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 illustrate our approach using an eCommerce Web service as an example of service composition. Moreover, we present a possible Java-based implementation of the proposed approach and present an extension to WS-Policy standard to specify privacy related assertions.*

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 (Alrifai, Risse, & Nejdl, 2012; Chao & Younas, 2005; Hammond, Keeney, & Raiffa, 2002; Jeong, Cho, & Lee, 2009; Tran & Tsuji, 2008; Wang, Chao, Lo, Huang, & Li, 2006) or on their trust and reputation level (Maximilien & Singh, 2004; Paradesi, Doshi, & Swaika, 2009; Wang, Chao, Lo, Farmer, & Kuo, 2009; Z. Xu, Martin, Powley, & Zulkernine, 2007). To the best of our knowledge, only few works have investigated privacy issues in service selection (Massacci, Mylopoulos, & Zannone, 2006; Squicciarini, Carminati, & Karumanchi, 2011) and composition (Hewett & Kijsanayothin, 2010; Tbahriti et al., 2011; W. Xu, Venkatakrishnan, Sekar, & Ramakrishnan, 2006). 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 identifies 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: 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 an eCommerce Web service as a running example. We also present a possible implementation of our approach as a Web service that does not require extending existing UDDI registries implementations. We also propose an extension to WS-Policy standard to allow service consumers and Web service providers to specify their privacy preferences and privacy policies by means of privacy assertions.

The remainder of the paper is structured as follows. In the next section we discuss related works. Then, we present a modeling framework for representing service orchestrations and users' privacy preferences and Web service providers' privacy policies. After that, we introduce the privacy-aware service composition and selection approach and a possible implementation of it in Java. We conclude the paper providing directions for evaluating the approach.

RELATED WORK

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

Service composition modeling: To model service composition and verify whether it satisfies safety and liveness properties, several languages, such as WS-BPEL (OASIS, 2007), or approaches, such as process algebra (Foster, Uchitel, Magee, & Kramer, 2007), Petri nets (Hamadi & Benatallah, 2003), model checking (Fu, Bultan, & Su, 2002), and finite state machines (Berardi, De Giacomo, Lenzerini, Mecella, & Calvanese, 2004) have been proposed. These approaches, however, either specify service orchestration architectures at a low level, which makes it difficult to analyze the overall inter-organizational relationships, or do not consider the autonomous and heterogeneity nature of services.

Contributions to service composition modeling also come from the requirement engineering community. In particular, goal-oriented approaches (Mahfouz, Barroca, Laney, & Nuseibeh, 2009; Singh, Chopra, & Desai, 2009) use goal models to represent the strategic business goals that drive the interactions in an orchestration of services, and the business goals of each participant in the 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. Similarly to (Mahfouz et al., 2009; Singh et al., 2009), we adopt a goal-oriented approach to model the functionalities offered by the orchestrator, the component services, and composite service. In particular, we use AND/OR trees where the semantics of nodes and arcs is based on the concepts defined by SI* (Massacci, Mylopoulos, & Zannone, 2010), a goal-oriented framework for security requirements elicitation and analysis.

Service composition: Service composition is the problem of aggregating services in such a way that given (functional and not functional) requirements are satisfied. Among approaches for secure service composition, it is worth mentioning the ones in (Carminati, Ferrari, & K. Hung, 2006; Paradesi et al., 2009). Carminati et al. (Carminati et al., 2006) present an approach to build a service composition where component services satisfy the security constraints imposed by service requestors and providers. Our approach, instead, builds a composition of services where the component services satisfy the privacy policy of the orchestrator and the privacy preferences of customers. Paradesi et al.(Paradesi et al., 2009) propose Wisp, an approach to Web service composition that computes the aggregated trust in the composition, and selects the composition that is most trustworthy. In addition, they present a method for comparing the different Web service compositions based on their trustworthiness. In our approach, instead, we compute an aggregated privacy level for each admissible service composition, and we select among all the admissible service compositions the one that is most privacy preserving.

The role of privacy in service composition has been investigated in (Hewett & Kijsanayothin, 2010; Tbahriti et al., 2011; W. Xu et al., 2006), where only services requiring the disclosure of less sensitive information and offered by trusted providers are selected in the composition. Tbahriti et al. (Tbahriti et al., 2011) propose a privacy-preserving composition approach to check compatibility among privacy requirements and policies within a composition. The compatibility matching is based on the notion of privacy subsumption. Hewett et al. (Hewett & Kijsanayothin, 2010) present a privacy-aware approach to service composition, where only services that require the disclosure of less sensitive information and that are offered by providers trusted by the customer are selected in the composition. Xu et al. (W. Xu et al., 2006) propose a framework that addresses customers' privacy concerns in the context of highly customizable composite web services. The framework provides automated techniques for checking at the customer site whether the privacy policies of the providers participating to the composition are compliant with a customer's preferences. The main difference between these proposals and our work is that they only identify a set of admissible service composition and do not select an optimal service composition with respect to the privacy preferences of the user like we do in our work.

Users' privacy concerns are often addressed by providing automated techniques for matching provider's privacy policies with customer's preferences (Cranor, Langheinrich, Marchiori, & Reagle, 2002a; Nyre, Bernsmed, Bo, & Pedersen, 2011; Tumer, Dogac, & Toroslu, 2005). The most prominent solution for policy matching is P3P (Platform for Privacy Preferences Project) (Cranor et al., 2002a). 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 (Cranor, Langheinrich, Marchiori, & Reagle, 2002b) and XPref (R Agrawal, Kiernan, Srikant, & Xu, 2005). In (Tumer et al., 2005) 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.

Service selection: Service composition might return a set of admissible services; thus, service ranking is needed to choose the best composition. QoS-based (Alrifai et al., 2012; Chao & Younas, 2005; Jeong et al., 2009; Tran & Tsuji, 2008; Wang et al., 2006), and trust-based (Maximilien & Singh, 2004; Paradesi et al., 2009; Wang et al., 2009; Z. Xu et al., 2007) service selection has been widely investigated in the literature. Privacy-aware service selection is addressed in (Squicciarini et

al., 2011). This work 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. In (Massacci et al., 2006)it is presented 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 (Banerjee, Karimi Adl, Wu, & Barker, 2011). 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 (Hammond et al., 2002) or collaborative filtering (Liu, Mehandjiev, & Xu, 2011) 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 automated, or are not applicable due to the nature of privacy criteria.

MODELING SERVICE COMPOSITION

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* (Massacci et al., 2010), 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 $A \cap T = \varnothing$. We use notation $s \triangleright t$ to indicate that a service $s \in A$ is of type $t \in T$. *Goals* represent the functionalities offered byservices, 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*.

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. eCommerce is a composite Web service which provides its customers with the possibility of login, search for items of interest, add them to their cart and pay. To this end, eCommerce relies on four types of service that are dynamically selected: eStore which provides the login and searching functionalities, Identity Provider which enables the user to login with one of her existing identities, Virtual Cart which provides the functionalities to create a cart, add items to the cart and checkout, and Point-of-Sale (PoS) which offers the functionalities to pay an order placed in the cart. Figure 1 illustrates the orchestrator model of eCommerce, while Figure 2 shows the list of symbols used through our examples and their meaning. eCommerce provides goal Sell

Goods, represented by the top oval. This goal is decomposed into sub-goals Login, Search, Manage Cart and Pay. Figure 3 and Figure 4 illustrate examples of component services of types Virtual Cart and Point-of-Sale.

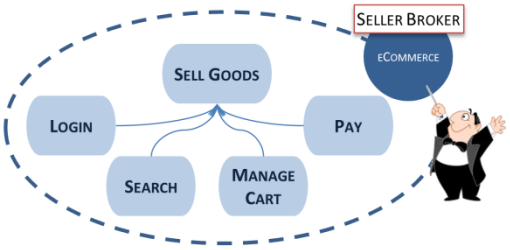

Figure 1: Orchestrator Model

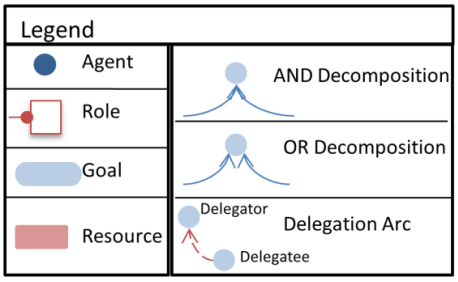

Figure 2: Legend

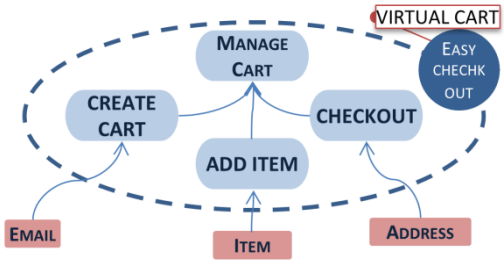

Figure 3: Component Service Model with AND Decomposition

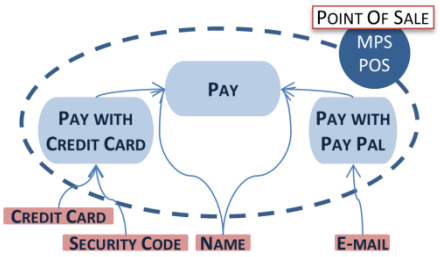

Figure 4: Component Service Model with OR Decomposition

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_1andS_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 re-delegate the achievement of (part of) the goal to another component service.

Example 2. Figure 5 shows the orchestration model obtained by merging the service model of eCommerce 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 eCommerce. Goal Pay can be provided by two point-of-sale services, MPS Point-of-Sale and Easy Point-of-Sale. Goals Login and Search can be fulfilled by eShop and ShopEasy. eShop delegates GroupBuy for searching for good deals. Goal Manage Cart is delegated to virtual cart services EasyCheckout and to FastCheckout. Goal Login can also be achieved using the identity provider MyIDP. The partner services may require different information to fulfill the goal they provide. For example, to fulfill goal Pay, MPS Point-of-Sale requires its customers to provide name, credit card and security code if goal Pay with Credit Card is chosen, or only email and name if goal Pay with PayPal is chosen. On the other side, Easy Point-of-Sale always requires name, credit card and address.

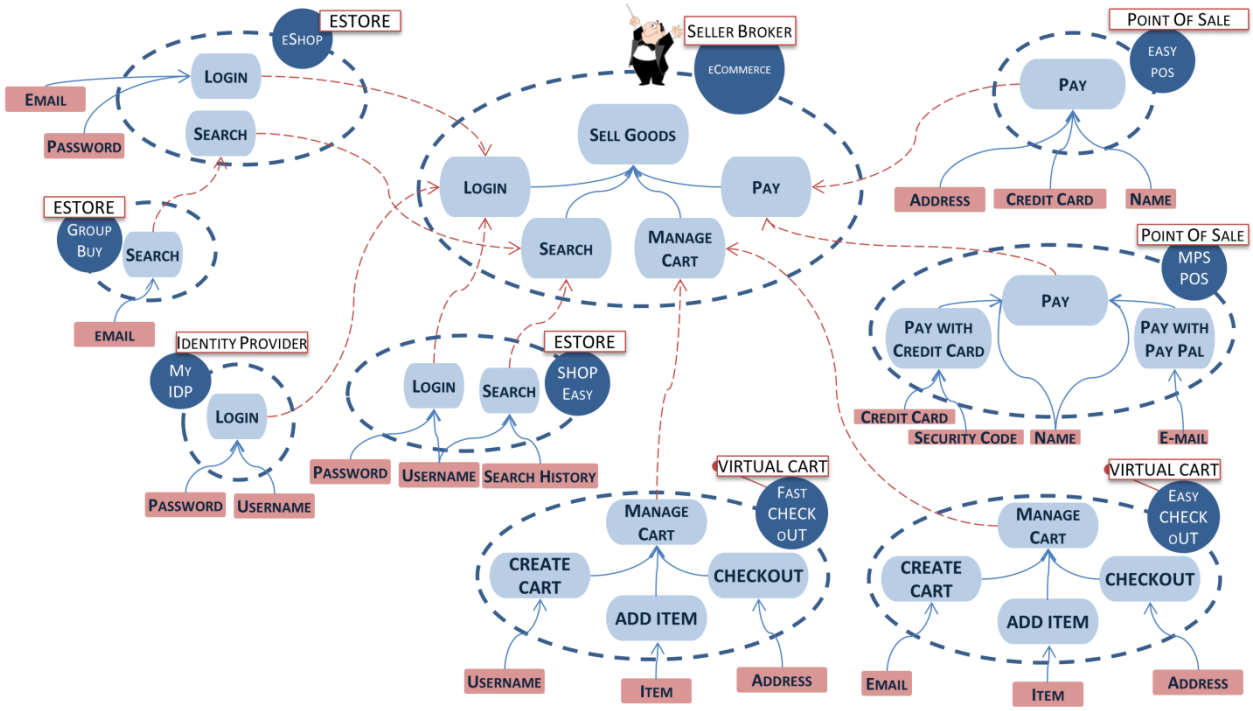

Figure 5: Example of Orchestration Model

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 $D_{Z,g0}$ such that $Z \subset V \cap R$ is a set of data items and g_0 is the root goal of S.

Example 3. Figure 6 shows a possible composite service within the orchestration model of Figure 5 where eCommerce's goals, Login, Search, Manage Cart and Pay are fulfilled by the services MyID, eShop (which delegates to GroupBuy), FastCheckout, and MPS Point-of-Sale, respectively. In turn, eShop delegates the fulfillment of Search to GroupBuy service, which offers deals to its customers.

MODELING PRIVACY

Nowadays, privacy protection holds the balance of power between business success and failure. This reflects the growing attention that users pay with whom their share their personal data and the increasing number of privacy regulations that are intended to safeguard citizens' right to privacy. Therefore, privacy is becoming a fundamental factor in the selection of service orchestrations. In this section, we define the privacy dimensions which are used to capture users' privacy preferences and services' privacy policies together with a language for their specification.

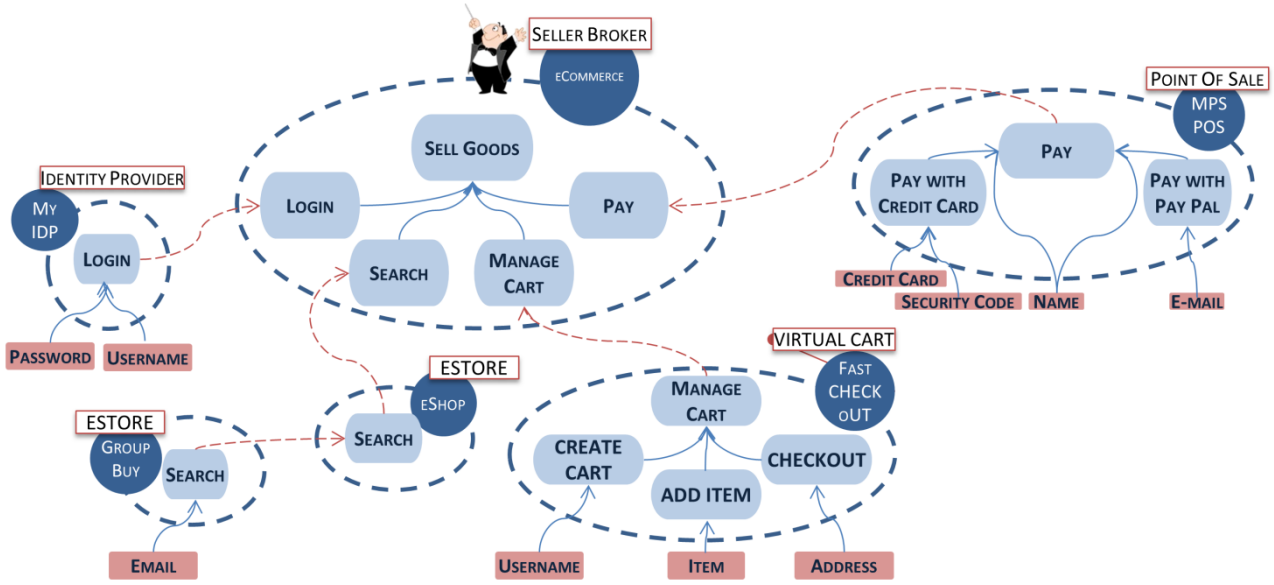

Figure 6: Example of Composite Service

Privacy Requirements

Many countries have promulgated data protection regulations (e.g., EU Directive 95/46/EC, HIPAA) establishing the rights and obligations of the actors involved in a data processing (Guarda & Zannone, 2009). Three main actors can be identified in these regulations, namely *data subject*, *data controller* and *data processor*. The data subject is defined as an identifiable person to whom the personal data refer. The data controller is the natural or legal person that determines the purposes and means of the data processing, and is responsible for the usage of collected data. The data processor is defined as a natural or legal person that processes personal data on behalf of the controller. The relationship between the controller and the processor must be governed by legal agreements binding both parties to ensure that the processor complies with the privacy practices defined by the controller and that the processor is bound by the same obligations as the controller.

Privacy regulations impose stringent requirements on the collection, processing and disclosure of personal data. These requirements include *fair and lawful processing*, *consent*, *purpose specification*, *minimality*, *minimal disclosure*, and *sensitivity* (Rajeev Agrawal, Kiernan, Srikant, & Xu, 2002; Guarda & Zannone, 2009). Fair and lawful processing states that data collection and processing should neither unreasonably intrude upon data subjects' privacy nor unreasonably interfere with their autonomy. Consent requires that personal data are collected and processed only if the data subject has given her explicit consent. Purpose specification states that personal data should be collected for specified and legitimate purposes and not processed in ways that are incompatible with the purposes for which data have been collected. Minimality imposes that data collection and processing is limited to the minimum necessary for achieving the specific purpose. This includes that personal data should be retained only for the time necessary to achieve the specific purpose. Minimal disclosure restricts the disclosure of personal data to third parties upon certain conditions.Finally, sensitivity states that the processing of personal data, which are particularly sensitive for the data subject, should be subject to more stringent protection measures.

Several data privacy taxonomies have been proposed to identify and define the privacy dimensions necessary to capture legal requirements (Agrawal et al., 2005; Barker et al., 2009; Cranor et al., 2002a; Guarda & Zannone, 2009). Here, we consider four privacy dimensions which are typically used to specify service providers' privacy policies and users' 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.

In the next sections we will show how these dimensions can be used for the specification of privacy policy and user preferences.

Privacy Policies and Preferences

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 these user concerns. On one hand, data protection legislation recognizes the right of users to control their data (Guarda & Zannone, 2009). 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 three privacy dimensions which are typically used to specify privacy: *purpose*, *visibility*, and *retention period*. Accordingly, privacy policies can be formally defined as follows.

Definition 4 (Privacy Policy). A privacy policy is a set of tuples $\langle d, p, v, \delta, \tau \rangle$ where:

- $d \in R$ denotes a data item;
- $p \in G$ is the purpose for which d can be collected;
- $v \subset \mathrm{A} \cup \mathrm{T}$ 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.

Example 4. The privacy policy of eCommerce is presented in Table 1. The policy specifies how eCommerce will use customers' data. For example, eCommerce will collect a customer's Name to fulfill purpose Login and it will maintain a copy of the data item for 36 months. Moreover, the policy states that eCommerce 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 Login and has different rules for different agents: if the component service is an instance of eStore 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 Virtual Cart, the National ID cannot be delegated further, and can be kept only for 12 months.

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 1 can be graphically represented using a model similar to Figure 1. The main difference between the two models is that nodes are annotated with visibility, (re)delegation depth and retentionperiod. Formally, a *privacy policy model* is a tuple $\langle \mathrm{V}, \mathrm{E}, \Gamma \rangle$ where $\langle \mathrm{V}, \mathrm{E} \rangle$ is the corresponding service model and Γ is theprivacy policy in tabular form. Given a privacy policy Γ and a purpose p, Γ^p =

$\{\langle d, v, \delta, \tau\rangle | \langle d, p, v, \delta, \tau \rangle \in \Gamma\}$. We say that a privacy policy Γ is *well-defined* if (i) $\forall \langle$d, p, v, δ, τ $\rangle \in \Gamma$ $v \neq \emptyset$ iff $\delta > 1$ and (ii) for every data item d and purpose p such that $\langle d, v, \delta, \tau\rangle \in \Gamma^p$, $\nexists \langle d', v', \delta', \tau' \rangle \in \Gamma^{p'}$ 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. Hereafter, we only consider well-defined privacy policies.

Table 1: eCommerce Privacy Policy

Data Item (d)	Purpose (p)	Visibility (v)	Depth (δ)	Retention (τ)
Username	Login	All	*	36
	Manage Cart	All	*	24
	Search	All	*	24
Password	Login	eStore, Identity Provider	2	24
Name	Pay	Point-of-Sale	*	24
Email	Login	All	*	24
	Manage Cart	All	*	24
	Pay	Point-of-Sale	*	36
	Search	All	*	36
Credit Card	Pay	Point-of-Sale	2	12
Security Code	Pay	Point-of-Sale	2	12
Search History	Search	All	*	36
Address	Manage Cart	Virtual Cart	3	18
	Pay	Point-of-Sale	3	12
Item	Manage Cart	All	*	36

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. We say that Γ_y complies with Γ_x, denoted as $\Gamma_y \rightsquigarrow \Gamma_x$, if $\forall p\ \forall \langle d_1, v_1, \delta_1, \tau_1\rangle \in \Gamma_y^p \exists \langle d_2, v_2, \delta_2, \tau 2_1\rangle \in \Gamma_x^p$ such that (i) $d_1 = d_2$; (ii) $\delta_1 < \delta_2$; (iii) $\tau_1 \leq \tau_2$.

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 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 6 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 $\langle d, p, \sigma, v, \delta, \tau \rangle$ 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;
- $v \subset A \cup T$ is the visibility of d for achieving p;
- $\delta \in \mathbb{N} \cup \{*\}$ is the (re)delegation depth which limits the sharing of d for achieving p;
- $\tau \in \mathfrak{R}$ is the retention period of d.

Table 2:Bob's privacy preferences

Data Item (d)	Sensitivity (σ)	Purpose (p)	Visibility (v)	Depth (δ)	Retention (τ)
Username	5	Login	All	*	36
	5	Manage Cart	All	*	24
	5	Search	All	*	24
Password	10	Login	Identity Provider	2	24
Name	5	Pay	Point-of-Sale	*	24
Email	5	Login	All	*	24
	5	Manage Cart	All	*	24
	5	Pay	All	*	36
	5	Search	All	*	36
Credit Card	10	Pay	MPS Point-of-Sale	2	12
Security Code	10	Pay	MPS Point-of-Sale	2	12
Search History	7	Search	All	*	36
Address	6	Manage Cart	Fast CheckOut	3	18
Item	6	Manage Cart	Fast CheckOut	3	36

Example 5. Let Bob be a new customer of eCommerce. Based on the privacy policy of eCommerce (Table 1), he specifies constraints on the collection and processing of his data. Bob's privacy preferences are presented in Table 2. Since Username and Email are usually required by service providers, Bob leaves their visibility to all. In contrast, he prefers that his Password is only given to an Identity Provider and that his real Name is only used for paying purposes. Bob also prefers that his Credit Card and Security Code are only disclosed to an agent he trusts, i.e. MPS Point-of-Sale and only for Pay purpose. Finally, Bob prefers to disclose Search History only to eStore and the Item he buys together with the Address where to ship, only to Fast CheckOut.

PRIVACY AWARE SERVICE SELECTION

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.

Algorithm 1: Service Composition

Input: S_u set of functionalities requested by user u,
$P_o = < V_o, E_o, \Gamma_o >$ 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 $P = < V, E, \Gamma >$;
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. $E = E \cup \{< S_u, root >\}$;
10. **while** Q is not empty **do**
11. extract s_i from Q;
12. **if** $s_i$not leaf node **then**
13. **for** $< Z, s_i > \in E_o$ **do**
14. $V = V \cup Z$;
15. $E = E \cup \{< Z, s_i >\}$;
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 $P_x = < V_x, E_x, \Gamma_x >$ be the policy model s.t. $s_i \in V_x$;
24. **if** s_i not leaf node **then**
25. **for** $< Z, s_i > \in E_x$ **do**
26. **if** $\Gamma_x^Z \rightsquigarrow \Gamma^{s_k}$ **then**
27. $V = V \cup Z$;
28. $E = E \cup \{< Z, s_i >\}$;
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. **elseif** s_i is a purpose node **then**
33. let $W = \{w | < d, \upsilon, \delta, \tau > \in \Gamma_x^{s_i} \wedge ((w \in \upsilon \cap A) \vee (w \rhd t \wedge t \in \upsilon \cap T))\}$;
34. **for** $w \in W$ **do**
35. let $P_w = < V_w, E_w, \Gamma_w >$ 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. $E = E \cup \{< \{s_j\}, s_i >\}$;
40. $\Gamma^{s_j} = \Gamma_w^{s_j}$;
41. insert (s_i, s_j) in S;

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 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 (line 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 with respect to 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 discarded 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. Figure 7 shows the composite services selected by the algorithm. The services that appear obfuscated in the figure have been discarded because their policies did not comply with Bob's preferences. Figure 8 shows the orchestration policy model based on Bob's privacy preferences (Table 2) together with the policies of the selected component services. The model describes four admissible composite services that can be employed to provide the functionalities requested by Bob (see Figure 10 for their description). Note that, for readability reasons, we have omitted the visibility field in the figure.

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.

PRIVACY POLICY MODEL FOR THE USER (BOB)

Γ_1^{login}

Data Item (d)	visibility	δ	τ
Username	All	*	36
Email	All	*	24
Password	Identity Provider	2	24

Γ_2^{search}

Data Item (d)	visibility	δ	τ
Username	All	*	24
Search History	All	*	36
Email	All	*	36

$\Gamma_3^{manage\ cart}$

Data Item (d)	visibility	δ	τ
Username	All	*	24
Email	All	*	24
Address	Fast Check Out	3	18
Item	Fast Check Out	3	36

Γ_4^{pay}

Data Item (d)	visibility	δ	τ
Name	Point-of-Sale	*	24
Email	All	*	36
Credit Card	MPS Point-of-Sale	2	12
Security Code	MPS Point-of-Sale	2	12

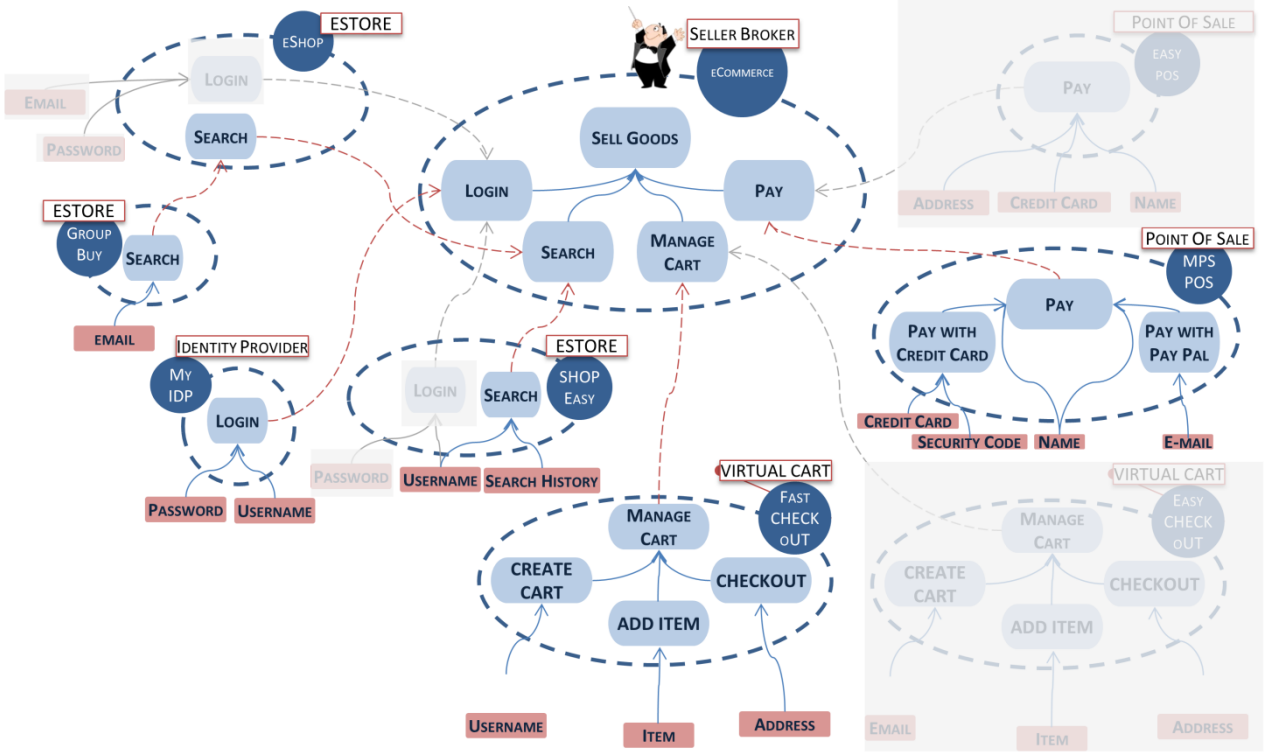

Figure 7: Service Composition

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.

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

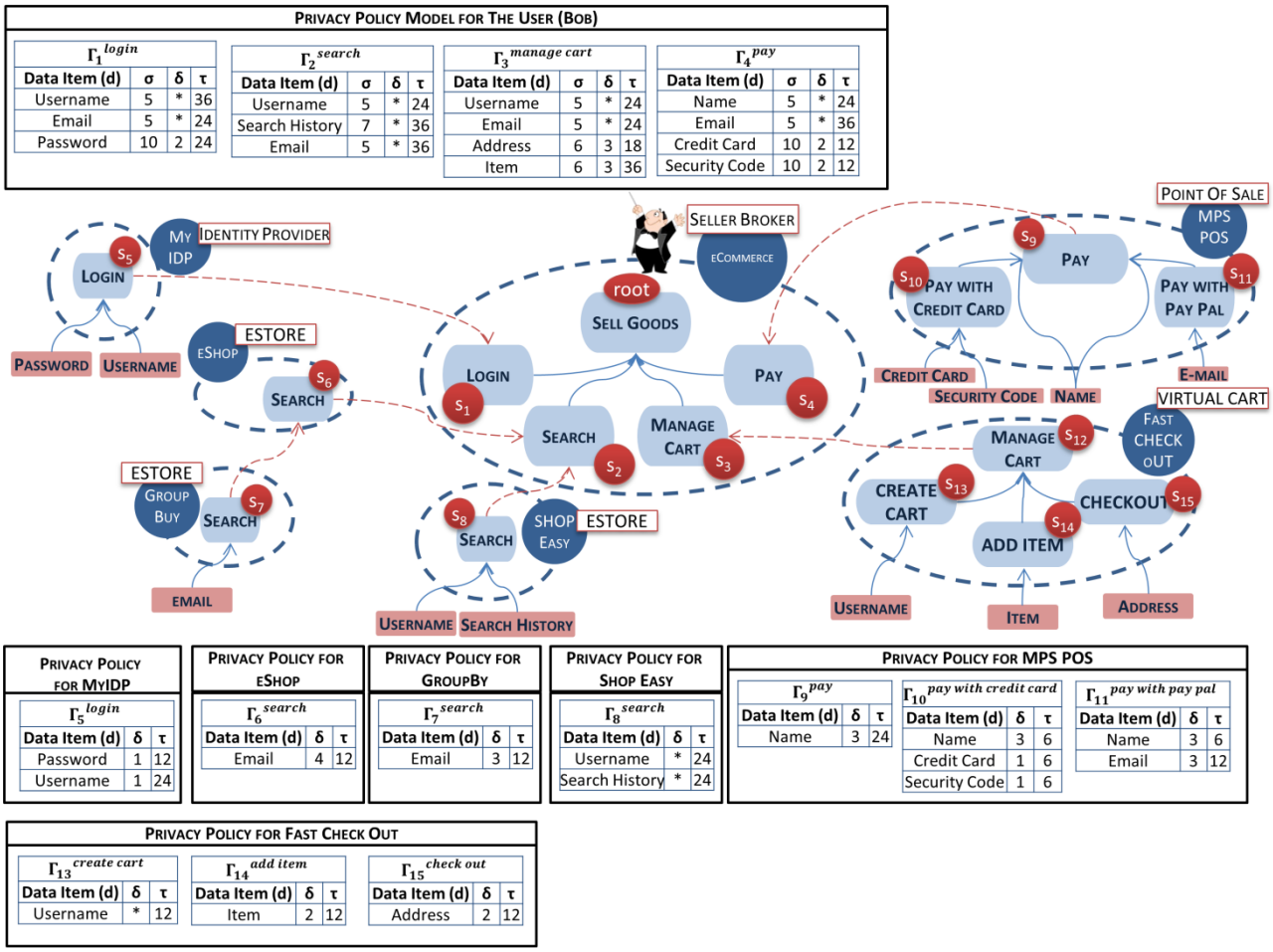

Figure 8: Orchestration Policy Model

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 $\bar{\Gamma} = \{\langle d, p, v, \delta, \tau \rangle | \langle d, p, v, \delta, \tau \rangle \in \Gamma \cap \Gamma_i\}$. The privacy level of the composite service is a vector $[\delta, \tau, \sigma]$ such that

- $\delta = avg(\sigma_j \delta_i | \langle d, p, v, \delta_i, \tau_i \rangle \in \bar{\Gamma} \wedge \langle d, p, \sigma_j, v, \delta_j, \tau_j \rangle \in \Pi_1)$
- $\tau = avg(\sigma_j \tau_i | \langle d, p, v, \delta_i, \tau_i \rangle \in \bar{\Gamma} \wedge \langle d, p, \sigma_j, v, \delta_j, \tau_j \rangle \in \Pi_1)$
- $\sigma = \sum_{\langle d,pi,vi,\delta i,\tau i \rangle \in \bar{\Gamma}} \sigma_j$ s.t. $\langle d, p_j, \sigma_j, v_j, \delta_j, \tau_j \rangle \in \Pi \wedge v_i \subset v_j \wedge (p_i = p_j \vee (\exists \langle p_j, Z \rangle \in E \text{ s.t. } p_i \in Z))$

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 Γ 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 $\delta_{max}, \tau_{max}, \sigma_{max}$ 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 (Liu et al., 2011) and, thus, the selection process cannot be automated.

Decision making should be simple and intuitive as well easy to review (Davis, 1989). 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 admissible component services, i.e. $min(\|\overline{\omega_i}\| \mid \omega_i \in \Omega^S)$.

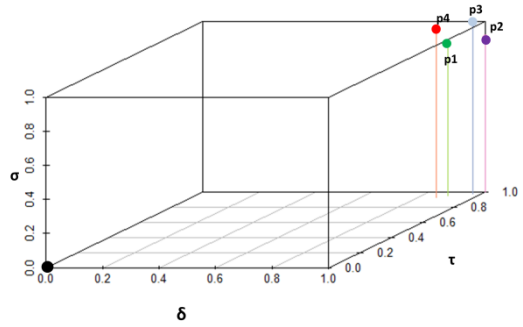

Figure 9: Privacy-preserving Composite Service Ranking - Graph Representation

CS	Description	δ	τ	σ	Norm
P1	$S_1, S_5, S_2, S_6, S_7, S_3, S_{12}, S_{13}, S_{14}, S_{15}, S_4, S_9, S_{10}$	0,8663	0,9270	0,9873	1,6077
P2	$S_1, S_5, S_2, S_8, S_3, S_{12}, S_{13}, S_{14}, S_{15}, S_4, S_9, S_{10}$	0,9659	0,9784	1,0000	1,7000
P3	$S_1, S_5, S_2, S_8, S_3, S_{12}, S_{13}, S_{14}, S_{15}, S_4, S_9, S_{11}$	1,0000	1,0000	0,9051	1,6790
P4	$S_1, S_5, S_2, S_6, S_7, S_3, S_{12}, S_{13}, S_{14}, S_{15}, S_4, S_9, S_{11}$	0,8963	0,9465	0,8924	1,5797

Figure 10: Privacy-preserving Composite Service Ranking: Admissible Composite Services

Example 7. Each admissible composite service (CS) in Figure 8 is represented as a 3D-point in Figure 9. The dimensions δ, v and σ as well as the norm for each composite service are presented in Figure 10. The height of a point represents its aggregated sensitivity, whereas the most right points are those with a higher depth, 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_4 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.

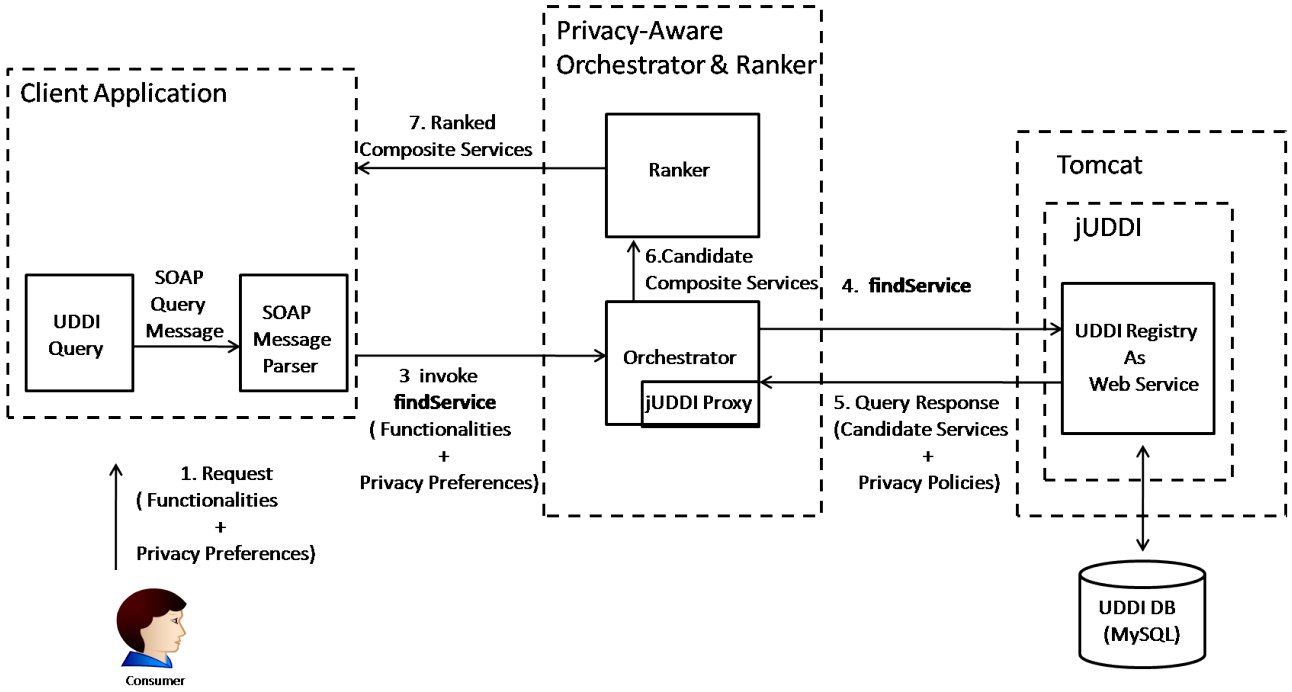

Figure 11: Prototype Implementation

SYSTEM ARCHITECTURE

In this section we describe a possible implementation in Java of our approach to privacy-aware Web service composition and ranking. As shown in Figure 11, our prototype consists of four main components: *Client Application*, *Privacy-Aware Orchestrator and Ranker*, *UDDI Registry*, and *UDDI Database*. The Client Application provides the service consumers with a user-friendly interface to specify the functionalities requested and their privacy preferences on how their data should be managed by the Web service providers. The Privacy-Aware Orchestrator and Ranker component provide two main functionalities: a) query the UDDI Registry to select Web services that match users' functional and privacy requirements for the composition and compose them according to schema; b) prioritize the admissible composite services based on their privacy level. The UDDI Registry offers the functionalities to describe and discover businesses, organizations, and other Web services providers, the Web services they make available, and the technical interfaces which may be used to access those services. All the information about Web service providers, the services they offer and their interfaces is stored in the UDDI Database.

To implement the prototype we have extended the current query capabilities of existing UDDI registries without the need of changing the UDDI registries data structure and API. The implementation is built on top of Java and Apache's jUDDI, which is an open source Java-based implementation of the UDDI standard. Using jUDDI, the UDDI Registry is a Web service implemented in Java running under Apache Tomcat application server. The Privacy-Aware Orchestrator and Ranker component acts as a proxy between the Client Application and the UDDI Registry. The component is implemented as a Web service which wraps the jUDDI client proxy. The jUDDI client proxy is an API that allows the Privacy-Aware Orchestrator and Ranker to query the UDDI Registry. The Privacy-Aware Orchestrator and Ranker component delegates the handling of standard UDDI queries to the UDDI Registry. In turn, the UDDI Registry processes the results of the query locally to determine all the possible Web service compositions for which service providers' privacy policies satisfy service consumers' privacy preferences. The Privacy-Aware Orchestrator and Ranker provides an interface with operation findService, which extends the traditional findService operation of UDDI standard with additional attributes like the privacy preferences of the service consumer.

When the service consumer places a request, a SOAP message containing the query to the UDDI Registry is generated by the Client Application. The SOAP message is analyzed by a SOAP message parser which is part of the Client Application: if the SOAP message contains a standard UDDI query, the findService operation of the UDDI Registry is invoked. Otherwise, the Privacy-Aware Orchestrator and Ranker component invokes findService which returns the Web services which match the functionalities requested by the service consumer along with the information necessary to invoke the services and to retrieve the privacy policy of the service provider. To allow the Privacy-Aware Orchestrator and Ranker generating privacy-preserving Web service compositions and to rank them, we assume that the privacy preferences and the service providers' privacy policies are expressed in WS-Policy which has been extended to support the specification of privacy-specific assertions. We introduce the extensions to WS-Policy in the next subsection.

Extending WS-POLICY to Specify Privacy Preferences and Policies

WS-Policy (Vedamuthu et al., 2007) is a W3C standard which allows service consumers and Web service providers to express their requirements and capabilities with respect to their interactions and agree on a mutual acceptable policy before interacting. WS-Policy policies are specified as a set of assertions that express capabilities or requirements of the policy subject. A WS-Policy policy can be associated either with a WSDL document entity (e.g., service, endpoint) or with a UDDI element.

Currently, WS-Policy supports assertions related to message integrity and confidentiality (e.g., which parts of a SOAP message need to be encrypted or signed), authentication and the use of algorithms. However, it does not allow the specification of privacy related assertions. Thus, to specify service consumers' privacy preferences and service providers' privacy policies, we have extended the WS-Policy standards by introducing new types of assertions. These assertions are listed in Table 3.

Table 3 WS-Policy Extension for Privacy

PRIVACY ASSERTION	DESCRIPTION
`<PrivacyAssertion>`	Assertion representing a privacy preference
`<Purpose>`	Purpose for which service consumer personal information can be collected
`<ProtectedParts>`	Service consumer personal information
`<Sensitivity>`	Sensitivity of personal information
`<AuthorizedEntities>`	Visibility of the personal information
`<RetentionPeriod>`	Retention period for personal information
`<Delegation>`	Re-delegation deph

`<PrivacyAssertion>` is the main container. `<Purpose>` denotes the purpose for which service consumer personal information can be collected. `<ProtectedParts>` specifies by means of XPath expressions the parts of the SOAP message carrying the service consumer personal information that need to be protected from unauthorized disclosure; `<Sensitivity>` specifies the sensitivity level of the parts of the SOAP message listed in `<ProtectedParts>` according to the service consumer. `<AuthorizedEntities>` specifies the URI of Web service entities that are entitled to access the service consumer personal information; `<RetentionPeriod>` is the maximum time for which the Web service provider can keep a copy of the service consumer personal information. `<Delegation>` denotes the delegation depth which limits the sharing of the service consumer

personal information with other Web service entities for achieving the purpose specified by <Purpose>.

We now illustrate how privacy assertions can be used in practice. We show how Bob's privacy preferences and eCommerce privacy policy can be represented using the privacy- related assertions.

```
<wsp:Policy xmlns:wsp="http://www.w3.org/ns/ws-policy" >
   <wsp:All>
     <PrivacyAssertion>
           <ProtectedParts>
              <sp:XPath>//Body/CreditCard</sp:XPath>
         </ProtectedParts>
           <Purpose>Manage Cart</Purpose>
           <Sensitivity>10</Sensitivity>
         <AuthorizedEntities>
              <AuthorizedEntity>
                   http://example.com/EasyCheckout
              </AuthorizedEntity>
            <AuthorizedEntities>
        <RetentionPeriod>12</RetentionPeriod>
        <Delegation>3</Delegation>
    </PrivacyAssertion>
   <wsp:All>

</wsp:Policy>
```

Figure 12: Bob's Privacy Preferences in WS-Policy

As shown in Example 5, Bob is willing to share his Credit Card with the trusted service EasyCheckout for fulfilling purpose Manage Cart. This preference is represented by <PrivacyAssertion> element shown in Figure 12. <PrivacyAssertion> has several sub-elements representing the different components of a privacy preference as specified in Definition 6. <ProtectedParts> includes an XPath expression that indicates the XML element CreditCard contained in the body of the SOAP message sent to invoke the operations offered by eCommerce Web service. The value of element <Purpose> denotes that CreditCard can be disclosed to fulfill Manage Cart purpose. The value of element <Sensitivity> specifies how sensitive CreditCard is for Bob on a scale from 1 to 10. <AuthorizedEntities> has a subelement <AuthorizedEntity> for each Web service entity that is entitled to access CreditCard. In this particular case, it has only one sub-element whose value denotes the EasyCheckout service. The value of element <RetentionPeriod> specifies that a copy of CreditCard can only be kept for 12 months. Finally, the value of element <Delegation> denotes that the re-delegation path has 3 as maximum length.

Figure 13 shows eCommerce's privacy policy expressed using the proposed privacy-related assertions. We focus on the privacy statements related to PhoneNumber. The statement is represented by <PrivacyAssertion>. <ProtectedParts> includes an XPath expression that

indicates the XML element PhoneNumber contained in the body of the SOAP message sent to invoke the operations offered by eCommerce Web service. The value of element <Purpose> denotes that PhoneNumber can be disclosed to fulfill Manage Cart purpose. <AuthorizedEntities> has no <AuthorizedEntity> sub-elements to denote that eCommerce can give a copy of PhoneNumber to any Web service entity. <RetentionPeriod> element value specifies that a copy of PhoneNumber will be kept for 36 months. <Delegation> element value denotes that the eCommerce can further share Phone Number without limitations.

```
<wsp:Policyxmlns:wsp="http://www.w3.org/ns/ws-policy" >
  <wsp:All>
        <PrivacyAssertion>
            <ProtectedParts>
             <sp:XPath>//Body/PhoneNumber</sp:XPath>
         </ProtectedParts>
         <Purpose>ManageCart</Purpose>
            <AuthorizedEntities/>
         <RetentionPeriod>36</RetentionPeriod>
         <Delegation>Unlimited</Delegation>
       </PrivacyAssertion>
   ……..……
    </wsp:All>
</wsp:Policy>
```

Figure 13: eCommerce service Privacy Policy in WS-Policy

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 conduct an extensive evaluation of our Java-based prototype. 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 ease of use, perceived usefulness, and intention to use according to the Technology Acceptance Model (TAM) proposed in (Davis, 1989).

REFERENCES

Agrawal, R., Kiernan, J., Srikant, R., & Xu, Y. (2005). XPref: a preference language for P3P. *Computer Networks*, *48*(5), 809–827.

Agrawal, R., Kiernan, J., Srikant, R., & Xu, Y. (2002). Hippocratic databases. *VLDB* (pp. 143-145). VLDB Endowment.

Alrifai, M., Risse, T., & Nejdl, W. (2012). A hybrid approach for efficient Web service composition with end-to-end QoS constraints. *ACM Transactions on the Web*, *6*(2), 1–31.

Banerjee, M., Karimi Adl, R., Wu, L., & Barker, K. (2011). Quantifying privacy violations. *Secure Data Management*, 1–17.

Barker, K., Askari, M., Banerjee, M., Ghazinour, K., Mackas, B., Majedi, M., Pun, S., et al. (2009). A data privacy taxonomy. *British National Conference on Databases* (pp. 42–54). Springer.

Berardi, D., De Giacomo, G., Lenzerini, M., Mecella, M., & Calvanese, D. (2004). Synthesis of underspecified composite e-services based on automated reasoning. *SOC* (pp. 105–114). ACM.

Carminati, B., Ferrari, E., & K. Hung, P. (2006). Security Conscious Web Service Composition. *ICWS* (pp. 489–496). IEEE.

Chao, K., & Younas, M. (2005). Fuzzy matchmaking for web services. *AINA* (Vol. 2, pp. 721–726). IEEE.

Cranor, L., Langheinrich, M., Marchiori, M., & Reagle, J. (2002a). *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*. W3C Recommendation.

Cranor, L., Langheinrich, M., Marchiori, M., & Reagle, J. (2002b). A P3P Preference Exchange Language 1.0 (APPEL1.0). W3C Recommendation.

Davis, F. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, *13*(3), 319–340.

Foster, H., Uchitel, S., Magee, J., & Kramer, J. (2007). Ws-engineer: A model-based approach to engineering web service compositions and choreography. *Test and Analysis of Web Services*.

Fu, X., Bultan, T., & Su, J. (2002). Formal verification of e-services and workflows. *Web Services, E-Business, and the Semantic Web*, 188–202.

Guarda, P., & Zannone, N. (2009). Towards the development of privacy-aware systems. *Information and Software Technology*, *51*(2), 337–350.

Hamadi, R., & Benatallah, B. (2003). A Petri net-based model for web service composition. *ADC* (pp. 191–200). Australian Computer Society.

Hammond, J., Keeney, R., & Raiffa, H. (2002). *Smart choices: A practical guide to making better decisions*. Broadway Books.

Hewett, R., & Kijsanayothin, P. (2010). Privacy and Recovery in Composite Web Service Transactions. *International Journal for Infonomics*, *3*(2), 240–248.

Jeong, B., Cho, H., & Lee, C. (2009). On the functional quality of service (FQoS) to discover and compose interoperable web services. *Expert Systems with Applications*, *36*(3), 5411–5418.

Liu, L., Mehandjiev, N., & Xu, D.-L. (2011). Multi-criteria service recommendation based on user criteria preferences. *RecSys* (p. 77). New York, New York, USA: ACM.

Mahfouz, A., Barroca, L., Laney, R., & Nuseibeh, B. (2009). Requirements-driven collaborative choreography customization. *ICSOC* (pp. 144–158). Springer.

Massacci, F., Mylopoulos, J., & Zannone, N. (2006). Hierarchical hippocratic databases with minimal disclosure for virtual organizations. *The VLDB Journal*, *15*(4), 370–387.

Massacci, F., Mylopoulos, J., & Zannone, N. (2010). Security requirements engineering: the SI* modeling language and the secure tropos methodology. *Advances in Intelligent Information Systems*, *265*, 147–174.

Maximilien, E. M., & Singh, M. P. (2004). Toward autonomic web services trust and selection. *Proceedings of the 2nd international conference on Service oriented computing - ICSOC '04*, 212.

Nyre, Å. A., Bernsmed, K., Bo, S., & Pedersen, S. (2011). A Server-side Approach to Privacy Policy Matching. *ARES* (pp. 609–614). IEEE.

OASIS. (2007). *Web Services Business Process Execution Language Version 2.0*.

Paradesi, S., Doshi, P., & Swaika, S. (2009). Integrating Behavioral Trust in Web Service Compositions. *ICWS* (pp. 453–460). IEEE.

Singh, M., Chopra, A., & Desai, N. (2009). Commitment-based service-oriented architecture. *IEEE Computer*, *42*(11), 72–79.

Squicciarini, A., Carminati, B., & Karumanchi, S. (2011). A Privacy-Preserving Approach for Web Service Selection and Provisioning. *ICWS*, 33–40.

Tbahriti, S., Mrissa, M., Medjahed, B., Ghedira, C., Barhamgi, M., Fayn, J., & Bernard, C. (2011). Privacy-Aware DaaS Services Composition. *Database and Expert Systems Applications*, 202–216.

Tran, V. X., & Tsuji, H. (2008). QoS Based Ranking for Web Services: Fuzzy Approaches. *NWeSP* (pp. 77–82). Ieee.

Tumer, A., Dogac, A., & Toroslu, I. (2005). A semantic-based user privacy protection framework for web services. *ITW* (pp. 289–305). Springer.

Vedamuthu, A., Orchard, D., Hirsch, F., Hondo, M., Yendluri, P., Boubez, T., & Ümit Yalçinalp. (2007). Web Services Policy 1.5. W3C Recommendation.

Wang, P., Chao, K., Lo, C., Huang, C., & Li, Y. (2006). A Fuzzy Model for Selection of QoS-Aware Web Services. *ICEBE*, 585–593.

Wang, P., Chao, K.-M., Lo, C.-C., Farmer, R., & Kuo, P.-T. (2009). A Reputation-Based Service Selection Scheme. *ICEBE* (pp. 501–506). IEEE.

Xu, W., Venkatakrishnan, V., Sekar, R., & Ramakrishnan, I. V. (2006). A framework for building privacy-conscious composite web services. *ICWS*, 655–662.

Xu, Z., Martin, P., Powley, W., & Zulkernine, F. (2007). Reputation-Enhanced QoS-based Web Services Discovery. *ICWS* (pp. 249–256). IEEE.