Publishing data for maximized reuse

Pieter Colpaert

Ghent University - iMinds - Multimedia Lab
and Open Knowledge Central
pieter.colpaert@okfn.org

Abstract. Two movements are currently influencing the owners of public datasets to open up what's inside their organization: the Web API movement and the Open Data movement. The first advocates open Web-services which can provide a specific use case of information. The second advocates raw data to be published to the Web to be able to get used, reused and redistributed. What are the advantages and disadvantages of both approaches? Where exactly do they part in their ideology and how can we get the best from both worlds?
This was the main question discussed during the keynote of the Services and Applications over Linked APIs and Data (SALAD) 2014 workshop at ESWC. In this paper I summarize the rationale and the conclusion made during the talk.

Keywords: Open Data, Linked Open Data, Linked Data Fragments, smart cities, data publishing, Web APIs

1 Introduction

The definition of Open Data[1] states that the only requirement for a dataset to be called open is that it is openly licensed. The definition helps advocating datasets to come out of the gray zone and enter the white zone, as illustrated in Figure 1.

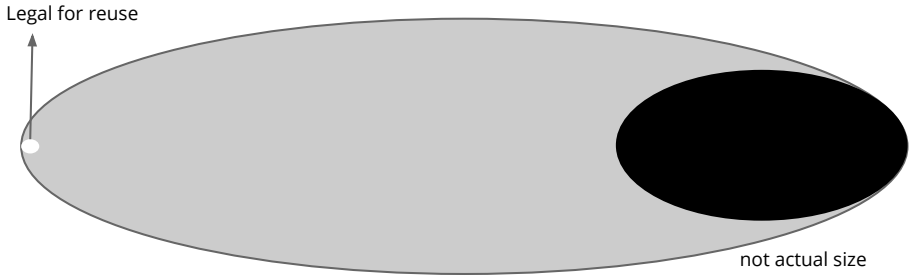

Fig. 1. Only a few datasets are openly licensed and help unlocking the full potential of the Web

The definition does not contain technical details on how to publish the data to the Web. When data owners want to publish the data, they seek answers within expert

[1] *http://opendefinition.org*

communities such as the audience of the Extended Semantic Web Conference (ESWC). Two groups of answers can be distinguished: 1) publishing the data to the Web as Open Data and 2) preparing data for certain use cases through a Web API. Which one is the right answer?

In the opening talk at the Services and Applications over Linked APIs and Data (SALAD) 2014 workshop held at ESWC 2014, we have discussed how the goals of data owners on the one hand and the needs of reusers on the other can be aligned. First, the reusers are identified. Then, the data owners are described. Finally, an answer is formulated towards whether we need to ask for Open Data, or we need to ask for APIs ready for a certain use case.

2 Three types of reusers

We introduce three types of reusers based on their goals, illustrated in Figure 2. The *first type* of reusers are the ones that require a direct interface to work on. The interface needs to deliver the right responses for their use case. For example, in transport, this API would be a route planner. Apps written upon this API are going to be route planning applications. A *second type* of reusers are the integrators. They integrate data from various sources in one system, shape the data to their needs, and code their applications on this integrated data service. A *third type* of reusers are the data experts (e.g., the owners themself or a data broker). They can decide how the global identifiers are formed, they can decide on the model and vocabulary of the data, etc.

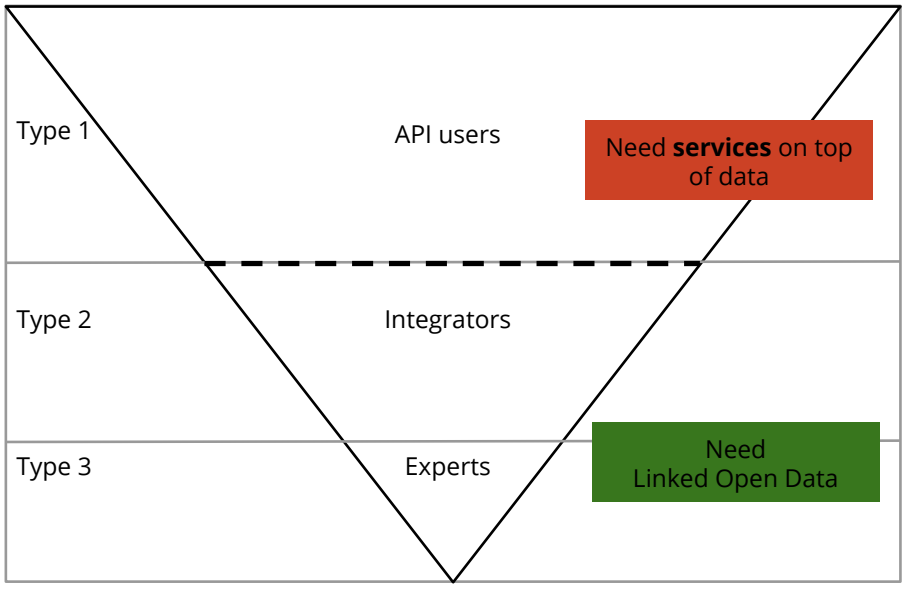

Fig. 2. Three types of Open Data reusers with different needs, ordered by the amount of people from that type that is able to reuse your dataset.

The needs of type 1 users vs. type 2 or 3 differ as API users don't need the data itself, but they need a service or a library on top of the data. This will allow them to reuse this data without thorough knowledge about how to apply the data: the knowledge is already present within the API or software library, typically created by type 2 or type 3 reusers.

3 Why data owners open up their data

Convincing data owners to publish their data to the Web is not an easy task. Why would they give everyone the right to use, reuse and redistribute the data? As part of the organization Open Knowledge, I have used various arguments in convincing data owners to do so. One with a directly visible impact is the "*free apps*" argument. App developers are going to create apps with their own business model, or merely voluntarily, on top of your data and you will not have to create an app for your data for every platform out there. The argument is not advisable as it may cause wrong expectations. Furthermore it might trigger data owners to create expensive open services ready for a certain use case rather than opening up the data itself.

A better argument is "*because they have to*". For example, the G8 Open Data charter, the Freedom of Information Act, the European directive on PSI, etc. put Open Data as the default, not as the exception.

Not everyone has to open their data by law. Another argument is to become and stay the *authoritative source* of the data. In that case opening up the data is advertising your URIs to the outside world. If everyone is using your URIs for the things you define, you have become the authoritative source of the data and are best in place to offer extra services on top of the data (such as service level agreements).

Last but not not least, is the argument of data never being correct. The more the data gets used, the more *feedback* will be provided. "One certain way to improve the data quality, is to improve its use", said Ken Orr in 1998 [3]. He was referring to the milenium bug, where he concluded that the real challenge did not lie in the datasets that get used every day, but the challenge was to fix the datasets that wouldn't get used often. He described data management systems as a Feedback Control System where maximizing the reuse will benefit the quality of the data, as there would be more feedback. With Open Data today, it is not much different: opening up the data opens the door towards more reuse and feedback, thus a higher quality dataset.

Within each of these arguments, there is a clear promise: opening up the data will raise the reuse of the data, which leads to more benefits. Once a data owner is convinced about this matter, the question follows: "how do we publish our data to the Web?".

4 Open Data vs. Web APIs

Data owners want to raise the reuse of their data. To this end, we introduce a distinction between data publishing and data services: *data publishing* focuses on keeping the data high available, while *data services* focus on serving the data for a specific use case. Different approaches are illustrated in Figure 3.

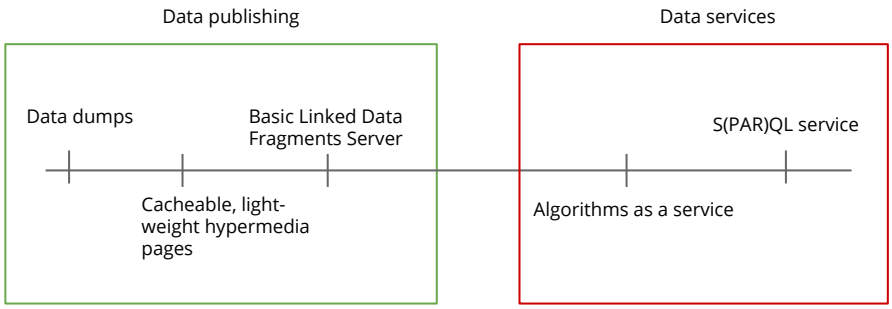

Fig. 3. The difference between data publishing and data services

Data dumps solve this prerequisite quite simply: the entire datadump is up for download, it's easy to host high availably (CDN hosts) and you can create copies of it without any problem. Yet, there are various pragmatic reasons not to just publish as a data dump: the data needs to be downloaded entirely before it can be queried, data that gets regular updates will need to get downloaded equally as much, etc.

Cacheable light-weight hypermedia pages are created by splitting the file into fragments and creating links between those fragments. The entire datadump can still be downloaded by following all the links and updates can be provided on separate resources, which can be downloaded separately. To query the data however, we would still need to download all the data locally.

Linked Data Fragments provide a way to extend these hypermedia interfaces with the ability to filter triple patterns and provide counts for these triple patterns [4]. This way, clients can query the Web for data using Basic Graph Patterns, while there are stil a finite amount of linked data fragments to be published with additional meta-data.

Algorithms or Software as a Service are a *service* on top of the data. They expose an API where the data is used to feed an algorithm. For instance, for public transport, this would be a route planning API. It becomes very expensive for the data owner when these API users try to download the entire data dump from this API, therefore this cannot be called data publishing anymore and this should be called data services.

Public querying endpoints such as SQL or SPARQL over HTTP are endpoints where you are able to query the data from the server-side. They allow user agents to request not only parts of the data, but also calculations of the different datasets. The availability of these querying endpoints are questionable [1]. One of the reasons given are the non-cacheability [4]. Also Hogan et al. made the case for having an alternate mechanism to query the data on the Web [2].

5 Conclusion

In this paper we have made the case for *data owners* to publish their data as Open Data instead of building Web APIs ready for certain use cases. This because datasets should be high available first, tools to query the dataset can be implemented by third parties, or

by the data owner itself as a complementary service. Furthermore, we want the data on the Web to come out of the gray zone (cfr. Figure 1) whether it is legal to use, reuse and redistribute the data. Therefore an open license has to be applied on the full dataset.

As indicated in Figure 2, we have distinguished 3 types of reusers, amongst which type 2 and type 3 need access to the data for use, reuse and redistribution. Type 1 users, which account for the most of the reusers, only need an API though. As data owners are not specialized at creating end-user applications, there is a not only a great opportunity for businesses to create apps on top of these datasets, but also for type 2 or type 3 users to create APIs for type 1 reusers.

References

1. C. Buil-Aranda, A. Hogan, J. Umbrich, and P.-Y. Vandenbussche. SPARQL web-querying infrastructure: Ready for action? In *The Semantic Web–ISWC 2013*, pages 277–293. Springer, 2013.
2. A. Hogan and C. Guyierrez. Path towards the Sustainable Consumption of Semantic Data on the Web.
3. K. Orr. Data quality and systems theory. *Communications of the ACM*, 41(2):66–71, 1998.
4. R. Verborgh, M. Vander Sande, P. Colpaert, E. Mannens, and R. Van de Walle. Web-Scale Querying through Linked Data Fragments. In *Proceedings of the 7th Workshop on Linked Data on the Web*, 2014.