The Europeana Linked Open Data Pilot Server

Nicola Aloia, Cesare Concordia, and Carlo Meghini

Istituto di Scienza e Tecnologie dell'Informazione,
National Research Council, Pisa, Italy
(nicola.aloia,cesare.concordia,carlo.meghini)@isti.cnr.it

Abstract. The Linked Data is a set of principles and technologies providing a publishing paradigm for sharing and reusing RDF data on the Web. The Linked Data Cloud is expanding at a very high speed since 2007, when the Linked Data Project was launched. Europeana, the European Digital Library, subscribes to the view of a web of data, and the distribution of cultural heritage data is one of the main objectives established by the Europeana Strategic Plan. The paper illustrates how Europeana publishes Linked Data, with focus on the technological approach adopted.

Keywords: Linked Data, Linked Data Server, Europeana.

1 Introduction

The Linked Data is a set of principles and technologies providing a publishing paradigm for sharing and reusing data on the Web [1]. In a well-known paper [3] Tim Berners-Lee, coined the term Semantic Web which advocated to extend the web of documents as "a web of data that can be processed directly and indirectly by machines", with the ability of discovering new resources through the interconnection of similar data. The Europeana project goal is to provide integrated access to digital objects from the cultural heritage organizations of all the nations of the European Union. To achieve this objective, Europeana provides a set of tools, such as the portal, a set of APIs for programmatic access to its resources, etc. Having the ability to provide metadata as Linked Open Data, is very important for Europeana to attract new users and new providers because the linked data paradigm enables the use of digital representations of cultural artifacts for generating knowledge [7]. For this reason, the implementation of Linked Open Data Pilot Server (LODPS) is an important step for Europeana, its partners and third parties. It paves the way towards achieving two crucial Europeana targets: enable connecting related data and makes them easily accessible using common Web technologies and enable everyone to access, reuse, enrich and share data.

Distributing the whole Europeana datasets as Linked Open Data (LOD) requires to process the existing Europeana metadata, coded according to the Europeana Semantic Elements (ESE), to obtain RDF descriptions as required by Linked Data approach (ESE enrichment and transformation), and to define an agreement with every data provider to publish their data as open data.

M. Agosti et al. (Eds.): IRCDL 2012, CCIS 354, pp. 241–248, 2013.

We decided to focus on finding solutions for the first issue, by creating a Linked Open Data Pilot server that exposes as Linked Data a subset of the Europeana content belonging to those providers, who want to make their data available on the web. Note that the Linked Open Data Pilot server is technically separated from the Europeana production server.

The approaches and technical solutions adopted for transforming ESE metadata into a richer and more flexible format and to link the Europeana data with other sources are described in [2].

This paper will describe in details the server built to publish Europeana Linked Data, showing its architecture, and the technical solutions adopted.

2 Linked Data Server

In [4] a number of best practices, known as the Linked Data Principles, are proposed. The basic idea is to use the architecture of the World Wide Web to share data on a large scale:

1. Use URIs as names for things. That is, use the URIs to report not only documents, but also objects and concepts of the real world.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL). That is, all URIs must be defererenceable, i.e. client applications use the HTTP protocol to look up the URI and to obtain a description of the resource identified by the URI, using standard notations.
4. Include links to other URIs. so that they can discover more things.

A Linked Data Server is an HTTP server application that offers the ability of discovering new resources through the interconnection of *similar* data and complies with the *Linked Data Principles.*

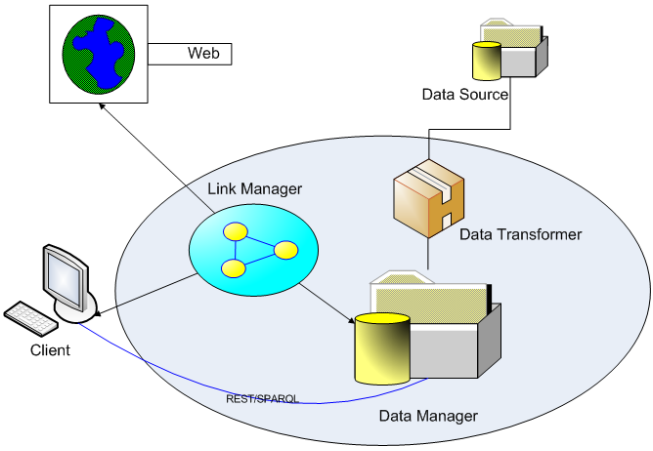

Fig. 1. Linked Data Server

A LD server can be logically divided in three components: a *Link Manager*, a *Data Transformer* and a *Data Manager* (Fig. 1). The role of the *Data Transformer* is to process the original dataset formatting and enriching it in order to publish it as linked data, and store it by means of the *Data Manager*. Generally speaking the *Data Manager* provides functionalities to index, search, access and maintain data. The *Link Manager* is the front end for the client application. It usually provides functionalities to process requests and format responses according to Linked Data specifications [1].

3 Web of Data: Making URIs Dereferenceable

The HTTP protocol was originally designed to manage HTML documents, i.e. compound hypermedia resources formatted according to a common rendering language. In Linked Data, instead, resources are not only documents; they can also be real world objects or abstract concepts. Moreover in the web of documents hypertext links are simply a way to access documents and don't contain information on the resource accessed. In the Linked Data paradigm, every resource is identified by a URI, and a Linked Data (LD) server must be able to dereference the URI i.e. to propose a description of the resource identified by the URI if this resource is not a document.

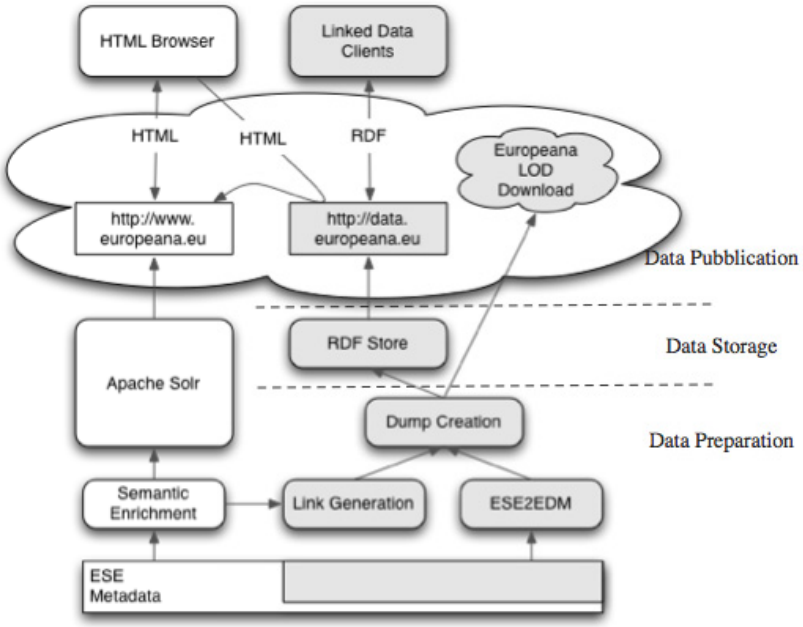

Fig. 2. LD server technical architecture

There are two main strategies to implement the URI dereference mechanism: 303 URIs and hash URIs, both are described in details in [6]. In summary:

- In the 303 URIs strategy if the server recognizes that the URI identify a real object or an abstract concept , it sends to the client a HTTP response code “303 See Other” and a link to a web document describing the resource, the client then asks for this document and the server returns it with HTTP code “200”
- In the hash URIs strategy the *fragment identifier* of a URI (the part of a URI that follows the # symbol) is used to identify real-world objects and abstract concepts, without creating ambiguity.

Advantages and disadvantages of both strategies are discussed in [6]; in essence: hash URIs have the advantage of reducing the number of necessary HTTP connections, which, in turn, reduces access latency, 303 URIs, on the other hand, is more flexible because the redirection target can be configured separately for each resource.

For the Europeana LD server we decided to adopt the 303 URIs strategy, the main reasons for this choice are explained in the following paragraph.

4 The Europeana Linked Data server

The following picture, taken from [2], shows the overall architecture of the Europeana Linked Data server.

According to the publishing steps individuated in [1] we can describe the server as follows:

- **Data preparation:** this step is executed by three server components. The ESE2EDM component that downloads the data from the Europeana dataset and maps the ESE metadata records into EDM information objects, the Link Generation component that enrich the EDM objects and creates the links to other Linked Data Sources and the Dump creation component that merges the results of the above components into a set of dump files. A detailed description of the algorithms and tools implementing these components can be found in [2].

  The output of this step consists of a number of RDF triples (currently: 115.769.306) that are stored a) in a set of dump files, b) in an RDF-Store. Every dump file contains RDF triples belonging to a specific collection; dump files are published and can be downloaded.
- **Data Storage:** The data storage is implemented using an RDF store. It is important to note that the dataset of the Europeana Linked Open Data Pilot is loaded in the RDF Store using a batch procedure and it does not change, this means that data manipulation is not critical in the Europeana Linked Data Server. On the contrary the response time for queries to the RDF Store is very critical since every data resource in the store is accessed via query.

  Results presented in [8], where performances and features the main RDF Store are compared, shows that the Virtuoso server is a good solution from the performance point of view. Moreover Virtuoso provides also a REST web service to perform SPARQL queries over HTTP, this feature is used to publish Europeana Linked Data via SPARQL.

- **Data Publication:** the component publishing Linked Data is implemented by a Web Server and by a library of Java servlets. The Web Server receive every request and redirect it to i) the download area if a dump file is requested, ii) the servlets library if, instead, a resource is requested. The servlets implement the 303 URIs dereference strategy. The implementation algorithm is based on the HTTP server-driven content negotiation mechanism [5], which enables HTTP clients and servers to negotiate a possible answer to a specific request. When a client requests a resource the LOD server checks the expected media type, if the request is for an HTML document a '303 redirection' is issued to the document describing the resource in the official Europeana Web server (www.europeana.eu). In case the client expects an RDF media type then the request URI is parsed to individuate the type of resource requested (a resource map, a proxy, an aggregation or an item) and the resource ID. Using these information a new URI is created and used as 303 redirect target. If the client accepts this redirection the URI is used as query parameter for a SPARQL Describe query. An example of interaction is shown in Fig. 3

Fig. 3. Dereferencing URI for an RDF resource

5 Accessing Linked Data

The Europeana LOD server provides three way to access linked data (see Table 1): via file transfer it is possible to download the whole dataset or specific collections, using the SPARQL GUI it is possible to query the LOD dataset to obtain single resources or collections of resources according to defined query filters, the HTTP/GET protocol allow to access resources via URI dereferencing.

As described in the previous chapter the implementation of the URI dereference strategy is based on the analysis of the "Accept" field value in the HTTP request header.

The role of this field in an HTTP request is to specify the acceptable media for the response, the value consists of a comma separated list of media types with the associated quality factor (a 'q:' followed by number in scale 0 1) i.e. the degree of preference indicated by the client for the specific media type, if the quality factor is not defined for a media type it is considered as 1. An example is the following:

Accept: application/rdf, text/html;q=0.9, text/plain;q=0.8

Table 1. Europeana Linked Data publishing methods

Publishing method / Data published	File Transfer	REST/SPARQL	HTTP/GET
Complete dataset	Download dataset dump	N.A.	N.A.
Collection of resources	Download collection(s) dump	SPARQL 'Select' query	N.A.
Single resource	N.A.	SPARQL 'Describe' query	URI dereference

The Accept header value is parsed to check if the request asks for an HTML document or if an RDF resource is needed. When an html document is requested, the client request is redirected (303 redirection) toward the document describing the resource in the Europeana server: www.europeana.eu.

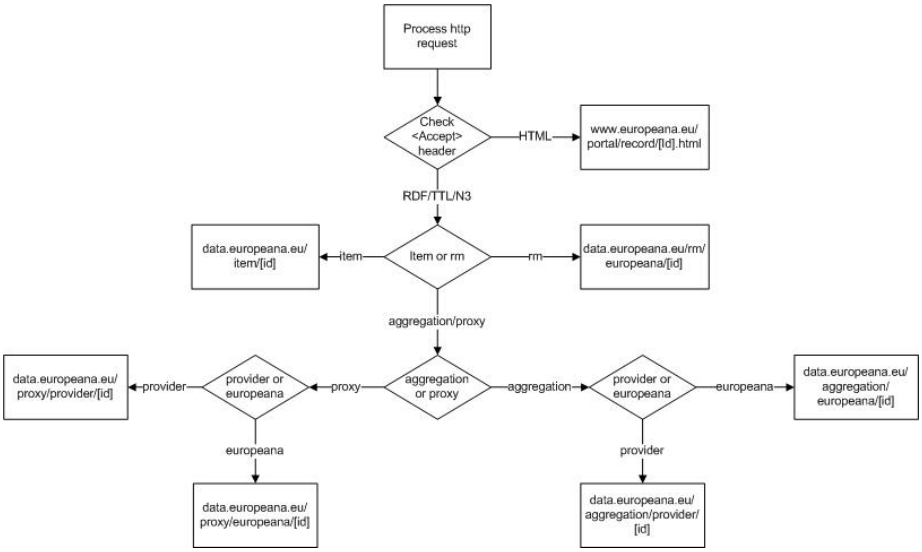

Fig. 4. HTTP request parsing

If instead the client asks for an RDF/TTL/N3 media type then the requested URI is parsed to individuate (i) the actual category of the resource requested (a resource map, a proxy, an aggregation or an item) and (ii) the resource ID.

The different categories of resources served by data.europeana.eu are [2]:

- **Item** (http://data.europeana.eu/item/[id]), a real-world object for which digital resources are available through Europeana
- **Resource Map** (http://data.europeana.eu/rm/europeana/[id]), a OAI-ORE resource map [10] indicating meta-level statements about the creation and publication of ORE data (ORE aggregations and their aggregated resources)

- **Provider aggregator** (http://data.europeana.eu/aggregation/provider/[id]) the digital resources submitted on an object by its provider, it also gives meta-information on the digital resource aggregation process, e.g., the name of the data provider
- **Europeana aggregator** (http://data.europeana.eu/aggregation/europeana/[id]) the digital resources maintained by Europeana for the object, it also gives meta-information on the data aggregation process, which is created by Europeana
- **Provider proxy** (http://data.europeana.eu/proxy/provider/[id]) gives all the data that applies to the real-world object, from the perspective of the data provider
- **Europeana proxy** (http://data.europeana.eu/proxy/europeana/[id]) gives all the data that applies to the real-world object, from the perspective of Europeana.

The LOD server gets a resource via a SPARQL 'DESCRIBE' query (i.e. a specific form of SPARQL query that returns a RDF graph describing the resource). The query is executed in the Europeana dataset stored in theVirtuoso triple-store.

The query result is parsed by the LOD server, formatted according to the requested media type and sent back to the client.

6 Conclusions and Next Steps

The Linked Open Data Pilot server publishes a subset of the Europeana dataset as Linked Data. It offers three different ways to clients for getting Europeana Linked Data: URIs dereferencing via the server located at http://data.europeana.eu, SPARQL queries via Web Service and data dump file downloading. The technology adopted and the code developed is open source [9]. The current activity on the Europena Linked Data pilot has three main goals: to increase the number of the Europeana content providers contributing to the Europeana Linked Data dataset, to refine the dataset quality by adding links to other Linked Data sets and to improve the implementation of the server functionalities. Concerning this last activity probably the biggest challenge is to improve the Data Store performances. Even if the Virtuoso Server query response time is acceptable for a pilot server, we're working to identify a solution applicable in a 'production' server, when potentially the whole Europeana Dataset could be published as Open Data. Another activity is to investigate other technical solutions adopted for the data publication to improve technical interoperability with other Linked Data servers.

References

1. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Morgan & Claypol Publishers (2011)
2. Haslhofer, B., Isaac, A.: data.europeana.eu: The Europeana Linked Open Data Pilot. In: DCMI Meetings and Conferences, DC 2011, The Hague (2011)
3. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American Magazine (May 17, 2001)
4. Berners-Lee, T.: Linked Data - Design Issues (2006), http://www.w3.org/DesignIssues/LinkedData.html

5. RFC 2616 – Hypertext Transfer Protocol – HTTP/1.1 – (Section 12: Content Negotiation)
6. Sauermann, L., Cyganiak, R.: Cool uris for the semantic web - w3c interest group note (2008), http://www.w3.org/TR/cooluris/
7. Gradmann, S.: Knowledge = Information in Context: on the Importance of Semantic Contextualiation in Europeana. Technical report, Berlin School of Library and Information Science, Humbold University (April 2010), http://www.scribd.com/doc/32110457/Europeana-White-Paper-1 (retrieved April 30, 2011)
8. Haslhofer, B., Roochi, E.M., Schandl, B., Zander, S.: Europeana RDF Store Report. Technical report, University of Vienna (retrieved April 30, (2011), http://eprints.cs.univie.ac.at/2833/
9. https://github.com/behas/slodr
10. http://www.openarchives.org/ore/