Citizen-empowered Linked Data Apps for Smarter Cities

Diego López-de-Ipiña1, Sacha Vanhecke2, Oscar Peña1, Tom De Nies2 and Erik Mannens2

1Deusto Institute of Technology, DeustoTech, University of Deusto, Avda. Universidades 24, 48007 Bilbao, SPAIN

2Ghent University – iMinds – Multimedia Lab

{[dipina, oscar.pena}@deusto.es](mailto:dipina,%20oscar.pena%7d@deusto.es),   
{[sacha.vanhecke,](mailto:sacha.vanhecke,%20) tom.denies, [erik.mannens}@ugent.be](mailto:erik.mannens%7d@ugent.be)

**Abstract.** Smart cities can be defined as a means of making available all the services and applications enabled by ICT to citizens, companies and authorities that are part of a city’s system. In contrast with the view of big corporations promoting holistic “smart city in a box” solutions, this work suggests that smarter cities can be achieved by simply leveraging already available infrastructure such as Open Government Data, sensor networks deployed in cities and the citizens’ smartphones. A platform, namely IES CITIES, is contributed that facilitates the generation of citizen-centric apps that exploit urban data in different domains. Moreover, user supplied data is used to complement, enrich and enhance the datasets managed by councils.

**Keywords.** Smart City, Linked Data, Apps

1. Introduction

Smart cities aim to increase citizens’ quality of life and improve the efficiency and quality of the services provided by governing entities and businesses. A truly smart city should be aware of the special needs of all its citizens, particularly those with disabilities or about to lose autonomy.

The IES Cities project is defined to promote user-centric and user-provided mobile services that exploit open data and user-supplied data. Its main contribution is to define an open Linked Data apps-enabling technological platform. Such platform will be deployed in different cities across Europe, allowing the citizens to produce and consume internet-based services (apps) based on their own and external open data related to the cities.

Something especially remarkable about IES Cities is that no project before has considered so much the extent of the impact that the users may have on improving, extending and enriching the open data in which services are usually based. The main stakeholders of our envisaged smart city-enabling platform through urban apps ecosystem are mainly the citizens, but also the SMEs and public administration of different cities.

This work follows the “Apps for Smart Cities Manifesto” [1] against the “smart city in a box” [2] solutions offered by big corporations such as IBM, CISCO or SIEMENS. The main statement behind this manifesto is the following: “The city must become like the Internet, i.e. enabling creative development and easy deployment of applications which aim to empower the citizen”. It highlights five “technologies that matter” for cities in 2020: mobile broadband; smart personal devices; government-sponsored cloud; open-source public databases to promote grassroots innovation, and “public interfaces.”

The IES CITIES platform addresses three main scientific challenges: a) extract and adapt heterogeneous structured and non-structured data from council repositories, web sites and social networks, b) validate, promote and integrate user-provided data with open government data and c) facilitate the development of urban apps by end developers. In order to achieve this, it defines a mobile application that enables users to get an overview of available mobile web services based on their location and execute these services. Significantly, the platform does not only enable the consumption of the open data, but also allows users to validate it and to provide their own, supported by a solid mechanism for keeping track of provenance information.

The structure of the paper is as follows. Section 2 talks about related work. Section 3 details the IES CITIES platform architecture.

1. Related work

This work seeks user participation and contribution to a city’s datasets or knowledge base by enabling them to contribute with data through their smartphones’ urban (IES CITIES-compatible) apps. However, the quality of such data may significantly vary from a citizen to other. It is important to be able to assess and value the provided data promoting valuable and trustable information and decrementing and eventually removing low value data. The W3C has created the PROV Data Model [3] for provenance interchange on the web. The provenance of digital objects represents their origins. PROV is a specification to express provenance records, which contain descriptions of the entities and activities involved in producing and delivering or otherwise influencing a given object. Provenance can be used for many purposes, such as understanding how data was collected so it can be meaningfully used, determining ownership and rights over an object, making judgements about information to determine whether to trust it, verifying that the process and steps used to obtain a result complies with given requirements, and reproducing how something was generated [9]. Furthermore, PROV was deliberately kept extensible, allowing various extended concepts and custom attributes to be used. For example, the Uncertainty Provenance (UP) [10] set of attributes can be used to model the uncertainty of data, aggregated from heterogeneously divided trusted and untrusted sources, or with varying confidence.

Lately, some JSON query languages [4] have emerged that intend to enable end-users to more easily exploit the JSON data model. In our view, urban apps will be assembled from structured and non-structured data in the form of RDF, CSV or even HTML pages. However, information in such models can be mapped into JSON, a *lingua franca*, for web and mobile developers. Therefore, an assumption of this work is to consider that query for data may be specified as JSON.

TheDataTank (TDT) [5] is a distributed open-source web framework that serves as a dataset adaptor to transform online data of different formats into a user-friendly Restful API.

Talk about provenance data, JSON query APIs, frameworks to extract structured, semi-structured and unstructured data. Other manifestation of the SmartCity manifesto. mCiudad, MUGGES, uService

ADAPTA

The DataTank

Crowdsourcing

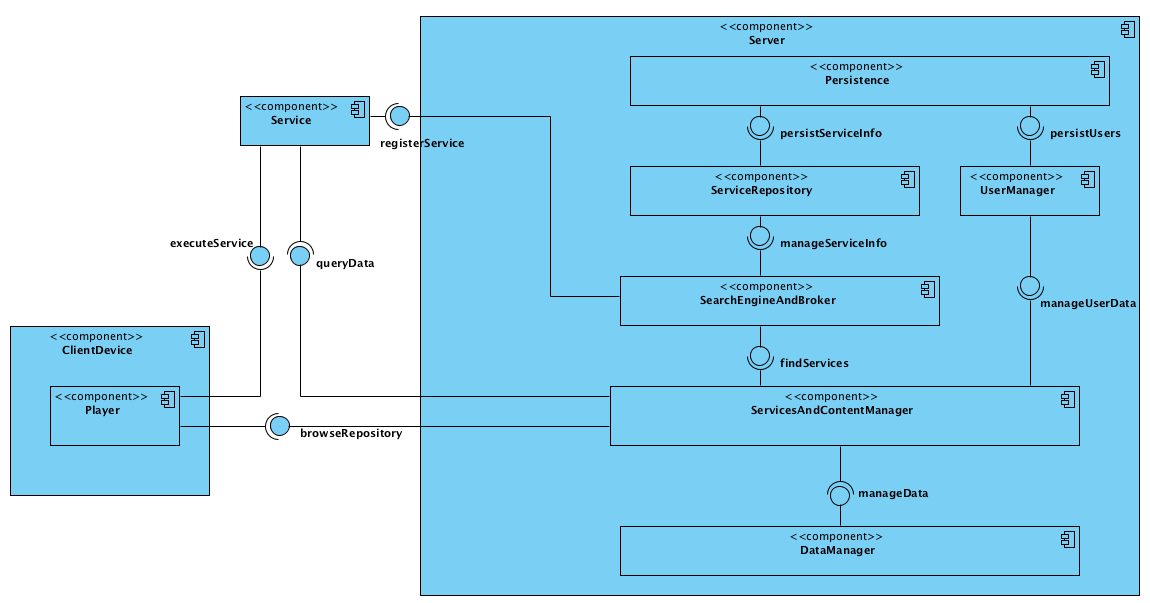
1. The IES CITIES platform

On the highest level, the Web Service architecture is applied and the IES Cities platform is divided in three parts, namely the core server, the IES Cities Player client, and IES Cities Services developed by any third party. The IES CITIES platform consists of two main components, namely the server and the client. The main role of the server is to enable the retrieval and the storage of data provided by both users and public infrastructures, through a set of services. These services can be predefined, but the goal is to motivate independent developers to develop and register new services as new datasets are added to the platform. In order to keep track of both the registered IES Cities Services and users, the server also has to persist any related information. Finally, it allows users to find and access services based on a degree of relevancy. The client, an application that has to be installed on the user's mobile device, serves as the communication portal to browse and run services. From this description, the following actors can be associated with the IES Cities platform:

* *User*. The citizens that will use the IES Cities Player to execute IES Cities Services.
* *Service Developer*. Developers of new IES Cities Services, no necessarily related to the datasets they want to use.
* *City Councils*. Public administrations providing the open datasets. They are considered the main data publishers of the platform.
* *IES Cities Service*. Any web applications that will actually perform operations on the open data and make it useful to end-users.

The IES CITIES architecture is depicted in Fig. 1, comprising the following components:

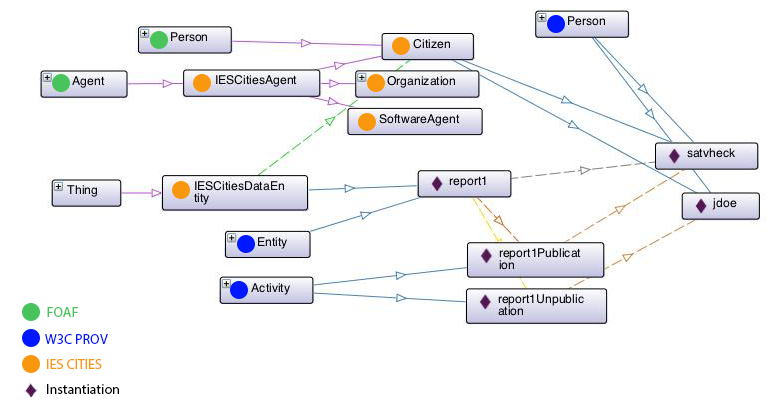
* *Player*. This is the component running on the client device. It is responsible for the user interaction, allowing both service browsing and execution.
* *ServiceAndContentManager*. This will be the main entry point to the platform's core functionality, offering an interface for both application and data querying. Using these interfaces, clients are able to access information about available services, and services can perform queries on all active datasets.
* *ServiceRepository.* The repository is used for fetching and storing information about IES Cities Services.
* *UserManager.* This component is responsible for managing any information related to users. In the future, it might also track usage of the platform by gathering activity logs for each individual service, for example to allow billing for certain smart city services.
* *SearchEngineAndBroker.* Together, the Broker and the Search Engine components allow for service registration and retrieval, the latter possibly according to a set of parameters.
* *PersistenceManager.* This component will manage all information concerning users and services that needs to be persisted permanently. This can be done in a file or a database.
* *DataManager.* This component is responsible for the communication with the publication engine and the Linked Data storage server. In the future, this component should allow the translation of non-structured information from different external sources. This component also keeps track of the provenance of the information gathered from different sources.



**Fig. 1.** Architectural Design

Internally, IES CITIES maintains data about the following entities:

* *ServiceInfo* entities contain all necessary information regarding registered IES Cities Services. Its main properties are the name of the service, a description about it, and its physical location, which is required for executing it. Each service can also be associated with a certain area of a city in order to allow retrieving it based on the detected location of the user's device. In order to find services using the search engine, there is a mapping between services and a set of keywords.
* *User* entity represents the users of the platform which are citizens, system administrators, public bodies and service developers. This entity consists of a first name, a last name, a password and a unique username
* *Review* entity allows users to write their opinion about a service. It consists of both a service's ID and a user's ID, a rating, e.g. between 1 and 5, and a brief comment.
* *Group* entity allows assigning roles such as “user”, “admin” or “developer”.



**Fig. 2.** IES Cities RDF schema

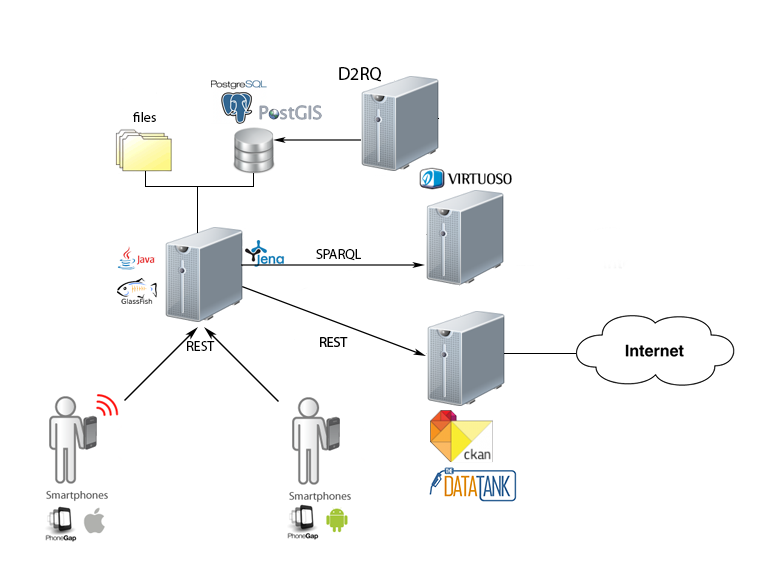
* 1. Linked Data Design

Involving end-users in the creation of Linked Open Data implies that this data should be published according to some RDF specifications, adding some particular characteristics to the data that associate it to the IES Cities platform, and enforcing its reusability inside or outside the context of the IES Cities platform. Since the reuse of existing terms is highly desirable, several widely used vocabularies were considered, such as Dublin Core, which defines general metadata attributes such as title, creator, date and subject, and FOAF, that defines terms for describing persons, their activities and their relations to other people and objects. Reusing terms maximizes the probability that data can be consumed by applications that may be tuned to such well-known vocabularies, without requiring further pre-processing of the data or modification of the application[6].

The IES Cities RDF Schema (see Fig. 2) defines two general classes. Firstly, the base for any data generated through the platform is the class IESCitiesDataEntity, which should at least include a DC terms identifier property to easily query the data and if possible a description. Secondly, in order to incorporate the users of the platform in the model, a general IESCitiesAgent class is defined, that extends FOAF’s Agent, and several subclasses such as Citizen, inheriting from FOAF’s Person class, Organization and SoftwareAgent.

Furthermore, because of the involvement of third parties in the management of open datasets, it is necessary to incorporate metadata that enables tracking down the creator of a particular data entity, known as provenance data. A widely deployed vocabulary that allows representing basic provenance data is Dublin Core. However, a more complete provenance ontology that better fits the needs of the IES Cities project by separating the data from actual creation, modification and invalidation processes, is the PROV Ontology [7]. The PROV-O terms (classes and properties) are grouped into three categories of incremental complexity, which facilitates the process of making a comprehensive provenance model. These categories are: Starting Point terms, Expanded terms, and terms for Qualifying relationships. Starting Point classes, such as Entity, Activity and Agent, provide the basis for the rest of the PROV Ontology. Expanded terms provide additional terms that can be used to relate classes. These two categories are used in the IES Cities vocabulary to associate users as the publishers or invalidators of particular data. Finally, Qualifying terms provide more detailed information about the binary relations between Starting Point and Expanded properties, enabling the persistence of more information about the reason for an invalidation activity, such as a comment and a corresponding time stamp.

Other topics that are strongly related to provenance are credibility of data and trustworthiness of its author. By adding these concepts to IES Cities datasets, it becomes possible to prioritize the processing of certain data over less reliable data. For the designed data model, a similar idea is used as for the Trust Ontology[[1]](#footnote-1), which rates trustworthiness on a scale of 1 to 10. A neutral credibility score is assigned to each IESCitiesDataEntity at initialization, which increments or decrements by one each time a citizen votes the corresponding data up or down respectively. This enables for example to invalidate data in an automated manner whenever a lower limit for the data’s credibility is reached



**Fig. 3.** Deployment of IES CITIES platform

* 1. Implementation details

Fig. 3 shows the actual realization of the IES CITIES platform. Using a native looking application, users use their mobile device to access the services registered in the platform. All information about the services and the users of the IES Cities platform is persisted in a database, such as PostgreSQL, and all the open datasets provided by city councils will be accessible through a publication engine such as CKAN4[[2]](#footnote-2) or TheDataTank5[[3]](#footnote-3). The core server will access these publication engines each time a particular dataset is requested in order to resolve its location. For this work, a Virtuoso RDF store was installed in order to maintain some example linked datasets and to store data generated and validated using the IES Cities platform.

Fetching open data is realized by defining a generic JSONformatted query and sending it to the server’s RESTful “/data/” interface, together with the name of the requested datasets. This JSON query mainly consists of key/value pairs to specify required fields and optionally some parameters. The DataManager component will take care of resolving the location of the requested dataset from the publication engines, after which it will transform the JSON query into a query language specific one, depending on the nature of the data’s resolved endpoint. Currently, a query builder has been implemented for both SPARQL, the query language for RDF, and SPECTQL, the query language used by TheDataTank. After executing the query over the endpoint, the resulting data is returned, in JSON format. An example of the used JSON queries is shown below:

{

"type":"data",

"requested”:{ "predicate1":"object1", "predicate2":"object2" },

"optional" :{ "predicate3":"object3" },

"given" :{ "predicate2":{ "type":"string", "value":"object2\_value" }}

}

A similar mechanism is also used for the generation and validation of data, using the SPARQL/Update extension and adding provenance meta-data compliant to the PROV-O vocabulary. This makes it, however, only available for linked datasets.

The client application, the IES Cities Player, was developed using PhoneGap6[[4]](#footnote-4), which serves as a wrapper for applications that are purely based on standard web-based technologies. It currently supports 7 mobile platforms, the main ones being iOS, Android andWindows Mobile. The GUI of the player was completely written in HTML5 and CSS3, using the JavaScript libraries jQuery and jQuery Mobile. Using AJAX, simple HTTP requests are sent to the RESTful “/service/” interface of the IES Cities server. This interface returns information in JSON format, which in turn can easily be parsed using basic JavaScript.

D:\Deusto\Docs\UCAmI2013\SearchingViewingServices.tif

**Fig. 4.** Searching and browsing servicers.

The overview of services based on the user’s location and ordered by average rating is shown in Fig. 4 (left hand side). Apart from location based lookup, users can also browse services using based on a set of keywords. As the user selects a service, another request is sent to get more specific information on it. A description is printed to the screen, along with a list of reviews by other users and the average rating (right hand side of Fig. 4). By clicking the Start button, the player will redirect the user to the actual service.

1. Validation

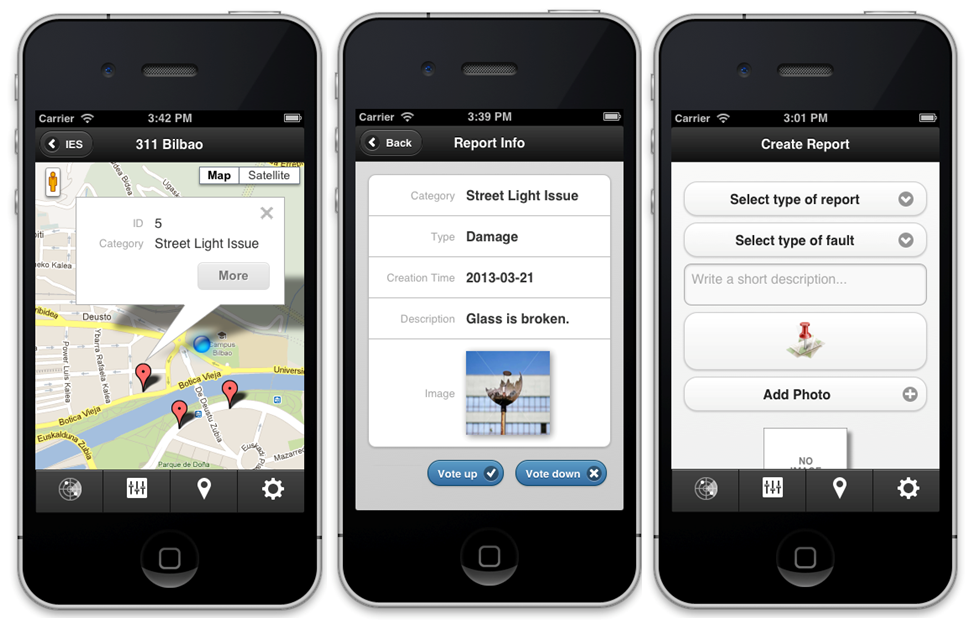
Two proof-of-concept IES Cities Services were implemented and deployed on a local network to evaluate the IES Cities platform’s usability for users, developers and city councils (data providers). The first service, 311 Bilbao (Fig. 5), uses Linked Open Data to get an overview of reports of faults in public infrastructure. It demonstrates how a developer can create complex mobile web services relying on semantic data, without technical knowledge of the query language SPARQL. He only needs to create a query using in the generic, well-defined JSON format discussed in the previous section, and send it to the IES Cities platform’s RESTful “/data/” interface. This interface will return the result using the JSON format as well.

During initialization, the service queries for reported faults, and displays the result on a map (a). This is achieved using the Google Maps JavaScript API. The requested information is kept minimal in order to minimize performance losses. Using the filter functionality on the second tab (b), a user can choose to see reports of only a certain type. By clicking on the marker of a particular report, the ID and the underlying nature of the reported fault is displayed. Meanwhile, more information about the report is also being queried in the background. By the time the user decides to inspect the report, all information is already retrieved, allowing the information page (c) to be displayed instantaneously. On this page, users can read the full description of the report and watch a full screen image, if any. Here, they are also able to vote the credibility of the report up or down. Finally, they can create their own reports (d), specifying the faults nature, its location, a description and optionally an image, either using their built-in camera or the device’s library.

From the data owner’s point of view, enrichment of their datasets by third parties, such as users of the 311 Bilbao application, revealed two problems. The first one is the fact that data does not need to be approved before being published and that there is no mechanism to control the amount of data a citizen can add. Intelligisense techniques can be used so that the user is suggested related reports earlier submitted before submitting its own.

This makes it impossible to prevent malicious users from polluting the datasets with garbage data. Other citizens can vote invalid data down, but the data will only actually be declared invalid once a lower limit of its credibility is reached.

Secondly, there is still the need for a way to differentiate the default trustworthiness of the different authors such as citizens and city councils. This might be important in the future, if the owner of the dataset wants to prioritize the processing of data from particular sources.



**Fig. 5.** 311 Bilbao IES CITIES service.

The second service, AirQual Madrid, demonstrates the added value of using The DataTank as a publication engine for IES Cities. For this service as well, the source data, now published in CSV format, could instantaneously be queried using the generic JSON format through the RESTful interfaces.

Portability evaluation was done by deploying the PhoneGap application on both iOS 5 (simulator) and Android 4.1 (Samsung Galaxy SIII mini), proving though that several of the most recent versions of PhoneGap still have different compatibility issues with both operating systems. A brief evaluation of performance was carried out as well, revealing the transmission time between services and mobile devices to be the bottleneck of the system, but still resulting in execution times below a second.



**Fig. 6.** AirQual IES CITIES service.

1. Conclusion and Further Work

This work has shown that the IES CITIES platform can be used by developers to facilitate the use of open data from arbitrary formats, relying on CKAN and custom data parsers or on the even easier to use adaptor functionality of The DataTank. Moreover, the easy to use mobile web services demonstrate how citizens’ involvement in the management of a city can be increased by allowing them to actively participate in the creation of new data and the validation of existing open data. These services also show the IES Cities platform’s added value for public bodies, who can easily publish their open data in different non-proprietary formats, while making them accessible as common machine-readable formats through uniform REST interfaces

Further work should address the following issues. Current use of provenance data might still not be comprehensive enough. More complex, derived provenance information should be incorporated, allowing to trace back the full revision chain of data modifications. Recent research has shown that the reconstruction of provenance, when it is (partially) missing, is feasible [11] and can be incorporated to complement our tracking approach. The current mechanism for defining credibility should also be extended. Instead of only setting a default score at creation time of the data, some level of ‘trust’ should be calculated as well, in a principled and algorithmic manner, based on the data’s actual source and its changes [8], making it possible to prioritize the processing of data of more trusted parties, such as official public bodies, over data provided by citizens. The reputation of certain sources is to be calculated over time, and used to set an initial value for a source’s credibility. However, in addition to reputation, provenance is an essential component towards trust assessment as well [12]. Applying automatic reasoning over the provenance of the data, seems a feasible method to implement these principles, as it only requires the addition of lightweight annotations to the provenance that is currently tracked [10]. These annotations allow for an on-the-fly selection of data, based on the preference and trustworthiness of its source trace. Lastly, the privacy concerns should be dealt with in an appropriate manner, for example by defining a comprehensive “Terms of service” document presented before registering in the IES Cities platform or opening a service, notifying the user of what data will be published.

References

1. The Apps for Smart Cities Manifesto, <http://www.appsforsmartcities.com/?q=manifesto>, last accessed: 10/07/2013
2. Townsend, A., Maguire, R., Liebhold, M. and Crawford, M. (2010). A planet of civic laboratories: The future of cities, information, and inclusion. Institute for the Future, Technology Horizons Program, 124 University Ave., 2nd fl. Palo Alto, CA, USA. Last accessed 18 January 2011.
3. W3C, PROV Model Primer. Working Group Note 30 April 2013, <http://www.w3.org/TR/2013/NOTE-prov-primer-20130430/>
4. JSONiq – The JSON Query Language, <http://www.jsoniq.org/>, last accessed 11/07/2013
5. The Data Tank – an open data management systems maintained by OKFN Belgium, <http://thedatatank.com/>
6. Tom Heath and Christian Bizer, Linked Data: Evolving the Web into a Global Data Space, Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, first edition, 2011.
7. Timothy Lebo, Satya Sahoo, Deborah McGuinness, and eds., “Prov-o: The prov ontology,” http://www.w3.org/TR/2013/REC-prov-o-20130430/, April 2013, Accessed: 2013-05-10.
8. Harry Halpin, “Provenance: The missing component of the semantic web for privacy and trust,” in Trust and Privacy on the Social and Semantic Web (SPOT2009), workshop of ESWC, 2009.
9. Paul Groth and Yolanda Gil, Editorial - Using Provenance in the Semantic Web, in Web Semantics: Science, Services and Agents on the World Wide Web, Elsevier, 2011
10. De Nies, Tom, Sam Coppens, Erik Mannens, and Rik Van de Walle. "Modeling uncertain provenance and provenance of uncertainty in W3C PROV." In *Proceedings of the 22nd international conference on World Wide Web companion*, pp. 167-168. International World Wide Web Conferences Steering Committee, 2013.
11. Magliacane, Sara. "Reconstructing provenance." In *The Semantic Web–ISWC 2012*, pp. 399-406. Springer Berlin Heidelberg, 2012
12. Ceolin, Davide, Paul T. Groth, Willem Robert van Hage, Archana Nottamkandath, and Wan Fokkink. "Trust Evaluation through User Reputation and Provenance Analysis." In *URSW*, pp. 15-26. 2012

1. http://trust.mindswap.org/trustOnt.shtml [↑](#footnote-ref-1)
2. http://ckan.org [↑](#footnote-ref-2)
3. http://thedatatank.com [↑](#footnote-ref-3)
4. http://phonegap.com/ [↑](#footnote-ref-4)