How to Enhance Privacy within DaaS service Composition?

Salah-Eddine Tbahriti, Chirine Ghedira, Brahim Medjahed, Michael Mrissa, and Djamal Benslimane

***Abstract* — The composition of DaaS (Data-as-a-Service) services is a powerful solution for building value-added applications on top of existing ones. However, privacy concerns are still among the key challenges that keep hampering DaaS composition. Indeed services may follow different, conflicting privacy specifications with respect to the data they use and provide. In this paper, we propose an approach for privacy-aware composition of DaaS services. Our approach allows specifying privacy requirements and policies and verifying the compatibility of services involved in a composition. We propose an adaptation protocol that makes it possible to reconcile the privacy specifications of services when incompatibilities arise in a composition. We validate the applicability of our proposal through a set of experiments.**

***Index Terms*—DaaS, composition, service, adaptation, privacy.**

I. Introduction

SERVICES of type DaaS (Data-as-a-Service) have been considered during the last few years as first-class objects that can manipulate data much like database management systems do [2][17]. They also have started to be a popular medium for data publishing and sharing on the Web. Besides, modern enterprises across all spectra are moving towards service-oriented architectures by wrapping their data sources in DaaS services for more efficient data integration [2][6][17].

DaaS Composition consists in combining several DaaS services to realize Business-to-Business (B2B) interactions described according to a business process [3][4][5][1]. While initial service composition approaches have been a powerful solution for building value-added services on top of existing ones, the issue of privacy is still considered as an important topic in the field of service computing [1][16][31][33]. Indeed, despite important efforts aimed at preserving privacy [32], privacy leakage incidents on the Web continue to make the headlines. As example, in 2011, 535 breaches, involving a combined 30.4 million sensitive records have been identified [45]. Besides, the emergence of analysis tools makes it easier to analyze and synthesize huge volumes of information, hence increasing the risk of privacy violation. According to a recent report [44], the number of reported electronic health data breaches has increased by 32% from the year 2010 and electronic medical data breaches cost the industry about $6.5 billion. The concept of privacy itself generates much debate. On the one hand, some might argue that the term *privacy* can be applied only to humans and not to institutions. On the other hand, there is no unanimous agreement about which information should be considered private. For example, some individuals choose to publish personal information such as pictures, videos, and their phone number, while others keep this information private and under no circumstances want it to become public. During service composition, the issue of privacy is more challenging task. Let us illustrate some privacy challenges through the following scenario.

A. Scenario and Challenges

We consider the following epidemiologist's query Q (as a part of a global request *R*): "*What are the ages, genders, zips, DNA and salaries of patients infected with H1N1; and what are the global weather conditions of the areas where these patients reside*?" and a subset of services shown in Table 1.

TABLE I
A SUBSET OF DAAS SERVICES

DaaS services	Semantics services Description
$S_{1.1}(\$x, ?s)$ $S_{1.2}(\$x, ?s)$	Returns "*SSN"* of patient infected with a disease= "*x*"
$S_{2.1}(\$s, ?d, ?g)$ $S_{2.2}(\$s, ?d, ?g)$	Returns *d* ="*DoB*", and *g* ="*gender*" of patient identified by *s* ="*SSN"*
$S_{3.1}(\$s, ?z, ?p)$	Returns *z* ="*zip*", and *p* ="*salary*" of patient identified by *s* ="*SSN"*
$S_{4.1}(\$s, ?n)$ $S_{4.2}(\$s, ?n)$	Returns *n* ="*DNA*" of patient identified by *s* ="*SSN"*
$S_{5.1}(\$z, ?w)$	Returns *w* = "*Weather-condition*" of address *z* ="*zip*"

We have proposed in [5] a mediator-based approach to compose services (based on a query-rewriting algorithm) and answer this kind of queries. In this approach, the mediator selects, combines and orchestrates (i.e., gets output data from a service and uses it as input data to call another service) services to answer queries. It also carries out all the interactions between composed services (i.e., relays exchanged data among interconnected services in the composition). The result of the composition process is a *composition plan*, *CP* (depicted in Fig. 1), which consists of a

• Salah-Eddine Tbahriti, Michael Mrissa, and Djamal Benslimane are members of the LIRIS lab., UMR5205 CNRS, Université de Lyon, 69622, Villeurbanne, FRANCE
E-mail: firstname.lastname@liris.cnrs.fr
• Chirine Ghedira is member with IAE lab., Université Jean-Moulin Lyon 3, 69355 Lyon cedex 08, FRANCE.
E-mail: chirine.ghedira-guegan@univ-lyon3.fr
• Brahim Medjahed is member with the Department of Computer and Information Science, University of Michigan-Dearborn, 4901 Evergreen Road, Dearborn 48128 USA
E-mail: Brahim@umd.umich.edu

set of services that must be executed in a particular order depending on their access patterns (i.e., the connections between their input and output parameters). Input parameters are identified with a first "$" character and output parameters with a "?". Hence, service *S*($*a*, ?*b*) requires an input value *a* and provides an output value *b*. Then, Q can be answered as follows: First, $S_{1.1}$ is invoked with *H1N1* as input value, then for each obtained *SSN*, $S_{4.1}$, $S_{2.2}$ and $S_{3.1}$ are invoked to obtain their *DNA*, *DoB (i.e. date-of-birth)*, *zip* and *salary*. Finally, $S_{5.1}$ is invoked with the patients' *zip* to get information about the *weather-conditions* (note that other solution *CP* can be found with the services of Table I).

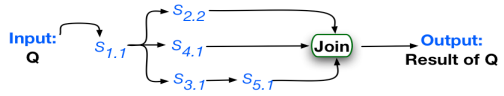

Fig. 1. Composition plan of Q.

In fact, services in *CP* may have conflicting privacy concerns with respect to their exchanged data. Some services may require some input data that other participating services cannot disclose because of their privacy specifications. For instance, let us assume that $S_{1.1}$ discloses its data (i.e., *SSN*) to a third-party service for use with a "*limited time*" restriction. $S_{3.1}$ meanwhile attests that it keeps collected data (i.e., *SSN*) for an "*unlimited time*". $S_{1.1}$ and $S_{3.1}$ are incompatible in terms of privacy with respect to *SSN*. $S_{1.1}$ (which provides *SSN)* judges that a long retention of *SSN* by a third-party is a risk for privacy, while $S_{3.1}$ would use that data as long as possible to perform several tasks that are not considered as a privacy risk. Such a conflict invalidates the *CP* of Fig. 1 in terms of privacy. Then, it becomes important on the one hand to extend service descriptions with privacy specifications, and on the other hand to insure the privacy compatibility of services selected for a composition.

B. Summary of Contributions

The previous scenario calls for a solution that must be expressive enough to capture the different needs for privacy concerns of services as well as simple and coherent with our previous service composition algorithm [5]. Since composing services is already a complex task, any target solution should involve minimal processing costs. Existing approaches based on secure multi-party computation [39] are usually characterized by their high computation time and complexity, which makes them impractical for database operations working over a large number of elements [34]. Data privacy through access control is among the classical goals of data management with countless proposals, e.g., [35]. However, our system is designed to be open, which means that the mediator (in charge of composing services and answering queries) does not have any knowledge about the requester of data. Such a circumstance makes traditional access control mechanisms ineffective, as they are mainly based on preliminary authentication of the requester.

In this paper, we focus on the privacy issue from the point of view of data usage and expectation during the design phase of DaaS composition. We build our contribution around:

Formal Model for Privacy Specification: to capture and reason about privacy concerns from a service perspective. Our proposed model allows each service *S* to define *Privacy Policies PP* (specifying how *S* manages collected data) and *Privacy Requirements PR* (specifying how *S* expects consumers to manage the data it provides). Our privacy model is defined with both expressiveness and simplicity in mind.

Privacy Compatibility-aware Composition: detecting incompatibilities between the *PR* and *PP* of services involved in a composition is a core concept of our approach. Our matching algorithm is based on the notion of privacy subsumption and on a cost model. Then, we extend our service composition approach to take into account the privacy specifications and compatibility of services.

Privacy-aware Adaptation: our third contribution is devoted to resolve detected incompatibilities by allowing services to define adaptation sets in order to obtain valid composition plans and enhance the efficiency of our system of composition. We introduce an adaptation protocol to automatically reconcile the adaptation sets in order to make the *PR* and *PP* of conflicting services compatible. The adaptation of *PR* and *PP* of service is decided by service reputations, individual' requirements subsumption by *PR* of service and a cost function. We also devise protocols to speed-up the adaptation process.

C. Paper Organization

Our paper is structured as follows. We overview the basic definitions for modeling and composing DaaS services in Section 2. Then, we describe our privacy model in Section 3. We show how our DaaS composition approach is extended within privacy compatibility in Section 4. We introduce our adaptation approach in Section 5 and detail how privacy compatibility in the composition is reached with our adaptation protocols. We present our experiments in Section 6 and discuss related work in Section 7. We discuss obtained results and future work in Section 8.

II. Background: The PAIRSE Project

The approach presented in this paper is implemented as a part of the PAIRSE project[1], which deals with privacy issues in P2P data sharing environments in the area of epidemiological research [29]. To support the decision process, scientists consider multiple data sources related to patients' data. Data are provided via DaaS services, which are set in an unstructured and unstable P2P network. In this paper, we consider DaaS services that only provide data. DaaS services are modeled as RDF views over domain ontologies to capture the semantic relationships between their input and output parameters. Fig. 2 summarizes the architecture of our project.

[1] This research project is supported by the French National Research Agency under grant number ANR-09-SEGI-008. URL https://picoforge.int-evry.fr/cgi-bin/twiki/view/Pairse/Web/

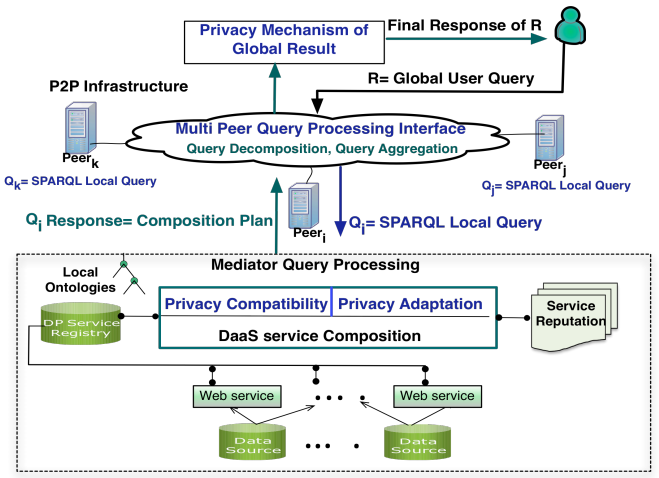

Fig. 2. PAIRSE global architecture.

The *Multi-Peer Query Processing Interface* component is in charge of answering the global user query. The latter has to be split into local queries. The component determines which peer is able to solve a local query. Each local query is expressed in SPARQL, the *de facto* query language for RDF [7]. Each peer includes a *Mediator Query Processing* component. Such a mediator selects the services that can be combined to answer the local query using a *DaaS Composition* component. Then, it processes the interactions to be performed between the composed services and generates a set of composition plans (*CP*), where each *CP* is a set of services that can answer the related local query. Once a *CP* is found, the *Privacy compatibility* component will check if all services in *CP* are compatible with respect to their privacy specifications. Subsequently, the different *CP* related to local queries are aggregated and executed to calculate an entire result into a final *table* that will be released to user. A privacy-preserving mechanism (embedded into *Privacy Mechanism of Global Result* component) is applied to this table and aims at forbidding all misuses of the data privacy. However, this last step is not detailed in this paper.

III. Formal Model for Privacy Specification

In this section, we build on our recent proposals [9][10] for privacy specification to enhance our privacy model with adaptation features for service composition. This model allows a S[2] provider to define on the one hand a set of privacy requirements (noted as $\boldsymbol{PR^S}$) specifying a set of privacy expectations that a third-party service must meet to consume S's provided data, and on other hand, a set of privacy policies (noted as $\boldsymbol{PP^S}$) specifying the set of privacy practices applicable on any data that S collects.

A. Privacy Specification Model

In order to define an expressive model of privacy for Web service, it is necessary first to examine the nature of data and to formally describe what we mean by privacy, so that we can argue that we protect such private data. In our case, the term of privacy relates to the right of an entity to determine why, for whom, and for how long some information should be released. Due to the privacy subjectivity, each service has to identify which data are considered as *private* (noted as *rs*). If S provides some private data *rs* then a set of privacy requirements applies to *rs*, and if S collects some private data *rs* then a set of privacy policies also applies to *rs*. The specification of these sets is based on privacy rules.

1) Privacy Rule

A privacy rule R_i is defined by a tuple (T_i, D_i, G_i) where T_i is the topic of R_i giving the privacy facet. For instance, the topic can describe[3]: *purpose, recipient* or *retention*. *Purpose* topic states the intent for which a given private data *rs* (collected or provided by S) will be used; the *recipient* topic mentions if and to whom *rs* can be revealed; the *retention* topic specifies whether and until when *rs* is stored by a third-party service. Then, for each topic T_i a set D_i defines the value domain of the topic. The definition of D_i is based on an ontology domain. For example, we consider the privacy rule R_1 which corresponds to the topic T_1 ="*recipient*" and the domain D_1 = {"*public*", "*government*", "*private-lab*", "*research-lab*", "*hospital*", "*university*"}. We define G_i ={"*total*", "*partial*"} as a granularity indicator, which states whether or not the data in *rs*, on which R_i applies, represent the totality of the service input or output. For instance, if the output of S contains n attributes and *rs* is composed of n' attributes where $|n'| < |n|$ then G_i="*partial*". In this paper, for the sake of simplicity we only consider the case where G_i="*total*"[4]. The definition of privacy rules, called *Rule Set* ($\mathcal{RS}$), is described independently of any private data and maintained by the administrators of the PAIRSE system.

2) Privacy Assertion

The application of a rule $R_i = (T_i, D_i, G_i)$ on private data *rs* is a *privacy assertion* noted as $A(R_i, rs) = pf$. S specifies its privacy concerns for *rs* through $A(R_i, rs)$ with a propositional formula $pf = (v_{ip} \wedge \ldots \wedge v_{iq})$ where $v_{ip},\ldots,v_{iq} \in D_i$. For example, we consider *rs*=*DoB* and R_1 which corresponds to the topic T_1 ="*recipient*" and the domain D_1. A privacy assertion on *rs*="*DoB*" through R_1, which states that *rs* will be shared with *government* agencies and *research-lab*, is noted as $A(R_1, DoB)$="*government* ∧ *research-lab*".

3) Privacy Requirements PR^S

A service S providing some private data *rs* as its output specifies a set of privacy requirements, denoted as PR^S, in terms of usage expectations that a third-party service must meet to consume *rs*. Initially, the S provider has to select and identify the set (noted $\mathcal{P}_{out}$) of its own private data. Secondly, for each element $rs \in \mathcal{P}_{out}$ S selects the privacy rules R_i from $\mathcal{RS}$ and instantiates them with an assertion: $A_j(R_i, rs_k) = pf$. In addition, the provider of S may give to each assertion A_j a

[2] In the rest of this paper, the symbol S refers to "DaaS service"

[3] The PAIRSE administrator with respect to the privacy laws in Europe and United-States can define other kinds of topic.

[4] Our model fully supports partial granularity without any modifications.

logical value *Neg*={"*T*", "*F*"}, which indicates if A_j could be adapted or not (cf. Section 5). Hence,
$PR^S = \{(A_j(R_i, rs_k)=pf, Neg), j \leq |PR^S|, i \leq |\mathcal{RS}|, k \leq |\mathcal{P}_{out}|, rs_k \in \mathcal{P}_{out}, R_i \in \mathcal{RS}, Neg=T$ iff A_j can be adapted$\}$.

4) Privacy Policy PP^S

A service *S* requesting some private data *rs'* (where $rs' \neq rs$ referring output data of *S*) as input specifies its privacy policy, noted PP^S, stating how *S* is going to use *rs'*. We define the set $\mathcal{P}_{in}$ of private data that is collected. For each $rs' \in \mathcal{P}_{in}$, *S* specifies assertions through the instantiation of R_i, and indicates for each assertion A_j a logical value *Neg*, which indicates whether or not A_j can be adapted. Then, we define $PP^S = \{(A_j(R_i, rs'_k)=pf, Neg), j \leq |PP^S|, i \leq |\mathcal{RS}|, k \leq |\mathcal{P}_{in}|, rs'_k \in \mathcal{P}_{in}, R_i \in \mathcal{RS}, Neg = T$ iff A_j can be adapted$\}$.

This distinction between provided and collected private data is crucial to describe the privacy requirements and policy of DaaS services. The content of and the assertions that apply to $\mathcal{P}_{out}$ and $\mathcal{P}_{in}$ vary from a service to another. Let us illustrate our definition with a concrete example:

Example 1: We consider two rules defined in $\mathcal{RS}$:
$R_1 = (T_1, D_1, G_1)$ where
- T_1 = "*recipient*",
- D_1 = {"*public*", "*government*", "*private-lab*", "*research-lab*", "*hospital*", "*university*"},
- G_1 = "*total*".

$R_3 = (T_3, D_3, G_3)$ where
- T_3 = "*retention*",
- D_3 = [0, 1,.., *Unlimited*] (defining retention in days),
- G_3 = "*total*".

$S_{1.1}$ and $S_{3.1}$ of Table I specify their privacy as follows. $S_{1.1}$ considers $\mathcal{P}_{out}$={*SSN*} and $S_{3.1}$ considers $\mathcal{P}_{in}$={SSN}. $S_{1.1}$, $S_{3.1}$ defines its *PR* respectively *PP* as:
- $PR^{S1.1}$ = {(A_1 (R_1, *SSN*) = "*hospital*", *Neg* =*T*); (A_3 (R_3, *SSN*) = "*10*", *Neg* = *T*)}.
- $PP^{S3.1}$ = {($A_{1'}$(R_1, *SSN*) = "*research-lab*", *Neg* =*T*); ($A_{3'}$(R_3, *SSN*) = "*100*", *Neg* = *T*)}.

$PR^{S1.1}$ states that $S_{1.1}$ provides *SSN* only to the "*hospital*" recipient ($A_1.R_1$), *and* with a right of data usage limited to *10 days* ($A_1.R_3$). $PP^{S3.1}$ states that $S_{3.1}$ shares collected *SSN* with any users identified as "*research-lab*" ($A_{1'}$, R_1), *and* that it keeps them for *100 days* ($A_{3'}$, R_3). We note that $S_{3.1}$ considers $\mathcal{P}_{out}$={*zip*, *salary*} and for each *rs* it will specify its corresponding *PR*. ♦

B. Privacy Annotation for DaaS

The *WSDL* standard gains considerable momentum as the language for Web service description. However, WSDL provides no support for privacy description of services. Existing standards such as WS-Security [43] focus on "access control"-oriented vision of privacy and does not offer the possibility for describing requirements and policies as with our model. Moreover, policy adaptation is not supported. In order to describe the privacy concerns of Web services, we extend WSDL with privacy references [8]. We exploit its extensibility elements to associate service operations, inputs and outputs with their corresponding PR^S and PP^S. More precisely, we extend the *operation*, *input* and *output* elements with a "*privacy-reference*" attribute that contains a link to a privacy file, thus keeping a clear separation between the functional description of the service (WSDL) and its privacy concerns (PR^S and PP^S described in the privacy XML file).

IV. Privacy Compatibility within Service Composition

There are two different aspects to be considered when dealing with the privacy issue in the context of service composition. The first aspect is related to describing Web service concerns with respect to privacy. The proposed model in Section III is devoted to that. The second aspect relates to evaluating how services can work together in the composition. As a result to a local query *Q*, the mediator returns a set of composition plans $\mathcal{CP} = \{CP_1,\ldots,CP_n\}$. Each $CP_l \in \mathcal{CP}$ (where $1 \leq l \leq n$) answers *Q*. In order to validate CP_l in terms of privacy, we check the compatibility between the sets of *PR* and *PP* of concerned services with respect to their order in CP_l. In this section, we explain how privacy compatibility is verified within composition.

A. Dependency Graph

The algorithm presented in [5] construct $CP_l \in \mathcal{CP}$ by building a *dependency graph* DG_l as a directed acyclic graph in which nodes represent services and edges correspond to functional dependencies between services. The execution order of services in DG_l depends on the connections between their inputs and outputs parameters as described in the dependency graph. If a service S_j has some input *$x* obtained from the output ?*y* of a service S_i then S_j must be executed after S_i in CP_l; we say that S_j depends on S_i. Fig. 3 depicts the *DG* of *CP* represented in Fig. 1 (related to *Q*) and shows the different steps that determine the execution order of CP_l. The main issues discussed in the following are: how to make sure that the privacy specifications of services in CP_l are compatible? How to deal with an eventual incompatibility?

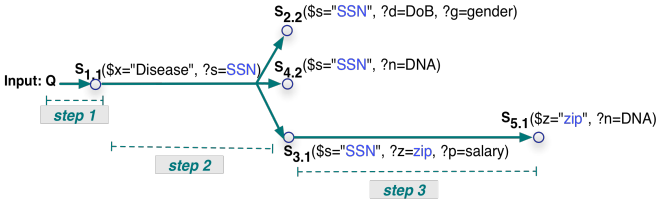

Fig 3. *DG* related to the *CP* of Q *of the motivating example*

1) Privacy Subsumption

Defining an assertion $A(R_i, rs)=pf$ involves assigning value(s) from D_i to T_i (of R_i). Let us consider D_1={"*public*", "*government*", "*private-lab*", "*research-lab*", "*hospital*", "*university*"} which represents a domain values for T_1 ="*recipient*" (related to R_1). In terms of privacy, let us assume that the value *public* is more general than any other values in D_1, then, any *rs'* recipient declared as *public* (i.e., shared with any entity) also includes *government*, *research-lab*, etc. with respect to privacy. In order to capture privacy relationships among domain values, we introduce the notion of privacy subsumption (noted $\sqsubseteq_p$). For each domain D_i, a corresponding

matrix $\mathcal{M}_p$: $D_i \times D_i$ is defined, by PAIRSE administrators upon a common ontology, to compute the privacy subsumption of all values in D_i. For instance, the following subsumptions can be stated: *government* $\sqsubseteq_p$ *public*; *research-lab* $\sqsubseteq_p$ *public*; *university* $\sqsubseteq_p$ *public*. Note that privacy subsumption is transitive like the typical "is-a" relationship. We use $\sqsubseteq_p$* to refer to the transitive closure of $\sqsubseteq_p$. In what follows, we explain how the PAIRSE mediator checks privacy compatibility of services connected in a *DG*.

B. Checking the Privacy Compatibility

Checking the privacy-compatibility between *PR* and *PP* of 2 services in a *CP* involves making sure that the assertions in $PP^{S'}$ are respecting the assertions in PR^{S} for all the data flows described in the *DG* of the *CP*.

1) Privacy Compatibility Matching Algorithm

In order to check the compatibility between assertions of two services *S* and *S'* respectively, we introduce a Privacy Compatibility Matching algorithm (*PCM*).

Algorithm: *PCM*

input : $\mathtt{PR}^{S}$=$\{(A_j(R_i, rs_k))$, $j \leqslant |\mathtt{PR}^{S}|$, $i \leqslant |RS|$, $k \leqslant |P_c|$, $rs_k \in P_c$, $R_i \in RS\}$ (assertion of privacy requirements)
input : $\mathtt{PP}^{S'}$ =$\{(A_{j'}(R_i, rs'_k))$, $j' \leqslant |\mathtt{PP}^{S'}|$, $i \leqslant |RS|$, $k \leqslant |P_p|$, $rs'_k \in P_p$, $R_i \in RS\}$ (assertion of privacy policy)
output: InC (The set of incompatible assertion couple);

1 **foreach** $rs_k = rs'_k$ **do**
2 **for** $i = 1$, $i \leqslant |RS|$ **do**
3 **for** $j = 1$, $j \leqslant |PR^S|$ **do**
4 **for** $j' = 1$, $j' \leqslant |PP^{S'}|$ **do**
5 **if** $(A_{j'}(R_i, rs'_k) \sqsubseteq (A_j(R_i, rs_k)$ **then**
6 $A_j(R_i, rs_k)$ is compatible with $A_{j'}(R_i, rs'_k)$
7 **else** InC $\leftarrow (A_j(R_i, rs_k)$, $A_{j'}(R_i, rs'_k))$

The semantics of *PCM* is described as follows: For each assertion $A \in PR^S$ (*A* is defined on *rs*) and $A' \in PP^{S'}$ (*A'* is defined on *rs'*) to be compatible, *A* must subsume *A'* in terms of privacy (noted $A' \sqsubseteq_p A$). Privacy subsumption is reached when *A* and *A'* are specified on the same data (i.e., *rs*=*rs'*), with the same privacy rule and at the same granularity (G_i=$G_{i'}$), (lines 1-4 of the algorithm *PCM*) where $R_{i'} = (T_{i'}, D_{i'}, G_{i'})$. Besides, with respect to the matrix $\mathcal{M}_p$ defined above the expectation of *S* as stated by *pf* should subsume the practice of *S'* as given by *pf'*. In other words, *pf'* should be true each time where *pf* is true. For instance, if *pf* ="*government* ∧ *research-lab*" and *pf'* ="*government*", then *pf* ⇒ *pf'* (where ⇒ is the symbol for implication in propositional calculus). Hence, *A* subsumes *A'* (noted $A' \sqsubseteq_p A$) (lines 5-6 of the *PCM* algorithm). Although some literals used in *pf* are syntactically different from the ones used in *pf'*, they may be semantically related via privacy subsumption relationships. For instance, let us assume that *pf* ="*public* ∧ *research-lab*" and *pf'* ="*university*". Since *university* $\sqsubseteq_p$ *public*, we can state that *public* ⇒ *university*. In this case, *PCM* recognizes that *pf* ⇒ *pf'* and hence $A' \sqsubseteq_p A$. *PCM* returns a set, noted *InC*, (line7) containing assertions couples (in *PR* and *PP* respectively) that are not compatible. We note that the compatibility check is *not symmetric* and that privacy subsumption is satisfied iff all assertions in PR^S *subsume* all assertions in $PP^{S'}$. If *InC*={}, then PR^S and $PP^{S'}$ are compatible.

2) Mediator Operation

From the PAIRSE mediator point of view, any service S_c which depends on S_p (with respect to the dependency order of corresponding *DG*) is showed as a *consumer* to some data provided by S_p and the latter is showed as a *producer*. For each edge in *DG*, the PAIRSE mediator extracts the dependent *rs* and assertions in PR^{Sp} (related to *rs*) of the producer service S_p (since PR^{Sp} specifies the requirements of S_p on the usage of its *rs*) and assertions in PP^{Sc} (related to *rs*) of the consumer service S_c (since PP^{Sc} specifies the usage S_c makes of the collected data) and checks the compatibility of these assertions by using the *PCM* algorithm. Then, a given CP_l is considered as privacy-compatible if the privacy compatibility is fully satisfied for all the dependencies in DG_l.

The mediator is committed through an *e-contract* [27] to only validate a CP_l iff *InC*={}. In other words, if at least one dependency in CP_l (regarding *rs*) presents incompatible assertions in *PR* and *PP*, then CP_l violates privacy and is discarded from $\mathcal{CP}$.

Example 2: Let us consider *DG* of Fig. 3. Firstly, the mediator identifies service consumers, producers and data dependencies from *DG*. The attribute *s*="*SSN*" is an input parameter for $S_{2.2}$, $S_{3.1}$ and $S_{4.1}$, and an output parameter for $S_{1.1}$. Therefore $S_{2.2}$, $S_{3.1}$ and $S_{4.1}$ depend on $S_{1.1}$ for providing *s*="*SSN*". Similarly, *z*="*zip*" 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, $S_{2.2}$ and $S_{4.1}$ are considered as consumers, while $S_{1.1}$ is considered once as a consumer and once as a producer. The same reasoning is observed for $S_{3.1}$. In step 1 of the execution order (see Fig. 3), the producer is the query *Q*, and the consumer is $S_{1.1}$. So, if we consider that $S_{1.1}$ considers defined a *PP* for *rs*="*Disease*", then the mediator checks the compatibility of PR^Q and $PP^{S1.1}$, where PR^Q is the set of privacy requirements provided with the user query regarding the "*Disease"* attribute. In step 2, the producer is now $S_{1.1}$, consumers are $S_{2.2}$, $S_{3.1}$ and $S_{4.1}$ and the dependent data is "*SSN*". Similarly, the mediator checks the compatibility of $PR^{S1.1}$ and $PP^{S2.2}$, $PR^{S1.1}$ and $PP^{S3.1}$, $PR^{S1.1}$ and $PP^{S4.1}$ (we also assume that $S_{1.1}$ defined assertions in *PR* for *rs*="*SSN*"). In step 3, $S_{3.1}$ is the producer for $S_{5.1}$ and *rs*="*zip*" and the compatibility of assertions defined on *rs*="*zip*" in $PR^{S3.1}$ and $PP^{S5.1}$ respectively is checked. For instance, we consider sets $PP^{S3.1}$ and $PR^{S3.1}$ (described in example 1) the compatibility of $PP^{S3.1}$ and $PR^{S3.1}$ at step 2 is not held according to the *PCM* algorithm, since "*hospital*" ⇒ "*research-lab*" and "10" ⇒ "100" are false according to privacy subsumption. Then *InC*={(A_1, $A_{1'}$), (A_3, $A_{3'}$)}. ♦

The mediator should discard any *CP* which is subject to privacy incompatibility from the response set $\mathcal{CP}$. However, sometimes no compatible *CPs* are found, in which case it becomes interesting to be able to adapt *a priori* incompatible *CPs* in order to reach a solution. We intend (to help scientists in achieving their epidemiological tasks) and avoid as possible such empty responses in order to improve the usefulness of the system. In the next section, we propose an adaptation mechanism to reach compatibility between services in *DG*.

V. Privacy-Aware Adaptation

In the previous section, we showed how privacy is checked within composition plans using dependency graphs and our *PCM* algorithm. In order to improve the flexibility and adaptability of our system, we propose a privacy-aware adaptation approach to reach a compatible CP_l from an incompatible one, to be detailed in the following.

A. Overview of the Adaptation approach

In Section II, we show how our mediator selects a service from several candidate services to answer a sub-part of the user query. Several approaches in literature use non-functional quality of service (QoS) properties to select services [38][40], where services provide contracts that can guarantee a certain level of QoS. Contract compliance can be evaluated via a reputation mechanism [41]. We use a similar notion to define a non-functional "*reputation*" property as a criterion to select services during composition. The mediator ranks reputation of services according to their availability for composition (cf. Fig. 2). Reputation is defined in formula (1) as the number of times that *S* has adapted its *PR/PP*, divided by the number of times *S* received PR^S/PP^S adaptation requests from the mediator. The more *S* is willing to adapt its PR^S/PP^S, the higher is its reputation. Each adaptation is weighted with a time factor (denoted as Δrt) that decreases the importance of the oldest adaptations in order to keep services active and make reputation a up-to-date indicator.

$$Reputation(S) = (\Delta rt \times Acc_{Adapt(PR/PP)})/Req_{Adapt(PR/PP)} \quad (1)$$

Consequently, service providers will be more and more conscious about their reputation [26][30]. This is the main motivation that makes service willing to adapt their *PR/PP* (i.e., assertions assigned with *Neg=T*) while preserving privacy. Our adaptation approach works as follows. The provider formulates two boundary values (detailed below) between which it generates an adaptation set. Please note that the *S*'s provider is able to take such decisions only if $PR^S \sqsubseteq_p$ (subsumes) the individuals' requirements whose data are provided by *S*. If the adaptation set is related to *PP*, *S*'s provider should refer to a cost function to take the decision of adapting *PP* (cf. Section V.B). This cost function gives provider gain estimation for keeping or adapting *PP*. Thus, in case of incompatibility, the mediator checks if the concerned services are willing to adapt their assertions in *PR/PP* in *InC* and automatically carry out a reconciliation of adaptation sets through automated protocols.

B. Service Adaptation Strategy

The adaptation set should supply *S* with the means to express other alternatives for their own PR^S/PP^S. Adaptation is cautiously operated with respect to pre-defined set of conditions without any privacy loss with respect to initial PR^S/PP^S.

1) PR Adaptation

If the provider is abundantly perceptive to participate in composition and increase its services reputation, that does not mean in any way that *PR* should be relaxed at the expense of privacy. For that aim, the *PR* adaptation strategy of service *S* is defined according to the individuals' own requirements that are given to *S* when individuals' data are integrated. The requirements that come from individuals are always respected, since the service *S* takes advantage of the fact that its own requirements must always be more restrictive than what is effectively required by individuals (i.e., $PR^S \sqsubseteq_p PR$ of individuals whose data are provided by *S*). Individuals can unequally value the assertions. For instance, some individual's requirements about *SSN* may be stronger than his/her requirements for "*zip*". Besides, some individual may consider an assertion more essential than another, even if both assertions are about the same *rs*. For example, an individual 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*. All these privacy features are taken into account through an e-agreement [27] between the service provider and concerned individuals. Thus, when the provider of a service *S* specifies the PR^S of the service, it selects, with respect to the individuals' requirements, the assertions A_j in PR^S that it is willing to adapt and assigns them the value *Neg=T*. Then, for each *A* with (*Neg=T*), *S* specifies an alternative *value set* of *A* called *Adaptation Set* noted T_A^S which is defined as follows:

$$T_A^S = (D^T, f_A^S) \quad (2)$$

where $D^T \subseteq D_i$ and f_A^S is the adaptation function of assertion *A* defined as, $f_A^S : D^T \rightarrow [0, 1]$. $f_A^S(v_i)$ is called the grade of v_i in T_A^S and it is a float value $\in [0, 1]$ (where $v_i \in D^T$). D^T is a finite set and f_A^S is an injective function on D^T (i.e., it does not exist two elements from D^T that have the same grade). Each element in T_A^S is indicated with its grade (denoted as $f_A^S(v_i)/v_i$). Hence, T_A^S is totally ordered. *S* exposes T_A^S to the mediator as follows:

$$T_A^S = \{f_A^S(v_L)/v_L, \ldots, f_A^S(v_i)/v_i, \ldots, f_A^S(v_U)/v_U\} \quad (3)$$

where the values v_L and v_U are respectively the lower and upper bound for f_A^S. Hence, the higher is the grade of v_i, the more adaptable is the corresponding assertion. The set T_A^S is characterized as a gradual set [12] since each element v_i is associated with a grade $f_A^S(v_i)$. T_A^S contains all the possible values that *S* will take in descending order to adapt its assertion *A* (instead of the initial value of *A*).

2) PP Adaptation

PP adaptation differs from *PR* adaptation. Indeed, when a

provider specifies its *PP*, it takes into consideration (in addition to the privacy features and their impact on its reputation) other QoS features that may help improving its performance. Studies have demonstrated how personal data, such as information captured by the index of desktop user-trace, local analyses, etc. can be used in order to enhance QoS, for example with personalization functionalities and consequently greatly improve the relevance of service behavior [25]. However, the usage and storage of such information may conflict with the *PR* of other services. Obviously, the foremost challenge for *S* provider is to take the best decision between keeping its PP^S unchanged and adapting them. Inevitably, a cost function on privacy-efficiency of trade-offs is needed in order to measure the gain earned in terms of reputation value by adapting PP^S and the gain earned by keeping PP^S unchanged. We give service providers the ability to use the following cost function for evaluating the best choice by using the two measures $U_{Rep}^{PP^S}$ and $U_{Pri}^{PP^S}$ as follows:

$$C_S^{Ad\text{-}Ke} = \psi(U_{Rep}^{PP^S}, U_{Pri}^{PP^S}) \quad (4)$$

where the superscript parameter *Ad-Ke* of $C_S^{Ad\text{-}Ke}$ refers to the *Adaptation-Keeping* of the PP^S ; $U_{Rep}^{PP^S}$ is a utility function based on formula (1) that measures the reputation gain earned by adapting PP^S. $U_{Pri}^{PP^S}$ is a utility function measuring the reputation gain when PP^S is kept unchanged. If $U_{Rep}^{PP^S} \geq U_{Pri}^{PP^S}$, $C_S^{Ad\text{-}Ke}$ returns an estimation set, denoted as $PP^S{}_{Ad}$, of the relevant assertions that affect the overall process of PP^S adaptation. These assertions are assigned with *Neg=T* (obviously, $PP^S{}_{Ad} \subseteq PP^S$). The function $C_S^{Ad\text{-}Ke}$ is inspired from the numerous economic models proposed in [24]. Then guided by $C_S^{Ad\text{-}Ke}$, *S* generates adaptation set for the assertions with *Neg =T*. It follows the same procedure as *PR* adaptation and defines its T_A^S according to the formulas (2) and (3).

Example 3: Assume that $S_{1.1}$, $S_{3.1}$ have found it advantageous to adapt their *PR* (while *PR* $\sqsubseteq_p$ *PR of individuals* whose data are provided by $S_{1.1}$, $S_{3.1}$), resp. *PP* (according to formula (4)). The adaptation sets are defined (according to formulas (2) and (3)) as follows:

For A_1, $S_{3.1}$ defines:

$T_{A_1}^{S_{1.1}}$ = {0.6/*private-lab*, 0.5/*university*, 0.1/*government*}

For $A_{1'}$, $S_{3.1}$ defines:

$T_{A_{1'}}^{S_{3.1}}$ = {0.8/*university*, 0.6/*government*, 0.3/*hospital*}

The grade of each value in $T_{A_1}^{S_{1.1}}$, resp. in $T_{A_{1'}}^{S_{3.1}}$, is assigned according to the function $f_{A_1}^{S_{1.1}}$, resp. $f_{A_{1'}}^{S_{3.1}}$. For instance, the element "0.6/*private-lab*" in $T_{A_1}^{S_{1.1}}$ is the most easily adaptable element and illustrates the upper bound. For A_3, $S_{1.1}$ defines a function $f_{A_3}^{S_{1.1}}$ and $S_{3.1}$ defines for $A_{3'}$ a function $f_{A_{3'}}^{S_{3.1}}$. The corresponding sets $T_{A_3}^{S_{1.1}}$ and $T_{A_{3'}}^{S_{3.1}}$ are depicted in the Fig. 4.

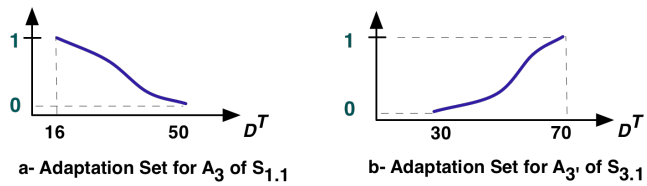

Fig 4. Adaptation Set for A_3 and $A_{3'}$

For example, the element "16" in $T_{A_3}^{S_{1.1}}$ illustrates the upper bound. Then, "1/16" it is the best value of adaptation for $S_{1.1}$ regarding retention (which is initially defined as "10"). ♦

The provider of *S* updates its adaptation sets when: (a) *S* has continually been invoked during the last Δ*time* and *S* was frequently not compatible and (b) the cost function $C_S^{Ad\text{-}Ke}$ defined in formula (4) helps defining which assertions should be adaptable. Thanks to our reconciliation protocols presented below, the mediator calculates among these sets, the couples of assertions that will be compatible. The best alternative for both consumer and producer service will be automatically adopted.

C. Reconciliation Protocols

Drawing on the previous definitions, we now introduce the protocols for *PR* and *PP* reconciliation. In our context, reconciliation has for objective to determine which adapted assertions two parties have in common. In the course of conventional reconciliation protocols, at least one of the two parties learns more of the other party's policies than what the two parties have in common. This fact is a major problem in situations where privacy is of primary concern. Our objective for privacy-aware reconciliation is to reveal nothing more about a party's adapted assertions than the adapted assertions the two parties have in common. Since it is the mediator that performs the composition, it is also in charge of the adaptation process. The first constraint is that neither the consumer service S_c nor the producer service S_p are able to know about the adaptation function of the other entity. Another important constraint is that the mediator also appoints through the *e-contract* [27] its commitment not to disclose any information related to the adaptations sets of services. It handles the adaptation protocol, which is established by the exchanges of digital credentials of services that do not have any knowledge on the adaptation sets of each other.

1) The basic protocol

Our adaptation protocol looks for the best value, noted v_r, common to the producer and consumer adaptation sets $T_{A_j}^{S_p}$ and $T_{A_{j'}}^{S_c}$ respectively defined for A_j, $A_{j'}$. The value v_r is calculated as follows:

$$Max(Min(T_{A_j}^{S_p}, T_{A_{j'}}^{S_c})) \quad (5)$$

where $Min(T_{A_j}^{S_p}, T_{A_{j'}}^{S_c})$ is a gradual set and represents the intersection of two gradual sets; $T_{A_j}^{S_p}$ and $T_{A_{j'}}^{S_c}$. The operator

Min[5] is a conjunction gradual operator. Hence, v_r is calculated as the element with the highest grade in the set $Min(T_{A_j}^{S_p}, T_{A_{j'}}^{S_c})$ and the operator *Max*[6] of formula (5) is a disjunction gradual operator. We note that there are a variety of methods available in the literature to calculate the intersection and disjunction of gradual sets [12]. We rely on the *Min* and *Max* operators since they keep most properties of classical logic.

Example 4: We consider the previous adaptation sets of example 3. According to (5), the best value v_{r1} that reconciles these sets is computed as follows.

v_{r1}=*Max*(*Min*(
{0.6/*private-lab*, 0.5/*university*, 0.1/*government*},
{0.8/*university*, 0.6/*government*, 0.3/*hospital*}))
v_{r1} = *Max* ({0.5/*university*, 0.1/*government*}) = 0.5/*university*.

The same reasoning is applied to compute v_{r2} to reconcile adaptation sets related to A_3 and $A_{3'}$. Upon successful reconciliation the consumer is granted access to the required data for the composition. The mediator requests $S_{1.1}$ and $S_{3.1}$ to adapt respectively their *PR* and *PP* with respect to assertions
A_1="university", A_3="v_{r2}" for $S_{1.1}$ and
$A_{1'}$="university", $A_{3'}$="v_{r2}" for $S_{3.1}$ where
v_{r2} =*Max*(*Min* ($T_{A_3}^{S_{1.1}}$, $T_{A_{3'}}^{S_{3.1}}$)). ♦

A given service defines an original *PR* (resp. *PP*) and may have several adapted versions of that *PR* (resp. *PP*), which result from the adaptation protocols. The mediator manages these versions and for each one of them, it indexes them with the related service and the corresponding CP_l. If *S'* does not define any adaptation set (i.e., all assertions are assigned with *Neg=F*) and its current $PP^{S'}$ is not compatible with PR^S of *S* which has a adaptation strategy, then the mediator refers to the formula (1) and looks for a candidate *S''* that is exactly equivalent to *S'* (i.e., *S"* has the same input and output like *S'*) and that is willing to adapt its $PP^{S''}$ (if the current $PP^{S''}$ are not compatible). Incompatible CP_l with the smallest *InC* are prioritized and selected prior to the adaptation process. The protocol described above is related to the case where one consumer depends on one producer in CP_l. We propose the following protocols for multiple dependencies.

2) Protocols for Multiple Dependencies

With respect to the dependency steps in *DG* of *CP*, the mediator can execute, in addition to the above protocol, one of the following protocols in order to find the best value that reconciles adaptation sets of services.

[5] The Intersection of two gradual sets A and B with membership functions μA and μB respectively is defined as the minimum of the two individual membership functions. This is called the minimum criterion μA∩B=Min(μA, μB).

[6] The Union of two gradual sets A and B with membership functions μA and μB respectively is defined as the maximum of the two individual membership functions. This is called the maximum criterion μA∪B=Man(μA, μB)

a) Protocol for 1-N Adaptation sets

In the 1-n case, during the *step m* of *DG*, the inputs of *N* consumers are connected to the output of one producer and these *N+1* services are not compatible. According to the *PCM* algorithm, InC={(A_1, $A_{1'}$), (A_1, $A_{2'}$),...,(A_1, $A_{n'}$)} (where $A_{1'}$ is an assertion of consumer S_{c1}, $A_{2'}$ is an assertion of consumer S_{c2}, and so on). In addition, if *N* consumers S_{c1} to S_{cn} are willing to respectively adapt their *PP*, then we have *N* sets, $T_{A_{1'}}^{S_{c1}}$,...,$T_{A_{n'}}^{S_{cn}}$ and v_r is calculated as follows:

$$v_r = Max(Min(T_{A_1}^{S_p}, Min(T_{A_{1'}}^{S_{c1}}, ..., T_{A_{n'}}^{S_{cn}})))$$

b) Protocol for M-1 Adaptation sets

Similarly in the m-1 case, during *step m* of *DG*, the input of one consumer is connected to the output of *M* producers and these *1+M* services are not compatible. Hence, InC={(A_1, $A_{1'}$), (A_2, $A_{1'}$), ..., (A_m, $A_{1'}$)}(where A_1 is an assertion of producer S_{p1}, A_2 is an assertion of producer S_{p2} , and so on). Then, if we consider that the *M* producers, S_{p1} to S_{pm}, have adaptation sets, then we have *M* sets, $T_{A_1}^{S_{p1}}$,...,$T_{A_m}^{S_{pm}}$ and v_r is calculated as follows:

$$v_r = Max(Min(Min(T_{A_1}^{S_{p1}}, ..., T_{A_m}^{S_{pm}}), T_{A_{1'}}^{S_c}))$$

D. Discussion

In order to analyze our privacy adaptation approach, we discuss the two following factors:

1) Adaptation Soundness

It is worth noting that the pieces of information services could deduce from the adaptation protocols is equivalent to knowing what can be deduced from $Max(Min(T_{A_j}^{S_p}, T_{A_{j'}}^{S_c}))$. While deviation from the adaptation strategy may imply privacy violations, a malicious service will not necessarily benefit from deviation in terms of maximizing its adaptation set values. This is due to the fact that the combined adaptation of the sets values (i.e., $Min(T_{A_j}^{S_p}, T_{A_{j'}}^{S_c})$) depends on the value grade of the honest service. In order to profit from the deviation, a malicious service would need to know the honest service's value grade in the adaptation set at reconciliation time and that is not possible since neither consumer S_c nor producer S_p are able to know about the adaptation strategy of the other party. Consequently, our adaptation protocols are secure at the applicative level since no information about adaptation can be disclosed outside the mediator. Moreover, these protocols are sound since the value v_r (of formula (4)), if it exists, is consistent with both adaptation sets and hence, supports the determination of all mutually compatible *PR/PP* on one round with no further adaptation.

2) Privacy Adaptation vs. Reputation Increase

The principles of privacy adaptation could be compared to anonymization approaches that aim at finding the best ratio between data protection and data utility. The cost function (given by formula (4)) is devised to give an estimation of the best choice between keeping and adapting *PPs*. In our work, services propose their own privacy requirements (*PRs*), which

must always be more restrictive in terms of privacy than what individuals require (we use the notion of privacy subsumption, which has been formally defined in our work, to verify this property). Then, the adaptation is made possible between the requirements of the service (that can be adaptable) and the requirements previously expressed by individuals (data owners). Hence, the privacy requirements that come from individuals are always respected, and the service takes advantage of the fact that its own requirements are always more restrictive than what is effectively required by individuals. Thus, adaptation, if applicable, is strictly bound between the requirements of services and those of individuals.

However, we argue that a compatible composition plan (CP_l) is not entirely protected. Several types of attack [13][32] can be performed against composition execution (i.e., a the result table of CP_l execution) in order to disclose the identity of published data. The *Privacy Mechanism of Global Result* component (cf. Fig. 2) is devoted to deal with this issue. Service providers need to be aware of this privacy-preserving mechanism that will be applied to the result table of CP_l. In this paper we focus on providing individuals with the means to express their privacy requirements so that they are respected by services. Hence, our goal is not to evaluate how much information can be inferred with respect to attacker's knowledge. Indeed, quantifying and modeling attacker is a NP-hard problem [28]. However, with respect to this challenge, the solution we deem the most appropriate is based on the work detailed in [28] that allows to efficiently model the attacker's knowledge through several dimensions. The approach allows calculating the probability for an adversary to re-identify the data contained in T_{CP} (in our case T_{CP} being the table of the compatible CP execution). The goal of an adversary is to predict whether a target individual t (contained in T_{CP}) has a target sensitive value s. In making this prediction, the attacker has access to the released candidate of T^*_{CP}, where T^*_{CP} is a result of k-anonymity on T_{CP}, as well as his own knowledge K. The objective is to calculate the function: $max_{t,s}$ *Prediction*(t has $s \mid K, T^*_{CP}$) that allows to calculate the **breach probability**, which represents the adversary's confidence in predicting the sensitive value s of the *least protected* individual t (in T^*_{CP}). The result of this function must be $< c$, where c is a threshold value defined by the mediator. The final released table to be returned to the requester must verify $max_{t,s}$ *Prediction*(t has $s \mid K, T^*_{CP}$) $< c$.

VI. Prototype and Experiments

In this section, we illustrate the viability of our approach. First, we show how we implemented our prototype for querying and composing DaaS services while ensuring privacy compatibility with our algorithms. Then, we show a set of experiments to analyze the impact of the *PCM* algorithm and adaptation on service composition.

A. Prototype Architecture

The architecture of our prototype is organized into four layers as depicted in Fig. 5. The first layer contains a set of Oracle/MySQL databases that store the medical data. The second layer includes a set of proprietary applications; each application accesses databases from the first layer. These proprietary applications are exported as DaaS services to the system. These services constitute the third layer. The WSDL files of DaaS services in the third layer are annotated with RDF views enriched with *PR* and *PP* annotations. Annotated description files are published to service registries. The upper layer includes a Graphical User Interface (GUI) and a Web Service management system (WSMS). Users access the system via the GUI and submit queries to the composition system. WSMS is composed of several modules: Interactive Query Formulator, RDF Query Rewriter, Service Locator, Composition Plan Generator, Execution Engine, and Up-Cast/Down-Cast Message Transformer.

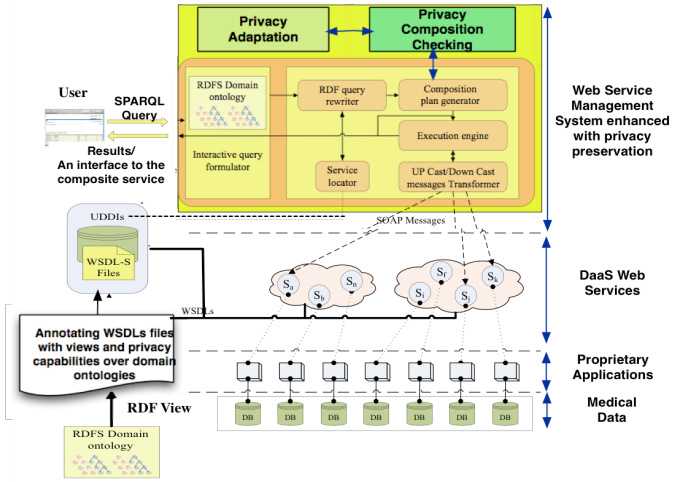

Fig 5. Architecture of the PAIRSE prototype

We used the deployment kit bundled with GWT (Google Web Toolkit) and the Apache Tomcat 6 to build and deploy our DaaS services (running on Mac OS X 10.6.8).

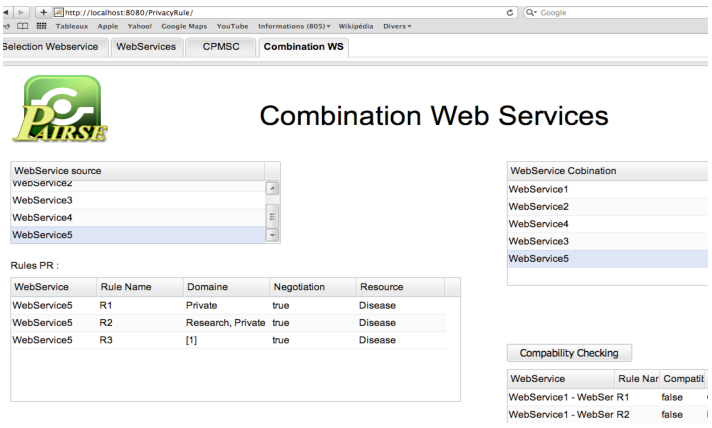

Fig 6. The PAIRSE Platform

Fig. 6 shows the *WebService, CPMSC* (*Compatibility-Privacy-Matching of service Composition*) and *Combination WS* components of the main interface implementing our prototype platform. Through the *WebService* component, service providers specify their *PP* and *PR* as well as adaptation sets. The *CPMSC* component implements the *PCM* algorithm to check assertion compatibility. If any compatible *CP* is found

the *Combination WS* component is called and performs the adaptation approach among candidate CP_l.

B. PCM Evaluation

To evaluate the impact of privacy compatibility checking on composition process, we implemented our PCM algorithm (described in Section IV.A.2) in Java and integrated it into the composition algorithm. We ran the composition system with and without compatibility checking. Each service defined it *PR* and *PP*. *Time1* in Table II is the time to compute *CP* for Q without checking compatibility (we have obtained eight different *CP*). *Time2* is the time to compute *CP* for Q with checking the privacy compatibility; only two *CP* were valid. ($\mathcal{CP}$={CP_3, CP_4}). Obviously, *PCM* does not introduce an important cost; the time needed to apply *PCM* is less than 60 milliseconds for query Q (given in our previous scenario).

TABLE II
COMPOSITION PLANS ANSWERING Q WITHOUT AND WITH PCM ENFORCING

Composition without PCM	*Time1*(ms)
$\boldsymbol{CP_1}$=($S_{1.1}$, $S_{2.1}$, $S_{3.1}$, $S_{4.1}$, $S_{5.1}$) $\boldsymbol{CP_2}$=($S_{1.1}$, $S_{2.1}$, $S_{3.1}$, $S_{4.2}$, $S_{5.1}$) $\boldsymbol{CP_3}$=($S_{1.1}$, $S_{2.2}$, $S_{3.1}$, $S_{4.1}$, $S_{5.1}$) $\boldsymbol{CP_4}$=($S_{1.1}$, $S_{2.2}$, $S_{3.1}$, $S_{4.2}$, $S_{5.1}$) $\boldsymbol{CP_5}$=($S_{1.2}$, $S_{2.1}$, $S_{3.1}$, $S_{4.1}$, $S_{5.1}$) $\boldsymbol{CP_6}$=($S_{1.2}$, $S_{2.2}$, $S_{3.1}$, $S_{4.1}$, $S_{5.1}$) $\boldsymbol{CP_7}$=($S_{1.2}$, $S_{2.1}$, $S_{3.1}$, $S_{4.2}$, $S_{5.1}$) $\boldsymbol{CP_8}$=($S_{1.2}$, $S_{2.2}$, $S_{3.1}$, $S_{4.2}$, $S_{5.1}$)	*680*
Composition with PCM	***Time2*** **(ms)**
$\boldsymbol{CP_3}$=($S_{1.1}$, $S_{2.2}$, $S_{3.1}$, $S_{4.1}$, $S_{5.1}$) $\boldsymbol{CP_4}$=($S_{1.1}$, $S_{2.2}$, $S_{3.1}$, $S_{4.2}$, $S_{5.1}$)	***738***

In order to demonstrate the feasibility of our privacy compatibility approach, we applied the prototype to a real scenario drawn from the healthcare domain. In the context of the PAIRSE project, we were provided with access to 100 medical Web services. For each service, we have randomly generated PR and PP regarding the manipulated resources (i.e., inputs and outputs). Assertions were generated randomly and stored in XML files. The computational complexity of our *PCM* algorithm is of order $O(n^2)$. The total number of assertions that must be checked among PR^S (containing *n* assertions) and $PP^{S'}$ (containing *m* assertions) that are related to the same private data and with respect to one dependency in *CP* is $n \times m$. We conducted a set of experiments to analyze the scalability of our *PCM* as the size of *PR* and *PP* increases according to the number of data and number of assertions. Then, we have generated a synthetic composition plan for which we changed the number of services (i.e., the size of CP). Fig. 7 shows the performance of *PCM* as the composition, *PR*, and *PP* size increase, for values of *k* ranging between 3 and 7 (where *k* is the number of *rs*). Each service in the composition plan had *k* sets of data, associated with 13 assertions each. In Fig. 7-(b), each service in *PC* had *k* sets of data, associated with 40 assertions each (we have generated more than 7 privacy rules). The reported durations in both parts of Fig. 7 are the durations needed to check the privacy compatibility. Thus, in Fig. 7-(a), the execution time of *PCM* slightly increases when *k* increases (105ms for *PR*=13, *PP*=13, *k*=3 and 170ms for *k*=7, 100 services are processed). In Fig. 7-(b), a small increase (execution time) is observed (150ms for *PR*=40, *PP*=40, *k*=3 and 210ms for *k*=7, and 100 services are processed). The time requirements of *PCM* are linear with the number of assertions, and we expect *PCM* to scale well on larger sets of *PR* and *PP*.

C. Adaptation Evaluation

In a second set of experiments, we simulated the performance of our adaptation approach. The adaptation sets $T_A^{S_i}$ ($1 \geq i \geq 100$) were generated randomly and each one of them is defined on D_T =[0,...,100] with respect to the "*Retention*" topic.

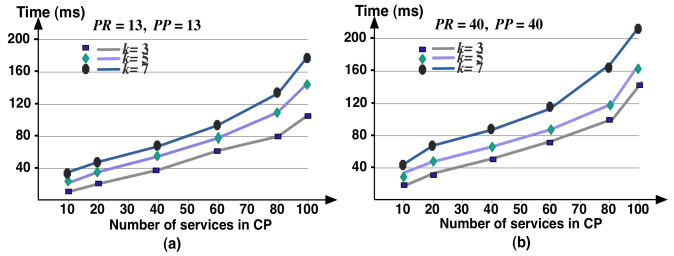

Fig 7. PCM Evaluation

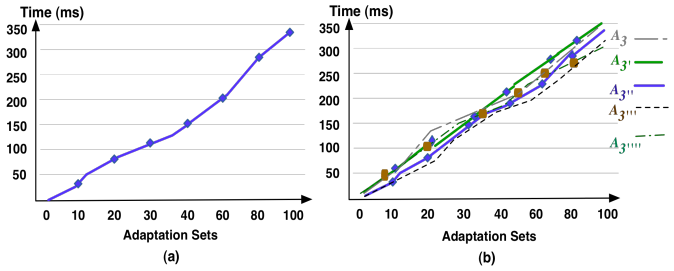

Fig 8. Adaptation Evaluation

Fig. 8 shows the obtained results. The reported durations includes time needed to compute $Max(Min(T_A^{S_1},...,T_A^{S_{100}}))$ and to adapt the related assertions of involved services. Fig. 8-(a) shows that even for a large number of adaptation sets (e.g., 100), the adaptation time remains negligible (344 milliseconds for 100 sets) compared to *Time1* needed for computing the compositions for a given query. Note that in the general case we may have *P* adaptation processes related to *P* different privacy assertions at the same level of dependency graph and in such case they could be all carried out in parallel. Fig. 8-(b) shows the performance when we execute five adaptation strategies. Time execution are close; the largest interval is between [60ms - 145ms] related to the sets of A_3 and $A_{3''''}$.

VII. RELATED WORK

In the following, we review the close work in related areas and discuss how our work leverages and advances the current state-of-the-art techniques.

A. Privacy-aware Data Modeling

A typical example of modeling privacy is the Platform for Privacy Preferences (P3P)[22]. This standard is destined to

specify privacy policies, but presents several limitations, for example it does not support adaptation, as mentioned previously in the paper. The approach in [15] assumes 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 by using a DAML-S ontology. Individuals specify their privacy preferences in different permission levels on the basis of domain specific service ontology. However, privacy preferences do not include the point of view of individuals over data usage restrictions. The approach described in [33] 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 a logic-based language. A policy is utilized in service selection and invocation. OWL-S profile is then extended with policies. Privacy constraints are not related to the published data but rather to the service. An interesting approach is described in [11]. It aims at providing an expressive form of authorization rules which define on the join path of relations and they also devise an algorithm to check if a query with given query plan tree can be authorized using the explicit authorization rules. The approach of [42] is typically based on the definition of an *access pattern* associated with each relation/view that defines how it can be accessed. It is rather devoted to deal with the optimization issue for conjunctive queries. In our work, privacy data is specified and may be related to individuals, data and service providers, and not only to the provided data, and allows defining complex restrictions on data usage (recipient, purpose, retention, and possibly other aspects) rather than simply defining access policies.

B. *Privacy-aware Service Composition*

Despite the relatively large body of work in the area of service composition, few efforts have specifically addressed the issue of privacy in service composition. In [20] a framework for enforcing data privacy in workflows is described. Privacy-preserving mechanism for data mashups is presented in [14]. Both these works aim at integrating private data from different data providers in a secure manner. The authors in [16] discuss the integration and verification of privacy policies in SOA-based workflows. Previous approaches focus on algorithms (such as *k*-anonymity) for preserving data privacy in a given table, while in our work we go further and propose a model that takes into account usage restrictions and client requirements. The works in [16][17] use third parties as database service providers. None of the previous works appear to be fully related to ours, as our proposal is situated at the level of privacy specification. We allow every service to outline its specification for the use and expectation of the manipulated data. Consequently, our work can be seen as a complement to these previous works. Additionally, during the final execution of the composition a mechanism of type *k*-anonymity can be applied to forbid all misuses of the data.

C. *Privacy-aware Adaptation*

The proposal in [18] is based on privacy policy lattices, which is created for mining privacy-preference/service-item correlations. Using this lattice, privacy policies can be visualized and privacy negotiation rules can then be generated. The Privacy Advocate approach [19] consists of three main units: privacy policy evaluation, signature and entity preference units. The negotiation focuses on data recipients and purpose only. An extension of P3P is proposed in [21]. It aims at adjusting a pervasive P3P-based negotiation mechanism for a privacy control. It implements a multi-agent negotiation mechanism on top of a pervasive P3P system. The approach proposed in [23] aims at accomplishing privacy-aware access control by adding negotiation protocol and encrypting data according to classification levels. Previous works are suffering from two major short- comings: The first one is the "take-it-or-leave-it" principle, i.e. a service can only accept or refuse the other service's proposal as a whole. The second is the "one-size-fits-all" principle: once the service producer has designed its privacy policy, it will be proposed to all interested services no matter what their requirements are. Languages such as *XACML*[37] or *ExPDT*[36] can be deployed over a variety of enforcement architectures. They are on the one hand syntactically expressive enough to represent complex policy rules, and offer on the other hand a formal semantics for operators to reason about policies, e.g. their conjunction and recently difference. Unfortunately, they do not provide adaptation mechanism when incompatibility occurs. Our privacy-aware adaptation approach goes beyond previous approaches and aims at ensuring privacy compatibility of involved services in the composition without any additional overload. It allows services to define adaptation sets and reconciles them through dynamic protocols for finding the best choice without any loss of privacy.

VIII. Conclusion and Future directions

In this paper, we proposed a semantic and formal privacy model for DaaS services. This model allows services to specify their privacy expectation/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 requirements with rich expression capabilities and in a flexible manner. We introduced a cost protocol bas on our model for checking the compatibility of privacy requirements and policies. We have presented a privacy compatibility-aware DaaS composition approach to resolve privacy concerns at the composition time. We also proposed a privacy-aware adaptation approach based on service reputation to tackle the incompatibilities between requirements and policies using dynamic reconciliation protocols. We have presented a gradual-based approach that applies flexible modeling to characterize the adaptation sets. As future work, we intend to deal with the issue of finding the set of quasi-identifier attributes (*QI*). The process of obtaining k-anonymity for a

given table is to anonymize the QI quasi-identifier attributes values. Consequently, when QI is not properly determined, the appearance of QI attribute values in a released table may give out private information. Another perspective worth studying in the context of service composition is the role of anonymization. We would like to study whether anonymization methods can mitigate the vulnerability and keep the utility of anonymized data useful.

IX. REFERENCES

[1] Q. Yu, X. Liu, A. Bouguettaya, and B. Medjahed, "Deploying and managing Web services: issues, solutions, and directions," The VLDB Journal, vol. 17, pp. 537–572, May 2008.

[2] M. Carey, "Data delivery in a service-oriented world: the BEA aquaLogic data services platform," ACM SIGMOD, pp. 695–705, 2006.

[3] A. H. H. Ngu, M. P. Carlson, Q. Z. Sheng, and H.-y. Paik, "Semantic-based mashup of composite applications," IEEE Trans. Serv. Comput., vol. 3, no. 1, pp. 2–15, January 2010.

[4] T. Weise, S. Bleul, D. Comes, and K. Geihs, "Different Approaches to Semantic Web Service Composition," International Conference on Internet and Web Applications and Services. Washington, DC, USA, pp. 90–96, 2008.

[5] M. Barhamgi, D. Benslimane, and B. Medjahed, "A Query Rewriting Approach for Web Service Composition," IEEE Transactions on Services Computing, vol. 3, no. 3, pp. 206–222, 2010.

[6] R. Vacula, H. Chen, R. Neruda, and K. Sycara, "Modeling and discovery of data providing services," International Conference on Web Service. pp. 54–61, 2008.

[7] E. Prud'hommeaux and A. Seaborne, "SPARQL Query Language for RDF," W3C, Tech. Rep., 2006. Available on: http://www.w3.org/TR/rdf- sparql-query/

[8] M. Mrissa, S.-E. Tbahriti, and H.-L. Truong, "Privacy model and annotation for DaaS ," European Conference on Web Services. pp. 3–10, 2010.

[9] S.-E. Tbahriti, M. Mrissa, B. Medjahed, C. Ghedira, M. Barhamgi, and J. Fayn. "Privacy-aware DaaS Services Composition," International Conference on Database and Expert Systems Applications, pp. 202–216, 2011.

[10] S.-E. Tbahriti, B. Medjahed, Z. Malik, C. Ghedira, and M. Mrissa, "Meerkat A Dynamic Privacy Framework for Web Services," IEEE/WIC/ACM International Conference on Web Intelligence, pp. 418–421, 2011.

[11] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, "Controlled Information Sharing in Collaborative Distributed Query Processing," International Conference on Distributed Computing Systems 2008.

[12] H. J. Zimmermann, "Fuzzy sets and its Applications" Book. ISBN: 0-7923-7435-5, 4th Edition. 2001.

[13] A. Machanavajjhala, J. Gehrke, and M. Götz, "Data publishing against realistic adversaries," VLDB, pp. 790–801, 2009.

[14] N. Mohammed, B. C. M. Fung, K. Wang, and P. C. K. Hung, "Privacy-preserving data mashup," EDBT, pp. 228–239, 2008.

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

[16] Y. Gil and C. Fritz, "Reasoning about the appropriate use of private data through computational workflows," In Intelligent Information Privacy Management, pp. 69–74, 2010.

[17] H. Hacigümüs, B. Iyer, C. Li, and S. Mehrotra, "Executing sql over encrypted data in the database-service-provider model," ACM SIGMOD, pp. 216–227, 2002.

[18] Y. Lee, D. Sarangi, O. Kwon, and M.-Y. Kim, "Lattice based privacy negotiation rule generation for context-aware service," International Conference on Ubiquitous Intelligence and Computing, pp. 340–352, 2009.

[19] M. Maaser, S. Ortmann, and P. Langendörfer, "The privacy advocate: Assertion of privacy by personalized contracts," In J. Filipe and J. A. M. Cordeiro, editors, WEBIST (Selected Papers), vol. 8, pp. 85–97, 2007.

[20] S. Agrawal and J. R. Haritsa, "A framework for high-accuracy privacy-preserving mining," ICDE, pp. 193–204, 2005.

[21] O. Kwon, "A pervasive p3p-based negotiation mechanism for privacy-aware pervasive e-commerce". Journal of Decis. Support Syst., vol. 50, no. 1, pp. 213–221, 2010.

[22] W3C, The Platform for Privacy Preference Specification, 2004.. Available: http://www.w3.org/TR/P3P11/ [25] W. D. W3C,ThePlatform for Privacy Preferences 1.1(P3P1.1)

[23] H.-A. Park, J. Zhan, and D. H. Lee, "Privacy-aware access control through negotiation in daily life service," IEEE ISI 2008 PAISI, PACCF, and SOCO international workshops on Intelligence and Security Informatics, PAISI, PACCF and SOCO, pp. 514–519, 2008.

[24] P. Persiano, I. Visconti, "An anonymous credential system and a privacy-aware PKI," Lecture Notes in Computer Science 2727/2003, pp. 27–38, 2003.

[25] Y. Xu, B. Zhang, and K. Wang, "Privacy-enhancing personalized web search", WWW, pp. 591–600, 2007.

[26] S. Nepal, Z. Malik, and A. Bouguettaya. "Reputation Management for Composite Services in Service-Oriented Systems". International Journal of Web Services Research (IJWSR), 8(2), pg. 29-52, April-June 2011.

[27] H.-L. Truong, S. Dustdar, J. Götze, T. Fleuren, P. Müller, S.-E. Tbahriti, M. Mrissa and C. Ghedira, "On Exchanging Data Agreements in the DaaS Model" In APSCC, pp. 153–60, 2011.

[28] B.-C. Chen, K. LeFevre, R. Ramakrishnan: Adversarial-knowledge dimensions in data privacy. VLDB J. 18(2): 429-467, 2009.

[29] E. Nageba, B. Defude, F. Morvan, C. Ghedira, J. Fayn "Data Privacy Preservation in Telemedicine: The PAIRSE project," Journal. Stud Health Technol Inform. Vol. 169, pg. 661-5, 2011.

[30] E.M Maximilien, M.P Singh, "Conceptual Model of Web Services Reputation". ACM SIGMOD Record, vol. 31, no. 4, pp. 36-41, 2002.

[31] H. Takabi, J.B.D. Joshi, G. Ahn, "Security and Privacy Challenges in Cloud Computing Environments," IEEE, Security & Privacy, vol. 8, no. 6, pp. 24–31, 2010.

[32] B. C. M. Fung, K. Wang, R. Chen and P. S. Yu, "Privacy-Preserving Data Publishing: A Survey of Recent Developments" In Journal of ACM Computing Surveys, vol. 42, no. 4, pp. 1–53, 2010.

[33] 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.

[34] R. Agrawal, A. Evfimievski, and R. Srikant. "Information sharing across private databases". ACM SIGMOD, pp. 86–97, 2003.

[35] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, and P. Samarati, "Access Control Policies and Languages," International Journal of Computational Science and Engineering (IJCSE), vol. 2, no.3, pp. 94–102, 2007.

[36] S. Sackmann and M. Kähmer, "ExPDT: A policy-based approach for automating compliance", Wirtschaftsinformatik Journal, vol. 50, no. 5, pp. 366–374, 2008.

[37] OASIS. Extensible access control markup language. http://www.oasis-open. org/committees/xacml/, 2008.

[38] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. "Qos-aware middleware for web services composition," IEEE Trans. Software Eng., 30(5). pp. 311–327, 2004.

[39] B. Pinkas, "Cryptographic Techniques for Privacy-Preserving Data Mining," SIGKDD Explorations, vol. 4, no. 2, pp. 12–19, 2002.

[40] M. Kähmer, M. Gilliot, and G. Müller. "Automating privacy compliance with expdt". IEEE Conference on E-Commerce and E-Services," pp. 87–94, 2008.

[41] R. Jurca and B. Faltings. "Reputation-based service level agreements for web services". International Conference on Service Oriented Computing, pp. 396–409, 2005.

[42] A. Cali, D. Martinengli, "Querying data under access limitations," International Conference on Data Engineering 2008

[43] IBM, Microsoft. Security in a Web services world: A proposed architecture and roadmap, (Apr. 2002); www-106.ibm.com/developer-works/webservices/library/ws-secmap/

[44] ID Experts. Data breaches cost, 2011. Available on: http://www2.idexpertscorp.com/press/healthcare-news/data-breaches-cost-the-healthcare-industry-an-estimated-65-billion/

[45] Privacy Rights Clearinghouse. Archive for the privacy incidents, 2011. URL http: //privacyguidance.com/blog/category/privacy-incidents/.