QuerioCity: A Linked Data Platform for Urban Information Management

Vanessa Lopez, Spyros Kotoulas, Marco Luca Sbodio,
Martin Stephenson, Aris Gkoulalas-Divanis, and Pól Mac Aonghusa

Smarter Cities Technology Centre, IBM Research, Ireland

Abstract. In this paper, we present QuerioCity, a platform to catalog, index and query highly heterogenous information coming from complex systems, such as cities. A series of challenges are identified: namely, the heterogeneity of the domain and the lack of a common model, the volume of information and the number of data sets, the requirement for a low entry threshold to the system, the diversity of the input data, in terms of format, syntax and update frequency (streams vs static data), and the sensitivity of the information. We propose an approach for incremental and continuous integration of static and streaming data, based on Semantic Web technologies. The proposed system is unique in the literature in terms of handling of multiple integrations of available data sets in combination with flexible provenance tracking, privacy protection and continuous integration of streams. We report on lessons learnt from building the first prototype for Dublin.

1 Introduction

Governments are increasingly making their data accessible to promote transparency and economic growth. Since the first data.gov initiative launched by the US government, many city agencies and authorities have made their data publicly available through content portals: New York City[1], London[2], San Francisco[3], Boston[4], and Dublin[5], to name a few.

Through these efforts, a large number of data sets from many different domains became available, allowing enterprises and citizens to create applications that can mash up data. However, the data sets shared in these portals come in different formats (csv, xml, kml, pdf,..), do not link to other sources on the Web, are heterogeneous and of variable quality. Semantic Web technologies have been adopted as a valuable solution to facilitate large-scale integration, sharing of distributed data sources and efficient access to government data [1,2]. However, converting raw government data into high quality Linked Government Data is

[1] http://www.nyc.gov/html/
[2] http://data.london.gov.uk/
[3] http://datasf.org/
[4] http://www.cityofboston.gov/doit/databoston/app/data.aspx
[5] http://www.dublinked.ie

P. Cudré-Mauroux et al. (Eds.): ISWC 2012, Part II, LNCS 7650, pp. 148–163, 2012.

costly [3,4], and there is a lack of practical approaches for converting and linking government data at scale [5]. Consequently, linked RDF data is sparse and often limited to metadata, and content in data portals is difficult to consume by web developers and users. In addition, dealing with dynamic data sources like Streams and allowing for multiple integrations of Linked Data remains an open problem.

We have developed QuerioCity, an open urban information management platform, based on semantic technologies, to easily capture and consume urban open data, with a particular focus on transforming, integrating and querying heterogenous semi-structured data in an open and large environment. The key research questions challenged were: How to represent and manage city-scale data as an information resource in practical and consumable ways? What are the challenges involved in creating an urban information management architecture with acceptable performance levels? What are the benefits and costs of using Linked Data technologies to allow people and systems to interact with the information ecosystem of a city? How can we provide privacy protection and capture provenance in an open world where traditional notions of information governance and control may no longer apply? How do the answers to the questions above change when dealing with streams instead of static data?

This paper describes the challenges, findings and lessons learnt from the first prototype of QuerioCity. The platform differs from existing open data initiatives in the following aspects: A strong focus on operational *live data* coming from physical sensors (transport, water and energy) or social media, the ability to *mix public and restricted data*, the transformation of raw data and metadata coming from data publishers in *various formats and structures* into linked open data at *enterprise scale*, and the ability to *detect and thwart privacy threats*.

In this paper, we are using the data portal for Dublin, named Dublinked, as a use-case, providing valuable real-world data, insight and challenges. Dublinked provides a real experimental validation for this research, a live and scalable scenario, where real-world data sets are obtained from four city authorities in the Dublin area: Dublin City, Dun Laoghaire-Rathdown, South Dublin and Fingal County Councils. The methods and technologies proposed in this paper are gradually rolled-out in Dublinked.

The rest of the paper is structured as follows. Section 2 presents the state-of-the-art and elaborates on the challenges of dealing with Urban data. We describe the rationale of QuerioCity in Section 3, along with its enterprise software components in Section 3.4. Our discussion, in Section 4, includes lessons learned and an analysis of the costs and benefits of Semantic Technologies.

2 Related Work and Research Challenges

The rise of the Open Data movement has contributed to many initiatives whose aim is generating and publishing government and geographical data according to Linked Data principles, such as OpenStreetMaps[6] and OrdnanceSurvey[7].

There are automated approaches for turning tabular data into a Semantic Web format. Pattern-based methods for re-engineering non-ontological resources into ontologies [8] are based on the use of thesauri, lexica and WordNet for making explicit the relations among terms. TARTAR [9] automatically generates knowledge models out of tables. In this system, grounded in the cognitive table model introduced by Hurst [10], a table is handled from a structural, functional and semantic point of view by respectively identifying homogeneous regions (group of cells) in a table, distinguishing between attribute cells and instance cells, and then finding semantic labels for each region content with the help of WordNet. The coverage of these approaches depends on WordNet or an ontology that models the domains of interest. To annotate tables on the Web and improve search, [11] uses a column-based approach. A class label is attached to a column if a sufficient number of the values in the column are identified with that label in some "is-a" databases extracted from the Web.

In [2], a new dataset-specific ontology is constructed for each dataset, representing only the data stored in the particular database. To convert this data into RDF, scripts are developed in correspondence with their manually-designated and built ontologies.

A number of tools for automatically converting tabular data (mostly CSV) into RDF also exist, such as RDF123 [12]. W3C defines a standardised mapping language R2RML[6] and an approach for converting relational databases to RDF. In this W3C candidate recommendation, the first row is used to suggest properties and each other row refers to entities, with one of the columns uniquely identifying the entity. This approach is used, for example, in the Datalift project [13] to automate the conversion from the source format to "raw RDF", before transforming it to "well-formed" RDF by using selected vocabularies and SPARQL construct queries.

The approach presented in [3] is based on Google Refine for data cleaning and a reconciliation service extended with Linked Data capabilities to enable exporting tabular data into RDF, while keeping provenance description represented according to the *Open Provenance Model Vocabulary* [14]. In our experience with Dublinked, asking the users to use tools such as Google Refine and define templates to guide the conversion process into RDF have limited fitness-for-use for the non-expert.

More often than not, urban data is sourced from legacy non-relational systems or spreadsheets made for consumption by humans. Urban data does not follow a relational model, it is highly heterogeneous and the structure is unknown (from static data to spatial-temporal data obtained from physical sensors). State-of-the-art approaches do not solve the entity recognition and type identification problem since data does not always come in a tabular format, and when it does, we cannot assume that each row is an entity, or that all the entities described are explicitly labeled in the table. For instance, Dublin City Council (DCC) published a dataset about energy consumption in the City Council Civic offices, as part of an initiative to reduce its carbon footprint by 2030. The dataset

[6] `http://www.w3.org/TR/r2rml/`

contains files, partly represented in Figure 1, with energy readings recorded every 15 minutes. These readings are split in different files for the new building blocks (Block 1 & 2) and the old ones (not shown in the figure). The location (DCC civic offices) and kind of data described in the data sets (energy measurements) is given in the text description in the metadata. In the content, the first row contains a unique cell with the blocks where the measurement was taken, the first column in the table represents dates, and the second column after the third row is the time of the day when the measurement was taken. Thus, each row of this dataset contains multiple entities of type measurement with properties to represent a given value and a given timestamp. *Automated methods have focused on relational data and are not yet able to deal with such structures.*

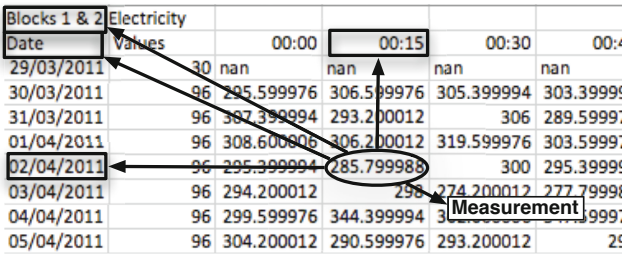

Fig. 1. Data sets about energy measurement in Dublin civic offices

As stated in [2], it is not realistic to assume that an organisation will subscribe to a single schema, or that different organisations will agree in a common semantic model. For instance, in Dublinked, each of the four county councils have published a data set about street lighting consisting in an inventory of pole locations. DCC represents this information in two CSV files: one including an ID and a name, and a second including an ID, Irish Grid spatial projections (easting, northing) and a location description. Fingal County council includes street name, outside / opposite house number, and easting / northing coordinates. Public Lighting locations in South Dublin County are described with IG spatial coordinates (ITMEast, ITMNorth), road names, and a set of "location" such as cul-de-sac, front, junction, school, lane way, etc. Dun Laonghaire county released the data as shape files. *Although all four data sets are valid, in the same domain, and machine-processeable, they are far from a common model.*

Urban information often comes in *Streams*. Although there have been efforts in integrating streams with Linked Data [15], the effort to do so has been centralized and manual.

Privacy has been traditionally offered on individual data sets of simple data types, where sensitive inferences are easy to predict. In Linked Data, and in an urban information management domain, data sets are characterized by the four Vs (velocity, variety, volume and veracity) and sensitive inferences can be drawn across multiple data sets and are thereby difficult to predict.

Summarizing, managing urban information raises challenges in terms of:

- *Fitness-for-use.* The users of the system are not data integration experts and not qualified to use industry data integration tools. Furthermore, they are not able to query data using structured query languages.
- *Domain modeling.* The domain of the information is very broad and open. As such, generating and mapping data to a single model is infeasible or too expensive. Even if such a model was to be created, it's complexity would hamper its use, given the target audience of the system.
- *Global integration.* Addressing the information needs for solving problems in an urban environment requires integration with an open set of external data sets. Furthermore, it is desirable that city data becomes easily consumable by other parties.
- *Scale.* The data in a city changes often (streams), is potentially very large and it is interlinked with an open set of external data.
- *Privacy.* Data sets may contain sensitive information that needs to be privacy-protected prior to their sharing. Even more, the linkage of data sets may lead to sensitive inferences which have to be blocked in accordance with legislation.

3 The QuerioCity Approach

We propose an approach where the integration effort is fundamentally incremental and split between the two major roles in the system: *data publisher* and *data consumer*. Data publishers need interactive tools and patterns to "lift" the data as much as possible, adding meaning to data through semantic annotations, linking across data sets and (partially) with large existent corpora in the Web, and protecting any sensitive information. On the other side, data consumers pull the data in order to fulfill their needs through searches for potentially relevant data sets, and by executing complex queries across data sets in an intuitive and exploratory fashion. In the process, the integration effort of consumers is reusable.

The data currency of Queriocity is a *Dataset*[7]. *Datasets* consist of the *Metadata* described in Section 3.1, a number of data *Graphs* or data *Streams*, and *Provenance* information, described in Section 3.2.

Figure 2 illustrates the overall approach taken in QuerioCity. Vertically, we illustrate the progression from raw source data to consumption. Initially, the system archives and catalogs *metadata* of the content (for example, keywords and publishing date), enabling rich queries over this meta-information, in combination with full-text search using Lucene[8] inverted indexes.

Good metadata is important in order to allow individuals to easily discover relevant data. However, this is not enough to answer queries that span various

[7] In this paper, we will refer to *Dataset* as a data artifact. We will refer to data set as the abstract notion.

[8] `http://lucene.apache.org`

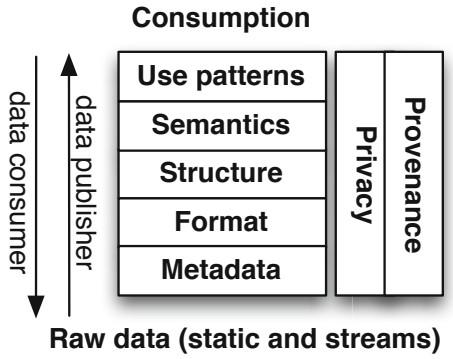

Fig. 2. The QuerioCity approach

data sets. By transforming the input into a uniform *format*, we are allowing queries where the data consumer knows the structure and the content of each file (possible by manual examination of the input). By harmonizing the *structure* of the various inputs, the user is able to query based on manually-mapped properties across data sets. Integrating the data into a common *semantic structure* allows transparent querying across data sets, without the need for manual mappings. Finally, given the heterogeneity (and consequently the semantic complexity) of the data, inferring *use patterns* allows users to create and share meaningful "views" over the data.

There are some issues that span all aspects of the system. Given the open nature of the integration process, it is imperative that the system records *information provenance*. In addition, given the potentially sensitive nature of the information, the system provides functionality for detecting *privacy threats* and tools for *anonymization* [16].

In what follows, we go into more detail for some of the key components of QuerioCity.

3.1 Metadata

In any data portal, and in Dublinked in particular, the metadata provides a fairly rich description of the data sets, which is critical for discovery and navigation. Some fields are given by the publisher, such as title, subject, keywords, publishing agency, license terms, collection purpose, period of coverage; and some others are updated automatically, such as date of creation, latest update and download links.

In this section, we outline, on the one hand, the process of generating Linked Data from Dublinked metadata: generation of the ontology model and the RDF data, and alignment of the datasets and links to external sources; and on the other hand how the generated Linked Data is used for user consumption and for building a publishing interface.

The ontological model created to represent the metadata catalogs is based on standard and widely used vocabularies, namely dublin core[9], FOAF[10] and DCAT[11]. New resources are accessible through HTTP following the W3C best practices for publishing Linked Data: resolving a dereferenceable URI gives relevant facts about the entity across all metadata sources.

Datatypes are standardised (e.g., using *xsd:date* and *dcterms:PeriodOfTime*) and instances are created for representing agencies, spatial administrative areas, access rights, update frequency, distribution format, keywords and categories. Consistent metadata is created by defining the range of properties as a given type. For instance, spatial administrative areas can refer to instances of county councils (Dun Laoghaire-Rathdown, Fingal, SouthDublin), city councils (Dublin city council), city regions (Greater Dublin, South East Inner City, Dublin city centre), and so on.

We are getting lot of value in standardisation and cleansing of the original data by virtue of this approach, e.g., spelling mistakes and duplicates are eliminated (e.g.,"DLR" council" and "Dun Laoghaire Rathdown County Council" are alternative labels for the same entity) and datasets are linked by the area they cover, the publishing agency, publishing date, etc.

The metadata is also linked to authoritative and external sources on the Web. Categories and keywords are mapped to the Integrated Public Sector Vocabulary (IPSV), which is a controlled vocabulary for populating the e-Government Metadata Standard that UK public sector organisations are required to comply with. The IPSV is available as an RDF Schema (based on SKOS) by the esd-toolkit[12]. Keywords are not fixed a priori, but categories are predefined in the ontology model. New categories are automatically added only if they have a direct mapping with an IPSV category. String distance metrics (Cohen et al., 2000), mainly a combination of Jaro and TFIDF, are used to automatically map the metadata to the best matching terms in IPSV (including all labels) and to avoid duplicated, e.g., keywords such as "coast" and "coastline" are both matched to the IPSV entity ipsv#527 "Coasts". However, noise and inaccuracies can be introduced by automatic approaches, e.g., "asset management" is matched to the IPSV term "waste management". These inaccuracies can be avoided at publishing time with the use of a semantically-aware publishing interface.

Equivalence *owl:sameAs* links to corresponding DBpedia entities are added, if any, to describe subjects and locations such as administrative counties. Linking to IPSV and external sources improves interoperability and discoverability of related datasets, for instance datasets are not just related because they use the same keyword, but also because they link to related, broader or narrower categories in IPSV. Furthermore, as shown in the examples in Section 2, for many of the datasets the entity described by the content is only explicitly mention in the metadata description. Linking the metadata to entities in DBpedia and IPSV

[9] `http://dublincore.org/specifications`
[10] `http://xmlns.com/foaf/spec`
[11] `http://www.w3.org/TR/vocab-dcat/`
[12] `http://doc.esd.org.uk/IPSV/2.00.html`

allows us to solve the type identification problem in many cases. Using the pole locations dataset, which is linked to the IPSV terms *street lighting* (ipsv#1404), *Street furniture* (ipsv#3103) and *Road and pathway maintenance* (ipsv#3025), the entities described by the data are of the most specific IPSV entity type, and also DBpedia category, "street lighting" (also known as "lamp posts").

As mentioned before, for newly published datasets we use a semantically-aware publishing interface that allows us to (1) validate on the fly external or internal mappings while the publisher fills in the metadata fields, (2) limit the user input to a set of instances of the appropriate range for a given property, or (3) allow the user to input free text but use existent metadata and the IPSV vocabulary to present suggestions to asset publishers annotating their metadata. For example, if the user start writing the keyword "parking", the system will propose further refining the input with the annotations "car park", "car parking permits", "resident parking", "disabled parking", "parking fines", "parking meters".

Besides a SPARQL endpoint, ranked searches and RESTful services are also provided. Figure 2 shows a summary view of the RDF metadata in Dublinked for the 209 datasets currently available, about 27K triples.

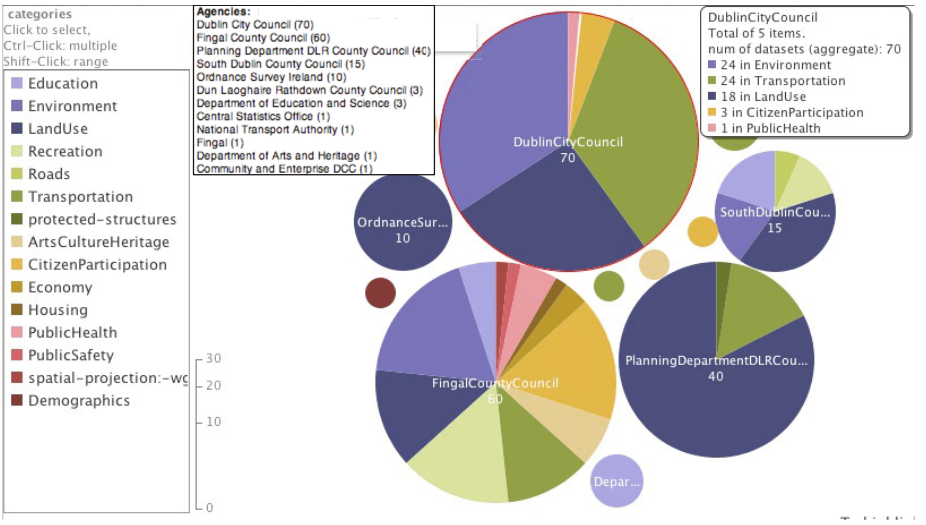

Fig. 3. Screenshot illustrating metadata from Dublinked by category and agency

In the future, we intend to use content to automatically push into the metadata information about the bounding box covered by the data. As such different datasets in different topics can also be dynamically link by spatial properties. Currently this field exists in the metadata, but none of the publishers fill it in.

3.2 Content

QuerioCity takes an incremental approach to managing content. Referring to figure 2, after extracting the relevant metadata, data sets go through the following steps towards consumption:

Format. All datasets in the system are preserved in their original formats. In addition, we transform datasets with known file formats to a simple RDF representation. This representation is not intended to capture semantics, but rather to provide a convenient and uniform way to represent the content of file. For example, the CSV file in figure 1, would be converted in triples of the form `:c1 a Cell. :c1 col "1". :c1 row "1". :c1 value "Blocks 1 & 2"`. This choice is motivated by the fact the it is not always possible to automatically convert CSV files to representations of entities.

Structure. Once homogeneous data areas are identified and validated with the user, pre-defined templates can be used as semantic masks that capture the intent of the publisher and guide the extraction of entities, making explicit the relations that hold between the entities described on the tables. Templates can be defined a priori but they can also be learned through user interaction and saved to be reused. The three dominant structures for tabular data in Dublinked are: geographically referenced entities (i.e. tables with two columns for longitude and latitude), measurements with a single entity per row and a column indicating the temporal aspect, and structures similar to that in Figure 1, representing measurements that reference time through both a given column and a row.

Semantics. The platform leverages semantic data types (geographical coordinates, dates, etc.) and automatically converts units of measurement. *Owl:sameAs* and *owl:equivalentAs* properties are used to link entities, eliminating the need for tight physical integrations imposed by relational databases and adopting a pay-as-you-go approach. These properties are discovered using a combination of existent reconciliation and state-of-the-art mapping techniques to detect common types and entity co-reference as well as user input. In addition, we can consume services provided by the `http://sameAs.org` web site for getting co-referent URIs for a given URI.

Use Patterns. Given the open nature of the domain, QuerioCity allows for multiple integrations and usage paths for Datasets. As we explain in the next paragraph, the system allows for efficient maintenance of such paths.

Data Model. In the QuerioCity data model, instance and schema information is stored in separate Named Graphs from the Metadata. Graphs are shared between Datasets (i.e. a Dataset may reference multiple Graphs, and a Graph can be referenced by multiple Datasets). Data integration tasks entail creating new Datasets. Instance data is append-only and schema information is read-only. This provides, possibly externally referenced Datasets with stability, while avoiding unnecessary data duplication.

Streams are handled using the same model. In the context of our system, Streams are analogous to Graphs; a Dataset can reference a set of Streams using URIs (possibly, in combination with Graphs). We make a distinction between

two types of queries over streams: querying over historical stream information, by generating RDF on-demand, and querying over live Streams. For live streams we use IBM Infosphere Streams[13], with a custom extension that allows for referring to data fields from a stream within a C-SPARQL [17] query.

To process historical stream data, we developed a REST API which transforms on-demand a time window of an archived data stream into RDF. Archived streams are stored as CSV files on a file system. The platform indexes the CSV files, and can convert a time window (potentially spanning multiple files) into an RDF Graph that is stored in an RDF strore for future use. The stream-data-to-RDF transformation process is asynchronous, because, depending on the requested time window, the processing time may be considerable.

As an example, we provide some details about the stream with information about Dublin buses. The stream provides information about approximately 600 active buses (bus line, location, delay, congestion, etc.) with updates every 20 seconds, and it is archived on a daily basis. We currently have 26 GB of bus stream data. On average, one line of a bus stream CSV file is transformed into 10 RDF triples, and we achieve a throughput of around 13000 triples / sec.

Querying. In the Semantic Web, there are three main approaches towards data integration: Query rewriting, dataset transformations (e.g. through mappings/links) and using reasoning. In QuerioCity, we take the reasoning approach, that presents the distinct advantages that the data integration is both transferable to other datasets and concise while the other two methods each present one of these advantages. In fact, integration through this method does not result in complicated queries: For example, a SPARQL query over a given Dataset would be in the form: `SELECT ... WHERE {?d rdf:type void:Dataset. ?g void:inDataset ?d. GRAPH ?g {...}}`.

Privacy. Compared to state-of-the-art platforms, an important differentiator of QuerioCity is privacy provisioning. Data privacy in QuerioCity is offered both at the dataset-level and on the graph-level (i.e., Linked Data). In particular, datasets that are uploaded to the platform undergo a semi-automated vulnerability check that aims at identifying potential privacy leaks. Based on the outcome of this process and on user input, the data is subsequently privacy-protected prior to being shared. As an example of the dataset-level privacy-protection mechanism, assume that a data publisher wishes to share the dataset (a) shown in Figure 4[14]. Applying vulnerability checking on this dataset, reveals that attributes *Position* and *Department* can lead to re-identification attacks, because their attribute-values combination is unique for some individuals. Moreover, the data publisher may indicate that *Salary* is a sensitive attribute that should be

[13] `http://www-01.ibm.com/software/data/infosphere/streams/`

[14] This is a sample of a dataset recently published online by the City of Chicago. Conforming to the US laws, the dataset lists current government employees, complete with full names, departments, position, and annual salaries. In our example, we consider a de-identified version of the dataset. The complete dataset is available at: `https://data.cityofchicago.org/Administration-Finance/Current-Employee-Names-Salaries-and-Position-Title/xzkq-xp2w`

Empl. ID	Position	Department	Salary
ID1	ADMIN. ASS	TRANSPORTN	$50280
ID2	ADMIN. ASST	INSPECTOR GEN	$70164
ID3	ADMIN. ASST	MAYOR'S OFF.	$40008
ID4	ADMIN. ASST	MAYOR'S OFF.	$62496
ID5	FIN. OFFICER	LAW	$80256
ID6	FIN. OFFICER	STREETS	$91864

(a)

Position	Department	Salary
ADMIN. ASS	*	$50K-$75K
ADMIN. ASST	*	$50K-$75K
ADMIN. ASST	MAYOR'S OFF.	$40K-$65K
ADMIN. ASST	MAYOR'S OFF.	$40K-$65K
FIN. OFFICER	LAW/STREETS	$80K-$95K
FIN. OFFICER	LAW/STREETS	$80K-$95K

(b)

Fig. 4. Example table before and after privacy protection

disclosed in ranges. Accordingly, an anonymization of this dataset, which protects individuals from re-identification attacks, can result in the dataset (b) shown in Figure 4.

Provenance. We keep both dataset-level provenance and graph-level provenance, storing *derivedFrom* relationships for both Datasets and Graphs. Graph-level provenance is tunable to the resolution required, by splitting Graphs. In the extreme case, we can keep a single graph per triple, so as to have triple-level provenance. Needless to say, this will have a negative impact on performance and we have yet to encounter the need for it. In QuerioCity, provenance is *operational* with regard to privacy. When a privacy threat is detected for a given Dataset, it to not sufficient to protect this Dataset in isolation, since the process to generate it could be repeated. We use the *derivedFrom* relations and protect *all* Datasets that were used to create the Dataset with privacy vulnerabilities.

3.3 Putting It All Together

We illustrate the design rationale through the example in Figure 5. When importing a data set into the system, we create two Datasets: Dataset A links to the Metadata (not shown in the figure) and a pointer to the URL of the source files, similar to a standard content portal. Dataset B links to the *same* metadata and the simple RDF representation described in the previous paragraph. Extracting entities would require structural changes to the RDF representation, which means that a new Graph will be generated to which the Dataset C will refer to. It is further possible to use standard datatypes (e.g. normalize spatial projections to WSG84[15]). In QuerioCity, this is accomplished by creating new properties with such values. Dataset D can refer to *both* the Graph containing the entities and a new Graph containing such new properties. Finally, we can also map Dataset D to Dataset F, to generate Dataset E. In this case, Dataset E can refer to the Graphs of Dataset D, Dataset F and a Graph containing the mappings. For each of these Datasets, we keep coarse-grain provenance information.

3.4 Deployment

In this section, we describe an internal deployment of the QuerioCity platform, consisting mainly of IBM technology, with some open-source components. The

[15] W3C Basic Geo (WGS84 lat/long) Vocabulary `www.w3.org/2003/01/geo/`

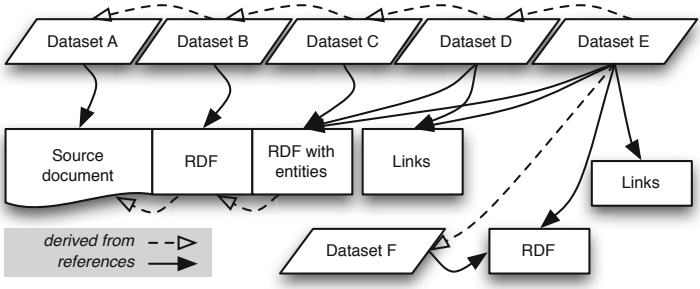

Fig. 5. Example for content data model

use well established commercial components, which can be clustered as required, ensures scalability and robustness. Figure 6 outlines the main software components of our system.

The Secure FTP (*SFTP*) server component allows publishers to securely upload multiple files for publishing. Such files are then mirrored on the Storage Area by the the *Publish & Sync App*. We use a fibre-connected *IBM Storage Array Network* to store the published data: this is secure, resilient and recoverable. *IBM InfoSphere Streams* provides the processing capabilities for live streams. The platform provides data access control using *IBM Tivoli Directory Server (TDS)*, which stores the credentials of users that can access restricted data sets or publish data sets. Users can retrieve data sets through HTTP: the platform uses *IBM HTTP Server (IHS)* to answer these requests after having checked the user credentials with *TDS*. We use *IBM WebSphere Application Server (WAS)* to host the (i) QuerioCity Data Layer, (ii) the Web user interface for searching and publishing metadata, and (iii) a set of REST APIs for searching, querying, browsing and downloading data. Finally, a SPARQL endpoint is available to work directly on RDF data, which are stored on a DB2 RDF store.

QuerioCity is based on commercial, enterprise-grade software. Critical components, such as the WAS, DB2 and the HTTP server can be clustered as required, providing scalability and robustness.

4 Lessons Learned

The infrastructure to run enterprise Semantic Web applications is finally mature. As described in Section 3.4, QuerioCity runs almost exclusively on enterprise-grade components. Nevertheless, emerging research technologies in Linked Data and semantic integration are well behind similar technologies for relational data. Selecting the appropriate ontology for describing a dataset, if it exists, and linking across datasets, are tedious tasks, which require significant expertise and lead to publication processes that are not scalable. A significant obstacle in building an urban Linked Data platform is to strike a balance between automatic approaches and user interaction to achieve continuous and incremental integration of streams.

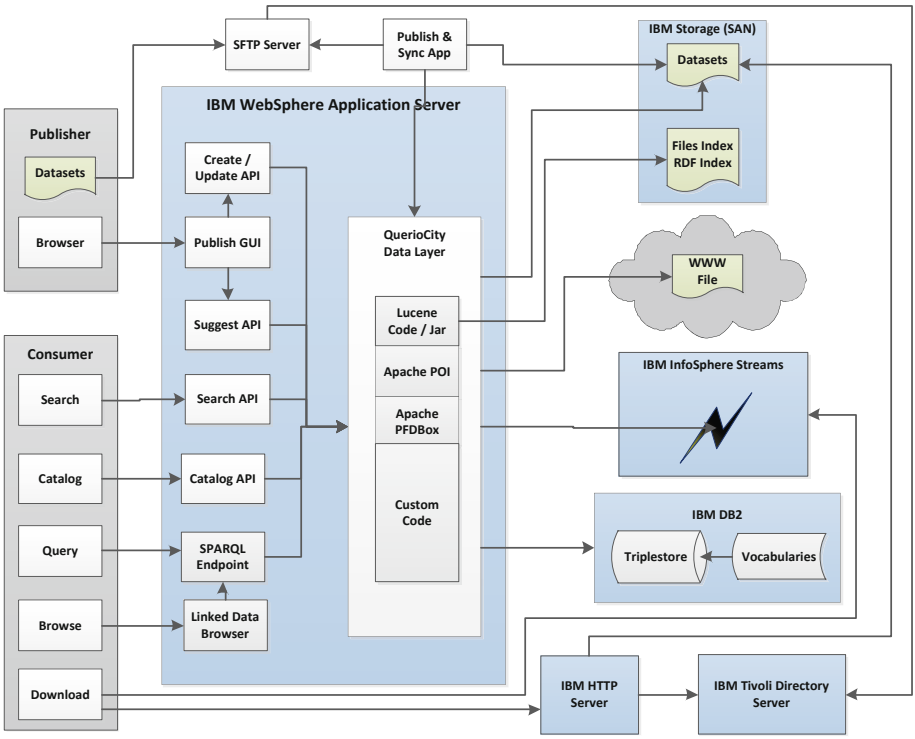

Fig. 6. Software components in QuerioCity

A pay-as-you-go semantic approach mitigates risk, since it allows publishers to partially complete and re-use integration tasks. Data need to be delivered in any shape and format with minimum cost to encourage participation and engage data providers. More work is required to find practical and inexpensive approaches to semantically annotate and RDFize the content of the collected data, provide insights into its quality and allow seamless data integration that can augment the value of the data.

In our experience, visualisations with high levels of aggregation, such as the one displayed in Figure 3, are preferable for citizen and city officials.

Visualizations are layered on SPARQL query results to enable citizens make use of and benefit from open data. As city data is situated on a specific temporal and geographical context, further insight is given by comparing datasets through spatial-temporal visualizations or heat maps (optionally points of interest can be extracted from sources such as Linked Open Street Maps [6]). By integrating diverse information sources, and making them consistently queriable, we enable the next generation of visual analytics.

The Benefits and Costs of Semantics. While relational databases deliver excellent performance and data integrity, triple stores allow for data storage without the need of prior schema definition. Thus, triple stores are more suitable than relational databases for situations where underlying data structures and schemata are changing, and where each dataset needs its own schema.

Relational schemata are not easily extensible; adding new datasets and relationships across data requires new link tables (with the foreign keys of rows to be linked) or very generic table structures. Furthermore, changes at the schema or the data definition level need to be reflected in the applications accessing the data.

QuerioCity is not the first attempt at developing an infrastructure for managing Urban Information in IBM Ireland Research Lab. Previously, we had investigated a relational schema to model data for Dublinked. This schema quickly became complicated, as it needed to be extended with semantic types and be amenable to extensions. Moreover, providing inference capabilities in a relational infrastructure proved particularly cumbersome. We got positive feedback from experienced engineers who had not worked with semantic technologies before with regard to (i) the flexibility of not having a fixed schema, (ii) the fact that SPARQL queries did not have to be adapted when the schema would change and (iii) the fact that Semantic Web technologies are in principle compatible with enterprise systems. Furthermore, the fact that RDF references are global and can be collected from many sources is proven invaluable. In contrast, the main inhibitors reported were the uncertainty over the implications of inference to the security model and the paradigmatic shift of not being able to inspect data in a relational schema.

The main drawback with using an RDF representation lies in the sparsity of the format. We report on statistics for 4959 files in CSV format from Dublinked.

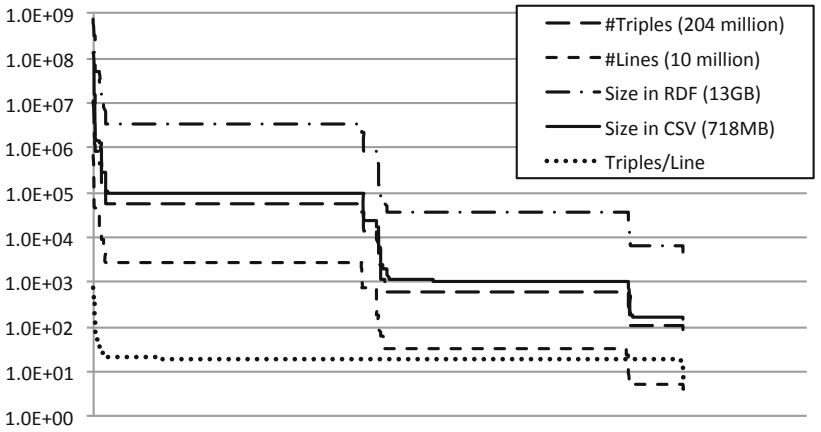

Fig. 7. Distribution of size metrics for data from Dublin. The values in parentheses represent aggregates. For readability, we have ordered values.

In Figure 7, we show the distribution of the number of triples when converted to the simple RDF format, the number of lines in the CSV files, the size of the RDF in Turtle format, the size of the input CSV files and the number of triples per CSV line. We observe that (i) the number of triples is 20 times larger than the number of lines, (ii) the size of the data in RDF format is 18 times the size of the CSV data, and (iii) the vast majority of CSV files contain between 3 and 7 columns, corresponding to 10-20 triples. Connected to this, there is also significant cost in indexing this data.

5 Conclusions and Future Work

In this paper, we presented QuerioCity, a Linked Data-based approach for managing the information of a city. The novelty of our approach lies in the flexibility of the integration, the provisioning for privacy, the efficiency of the storage model and the ability to handle streams. The main lessons learned concern tackling the domain complexity of city data and the maturity of related technologies, while the main benefits and costs of a semantic representation lie in flexibility and sparsity.

The QuerioCity platform creates an opportunity to further understand how citizens and city officials use the system and what are the typical questions they are trying to answer. Query logs obtained from users explorations and applications (SPARQL queries and consecutive HTTP requests) can be used to create models of usage to enhance the search and explorations with information more commonly sought by users, and learn how linked urban data can be exploited to answer potentially complex information needs and user queries.

In future work, we also plan to investigate an urban information management platform across cities, through the use of a federated catalog and indexes for data distributed in different cities repositories that will allow to discover, fuse and compare data and cities. Considering the complexity of the domain and the heterogeneity of the information, natural querying that scales becomes of paramount importance. With breakthroughs such as Watson allowing complex queries in natural language, we further plan to investigate methods to perform quantitative queries in an open domain.

Acknowledgments. The authors would like to thank the authorities in the Dublin area for providing datasets and NUI, Maynooth for their contribution in the development of Dublinked.

References

1. Berners-Lee, T.: Putting government data online (2009)
2. Alani, H., Dupplaw, D., Sheridan, J., O'Hara, K., Darlington, J., Shadbolt, N., Tullo, C.: Unlocking the Potential of Public Sector Information with Semantic Web Technology. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 708–721. Springer, Heidelberg (2007)

3. Maali, F., Cyganiak, R., Peristeras, V.: A Publishing Pipeline for Linked Government Data. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 778–792. Springer, Heidelberg (2012)
4. Ding, L., Lebo, T., Erickson, J.S., DiFranzo, D., Williams, G.T., Li, X., Michaelis, J., Graves, A., Zheng, J.G., Shangguan, Z., Flores, J., McGuinness, D.L., Hendler, J.: Twc logd: A portal for linked open government data ecosystems. Web Semantics (2011) (in press)
5. Sheridan, J., Tennison, J.: Linking uk government data. In: LDOW. CEUR Workshop Proceedings, vol. 628. CEUR-WS.org (2010)
6. Auer, S., Lehmann, J., Hellmann, S.: LinkedGeoData: Adding a Spatial Dimension to the Web of Data. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 731–746. Springer, Heidelberg (2009)
7. Goodwin, J., Dolbear, C., Hart, G.: Geographical linked data: The administrative geography of great britain on the semantic web. Transaction in GIS 12(1), 19–30 (2009)
8. García-Silva, A., Gómez-Pérez, A., Suárez-Figueroa, M.C., Villazón-Terrazas, B.: A Pattern Based Approach for Re-engineering Non-Ontological Resources into Ontologies. In: Domingue, J., Anutariya, C. (eds.) ASWC 2008. LNCS, vol. 5367, pp. 167–181. Springer, Heidelberg (2008)
9. Pivk, A.: Automatic ontology generation from web tabular structures. AI Communications 19, 2006 (2005)
10. Hurst, M.: Layout and language: Challenges for table understanding on the web, pp. 27–30 (2001)
11. Venetis, P., Halevy, A., Madhavan, J., Paşca, M., Shen, W., Wu, F., Miao, G., Wu, C.: Recovering semantics of tables on the web. VLDB Endow. 4(9), 528–538 (2011)
12. Han, L., Finin, T., Parr, C.S., Sachs, J., Joshi, A.: RDF123: From spreadsheets to RDF. In: Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 451–466. Springer, Heidelberg (2008)
13. Scharffe, F., Atemezing, G., Troncy, R., Gandon, F., Villata, S., Bucher, B., Hamdi, F., Bihanic, L., Kepeklian, G., Cotton, F., Euzenat, J., Fan, Z., Vandenbussche, P.-Y., Vatant, B.: Enabling linked-data publication with the datalift platform. In (AAAI 2012) Workshop on Semantic Cities (2012)
14. Zhao, J.: Open provenance model vocabulary specification. Tech. rep., University of Oxford (2010), `http://open-biomed.sourceforge.net/opmv/ns.html`
15. Le-Phuoc, D., Parreira, J., Hausenblas, M., Han, Y., Hauswirth, M.: Live linked open sensor database. In: Proceedings of the 6th International Conference on Semantic Systems, I-SEMANTICS 2010, pp. 46:1–46:4. ACM, New York (2010)
16. Fung, B.C.M., Wang, K., Chen, R., Yu, P.S.: Privacy-preserving data publishing: A survey of recent developments. ACM Comput. Survey 42(4), 14:1–14:53 (2010)
17. Barbieri, D., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: Querying rdf streams with c-sparql. SIGMOD Record 39(1), 20–26 (2010)