Privacy-Aware DaaS Services Composition

Salah-Eddine Tbahriti[1], Michael Mrissa[1], Brahim Medjahed[2], Chirine Ghedira[1], Mahmoud Barhamgi[1], and Jocelyne Fayn[3]

[1] Claude Bernard Lyon1 University, 69622, Villeurbanne, France
firstname.lastname@liris.cnrs.fr
[2] Department of Computer and Information Science, University of Michigan-Dearborn, 4901 Evergreen Road, Dearborn, MI 48128, USA
brahim@umd.umich.edu
[3] EA 4171: MTIC, Hopital Cardiologique de Lyon, Bat B13, 28 av. Doyen Lepine - 69677 BRON cedex, France
jocelyne.fayn@insa-lyon.fr

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 privacy concerns that traditional privacy models do not handle since they only focus on the service interface without taking into account privacy constraints related to the data exchanged with a DaaS during its invocation. In addition, DaaSs compositions may reveal also privacy-sensitive information. In this paper we propose a privacy formal model in order to extend DaaS descriptions with privacy capabilities. The privacy model allows a service to define a *privacy policy* and a set of *privacy requirements*. We propose also a privacy-preserving DaaS composition approach allowing to verify the compatibility between privacy requirements and policies in DaaS composition. We validate the applicability of our proposal with some experiments.

Keywords: Privacy, DaaS services, Composition, Dependency.

1 Introduction

Recent years have witnessed a growing interest in using Web services as a reliable medium for data publishing and sharing. This new type of services is known as *Data-as-a-Service* services [4] [17], corresponds generally to calls over data sources. While individual DaaS services may provide interesting information alone, in real scenarios like epidemiological studies, users' queries require the invocation of several services. The DaaS composition is a powerful solution for building value-added services on top of existing ones [15] [20]. In the context of our project PAIRSE[1] we proposed in [2] a mediator-based approach to compose DaaSs. In that approach the proposed mediator answers users complex queries by combining available DaaSs and carries out all the interactions between the

[1] This research project is supported by the French National Research Agency under grant number ANR-09-SEGI-008.

A. Hameurlain et al. (Eds.): DEXA 2011, Part I, LNCS 6860, pp. 202–216, 2011.

composed services. Depending on available DaaSs, the mediator may return a set of DaaS compositions all answering the same query. However, DaaS compositions in that approach may reveal privacy-sensitive information. Privacy preservation is indeed still one of the most challenging problems in DaaS composition. In this paper we address the privacy issue in DaaS composition. We propose a privacy formal model in order to extend DaaS descriptions with privacy capabilities. The privacy model, goes beyond traditional data-oriented models, by allowing a service to define a *privacy policy* (specifying how it treats its collected data) and a set of *privacy requirements* (specifying how it expects consumer services to treat its provided data) by defining a set of privacy rules. We propose also an annotation mechanism to link DaaSs to their defined privacy polices and requirements.

Component DaaSs in a composition may have different privacy concerns, thus leading to an incompatibility problem between the privacy policies and requirements of interconnected services. The second contribution is a privacy-aware DaaS Composition. We devise a compatibility matching algorithm to check the privacy compatibility among privacy requirements and policies within a composition. The compatibility matching is based on the notion of privacy subsumption and a cost model. A matching threshold is set up by a given service to cater for partial and total privacy compatibility.

Our paper is structured as follows. First, we overview related work in Section 2. We then describe our privacy model in Section 3. Then, we introduce the notion of compatibility between privacy policies and requirements in Section 4, and illustrate its importance in the context of DaaS composition and will show how our DaaS composition approach is extended within privacy-preserving in Section 5. We present our experiments in Section 6 and discuss future work in Section 7.

2 Related Work

Our work is inspired and informed by a number of areas. We briefly review the closely related areas below and discuss how our work leverages and advances the current state-of-the-art techniques.

2.1 Privacy Aware-Data Modeling

A typical example of modeling privacy is P3P [19] standard. It encodes privacy policies in XML for Web sites and specifies the mechanisms to locate and transport privacy policies. However, the major focus of P3P is to enable only Web sites to convey their privacy policies. The work in [18] aims at specifying DAML-S ontology to answer two questions: how sensitive the information is; and under what conditions the information has that sensitive degree. Regarding that, 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. However, privacy preferences do not include the point of view of individuals over the data usage. An

approach on the feasibility of achieving a balance between consumers privacy and provider search has been proposed in [21]. It allows client to collect, summarize, and organize their personal information into a hierarchical profile. Through this profile, the client controls which portion of its private information is exposed to the provider by adjusting a threshold. The work in [16] aims at protecting the content of client queries and the retrieved documents. It proposes a schema for a provider to perform similarity-based text retrieval while protecting clients search activities. In our work, privacy resource is specified and may be related to client, Data and Service providers levels, and not only to the provided data.

2.2 Privacy Aware-Composition

The works in services composition are closely inspired from workflow and Data mashups composition. In [7] a framework for enforcing data privacy in workflows is described. In [8], the use of private data is reasoned for workflows. Privacy-preserving mechanism for data mashup is represented in [13]. It aims at integrating private data from different data providers in secure manner. The authors in [12] discuss the integration and verification of privacy policies in SOA-based workflows. The previous approaches, related to data mashup and workflows, focus on using algorithms (such as k-anonymity) for preserving privacy of data in a given table, while in our work we go further and propose a model that also takes into account usage restrictions and client requirements. The works [9] [10] [6] propose using third parties as database service providers without the need for expensive cryptographic operations. However the proposed schemes do not allow queries to execute over the data of multiple providers and do not take into account the privacy issue regarding service provider and data consumer, which is the main focus of our work. In the filed of data integration, several efforts have been made to either preserve the privacy of individuals using sanitized techniques [1] [3] or to preserve the privacy of the datasource while running data integration algorithms over multiple databases using cryptographic techniques [5]such as *secure multi-party computation* and *encryption*. In contrast to the existing approaches, in this paper we introduce a service-oriented privacy model for DaaS that goes beyond "traditional" data-oriented privacy approaches. *Input/output* data as well as *operation* invocation may reveal sensitive information about services and hence, should be subject to privacy constraints.

3 Privacy Description Model

In this section, we propose a formal model to specify the privacy capabilities attached to DaaS service (simply service) description. With this model, a service $\mathtt{S}$ will define a *privacy policy* (noted as $\mathtt{PP}^{\mathtt{S}}$) specifying the set of privacy practices applicable on any collected data and *privacy requirements* (noted as $\mathtt{PR}^{\mathtt{S/T}}$) specifying the $\mathtt{S}$'s set of privacy conditions that a third-party service $\mathtt{T}$ must meet to consume its data. Indeed, privacy is a very subjective notion, for instance, a given service may consider an input parameter provided to a third-party service

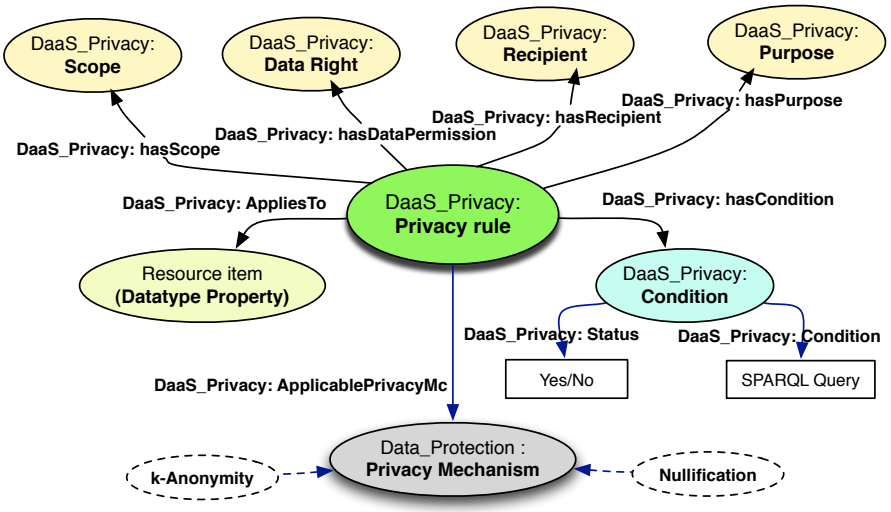

Fig. 1. Graph-based representation of a privacy rule

as private; another may view the information stating that the service invoked a specific operation of a given third-party service as private. Our model relies on the definition of privacy resource and privacy rule. Different types of information may be subject to privacy. We refer to such information as privacy resources (simply resources). To take into account the type of resources, we introduce the notion of privacy level (simply level). A graph-based representation of our model is presented in Figure 1.

3.1 Privacy Level

We define two privacy levels: data and operation. The data level deals with the data privacy. The resources (i.e., *Resource item* in Figure 1.) refer to input and output parameters of a service (e.g., defined in WSDL). For instance, service S_a has an operation op_a called *Patent-research* that takes as input a *user query* and returns as output *PatentResults*. The *user query* and *PatentResults* (i.e., input and output, resp.) may be both viewed as private; they are hence defined as data resources. The operation level copes with the privacy about operation's invocation. Information about operation invocation may be perceived as private independently on whether their input/output parameters are confidential or not [11]. For instance, let us consider a scientist that has found an invention about the causes of some infectious diseases, he invokes op_a to search if such an invention is new before he files for a patent. When conducting the query, the scientist may want to keep the invocation of op_a, query and result of query (i.e., the op_a input, op_a output resp.) private, perhaps to avoid part of his idea being stolen by a competing company. We give below the definition of the privacy level.

Definition 1. Let *rs* be a privacy resource of a service `S`. The privacy level L of *rs* is defined as follows: (i)L="data" if *rs* is an input/output of `S` operation; (ii)L="operation" if *rs* is information about `S`'s operation. ◊

3.2 Privacy Rule

The sensitivity of a resource may be defined according to several dimensions called *privacy rules*. We call the set of privacy rules *Rules Set*(RS). We define a privacy rule by a *topic*, *level*, *domain*, and *scope*. The *topic* gives the privacy facet represented by the rule. For instance, given the representation of privacy rule in Figure 1, the topic may include: the data right, the recipient and the purpose. The "purpose" topic states the intent for which a resource collected by a service will be used; the "recipient" topic specifies to whom the collected resource can be revealed. The *level* represents the privacy level on which the rule is applicable. The domain of a rule depends on its level. Indeed, each rule has one single level: "data" or "operation". We use the terms data and operation rule to refer to a rule with a "data" and "operation" level, respectively. The *domain* is a finite set that enumerates the possible values that can be taken by resources according to the rule's topic. For instance, a subset of domain for a rule dealing with the right topic is {"no-retention", "limited-use"}. The *scope* of a rule defines the granularity of the resource that is subject to privacy constraints. We consider two cases: operation and data rules. In the former case, several parts of a service log entry may be viewed as private. Services assign one of the values "total" or "partial" to the scope of their operation resources. If an operation resource is assigned a "total" scope for a given rule, then the whole entry of that operation in the service log is private. Otherwise (i.e., the assigned scope is "partial"), only the ID of the service that invoked the operation is private. In the case of data rules, we consider data resources as atomic. Hence, the only scope value allowed in this situation is {"total"}. "Partial" scope may also be considered for complex data resources (e.g., array structure). In this case, only part of an input/output parameter is private. However, this issue is out of the scope of this paper. Two rules at most are created for each topic: one for data and another for operations.

Definition 2. A privacy rule R_i is defined by a tuple (T_i, L_i, D_i, S_i) where:

- T_i is the topic of R_i,
- $L_i \in$ {"data", "operation"} is the level of the rule,
- D_i is the domain set of R_i; it enumerates the possible values that can be taken by T_i with respect to *rs*,
- S_i is the scope of R_i where S_i= {"total","partial"} if L_i="operation" and S_i={"total"} if L_i="data". ◊

For instance, we give two examples of rules R_1 and R_2, where R_1 = (T_1, L_1, D_1, S_1) with T_1="recipient", L_1="data", D_1= {`public`, `government` , `federal tax`, `research` } and S_1 ={"total"} R_2 = (T_2, L_2, D_2, S_2) with T_2="recipient", L_2="operation", D_1= {`public`} and S_2 ={"total", "partial"}. Our objective is to propose formal privacy model with a fine granularity that allows to add, modify (e.g., add new topic) and delete rules at anytime but also to check formally the compatibility between rules among service. It is therefore important to examine how privacy rules can be instantiated which is the focus of the subsequent section.

3.3 Privacy Assertion

The services will use privacy rules to define the privacy features of their resources. The application of a rule R_i=(T_i, L_i, D_i, S_i) on a resource rs is a *privacy assertion* $A(R_i, rs)$ where rs has L_i as a level. $A(R_i, rs)$ states the granularity of rs that is subject to privacy. The granularity g belongs to the scope S_i of the rule. For instance, g is equal to partial if only the ID of the operation invoker is private. $A(R_i, rs)$ also indicates D_i's values that are attributed to rs. For example, let us consider the rule R_1. A privacy assertion on rs according to R_1 may state that rs will be shared with government agencies and research institutions. We use the *propositional formula* (pf)"`government`" $\wedge$ "`research`" to specify such statement.

Definition 3. A privacy assertion $A(R_i, rs)$ on a resource rs is defined by the couple (pf, g); $pf = v_{ip} \wedge ... \wedge v_{iq}$ according to R_i=(T_i, L_i, D_i, S_i), where v_{ip},..., $v_{iq} \in D_i$; $g \in S_i$ is the granularity of rs. $\Diamond$

3.4 Privacy Policy

A service `S` will define a *privacy policy*, $\texttt{PP}^{\texttt{S}}$, that specifies the set of practices applicable to the collected resources. Each service has its own perception of what should be considered as private. Defining the privacy policy $\texttt{PP}^{\texttt{S}}$ of `S` is performed in two steps. First, the service `S` identifies the set (noted P_p) of all privacy resources. Second, `S` specifies assertions for each resource rs in P_p. Deciding about the content of P_p and the rules (from RS) to apply to each resource in P_p varies from a service to another. $\texttt{PP}^{\texttt{S}}$ specifies the way `S` (i) treats the collected resources (i.e., received through the mediator), (ii) expects any third-party services to treat resources provided as output when `S` operation will be invoked. We consider three cases: (a) rs is an input data, (b) rs is an output data, and (c) rs is an operation. If rs is an input data or operation (cases (a) and (c)), then $A(R_i, rs)$ states what will a service `S` do with rs according to R_i. If rs is an output data (case (b)), then `S` defines two assertions for rs according to R_i; the first, noted $A(R_i, rs^E)$, gives `S`'s expectation; the second, $A(R_i, rs^P)$, denotes `S`'s practice:

- Expectation: $A(R_i, rs^E)$ states what service `S` expects a third-party service to do with rs (provided as the output of `S` operation) according to R_i.
- Practice: $A(R_i, rs^P)$ states what service `S` will do with rs according to R_i.

For instance, let us consider a scientist that would like to conduct some experiments. Through mediator, the operation op_b of the service S_b will be invoked. op_b takes as input a `patient-disease` and returns as output the `SSN` (social security number) of the patient. The service S_b (which owns operation op_b) expects that third-party services will use the given output of op_b according to its expectations since `SSN` is a data with higher privacy sensitivity. We give below a definition of privacy policy and rs_k refers rs_k^E or rs_k^P if rs_k is an output data.

Definition 4. The *privacy policy* of a service `S` is defined as $\texttt{PP}^{\texttt{S}}$= $\{A_j(R_i, rs_k)$, $j \leqslant |\texttt{PP}^{\texttt{S}}|$, $i \leqslant |RS|$, $k \leqslant |P_p|$, $rs_k \in RS\}$

3.5 Privacy Requirements

A service S will define a *Privacy Requirements* $\mathtt{PR}^{\mathtt{S/T}}$ stating S's assertions describing how S expects and requires a third-party service T should use its resources. Before creating $\mathtt{PR}^{\mathtt{S/T}}$, S first identifies the set (noted P_c) of all its privacy-sensitive resources. $\mathtt{PR}^{\mathtt{S/T}}$ assertions describe the following requirements:

- The way S expects T to treat the privacy of input data, output data (e.g., experiment results returned by a service), and information about operation invocation; and
- The way S treats the privacy of any output data returned by T, through the mediator.

The aforementioned requirements are expressed via privacy assertions. Similarly to privacy policies, requirements on outputs express service's expectations (noted $A(R_i, rs^E)$) and practices (noted $A(R_i, rs^P)$). For instance, the output of operation invoked (owned by a third-party service) by S concerns primary S and S may be sensitive about how third-party service owned the invoked operation, will treat the output of the invoked operation regarding retention time. S may unequally value the assertions specified in $\mathtt{PR}^{\mathtt{S/T}}$. For instance, S owns SSN and zip_code data, S's requirements about SSN may be stronger than its requirements for zip_code. Besides, S may consider an assertion more essential than another, even if both assertions are about the same resource. For example, S may view the rule constraining the recipients of SSN as more valuable than the rule stating the duration for which the service can retain SSN. For that purpose, S assigns a weight w_j to each assertion $A(R_i, rs)$ in $\mathtt{PR}^{\mathtt{S/T}}$. w_j is an estimate of the significance of $A(R_i, rs)$. The higher is the weight, the more important is the corresponding assertion. Each weight is decimal number between 0 and 1. The total of weights assigned to all assertions equals 1:

- $\forall\ j \leqslant |\mathtt{PR}^{\mathtt{S/T}}| : 0 < w_j \leqslant 1$,
- $\sum_{j=1}^{k} w_j = 1$, where $k = |\mathtt{PR}^{\mathtt{S/T}}|$

In the real cases, the service S may be willing to update some of their privacy requirements. For instance, it may agree to relax constraints about the disclosure of their zip_code if the mediator requests that in exchange to offer it incentives such as discounts. However, S will probably be more reluctant to loosen conditions about the disclosure of their names. To capture this aspect, S stipulates whether an assertion $A(R_i, rs)$ is *mandatory* or *optional* via a boolean attribute M_j attached to assertion$A$.

Definition 5. The *privacy requirements* of a service S on third service T is defined as $\mathtt{PR}^{\mathtt{S/T}}$= { $(A_j(R_i, rs_k), w_j, M_j)$, $j \leqslant |\mathtt{PR}^{\mathtt{S/T}}|$, $i \leqslant |RS|$, $k \leqslant |P_c|$, $rs_k \in P_c$, $R_i \in RS$, w_j is the weight of A_j, M_j=True iff A_j is mandatory }. ◊

Other specific conditions, related to the context application, may be specified with SPARQL conditions (as showed in Figure 1). Furthermore, services may use

privacy protection mechanism (like k-anonymity) to sanitize its data(Figure 1). Due to the space limitation, details of these two characteristics will be discussed in another future work.

4 Privacy Compatibility

In this section we introduce the notion of compatibility between privacy policies and requirements according the notion of privacy subsumption.

4.1 Privacy Subsumption

Let us consider a rule R_i=(T_i, L_i, D_i, S_i). Defining an assertion $A(R_i, rs)$=(pf, g) for rs involving assigning value(s) from D_i to the propositional formula pf of A. The values in D_i are related to each other. For instance, let us consider the domain {`public`, `government`, `federal tax`, `research`} for a rule dealing with topic T_i="recipient". The value `public` is more general than the other values in D_i. Indeed, if the recipient of rs is declared public (i.e., shared with any entity), then the recipient is also government and research. Likewise, the value `government` is more general than `research` since the research **is-a** government agency. To capture the semantic relationship among domain values, we introduce the notion of *privacy subsumption* (noted $\sqsubseteq$). For instance, the following subsumptions can be stated: `government` $\sqsubseteq$ `public`; `research` $\sqsubseteq$ `government`. Note that privacy subsumption is transitive since it models the "is-a" relationship. We use $*$ to refer to the transitive closure of $\sqsubseteq$.

Definition 6. Let $D_i = \{v_{i1}, ..., v_{im}\}$ be the domain of a privacy rule R_i. We say that v_{ip} is subsumed by v_{iq} or v_{iq} subsumes v_{ip}, ($1 \leqslant p \leqslant m$ and $1 \leqslant q \leqslant m$) noted $v_{ip} \sqsubseteq v_{iq}$, iff v_{iq} is more general than v_{ip}. $\Diamond$

We generalize the notion of privacy subsumption to assertions. Let us consider an assertion $A(R_i, rs)$=(pf, g) representing an expectation of `S` (resp., `T`) and another assertion $A'(R_i^{'}, rs')$=(pf', g') modeling a practice of `T` (resp., `S`). In order for A and A' to be compatible, they must be specified on the same rule (R_i=$R_i^{'}$), the same resource (rs=rs'), and at the same granularity (g=g'). Besides, the expectation of `S` (resp., `T`) as stated by pf should be more general (i.e., subsumes) than the practice of `S` (resp., `T`) as given by pf'. In other words, if pf is true, then pf' should be true as well. For instance, if pf="`government` $\wedge$ `research`" and pf'='`government`", then $pf \Rightarrow pf'$ (where $\Rightarrow$ is the symbol for implication in propositional calculus). Hence, A is more general than A' or A subsumes A' (noted $A' \sqsubseteq A$).

Although some literals used in pf are syntactically different from the ones used in pf', they may be semantically related via subsumption relationships. For instance, let us assume that pf="`public` $\wedge$ `research`" and pf'="`federal tax`". Since `federal tax` $\sqsubseteq$ `public`, we can state that `public` $\Rightarrow$ `federal tax`. In this case, we can prove that $pf \Rightarrow pf'$ and hence, $A' \sqsubseteq A$. To deal with the issue of having different literals in propositional formulas, we use the following

property: if $v_{ip} * v_{iq}$ (i.e., v_{iq} directly or indirectly subsumes v_{ip}) then $v_{iq} \Rightarrow v_{ip}$.

Definition 7. Let us consider assertions $A(R_i, rs)=(pf, g)$ and $A'(R_i^{'}, rs')=(pf', g')$. A' is subsumed by A or A subsumes A', noted $A' \sqsubseteq A$, if $R_i=R_i^{'}$, $rs=rs'$, $g=g'$, and $pf \Rightarrow pf'$. $\Diamond$

4.2 Privacy Compatibility Matching Algorithm

The aim of Privacy Compatibility Matching algorithm *PCM* is to check that assertions in $\texttt{PR}^{\texttt{S/T}}$ and $\texttt{PP}^{\texttt{T}}$ are related via subsumption relationships (cf. Definition 7). As mentioned in 3.2 and 3.3, both $\texttt{PR}^{\texttt{S/T}}$ and $\texttt{PP}^{\texttt{T}}$ contain expectations and practices. *PCM* matches expectations in $\texttt{PR}^{\texttt{S/T}}$ to practices in $\texttt{PP}^{\texttt{T}}$ and expectations in $\texttt{PP}^{\texttt{T}}$ to practices in $\texttt{PR}^{\texttt{S/T}}$. *PCM* deals with the following three cases:

Case (a) *PCM* matches a $\texttt{PR}^{\texttt{S/T}}$ assertion $A(R_i, rs)$ where rs is an input or operation usage, to an assertion $A'(R_i^{'}, rs')$ in $\texttt{PP}^{\texttt{T}}$. In this case, $A(R_i, rs)$ is a S's expectation and $A'(R_i^{'}, rs')$ is a $\texttt{PP}^{\texttt{T}}$ practice. If $A' \sqsubseteq A$ then A' and A are matched.

Case (b) *PCM* matches a $\texttt{PR}^{\texttt{S/T}}$ assertion $A(R_i, rs^E)$ where rs^E is an output, to an assertion $A'(R_i^{'}, rs'^P)$ in $\texttt{PP}^{\texttt{T}}$. In this case, $A(R_i, rs^E)$ is a S's expectation and $A'(R_i^{'}, rs'^P)$ is a $\texttt{PP}^{\texttt{T}}$ practice. If $A' \sqsubseteq A$ then A' and A are matched.

Case (c) *PCM* matches a $\texttt{PR}^{\texttt{S/T}}$ assertion $A(R_i, rs^P)$ where rs^P is an output, to an assertion $A'(R_i^{'}, rs'^E)$ in $\texttt{PP}^{\texttt{T}}$. In this case, $A(R_i, rs^P)$ is a S's expectation and $A'(R_i^{'}, rs'^E)$ is a $\texttt{PP}^{\texttt{T}}$ practice. If $A' \sqsubseteq A$ then A' and A are matched.

Two options are possible while matching $\texttt{PR}^{\texttt{S/T}}$ and $\texttt{PP}^{\texttt{T}}$. The first option is to require full matching. This is not flexible since some DaaS consumers may be willing to use a DaaS producer even if certain of their privacy constraints are not satisfied. For that purpose, we present a *cost model*-based solution to enable *partial matching*. The cost model combines the notions of *privacy matching degree* and *threshold*. Due to the large number and heterogeneity of DaaS services, it is not always possible to find policy $\texttt{PP}^{\texttt{T}}$ that fully matches a S's requirement $\texttt{PR}^{\texttt{S/T}}$. The *privacy matching degree* gives an estimate about the ratio of $\texttt{PR}^{\texttt{S/T}}$ assertions that are matched to $\texttt{PP}^{\texttt{T}}$ assertions. We refer to $\mathbf{m} \subset \texttt{PR}^{\texttt{S/T}}$ as the set of all such $\texttt{PR}^{\texttt{S/T}}$ assertions. The degree is obtained by adding the weights of all assertions in $\mathbf{m}$: $\text{Degree}(\texttt{PR}^{\texttt{S/T}}, \texttt{PP}^{\texttt{T}}) = \sum w_j$ for all assertions $(A_j(R_i, rs_k), w_j, M_j) \in \mathbf{m}$. The privacy matching *threshold* τ gives the minimum value allowed for a matching degree. The value of τ is given by the client and gives an estimate of how much privacy the consumer is willing to sacrifice. As mentioned in 3.5, we give consumer the possibility to control their "core" privacy requirements by associating a mandatory attribute M_j to each assertion $(A_j(R_i, rs_k), w_j, M_j)$ in $\texttt{PR}^{\texttt{S/T}}$.

5 Privacy-Aware DaaS Composition

We aims at extending the composition approach described in [2] to deal with privacy preserving according three step. First, a functional selection of DaaS is performed. taking as input a user query. Second, the privacy requirements and policy attached to service are fetched, thanks to the annotation approach developed in [14]. Third, the compatibility of privacy requirements and policies of the services with respect to the composition is checked. In this section we give details about these steps.

5.1 Fetching DaaS Annotations

The model developed above provides a formal background to specify privacy requirements $\texttt{PR}^{\texttt{S/T}}$ and policy $\texttt{PP}^{\texttt{S}}$ of service `S`. To make these privacy capabilities concretely available on service, we link them to services via an annotation of the service. Our previous work in [14] provided a complete description about the privacy annotation extensibility. We remind how we annotate the major description formats for DaaS (WSDL and REST annotations) according to the aforementioned privacy model. Indeed, the specifications of WSDL allows for the addition of new XML elements and attributes in certain locations inside a WSDL file. We exploit these extensibility elements to associate the services operations, interface inputs and outputs with their corresponding capability files. Specifically, for each interface, operation, input and output elements, we define a new child element called "`privacy-capability`" to hook assertions of $\texttt{PR}^{\texttt{S/T}}$ and assertions of $\texttt{PR}^{\texttt{S}}$ with to `S` descriptions. 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 are utilized to annotate 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. During the composition, the privacy requirement description file of component service is compared to this describing the privacy policy of service within composition as explaining in the above subsection.

5.2 Checking Privacy within Composition

We aims at extending the previous composition approach to deal with privacy preserving. Let us consider services in Table 1 and the following epidemiologist's query Q "What are Ages, Genders, address, DNA, salaries of patients infected with *H1N1*; and what are the global weather conditions of the area where these patients reside?". The mediator is considered as a trusted entity. It manage the composition and handles all the interactions among services. It answer Q by composing the relevant services as follows: Firstly, the invocation of $S_{1.1}$ with *H1N1*, then for each obtained patient, $S_{4.1}$ is invoked to obtain their `DNA`, $S_{2.2}$ and $S_{3.1}$ to obtain `date_of_birth`, `zip_code` and `salary` of obtained patients. Finally, $S_{5.1}$ with patients'`zip_code` to get information about

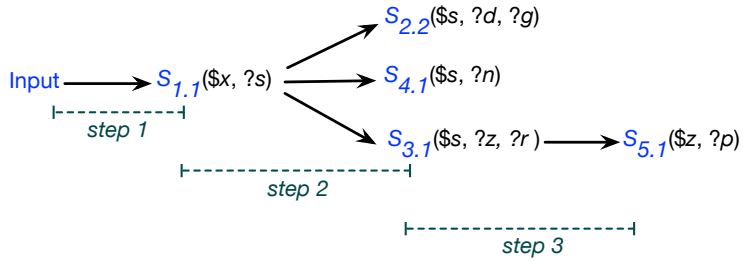

Fig. 2. Dependency Graph of query Q

the `weather_conditions`. Selected services need to be executed in a *particular order* in the composition plan depending on their inputs and outputs. To construct the composition plan the algorithm establishes a dependency graph *DG* (Figure 2) in which the nodes correspond to services and the edges correspond to dependency constraints between component services. If a service S_j need an input x that can be provided from an output y of S_i then S_j must be preceded by S_i in the composition plan; we say that there is a *dependency* between S_i and S_j (or S_j depends on S_i). Consequently, the mediator recognizes that services $S_{2.2}$, $S_{3.1}$, $S_{4.1}$ depend on $S_{1.1}$ since they have a same input y (i.e. `SSN`) which is provided as an output of $S_{1.1}$ and $S_{5.1}$ depends on $S_{3.1}$.

In order to take privacy into account, if S_j depends on S_i, then S_j is showed as a consumer to some data provided by S_i and this latter is showed then as a producer from the mediator point of view. In other words, the mediator considers the privacy requirements $\texttt{PR}^{S_i/\texttt{T}}$ for service S_i (since $\texttt{PR}^{S_i/\texttt{T}}$ specifies S_i' conditions on the usage of its concerning data) and privacy policy $\texttt{PP}^{S_j}$ for service S_j (since $\texttt{PP}^{S_j}$ specifies S_j' usage on the collected data) and checks the compatibility of $\texttt{PR}^{S_i/\texttt{T}}$ and $\texttt{PP}^{S_j}$ by using the privacy compatibility matching algorithm *PCM* (Section 4.2) within services order in *DG*.

For instance, let us consider *DG* in Figure 2. The mediator identifies firstly, from *DG*, services type (i.e., consumer services, and producer services) and resources related to each dependency. The parameter s is an input parameter for the services $S_{2.2}$, $S_{3.1}$ and $S_{4.1}$ while it is an output parameter for $S_{1.1}$ and therefore $S_{2.2}$, $S_{3.1}$ and $S_{4.1}$ depend on $S_{1.1}$. Note that input parameters begins with "$" and output parameters by "?". Similarly, the parameter z is an input parameter for $S_{5.1}$ and an output parameter for $S_{3.1}$, therefore $S_{5.1}$ depends on $S_{3.1}$. Consequently, mediator considers $S_{2.2}$, and $S_{4.1}$ as consumers services, while $S_{1.1}$ is considered once as a consumer (since it input is provided by the *input*) once as a producer (since it provides output for others services). The same reasoning is observed for $S_{3.1}$. In *step 1* the producer is the *input* (i.e., the user query), consumer is $S_{1.1}$ and the private resource rs ="`Patient Disease`". The mediator checks the compatibility of $\texttt{PR}^{input/\texttt{T}}$ and $\texttt{PP}^{S_{1.1}}$. In *step 2* the producer is $S_{1.1}$ and consumers are $S_{2.2}$, $S_{3.1}$, $S_{4.1}$ and the private resource rs ="`SSN`". The mediator checks the compatibility of $\texttt{PR}^{S_{1.1}/\texttt{T}}$ and $\texttt{PP}^{S_{2.2}}$, $\texttt{PR}^{S_{1.1}/\texttt{T}}$ and $\texttt{PP}^{S_{3.1}}$, $\texttt{PR}^{S_{1.1}/\texttt{T}}$ and $\texttt{PP}^{S_{4.1}}$, In *step 3* $S_{3.1}$ is now the producer for $S_{5.1}$ and rs ="`zip_code`" and the compatibility of $\texttt{PR}^{S_{3.1}/\texttt{T}}$ and $\texttt{PP}^{S_{5.1}}$ is checked.

Table 1. A subset of PAIRSE's DaaSs

DaaS services	Semantics services Description
$S_{1.1}(\$x, ?s)$ $S_{1.2}(\$x, ?s)$	Returns patients SNN s, infected with a disease x
$S_{2.1}(\$s, ?d, ?g)$ $S_{2.2}(\$s, ?d, ?g)$	Returns date_of_birth, d, and gender, g, of a patient identified by s
$S_{3.1}(\$s, ?z, ?r)$	Returns zip_code, z, and salary, r, of a patient identified by s
$S_{4.1}(\$s, ?n)$ $S_{4.2}(\$s, ?n)$	Returns DNA, n, of a patient identified by s
$S_{5.1}(\$z, ?w)$	Returns Weather_condition, w, of a address z

6 Evaluations

To demonstrate the feasibility of our approach to privacy-preserving DaaS composition, we applied it to a real scenario drawn from the healthcare domain. In the context of the PAIRSE project [2] we were provided with access to /411/ medical Web services defined on top of /23/ different medical databases (oracle databases) storing medical information (e.g., diseases, medical tests, allergies, etc) about more than /30,000/ patients. Among these services Table1 shows the services that pertain to the query Q in our running example. All services were deployed on top of a GlassFish web server. The resources are related to a particular type of medical data (e.g., ongoing treatments, Allergies). For each service, we have randomly generated privacy requirements and privacy policy with regard to /10/ values D_i set for R_i topic = "medical recipients" (e.g., researcher, physician, nurse, etc) and different values for R_i topic = "purpose" (e.g., scientific research, academic laboratory, government, etc.). These privacy requirements and policies are used to annotate the service description files in accordance with the mechanisms presented in section 5. Our algorithms are implemented in Java and run on a Intel Core Duo 2.53 GHz and 4GB RAM running Windows 7.

Table 2. Possible compositions that answer Q without and with privacy preservation

Compositions without privacy preservation	Compositions with privacy preservation
$C_1 = \{S_{1.1}, S_{2.1}, S_{3.1}, S_{4.1}, S_{5.1}\}$	$C_3 = \{S_{1.1}, S_{2.2}, S_{3.1}, S_{4.1}, S_{5.1}\}$
$C_2 = \{S_{1.1}, S_{2.1}, S_{3.1}, S_{4.2}, S_{5.1}\}$	$C_4 = \{S_{1.1}, S_{2.2}, S_{3.1}, S_{4.2}, S_{5.1}\}$
$C_3 = \{S_{1.1}, S_{2.2}, S_{3.1}, S_{4.1}, S_{5.1}\}$	
$C_4 = \{S_{1.1}, S_{2.2}, S_{3.1}, S_{4.2}, S_{5.1}\}$	
$C_5 = \{S_{1.2}, S_{2.1}, S_{3.1}, S_{4.1}, S_{5.1}\}$	
$C_6 = \{S_{1.2}, S_{2.1}, S_{3.1}, S_{4.2}, S_{5.1}\}$	
$C_7 = \{S_{1.2}, S_{2.2}, S_{3.1}, S_{4.1}, S_{5.1}\}$	
$C_8 = \{S_{1.2}, S_{2.2}, S_{3.1}, S_{4.2}, S_{5.1}\}$	

[2] https://picoforge.int-evry.fr/cgi-bin/twiki/view/Pairse/Web/

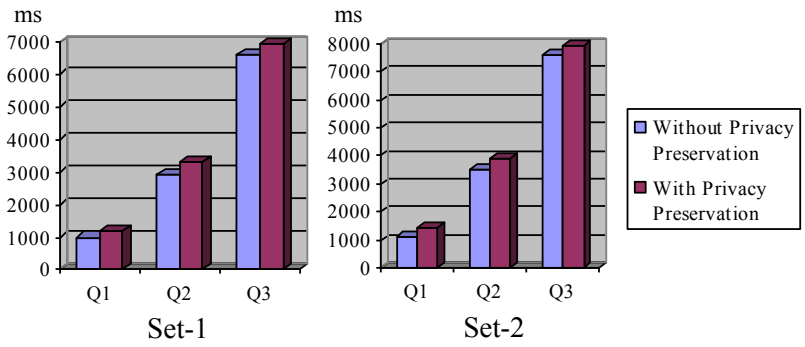

Fig. 3. The Experimental Results

Table 2 shows in the first column the different DaaSs compositions the composition would give without applying our privacy compatibility matching algorithm *PCM*. Much of these composition may violate the privacy requirements of involved services. The second column shows the possible compositions when *PCM* within composition approach (of section 5.1) is applied. These compositions do preserve the privacy requirements of involved services. We conducted a set of experiments to measure the cost incurred in privacy preservation while composing DaaS. We considered two sets of queries. The first one included queries about a given patient, each with a different size: Q_1 requests the "Personal information" of a given patient p_i, Q_2 requests the "Personal information", "Allergies" and "Ongoing Treatments" of p_i, and Q_3 requests the "Personal information", "Allergies", "Ongoing Treatments", "Cardiac Conditions" and "Biological Tests" of p_i. The second set uses the same queries Q_1, Q_2 and Q_3 but for all of patients living in Lyon. All queries were posed by the same actor (researcher) and for the same purpose (medical research). Figure 3 depicts the results obtained for the queries in sets 1 and 2,(the time shown includes both the DaaS composition construction time and the DaaS composition execution time). Set-2 (as opposed to Set-1) amplifies the cost incurred by Set-1 at the composition "*execution phase*" by a factor equals to the number of returned patients. The results for Set-1 show that privacy handling adds only a slight increase in the query rewriting time (note that the composition execution time is neglected for one patient). This is due to the fact that the number of services used to retrieve privacy requirements is limited compared to the number of services used to retrieve data (10 versus 411 in our experiments). The results for Set-2 show that the extra time needed to handle privacy in the the composition process is still relatively low if compared to the time required for answering queries without addressing privacy concerns.

7 Conclusion

In this paper, we proposed a dynamic and formal privacy model for DaaS services. The model deals with privacy at two different levels: the data (inputs and outputs) and operation levels. Services specify their privacy concerns/practices via privacy requirements and policies, respectively. Both privacy requirements

and policies refer to rules that may be added, deleted, and modified at any time. The granularity of our privacy model allows defining the widest range of policies and requirement with rich expression capabilities and flexibly manner. We introduced a cost model-based protocol for checking the compatibility of privacy requirements and policies. We have presented a preserving-privacy DaaS composition approach to resolve privacy concerns at the composition time. As future work, we plan to extend our privacy-preserving DaaS composition approach to tackle the incompatibilities between requirements and policies using a dynamic reconciliation mechanism. The reconciliation of requirements and policies will be carried out based on some negotiation protocols. We intend also to study and improve the scalability of our proposed privacy-aware composition approach.

Acknowledgment. This work has partially funded by the Region of Rhône-Alpes.

References

1. Agrawal, S., Haritsa, J.R.: A framework for high-accuracy privacy-preserving mining. In: Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, pp. 193–204. IEEE Computer Society, Washington, DC, USA (2005)
2. Barhamgi, M., Benslimane, D., Medjahed, B.: A Query Rewriting Approach for Web Service Composition. IEEE Transactions on Services Computing, TSC (January 2010)
3. Bertino, E., Yang, Y.: Privacy and ownership preserving of outsourced medical data. In: ICDE, pp. 521–532 (2005)
4. Carey, M.: Declarative data services: This is your data on soa. In: Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications, p. 4. IEEE Computer Society, Washington, DC, USA (2007)
5. Clifton, C., Kantarcioğlu, M., Doan, A., Schadow, G., Vaidya, J., Elmagarmid, A., Suciu, D.: Privacy-preserving data integration and sharing. In: DMKD 2004: Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 19–26. ACM, New York (2004)
6. Feder, T., Ganapathy, V., Garcia-Molina, H., Motwani, R., Thomas, D.: Distributing data for secure database services. Technical Report 2007-23, Stanford InfoLab (June 2007)
7. Gil, Y., Cheung, W., Ratnakar, V., kin Chan, K.: Privacy enforcement in data analysis workflows. In: Finin, T., Kagal, L., Olmedilla, D. (eds.) Proceedings of the Workshop on Privacy Enforcement and Accountability with Semantics (PEAS 2007) at ISWC/ASWC 2007, Busan, South Korea. CEUR Workshop Proceedings, vol. 320. CEUR-WS.org (November 2007)
8. Gil, Y., Fritz, C.: Reasoning about the appropriate use of private data through computational workflows. In: Intelligent Information Privacy Management, Papers from the AAAI Spring Symposium, pp. 69–74 (March 2010)
9. Hacigümüş, H., Iyer, B., Li, C., Mehrotra, S.: Executing sql over encrypted data in the database-service-provider model. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, SIGMOD 2002, pp. 216–227. ACM, New York (2002)

10. Hore, B., Mehrotra, S., Tsudik, G.: A privacy-preserving index for range queries. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDB 2004, pp. 720–731. VLDB Endowment (2004)
11. Kawamoto, J., Yoshikawa, M.: Security of social information from query analysis in daas. In: Proceedings of the 2009 EDBT/ICDT Workshops, EDBT/ICDT 2009, pp. 148–152. ACM, New York (2009)
12. Lee, Y., Werner, J., Sztipanovits, J.: Integration and verification of privacy policies using DSML's structural semantics in a SOA-based workflow environment. Journal of Korean Society for Internet Information 10(149), 09/2009 (2009)
13. Mohammed, N., Fung, B.C.M., Wang, K., Hung, P.C.K.: Privacy-preserving data mashup. In: EDBT 2009: Proceedings of the 12th International Conference on Extending Database Technology, pp. 228–239. ACM, New York (2009)
14. Mrissa, M., Tbahriti, S.-E., Truong, H.-L.: Privacy model and annotation for DaaS. In: Antonio Brogi, G.A.P., Pautasso, C. (eds.) European Conference on Web Services (ECOWS), pp. 3–10 (December 2010)
15. Ngu, A.H.H., Carlson, M.P., Sheng, Q.Z., Paik, H.-y.: Semantic-based mashup of composite applications. IEEE Trans. Serv. Comput. 3, 2–15 (2010)
16. Pang, H., Shen, J., Krishnan, R.: Privacy-preserving similarity-based text retrieval. ACM Trans. Internet Technol. 10, 4:1–4:39 (2010)
17. Truong, H.L., Dustdar, S.: On analyzing and specifying concerns for data as a service. In: Kirchberg, M., Hung, P.C.K., Carminati, B., Chi, C.-H., Kanagasabai, R., Valle, E.D., Lan, K.-C., Chen, L.-J. (eds.) APSCC, pp. 87–94. IEEE, Los Alamitos (2009)
18. Tumer, A., Dogac, A., Toroslu, I.H.: A semantic-based user privacy protection framework for web services. In: Mobasher, B., Anand, S.S. (eds.) ITWP 2003. LNCS (LNAI), vol. 3169, pp. 289–305. Springer, Heidelberg (2005)
19. W3C. The Platform for Privacy Preference Specification (2004)
20. Weise, T., Bleul, S., Comes, D., Geihs, K.: Different approaches to semantic web service composition. In: Proceedings of the 2008 Third International Conference on Internet and Web Applications and Services, pp. 90–96. IEEE Computer Society, Washington, DC, USA (2008)
21. Xu, Y., Wang, K., Zhang, B., Chen, Z.: Privacy-enhancing personalized web search. In: Proceedings of the 16th international conference on World Wide Web, WWW 2007, pp. 591–600. ACM, New York (2007)