state-of-art of data management and will remain a prevalent source of data in enterprises. Therefore the interest of the research community[11,12] has gathered around the development of mapping approaches and techniques in moving from RDB to RDF data. These approaches will enable businesses to:

- Integrate their RDB with another structured source in RDB, XLS, CSV, etc. (or unstructured HTML, PDF, etc.) source, so they must convert RDB to RDF and assume any other structured (or unstructured) source can also be in RDF.
- Integrate their RDB with existing RDF on the web (Linked Data), so they must convert to RDF and then be able to link and integrate.
- Make their RDB data to be available for SPARQL or other RDF-based querying, and/or for others to integrate with other data sources (structured, RDF, unstructured).

Two key points should be taken into consideration and addressed within the enterprise (see Fig. 8):

Definition of the Mapping Language from RDB2RDF

Automatic mappings provided by tools such as D2R[13] and Virtuoso RDF Views provide a good starting point especially in cases when there is no existing Domain Ontology to map the relational schema to. However, most commonly the manual definition of the mappings is necessary to allow users to declare domain-semantics in the mapping configuration and take advantage of the integration

[11] http://www.w3.org/2011/10/rdb2rdf-charter.html

[12] http://www.w3.org/DesignIssues/RDB-RDF.html

[13] http://d2rq.org/d2r-server

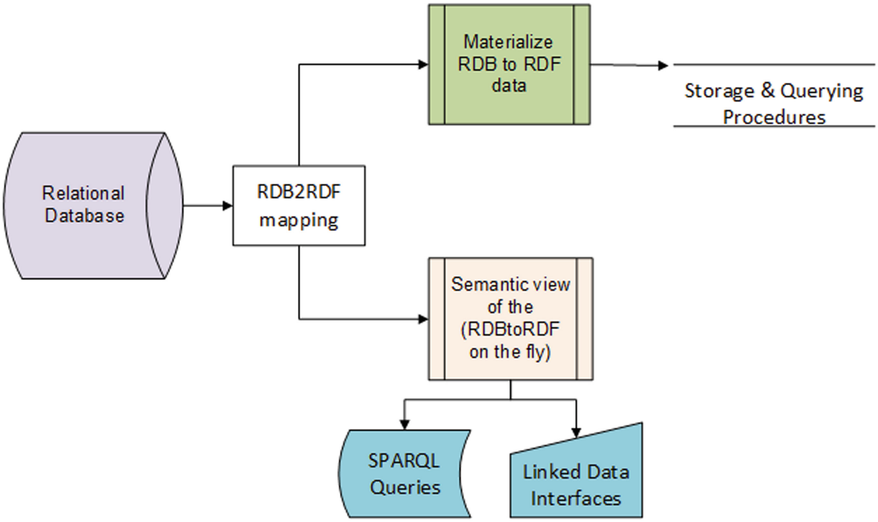

Materialize RDB to RDF data
Storage & Querying Procedures
Relational Database
RDB2RDF mapping
Semantic view of the (RDBtoRDF on the fly)
SPARQL Queries
Linked Data Interfaces

Fig. 8. RDB2RDF publishing pattern

and linking facilities of Linked Data. R2RML[14], a W3C recommendation language for expressing such customized mappings, is supported from several tools including Virtuoso RDF Views and D2R.

Materializing the Data
A common feature of RDB2RDF tools is the ability to create a "semantic view" of the contents of the relational database. In these cases, an RDF version of the database is produced so that content can be provided through a SPARQL endpoint and a Linked Data interface that works directly on top of the source relational database, creating a virtual "view" of the database. Such a "semantic view" guarantees up-to-date access to the source business data, which is particularly important when the data is frequently updated. In contrast, generating and storing RDF requires synchronization whenever either the source data model, the target RDF model, or the mapping logic between them changes. However, if business decisions and planning require running complicated graph queries, maintaining a separate RDF store becomes more competitive and should be taken under consideration.

5.5.2 Publishing Pattern for Excel/CSV Data

When the original data reside in Excel or CSV format, describing them with RDF would be a first step of a publishing pattern while hosting and serving it on the Web follows. LODRefine is a stack component, well-suited to automating and easing the "RDFizing" procedure. Usage brings direct added business value:

- powerful cleaning capabilities on the original business data.

[14] http://www.w3.org/TR/r2rml/

- reconciliation capabilities, in case it is needed, to find similar data in the LOD cloud and make the original business data compatible with well-known Linked Data sources.
- augmenting capabilities, where columns can be added from DBpedia or other sources to the original data set based on the previous mentioned reconciliation services.
- extraction facilities when entities reside inside the text of the cells.

5.5.3 Publishing Pattern for XML Data

When the original data is in XML format an XSLT transformation to transform the XML document into a set of RDF triples is the appropriate solution. The original files will not change; rather a new document is created based on the content of the existing one. The basic idea is that specific structures are recognized and they are transformed into triples with a certain resource, predicate and value. The LOD2 stack supports XML to RDF/XML XSLT transformations. The resulting triples are saved as an RDF/XML graph/file that can follow the same hosting and serving procedures explained in the previous section.

5.5.4 Publishing Pattern for Unstructured Data

Despite the evolution of complex storage facilities, the enterprise environment is still a major repository paradigm for unstructured and semi-structured content. Basic corporate information and knowledge is stored in a variety of formats such as PDF, text files, e-mails, classic or semantic annotated websites, may come from Web 2.0 applications like social networks or may need to be acquired from specific web API's like Geonames[15], Freebase[16] etc. Linked Data extraction and instance data generation tools maps the extracted data to appropriate ontologies en route to produce RDF data and facilitate the consolidation of enterprise information. A prominent example of a tool from the LOD2 stack that facilitate the transformation of such types of data to RDF graphs is Virtuoso Sponger.

Virtuoso Sponger[17] is a Linked Data middleware that generates Linked Data from a big variety of non-structured formats. Its basic functionality is based on Cartridges, that each one provides data extraction from various data source and mapping capabilities to existing ontologies. The data sources can be in RDFa format[18], GRDDL[19], Microsoft Documents, and Microformats[20] or can be specific vendor data sources and others provided by API's. The Cartridges are highly customizable so to enable generation of structured Linked Data from virtually any resource type, rather than limiting users to resource types supported by the default Sponger Cartridge collection bundled as part of the Virtuoso Sponger.

[15] http://www.geonames.org/
[16] http://www.freebase.com/
[17] http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtSponger
[18] http://rdfa.info
[19] http://www.w3.org/TR/grddl/
[20] http://microformats.org/

The PoolParty Thesaurus Server[21] is used to create thesauri and other controlled vocabularies and offers the possibility to instantly publish them and display their concepts as HTML while additionally providing machine-readable RDF versions via content negotiation. This means that anyone using PoolParty can become a W3C standards compliant Linked Data publisher without having to know anything about Semantic Web technicalities. The design of all pages on the Linked Data front-end can be controlled by the developer who can use his own style sheets and create views on the data with velocity templates.

DBpedia Spotlight[22] is a tool for automatically annotating mentions of DBpedia resources in text, providing a solution for linking unstructured information sources to the Linked Open Data cloud through DBpedia. DBpedia Spotlight recognizes that names of concepts or entities have been mentioned. Besides common entity classes such as people, locations and organisations, DBpedia Spotlight also spots concepts from any of the 320 classes in the DBpedia Ontology. The tool currently specializes in English language, the support for other languages is currently being tested, and it is provided as an open source web service.

Stanbol[23] is another tool for extracting information from CMS or other web application with the use of a Restful API and represents it as RDF. Both Dbpedia Spotlight and Stanbol support NIF implementation (NIF will soon become a W3C recommendation) to standardise the output RDF aiming on achieving interoperability between Natural Language Processing (NLP) tools, language resources and annotations.

5.5.5 Hosting and Serving

The publishing phase usually involves the following steps:

1. storing the data in a Triple Store,
2. make them available from a SPARQL endpoint,
3. make their URIs dereferenceable so that people and machines can look them up though the Web, and
4. provide them as an RDF dump so that data can easily be re-used.

The first three steps can be fully addressed with a LOD2 stack component called Virtuoso, while uploading the RDF file to CKAN[24] would be the procedure to make the RDF public.

OpenLink Virtuoso Universal Server is a hybrid architecture that can run as storage for multiple data models, such as relational data, RDF, XML, and text documents. Virtuoso supports a repository management interface and faceted browsing of the data. It can run as a Web Document server, Linked Data server and Web Application server. The open source version of Virtuoso is included in the LOD2 stack and is widely used for uploading data in its Quad store, it

[21] http://www.poolparty.biz

[22] https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki

[23] http://stanbol.apache.org/

[24] http://ckan.org/

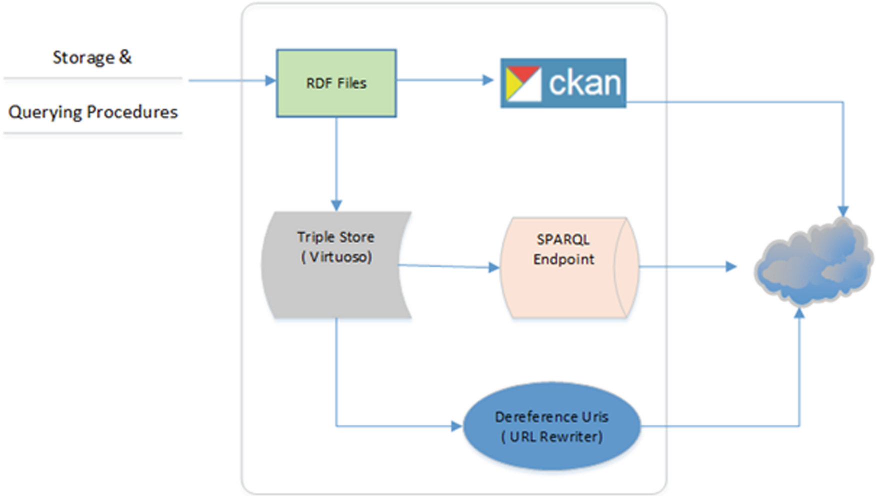


Fig. 9. Publishing pattern for registering data sets

offers a SPARQL endpoint and a mechanism called URL-Rewriter to make URIs dereferenceable.

According to the fourth step, sharing the data in a well-known open datahub such as CKAN will facilitate their discovery from other businesses and data publishers. The functionality of CKAN is based on packages where data sets can be uploaded. CKAN enables also updates, keeps track of changes, versions and author information. It is advised as good practice to accompany the data sets with information files (e.g. VOID file) that contain relevant metadata (Figs. 9, 10).

5.6 Interlinking - The Creation of 5-Star Business Data

5-Star business data[25] refers to Linked Open Data, the 5 stars being:

1. data available on the web with an open-data license,
2. the data is available in a machine readable form,
3. the machine readable data is in a non-proprietary form (e.g. CSV),
4. machine readable, non-proprietary using open standards to point to things,
5. all the above, linked with other data providing context.

To get the full benefits of linked data with the discovery of relevant new data and interlinking with it, requires the 5^{th} star, but that does not mean that benefits are not derived from the Linked Data approach before that point is reached. A good starting point can be business registers such as Opencorporates[26] or

[25] http://5stardata.info/
[26] https://opencorporates.com/

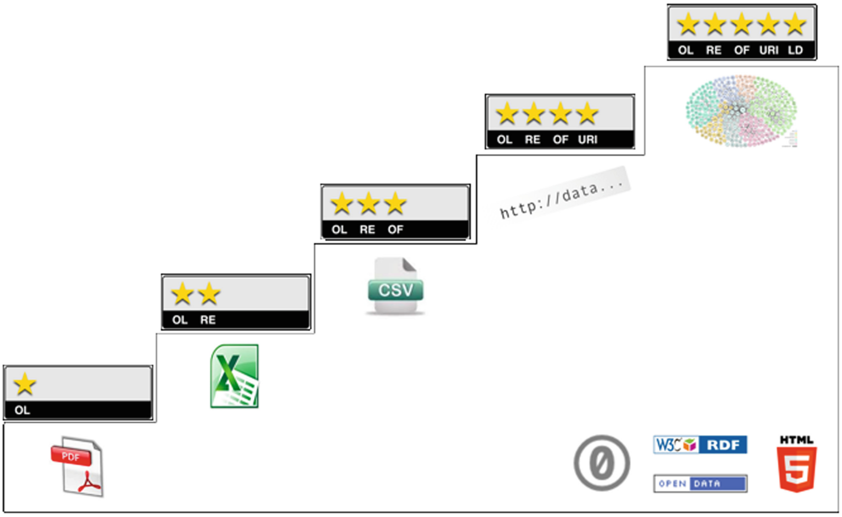


Fig. 10. 5 star data

UK Companies House[27] that contain the metadata description of other companies. The discovery of more related business data can further be facilitated with Linked Data browsers and search engines like SigmaEE[28]. However, the implementation of interlinking between different data sources is not always a straightforward procedure. The discovery of joint points and the creation of explicit RDF links between the data in an automated way can be supported with tools both included in the Interlinking/Fusion LOD2 life cycle.

The process that is referred to as interlinking is the main idea behind the Web of Data and leads to the discovery of new knowledge and their combinations in unforeseen ways. Tools such as SILK[29] offer a variety of metrics, transformation functions and aggregation operators to determine the similarity of the compared RDF properties or resources. It operates directly on SPARQL endpoints or RDF files and offers a convenient user interface namely Silk Workbench.

5.7 Vocabulary Mapping

Sometimes, an enterprise may need to develop a proprietary ontology when applying Linked Data principles. Mapping the terms that were used for publishing the triples with terms in existing vocabularies will facilitate the use of the enterprise data from third-party applications. A tool that supports this kind of mapping is R2R[30].

[27] http://www.companieshouse.gov.uk/
[28] http://sig.ma
[29] http://lod2.eu/Project/Silk.html
[30] http://wifo5-03.informatik.uni-mannheim.de/bizer/r2r/

R2R searches the Web for mappings and apply the discovered mappings to translate Web data to the application's target vocabulary. Currently it provides a convenient user interface that facilitates the user in a graphical way to select input data from a SPARQL endpoint as well as from RDF dumps, create the mappings and write them back to endpoints or RDF files.

6 Conclusion

In this chapter, we discussed the best practices to deploy in an enterprise application to ensure a full LOD paradigm compliant semantic dataflow. We also saw that deploying LOD tools and procedures does not necessary requires to start the IT design from scratch but can be deployed on top of existing applications. This guarantees low cost deployment and integration.