Privacy-Aware Web Service Composition and Ranking

Elisa Costante*, Federica Paci†, Nicola Zannone*
*Eindhoven University of Technology, The Netherlands
{e.costante,n.zannone}@tue.nl
†University of Trento, Italy
paci@disi.unitn.it

***Abstract*—Service selection is a key issue in the Future Internet, where applications are built by composing services and content offered by different service providers. Most existing service selection schemas only focus on QoS properties of services such as throughput, latency and response time, or on their trust and reputation level. By contrast, the risk of privacy breaches arising from the selection of component services whose privacy policy is not compliant with customers' privacy preferences is largely ignored. In this paper, we propose a novel privacy-preserving Web service composition and selection approach which (i) makes it possible to verify the compliance between users' privacy requirements and providers' privacy policies and (ii) ranks the composite Web services with respect to the privacy level they offer. We demonstrate our approach using a travel agency Web service as an example of service composition.**

I. Introduction

The Future Internet will be characterized by a new generation of applications built by composing services and data from different providers and organizations in order to provide users with added-value services tailored to their needs. Web services play a key role in realizing this vision because they can be advertised, located, and composed over the Internet using standards like WSDL, UDDI and BPEL, respectively. Typically, Web service composition is represented by a plan consisting of tasks that, at run-time, are instantiated to the actual services satisfying users' requirements. Due to the increasing number of services available offering similar functionalities, it is hard for users to select an optimal service composition among a list of candidate services that satisfy their needs. Therefore, service selection is a key challenge in the Future Internet.

The literature offers a large amount of work on Web service composition and selection. Most of the existing approaches focus on the identification of optimal Web services among a set of candidates based on constraints on the Quality of Service (QoS) performance of the candidates [1], [2], [3], [4], [5], or on their trust and reputation level [6], [7], [8], [9]. To the best of our knowledge, only few works have investigated privacy issues in service selection [10], [11] and composition [12], [13], [14]. Despite the limited effort, privacy plays a major role in Web service composition and selection. The orchestrator usually collects a large amount of personal data about their clients and eventually shares these data with the service providers providing the orchestrated services. This, however, may lead to risks of data misuse. For instance, a service provider may use client data for unlawful purposes. As a consequence, more and more users are considering privacy practices adopted by Web service providers as an important factor for service selection: users will more likely use Web services that customize the service provision based on users' privacy preferences.

In this paper, we propose an approach to assist both users and Web service providers in composing and selecting optimal services with respect to their privacy preferences. We use AND/OR trees to represent the orchestration schema, component services and their privacy policies. Based on this representation, we present an algorithm that determines the Web service compositions compliant with user privacy preferences. To help them to select the best Web service composition, our approach ranks admissible composite Web services (i.e., composite services whose privacy policy satisfy user preferences) with respect to their *privacy level*. The privacy level quantifies the risk of misuse of personal data based on three dimensions: the sensitivity, visibility and retention period of information.

The contribution of this paper is three-fold. First, we propose a fine-grained model to express Web service providers' privacy policies and users' privacy preferences based on several privacy dimensions – sensitivity, purpose, retention period, visibility – while other approaches to privacy-aware service composition only consider one dimension, e.g. sensitivity or visibility. Second, we propose a Web service composition algorithm which merges into a single step the selection of services that satisfy users' functional requirements and the selection of services compliant with users' privacy requirements, while most existing approaches execute these two steps separately. Last but not least, we rank composite services with respect to the level of privacy they offer, while other approaches only focus on the generation of a privacy-preserving composition. We illustrate our privacy-aware composition and selection process using a travel agency Web service as a running example.

The remainder of the paper is structured as follows. Section II discusses related work. Section III presents a modeling framework for representing service orchestrations, users' privacy preferences and Web service providers' privacy policies. Section IV presents the privacy-aware service composition and selection process. Section V concludes the paper providing directions for future work.

978-0-7695-5025-1/13 $26.00 © 2013 IEEE
DOI 10.1109/ICWS.2013.27

II. Related Work

Our work is related to the fields of *service composition modeling*, *service composition*, and *service selection*.

a) Service composition modeling: To model service composition and verify whether it satisfies properties like safety and liveness, several languages, such as WS-BPEL [15], or approaches, such as process algebra [16], Petri nets [17], model checking [18], and finite state machines [19], have been proposed. Contributions to service composition modeling also come from the requirement engineering community, where goal-oriented approaches [20], [21] are used to represent strategic business goals. Similarly, we adopt a goal-oriented approach to model service composition. The advantage of such an approach is that it provides the abstraction necessary to represent privacy policies without getting bogged down into the functioning of Web services.

b) Service composition: Service composition is the problem of aggregating services in such a way that given (functional and not functional) requirements are satisfied. The role of privacy in service composition has been investigated in [13], where only services requiring the disclosure of less sensitive information and offered by trusted providers are selected in the composition. Users' privacy concerns are often addressed by providing automated techniques for matching provider's privacy policies with customer's preferences [12], [14], [22], [23], [24]. The most prominent solution for policy matching is P3P (Platform for Privacy Preferences Project) [22]. P3P aims to assist service providers in specifying their privacy practices on the Web, and users in matching such practices against their preferences. To automate the matching process, P3P has been complemented with privacy preferences languages such as APPEL [25] and XPref [26]. In [23] service composition is the result of a negotiation phase between user privacy preferences (describing the type of access to each piece of personal information) and the Web service policy statement (specifying which information is mandatory and which is optional to use a service). Here, the outcome of the negotiation indicates what personal information the user should disclose to the service provider. However, these techniques only focus on the relation between a server and a client. In contrast, our work uses a privacy policy matching approach to build the model of admissible service compositions. In addition, our work goes beyond pure service composition: we also identify the most privacy preserving composition.

c) Service selection: Service composition might return a set of admissible services; thus, service ranking is needed to choose the *best* composition. QoS-based [1], [2], [3], [4], [5] and trust-based [6], [7], [8], [9] service selection has been widely investigated in the literature. Privacy-aware service selection is addressed in [11] which presents a comprehensive framework to protect users' and service providers' privacy needs at selection time. Users' criteria are matched against Web services' attributes in a private fashion such that both criteria and service attributes are kept private. This approach mainly focuses on protection of service provision rules from unwanted disclosure, while our goal is to select the most privacy preserving composition. Massacci et al. [10] present an approach to service selection based on the sensitivity of data to be disclosed for the service provision. In contrast, we consider a number of criteria characterizing privacy policy and user preference for selecting the optimal service composition. Similar criteria are also considered in [27]. However, these criteria are not used to assess the privacy level of services. Rather, they are used to capture discrepancies between what stated in privacy policies and what is done in practice. To allow service ranking, we aggregate the identified criteria using an approach based on the norm. Although more complex solutions like swap [28] or collaborative filtering [29] have been proposed to assist users in multi-criteria decision making, such solutions either require a high level of user interactions and thus cannot be automatized, or are not applicable due to the nature of privacy criteria.

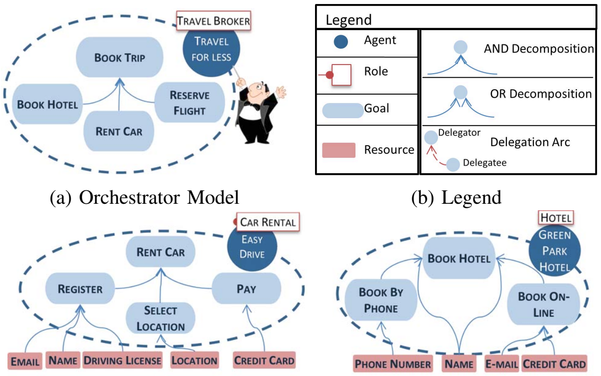
III. Modeling Service Composition and Privacy

In this section we introduce the models to represent Web service orchestration, privacy policies and user privacy preferences on which our approach is based.

A. Modeling Service Orchestration

In Web services composition typically there is an *orchestrator* which combines the functionalities provided by other services usually denoted as *component services* to satisfy users' requests. Several services may be able to provide the same functionality requested by the user. The service resulting from the orchestration is called *composite service*. We model the composition schema as an *orchestrator model*, each component service as a *component service model*, and all possible alternative instantiations of the schema as a *service orchestration model*.

We represent these models as AND/OR trees where the semantics of nodes and arcs is based on the concepts defined by SI* [30], a goal-oriented framework for requirements elicitation and analysis. SI* employs the notions of *actor*, *goal*, *resource*, *decomposition* and *delegation*. *Actors* are active entities that have strategic goals and perform actions to achieve them. Actors can be *agents* or *roles*: agents are used to represent the orchestrator and component services, and roles to represent the types of services. The sets of agents and roles are denoted A and T respectively, with $A \cap T = \emptyset$. We use notation $s \rhd t$ to indicate that a service $s \in A$ is of type $t \in T$. *Goals* represent the functionalities offered by services, while *resources* represent data produced/consumed by a goal. The sets of goals and resources are denoted G and R, respectively. *Decomposition* is used to refine a goal: AND decomposition refines a goal into subgoals and resources needed to achieve the goal, while OR decomposition defines alternatives to achieve a goal. *Delegation* marks a formal passage of responsibility or authority from an actor (*delegator*) to another actor (*delegatee*) to achieve a goal. We use these concepts to define the notion of *service model*.



(a) Orchestrator Model

(b) Legend

(c) Component Service Model with AND Decomposition

(d) Component Service Model with OR Decomposition

Fig. 1. Examples of our Modeling

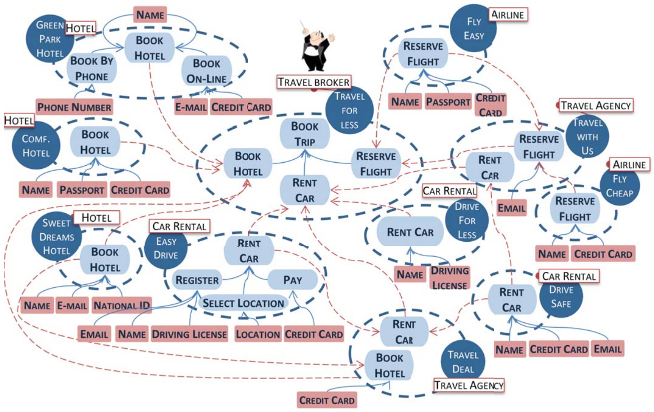


Fig. 2. Example of Orchestration Model

Definition 1 (Service Model): A *service model* S is a pair $\langle V, E\rangle$ where: $V = G \cup R$ is the set of nodes; E is the set of decomposition arcs $\langle Z, g\rangle$ connecting a node $g \in G$ to a non-empty set $Z \subseteq V$.

Example 1: TravelForLess is a composite Web service which provides its customers deals including hotel reservations, flight bookings, car rentals, or any combination of these travel options.To this end, TravelForLess relies on partner travel agencies, hotels chains, airline companies and car rental agencies that are dynamically selected. Fig. 1a illustrates the orchestrator model of TravelForLess, while Fig. 1b shows the list of symbols used through our examples and their meaning. TravelForLess provides goal book trip, represented by the top oval. This goal is decomposed into sub-goals book hotel, reserve flight and rent car. Figs. 1c and 1d illustrate examples of component services of types Car Rental Agency and Hotel. □

The service orchestration model is obtained by merging the service models associated with the orchestrator and all component services. In particular, we merge the service model of the outsourcer with the service model of the subcontractor by linking the goal of the former with the corresponding goal (with the same name) occurring in the service model associated with the latter. Intuitively, goals with the same name represent the same functionality and, therefore, can be considered equivalent (although they may require different data items or can be decomposed differently). Let S_1 and S_2 be two service models. We write $n_1 \equiv n_2$ to denote that $n_1 \in S_1$ and $n_2 \in S_2$ are equivalent. Arcs linking nodes across service models are called *delegation arcs*. If more than one component service can fulfill the goal, each such component service is linked to the goal of the outsourcer. Notice that a component service may not have the capabilities to fully achieve a goal. In this case, the component service may redelegate the achievement of (part of) the goal to another component service.

Example 2: Fig. 2 shows the orchestration model obtained by merging the service model of TravelForLess with the ones of the candidate component services. In the figure, delegation arcs are represented as dashed arrows. The model represents all possible alternatives to fulfill the goals of TravelForLess. Goal rent car can be provided by two car rentals, DriveForLess and EasyDrive, and by two travel agencies, TravelDeal and TravelWithUs. Goal book hotel can be fulfilled by SweetDreamsHotel, GreenParkHotel and Comfort Hotel, while goal reserve flight is delegated to airline company FlyEasy and to travel agency TravelWithUs. The partner services may require different information to fulfill the goal they provide. For example, to fulfill goal book hotel, SweetDreamsHotel requires its customers to provide name, email, credit card and national ID, while ComfortHotel requires name, passport, and credit card. □

A composite service is a particular sub-tree of the service orchestration model which represents a possible alternative to fulfill its root goal. Before formally defining a composite service, we introduce the notion of *decomposition path*.

Definition 2 (Decomposition Path): Let $S = \langle V, E\rangle$ be a service orchestration model, $Z \subset V$ be a non-empty set of goals and resources, and g be a goal in V. A *decomposition path* $D_{Z,g}$ is a set of arcs $E' \subset E$ such that either $g \in Z$, or there exists a decomposition arc $\langle T, g\rangle \in E'$ and there are decomposition paths $D_{Z,x} \in E'$ for each $x \in T$.

Definition 3 (Composite service): Let $S = \langle V, E\rangle$ be a service orchestration model. A *composite service* is a decomposition path D_{Z,g_0} such that $Z \subset V \cap R$ is a set of data items and g_0 is the root goal of S.

Example 3: Fig. 3 shows a possible composite service within the orchestration model of Fig. 2 where Travel ForLess's goals, book hotel, rent car, and reserve flight are fulfilled by the services GreenParkHotel, EasyDrive, and TravelWithUs, respectively. In turn, TravelWithUs delegates the fulfillment of reserve flight to FlyEasy. □

B. Modeling Privacy

To complete the interaction with a Web service (composite or simple), the user has to disclose her personal information to the service. However, users may be concerned about disclosing their personal data. Data protection legislation aims to address

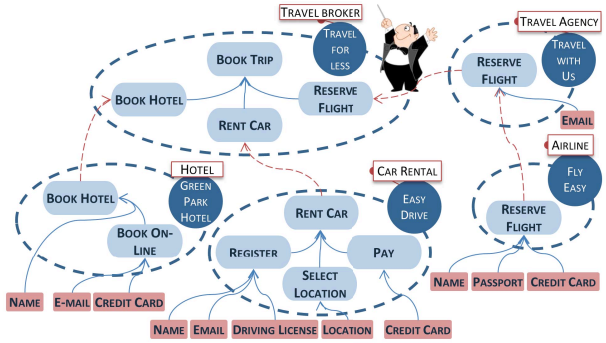


Fig. 3. Example of Composite Service

these user concerns. On one hand, data protection legislation recognizes the right of users to control their data [31]. To this end, they may define *privacy preferences* which specify constraints on the collection and processing of their data. On the other hand, Web service providers (both the orchestrator and component services) are obliged by law to publish *privacy policies* in which their privacy practices are declared.

Here, we consider four privacy dimensions which are typically used to specify privacy policies and privacy preferences: *purpose* defines the reason(s) for data collection and usage; *visibility* defines to whom data can be disclosed; *retention period* defines how long data can be maintained; *sensitivity* represents the data subject's perception of the harm the misuse of her data can cause to her. Based on these privacy dimensions, privacy policies can be formally defined as follows.

Definition 4 (Privacy policy): A privacy policy is a set of tuples $\langle d, p, \nu, \delta, \tau \rangle$ where: $d \in R$ denotes a data item; $p \in G$ is the purpose for which d can be collected; $\nu \subset A \cup T$ is the visibility of d for achieving p; $\delta \in \mathbb{N} \cup \{*\}$ represents the (re)delegation depth which is used to limit the sharing of d for achieving p (Depth 1 means that no further sharing is allowed, n means that $n-1$ further steps are allowed, and depth "*" means unlimited sharing); $\tau \in \Re$ represents the retention period (here in months) of d for achieving p.

Although the notation introduced in Definition 4 makes it possible to capture the privacy dimensions necessary to specify privacy policies, it makes it difficult to understand and reason on the specified privacy policies. To this end, we represent privacy policies as AND/OR trees where nodes model the purposes in the policy's tuples and data items protected by the policy. This representation resembles the service model. For instance, the privacy policy in Table I can be graphically represented using a model similar to Fig. 1a. The main difference between the two models is that nodes are annotated with visibility, (re)delegation depth and retention period. Formally, a *privacy policy model* is a tuple $\langle V, E, \Gamma \rangle$ where $\langle V, E \rangle$ is the corresponding service model and Γ is the privacy policy in tabular form. Given a privacy policy Γ and a purpose p, $\Gamma^p = \{\langle d, \nu, \delta, \tau \rangle | \langle d, p, \nu, \delta, \tau \rangle \in \Gamma\}$. We say that a privacy policy Γ is *well-defined* if (i) $\forall \langle d, p, \nu, \delta, \tau \rangle \in \Gamma\ \nu \neq \emptyset$ iff $\delta > 1$ and (ii) for every data item d and purpose p such that $\langle d, \nu, \delta, \tau \rangle \in \Gamma^p$, $\nexists \langle d', \nu', \delta', \tau' \rangle \in \Gamma^{p'}$ with p' sub-purpose of p and $d = d'$. Intuitively, the first condition states that the visibility can be defined if and only if the delegation depth is greater than 1, and the second imposes that the privacy policy for a data item is not redefined during policy refinement. In this paper, we only consider well-defined privacy policies.

Data Item (d)	Purpose (p)	Visibility (ν)	Depth (δ)	Retention (τ)
Name	Rent Car	All	*	36
	Book Hotel	All	*	24
	Reserve Flight	All	*	24
Email	Rent Car	All	*	24
	Book Hotel	All	*	24
	Reserve Flight	All	*	36
Credit Card	Rent Car	Car Rental, Travel Agency	3	12
	Book Hotel	Hotel, Travel Agency	2	12
	Reserve Flight	Airline	2	12
Passport	Reserve Flight	Travel Agency, Airline	2	12
Driving License	Rent Car	Travel Agency, Car Rental	3	12
Phone Number	Rent Car	All	*	36
	Book Hotel	All	*	36
	Reserve Flight	All	*	36
National ID	Reserve Flight	Travel Agency	3	18
		Airline	2	12
	Book Hotel	Travel Agency, Hotel	3	12
Location	Rent Car	Car Rental	2	12

TABLE I
TravelForLess's PRIVACY POLICY

To compare the privacy policy of different services, we introduce the notion of *policy compliance*. We say that the privacy policy of a service complies with the privacy policy of another service if the former is more restrictive than the latter. Policy compliance is formally defined as follows.

Definition 5 (Policy compliance): Let Γ_x and Γ_y be two well-defined privacy policies. Γ_y complies with Γ_x, denoted as $\Gamma_y \rightsquigarrow \Gamma_x$, if $\forall p\ \forall \langle d_1, \nu_1, \delta_1, \tau_1 \rangle \in \Gamma_y^p\ \exists \langle d_2, \nu_2, \delta_2, \tau_2 \rangle \in \Gamma_x^p$ such that (i) $d_1 = d_2$; (ii) $\delta_1 < \delta_2$; (iii) $\tau_1 \leq \tau_2$.

Example 4: The privacy policy of TravelForLess is presented in Table I. The policy specifies how TravelForLess will use customers' data. For example, TravelForLess will collect a customer's name to fulfill purpose rent car and it will maintain a copy of the data item for 36 months. Moreover, the policy states that TravelForLess can disclose customers' name to services of any type (which is denoted by "all"). Since the depth is set to *, any service receiving directly or indirectly a copy of name can further share it with no limitation. Customers' national ID can be collected only for purpose reserve flight and has different rules for different agents: if the component service is an instance of Travel Agency the national ID can be shared with other services and can be stored up to 18 months; in the case the component service is an instance of Airline, the national ID cannot be delegated further, and can be kept only for 12 months. □

When interacting with the orchestrator, a user should analyze the policy of the orchestrator and decide whether it is acceptable. The user can refine the policy by limiting the requested functionalities and restricting the use of data items. In particular, she can restrict the visibility of a certain data item by denying sharing it with a certain type of service or selecting specific component services. In addition, the user may decide

Data Item (d)	Sensitivity (σ)	Purpose (p)	Visibility (ν)	Depth (δ)	Retention (τ)
Name	5	Rent Car	All	*	36
		Book Hotel	All	*	24
Email	5	Rent Car	All	*	24
		Book Hotel	All	*	24
Credit Card	10	Rent Car	Travel With Us	3	12
	8	Book Hotel	Travel Deal, Green Park Hotel	2	12
Driving License	9	Rent Car	Drive For Less	3	12
National ID	6	Book Hotel	Travel Agency, Hotel	3	12

TABLE II
Bob's PRIVACY PREFERENCES

to not disclose a certain data item. Finally, the user should define the sensitivity of each data item, which may vary from purpose to purpose. Users, however, are not allowed to change the delegation depth and retention period. This is because these attributes are often constrained by the business model of the orchestrator as well as by the requirements imposed by the legal framework in force (e.g., telecommunications data have to be stored for six to 24 months according to the EU Directive on data retention). The result of this refinement process represents the privacy preferences of the user.

We formally specify users' privacy preferences as follows.

Definition 6 (Privacy preferences): The privacy preferences of a user are a set of tuples $\langle d, p, \sigma, \nu, \delta, \tau \rangle$ where: $d \in R$ denotes a data item; $p \in G$ is the purpose for which d can be collected; $\sigma \in [1, 10]$ is the sensitivity of d; $\nu \subset A \cup T$ is the visibility of d for achieving p; $\delta \in \mathcal{N} \cup \{*\}$ is the (re)delegation depth which limits the sharing of d for achieving p; $\tau \in \Re$ is the retention period of d.

Example 5: Let Bob be a new customer of TravelForLess. He wants to book a trip to Barcelona but, since he is afraid to fly, he only wants to book a hotel and rent a car. Based on the privacy policy of TravelForLess (Table I), he specifies constraints on the collection and processing of his data. Bob's privacy preferences are presented in Table II. Since name and email are usually required by service providers, Bob leaves their visibility to all. In contrast, he prefers that his credit card is only disclosed to agents he trusts, i.e. TravelWithUs, TravelDeal and GreenParkHotel. Bob also restricts the access to his driving license only for the purpose of renting a car, and the national ID only for booking a hotel. Finally, Bob prefers to be contacted by email and thus he is not willing to disclose his phone number. □

IV. Privacy-aware Service Selection

Figure 4 shows the architecture of our approach which consists of two main components: a) the *Privacy-Aware Orchestrator* queries the *Service Repository* to select Web services that match users' functional and privacy requirements for the composition and composes them according to schema; b) the *Privacy Aware Ranker* prioritizes the admissible composite services based on their privacy level. To avoid interoperability issues between clients and Web service providers, we assume that privacy preferences and privacy policies are expressed in the WS-Policy standard extended with privacy-specific assertions.

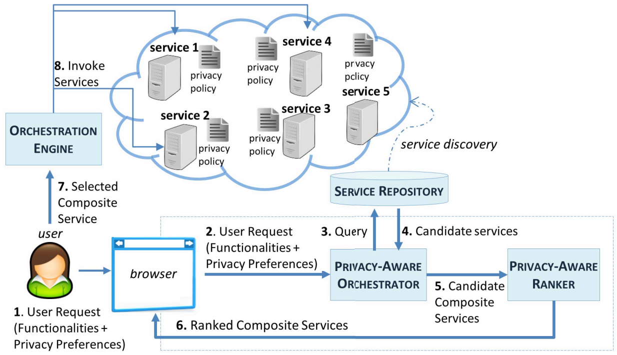


Fig. 4. Privacy-Aware Service Composition and Ranking Architecture

In what follows we describe in details the operations performed by the architectural components.

A. Service Composition

Service orchestrators usually do not provide the functionalities required by a client directly but they outsource the provision to specialized services. Nonetheless, according to the EU privacy regulation, they are liable for the actions performed by the subcontractors. Therefore, an orchestrator is willing to select a component service only if the privacy policy of the component service complies with its policy and user privacy preferences. The aim of the service orchestration composition step is to identify *admissible composite services*, i.e. those composite services that comply with the user preferences and legal requirements.

After a user has defined her privacy preferences through the refinement of the orchestrator's privacy policy (see Example 5), the orchestrator uses those preferences to identify admissible composite services. Admissible composite services are determined using Algorithm 1. The algorithm builds the privacy model of the service orchestration that includes only those component services whose privacy policy complies with the privacy preferences of the user (for the sake of simplicity, here we omit sensitivity in the user preferences, and represent them using the notation for privacy policies; sensitivity is used in the next step). The algorithm first identifies the portion of the policy model of the orchestrator related to the functionalities required by the user (lines 5-20). The policy associated with a purpose is propagated to sub-purposes (lines 16-17). Intuitively, a purpose inherits the constraints from the higher level purpose. This makes it possible to check the consistency of policies along the service orchestration model.

When the policy of the orchestrator is fully analyzed, the algorithm identifies the component services which offer the functionalities required by the user and whose privacy policy is compliant with the privacy policy of the service delegating the service to them (lines 21-41). If the node to be analyzed is not a leaf node of the policy (line 24), the algorithm checks whether the policy associated with the subnodes of that node complies with the policy associated with the leaf node in the policy of the service delegating the provisioning

Algorithm 1: Service Composition

```
Input: $S_u$ set of functionalities requested by user $u$,
       $P_o = \langle V_o, E_o, \Gamma_o \rangle$ privacy policy model of the orchestrator
       augmented with the privacy preferences of $u$, $\mathcal{P}$ set of the
       privacy policy models of component services
Output: $P$ privacy policy model of the service orchestration
1  let $P = \langle V, E, \Gamma \rangle$;
2  let $V = \{root\}$, $E = \emptyset$, $\Gamma = \emptyset$;
3  make $Q$ empty ;  //$Q$ is a queue containing the nodes to be
   visited
4  make $S$ empty ;   //$S$ is a queue containing pairs of nodes
   where the first element represents the reference node
   and the second represents the node to be visited
5  for $s \in S_u$ do
6      $V = V \cup \{s\}$;
7      $\Gamma^s = \Gamma_o^s$;
8      insert $s$ in $Q$;
9  $E = E \cup \{\langle S_u, root \rangle\}$;
10 while $Q$ is not empty do
11     extract $s_i$ from $Q$;
12     if $s_i$ not leaf node then
13         for $\langle Z, s_i \rangle \in E_o$ do
14             $V = V \cup Z$;
15             $E = E \cup \{\langle Z, s_i \rangle\}$;
16             for $s_j \in Z$ do
17                 $\Gamma^{s_j} = \Gamma^{s_i} \cup \Gamma_o^{s_j}$;
18                 insert $s_j$ in $Q$;
19     else
20         insert $(s_i, s_i)$ in $S$
21 while $S$ is not empty do
22     extract $(s_k, s_i)$ from $S$;
23     let $P_x = \langle V_x, E_x, \Gamma_x \rangle$ be the policy model s.t. $s_i \in V_x$;
24     if $s_i$ not leaf node then
25         for $\langle Z, s_i \rangle \in E_x$ do
26             if $\Gamma_x^Z \rightsquigarrow \Gamma^{s_k}$ then
27                 $V = V \cup Z$;
28                 $E = E \cup \{\langle Z, s_i \rangle\}$;
29                 for $s_j \in Z$ do
30                     $\Gamma^{s_j} = \Gamma^{s_i} \cup \Gamma_x^{s_j}$;
31                     insert $(s_k, s_j)$ in $S$;
32     else if $s_i$ is a purpose node then
33         let $W = \{w | \langle d, \nu, \delta, \tau \rangle \in \Gamma_x^{s_i} \wedge ((w \in$
           $\nu \cap A) \vee (w \rhd t \wedge t \in \nu \cap T))\}$;
34         for $w \in W$ do
35             let $P_w = \langle V_w, E_w, \Gamma_w \rangle$ be the policy model of $w$;
36             if $\exists s_j \in V_w$ s.t. $s_j \equiv s_i$ then
37                 if $\Gamma_w^{s_j} \rightsquigarrow \Gamma^{s_k}$ then
38                     $V = V \cup \{s_j\}$;
39                     $E = E \cup \{\langle \{s_j\}, s_i \rangle\}$;
40                     $\Gamma^{s_j} = \Gamma_w^{s_j}$;
41                     insert $(s_i, s_j)$ in $S$;
```

of the functionality (called *reference node*) (line 26). If it is compliant, the nodes are added to the policy model of the orchestration (lines 27-31).

If the node to be analyzed is a leaf node of the policy, the algorithm checks whether it is a purpose node (line 32). This case corresponds to situations in which the service does not have the capability to provide the functionality and outsources its provision to another service. Visibility is used to determine which component services should be considered in the orchestration (line 33). A component service in the visibility is considered by the algorithm if it actually offers the required functionality (line 36). If the policy associated with the new node complies with the policy of the outsourcer (line 37), the node together with a delegation arc is added to the policy model of the orchestration (lines 38-40).

The privacy policy model returned by Algorithm 1 corresponds to the privacy policy regulating the service orchestration. The composite services in the policy model of the service orchestration are the admissible composite services.

Proposition 1: Let Π be the privacy preferences of a user and P the privacy policy model of the service orchestration obtained through Algorithm 1 wrt Π. The privacy policy of every composite service $p \in P$ complies with Π. □

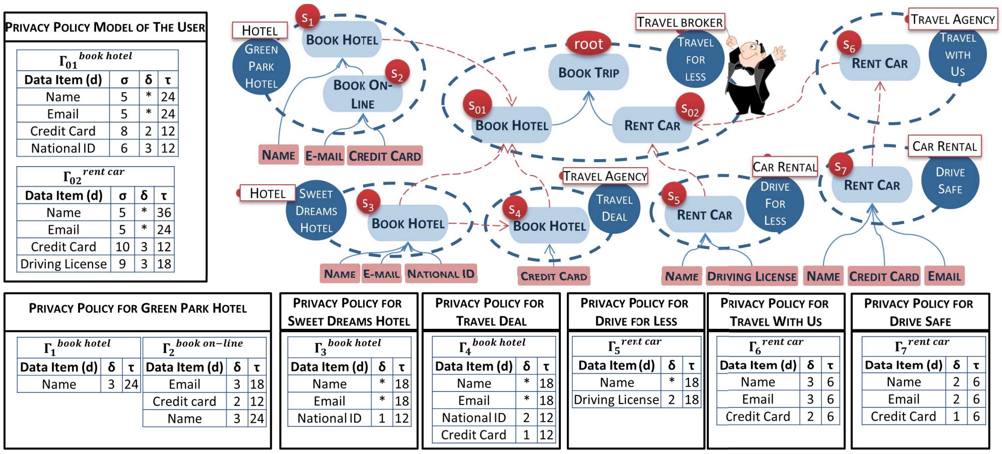
The proof is by induction on the depth of the privacy policy model of the service orchestration. Notice that some composite services compliant with user privacy preferences may be discarted as compliance of the policy of a service is verified against the policy of the outsourcer (which may be more restrictive than user privacy preferences). This reflects the fact that, by law, the outsourcer is liable for the subcontractor. Therefore, a service would outsource (part of) its duties only to those services whose privacy practices are acceptable for it.

Example 6: Fig. 5 shows the orchestration policy model based on Bob's privacy preferences (Table II) together with the policies of the selected component services. The model describes six admissible composite services that can be employed to provide the functionalities requested by Bob (see Fig. 6b for their description). Note that, for readability reasons, we have omitted the *visibility* field in the figure. □

B. Composite Service Ranking

More than one composite service may satisfy a user's privacy preferences. In order to support the user in the decision making, we prioritize admissible composite services according to their privacy level. Intuitively, a composite service is more privacy-preserving if it requires the disclosure of less sensitive data as well as it retains data for less time and its constraints on their delegation are more restrictive.

To assess and compare the privacy level of admissible composite services, we represent their privacy policy in a three dimensional graph whose axes represent retention period, (re)delegation depth and sensitivity. In Definition 4 the privacy policy is defined as a set of tuples. The overall privacy level with respect to retention period and (re)delegation depth is obtained by aggregating the values of these dimensions in the tuples forming the policy of the composite service. Retention period and (re)delegation depth are weighted with respect to the sensitivity of the data item. This is to reflect the higher privacy risk of storing high sensitive data for a long time and potentially sharing them with more services. The sensitivity value associated with a composite service is given by the sum of the sensitivity of all data items that have to be shared for the execution of the component service. Notice that, although sensitivity is considered "twice", it has a different impact on the privacy level. While sensitivity as a dimension is used to measure the amount of information that needs to be disclosed by the user, sensitivity as a weight for retention period and (re)delegation depth is used to characterize the privacy risks associated with these two dimensions.



Privacy Policy Model of The User

$\Gamma_{01}^{book\ hotel}$			
Data Item (d)	σ	δ	τ
Name	5	*	24
Email	5	*	24
Credit Card	8	2	12
National ID	6	3	12

$\Gamma_{02}^{rent\ car}$			
Data Item (d)	σ	δ	τ
Name	5	*	36
Email	5	*	24
Credit Card	10	3	12
Driving License	9	3	18

Privacy Policy for Green Park Hotel

$\Gamma_{1}^{book\ hotel}$		
Data Item (d)	δ	τ
Name	3	24

$\Gamma_{2}^{book\ on-line}$		
Data Item (d)	δ	τ
Email	3	18
Credit card	2	12
Name	3	24

Privacy Policy for Sweet Dreams Hotel

$\Gamma_{3}^{book\ hotel}$		
Data Item (d)	δ	τ
Name	*	18
Email	*	18
National ID	1	12

Privacy Policy for Travel Deal

$\Gamma_{4}^{book\ hotel}$		
Data Item (d)	δ	τ
Name	*	18
Email	*	18
National ID	2	12
Credit Card	1	12

Privacy Policy for Drive for Less

$\Gamma_{5}^{rent\ car}$		
Data Item (d)	δ	τ
Name	*	18
Driving License	2	18

Privacy Policy for Travel With Us

$\Gamma_{6}^{rent\ car}$		
Data Item (d)	δ	τ
Name	3	6
Email	3	6
Credit Card	2	6

Privacy Policy for Drive Safe

$\Gamma_{7}^{rent\ car}$		
Data Item (d)	δ	τ
Name	2	6
Email	2	6
Credit Card	1	6

Fig. 5. Example of Service Composition

We represent the privacy level of a composite service as a three dimensional vector.

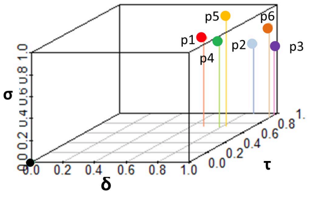
Definition 7 (Privacy level): Let Γ_0 be the privacy policy of the orchestrator, $\Gamma_1, \ldots, \Gamma_n$ the privacy policies of component services, $P = \langle V, E, \Gamma \rangle$ the privacy policy model of a composite service, and Π the privacy preference of a user. Let $\overline{\Gamma} = \{\langle d, p, \nu, \delta, \tau \rangle | \langle d, p, \nu, \delta, \tau \rangle \in \Gamma \cap \Gamma_i\}$. The privacy level of the composite service is a vector $[\delta, \tau, \sigma]$ such that

- $\delta = avg\left(\sigma_j \delta_i | \langle d, p, \nu, \delta_i, \tau_i \rangle \in \overline{\Gamma} \wedge \langle d, p, \sigma_j, \nu, \delta_j, \tau_j \rangle \in \Pi\right)$
- $\tau = avg\left(\sigma_j \tau_i | \langle d, p, \nu, \delta_i, \tau_i \rangle \in \overline{\Gamma} \wedge \langle d, p, \sigma_j, \nu, \delta_j, \tau_j \rangle \in \Pi\right)$
- $\sigma = \sum_{\langle d, p_i, \nu_i, \delta_i, \tau_i \rangle \in \overline{\Gamma}} \sigma_j$ s.t. $\langle d, p_j, \sigma_j, \nu_j, \delta_j, \tau_j \rangle \in \Pi \wedge \nu_i \subset \nu_j \wedge (p_i = p_j \vee (\exists \langle p_j, Z \rangle \in E \text{ s.t. } p_i \in Z))$

Note that in Γ some tuples are duplicated because Algorithm 1 propagates them to sub-purposes, while the original policies $\Gamma_0, \Gamma_1, \ldots, \Gamma_n$ may contain tuples that are not applicable for the given composite service. The set of tuples $\overline{\Gamma}$ contains only the tuples that are relevant for the composite service and does not contain duplicates. Moreover, notice that every tuple in Γ has a counterpart in Π. If this is not the case, then the composite service is not admissible and therefore it would not be considered at this stage.

The dimensions obtained above range in different scales. To make them comparable, they need to be normalized. Also, when the (re)delegation depth is unlimited ($\delta = *$), for the sake of computation, we bound its value to 10. Let S be the set of admissible component services and Ω^S the vector space containing the privacy level of the services in S. Let δ_{max}, τ_{max}, σ_{max} be defined as follows: $\delta_{max} = max(\delta_i \mid [\delta_i, \tau_i, \sigma_i] \in \Omega^S)$, $\tau_{max} = max(\tau_i \mid [\delta_i, \tau_i, \sigma_i] \in \Omega^S))$, $\sigma_{max} = max(\sigma_i \mid [\delta_i, \tau_i, \sigma_i] \in \Omega^S))$. Let $\omega_i = [\delta_i, \tau_i, \sigma_i] \in \Omega^S$ be the privacy level of $s_i \in S$, its normalized privacy level $\overline{\omega_i}$ is obtained dividing each component of the vector for the corresponding maximum value, i.e. $\overline{\omega_i} = \left[\frac{\delta_i}{\delta_{max}}, \frac{\tau_i}{\tau_{max}}, \frac{\sigma_i}{\sigma_{max}}\right]$.

If the normalized vector corresponding to a composite service is optimal with respect to all dimensions, such a composite service is the most privacy-preserving composite service. Otherwise, the most privacy-preserving composite service should be determined by analyzing the components forming the privacy level. However, end-users often are not able to understand the consequences of their privacy preferences. In addition, requiring the user to specify additional information makes the level of her involvement too high [29] and, thus, the selection process cannot be automated.



(a) Graph Representation

CS	Description	δ	τ	σ	Norm
P1	$S_{01}, s_1, s_2, s_{02}, s_6, s_7$	0.67	0.75	0.80	1.29
P2	$S_{01}, s_1, s_2, s_{02}, s_5$	0.85	1.00	0.64	1.46
P3	S_{01}, s_3, s_{02}, s_5	1.00	0.96	0.63	1.52
P4	$S_{01}, s_3, s_{02}, s_6, s_7$	0.79	0.72	0.79	1.33
P5	$S_{01}, s_4, s_3, s_{02}, s_6, s_7$	0.82	0.74	1.00	1.49
P6	$S_{01}, s_4, s_3, s_{02}, s_5$	0.99	0.93	0.84	1.59

(b) Admissible Composite Services

Fig. 6. Privacy-preserving Composite Service Ranking

Decision making should be simple and intuitive as well easy to review [32]. Therefore, instead of asking the user to set her priorities over the privacy dimensions, we aggregate them using an approach based on the norm. Intuitively, the privacy of a composite service is computed as the average of the criteria forming the privacy level. Given a privacy level $\omega_i \in \Omega^S$, we denote the norm of its normalization as $\|\overline{\omega_i}\|$. The composite service, for which the norm of its normalized privacy level $\|\overline{\omega_i}\|$ is the lowest, is most privacy-preserving composite service, i.e. $min(\|\overline{\omega_i}\| \mid \omega_i \in \Omega^S)$.

Example 7: Each admissible composite service (CS) in Fig. 5 is represented as a 3D-point in Fig. 6a. The dimensions δ, ν and σ as well as the norm for each composite service are presented in Fig. 6b. The height of a point represents its aggregated sensitivity, whereas the most right points are those with a higher depths, and those more in the back have a longer retention period. Intuitively, we prefer those composite services represented by the lowest, left-most, front-most points on the graph. The norm gives a precise measure of the privacy level of composite services and, thus, makes it possible to distinguish the most privacy-preserving composite service, represented by p_1 in our example. □

Notice, however, that the framework is flexible enough to allow users to account more a particular dimension by specifying weights for the dimensions. These weights can be used to calculate the (weighted) average of the privacy level. For instance, a user can select the composite service that requires the less sensitive data release by setting the weight for the first two components to 0.

V. Conclusions

We have presented a novel approach to assist users and Web service providers in the composition and selection of composite services that are more privacy preserving. With respect to other proposals for privacy-preserving Web service composition, our approach supports the specification of fine-grained privacy policies and preferences based on different privacy dimensions, i.e. purpose, visibility, retention period and sensitivity. In addition, our approach ranks the generated composite Web services with respect to their privacy level, which quantifies the risk of unauthorized disclosure of user information based on sensitivity, visibility and retention period.

As future work, we are planning to implement our approach in Java and to conduct an extensive evaluation. First, we will evaluate its performance with respect to the number of candidate Web services, the complexity of the privacy policies of the orchestrator and component services, and to the (re)delegation depth. Then, we will conduct a controlled experiment with master students in computer science to evaluate participants' *perceived easy of use*, *perceived usefulness*, and *intention to use* according to the Technology Acceptance Model (TAM) proposed by Davis in [33].

Acknowledgements This work is funded by EU-FP7-IST-NoE-NESSOS, EU-EIT-ICTLabs, EU-FP7-SEC-SECONOMICS, PAT-TRISE projects and by Dutch national program COMMIT through the THeCS project.

References

[1] M. Alrifai, T. Risse, and W. Nejdl, "A hybrid approach for efficient web service composition with end-to-end qos constraints," *TWEB*, vol. 6, no. 2, pp. 7:1–7:31, 2012.

[2] K.-M. Chao, M. Younas, C.-C. Lo, and T.-H. Tan, "Fuzzy matchmaking for web services," in *Proc. of AINA*. IEEE, 2005, pp. 721–726.

[3] B. Jeong, H. Cho, and C. Lee, "On the functional quality of service (fqos) to discover and compose interoperable web services," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5411–5418, 2009.

[4] V. X. Tran and H. Tsuji, "QoS Based Ranking for Web Services: Fuzzy Approaches," in *Proc. of NWeSP*, 2008, pp. 77–82.

[5] P. Wang, K.-M. Chao, C.-C. Lo, C.-L. Huang, and Y. Li, "A Fuzzy Model for Selection of QoS-Aware Web Services," in *Proc. of ICEBE*. IEEE, 2006, pp. 585–593.

[6] E. M. Maximilien and M. P. Singh, "Toward autonomic web services trust and selection," in *Proc. of SOC*. ACM, 2004, pp. 212–221.

[7] S. Paradesi, P. Doshi, and S. Swaika, "Integrating behavioral trust in web service compositions," in *Proc. of ICWS*. IEEE, 2009, pp. 453–460.

[8] P. Wang, K.-M. Chao, C.-C. Lo, R. Farmer, and P.-T. Kuo, "A reputation-based service selection scheme," in *Proc. of ICEBE*. IEEE, 2009, pp. 501–506.

[9] Z. Xu, P. Martin, W. Powley, and F. Zulkernine, "Reputation-Enhanced QoS-based Web Services Discovery," in *Proc. of ICWS*. IEEE, 2007, pp. 249–256.

[10] F. Massacci, J. Mylopoulos, and N. Zannone, "Hierarchical hippocratic databases with minimal disclosure for virtual organizations," *VLDB J.*, vol. 15, no. 4, pp. 370–387, 2006.

[11] A. Squicciarini, B. Carminati, and S. Karumanchi, "A privacy-preserving approach for web service selection and provisioning," in *Proc. of ICWS*. IEEE, 2011, pp. 33–40.

[12] S.-E. Tbahriti, M. Mrissa, B. Medjahed, C. Ghedira, M. Barhamgi, and J. Fayn, "Privacy-Aware DaaS Services Composition," in *Database and Expert Systems Applications*, ser. LNCS 6860. Springer, 2011, pp. 202–216.

[13] R. Hewett and P. Kijsanayothin, "Privacy and recovery in composite web service transactions," *International Journal for Infonomics*, vol. 3, no. 2, pp. 240–248, 2010.

[14] W. Xu, V. N. Venkatakrishnan, R. Sekar, and I. V. Ramakrishnan, "A framework for building privacy-conscious composite web services," in *Proc. of ICWS*. IEEE, 2006, pp. 655–662.

[15] OASIS, "Web Services Business Process Execution Language Version 2.0," OASIS Standard, 2007.

[16] H. Foster, S. Uchitel, J. Magee, and J. Kramer, "Ws-engineer: A model-based approach to engineering web service compositions and choreography," in *Test and Analysis of Web Services*. Springer, 2007, pp. 87–119.

[17] R. Hamadi and B. Benatallah, "A Petri net-based model for web service composition," in *Proc. of ADC*. Australian Computer Society, Inc., 2003, pp. 191–200.

[18] X. Fu, T. Bultan, and J. Su, "Formal verification of e-services and workflows," in *Web Services, E-Business, and the Semantic Web*, ser. LNCS 2512. Springer, 2002, pp. 188–202.

[19] D. Berardi, G. D. Giacomo, M. Lenzerini, M. Mecella, and D. Calvanese, "Synthesis of underspecified composite *e*-services based on automated reasoning," in *Proc. of SOC*. ACM, 2004, pp. 105–114.

[20] A. Mahfouz, L. Barroca, R. C. Laney, and B. Nuseibeh, "Requirements-driven collaborative choreography customization," in *Proc. of ICSOC*, ser. LNCS 5900. Springer, 2009, pp. 144–158.

[21] M. P. Singh, A. K. Chopra, and N. Desai, "Commitment-based service-oriented architecture," *IEEE Computer*, vol. 42, no. 11, pp. 72–79, 2009.

[22] L. Cranor, M. Langheinrich, M. Marchiori, and J. Reagle, "The Platform for Privacy Preferences 1.0 (P3P1.0) Specification," W3C Recommendation, Apr. 2002. [Online]. Available: http://www.w3.org/TR/P3P/

[23] A. Tumer, A. Dogac, and I. H. Toroslu, "A semantic-based user privacy protection framework for web services," in *Proc. of ITWP*, ser. LNCS 3169. Springer, 2005, pp. 289–305.

[24] A. Nyre, K. Bernsmed, S. Bo, and S. Pedersen, "A server-side approach to privacy policy matching," in *Proc. of ARES*, 2011, pp. 609 –614.

[25] L. Cranor, M. Langheinrich, M. Marchiori, and J. Reagle, "A P3P Preference Exchange Language 1.0 (APPEL1.0)," W3C Recommendation, Apr. 2002. [Online]. Available: http://www.w3.org/TR/P3P-preferences/

[26] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "XPref: a preference language for P3P," *Computer Networks*, vol. 48, no. 5, pp. 809–827, 2005.

[27] M. Banerjee, R. K. Adl, L. Wu, and K. Barker, "Quantifying privacy violations," in *Secure Data Management*, ser. LNCS 6933. Springer, 2011, pp. 1–17.

[28] J. S. Hammond, R. L. Keeney, and H. Raiffa, *Smart choices : a practical guide to making better life decisions*. Broadway Books, 2002.

[29] L. Liu, N. Mehandjiev, and D.-L. Xu, "Multi-criteria service recommendation based on user criteria preferences," in *Proc. of RecSys*. ACM, 2011, pp. 77–84.

[30] F. Massacci, J. Mylopoulos, and N. Zannone, "Security Requirements Engineering: The SI* Modeling Language and the Secure Tropos Methodology," in *Advances in Intelligent Information Systems*, ser. Studies in Computational Intelligence. Springer, 2010, vol. 265, pp. 147–174.

[31] P. Guarda and N. Zannone, "Towards the Development of Privacy-Aware Systems," *Information and Software Technology*, vol. 51, no. 2, pp. 337–350, 2009.

[32] T. Saaty, "How to make a decision: The Analytic Hierarchy Process," *EJOR*, vol. 48, pp. 9–26, 1990.

[33] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Q.*, vol. 13, no. 3, pp. 319–340, Sep. 1989.