OPEN DATA FOR THE MASSES
Unleashing Personal Data into the Wild

Giuseppe Ciaccio and Marina Ribaudo
DISI, Università di Genova, Via Dodecaneso 35, Genova, Italy

Keywords: Open Data, Open API, Government 2.0, OAuth.

Abstract: In these years the public administration is undergoing a deep transformation, driven by a greater demand for transparency and efficiency in a participative framework involving nonprofit organizations, businesses, and citizens, with the modern network infrastructures as a common medium. The Open Data movement is considered to be one of the keys of this change. In this position paper we argue that the work done so far in the Open Data field, i.e. offering massive public datasets, is just a preliminary answer. We argue that a few open standards concerning online authorization, access control, and data exportation, are emerging; these standards, if adopted by the public administration (but also business companies and any other organization), will trigger the release of a much wider and more useful wave of open data, able to sustain a new generation of helpful online personalized services, based on personal data whose ownership is given back to individual people.

1 INTRODUCTION

The past 15 years have seen a fundamental transition from the ”traditional” web to the so called Web 2.0 (T. O’Reilly, 2005), showing the power of the users in co-creating new content and adding value to already existing knowledge. Web 2.0 suggestions and the idea of “Web-as-participation-platform” have changed different aspects of our everyday life. We recall, for instance, the educational context where the term *e-learning 2.0* has been coined to stress the need of changes in the pedagogical models for the students of the new millennium, or in the media sector where social networking and media-sharing websites, and the increasing prevalence of mobile devices, have made news available to people almost in real time all over the world.

Also the public administration is not immune from this participatory scenario and terms like *e-democracy* or *Government 2.0* have been coined aiming for a broader and more active citizen participation in today’s representative democracy. The memorandum on *Trasparency and Open Government* signed by President Obama (Obama, 2009) has fostered a new era for the public sector in which *transparency*, *participation*, and *collaboration* should become central in the democratic decision process. An example is the bottom-up approach in which government agencies place high value on the collaboration with citizens, supporting the learning from the crowds (Surowiecki, 2005). One term used in this case is *citizen-sourcing* and we refer to (Nam, 2012) for a discussion of this matter.

The use of open standards to foster innovation, the proposal of simple solutions that can autonomously evolve thanks to the community of developers, the design for participation, the disclosure of information that the public can easily access, all of these issues are currently being discussed. Open Data, in particular, is considered the coin of the Internet business model and the public sector should produce data, making it available to the private sector and to citizens who have the economic interest, the technical competencies, and the creativity to show how different data can be put in use to build new applications and services.

Several public administrations have started releasing massive datasets as Open Data on their websites. The first catalog of Open Data has been published in May 2009 by the United States Government[1], followed by the experience of the British Government[2] with the release of its national portal in September 2009. Some Open Data portals have been launched in Italy too. We mention here the case of the administration of the Piedmont region, Italy, and its portal[3] offering political data, traffic and local transportation

[1]http://data.gov
[2]http://data.gov.uk
[3]http://www.dati.piemonte.it

Ciaccio G. and Ribaudo M..
OPEN DATA FOR THE MASSES - Unleashing Personal Data into the Wild.
DOI: 10.5220/0003960502010206
In *Proceedings of the 8th International Conference on Web Information Systems and Technologies* (WEBIST-2012), pages 201-206
ISBN: 978-989-8565-08-2
Copyright © 2012 SCITEPRESS (Science and Technology Publications, Lda.)

data, touristic and cultural data, cartographic maps, mostly in the form of comma-separated values or geospatial vector data in the case of geographic information. To the best of our knowledge, the datasets made available at the portal have static character (that is, they will not reflect changes occurred after the release date) and are of two kinds, namely: aggregated and anonymized data (e.g. number of children for each school in the region); and identification data of public entities (e.g. names and addresses of restaurants in the region). We believe this is a general issue affecting the whole current generation of Open Data. Indeed no data concerning individual people may been released, because of obvious privacy reasons. On the other hand, data of public entities (e.g. list of restaurants) are already available on the web, although possibly not in an organized form. Thus, the only new thing in the current wave of Open Data is the massive release of aggregated anonymized data, which can be of great interest for statistical reasons but of limited use for building services. In our opinion, the lack of data concerning individual people, along with the static nature of the datasets, are weakness points of the current generation Open Data; without personal data and without "freshness", it is indeed impossible to build useful services tailored to the actual needs of a given individual at a given time[4].

At this point, we wondered about what would be a useful complement to the Open Data idea. Personal and sensitive data cannot be released without access control and online permission by the individual. If we need data that are always "fresh", we need to abandon the concept of dataset being released once for ever. It would then make much more sense to leave the data where they are, namely, the back ends of the numerous websites, and let the webservers export them to the web using some sort of accepted standard at the front end. As we will discuss in this position paper, we do not need to devise anything really new: it is just a matter of leveraging existing technologies and standards, and then the massive amount of personal and even sensitive data concerning individuals would become open, thus available for a new wave of innovative and personalized services, yet preserving an acceptable degree of privacy.

The key point is that there now exist standard technologies for online authorization, by which an individual can exert access control over personal data regardless of the physical location where the data are actually stored and managed; it is only necessary that the manager of data (a public administration dealing with citizen data, but also a business company managing client data) conforms to these authorization technologies and APIs. Another point is that there now exist mature technologies for representing and exporting data items (e.g. records of a back end database). A third point is that there would be no need to change the internal organization of data at the back ends, so no need for coordination among the many entities that currently manage our personal data. Just glue these pieces together, and get what we call the "Open Data for the masses".

Last but not least, with these technologies each individual would regain ownership on personal data, after decades in which "the owner" could only be the same as "the manager"; this is very important for a true ecosystem of online services to grow, free from the monopolistic control of data managers improperly acting as owners.

This position paper is structured as follows. Section 2 briefly summarizes the current standard ingredients of the Web 2.0. Section 3 describes the new emerging standards and technologies for data exportation, authorization and access control; access control by user permission, based on cryptographic credentials, is the ingredient the lack of which whould make it impossible to safely export personal data to the web. Finally, in Section 5 the paper concludes suggesting what we consider the open points to be addressed by near-term research in order for the proposed approach to become effective.

2 WEB 2.0 TECHNOLOGIES

Data Formats, Web API. In order to go over the HTML page building block, new and more generic languages to model structured data, independently from their future usage, such as XML and JSON, have been standardised. Using HTTP as a communication protocol, each website can nowadays export its data and services by offering its own set of web APIs, accessible to other software developers.

SOA, REST. The interaction of different applications, possibly running on heterogeneous architectures, has led at the beginning of the 2000 to the development of the so-called *Service-Oriented Architecture* (SOA), Amazon Web Services[5] being the first case of success with the introduction of its cloud computing platform in which e-commerce applications are built by means of separate cooperating services.

[4]It is not by chance, that the *Apps For Italy* developer contest (http:/www.apps4italy.org), calling for interesting Open Data applications, has shifted the currently open submission deadline from February 10th to April 30th, 2012.

[5]http://aws.amazon.com

A lightweight alternative to SOA is the REST architectural style (Fielding, 2000) that has been developed in parallel with the version 1.1 of the HTTP protocol and it is therefore deeply influenced from the architecture of the web.

Mashup. The availability of web APIs has had a strong impact on the creation and fruition of new information following a mashup approach. On the one hand, we have assisted to the growth of websites built by software developers easily integrating data coming from different sources instead of building them by themselves. On the other hand, by exploiting third parties APIs new applications have emerged. We recall here Facebook applications which can be executed on remote servers or directly on the clients (e.g., in form of Java applets or iPhone and Android apps).

Towards Web 3.0. Quoting (T. O'Reilly, 2005) *"Data is the Next Intel Inside"*, we live in the era of the *big data*, datasets that grow so large and are so hetereogeneous that it becomes difficult to work with them using traditional database management tools. Massively parallel software running on tens, hundreds, or even thousands of servers can be used to cope with the size of the available datasets. Moreover, "intelligent" techniques are required to move beyond a mere syntactic use of this volume of data, not necessarily stored in searchable databases. The web is (slowly) evolving towards the so called *Semantic Web* whose resources (HTML pages, files of different formats, services) have an associated semantics defined via explicit metadata. Emerging technologies in this context are the *Resource Description Framework* (RDF, nd), a standard model for data interchange on the web, with the associated query language (SPARQL, 2004), the *Web Ontology Language* (OWL, 2004), and the *Linked Data* (Berners-Lee, 2006) describing a method of publishing structured data so that it can be interlinked and become more useful.

3 TOWARDS A STANDARD FOR WEB DATA ACCESS

Standardizing the web APIs being offered by websites seems to be infeasible as a goal. A lesser goal, which has partly been achieved, concerns the standardization of a typical subset of any web API, namely, the API subset for data access. Experience in the field of web application shows that, like with any information system, it is almost always necessary that a web APIs for data access supports the following basic operations: read a subset of data items that match a rule given as a simple logical expression, read only some attributes from a given subset of data items, replace a given data item with a modified version of itself, modify some attributes of a given data item, delete a given data item. Data items may be represented by one of the languages now established as a standard in the field and, basically, they are XML documents or JSON objects.

But this is not enough: each data item must be given a unique "name", if we want to give meaning to operations like "delete a given data item". Following the REST approach, item names should be URIs in the same way as any other resource name on the web.

3.1 Google GData and Microsoft OData

Among the various options for associating a URI with an XML document, an interesting approach is the one already in use for representing RSS or Atom feeds (RSS, nd; ATOM, 2005). These *feeds* are brief summaries of fresh items recently appeared on a website. Each *feed* appears as an XML document exported by a website; it contains a brief summary of a fresh item appeared on the website plus the URI where the full item can be accessed. Modern browsers read feeds from websites selected by the user; in this way, the user will get the latest news appeared on her preferred websites without explicitly searching them. Feeds may also be imported from other websites, so that they can cite an item appeared elsewhere and point the original website for the full item, an activity known as *syndication*.

In 2007 Google created a web API, inspired to Atom, aimed at exporting, querying, and modifying data items; such an API has been then integrated into the web APIs of many Google's services (Google Calendar, Google Analytics, Google Maps, Picasa, and others). This web API is known as *Google Data Protocol* (GData, 2007).

With GData it is possible to represent structured data with XML or JSON, associate a unique URI to each data item, select items by substring match and extract them in full or in part, create new data items, replace/modify/delete items identified by URI[6].

Google has also created useful documentation for GData developers, plus a set of library stubs for the main web programming frameworks. A GData module for the Drupal CMS exists as well (M. Cotterell, 2008). GData is currently used by Google services

[6]The usage example reported at http://code.google.com/apis/gdata/docs/2.0/basics.html is very illustrative of the basic GData features.

but the specification has been released under a licence that grants free use to anybody (GDataLicence, nd); the goal seems thus to push towards establishing an open standard of general use on the web.

In 2010 Microsoft (E. Minardi, 2010) has proposed the *Open Data Protocol*, a web API conceptually similar to GData[7], whose specification is publicly available for implementation under a license called "Microsoft Open Specification Promise" (ODataLicence, nd). In this case too it seems that the goal is to contribute to some kind of standard web API for data access.

3.2 Data Access Upon Authorization

Read access to data classified as "sensitive" or even just "personal" raises obious privacy concerns; access in write mode to whatever kind of data poses additional security issues, related to integrity of information stored in the back end. All these concerns have greatly hampered exporting data of this kind to the web so far.

Most of the fear surrounding the exportation of personal or sensitive data by web servers is rooted in a poor knowledge of the existing cryptographical tools and algorithms which, if properly implemented and used, allow to attain a security level comparable to the one of a system disconnected from the network. The privacy and security requirements, indeed, can be fulfilled by submitting suitable credentials for access control along with data requests, and using session encryption against eavesdropping and man-in-the-middle attacks.

Session encryption is performed routinely by HTTPS. Access credential can be implemented by means of *cryptographic tokens* issued by an authorization server, and carrying unforgeable grants valid for accessing a given resource within a possibly limited time span. The authorization technology is now mature and there are indeed some standardization attempts. Google, for example, has chosen to conform to the OAuth authorization protocol (OAuth, 2006; OAuth2.0, 2010; E. Hammer-Lahav, 2010; OAuthExample, nd; GDataOAuth, nd), thus showing an interest in establishing OAuth as an open authorization standard. Facebook is another such example.

With OAuth 2.0, the authorization server is logically distinct from the server providing the web API for data access. An application that needs to query a data server must first have registered itself with the data server, then it has to digitally sign any request to that server's web API and provide, along the request, an authorization token valid for that server and the specific data resource being queried; such a token must have previously been obtained by the authorization server, which may release it upon verification of the (permanent, or one-time) permission by the party who the data (in this case, sensitive or personal data) refer to. It is thus possible to separate the data *manager* (e.g. the administrator of the data server) from the data *owner* (e.g. a person, citizen, business enterprise or client). The owner can give permission for specific applications to access given data items on given data servers for a given amount of time and according to specified access patterns (e.g. only for read, or read/modify, or full access) without revealing any authentication credential (like username or password) to the application or the data manager. Only the authorization server needs to get authentication credential from the user, in order to assess the validity of the permissions.

More authorization servers may exist, of course, under control of as many independent administration domains, but they could coexist within a single federated authentication domain by means of the OpenID standard (OpenID, nd; OpenIDGoogle, nd) or *Single Sign-On* approaches.

3.3 Open Data for the Masses

The separation between data manager and data owner allows to extend the realm of Open Data to the huge domain of personal, and even sensitive, data items; such data can well become Open without becoming "Public" because, thanks to an authorization server, there may exist an *owner* who ultimately grants or denies access permission to the data items selectively, and independently of the possibly many administration domains where the data are physically stored and logically managed.

Some innovative use cases become suddenly feasible. For instance, let us consider individual medical data, also known as *electronic health records* (EHR), whose adoption is currently being strongly promoted by the Obama Administration in the United States[8]. By implementing EHRs as Open Data with authorization, a person may grant access permission for her own EHR to applications run by public hospitals while denying permission to insurance companies. Another use case might be found the domain of income and taxation, where it would be very useful to have an application for automatic statement of income. Ideally, data items concerning income of a citizen could be exported by the employer, from a

[7]A sufficiently clear example of usage is reported at http://www.odata.org/developers/protocols/operations.

[8]https://www.cms.gov/EHRIncentivePrograms

server of his own as well as a third party data centre; data items concerning, say, real estate or other properties owned by the citizen and subject to taxation could be exported by a public registry; and, say, data items concerning deductible medical expenses could be exported by the respective hospitals or specialists who provided the healthcare service or medical intervention. All of these items could be exported as Open Data subject to access permission by the owner (the citizen, and a few public authorities). With the one-time consent of the citizen, an application run by the citizen herself or by an accountant of her choice could access these data and help building a statement of income for the citizen. The same data could also be exploited for auditing purposes by the public tax authority, who could be just one of the owners of the data items in addition to the citizen. Final income data, once submitted to the office of tax and revenue, could be in turn exported as Open Data by the office of tax itself and exploited for computing the due balance of those social services whose fare depends on the stated income of the citizen.

To sum up, by coupling an Open Data approach with a suitable authorization technique, it would be possible to deploy personalized and thus much more helpful online services that need, or make only sense with, personal and sensitive data of individual users. This would open to supporting a great number of routinary tasks of various kinds (bureauocratic, commercial, informational, or merely recreational), even involving a number of distinct and independent parties. The only effort required to each party is to overcome any unmotivated security/privacy anxiety and unleash data items “into the wild”, adhering to a standard web API and a standard authorization scheme (like GData and OAuth). The back end organization of data need not be changed in any way; this is another point of strenght of this approach, as the various data managers would keep their reciprocal independency.

4 RELATED WORK

In the public administration domain we could not find proposals similar to our for delivering services built upon personal or sensitive data and using already available open standards.

The work in (Wallis et al., 2011) proposes the separation of data (personal information and user-generated content) from the web applications willing to use them. This is made possible thanks to the introduction of a model of a distributed online data storage. The application domain is that of Web 2.0 applications: instead of filling multiple user profile forms to grant access to different services, users should store their personal data onto a possibly distributed data storage, responsible of maintaining a single fresh copy of the data. Upon authorization of data owner, this storage service can provide access grant to web applications following a publish/subscribe paradigm and thanks to the definition of an API offered (1) to users to store/modify their data and (2) to web applications to access (in read mode) to the same data upon authorization, as specified by specific policy rules that define what a given web application can do with given user data. Differently from our proposal, this approach does not take advantage of the already existing standards, despite the authors themselves recognize that “*For the DDS model to become widely utilised the DDS API will need to be adopted as a standard*” (p. 58).

The approach we propose is very similar to the vision of the *Data Portability Project*[9] launched in 2008, as written on the associated website:

> “Data portability is a new approach, where it is easier to use and deliver services. This frictionless movement through the network of services fosters stronger relationships between people and services providers and helps build a healthy networked ecosystem.”

Unfortunately, we could not find examples of applications developed having this goal in mind. The technology to enable data portability does exist but opening up data to build helpful and competing services still seems unsafe.

5 OPEN POINTS FOR RESEARCH

Indeed, the first short-term goal is to identify a number of simple use cases and try to develop prototype applications that make use of exported data subject to user authorization. But there are a number of aspects that need to be investigated in order for our proposed approach to become effective and cover a broader range of use cases.

One of the open points is related to distributed activities that may incur unpredictable delays before delivering an answer to the user; as an easy example, think at an online bureaucratic task requiring some kind of permission or signature by a human officer in order to proceed. A naive application would remain idle, waiting for an unpredictable time, and this would not be acceptable from the user side. In these cases the API should support some sort of state transition for the application, so that it could be put in a sort

[9]http://www.dataportability.org

of stand-by mode, with the possibility of sending an asynchronous notification to the user (by SMS, email, or other messaging) whenever the application is ready to proceed.

A more complex scenario where some research effort could be necessary, concerns tasks that require reservation of multiple resources in an all-or-nothing fashion, or, in other words, distributed transactions. It would be easy to draw a number of use cases where a distributed transaction is in order; think of, for instance, the usual example where a guy wants to make reservations for his holiday time, and must match the availability of a seat on a flight with the availability of a room in a hotel, the choice of location depending on the very possibility to find both resources in the required dates. Distributed transaction is an established topics of research in the field of distributed computing [10], but the community seems not to have reached a consensus on whether could distributed transaction be implemented in a REST style (Little, 2009; Musgrove, 2009; Marinos et al., 2009; Carlyle, 2009), yet REST is the style of all the web APIs discussed in Section 3.

[10]In SOA, the concept of transaction is encapsulated into a specific kind of web service called *WS Transaction* (WS-TX, 2004)

REFERENCES

ATOM (2005). Atom Syndication Format. http:// tools.ietf.org/html/rfc4287.

Berners-Lee, T. (2006). Linked Data. http:// www.w3.org/ DesignIssues/LinkedData.html.

Carlyle, B. (2009). The REST Statelessness Constraint, in Sound Advice - Blog. http://soundadvice.id.au/ blog/2009/06/13/.

E. Hammer-Lahav (2010). Introducing OAuth 2.0. http:// hueniverse.com/2010/05/introducing-oauth-2-0/.

E. Minardi (2010). Introduzione ad Open Data Protocol (aka odata). http://www.booom.it/wordpress/tag/ open-data-protocol/.

Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. http:// www.ics.uci.edu/ ∼fielding/pubs/dissertation/ top.htm.

GData (2007). Google Data Protocol. http:// code.google.com/apis/gdata/.

GDataLicence (n.d.). Google Data Protocol license. http:// code.google.com/apis/gdata/patent-license.html.

GDataOAuth (n.d.). Using OAuth with the Google Data APIs. http://code.google.com/apis/gdata/articles/ oauth.html.

Little, M. (2009). REST and transactions? http:// www.infoq.com/news/2009/06/rest-ts.

M. Cotterell (2008). GData Module for Drupal. http:// groups.drupal.org/node/10142.

Marinos, A., Razavi, A., Moschoyiannis, S., and Krause, P. (2009). RETRO: A (hopefully) RESTful Transaction Model. https://docs.google.com/View? id=ddffwdq5_2csz22wfd&pageview=1&hgd=1.

Musgrove, M. (2009). Transactional support for JAX RS based applications. https://community.jboss.org/wiki/ TransactionalsupportforJAXRSbasedapplications.

Nam, T. (2012). Suggesting frameworks of citizen-sourcing via Government 2.0. *Government Information Quarterly*, 29(1):12 – 20.

OAuth (2006). OAuth authentication protocol. http:// oauth.net.

OAuth2.0 (2010). OAuth authentication protocol version 2.0. http:// oauth.net/2/.

OAuthExample (n.d.). OAuth 1.0 for Web Applications. http://code.google.com/apis/accounts/docs/ OAuth.html.

Obama, B. (2009). Memorandum for the Heads of Executive Departments and Agencies: Transparency and Open Government. Retrieved January 29, 2012, from http://www.whitehouse.gov/the_press_office/TransparencyandOpenGovernment.

ODataLicence (n.d.). Microsoft Open Data Protocol license. http://www.microsoft.com/openspecifications/ en/us/programs/osp/default.aspx.

OpenID (n.d.). OpenID Foundation. http://openid.net/.

OpenIDGoogle (n.d.). Federated Login for Google Account Users. http://code.google.com/apis/accounts/docs/ OpenID.html.

OWL (2004). Web Ontology Language. http:// www.w3.org/2004/OWL/.

RDF (n.d.). Resource Description Framework. http:// www.w3.org/RDF/.

RSS (n.d.). Really Simple Syndication (RSS). http:// www.rssboard.org/.

SPARQL (2004). SPARQL Query Language for RDF. http://www.w3.org/TR/rdf-sparql-query/.

Surowiecki, J. (2005). *The Wisdom of Crowds*. Doubleday.

T. O'Reilly (2005). What Is Web 2.0. http://oreilly.com/ web2/archive/what-is-web-20.html.

Wallis, M., Henskens, F., and Hannaford, M. (2011). Web 2.0 Data: Decoupling Ownership from Provision. *International Journal on Advances in Internet Technology*, 4(1 & 2):47–59.

WS-TX (2004). Web Services Transactions specifications. http://www.ibm.com/developerworks/library/specification/wstx/.