Privacy model and annotation for DaaS

Michael Mrissa, Salah-Eddine Tbahriti
Université de Lyon, CNRS
Université Lyon 1, LIRIS UMR5205
Villeurbanne, F-69622, France
email: {firstname.lastname}@liris.cnrs.fr

Hong-Linh Truong
Distributed Systems Group
Vienna University of Technology
Vienna, Austria
email: truong@infosys.tuwien.ac.at

***Abstract*—Data as a Service (DaaS) builds on service-oriented technologies to enable fast access to data resources on the Web. However, this paradigm raises several new concerns that traditional privacy models for Web services do not handle. First, the distinction between the roles of service providers and data providers is unclear, leaving the latter helpless for specifying and verifying the enforcement of their data privacy requirements. Second, traditional models for privacy policies focus only on the service interface without taking into account privacy policies related to data resources. Third, unstructured data resources, as well as user permissions and obligations related to data that are utilized in DaaS are not taken into account.**

In this paper, we study data privacy as one of these concerns, which relates to the management of private information. The main contribution of this paper consists in: 1) devising a model for making explicit privacy constraints of DaaS, and 2) on the basis of the proposed privacy model, developing techniques that allow handling the privacy concern when querying DaaS. We validate the applicability of our proposal with some experiments.

I. Introduction

Building on the advantages of the service-oriented model (syntactic interoperability and programmatic access to remote functions), the concept of Data as a Service (DaaS) is now widely developed, as we can observe several endpoints available on the Web, such as StrikeIron data service[1], Infochimps data service[2], and the UN Data API project[3]. However, the central place of data in DaaS draws the attention to several concerns that are already well-known in the database domain, such as data quality, data context, etc. In order to support the data consumer in selecting DaaSs and correctly utilizing data offered by DaaSs, the various concerns of DaaS, including specific concerns such as data semantics, quality and usage, and more traditional QoS concerns such as performance or price, should be made explicit [1]. In this paper, we study data privacy as one of these concerns, which relates to the management of private and sensitive information. Indeed, data privacy is of primary importance as a major limitation to a massive adoption of DaaS [2]. In effect, while the possibility to query DaaSs brings many advantages to the users, it also increases the risks of data disclosure with the combination of several data sources. Protection of data privacy then becomes a central problem.

[1]http://www.strikeiron.com/
[2]http://infochimps.org/
[3]http://www.undata-api.org/

While several techniques have been proposed for data privacy-protection, e.g., before data are published [3], as well as several data privacy models have been proposed for Web services [4], [5], [6], [7], we observe two issues: (i) a clear gap of data privacy-preservation in the lifecycle of data publishing through DaaS and (ii) a lack of suitable techniques to deal with different types of data offered by current DaaSs in the Internet and cloud environments. With respect to the first issue, existing privacy models are typically associated with Web services but they do not address the privacy concern at the data provider level, therefore data providers are totally dependent from service providers to specify and enforce their privacy policies. In the second issue, current Web services privacy models do not support unstructured data, such as documents and zipped dataset, which are very popular in DaaSs and are typically offered by REST-based DaaSs. Another point is that existing models do not separate from service providers and data providers and do not provide data rights associated with data offered by Web services, while data rights are key factors to invoke third parties in the enforcement of data privacy concerns. To address these issues, the main contribution of this paper consists in: 1) devising a privacy model for making generic, explicit privacy constraints of DaaS and data providers, also covering explicit techniques used by DaaSs, data rights, unstructured data, and 2) based on the proposed model, developing techniques that allow handling the privacy concern when querying DaaSs.

This paper is organized as follows. Sect. II overviews related works and shows the need for privacy-aware DaaS querying. Sect. III shows our model for representing privacy constraints for DaaS, Sect. IV details how such privacy constraints are integrated into DaaS descriptions. Sect. V shows our experiments and discusses our results, and Sect. VI highlights some trends for future work.

II. Motivation and Related Work

A. Motivation

Our motivation is based on special characteristics of DaaS and the nature of data in DaaS. In effect, both impact the management of privacy concerns, and the privacy model presented in this paper should take into consideration this impact. In our view, the nature of data in DaaS is characterized by the following aspects: domain (i.e. business, e-science, and e-government), form (e.g., structured, dataset, and document), and purpose (e.g., free,

Published Privacy Requirements		Data Provider's Purpose			Data Form	
Category	**Requirements**	**Organizational work**	**Pay-per-use**	**Free/Public**	**Structured**	**Unstructured**
concern	privacy-preserving methods		+	+	+	+
	types of privacy data	+	+	+	+	+
	data rights		+	+	+	+
scope	individual data resources	+	+	+	+	+
	service operation	+	+	+	+	
	service as a whole	+	+	+	+	

Table I
REQUIREMENTS FOR DAAS PROVIDERS TO PUBLISH PRIVACY CONCERNS

commercial, and inter-organizational work). DaaS also makes a specific distinction with respect to the actors it interacts with. The DaaS provider (service provider) is clearly distinguished from the data provider, which could be, for example, public organizations, enterprises, and individual persons. Therefore, DaaS providers have to provide an extensible and customizable mechanism for data providers to make sure their published data is compliant with privacy-preserving rules and to guarantee data consumers know the restrictions applied on the use of data offered by DaaSs.

While DaaSs still are Web services, they present some characteristics that require extensions to typical privacy models developed for Web services. For example, within a DaaS, privacy policies may be associated to datasets (data resources) owned by different data providers, even within the same DaaS. Such a privacy concern is currently not handled in service-based environments where privacy models are related to service operations and I/O messages.

With respect to deployment, DaaS could be (i) part of (multi-)organizations (e.g., in e-government and e-healthcare), where access to private information is regulated with user roles and (ii) on the Internet and cloud with/without roles (like in cloud or several public data services).

With respect to the data offered, DaaS could be accessed as a structured-based DaaS (i.e. XML content, can be queried) or it could deliver unstructured data (zip or spreadsheet files). Structured-based DaaSs are well supported in (multi-)organizations with respect to privacy (most Web services), while unstructured-based DaaSs typically provide unstructured datasets that encapsulate data in a particular (compressed or proprietary) format. Therefore, existing privacy models that focus on the service, operation and I/O levels are not sufficient. New models that focus on data and handle its different forms are required.

With respect to data usage, there is a need for a regulation that restricts users from accessing or divulgating sensitive data. For instance, DaaS like Infochimps[4] provides several datasets but cannot ensure that privacy rules related to the delivered datasets will be respected.

Table I summarizes requirements for publishing privacy-related information in DaaSs. Such requirements impact DaaS providers on the ways privacy should be dealt with, and require new approaches for ensuring the respect of data privacy requirements. In the following, we propose a privacy model for DaaS that takes into account the diverse aforementioned aspects (type of deployment, form of data, usage regulation).

[4]http://www.infochimps.org/

B. Related work

1) Modeling the Privacy Concern for DaaS: In [5] privacy only takes into account a limited set of data fields and rights. Data providers specify how to use the service (mandatory and optional data for querying the service), while individuals specify the type of access for each part of their personal data contained in the service: *free*, *limited*, or *not given* using a DAML-S ontology. This work is very relevant to our work. However, privacy preferences do not include the point of view of individuals (data providers) over data usage restrictions.

In [6], Ran propose a discovery model that takes into account functional and QoS-related requirements, and in which QoS claims of services are checked with external components that act as *certifiers*. The authors refer to the privacy concern with the term *confidentiality*, and some questions are raised about how the service makes sure that the data are accessed and modified only by authorized personals.

The approach described in [7] is based on the definition of fine-grain security markup of service parameters in profile and process models by the addition of annotations about the security and privacy policies of services expressed in the logic-based language *Rei* [8]. A policy is utilized in service selection and invocation. OWL-S profile is then extended with policies. In this work, privacy constraints are not related to the published data but rather to the service.

2) Privacy and DaaS Composition: While one of the major advantages of SOA is the possibility to compose services, the combination of data originating from several DaaSs may increase the risks of privacy violations. As a consequence, privacy has also been explored in the context of DaaS composition.

The approach in [4] proposes an ontology-based declarative framework for discovering, composing and querying government Web Services while respecting privacy. Three types of privacy are discussed: user privacy, service privacy and data privacy. By allowing individuals to describe their privacy preferences, the system provides mechanisms that control access to this information on both client and service sides. Policy enforcement is still an open problem and access control mechanisms are not sufficient to solve privacy aspects such as data retention obligation.

Therefore, we rather rely on a formal privacy model that is backed with access control mechanisms for handling service and data privacy.

A composition of DaaSs is also a workflow. Gil et al. [9] describe a framework for enforcing data privacy in workflows. In [10], the use of private data is reasoned for workflows. Privacy-preservation for data mashup is represented in [11]. Lee et al. [12] discuss the integration and verification of privacy policies in SOA-based workflows. Data mashups and workflows focus on using algorithms (such as k-anonymity) for preserving privacy, while in our work we go further and propose a model that also takes into account usage restrictions.

3) Privacy and Data Integration: In the field of data integration, several algorithms have been proposed to ensure data privacy, such as k-anonymity [13] and alternative approaches [14] where sensitive attributes are preserved from identification. Some framework are also proposed in the field of data mining [15], relying on privacy policies, views and purpose ; and in peer data management systems (PDMS) where data are broadcasted over the network using *noise insertion* and *commutative encryption methods* to ensure its non-disclosure.

However, the works that deal with privacy in the context of data integration deal with structured data. Hence, they apply data transformation algorithms, or they rely on role-based mechanisms to ensure privacy compliance. We notice the mechanisms for handling privacy in the case of unstructured data or cloud-based DaaS are missing.

III. A MODEL FOR PRIVACY CONCERN

When data is published via DaaS, it is the responsibility of the data publisher to ensure that the data to be published will be compliant with data privacy laws. Therefore, internally, each DaaS can implement different techniques to enforce data privacy, such as those described in [3], for data providers. Our particular concern here is that, in order to support the data consumer to comply with the privacy laws by means of querying and to understand limitations due to such laws, both DaaS service and data providers have to publish data privacy capabilities that might be associated with their services and published data. The *data privacy capabilities* of DaaS describe how a DaaS can ensure privacy data and support privacy-related data query.

First, we assume that each data provider knows the different types of privacy concerns that exist in their data (e.g., using mining, data provenance, data schemas, etc.). Second, we assume that DaaS has some internal data privacy enforcement. Therefore, in this section, we only focus on published privacy concerns that are important to data consumers. The published privacy concerns of DaaS are strongly dependent on how the DaaS provider supports privacy-preserving within its DaaS. However, we just provide a publishing mechanism for DaaS providers to describe the capability of their DaaS with respect to data privacy issues.

In our model, we consider that a service offers several service operations, each operation will process one or more data resources. We will focus on two types of data resources:

- structured data resources: a data resource is represented in a structured way, e.g., a complex XML data type, a relational data record, or a relational database table.
- unstructured data resources: a data resource is represented in unstructured way, such as images and zip file, so that its content can not be queried.

These two types of data resources are typically provided by DaaS. The data consumer wants to retrieve relevant data resources. With respect to data privacy, both service providers and data consumers want to ensure that they comply with privacy laws.

Given a DaaS, let DPC be the set of data privacy capabilities, $DPC = \{dpc_1, dpc_2, \cdots, dpc_n\}$. A data privacy capability, dpc is described as $dpc = (CPI, scope)$, where CPI is a set of conditions on privacy information, $scope$ describes the level of dpc as explained in the following.

We consider three levels of data privacy capabilities (DPC), $scope = \{service, operation, data\ resource\}$

- the service as a whole: privacy capabilities apply to all data resources returned by any service invocation. One example of such capabilities is *all names are anonymous*.
- service operation: privacy capabilities apply to all invocations of specific service operation. One example is that *emails in all data resources of subscribers are nullified*.
- data resource: privacy capabilities apply to data resources. One example is that *the real name of the user has been changed*.

The first two levels of privacy capabilities are managed by DaaS service providers, as the service and its operations are provided by the DaaS provider. The last level is managed by the data provider because the privacy capabilities are associated with individual data resources. As a result of these different responsibilities in managing privacy capabilities, we need a privacy model that is suitable to both DaaS providers and data providers.

Now we discuss the conditions on data privacy information (CPI). The conditions are established based on type of data and privacy operations and permissions. For the types of data that should be considered in privacy-preserving, we propose to rely on a privacy data tree (PDT), stimulated by a context dimension tree discussed in [16], the data categories in P3P[5] (such as “physical”,”financial”, “health”), and linked data models [17]. A conceptual view of a data privacy tree is described in Fig. 1. PDT includes domain-independent nodes, domain-specific nodes, and custom nodes. Examples of nodes in domain-independent subtree are personal information, financial information and health information. The PDT can be obtained by using different means, such as data mining, user specification, or pre-defined ontology. Domain-independent nodes specify

[5]http://www.w3.org/TR/P3P/

common types of private and sensitive data while domain-specific nodes specify types of private and sensitive data for particular domains. In our view, a domain specific node of PDT should be specified by privacy experts in that domain. In addition to that, we also consider custom nodes which are specific to particular DaaS or data providers. By utilizing several domain-specific, domain-independent and custom PDT nodes, several possibilities for specifying privacy relevant data can be defined. Furthermore, by using PDT the provider of DaaS can also specify *data rights* to indicate whether its data can or cannot be combined with other potential privacy data not provided by the DaaS but specified in the PDT. In our work, PDT is specified via an ontology which is incrementally built and is used differently by DaaS providers and data providers:

- DaaS provider: provides its own PDT. The provider also incrementally incorporates new domain-specific and domain-independent PDTs obtained from its data providers or specified for its supporting types of data.
- data provider: specifies privacy capabilites of its data based on its own PDT or DaaS provider's PDT. It allows DaaS providers to incorporate its PDT into the PDT of DaaS providers.

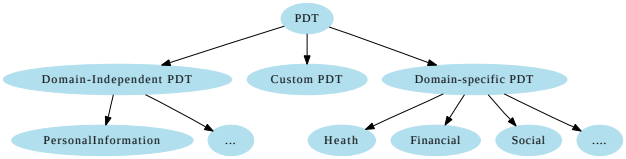

Figure 1. Conceptual model of the privacy data tree

Privacy operations are built upon existing privacy-preserving techniques (based on best practices). Examples of existing techniques are k-Anonymity [13], l-Diversity [18], and t-Closeness [19]. Similarly, data permission can be applied to sensitive data. The privacy data permissions are defined based on data permission and data licensing, such as to allow to use in research but not commercial or specify the responsibility of ensuring privacy policies of data consumers. Table II shows some examples of data permissions in literature.

Name	Description
non-commercial use	only use for non-commercial purposes
no-distribution	no distribution with a third party
no-integrity	protected from being created, changed or deleted by those who do not have permission to do so
no-linkage	use without linkage or composition with other sources

Table II
EXAMPLE OF DATA PERMISSION FOR DATA CONSUMERS

Let $PO = \{po_1, po_2, \cdots, po_n\}$ be the set of possible privacy operations. Let $UP = \{up_1, up_2, \cdots, up_n\}$ be the set of data permissions. A CPI for a dpc is defined by specifying possible operations and permissions applying to data items specified in PDT. Basically, $CPI = \{po(pdt) \cup up(pdt)\}$ where $po \in PO$, $up \in UP$, and $pdt \in PDT$. Conditions based on data privacy operations will give detailed information about changes that have been applied to data returned to data consumers (so data consumers can be sure that they do not worry about data compliance or can deal with missing/hiding/anonymous information). The impact of data privacy permission is that it requires the data consumers to perform certain data privacy-compliant responsibilities.

After having the concept, we can build our implementation by using existing vocabulary from PRIME ontologies[6], P3P, or Dublin core. Our data permissions are based on the Open Digital Rights Language (ODRL)[7] which allows describing digital rights management (DRM). This will involve the mapping from data privacy operations, privacy data tree, and conditions on privacy information to existing terms, vocabulary and concepts. Our prototype of the above-mentioned privacy capabilities model is based on OWL and RDF. Fig. 2 presents an overview of the main classes in our implementation. The PDT node links to data elements which can be specified inside or externally linked to the model.

IV. ANNOTATING DAAS DESCRIPTIONS WITH PRIVACY INFORMATION

The model developed above provides the necessary theoretical background to represent privacy capabilities. However, it is necessary to make these capabilities available to the users of DaaSs (human or agents). To do so, we link privacy capabilities to services via an annotation of their descriptions with the privacy capabilities of the service. In the following, we explain how we annotate the major description formats for DaaS (WSDL and REST annotations) according to the aforementioned model.

A. Privacy Annotation for WSDL-based DaaS

As WSDL 2.0 is the lastest W3C recommendation we first describe our annotation for this language, before detailing the minor changes required for WSDL 1.1 retro-compatibility. WSDL 2.0 descriptions provide interesting annotation capabilities, as shown with SAWSDL[8].

According to the specification[9], WSDL 2.0 allows element- and attribute-based extensibility on all the elements of a description, as long as the annotating elements and attributes are defined in some external namespace (i.e. not http://www.w3.org/ns/wsdl). Thus, we need to explore these elements and spot the places where annotations with privacy capabilities are the most relevant.

First, we choose to annotate WSDL 2.0 descriptions under the `interface` element that describes the abstract part of the service, in order to remain free from the different implementations described in the `binding` part. Then, considering our model and in particular its `scope` attribute, we choose to annotate WSDL descriptions at the three following places: `interface`, `operation`,

[6]https://www.prime-project.eu/ont/
[7]http://www.w3.org/TR/odrl/
[8]http://www.w3.org/TR/sawsdl/
[9]http://www.w3.org/TR/wsdl20/

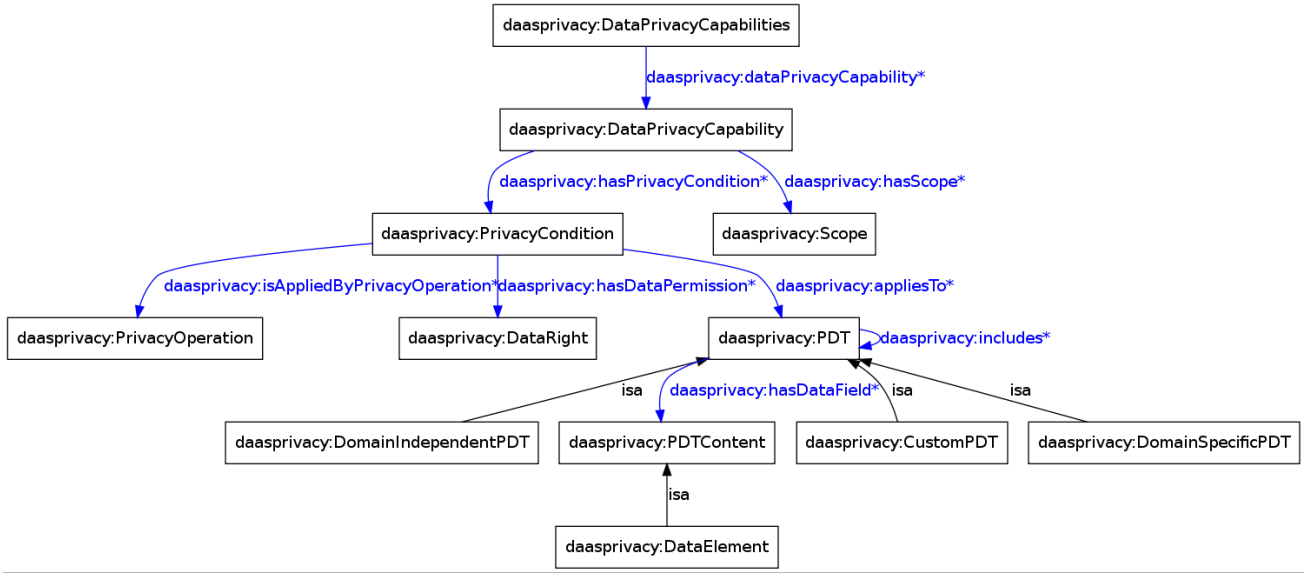

Figure 2. Main classes in the RDF of DaaS privacy capabilities prototype

`input` and `output`. In effect, these elements respectively correspond to the service-, operation-, and resource-levels defined in our privacy model.

For retro-compatibility sake, we also provide the following rules to adapt our WSDL 2.0 annotation to WSDL 1.1. The ”attrExtensions” element defined in SAWSDL is utilized to annotate WSDL 1.1 elements that do not support attribute extensibility, such as `operation` and `porttype`. The `porttype` element must be annotated as the ancestor of the `interface` WSDL 2.0 element, and message `part` elements must be annotated in replacement of `input` and `output` WSDL 2.0 elements. A sample annotation to WSDL 1.1 is presented in Listing 1, the complete file is available at http://liris.cnrs.fr/~mmrissa/doku.php?id=demos.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://jsonservice.
    liris.cnrs.fr/" name="JSONWSService"
  xmlns:pr="http://www.infosys.tuwien.ac.at/SOD1/
      dataconcerns/daasprivacy.owl#">
...
<portType name="JSONWS">
    <sawsdl:attrExtensions pr:dataprivacycapabilities=
        http://liris.cnrs.fr/~mmrissa/ECOWS/daasprivacy
        .rdf>
...
</portType>
...
</wsdl:definitions>
```

Listing 1. Excerpt of WSDL 1.1 annotation

B. Privacy Annotation for RESTful DaaS

RESTful services are typically described in a human-readable form via Web pages. Several works provide machine-interpretable descriptions for RESTful services, annotated into the HTML code and invisible to human readers [20], [21], [22]. In the following, we provide a brief overview of these works and we illustrate our privacy annotation with an extension to the MicroWSMO specification.

A simple model for describing RESTful services proposed in [20]. This model describes the service, its operations, their addresses (endpoint URI), HTTP methods and input/output messages, and serves as a support to other annotations as follows. MicroWSMO [22] adds ”model”, ”lifting” and ”lowering” attributes into hRESTS [20] descriptions in order to link I/O messages to ontology concepts and translate back and forth between concrete message encoding and semantic description. SA-REST [21] is a similar annotation that allows describing additional service aspects such as data formats of I/O messages or programming language bindings.

In order to enable the management of privacy concerns, we extend the MicroWSMO model proposed in [20] with additional classes and properties that describe privacy concerns. We define (Listing 2):

- an additional ”DataPrivacyPolicies” RDFS class that contains a link to the data privacy files attached to the service,
- an additional ”hasDataPrivacyPolicies” RDF property with domain {Service, Operation, Data Resource} and with range the DataPrivacyPolicies class.

```
@prefix hr: <http://www.wsmo.org/ns/hrests#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns
    #>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix pr: <http://www.infosys.tuwien.ac.at/SOD1/
    dataconcerns/daasprivacy.owl#>.

  # WSMO-Lite minimal service model
  wsl:Service a rdfs:Class.
  wsl:hasOperation a rdf:Property;
     rdfs:domain wsl:Service;
     rdfs:range wsl:Operation.
  wsl:Operation a rdfs:Class.
  wsl:hasInputMessage a rdf:Property;
     rdfs:domain wsl:Operation;
     rdfs:range wsl:Message.
  wsl:hasOutputMessage a rdf:Property;
     rdfs:domain wsl:Operation;
     rdfs:range wsl:Message.
```

```
wsl:Message a rdfs:Class.

# hRESTS properties added to the above model
hr:hasAddress a rdf:Property;
   rdfs:domain wsl:Operation;
   rdfs:range hr:URITemplate.
hr:hasMethod a rdf:Property;
   rdfs:domain wsl:Operation;
   rdfs:range xsd:string.
# a datatype for URI templates
hr:URITemplate a rdfs:Datatype.

# Extension for privacy description
pr:DataPrivacyPolicies a rdfs:Class.
pr:hasDataPrivacyPolicies a rdf:Property;
   rdfs:domain wsl:Service;
   rdfs:domain wsl:Operation;
   rdfs:domain wsl:Message;
   rdfs:range xsd:string.
```

Listing 2. Extended hRESTS service model in RDFS/N3

As explained in Section III, vocabularies for describing privacy concerns are partially domain-dependent. Thus, we advise to follow a RDFa-based syntax, as it is simpler for service designers to define their vocabularies in a namespace and then link the REST annotations to appropriate RDF elements. However, we highlight that our model extension -just like MicroWSMO- is not syntax-dependent, thus allowing both RDfa or microformat approach to be adopted.

C. Deployment of privacy policies

Both data providers and service providers need to agree on the best practice for attaching policies to the service, depending on the nature of the service and the nature of data. As shown in the following, policies are contained in RDF files, and the annotations contain a reference to the location of such a file. Direct annotation of service description with privacy policies is not recommended for maintenance purpose, in case these policies change. It is always the responsibility of the file reader to read all the privacy files of a service description and to determine the action to be taken. The annotations described above link privacy policies to WSDL and REST via different hooks in the description. These hooks have been chosen to correspond to the `scope` attribute of the privacy policy. Such a choice has the advantage to offer several alternatives for the deployment of privacy policies.

The first alternative is to group policies into the same file and to attach this file to the hook that offers the highest granularity level: the ”service” level. This alternative is interesting when the service does not rely on too many sources, so that a single privacy file is sufficient. The second alternative consists in grouping policies at the operation or message level. Such alternative is interesting in business corporations where each file reflects the policies attached to a specific operation, but it implies some redundancy when the same policy applies to several operations. The last alternative is to split policies into several files, each corresponding to a granularity level (service, operation, data resource). In such case, the maintenance of privacy policies is simplified and their clarity is increased with a clear separation between the different scopes of these policies. On the other hand, it requires several annotations at different places in the service description, and such decentralization raises consistency problems in case of conflicts between rules from different levels. In order to solve this problem, a priority on the smallest scope could be setup. For convenience, we have chosen the first deployment alternative in our experiments.

V. Experiments and Discussion

In order to validate our proposal, we have developed a sample scenario based on the Haiti earthquake dataset[10] from the Twitter.com social network. The size of this dataset is approximately 100MBs and its entries are in JSON format. The data includes several data fields relevant to privacy concerns such as `in_reply_to_screen_name`, `contributors`, `geo`, `name`, and `in_reply_to_user_id`. The published data already partially removes these sensitive data fields.

In our scenario, the data provider has bought the dataset from Twitter and is interested in publishing these data with different privacy protection levels, corresponding to different types of users (non-registered, registered, premium member etc.) that have restricted access to the different versions of data obtained from the original dataset. On top of these restrictions, Twitter has given several policies related to the usage of the released data. Users are given access to tweets published via the WSDL interface of our experimental DaaS.

This scenario highlights the different privacy requirements on the service and data provider levels:

- on the service provider level, application of its own and the data provider’s privacy requirements via privacy-preserving algorithms
- on the data provider level, specification of general limitations on data usage, privacy restrictions on some data values and limitations on the privacy algorithms to be applied on data

First, let us assume that the data provider wants to enforce the use of the `replacebyNULL` algorithm to preserve data privacy and, second, that there is an ontology in social science that describes the privacy data tree (PDT) for social networks (e.g., Twitter, Facebook, Youtube and Google Buzz). To fulfill the second assumption, we have built a simple PDT based on the recent proposed activity streams for social data in [23]. The data publisher wants to share the data as a free source for research on text mining but does not allow data consumers to combine this data with other social data sources because the data publisher is afraid of the use of techniques to de-anonymize social networks data [24]. These privacy policies are described in a single RDF-based policy file that is shown in Fig. 3. This file is referenced (linked) from the service description using the annotation proposed above. We have developed a Java™ servlet that acts as a user-friendly interface to the annotated DaaS and is hosted on a Glassfish™ server[11].

[10] http://infochimps.org/datasets/twitter-haiti-earthquake-data

[11] Demo and annotated descriptions of our sample service are available at http://liris.cnrs.fr/~mmrissa/doku.php?id=demos

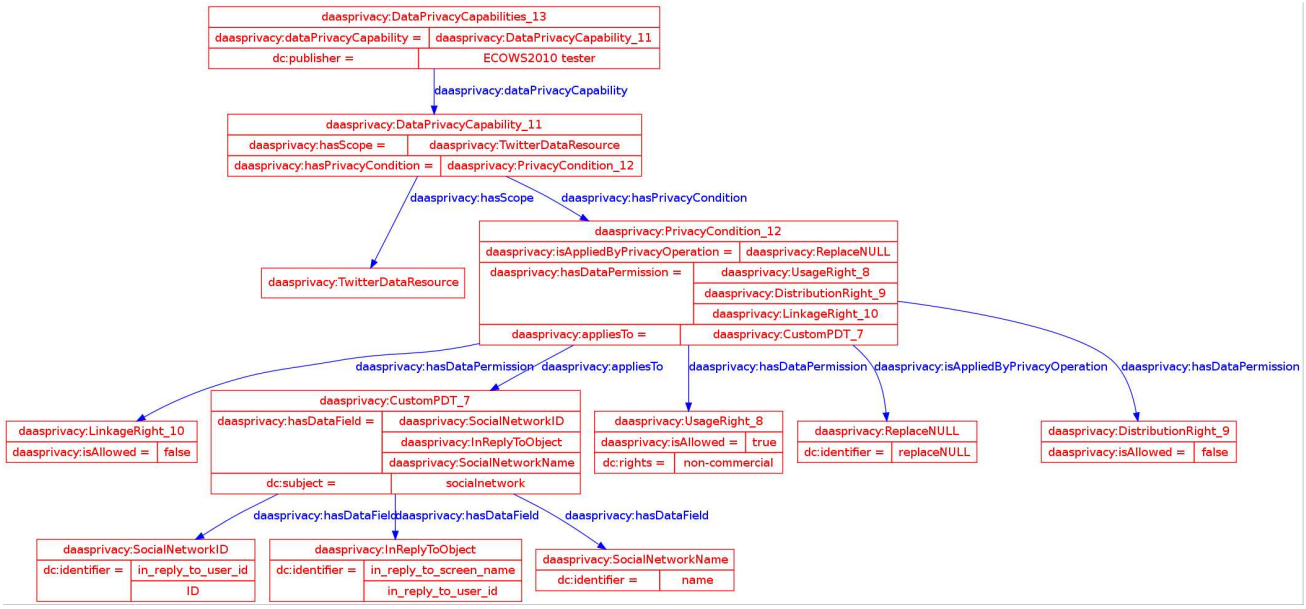

Figure 3. Example of privacy policies for twitter dataset

Now, in order to illustrate the use of privacy policies in DaaS services, let us explore the interactions between actors in our scenario:

- at any time, the data consumer may consult the DaaS description file and access to the linked privacy file to learn about the policies attached to this DaaS
- the data consumer sends a query to the DaaS
- the DaaS fetches data relevant to the query, and applies the privacy operations defined over the particular context of the query (data queried, type of user etc.)
- resulting data are sent back to the data consumer, with attached permissions or obligations if necessary.

In our scenario, service provider and data provider policies are combined to build the answer of the service. The service provider applies the following policies:

- non-registered users are given access to one tweet per query
- registered users are given access to several tweets per query but user names are replaced by NULL
- premium users are given access to several tweets per query with user names

and the data provider has specified the following policies:

- data must not be mixed with other data
- either name or localization of tweet authors are nullified
- only replacement by NULL is allowed on data values

In order to respect these policies, the provider has implemented the policies over the data queried. Access control has been implemented over the service, data is delivered together with a warning message on usage restrictions, and some values have been nullified according to the data provider's policy. The service holds a listener over the privacy file, so that when the data provider updates the file, the changes are directly reflected on the service. Please note that changes on data rights are easy to implement but substantial changes like the application of different privacy-preserving algorithms cannot be realized on-the-fly. Such perspectives are subject to future works.

VI. Conclusion

In this paper, we highlight several DaaS-related problems that traditional service-oriented technologies do not handle. We bring out the need for a clear distinction between the roles of service providers and data providers, for a better management of their privacy requirements. Also, we show the limitations of traditional Web service privacy models for taking into account privacy policies related to data resources, and for dealing with unstructured data resources, user permissions and obligations.

We address these problems and enable the management of the privacy concern in DaaS environments. We propose a model for representing privacy policies, together with annotations of the main service descriptions formats (WSDL and REST) with privacy policies. We illustrate the suitability of our model and show its concrete application with an experiment built on a use case using Twitter data.

Several possibilities are envisioned as future works. As short-term evolutions, on-the-fly reaction of the service provider to changes on privacy policies should improve our proposal, as well as tests on the fulfillment of the privacy requirements attached to data. Also, we intend to develop our annotation to other WSDL-based or REST-based service description formats. As a long-term evolution, an extension of our proposal to the context of DaaS composition is under study. The idea is to enable the composition of several DaaSs with privacy-aware mechanisms that allow enforcing individual privacy policies in the composition while respecting user permissions and obligations.

ACKNOWLEDGMENT

This work is partially supported by the Vienna Science and Technology Fund (WWTF), project ICT08-032.

REFERENCES

[1] H. L. Truong and S. Dustdar, "On analyzing and specifying concerns for data as a service," in *APSCC*, M. Kirchberg, P. C. K. Hung, B. Carminati, C.-H. Chi, R. Kanagasabai, E. D. Valle, K.-C. Lan, and L.-J. Chen, Eds. IEEE, 2009, pp. 87–94.

[2] L. F. Cranor, "Internet privacy - introduction," *Commun. ACM*, vol. 42, no. 2, pp. 28–31, 1999.

[3] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Comput. Surv.*, vol. 42, no. 4, pp. 1–53, 2010.

[4] A. Rezgui, M. Ouzzani, A. Bouguettaya, and B. Medjahed, "Preserving privacy in Web services," in *WIDM '02: Proceedings of the 4th international workshop on Web information and data management*. New York, NY, USA: ACM, 2002, pp. 56–62.

[5] A. Tumer, A. Dogac, and I. H. Toroslu, "A semantic-based user privacy protection framework for Web services," in *ITWP*, ser. Lecture Notes in Computer Science, B. Mobasher and S. S. Anand, Eds., vol. 3169. Springer, 2003, pp. 289–305.

[6] S. Ran, "A model for Web services discovery with QoS," *SIGecom Exchanges*, vol. 4, no. 1, pp. 1–10, 2003.

[7] L. Kagal, M. Paolucci, N. Srinivasan, G. Denker, T. Finin, and K. Sycara, "Authorization and privacy for semantic Web services," *IEEE Intelligent Systems*, vol. 19, no. 4, pp. 50–56, 2004.

[8] L. Kagal, T. Finin, and A. Joshi, "A policy based approach to security for the semantic Web," in *2nd International Semantic Web Conference (ISWC2003)*, September 2003.

[9] Y. Gil, W. Cheung, V. Ratnakar, and K. kin Chan, "Privacy enforcement in data analysis workflows," in *Proceedings of the Workshop on Privacy Enforcement and Accountability with Semantics (PEAS2007) at ISWC/ASWC2007, Busan, South Korea*, ser. CEUR Workshop Proceedings, T. Finin, L. Kagal, and D. Olmedilla, Eds., vol. 320. CEUR-WS.org, November 2007.

[10] Y. Gil and C. Fritz, "Reasoning about the appropriate use of private data through computational workflows," in *Intelligent Information Privacy Management, Papers from the AAAI Spring Symposium*, March 2010, pp. 69–74. [Online]. Available: gil-fri-aaai10ss.pdf

[11] N. Mohammed, B. C. M. Fung, K. Wang, and P. C. K. Hung, "Privacy-preserving data mashup," in *EDBT '09: Proceedings of the 12th International Conference on Extending Database Technology*. New York, NY, USA: ACM, 2009, pp. 228–239.

[12] Y. Lee, J. Werner, and J. Sztipanovits, "Integration and verification of privacy policies using DSML's structural semantics in a SOA-based workflow environment," *Journal of Korean Society for Internet Information*, vol. 10, no. 149, 09/2009 2009.

[13] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 571–588, 2002.

[14] X. Xiao and Y. Tao, "Anatomy: simple and effective privacy preservation," in *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment, 2006, pp. 139–150.

[15] C. Clifton, M. Kantarcioğlu, A. Doan, G. Schadow, J. Vaidya, A. Elmagarmid, and D. Suciu, "Privacy-preserving data integration and sharing," in *DMKD '04: Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*. New York, NY, USA: ACM, 2004, pp. 19–26.

[16] C. Bolchini, C. A. Curino, G. Orsi, E. Quintarelli, R. Rossato, F. A. Schreiber, and L. Tanca, "And what can context do for data?" *Commun. ACM*, vol. 52, no. 11, pp. 136–140, 2009.

[17] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data - the story so far," *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2009.

[18] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, p. 3, 2007.

[19] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *ICDE*. IEEE, 2007, pp. 106–115.

[20] J. Kopecký, K. Gomadam, and T. Vitvar, "hRESTS: An HTML microformat for describing RESTful Web services," in *Web Intelligence*. IEEE, 2008, pp. 619–625.

[21] J. Lathem, K. Gomadam, and A. P. Sheth, "Sa-rest and (s)mashups : Adding semantics to RESTful services," in *ICSC*. IEEE Computer Society, 2007, pp. 469–476.

[22] J. Kopecký, T. Vitvar, D. Fensel, and K. Gomadam, "hRESTS & MicroWSMO," STI International, Tech. Rep., 2009. [Online]. Available: http://cms-wg.sti2.org/TR/d12/

[23] M. Atkins, W. Norris, C. Messina, M. Keller, and R. Dolin, "Activity streams concepts and representations (draft)," http://activitystrea.ms/head/json-activity.html, July 5 2010, internet-Draft.

[24] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2009, pp. 173–187.