# Monetization Strategies for the Web of Data

Tobias Grubenmann
«supervised by Abraham Bernstein»
University of Zurich
Department of Informatics
Zurich, Switzerland
grubenmann@ifi.uzh.ch

## ABSTRACT

Inspired by the World Wide Web, the Web of Data is a network of interlinked data fragments. One of the main advantages of the Web of Data is that all of its content is processable by machines. However, this also has its drawbacks when it comes to monetization of the content: advertisements and donations—two important financial motors in the World Wide Web—do not translate into the Web of Data as they rely on exposing the user to advertisement/call for donations.

The remedy this situation, we propose two different monetization strategies for the Web of Data. The first strategy involves a marketplace where users can buy data in an integrated way. The second strategy allows third parties to promote certain data. In return, the sponsors pay money whenever a user follows a link contained in the sponsored data. We identified two different kind of data—commercial and sponsored data—which can benefit from the two respective monetization strategies. With our work, we propose solutions to the problem of financing the creation and maintenance of content in the Web of Data.

## KEYWORDS

Web of Data, Monetization, Marketplace, Integer Programming, Auction

**ACM Reference Format:**
Tobias Grubenmann. 2018. Monetization Strategies for the Web of Data. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3184558.3186568

## 1 PROBLEM

The *Web of Data* (WoD) is an extension of the World Wide Web (WWW) to facilitate the exchange and processing of data which is distributed over the Web. In the WoD, data is exposed using a machine-processable, semantically annotated data model, the *Resource Description Framework (RDF) data model*. Using this data model, *machines* can access data *autonomously* on behalf of humans according to some task specifications. Another advantage of the RDF data model is that data sources can be queried in a federated fashion without agreeing on a common scheme beforehand. Even though the WoD is an extension of the WWW, accessing data in the WoD is more similar to querying one big, decentralized database than browsing web pages. Whilst this has the before-mentioned advantages, it also has its disadvantages. One of these disadvantages is that most *monetization strategies* for data publishers from the WWW do not translate easily to the WoD. Typical monetization strategies in the WWW include *advertisement*, *donations*, and *subsidies*.

Advertisement is arguably the biggest financial motor in the WWW. The WWW offers the opportunity to customize advertisement to the user and target specific user groups in a way unprecedented in any other media. A *publisher* of Web content can embed ads into the *presentation* of the actual content to create the required impressions (which may be converted into clicks and actions) for which the advertiser is paying. If we would want to translate this advertisement mechanism to the WoD, we would need a way to embed advertisement into the presentation of the RDF data. However, unlike the markup language HTML, the RDF data model does not provide the means to *dictate* how certain data elements should be presented to the user. Indeed, the mere concept of machine-processable data opposes the idea of exposing the consumer of the data (which could be a human or a machine) to advertisement. For example, the query language SPARQL [11], which can be used to query RDF data, allows to *project* a query answer to a set of variables which are of interest. Thus, meta-data like advertisement can easily be filtered out. Of course, it is always possible to introduce advertisement at the interface between the human and the machine which accesses the WoD. However, such advertisements would be controlled by the publisher of the respective interface and not by the publisher of the original data. The data publisher would not benefit from such an advertisement strategy, which is the main concern of our research.

Donations are another important source of money in the WWW. This monetization strategy relies on users' *appreciation* of the Web content to a degree where they are voluntarily donating money to keep the service alive. If a publisher wants to finance its service through donations, a crucial part is to make the consumers *aware* of the need of financial support and instruct them how to donate. This problem should not be underestimated. As [6] shows, a significant part of the users is not aware that there is a possibility to donate to Wikipedia[1], a website which is mainly financed through donations, or do not know how to donate. To remedy this situation, publishers which do rely on donations occasionally put banners on their websites to call for donations, for example. In the WoD, this problem is much more pronounced. Similar to what we have already discussed about advertisement, it is also not straightforward to embed calls

[1] http://www.wikipedia.org


*WWW '18 Companion, April 23–27, 2018, Lyon, France*

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3186568

for donations into RDF data. In the end, calls for donations can be seen as a kind of advertisement where the advertiser is also the publisher. In addition, users might simply not be aware that they are using the data of a certain publisher, especially if the data is part of a federation. Even if the RDF data contains some attribution about the source of the data, similar to advertisement, such meta-data can be filtered out. As a result, publishers in the WoD might not be able to create the required *awareness* to trigger donations.

Besides advertisement and donations, subsidies play also an important role in financing Web content. Unlike the former two, subsidies do actually also apply to the WoD setting. Most current datasets on the WoD which are not maintained by enthusiasts are subsidized either by governments via data access laws or by research grants. However, subsidies cannot be considered an actual monetization strategy, as they basically delegate the problem to somebody else.

**Problem statement:** Without new monetization strategies, many promising datasets will be poorly maintained or disappear as there will be not enough funds to keep the data up-to-date and the servers running. To solve this problem, we identify, model, and evaluate two new monetization strategies for the WoD.

The first monetization strategy focuses on the consumer of the WoD data to finance the data publishers. In the second monetization strategy a third party, the sponsor, finances the data. These two monetization strategies can be applied to two different types of data: The former strategy is applicable to data for which a consumer is willing to pay, which we will call *commercial data*. The latter strategy is applicable to data some sponsor is willing to promote, which we will call *sponsored data*.

## 2 STATE OF THE ART

The state of the art can be divided into two parts: Data Markets and Sponsored Search Auctions.

### 2.1 Data Markets

The idea to charge customers for accessing data is already implemented in the WWW. *Bloomberg*[2], *LexisNexis*[3], and *Thomson Reuters*[4] charge customers high fees for accessing their data, primarily using subscription-based models. All these examples have in common that they can sell their whole data offering charging quasi-monopolistic prices [2]. Also, for relational databases, markets have been proposed which sell data using arbitrage-free pricing schemes [4, 12]. In the work of [17], the negative externalities of giving away data are considered when selling them using an auction mechanism.

Unfortunately, none of these marketplaces allow a customer to join datasets from different providers, which is an important aspect of the WoD scenario we are investigating. In addition, to the best of our knowledge, none of these marketplaces considers partial answers due to value or budget constraints.

In a pilot study, [19] laid the foundation of our research by simulating a marketplace for the WoD. In [13], we first introduced the idea of using a double-auction to sell data on the WoD. Finally, in [10] we introduced our model for a marketplace which allows customers to buy data from decentralized sellers in an integrated way. In contrast to previous models, we focused on maximizing the customers utility given all available data.

### 2.2 Sponsored Search Auctions

The Generalized Second Price (GSP), which was introduced by Google in 2002 and replaced the first price models, became an industry standard for sponsored search auctions [18]. The main advantage of the GSP auction is that it prevents "cycling" patterns, a situation where repeatedly prices gradually rise and then suddenly drop [5]. Another alternative to first price models is the Vickrey-Clarke-Groves (VCG) auction [3, 7, 16]. This auction has the advantage of being *truthful*, meaning that no bidder has an incentive to lie about his or her valuation of the outcome of the auction. [1] proposed a new auction mechanism where bidders can impose additional constraints on the exact location where they want their ad to appear, as a bidder might have a value for the mere appearance of the ad, even if the ad is not clicked by the user.

So far, none of these auction models have been applied to the WoD setting. One reason is that, as discussed in Section 1, online advertisement from the WWW does not translate to the WoD. However, as we have shown in [9], the auction techniques can nevertheless be applied in the WoD setting, although not in the way they were originally designed for.

## 3 PROPOSED APPROACH

Our basic approach is to formulate two different monetization strategies for commercial and sponsored data.

The main characteristics of commercial data is that the consumer has a higher *value* for *consuming* the data than any other entity (including, but not limited to, the data publisher) has in *exposing* the customer to the data. The value indicates the willingness to pay for consuming and exposing the data, respectively. Examples of commercial data are data about consumer behaviors or stock exchange data.

The main characteristics of sponsored data is that there is an entity, which we call the *sponsor*, that has a higher value for *exposing* the consumer to the data than the consumer has for *consuming* the data. Sponsored data is basically any data which is typically included in advertisements. In contrast to traditional advertisement, however, sponsored data is explicitly requested by the consumer and does not represent additional information or even a distraction from the requested data. Examples of sponsored data are information about hotels and restaurants or data about goods for sale.

### 3.1 A Marketplace for Commercial Data

To answer the first research question, we propose a marketplace for commercial data in the WoD. The main feature of our marketplace is that a customer can *combine* data from different data publishers in an integrated way, which means that the customer can access the data of all participating data publishers as if there would be only one big database. The customer does not have to pay for all the available data, however. Instead, a customer can submit a query and will be charged only for those triples (the smallest possible data fragments) which are precisely needed to form a certain query

[2] http://www.bloomberg.com
[3] http://www.lexisnexis.com
[4] http://www.thomsonreuters.com

answer. In addition, the customer can decide how small or big the query answer should be.

Without using such a marketplace, a user would have to buy individual triples directly from the different data publishers. While this is possible, there is a major drawback of this approach when data from more than one data publisher have to be joined: It is very hard for the customer to estimate which triples of one publisher will actually join with the triples of another publisher. Hence, a customer might either waste money on triples which do not join or miss a part of the query answer because some triples were not bought. As we have shown in [8], join estimation techniques are suffering a lot from false-positive matches in the WoD setting and hence, we cannot expect that a customer would be able to use such techniques to buy exactly those triples which are needed to form a specific query answer. Only a query execution can reveal the true contribution and value of the publishers' triples. Hence, we argue that a market for the WoD has to execute a given query on the publishers' data *before* the decision can be made which triples should be bought by the customer. Figure 1 shows the necessary steps in our marketplace:

(1) The marketplace receives a query from the customer and executes it on the available data.
(2) Only a certain part of the complete query answer is chosen by the customer.
(3) The customer only pays the marketplace the indicated price of the selected triples.

My first research question focuses on the customer's buying decision, which we call the *allocation problem*:

**RQ1** How to solve the allocation problem for a market for the WoD efficiently?

## 3.2 A Delayed-Answer Auction for Sponsored Data

For the second research question, we propose a slot-auction similar to the auctions used for sponsored search results. In this auction, sponsors pay money if a user follows a certain link contained in a query answer. To motivate sponsors to pay for such link visits, we need a way to prioritize those links which receive a higher bid from those links with lower or no bid. For this, we introduce delays into the delivery of the query answer. Records containing links with high bids are delivered more quickly than the records with lower bids. By introducing such different delays, we create a ranking of the different records of a query answer. The ranking we introduce works similar to the rankings of advertisement slots in sponsored search results. Hence, we can apply the auction techniques which originate from sponsored search auctions to this new setting.

Figure 2 illustrates the process of our new auction concept. Like in the marketplace from Section 3.1, we have a user who submits a query to our system and the query gets executed on the data of all participating publishers. This time, however, the user does not select a part of the query answer. Instead, the complete query answer is delivered to the user. Different delays are assigned to the different records of the query answer, depending on how much the sponsors bid on certain links contained in the records. Records with less delay have a higher chance of being considered by the user

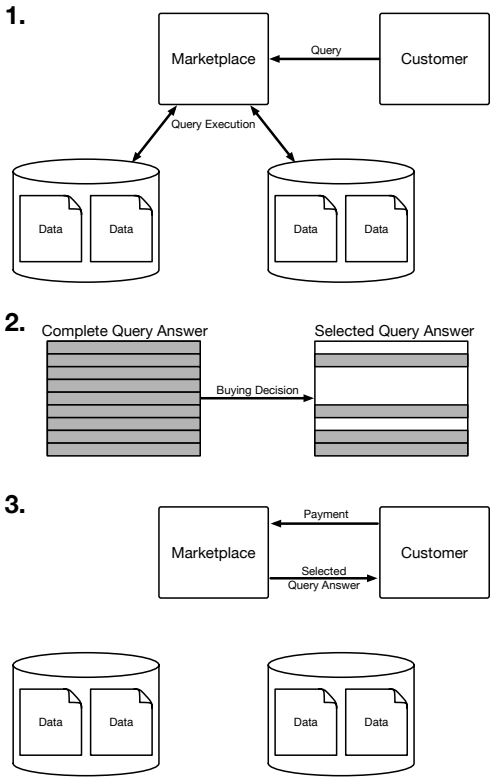


**Figure 1: The three steps from a query to the selected query answer.**

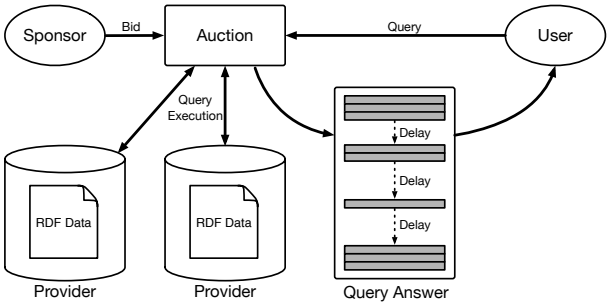


**Figure 2: A user gets a delayed query answer based on the bids of sponsors.**

and hence, the likelihood of a link visit is higher. If the user visits a certain link, the sponsor has to pay a certain price to the auctioneer. The prices are calculated using the VCG mechanism [3, 7, 16].

The second research question asks about the influence of the delays on the proposed auction:

**RQ2** How does the choice of the delays influence the proposed delayed-answer auction?

# 4 METHODOLOGY

The methodology of our research is slightly different for the two research questions.

## 4.1 Marketplace for Commercial Data

For the marketplace for commercial data, different parts of the market have to be modelled. The first part models how the necessary meta-data can be extracted from the different data publishers during query execution (part 1 in Figure 1). In our model, the publishers' data is organized into *data products*. A data product is a collection of RDF triples that all have the same meta-data (including the price). We use RDF statements to describe the meta-data about a data product. This allows us to access the data and the meta-data of the data products using a single, federated SPARQL query. Additionally, a user can use the meta-data directly in the query to restrict the query answer based on the specific needs.

A second important aspect of our model for the marketplace is the selection of the query answer (part 2 in Figure 1). Choosing an appropriate subset of the complete query answer can be modelled as an integer programming problem, assuming that the value for the query answer is a linear function with respect to the containing records. However, solving such a problem is NP-hard. If the value is not linear but monotonically decreasing, we cannot formulate the problem as an integer programming problem.

The query execution of our marketplace is implemented using the federated querying engine FedX [15]. In traditional federated SPARQL query execution, a SPARQL query is split up into subqueries which are sent to different sources. Combining the query answers from the different subqueries results in the complete query answer. Instead of sending the subqueries directly to the endpoints, we have to rewrite each subquery to retrieve also all meta-data about the different data products.

The selection of the query answer is implemented using two different approaches: (1) Using the commercial solver CPLEX[5]. This approach is only possible if the selection can be modelled as an integer programming problem. (2) Using our own greedy algorithm. This approach is also possible if the value is monotonically decreasing.

To evaluate the two different approaches, we are using two different metrics: the runtime of the algorithms (which establishes their feasibility) and the quality of their selection. The quality of a specific selection of a query answer is expressed by its *utility*, which is the *value* a user has for a specific selection minus the *price* the customer has to pay.

## 4.2 Delayed-Answer Auction

The model for the delayed-answer auction is based on the slot auctions for sponsored search results. In this model, we have different slots with decreasing likelihood of being selected by the user. Since a query answer is a set of records, there is no inherent ordering of the different records and hence, all records have initially the same likelihood of being selected. To create such an ordering artificially, we have to introduce different delays for different records. Each record inside a query answer gets assigned to a specific slot which determines the delay.

[5]https://www.ibm.com/jm-en/marketplace/ibm-ilog-cplex

Even though our delayed-answer auction is similar to sponsored search auctions, we need a new click model for this new setting. The model we are using assumes that a user is able to judge the relevancy of a link upon receiving the query answer and hence, will visit at most one link. Unlike links in the WWW, in the WoD, the user can judge the relevancy of a link by the information embedded in the query answer.

The delays for the different records are important parameters of our auction which have to be chosen by the auctioneer. These parameters add a new dimensionality to the slot auctions and do not have any counterpart in traditional slot auctions used in sponsored search auctions. Hence, it is crucial to understand how these parameters influence the auction and how an auctioneer can optimize on them. Hence, we perform a theoretical analysis of how the parameters can be optimized. In addition, we perform simulations which illustrates the choice of the parameters on social welfare (the total generated wealth) and revenue for the auctioneer.

# 5 RESULTS

We will now briefly sketch some of the results we already obtained for the two scenarios: the marketplace for commercial data (Research Question RQ1) and the delayed-answer auction (Research Question RQ2).

## 5.1 Marketplace for Commercial Data

To evaluate the two different approaches sketched in Section 4.1, we tested both algorithms on 17 FedBench [14] queries. We used the metrics described in Section 4.1 for this evaluation.

With respect to the utility, the first approach using CPLEX and the second approach using our greedy algorithm are very close to each other. We observed that our greedy algorithm reaches more than 90% of the utility of CPLEX in 15 out of 17 queries. For the other two queries, our greedy algorithm reached at least 80%. At the same time, the greedy algorithm runs in most cases between 1 and 3 orders of magnitude faster than CPLEX. Figure 3 compares the approach using CPLEX, which we call the Integer Rule, and our own algorithm, which we call the Greedy Rule.

In a different evaluation, we showed that CPLEX is very sensitive to the diversity of the query answer and that for some parameter ranges, CPLEX is not able to find an optimal solution within a time-limit of 12 hours.

As our results show, the two approaches under investigation—Integer Rule and Greedy Rule—can be used to solve the allocation problem. While using CPLEX guarantees an optimal solution, our own algorithm can achieve similar results in a much shorter timeframe, usually. This gives an answer to Research Question RQ1: The allocation problem can in most cases solved efficiently with both approaches. In some cases, the Integer Rule is superior to the Greedy Rule. In most cases, however, the Greedy Rule can be a good and fast alternative. Finally, for some cases, the Integer Rule is not feasible due to the excessive runtime.

## 5.2 Delayed-Answer Auction

We analyzed the properties of the delayed-answer auction and found that in general it is not possible to optimize both revenue and

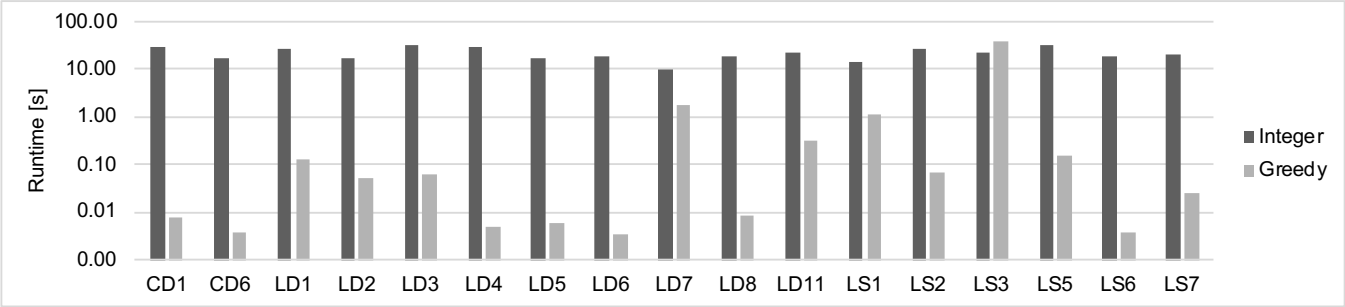


**Figure 3: Runtime in seconds when using CPLEX (Integer) and our own algorithm (Greedy) for the FedBench benchmark.**

social welfare with the same set of parameters for the delays. Optimal revenue and optimal social welfare have in common that they both require that part of the query answer is delayed indefinitely. Such an approach might damp the user's experience, however, who only receives part of all available data. Our simulations showed that the parameters for optimizing revenue and optimizing social welfare differ a lot when there are is a small set of bids which are much higher than the remaining bids.

The analysis gives a good overview of the influence of the choice of the parameters for the auction and thus, answers Research Question RQ2.

## 6 CONCLUSIONS AND FUTURE WORK

To create financial incentives to publish content in the Web of Data, we proposed two different strategies, one for commercial and one for sponsored data. As discussed in Section 5, we already established some results regarding these two strategies.

For the first strategy, we compared two different ways how a consumer can select a subset of a query answer and compared their performance with respect to runtime and utility. What is left for future work is to investigate how data providers can determine the best price for their data. We believe that such optimal price can be learned, for example by using reinforced learning techniques. We also need to explore how well subscription-based models can be applied to our new setting. Finally, our marketplace offers new possibilities for market-aware query optimizations which considers the costs we impose on the data providers by executing queries on their servers.

For the second strategy, we already did a theoretical analysis of our auction setting and a simulation which showed some interesting properties of our model. However, we are still missing an evaluation of our method using real-world data. For this, we plan to use datasets from traditional ad auctions for web search. We have to investigate how these datasets can be applied to our new setting.

## ACKNOWLEDGMENTS

I want to thank my advisor Abraham Bernstein for supervising my PhD research.

This work was partially supported by the Swiss National Science Foundation under grant #153598 (http://p3.snf.ch/project-153598).

## REFERENCES

[1] Gagan Aggarwal, Jon Feldman, and Shanmugavelayutham Muthukrishnan. 2007. Bidding to the Top: VCG and Equilibria of Position-Based Auctions. In *Approximation and Online Algorithms. WAOA 2006.*

[2] Yannis Bakos and Erik Brynjolfsson. 1999. Bundling Information Goods: Pricing, Profits, and Efficiency. *Management Science* 45, 12 (1999), 1613–1630. http://EconPapers.repec.org/RePEc:inm:ormnsc:v:45:y:1999:i:12:p:1613-1630

[3] Edward H. Clarke. 1971. Multipart Pricing of Public Goods. *Public Choice* 2 (1971), 19–33.

[4] Shaleen Deep and Paraschos Koutris. 2017. QIRANA: A Framework for Scalable Query Pricing. In *SIGMOD '17 Proceedings of the 2017 ACM International Conference on Management of Data*. 699–713.

[5] Benjamin Edelman and Michael Ostrovsky. 2007. Strategic Bidder Behavior in Sponsored Search Auctions. In *Decision Support Systems*, Vol. 43. 192–198.

[6] Ruediger Glott, Philipp Schmidt, and Rishab Ghosh. 2010. *Wikipedia survey – overview of results.* Technical Report. United Nations University MERIT.

[7] Theodore Groves. 1973. Incentives in Teams. *Econometrica* 41(4) (1973), 617–631.

[8] Tobias Grubenmann, Abraham Bernstein, Dmitry Moor, and Sven Seuken. 2017. Challenges of source selection in the WoD. In *Proceedings of the International Semantic Web Conference ISWC '17.*

[9] Tobias Grubenmann, Abraham Bernstein, Dmitry Moor, and Sven Seuken. 2018. Financing the Web of Data with Delayed-Answer Auctions. In *WWW 2018: The 2018 Web Conference.*

[10] Tobias Grubenmann, Daniele Dell'Aglio, Abraham Bernstein, Dmitry Moor, and Sven Seuken. 2017. Decentralizing the Semantic Web: Who will pay to realize it?. In *Proceedings of the Workshop on Decentralizing the Semantic Web (DeSemWeb).* http://ceur-ws.org/Vol-1934/contribution-01.pdf

[11] Steve Harris and Andy Seaborne. 2013. SPARQL 1.1 Query Language. https://www.w3.org/TR/sparql11-query/. (March 2013).

[12] Paraschos Koutris, Prasang Upadhyaya, Magdalena Balazinska, Bill Howe, and Dan Suciu. 2013. Toward Practical Query Pricing with QueryMarket. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data.* 613–624.

[13] Dmitry Moor, Tobias Grubenmann, Sven Seuken, and Abraham Bernstein. 2015. A Double Auction for Querying the Web of Data. In *The Third Conference on Auctions, Market Mechanisms and Their Applications.*

[14] Michael Schmidt, Olaf Görlitz, Peter Haase, Günter Ladwig, Andreas Schwarte, and Thanh Tran. 2011. FedBench: A Benchmark Suite for Federated Semantic Data Query Processing. *International Semantic Web Conference* (2011), 585–600.

[15] Andreas Schwarte, Peter Haase, Katja Hose, Ralf Schenkel, and Michael Schmidt. 2011. FedX: Optimization Techniques for Federated Query Processing on Linked Data. In *International Semantic Web Conference (1).* 601–616.

[16] William Vickrey. 1961. Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance* 16(1) (1961), 8–37.

[17] Xiang Wang, Zhenzhe Zheng, Fan Wu, Xiaoju Dong, Shaojie Tang, and Guihai Chen. 2016. Strategy-proof data auctions with negative externalities. In *Proceedings of the International Conference on Autonomous Agents Multiagent Systems (AAMAS).* 1269–1270.

[18] Christopher A. Wilkens, Ruggiero Cavallo, and Rad Niazadeh. 2017. GSP – The Cinderella of Mechanism Design. In *WWW '17 Proceedings of the 26th International Conference on World Wide Web.* 25–32.

[19] Mengia Zollinger, Cosmin Basca, and Abraham Bernstein. 2013. *Market-based SPARQL Brokerage with MaTriX: Towards a Mechanism for Economic Welfare Growth and Incentives for Free Data Provision in the Web of Data.* Technical Report IFI-2013.4.