Privacy-Preserving Data Publishing in Linked Data Mashup Architectures

Juan Manuel Dodero[1], Mercedes Rodriguez-Garcia[1] and Enrico Motta[2]

[1] Escuela Superior de Ingeniería, Universidad de Cádiz, Spain
[2] Knowledge Media Institute, The Open University, Milton Keynes, UK

Abstract. The mashup of microdata sources to form a data hub must fulfill a set of privacy preservation anonymity requirements that hinder data analysts to figure out sensitive information of the source datasets. This is relevant in a number of fields that include smart cities, electronic healthcare records and others. Linked data publishing architectures are not designed to adapt well to the requirements of existing approaches to sanitize the linked datasets, which do not always exploit the potential of semantics. Besides, the sanitizing protocols are not always controlled by a central coordinator. We propose a classification framework to decide on the distribution of control and partitioning of the dataset information models. Based on the framework, we define an approach to engineer privacy-preserving linked data mashups that defines the essential functionalities of privacy-preserving linked data publishing architectures. The classification framework and engineering method for data privacy preservation can have an implication for big data systems and emergent blockchain-based distributed ledgers.

Keywords: privacy preservation, data mashups, linked data architectures.

1 Introduction

Data mashups or hubs are combinations of information from multiple, independent origins into a single data source that can be queried through a single endpoint, thus serving data integration on demand. Data mashups constitute the basis of *Data-as-a-Service* (DaaS) architectures [24], aimed at reducing the cost of data management to support data scientists in mining combined data from disparate datasets so as to explore new knowledge.

However, sensitive information can be revealed when setting up a data mashup, so different privacy-preserving data publishing (PPDP) techniques such as data aggregation, noise addition and generalizations have been applied to the data that reside in each dataset [14]. Privacy preservation in data mashups is a relevant issue in diverse domains, including electronic business users' databases [27], electronic healthcare records [22,13,15] and smart city data hubs [34], among others.

1.1 An example on smart city data mashups

PPDP techniques focus on publishing personally identifiable information (i.e.

microdata[1]) about individuals. Because of privacy protection requirements, however, datasets are usually made public as aggregate data instead of microdata. For instance, the MK:Smart project (www.mksmart.org) provides citizens and companies with access to a number of aggregate data sources about diverse aspects of Milton Keynes town [8]. Provided by diverse institutions, such data include, among others, transport, average energy and water consumption of citizens and companies, which are compiled and stored in the MK Data Hub mashup (datahub.mksmart.org). The Milton Keynes City Council also provides the data hub with statistics about population growth, jobs, crime, marital status, religion and employment of their citizens as aggregate data that can be queried by place, ward, district, postcode and other forms of administrative aggregations. For that aim the MK Data Hub provides a public entity-centric API (Application Programming Interface). The MK Data Hub API and available datasets are very convenient and useful for citizens' open data efforts, but its analytical utility is limited to what can be observed on aggregate data, since microdata are not usually available. Doing so would require applying PPDP techniques on data providers (i.e. the city council, energy companies, etc.) and the resulting data mashup.

Despite the set of policies regulating the usage of each data source, and in spite of anonymizing microdata in each dataset, one cannot impede someone from knowing sensitive information by means of a *linking attack* to two or more datasets. For instance, even removing explicit identifiers, an individual's name in the City Council dataset DS_1(`address`, `birthdate`, `sex`, `postcode`, `name`, `taxes`) can be linked with another record in the energy consumption dataset DS_2(`birthdate`, `sex`, `postcode`, `electricityConsumption`, `gasConsumption`) through the combination of `postcode`, `birthdate` and `sex`. Each of these attributes does not uniquely identify a record owner, but their combination is a *quasi-identifier* that points to a unique or small number of records [19]. The linking attacker can thus notice that one house at a certain `address` might be unoccupied because its `electricityConsumption` and `gasConsumption` are almost nil. This can pose a threat about burglary, but it can be also a tool for tax agencies to investigate occupied rental houses that might have unpaid taxes from the lessor.

Even if anonymizing both datasets by means of generalization techniques on the quasi-identifiers of each dataset, there is the possibility that potential quasi-identifiers are split in both datasets that needs to be merged for analysis. For instance, let the City Council dataset schema be DS_1(`id`, `sex`, `defaulter`) and the energy consumption dataset schema be DS_2(`id`, `occupation`, `defaulter`, `electricityConsumption`, `gasConsumption`), as shown in Table 1. Assuming that a data analyst needs to combine DS_1 and DS_2 to predict default risks, DS_1 and DS_2 can be merged by matching the `id` field in a new integrated and then anonymized dataset *DS*. Then the `sex` and `occupation` attributes form a new quasi-identifier, which was not included in each dataset separately, so linking attack is still possible on these fields of the integrated dataset *DS*. After integrating the tables of both datasets, the (Female, Carpenter) individual on (`sex`, `occupation`) becomes unique and vulnerable to link sensitive information, such as address and energy consumptions.

[1] In Statistics, *microdata* is individuals' information consisting of properties that are recorded separately for every person who responds a survey; not to be confused with HTML *microdata*, which is commonly used in Web Engineering.

Table 1. Data tables from the City Council dataset (DS_1) and the Energy provider dataset (DS_2) that build up the data mashup *DS*

Shared		**DS_1**		**DS_2**		
ID	*default*	*sex*	*address*	*occupation*	*electricity Consumption*	*gas Consumption*
1-3	0y3n	M	A1	Sales	18	17
4-7	0y4n	M	A2	Ceramist	24	8
8-12	2y3n	M	A3	Plumber	25	10
13-16	3y1n	F	A4	Webmaster	20	17
17-22	4y2n	F	A5	Animator	31	11
23-25	3y0n	F	A6	Animator	34	10
26-28	3y0n	M	A7	Carver	32	12
29-31	3y0n	F	A8	Carver	30	14
32-33	2y0n	M	A9	Carpenter	33	11
34	1y0n	F	A10	Carpenter	29	15

Because the ultimate motivation underlying to data releases is to conduct analyses on the such data, anonymization should be done in a way that the protected data still retain as much analytical utility as possible; that is, the conclusions or inferences extracted from the analysis of the anonymized dataset should be similar to those of the original dataset. With the goal of balancing privacy and utility preservation, the PPDP methods [13,40] build the protected dataset by modifying the original quasi-identifying attributes while preserving certain statistical features. On the one hand, non-perturbative masking methods modify quasi-identifying attributes either by suppressing some of the data or by reducing their level of detail, such as generalization [32]. On the other hand, perturbative masking methods are based on distorting the quasi-identifying attributes by adding noise [7,25], data permuting [28] or data aggregating [10,11].

Most existing masking techniques poorly consider the semantics of nominal values and many times they manage individual attributes independently, thus neglecting the potential correlation between attribute pairs. For instance, numerical values such as `electricityConsumption` and `gasConsumption` on the Table 1 can be generalized by defining the intervals –e.g. [0,10), [10,20), [20,30) and [30,∞)– that mask the values of each microdata record, in order to sanitize the DS_2 dataset. On the contrary, the nominal values of the `occupation` column cannot be easily distorted by means of generalization techniques to sanitize the dataset. In previous works [26,31], distortion methods were improved to exploit the semantics provided by an ontology to better preserve the semantics underlying the nominal values. Therefore, nominal data have to be properly mapped to the instance values of an ontology of concepts that replace the original values of a nominal attribute in a dataset.

1.2 Mashup sanitizing approaches

There are two PPDP approaches when dealing with the manifold publishers that set

up a data mashup. The first one is *integrate-then-sanitize*, i.e., first integrates the distributed datasets by means of a common identifier, such as *SSN*, and then sanitize the quasi-identifying attributes from the integrated dataset using a PPDP masking method. As a result, the sanitized integrated dataset is expected to satisfy a given privacy model, such as *k*-anonymity [32]. In this approach, *k*-anonymity would not be completely satisfied for privacy-preserving distributed data mashups, because non-sanitized microdata should have to go through to the mashed-up database custodian. As a consequence, by knowing the original microdata, the data mashup holder may attempt to infer additional information (e.g., sensitive information) about their owners. The second approach, *sanitize-then-integrate*, provides better privacy guarantees because, before the data integration, each data publisher sanitizes its dataset locally. If a quasi-identifier formed by attributes spanning different data publishers is involved, this approach does not work because (i) sanitized datasets do not have identifying attributes to carry out the integration process and (ii) if it were possible to integrate the data, the resulting sanitized dataset would hardly fulfill the *k*-anonymity privacy requirement because the PPDP masking method needs as input the combination of the quasi-identifier of all involved datasets. To solve this issue, [38] proposes a similar approach to the integrate-then-sanitize strategy, which does not reveal the local data until it has been sanitized by generalization to satisfy *k*-anonymity. [27] extends this idea to distributed data mashup applications by establishing a collaboration among the data publishers. [34] also proposes a collaborative strategy to achieve *k*-anonymity on horizontally partitioned datasets. In these collaborative sanitization proposals, a communication among data publishers and/or between each data publisher and a central party or mashup *coordinator* is required.

In summary, should sanitization affect two or more datasets of a data mashup, the process must be collaboratively carried out by each dataset custodian. This paper proposes a novel approach to engineer the architecture of linked data publishing systems that takes into account two major requirements of the sanitizing solutions for privacy-preserving data mashups, namely where the *control* of the sanitizing protocol resides and how the data mashup schema is *partitioned*.

2 Semantic privacy-preserving data mashups

As many datasets can be involved in a data mashup, two aspects are relevant for privacy-preserving data publishing, so we are dealing with them independently in this section. First, the semantics and information model of the datasets is fundamental to solve data integration issues, which are common to other approaches in the databases field, such as the Extract-Transform-Loading (ETL) systems. Second, the dataset partitioning determines the requirements of the sanitizing protocol to be applied.

2.1 Semantics and conceptual mapping

Sanitized datasets are expected to satisfy a given privacy model, such as *k*-anonymity [32]. A sanitized integrated dataset satisfies *k*-anonymity if every combination of values on the quasi-identifiers is shared by at least *k* records.

Besides, usual perturbative PPDP approaches do not deal well with nominal data because of their mathematical operating principle [19]. For example, the noise addition mechanisms require computing the variance of the input data to generate noise sequences that reflect the degree of dispersion of the original values; the rank swapping mechanisms require sorting the input data to restrict the swap to a given rank-distance; and aggregation techniques typically use the mean to aggregate input data. As nominal data take values from a discrete and finite list of categories, which are usually expressed by words, a priori, it is not possible to carry out these operations. On the other hand, since nominal data utility is closely related to the preservation of *semantics* [37], any data transformation or calculation performed to anonymize data should carefully consider the *meaning* of the input values.

To enable a semantically-coherent protection of nominal data, recent PPDP proposals [3,26,31] exploit the formal knowledge modeled in ontologies. For that, prior to the masking process, the input nominal values are unequivocally associated with concepts in an ontology by means of a process named *interlinking* or conceptual mapping [2] (see Fig. 1). Following conceptual mapping, semantic PPDP methods will then be able to capture the semantics conveyed by nominal data. Specifically, these methods use the notion of semantic distance [4] to semantically compare the nominal values and so to detect how similar they are, and adaptations based on the semantic distance of the arithmetical operators involved in the masking process.

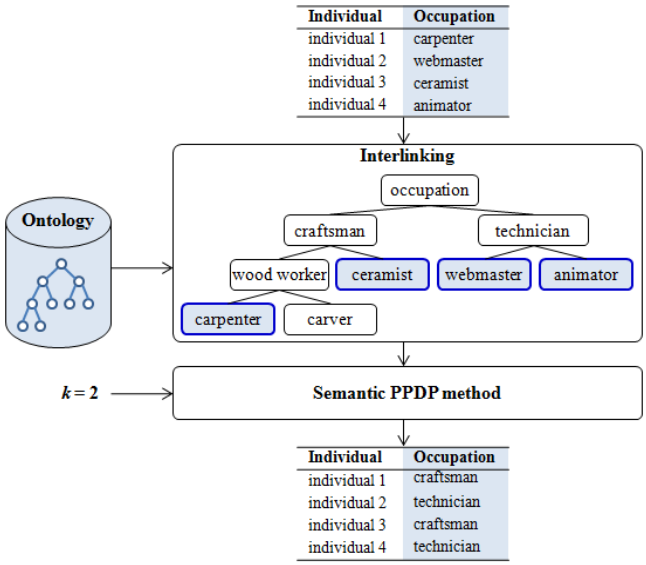

Fig. 1 Interlinking in semantic PPDP methods

2.2 Dataset partitioning

The mashup coordinator must discover how the data mashup is partitioned, i.e. horizontally or vertically (see Fig. 2), as explained in [6]. Such partitioning will condition the data integration and sanitization procedure. When the data mashup is *horizontally* partitioned, integration and sanitization processes must be delegated to

the mashup coordinator, as in the centralized integrate-then-sanitize approach. Unlike the centralized approach, however, the data publishers of a collaborative sanitization procedure will contribute their data in a privacy-preserving fashion by following an integration and sanitization protocol that can be managed by the mashup coordinator [34]. On the other hand, when the data mashup is *vertically* partitioned, the integration and sanitization processes must be delegated to the data publishers. Unlike the local sanitize-then-integrate approach, where each publisher independently sanitizes its data prior to sending them to the mashup coordinator, in the collaborative approach the sanitization has to be cooperatively performed by all data publishers involved in the mashup. In this context, the coordinator initiates the integration and sanitization protocol and remains in the background, looking forward to receiving the sanitized integrated dataset when the protocol is completed [27].

(a) Vertical partitioning

	Party 1		**...**		**Party N**	
ID + other shared attributes	non-sensitive attributes	sensitive attributes			non-sensitive attributes	sensitive attributes

(b) Horizontal partitioning

	Shared data schema		
	ID	non-sensitive attributes	sensitive attributes
Party 1			
...			
Party N			

Fig. 2 Dataset partitioning

3 Where sanitizing a data mashup?

The architectural patterns of linked data applications are discussed by [18] as a means to structure the software components that are comprised in the system (see Fig. 3). In this architecture, where do data sanitizing techniques have to be implemented to obtain a privacy-preserving data mashup?

On the top layer, the architecture of an LD application is usually made up of a number of *data access, integration and storage* modules (i.e. web access module, vocabulary mapping, identity resolution and quality evaluation). An extension has been implemented [16] based on an LD *API layer* on top of the data access and integration layer, which mediates between consumer applications and an integrated database. Eventually, pipelining all the functional modules of the data access and

integration layer leads to an integrated database which feeds the SPARQL endpoint or the API mediator module with RDF data.

In the bottom, the *publication* layer usually implements wrapper modules that, either by scraping [29] or enriching [21] web resources, add the required semantics to existing resources and datasets. Setting up a middleware module is also a strategy to reengineer existing applications to build such LD wrappers from a wide variety of data sources. When such distributed data sources have to be sanitized for privacy preservation, however, the architectural layer where sanitization must be implemented is not clear.

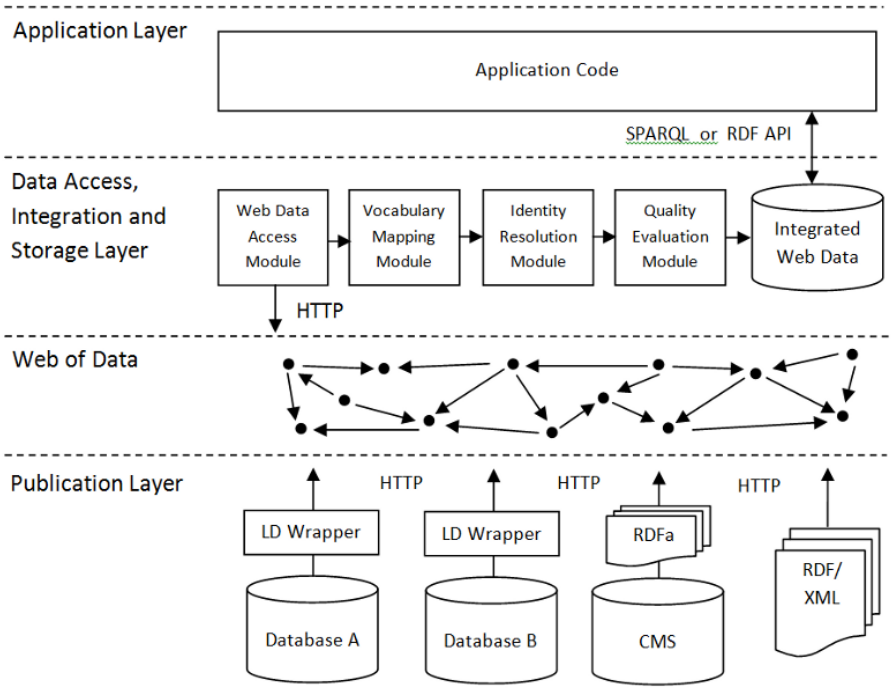

Fig. 3 Linked data applications architecture as described by [Heath et al., 2011]

The issue of what is the architectural layer that better fits data sanitization is not an exclusive concern of LD architectures. In distributed big data architectures based on the ETL paradigm, a data mashup application may also need several datasets from various data custodians and has to confront the challenge of privacy preservation at the same time (see Fig. 4) [20]. The location of data sanitization modules that implement the distributed algorithm for privacy preserving each pairwise dataset combination is not clear in the architectural design of ETL systems. In the architecture of Fig. 4, should it be part of the pre-validation, the ETL validation, or both?

To tackle an answer to the question about where does sanitizing should be carried out in a LD mashup or ETL architecture, an analysis of the main sanitizing approaches must be done. On the one hand, in the integrate-then-sanitize approach, it

seems reasonable to implement both the integration process and the PPDP masking techniques in the quality evaluation module of the data access and integration layer. However, should the data privacy requirements be implemented in this layer, *k*-anonymity would not be completely satisfied for privacy-preserving distributed data mashups, because non-sanitized microdata should have to go through all or some of the upper layer modules (i.e., data access, integration and storage layer) before being stored in the integrated, mashed-up database. On the other hand, as for the sanitize-then-integrate approach, it seems reasonable that PPDP masking techniques be implemented at the publication layer of a linked data application architecture and the integration process in the quality evaluation module of the data access and integration layer. However, if a quasi-identifier formed by attributes spanning different data publishers is involved, this approach does not work because.

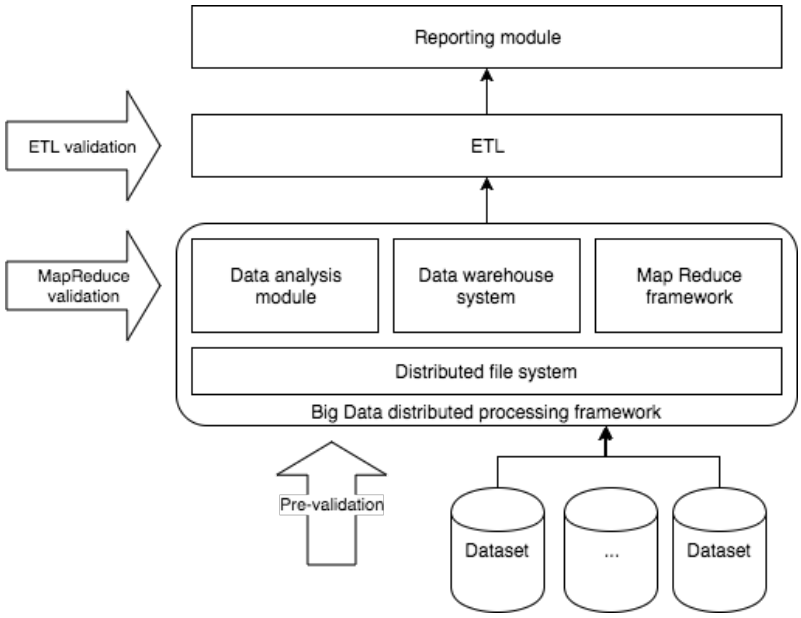

Fig. 4 Extract-Transform-Load paradigm in big data architectures [Jain et al., 2016]

As an answer to the issue of where data mashups should be sanitized, we need to either (1) implement anonymization and data integration techniques in the same architectural layer, or (2) to define a new architecture that reasonably does not disclose all the microdata that build up the privacy-preserving data mashup. The source of this architectural trade-off about PPDP issues is that existing LD architectures do not have into account the collaborative nature of the protocol for distributed dataset sanitizing. The data publishing functionalities are constrained to the data publication layer, but may affect other architectural layers, as discussed above.

4 Engineering privacy-preserving linked data mashups

The engineering PPDP approach proposed in this paper consists of a number of functions to be implemented in the modules of an LD application architecture (see Fig. 3). The following steps have to be taken before integrating LD datasets that come from existing data sources in a privacy-preserving fashion:

1. *Revealing the underlying data model*: A linked data model that is equivalent to the application schema is generated and published. This step can be readily carried out through existing wrapping tools, depending on the technology of the underlying data store –e.g. D2R server [5] or Virtuoso RDF Views [12] can be used with relational databases. To facilitate external linking with standard vocabularies, a set of mapping options can be configured. Thus, the web access module and vocabulary mapping module functionalities of the architecture are implemented in this stage.

2. *Linking the data instances*: The linked datasets retrieved from the internal data storage of the application can be explored and linked. This step is a function of the identity resolution module of the LD architecture. It can be made with the help of an external interlinking module [39], such as LIMES or Silk [30], that conceptually maps nominal data to values of the semantic model.

3. *Publishing the linked data API*: A controller API that follows the CRUD (Create-Read-Update-Delete) pattern to consume the LD resources can be generated in this phase. As a consequence, an extended description for the API mediator functionality are automatically produced. Yet the legacy web application might have existing operation implementations that already provide the right data that feeds the API mediator. If making such implementations public preserves the privacy requirements, they can be immediately revealed. Otherwise, they must be submitted to the next step.

4. *Privacy-preserving linked data access*: Since access to the generated linked data should be privacy-preserving, the appropriate PPDP technique must be implemented here. The approach for collaborative data sanitizing is explained in detail at the end of this section. It must be noted that, for the aim of this work, only the privacy preservation aspect has been considered. Nonetheless, other non-functional quality features (e.g. secure access control) can be also pipelined in this phase as additional quality requirements.

When it comes to implementing the data sanitizing protocol of step 4, two architectural concerns have to be considered: (1) who has *control* over the collaborative sanitizing protocol, and (2) how are the datasets *partitioned*. Fig. 5 depicts a two-dimensional classification framework that represents the coarse-grained options for both concerns.

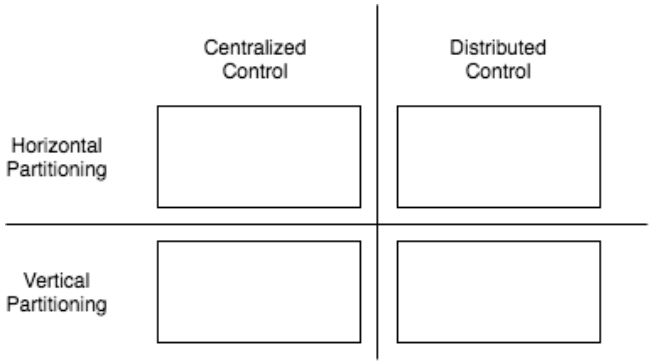

Fig. 5 Frame classification of collaborative sanitizing approaches according to the dataset partitioning and protocol control

A collaborative sanitizing protocol in which two or more distributed datasets are involved does not necessarily imply that control is also distributed. As explained above, there is often a need for a mashup coordinator to take decisions about if a given data publishing can be authorized, as well as for coordinating the sanitizing protocol. An example of centralized control is presented in [34], where the sanitizing process is carried out on a horizontally partitioned data mashup and managed by a coordinator. Another example of centralized control on horizontal partitioning is suggested in [22]. In this case, a leading part takes all decisions to recursively partition the quasi-identifier domain space in a top-down approach. On the other hand, a typical case of distributed control is proposed in [27], where the sanitizing process is cooperatively performed by all owners of a vertically partitioned data mashup, without a coordinator that manages the process or a fixed leading party that monopolizes the decision making. Nowadays, the growing role that is being played by the blockchain technologies for information registry and distribution eliminates the need of a centralized control, thus turning the spotlight on collaborative sanitizing protocols to distributed control approaches.

5 Cases of privacy-preserving linked data mashup publishing

As explained above, two kinds of dataset partitioning are considered when applying privacy-preserving sanitizing techniques for a distributed data mashup, namely horizontal partitioning and vertical partitioning. In horizontally partitioned datasets, each dataset custodian has a subset of the records defined over the same data attribute schema. Horizontal dataset partitioning case is common when several custodians have agreed upon a shared semantic model. For instance, Electronic Health Records (EHR) usually mash up data from several health organizations, such as hospitals and health care centers of different size and operating in different regions [9].

On the other hand, in vertically partitioned datasets, each data custodian has a subset of the attributes defined over the same set of records. They usually share an identifying attribute that enables to map records of the same individuals in the mashup. This is also a recurrent case in EHRs, since different stakeholders may keep a different data schema about the same individual, either at an inter-organizational

level (e.g. hospitals, clinical laboratories, radiological imaging centers, etc.) or intra-organizational level (e.g. physicians, pharmacists, nursing, diagnostic testing, etc.) [17]. Another common case of vertical partitioning is the Smart Cities application field. In the example described at the beginning of the paper, different players such as City Councils, energy and transport providers may form a vertically partitioned data mashup and their data needs to be sanitized if they are going to be exploited at the microdata level for an analytic purpose.

For instance, when looking at the MK Data Hub mashup, one can find relevant datasets with aggregated information about energy consumption[2] and demography[3] that can be query through a data-centric API[4]. Should the MK Data Hub aim at publishing microdata, to combine and publish such datasets would pose privacy concerns and can be exposed to *linking* attacks like the explained above. In MK Data Hub, for example, there might be records with `electricityConsumption` or `gasConsumption` equal to zero that might be exposed to an attack to discover the house addresses. To prevent such linking attacks, we can generalize *Carpenter* and *Carver* to *Wood worker* such that the (*Female*, *Carpenter*) individual becomes one of many female professionals.

The issue is that this generalization must be done collaboratively by both data holders. On the one hand, the integrate-then-sanitize approach must first integrate DS1 and DS2 and then generalize the DS table using sanitizing methods on a single table. This approach does not preserve privacy because any party holding the integrated table will know all private information from both parties. On the other hand, the sanitize-then-integrate approach first generalizes each table locally and then integrate the generalized tables. This approach does not guarantee k-anonymity to be achieved on the quasi-identifier (`sex`, `occupation`) by k-anonymizing on sex and occupation separately.

Regarding the control of the mashup sanitizing protocol, most existing application areas demand a central mashup coordinator to decide what data can be published and how. In smart cities scenarios like the MK Data Hub, for instance, given the number of datasets, it is difficult to envisage that, for any number of combinations, data custodians can agree a collaborative PPDP approach. Instead, the data coordinator or administrator, who has the responsibility to oversee all datasets, allows the combination of specific datasets and possibly part of their data models to carry out the PPDP procedure on the integrated mashup. In these cases, a centralized control approach, combined with a vertical or horizontal PPDP technique should suffice. In the EHR field, however, a centralized coordinator for privacy-preserving policies is not always available. For example, it is difficult to foresee a Europe-wide institution with the responsibility of control over the data that can be exchanged and mashed-up between datasets of different healthcare centers and service providers. Even at some nationwide level, such as Spain, this is hardly attainable. In such cases, a completely distributed control of the sanitizing protocol might be required when building up an EHR mashup composed by two or more individual EHR. The reasons

[2] https://datahub.mksmart.org/dataset/lower-layer-super-output-area-lsoa-domestic-electricity-consumption-2013-2/

[3] https://datahub.mksmart.org/dataset/mki-census-2011-demography/

[4] https://datahub.beta.mksmart.org/entity-lookup/

for needing a distributed control approach can be greater when blockchain-based technologies are mature enough to replace centralized EHRs and other data repositories with distributed ledgers [1]. These are cases of completely distributed control, either vertical or horizontally partitioned depending on the custody responsibilities for each involved dataset.

6 Discussion

Engineering an LD mashup publishing system involves two privacy-preserving functionalities. First, a configurable ontology mapping function that enables to bind nominal data in an existing application or dataset to the ontology concepts. This can be provided by existing interlinking solutions. And second, a sanitizing function that implements the privacy-preserving strategy at each dataset, thus avoiding to reveal sensitive data to the end user and other parties that intervene in the data mashup.

When publishing a data mashup, privacy preservation techniques must be carried out in collaboration between all data custodians involved in the mashup. Consequently, distributed collaboration protocols are important for the architectural design of data mashups and ETL approaches of big data systems. In particular, LD architectures have to be aware of data privacy concerns and implement privacy-preserving data publishing techniques in the presence of distributed data mashups.

A number of privacy-preserving data mashup algorithms have been proposed to securely integrate private data from multiple parties that collaborate in producing an integrated data mashup that satisfies a given k-anonymity requirement [27,24]. However, in the solutions proposed to integrate the datasets [27], data publishers need to exchange the identifying attribute with each other involved in the sanitizing process. As a consequence, publishers have additional information (i.e. a link between the identifying attribute and the sanitized quasi-identifier) to the published in the eventually sanitized dataset, thereby violating one of the requirements of collaborative data sanitizing. To solve this issue, it would be interesting to explore the use of pseudonyms for the identifier attributes during the sanitization process.

Distributed implementations of PPDP algorithms are becoming more relevant as microservice-based cloud architectures are implemented in the Semantic Web [23]. What is more disruptive, the blockchain paradigm change brings lots of implications concerning the distributed nature of data storage services and the privacy of distributed ledgers [36]. As long as metadata and linked data are going to be stored on blockchain technologies [33], there is the need for a completely distributed PPDP solution.

The solution proposed in this paper considers privacy protection in the engineering process as a primary requirement, as recommended by [13], instead of after the deployment of a new technology, such as the deployment of mobile devices with location-based services, sensor networks and social networks. The proposed method provides a privacy-preserving tool for individuals as well as for data publishers, by enabling record owners to have the opportunity to configure the protection of their own private information, before this is aggregated in a data mashup [35].

References

1. Azaria, A., Ekblaw, A., Vieira, T. & Lippman, A. (2016). MedRec: Using Blockchain for Medical Data Access and Permission Management, IEEE Int. Conf. on Open and Big Data, Vienna, Austria (pp. 25–30).
2. Batet, M., Erola, A., Sánchez, D., & Castellà-Roca, J. (2013). Utility preserving query log anonymization via semantic microaggregation. *Information Sciences*, 242, 49–63.
3. Batet, M., Erola, A., Sánchez, D., & Castellà-Roca, J. (2014). Semantic Anonymisation of Set-valued Data. In Proceedings of the 6th International Conference on Agents and Artificial Intelligence - Volume 1 (pp. 102–112). Portugal: SCITEPRESS - Science and Technology Publications.
4. Batet, M., Sánchez, D. (2015) A review on semantic similarity, in: *Encyclopedia of Information Science and Technology*, 3rd Edition, IGI Global, pp. 7575-7583.
5. Bizer, C., & Cyganiak, R. (2006). D2R Server - Publishing Relational Databases on the Semantic Web. In Proc. of the 5th International Semantic Web Conference, USA.
6. Casino, F., Domingo-Ferrer, J., Patsakis, C., Puig, D., & Solanas, A. (2015). A k-anonymous approach to privacy preserving collaborative filtering. *Journal of Computer and System Sciences*, 81(6):1000–1011.
7. Conway, R., & Strip, D. (1976). Selective partial access to a database. In Proceedings of the 1976 annual conference (pp. 85-89). ACM.
8. Daga, E., D'Aquin, M., Adamou, A., & Motta, E. (2016). Addressing exploitability of Smart City data. In IEEE International Smart Cities Conference (pp. 1–6).
9. Dechene, J. C. (2010). The challenge of implementing interoperable electronic medical records. Annals of Health Law, 19(1), 195–203.
10. Defays, D., Anwar, M. N. (1998) Masking microdata using micro-aggregation, *Journal of Official Statistics* 14(4):449–461.
11. Domingo-Ferrer, J., Martínez-Ballesté, A., Mateo-Sanz, J.M., Sebé, F. (2006) Efficient multivariate data-oriented microaggregation, *VLDB Journal*, 15(4):355–369.
12. Erling, O., & Mikhailov, I. (2009). RDF Support in the Virtuoso DBMS. In T. Pellegrini, S. Auer, K. Tochtermann, & S. Schaffert (Eds.), *Networked Knowledge - Networked Media. Integrating Knowledge Management, New Media Technologies and Semantic Systems* (Vol. 221, pp. 8–24). Springer.
13. Fung, B. C. M., Wang, K., Chen, R., & Yu, P. S. (2010). Privacy-preserving Data Publishing: A Survey of Recent Developments. *ACM Comput. Surv.*, *42*(4), 14:1--14:53.
14. Fung, B. C. M., Wang, K., Fu, A. W.-C., & Yu, P. S. (2010). *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques* (1st ed.). Chapman & Hall/CRC.
15. Goryczka, S., Xiong, L., & Fung, B. C. M. (2011). m-Privacy for collaborative data publishing. In *7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)* (pp. 1–10).
16. Groth, P., Loizou, A., Gray, A. J. G., Goble, C., Harland, L., & Pettifer, S. (2014). API-centric Linked Data integration: The OpenPHACTS Discovery Platform case study. *Web Semantics: Science, Services and Agents on the World Wide Web*, *29*, 12–18.
17. Häyrinen, K., Saranto, K., & Nykänen, P. (2008). Definition, structure, content, use and impacts of electronic health records: A review of the research literature. *International Journal of Medical Informatics*, 77(5), 291–304.
18. Heath, T., & Bizer, C. (2011). *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool.
19. Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E. S., Spicer, K., & de Wolf, P.-P. (2012). *Statistical Disclosure Control*. Wiley.
20. Jain, P., Gyanchandani, M., & Khare, N. (2016). Big data privacy: a technological perspective and review. *Journal of Big Data*, 3(1), 25.
21. Joksimovic, S., Jovanovic, J., Gasevic, D., Zouaq, A., & Jeremic, Z. (2013). An empirical

evaluation of ontology-based semantic annotators. In Proc. of the 7th Int. Conf. on Knowledge Capture (pp. 109–112).
22. Jurczyk, P., & Xiong, L. (2009). Distributed Anonymization: Achieving Privacy for Both Data Subjects and Data Providers. In Proc. of the 23rd Annual IFIP WG 11.3 Working Conf. on Data and Applications Security (pp. 191–207). Berlin: Springer-Verlag.
23. Khalili, A., Loizou, A., & van Harmelen, F. (2016). Adaptive Linked Data-Driven Web Components: Building Flexible and Reusable Semantic Web Interfaces. In H. Sack, E. Blomqvist, M. D'Aquin, C. Ghidini, S. P. Ponzetto, & C. Lange (Eds.), 13th International Semantic Web Conference (pp. 677–692). Springer.
24. Khokhar, R. H., Fung, B. C. M., Iqbal, F., Alhadidi, D., & Bentahar, J. (2016). Privacy-preserving data mashup model for trading person-specific information. *Electronic Commerce Research and Applications*, 17, 19–37.
25. Kim, J. J. (1986). A method for limiting disclosure in microdata based on random noise and transformation. In Proceedings of the section on survey research methods (pp. 303-308). American Statistical Association.
26. Martínez, S., Sánchez, D., & Valls, A. (2013). A semantic framework to protect the privacy of electronic health records with non-numerical attributes. *Journal of Biomedical Informatics*, 46(2): 294–303.
27. Mohammed, N., Fung, B. C. M., Wang, K., & Hung, P. C. K. (2009). Privacy-preserving Data Mashup. Proc. of 12th Int. Conf. on Extending Database Technology: Advances in Database Technology (pp. 228–239). New York, NY, USA: ACM.
28. Moore, R. A. (1996) Controlled data swapping techniques for masking public use microdata sets, Statistical Research Division Report Series RR 96-04, U. S. Bureau of the Census, Washington, DC, 1996.
29. Pol, K., Patil, N., Patankar, S., & Das, C. (2008). A Survey on Web Content Mining and extraction of Structured and Semistructured data. In *Emerging Trends in Engineering and Technology* (pp. 543–546).
30. Rajabi, E., Sicilia, M.A., & Sánchez-Alonso, S. (2014). An empirical study on the evaluation of interlinking tools on the Web of Data. *Journal of Information Science*, 40(5), 637–648.
31. Rodríguez-García, M., Batet, M., & Sánchez, D. (2017). A Semantic Framework for Noise Addition with Nominal Data. *Knowledge Based Systems*, 122(C), 103–118.
32. Samarati, P., & Sweeney, L. (1998). Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report SRI-CSL-98-04, SRI International.
33. Sicilia, M. A., Sánchez-Alonso, S., & García-Barriocanal, E. (2017) Deploying metadata on blockchain technologies, Metadata and Semantics Research Conference, Tallin.
34. Soria-Comas, J., & Domingo-Ferrer, J. (2015). Co-utile Collaborative Anonymization of Microdata. In V. Torra & T. Narukawa (Eds.), 12th Int. Conf. on, MDAI, Skövde, Sweden, September 21-23 (pp. 192–206).
35. Tao, Y. & Xiao, X. (2006). Personalized privacy preservation. In *Privacy-Preserving Data Mining*, Advanced Database Systems book series (pp. 461–485). Berlin: Springer-Verlag.
36. Third, A., & Domingue, J. (2017). Linked Data Indexing of Distributed Ledgers. Proc. of the 26th International Conference on World Wide Web Companion (pp. 1431–1436).
37. Torra, V. (2011). Towards knowledge intensive data privacy. In Data Privacy Management and Autonomous Spontaneous Security (Vol. 6514, pp. 1–7). Spriger-Verlag.
38. Wang, K., Fung, B. C. M., & Dong, G. (2005). Integrating Private Databases for Data Analysis. Proc. of IEEE Int. Conf. on Intelligence and Security Informatics (pp. 171–182).
39. Wölger, S., Siorpaes, K., Bürger, T., Simperl, E., Thaler, S., & Hofer, C. (2011). *A survey on data interlinking methods*. TR 2011-03-31, Semantic Technology Institute.
40. Xu, L., Jiang, C., Wang, J., Yuan, J., & Ren, Y. (2014). Information Security in Big Data: Privacy and Data Mining. *IEEE Access*, *2*, 1149–1176.