3-2012

Privacy Protection Framework with Defined Policies for Service-Oriented Architecture

David S. Allison
Western University, dallison.uwo@gmail.com

Miriam AM Capretz
Western University, mcapretz@uwo.ca

Hany F. ELYamany
Suez Canal University, hany_elyamany@ci.suez.edu.eg

Shuying Wang
Western University, swang259@uwo.ca

Journal of Software Engineering and Applications, **2012, 5, 200-215**
http://dx.doi.org/10.4236/jsea.2012.53026 Published Online March 2012 (http://www.SciRP.org/journal/jsea)

Privacy Protection Framework with Defined Policies for Service-Oriented Architecture

David S. Allison[1], Miriam A. M. Capretz[1], Hany F. EL Yamany[2], Shuying Wang[1]

[1]Department of Computer and Electrical Engineering, The University of Western Ontario, London, Canada; [2]Department of Computer Science, Suez Canal University, The Old Campus, Ismailia, Egypt.
Email: {dallison, mcapretz, swang259}@uwo.ca, hany_elyamany@ci.suez.edu.eg

Received December 27th, 2011; revised January 31st, 2012; accepted February 15th, 2012

ABSTRACT

Service-Oriented Architecture (SOA) is a computer systems design concept which aims to achieve reusability and integration in a distributed environment through the use of autonomous, loosely coupled, interoperable abstractions known as services. In order to interoperate, communication between services is very important due to their autonomous nature. This communication provides services with their functional strengths, but also creates the opportunity for the loss of privacy. In this paper, a Privacy Protection Framework for Service-Oriented Architecture (PPFSOA) is described. In this framework, a Privacy Service (PS) is used in combination with privacy policies to create privacy contracts that outline what can and cannot be done with a consumer's personally identifiable information (PII). The privacy policy consists of one-to-many privacy rules, with each rule created from a set of six privacy elements: collector, what, purpose, retention, recipient and trust. The PS acts as an intermediary between the service consumer and service provider, to establish an unbiased contract before the two parties begin sending PII. It is shown how many Privacy Services work together to form the privacy protection framework. An examination of what current approaches to protecting privacy in an SOA environment is also presented. Finally, the operations the PS must perform in order to fulfill its tasks are outlined.

Keywords: Privacy; Service-Oriented Architecture; Web Services; Fair Information Practices; Policies; Contracts

1. Introduction

Service-Oriented Architecture (SOA) presents many challenges and security is considered to be one of the most difficult [1]. Security is a far reaching area covering such topics as authorization, authentication, auditing and privacy [2]. Amid all of these different areas of study, privacy often gains the least attention [3,4]. This lack of attention and the lack of any suitable solution are strong motivating factors in creating a privacy protection solution.

Privacy has no single definition as it is subjective to each individual. The definition of privacy in this paper is the ability to control information about oneself that has not been released, and to retain some measure of control over the information that has. Though often used interchangeably, privacy differs from both confidentiality and secrecy. Confidentiality refers to how private information provided to a third party is protected from release. Secrecy differs from privacy by being less about control of information, and more about keeping information invisible. For example, one's age may be considered private, but it is not a secret that a person has an age.

SOA provides a solution to finding, utilizing and integrating many different services to meet the business requirements of a consumer. The usefulness of services in providing business solutions is directly linked to the amount of interactions between different services. This property of the SOA domain poses a unique and challenging problem for dealing with privacy protection. As an increasing number of services are composed together, often from multiple sources, it becomes easier for a consumer to unwittingly expose private information. A common approach to protecting consumers from this exposure is to provide pseudonyms to identifying information. However this solution is incomplete as even hidden identities can be deduced by tracking patterns of usage [3]. Similarly, by tracking only seemingly harmless information, such as only the websites an anonymous consumer visits, one can deduce to a reasonable certainty, information including the consumer's age, gender, race and location. With the ability to perform more complicated tasks through SOA such as Internet banking [5], the risk of exposing unique personally identifiable information (PII) becomes a reality. The release of PII, including credit card numbers and social insurance numbers, can lead to

serious problems such as identify theft and falsified transactions. Most SOA services do not explain how or if they will collect personal information and those that do often do so in complex and confusing language that the average consumer cannot understand. The former situation gives a consumer no comfort at all, while the latter creates a one-way mirror effect [6] where PII is asked for but consumers do not know how it will be used. A privacy solution should address these issues, alerting a consumer to how their information can and will be used.

Another concern for a privacy solution is that it should not rely too heavily on customization or input from the consumer it is designed to protect, both initially and during its operation. As privacy is subjective and difficult to define, many consumers are left unqualified to make decisions on their own privacy [7]. Consumers want security, but they do not want to see it working [8]. Thus once configured, the privacy solution should run as silently as possible, only alerting the consumer when absolutely necessary.

In our past work [9,10], we have outlined the creation of privacy elements that represent enough information to thoroughly protect a consumer's privacy. What the consumer should be able to do and know is determined based on accepted principles used to protect privacy around the world [11]. The privacy elements together form a single privacy rule. A set of privacy rules along with identifying information creates a privacy policy. Previous attempts at privacy policies have left the definition of each element vague [12] or specific to a single situation [13]. In this paper these element descriptions will be expanded and finalized, defining what each element can be and how each element will be compared. An additional goal is to produce a policy whose rules allow it to cover both general conditions as well as very specific situations. The privacy policy will be defined and constructed using XML. The comparison of two privacy policies, one from a service consumer and another from a service provider, is used to create a privacy contract. This comparison will be accomplished by a Privacy Service (PS). The PS will also handle negotiations with the service consumer if conflicts between the two privacy policies arise.

There are several goals that together form the scope of this paper, which will now be outlined. The primary goal is to create a Privacy Protection Framework for Service-Oriented Architecture (PPFSOA). The PPFSOA makes use of privacy policies that can accurately portray the privacy of a service in any situation. To meet this goal, the definition of a privacy policy is created which contains rules made of six privacy elements: collector, what, purpose, retention, recipient and trust. A privacy policy allows a service consumer and provider to outline how they wish to deal with personally identifiable information (PII). The privacy framework includes a PS which has the primary job of comparing privacy policies to create binding privacy contracts. This PS is an autonomous, loosely coupled service that can be published and discovered in a repository. These properties allow the PS to be used and reused by many different consumers and providers. The PS will be detailed, with its role in the service consumer-provider-broker relationship defined. As privacy is but one part of security, an additional goal of the PS is to work in conjunction with other security services within a larger security framework [14].

The rest of this paper is divided into sections. Section 2 presents work related to the field of SOA privacy. Section 3 outlines the privacy elements required to protect the privacy of a consumer. How the privacy elements are created from accepted privacy practices is shown. These privacy elements form the basis of privacy rules. Privacy rules compose a privacy policy. Finally, two privacy policies are compared to create a privacy contract. In Section 4 the PS is introduced, which has the job of comparing privacy policies to create privacy contracts. How the PS acts in the typical SOA service consumer-provider-broker relationship is explained. It is also explained in Section 4 how many Privacy Services work together to form a privacy protection framework. Section 5 examines the implementation of the PS by outlining the operations the PS must perform in order to fulfill its tasks. These operations include both the internal and external processes of the PS. Section 6 presents a discussion on the work done in this paper, while Section 7 presents a summary and outlines possibilities for future work.

2. Related Work

In this section, the novelty of our work will be highlighted by its comparison to other privacy protection approaches.

IBM has presented a complete security model of SOA applications [15]. The model is presented by focusing on a banking industry scenario and consists of three levels of security: Business Security Services, Security Policy Infrastructure and IT Security Services. IBM combines the task of providing authorization and privacy into one group of services. Authorization in this case, is determining if a consumer has the right to access information. Privacy is considered an extension of this definition, determining if a consumer has the right to access Personally Identifiable Information (PII). This differs from the approach in this paper as authorization and privacy are considered tasks for two separate services. The approach by IBM to determining privacy authorization is done through the use of privacy policies. IBM relies on the standard WS-Privacy to describe how service consumers and providers state their privacy preferences within a

policy and XACML [16] to define and evaluate a policy. This reliance on WS-Privacy is ultimately the weakness of the IBM approach, as no such standard currently exists. WS-Privacy has been long discussed, with an anticipated completion date of 2004 [17], however it has yet to be completed. With the absence of WS-Privacy, the framework given by IBM lacks the vocabulary required to provide a proper privacy solution.

The Platform for Privacy Preferences Project (P3P) is a standard created by the World Wide Web Consortium (W3C) [18] that provides websites with a standard format for stating their privacy preferences. Privacy policies expressed in P3P can be formatted into readable documents quickly and easily by software known as user agents. P3P is designed as a protocol for websites and does not translate into the SOA domain; however the basic approach of P3P does provide useful insights into protecting privacy. P3P was designed around the Fair Information Practices (FIP) developed by the Organisation for Economic Co-operation and Development [19]. It is from these same FIP that the metamodel presented here was developed. P3P is designed in XML and uses the OECD principles to create eight top level tags [20]: category, data, purpose, recipient, access, retention, disputes and remedies. These eight categories represent the OECD principles in a similar manner to what is described in this paper; however as P3P was not designed for an SOA environment, the set of tags offered by P3P differs from the values chosen in this paper. Another difference between this paper and P3P is that P3P does not allow for comparisons between the values for each tag. No option is considered more or less secure than another and therefore P3P is not directly enforceable [21].

The Organization for the Advancement of Structured Information Standards (OASIS) has created an XML-based, general purpose access control policy language known as eXtensible Access Control Markup Language (XACML) [16]. XACML is designed to support the requirements of most authorization systems by providing the syntax for a policy language and the semantics for processing the policies [22]. XACML is platform independent and supports directly enforceable policies [23]. XACML uses an abstract model for policy enforcement, where all requests to access a resource must travel through the abstract component known as the Policy Enforcement Point (PEP) [23]. The PEP first gathers the access requests and then requests authorization from the Policy Decision Point (PDP). The PDP evaluates the authorization request and makes a decision based on any applicable policies and attributes related to the request that it can find. The PDP makes the authorization decision, but has no control over the enforcement of the decision, which is the responsibility of the PEP [23].

Recently OASIS has developed a Web Service profile of XACML known as WS-XACML [24] that can be used in the context of Web Services for privacy policies. Any WS-XACML Assertion consists of Requirements and Capabilities [24]. XACMLPrivacyAssertion, a specific definition of WS-XACML for privacy, requires a proper policy vocabulary to describe its Requirements and Capabilities. P3P is often selected as this vocabulary.

Although both P3P and XACML can be used to express privacy policies, they have different roles. A P3P policy is able to express privacy in a high level, easily readable form. An XACML policy expresses the same privacy conditions as the P3P policy, but in terms that a computer access control mechanism can understand and enforce [25]. By using P3P inside of XACMLPrivacyAssertion, this high level expression of privacy can be converted into a lower level machine readable format.

Dürbeck, Schillinger and Kolter [26] identify security requirements for an eGovernment Semantic SOA (SSOA). One of the security requirements they focus on is privacy. The authors make no specific selection of a privacy language, but instead identify potential candidates. The languages they select are P3P [18], EPAL [27] and XACML [16]. Along with using one of these languages, the authors suggest allowing for different privacy preferences per each process. They also suggest that the service provider provide mechanisms to enforce the consumer's privacy preferences.

Yee and Korba [12] have created a privacy policy specifically for e-services which could be used in an SOA environment. The policy is derived from the Model Code for the Protection of Personal Information created by the Canadian Standards Association [28]. This Canadian model was based on the OECD guidelines and therefore has a similar list of principles [29]. From the guidelines of the CSA, Yee and Korba [12] extract five privacy elements: collector, what, purposes, retention time, and disclose-to. These five elements represent the same information as five of the six elements described in this paper, with the exception of trust. Also unlike this paper, the elements presented by Yee and Korba are not fully defined. Beyond some examples of what a document containing these elements would look like, no definitions for the possible values of the elements could be found. From the examples the authors present [12], it appears specific values for whatever system is using the policy is envisioned, rather than a more general set of definitions. One of the goals of the metamodel in our paper is to create policies that can be specific when the situation warrants it, or can be general to encompass the privacy of many situations.

Yee [30] has another work that outlines an approach for estimating the privacy protection capability of a Web service provider. It gives an approach to what types of

data and equations are required to estimate privacy in a provider. This provides an example of how estimation of privacy can be done for a provider service.

3. Privacy Protection for SOA

A privacy policy consists of privacy rules, which are in turn created from a set of privacy elements. In this section the privacy elements and rules that compose the privacy policy will each be explained and defined. The definition process will start from the most basic concept, the privacy element, and show how these privacy elements form privacy rules.

3.1. Privacy Elements from OECD Principles

A goal of this paper is the construction of a privacy contract. The most basic part of this privacy contract will be the individual privacy elements. These privacy elements are designed to build privacy rules that can be general enough to function in many environments that use services, while retaining the ability to be specific to one case if required. This ability will allow the privacy elements to form rules that thoroughly protect privacy in either the general or specific case. A justification for the selection of the privacy elements must first be outlined. The elements to be created will be based on the Fair Information Practices (FIP) developed by the Organisation for Economic Co-operation and Development [19]. The FIP created by the OECD were selected as the basis for the privacy elements in our framework because these guidelines have been used as the model for most of the privacy legislation throughout the world [11]. The OECD FIP consists of eight principles: Collection Limitation, Data Quality, Purpose Specification, Use Limitation, Security Safeguards, Openness, Individual Participation, and Accountability [19]. The information that is required to be exchanged between service consumer and provider in order to satisfy these principles must be extracted. This extraction process is further described in our previous works [9,10]. After this process, six privacy elements are found to be required, which are described as follows:

- Collector: The collector of the data.
- What: What type of data will be collected.
- Retention: The length of time the collected data can be stored.
- Purpose: The reasons for which the data is collected.
- Recipient: Which parties, if any, the data is allowed to be disclosed.
- Trust: The level of trust of a PS.

Not every principle outlined in the OECD guidelines has been addressed by the requirements outlined above. This is because a privacy metamodel can only fulfill every privacy concern when included within a larger security framework. The Security Safeguards principle states that the data must be protected against unauthorized access and release [19]. These concerns are addressed through the use of traditional security techniques, such as authentication, authorization and encryption.

The Accountability principle states the more abstract concern of holding the service provider responsible for complying with all the other principles [19]. Accountability presents a unique problem for the SOA environment as the ability to provide enforcement is difficult and often nonexistent. It is therefore decided that accountability will be managed through the use of auditing. With the addition of an Auditing Service (AdS), neither party would be able to deny how data has been used. These requirements fall outside the scope of this privacy metamodel. Legislation would also likely be required to assist with accountability, as the AdS cannot determine violations any provider makes within their own systems, for example retaining information longer than the agreed upon retention time.

With the six privacy elements identified, a formal definition for each must be created. This will be accomplished next through the creation of a definition to specify the range of values each element can consist of, and the criteria for comparison of each element. Collector, what, retention, purpose, recipient and trust are the six privacy elements. Together, these six elements form a single privacy rule. A privacy policy is created by combining one-to-many privacy rules together with an identifying owner tag. This privacy policy structure is shown in **Figure 1**. The privacy policy of a consumer contains no actual private information about that consumer, only their preferences for protection. As such, the policy itself is unclassified and can be passed between Privacy Services as required.

3.2. Privacy Elements for SOA

***Definition* 1 (*Privacy Policy*):** We define a privacy policy as a tuple $<PP, CP, f>$ where $pp = <C_p, W_p, RT_p, P_p, RC_p, T_p>$, $pp \in PP$ represents a privacy rule associated with elements of collector, what, retention, purpose, recipient and trust respectively on the service provider side, $cp = <C_c, W_c, RT_c, P_c, RC_c, T_c>$, $cp \in CP$ is the consumer's privacy rule similarly, f is the set of comparison rules used to match the corresponding privacy elements between PP and CP.

3.2.1. Collector Element

The collector element states the name of the organization or party who will be collecting the data.

***Definition* 2 (*Collector*):** On the provider side, C_p consists of a single name of the service provider; on the consumer side, C_c consists of either a set of possible service provider names or the term "*Any*". $f(C_c \rightarrow C_p) = true$ if $C_c =$ "*Any*" or $C_c \subseteq C_p$.

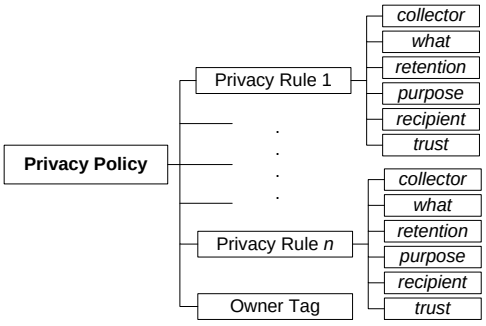

Figure 1. Privacy policy structure.

If "*Any*" is selected by the consumer, no comparison will be made between the fields. If a set of one or more names is specified by the consumer, this consumer set must contain the name specified by the provider in order for the comparison to be successful.

3.2.2. What Element

The what element allows the privacy policy to outline what types of private information will be collected. It is impossible to universally declare one piece of information more private than another since what is considered private information varies greatly between different individuals. Instead, individuals are allowed to rank their information according to four ordered levels, based on the levels of classification used by the government of the United States of America [31]. Though these levels are the same as the Bell-La Padula (BLP) model [32], the framework presented in this paper does not share the same properties as BLP, such as no write-down. This scheme of classification was selected because it is in use throughout the world and the vocabulary used is such that a layperson could easily discern the order in which the levels are ranked.

An individual acting as a service consumer would be required to sort a list of the most common types of private information into the four levels according to their own preferences. These four levels are Top Secret ("*TS*"), Secret ("*S*"), Confidential ("*C*"), and Unclassified ("*U*"). This information would be saved on the consumer side in a document called a What Element Ranking (WER). This requirement does place some responsibility on the individual, which should regularly be minimized. Due to the subjective nature of private information, this responsibility is unavoidable.

***Definition* 3 (*What*):** We define $PI = <PI_p, PI_c>$ to be private information required for the provider and consumer sides containing all of the different types of PII such as name, age, address, sex, and so on. H = {"*TS*", "*S*", "*C*", "*U*"} is a hierarchy set with order "*TS*">"*S*">"*C*">"*U*". On the provider side, $W_p = PI_p$ is a set of required private information. On the consumer side, $WER = \{PI_{TS}, PI_S, PI_C, PI_U\}$ is PI_c ordered with H and $W_c = h$, $h \in H$ is a one of four possible H values correspondingly. $f(W_c \rightarrow W_p) = true$ if $W_p.h \leq W_c$ and $W_p \subseteq wer$, $wer \in WER$, $wer.h = W_c$.

The set of values specified by the provider will be compared to the level selected by the consumer and their corresponding WER. In order for this element to be compared successfully, each of the values mentioned by the provider must be less than or equal to the level selected by the consumer. If any piece of information asked for by the provider is missing from the consumer's WER, the consumer will be informed and given the opportunity to add it.

3.2.3. Purpose Element

The purpose element is important in determining if a service consumer and service provider should be allowed to interact and share information. Purpose can be interpreted in two different ways. The first is to consider the purpose to be the goal of the service, such as for "*Identification*". The second interpretation is for purpose to outline the operational reasons for needing data access, and will consist of four possibilities, No Collection and No Distribution ("*NC&ND*"), Collection & No Distribution ("*C&ND*"), Collection and Limited Distribution ("*C&LD*"), and Collection and Distribution ("*C&D*"). In order to fulfill both of these interpretations, a purpose element that consists of two parts, a goal and an operation, will be created. The goal is required from the service provider in order to inform the consumer and to satisfy the OECD guidelines. If the consumer wishes to limit their data to a particular goal they have that option, or they can choose "*Any*" and allow any purpose as long as it satisfies the second criterion. This second criterion is the operation, which will consist of four ordered levels outlining the possible operational uses of data.

***Definition* 4 (*Purpose*):** A Purpose P is a tuple $<g, o>$ which contains a goal g, and an operation o. On the provider side, g = <*Records*, *Mapping*, *Identification*…> if $P = P_p$, while on the consumer side, g can be either "*Any*" or g = <*Records*, *Mapping*, *Identification*…> if $P = P_c$. For both sides, o is a value from four levels {"*NC&ND*", "*C&ND*", "*C&LD*", "*C&D*"} with the order "*NC&ND*" > "*C&ND*" > "*C&LD*" > "*C&D*". $f(P_c \rightarrow P_p) = true$ if ($P_c.g$ = "*Any*" or $P_c.g = P_p.g$) and $P_c.o \leq P_p.o$.

The creation of a hierarchy allows for the comparison of two privacy policies even if the specific goal of the provider has not been outlined by the consumer. As long as the consumer has defined a rule with the value "*Any*" selected in the goal portion of the purpose element, a valid comparison can be made if the consumer's corresponding operation level is less than or equal to the provider's operation level. This greatly reduces the total number of rules required in the privacy policy of a

consumer. If the consumer has specified any other value in the goal portion of the purpose element, two comparisons must be done in order to successfully compare the element. First, the goal portion of the purpose element of the consumer must equal the goal portion of the purpose element of the provider. Second, the level of operation selected by the consumer must be less than or equal to the level of operation selected by the provider.

3.2.4. Retention Element

Retention is an element that outlines how long a consumer's data may be stored by a provider. For the consumer, the retention element RT_c is an integer –1 or greater, used to state in days, how long data can be held by a provider. The value of –1 is used to represent the case where gathered information is allowed to be retained for an unlimited amount of time. This is useful when a consumer is not concerned with how long a particular piece of information is held. Zero is a valid input for RT_c and it represents that any data collected on the consumer must be deleted immediately upon completion of the service. On the provider side, the retention element RT_p is used to state how long they wish to retain a consumer's data. *RT* is a non-negative integer which represents the number of days past the completion of the service the data may be held. Zero is a valid input for RT_p, and represents that any data collected will be deleted immediately upon completion of the service.

***Definition* 5 (*Retention*):** Retention, *RT*, consists of a non-negative integer representing days. The consumer has the additional choice of selecting –1, which represents an unlimited amount of time. $f(RT_c \rightarrow RT_p)$ = true if $RT_c = -1$ or $RT_p \leq RT_c$.

3.2.5. Recipient Element

The recipient element is unique in its comparison by working in conjunction with both the purpose and collector elements. The recipient element is only compared if the Collection & Limited Distribution ("*C&LD*") level is specified in the operation portion of the consumer's purpose element. If this level is specified by the consumer, they must select from one of two options: "*Delivery*" or "*Approved*". If "*C&LD*" is not selected by the consumer, Recipient may be left blank and is ignored. "*Delivery*" is selected by the consumer if they will allow for any third party to be involved as long as it is required in order to deliver the original service. This option is also useful in situations where the consumer does not know which parties may be involved in the transaction but still wish for the transmission of their data to be limited. "*Approved*" allows for the consumer to specify a list of approved providers who are then allowed to have the consumer's data passed to them. If "*C&LD*" and "*Approved*" are selected, the names listed in the provider's recipient element must be a subset of the names listed in the consumer's collector element in order for a successful comparison.

***Definition* 6 (*Recipient*):** For the provider side, Recipient RC_p will consist of a set of names, listing each third party service provider who could possibly receive data from the original service provider. For the consumer side, RC_c will be empty or state "*Delivery*" or "*Approved*". $f(RC_c \rightarrow RC_p) = true$ if ($RC_c = \varnothing$ and $Pc.o \neq$ "*C&LD*") or ($P_c.o$ = "*C&LD*" and RC_c = "*Delivery*") or ($P_c.o$ = "*C&LD*" and RC_c = "*Approved*" and $RC_p \subseteq C_c$).

The recipient element outlines who is permitted to have the data passed to them. Since the consumer specifies who may receive their data with the collector element, the recipient element does not need to list any provider names. For the provider, recipient will consist of a set of names, listing each third party service provider who could possibly receive data from the original service provider.

3.2.6. Trust Element

The trust element gives the consumer a degree of control over what PS can be used to negotiate the privacy contract. Without this ability, the consumer would have no assurance that the policy comparison is being done without bias. The provider in this element provides the name of the PS it wishes to use. There are four levels of trust a consumer can select for a PS to have: High ("*H*"), Moderate ("*M*"), Low ("*L*") and Not Required/Not Ranked ("*NR*").

***Definition* 7 (*Trust*):** On the provider side, the trust element T_p is represented as the name of a privacy service. R = {"*H*", "*M*", "*L*", "*NR*"} is a hierarchy set with order "*H*" > "*M*"> "*L*" > "*NR*". On the consumer side, trust element $T_c = r$, $r \in R$ is one of four possible *R* values correspondingly. $f(T_c \rightarrow T_p)$ = true if $T_c \leq T_p.r$.

For the comparison of this element to be successful, the trust level of the PS the provider supplies must be at least as high as the level chosen by the consumer. Ratings are given to each PS by consumer or provider services that have previous experience using the PS. These ratings can be used to develop a trust metric. Trust metrics are algorithms that are able to predict the trustworthiness of an unknown user [33], or in the case of SOA, an unknown service. Trust metrics fall into two large categories, global and local. Global trust metrics contain one level of trust for each member in the community so every member has the same opinion of every other member. Local trust metrics allow for each member of the community to have a different opinion on each other member. Local trust metrics provide finer control over the levels of trust in a system, but due to their complexity, are far more computationally expensive than global trust metrics [33]. Due to the large number of po-

ssible services in an SOA, a global trust metric is recommended. The selection of a specific type of global trust metric and its use falls outside of the scope of this paper as there are many different trust metrics available [33], each of which should be considered in greater detail. This selected global trust metric will be used to determine the level of the service: high, moderate or low. If a service does not participate in the global trust metric or has yet to be assigned a proper trust level, the fourth level of not ranked will be used.

The question of who will carry out the trust classification is an important one. The classification can be carried out either internally by the company or party that provides the PS, or externally by an outside body. Internally would require no extra party be involved and therefore less work, but ultimately will be too unconvincing to a consumer. If each PS simply rates itself, the consumer would be unconvinced by the credibility of the rating. Therefore the latter option is required, that being an external body which would gather the ratings, generate and store the trust metrics. Such a body could be a trusted organization such as the W3C, OECD or local government. Ultimately there must be some motivation for services to not provide the governing body with false information. This must come in the form of legislation that provides punishment for breaks of privacy and for knowingly providing false data. A government acting as the trusted organization is the best solution as they can enact laws and provide enforcement. Such legislation has already been enacted by many countries around the world [34,35]. If a service resides outside the jurisdiction of a government that performs privacy trust rankings and therefore cannot be ranked, it will remain at the NR level. If a consumer wishes to use one of the NR ranked services they can knowing they are at further risk, otherwise a more local and ranked service can be used.

3.3. Providing Context

It is important that any privacy policy be able to provide specific context if required. This means that a consumer should be able to specify one situation where their information is released and another very similar situation where their information is not released. This is provided in the privacy policy presented here by having rules that include elements that allow for specific input, such as purpose, goal and collector, and allowing multiple rules per policy. With these tools available, a consumer will be able to create rules that allow for the release of different information in specific situations.

3.4. Providing Context

With the elements of the privacy policy now defined, it can be shown how this implementation improves on current, popular privacy policy implementations, such as P3P and XACML. P3P has a number of identified areas for improvement, and one of these areas is the lack of specificity in outlining the purpose for gathering data [36]. This is addressed in this section by the purpose element, which is sub-divided into two areas, goal and operation, allowing for greater descriptions. Another area P3P needs improvement in is transitivity, where privacy may not be protected when information is passed from one party to another [36]. This is addressed in the privacy policy model presented in this section through the development of hierarchies for elements that are not directly comparable, ensuring information is only passed to parties that are at least as secure as the original provider. XACML and P3P perform complementary services, where P3P is a high level, human readable vocabulary and XACML is a low level vocabulary understood by access control mechanisms [25]. As the privacy policy definition presented here improves upon P3P in an SOA environment, the presented privacy policy could replace P3P and work together with XACML.

4. Privacy Contract Agreement

With the elements that compose privacy rules now defined, along with how each should be compared, it next becomes necessary to outline how this comparison should be carried out. In this section the stages required to agree upon a privacy contract when establishing a connection between a service consumer and provider will be described. A Privacy Service (PS) supplied by a trusted third party will be used as an intermediary between the service provider and consumer. The third party PS is required to ensure the comparison of privacy policies is done correctly and without bias. The PS also allows for the final contract to be stored in a neutral location. The comparison of policies is carried out using the rules defined in the previous section. **Figure 2** demonstrates the five main stages in this process.

4.1. Publish Stage

Publish is the first stage, which requires the service provider to send information about itself to the service broker to be published. This information includes where the service is located, how to establish communication and what tasks the service can provide. The service broker acts as a repository, so this information can be advertised to be discovered by a service consumer. This stage is unchanged in the given scenario from a typical service publishing stage that does not concern itself with privacy.

4.2. Find Stage

In the next stage, Find, a service consumer sends a request to the service broker asking for a service provider

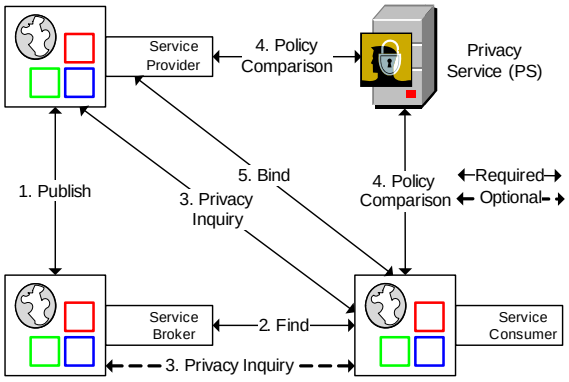

Figure 2. Contract agreement and contact stages.

to help accomplish a task. The service broker will return the information it has on a matching service. This stage is also unchanged from a typical service discovery scenario.

4.3. Privacy Inquiry Stage

Privacy Inquiry is the third stage and the first that is unique to the approach in this paper. Once the service consumer has retrieved the information on the service provider from the service broker, it queries the service provider to determine if the provider is willing to create a privacy contract. As the Privacy Protection Framework for Service-Oriented Architecture (PPFSOA) is an addition to the current approach of interaction in an SOA environment, there will possibly be services which do not implement it. If the consumer receives no reply from this early request, they know to look for a different provider. If the provider does utilize a privacy policy, they will respond with a message confirming this, as well as suggesting a PS. Due to a trust element being defined by the consumer, there is the chance that the PS suggested by the provider does not meet the consumer's desired level of protection. Since the PS is itself a service, it can and will be published to a service broker, for example the Universal Description, Discovery and Integration (UDDI) [37]. If the provider's PS is unacceptable to the consumer, they can choose to terminate the transaction or search the broker for a suitable PS and submit its details to the provider. This optional second half of the Privacy Inquiry stage is shown as a dotted line in **Figure 2**. If the provider for some reason rejects the PS, it can provide a counter offer or also terminate the transaction. The risk of the provider rejecting the PS should be low, as the main reason for rejection of the original PS by the consumer would be its low privacy ranking. Therefore any counter proposal should be of a PS that is of greater security, which also benefits the provider. However the provider does retain the option of rejection, at the risk of losing the consumer and any revenue from the consumer's patronage.

4.4. Policy Comparison Stage

In an example of policy comparison, there is a service consumer Ellen Doe and a service provider SaveRx. The service provided by SaveRx gives product information on drugs available for sale in their stores. SaveRx keeps records containing each consumer's name, address and date of birth. It also has the option to provide the consumer with directions to their closest SaveRx store through the use of a third party mapping service. These requirements are specified in the privacy policy of SaveRx as shown in **Figure 3**.

Ellen has previously created a WER document by ranking the different types of private information according to her own preferences. Ellen has also created a privacy policy, containing one rule. This rule outlines that Ellen is willing to allow anyone to collect her information as long as it is ranked Confidential or less. The two documents created by Ellen are shown in **Figure 3**. The different steps in the Policy Comparison Stage will now be explained and shown in context to this example.

4.4.1. Send Policy

In the first step, the service consumer sends the PS its privacy policy, along with a copy of its WER. Similarly, the service provider sends the PS its privacy policy. In the presented example, the three documents sent are shown in **Figure 3**.

4.4.2. Compare Policies

Using these three documents, the PS compares each element of the consumer rules to a corresponding provider rule using the comparisons previously outlined.

In the example, SaveRx is attempting to collect from Ellen her name, address and date of birth. The PS first ensures that each of these items can be collected by consulting the consumer's WER and policy. Ellen's policy only consists of one rule, which allows for the collection of information ranked confidential or lower. The collected WER states that all of name, address and date of birth are collectable under this rule.

Continuing the comparison of policies, the PS then checks each rule in the provider's policy against the available rules in the consumer's policy. The first rule for SaveRx attempts to collect a name, address and date of birth from Ellen. These three pieces of information are all addressed by the first rule in Ellen's privacy policy through comparison of the what element. This rule specifies that "*Any*" collector is permitted, so no comparisons are required for that element. The next element, retention, specifies that Ellen permits this information to be stored for a maximum of 30 days. SaveRx will only be storing the information for 7 days, so this element also results in a pass. The purpose element consists of two parts, a goal and an operation. The goal is compared first, and in the

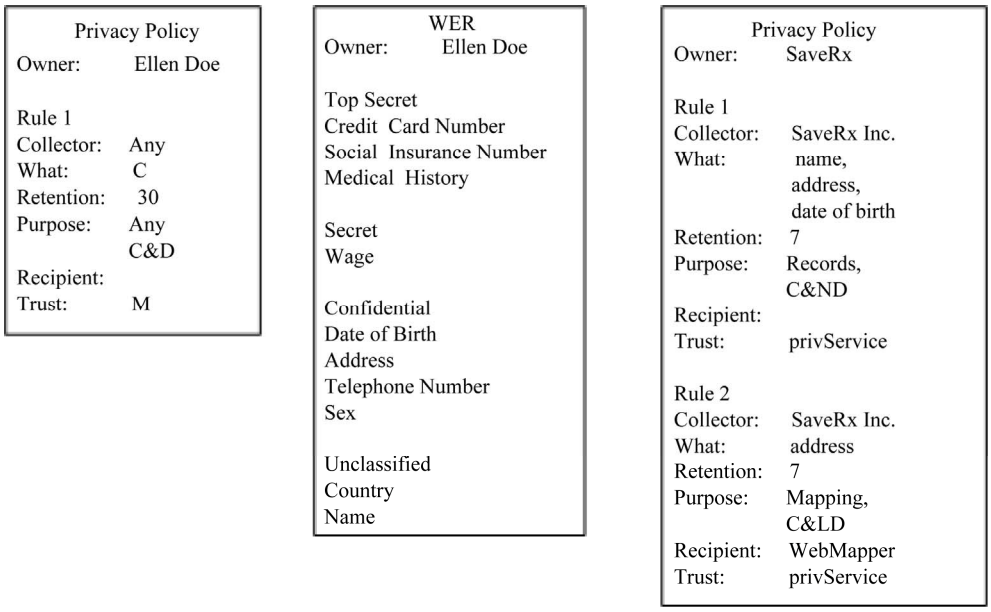

Figure 3. Policy document examples.

given scenario "Records" in SaveRx's policy is matched to the "*Any*" in Ellen's policy. The operation of the purpose elements are then compared and accepted, since SaveRx's "*C&ND*" is more secure than Ellen's "*C&D*". The fifth element, recipient, does not need to be compared since Ellen has not selected "*C&LD*" in purpose. The last element, trust, is not used directly in this step, as they were already used in the Privacy Inquiry stage. If the negotiation has proceeded this far, it can be assumed that the PS meets the consumer's requirements. Instead of a second comparison, the level of trust given by the consumer is simply recorded along with the name of the PS. After the first rule is successfully checked, the PS would then move to the second rule in SaveRx's policy. These comparisons would be similar to the comparisons done previously, with each resulting in a success in this case.

4.4.3. Inform Results

If any problems with the comparison of the WER or policies outlined in the previous Compare Policies step occur, the service consumer is informed and given the chance to change their policy or WER accordingly. If the change is made, the Compare Policies step is repeated using the changed documents. If the service consumer does not choose to change their privacy policy or WER, the transaction fails and the consumer returns to the Find stage to locate a new service provider. If the policies are matched successfully, the two parties are informed and asked to sign a finalized privacy contract that outlines their agreement.

In the given example, during the Inform Results step, the PS sends the consumer and provider a message informing both of the positive results gathered in the Compare Policies step. A request is made in this message for each party to sign a copy of the agreed upon privacy contract.

4.4.4. Sign Contract

In the final step, a copy of the signed privacy contract is kept by the PS for record keeping. By signing the contract, both parties signify that they understand and acknowledge the terms and are now bound to them. This signed copy is stored for future reference by the provider of the PS in a secure location. An acknowledgement message is sent by the PS to both the consumer and provider once both signatures have been received.

In this example, since the comparison was successful, both parties sign the final contract. The acknowledgement message is sent by the PS to both consumer and provider.

4.5. Bind Stage

The fifth and final stage is Bind, which occurs once the privacy contract has been agreed upon and signed by both service provider and consumer. Once this process has completed, the two parties are free to interact with each other. This stage is the same as the third stage in a typical service discovery scenario where privacy is not a concern.

4.6. Privacy Protection Framework for Service-Oriented Architecture

Now that it has been explained how the PS will work with policies to create privacy contracts, an overall view

of the privacy framework can be provided. This Privacy Protection Framework for Service-Oriented Architecture (PPFSOA) is shown in **Figure 4**.

The PPFSOA is an expansion of the Privacy Contract Agreement described earlier in this section. The Privacy Contract Agreement outlines how a single PS works with a single consumer-provider-broker relationship to provide privacy. The PPFSOA utilizes an Enterprise Service Bus (ESB) to implement this on a larger scale, providing privacy to many consumers at once. The ESB will allow for message routing between each PS, monitoring of traffic, language transformation, exception management and the duplication of a PS when traffic becomes heavy. With the use of an ESB, a number of Privacy Services running in parallel will be made available to meet the demands of the environment. Each individual PS will rely on the message routing abilities of the ESB to ensure that it receives messages from the correct consumer and provider, and that the messages sent by the PS reach their correct destination.

The PPFSOA can be adopted into current SOA environments since it requires minimal changes to the parties currently involved. The service broker, such as UDDI, requires no changes as the privacy service is advertised to it as a regular service. The service consumer and provider only require the addition of a privacy policy. An education campaign would be required to advertise the PPFSOA to providers and consumers who do not currently implement privacy protection. Software will be provided to assist in the creation of privacy policies. This software will use a graphical user interface (GUI) to allow the consumer or provider to set their privacy preferences to any available level, and will automatically create the policy in the correct XML format once each preference is selected. As previously mentioned, the PPFSOA is designed to sort between providers that do and do not implement privacy policies, so once a consumer has a privacy policy created, the framework will ensure that they only deal with providers that have their privacy preferences in mind.

5. Implementation

The PS now has its place in the service consumer-provider-broker relationship defined, along with what tasks it can perform. It also has its place defined in the context of a larger security framework. How the PS specifically carries out its role will be discussed next. The purpose of the PS is to act as an intermediary between the service consumer and provider and negotiate a privacy contract that both sides can agree upon. In the following section the operations that the PS will perform in this comparison process are detailed. Since this section deals specifically with the PS, in each of the operations listed below it is assumed that the service consumer and service provider have already completed up to and including the Privacy Inquiry stage in the Privacy Contract Agreement process, as outlined in the previous section. Sequence diagrams are also presented in this section to outline the series of events that take place through different policy comparison scenarios.

Figure 5 shows the sequence of events during the sim-

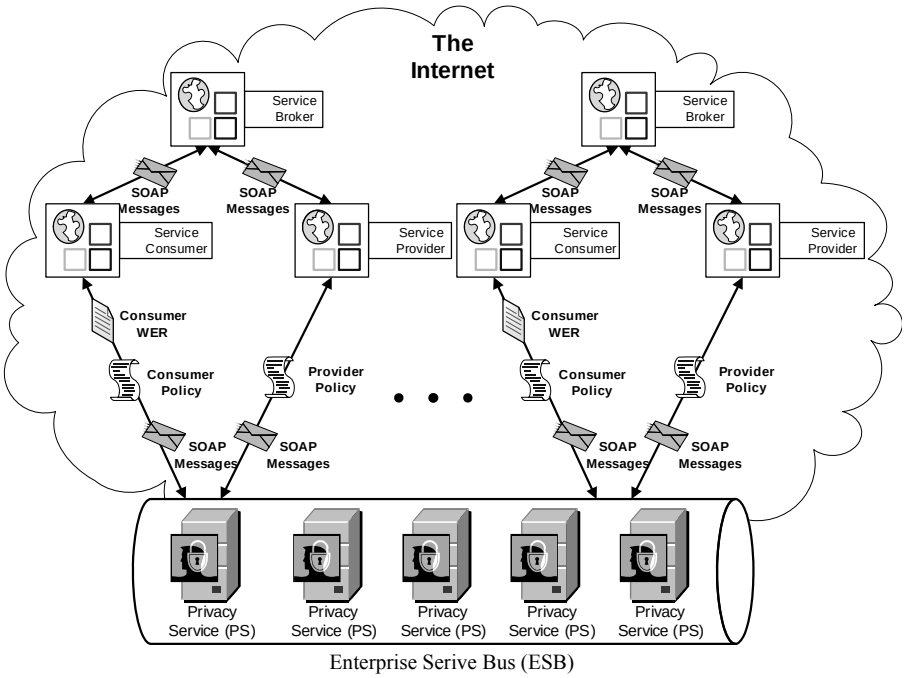

Figure 4. Privacy protection framework for SOA (PPFSOA).

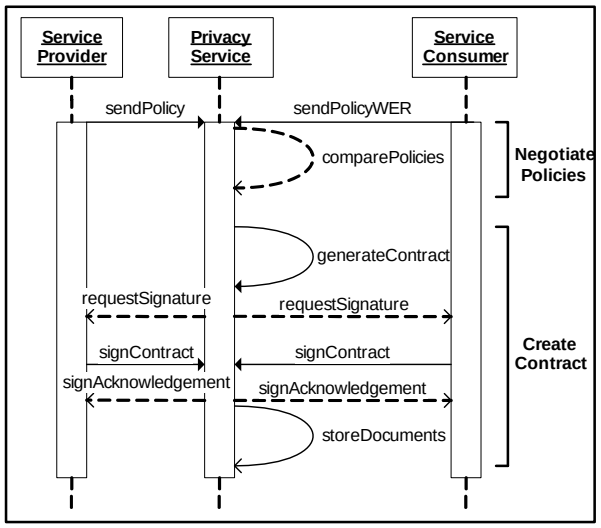

Figure 5. Successful comparison of policies.

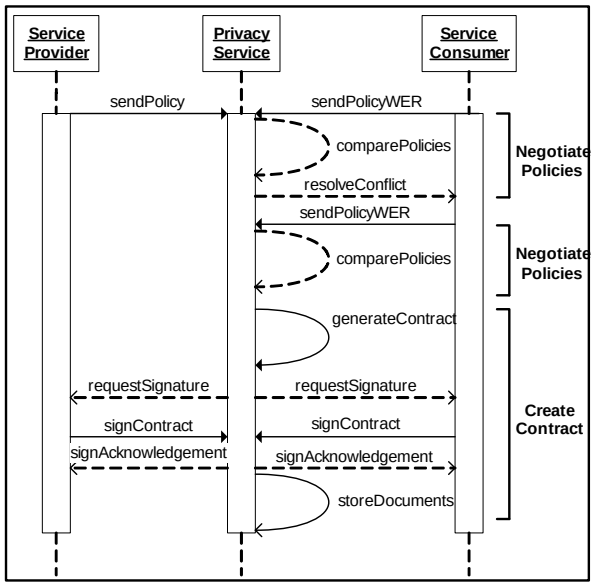

Figure 6. Successful comparison of policies following a conflict.

plest example of a successful comparison of policies. **Figure 6** demonstrates the sequence of events during another successful comparison of policies, however this time after encountering a conflict within the policies. **Figure 7** demonstrates a conflict that could not be resolved, resulting in an unsuccessful comparison of policies. **Figure 8** outlines the steps in a scenario where policies must be converted after a successful initial comparison in order to deal with a third party provider. Finally, **Figure 9** shows the steps taken when one party challenges the terms of a previously successful comparison of policies.

5.1. Negotiate Policies Operation

In this operation, the PS is the recipient of two messages, one from the service provider and one from the service consumer. The message sent by the service provider contains a copy of its privacy policy, while the message sent by the service consumer contains its privacy policy and a copy of its WER. Examples of these three documents can be seen in **Figure 3**. Using these three documents, the compare Policies operation is carried out by the PS, where each element of the consumer's rules is compared to a corresponding provider rule using the evaluations outlined earlier in this paper. If any problems in the match occur, a resolve Conflict message is sent to the service consumer. This message informs the consumer of the problem and suggests changes required to its privacy policy or WER. If the consumer replies with a new privacy policy and WER, the comparePolicies operation is repeated using these new documents and the previous service provider privacy policy.

5.2. Create Contract Operation

The Create Contract Operation is performed by the PS

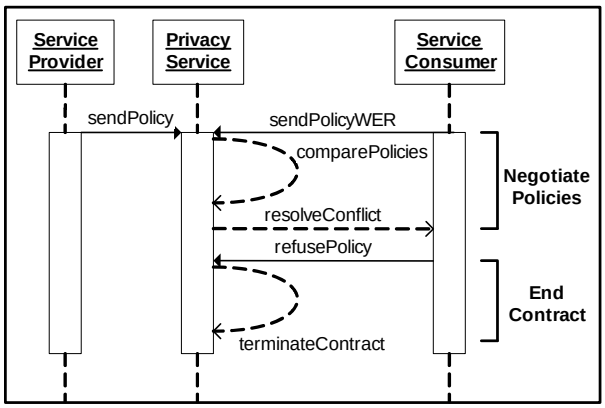

Figure 7. Unsuccessful comparison of policies.

once the Negotiate Policies Operation has completed. In this operation a privacy contract is first generated by taking the value provided by the consumer, or the provider if the provider's option is more secure. A message is sent to both the service consumer and provider stating that an agreement has been met and that a final signature is required. This requestSignature message also contains a copy of the agreed upon privacy terms for both parties to observe if they so require. The two parties sign the contract through the use of a secure identifiable process, such as Public Key Infrastructure (PKI) or XML Signature [38]. An acknowledgement message is sent to both parties informing them that the contract has been signed and is final. The privacy terms are combined with the names of the provider and consumer, a timestamp, and a signature from both parties. Together, these form one entire privacy contract. The signature acknowledgement signifies that the provider and consumer are now free to

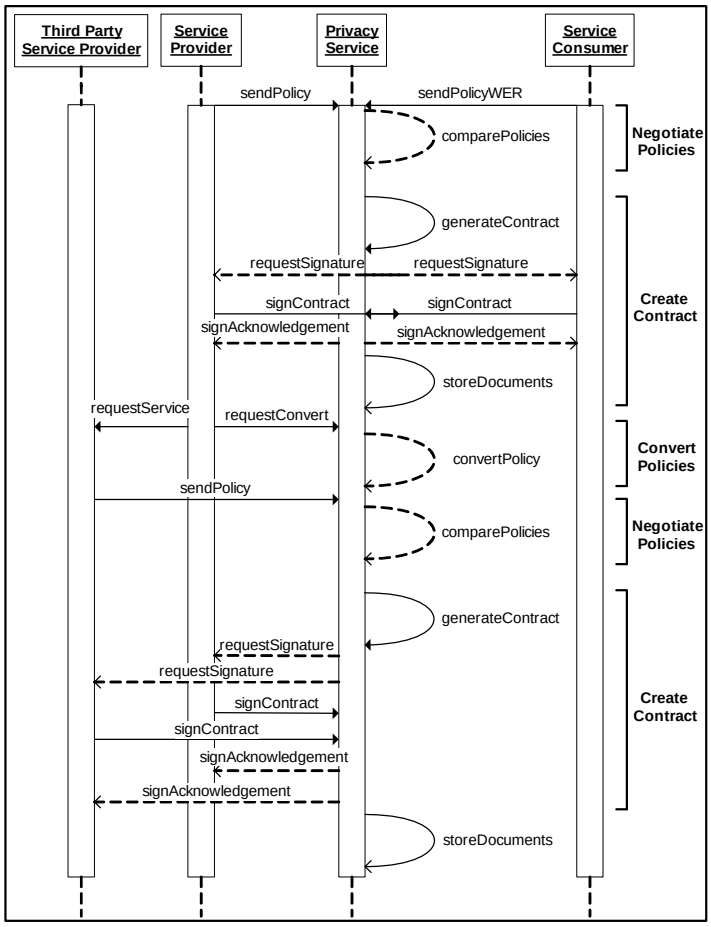

Figure 8. Convert policies and successful comparison with a third party.

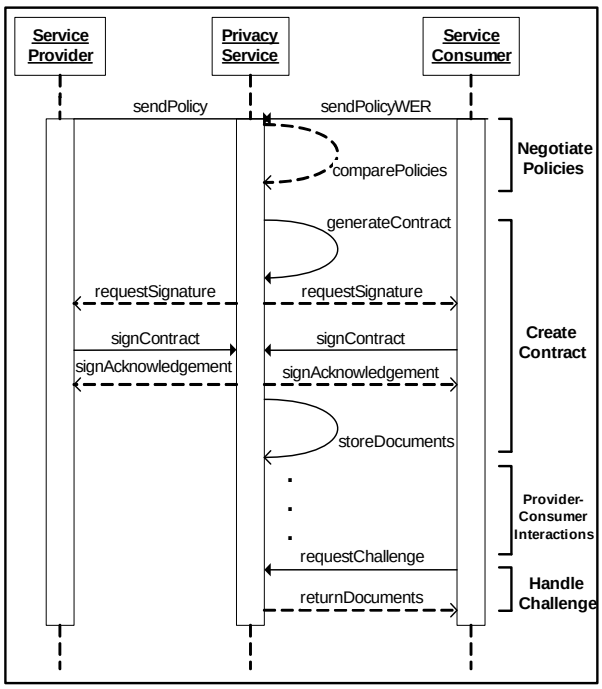

Figure 9. Challenge of contract following interactions.

interact. A copy of the contract is then stored by the PS in a secure database, along with both policies and the consumer's WER. These documents are stored alongside the contract in order to be available in case a challenge is made to the validity of the agreement. Examples of the Create Contract Operation are shown in **Figure 5**, **Figure 6**, **Figure 8** and **Figure 9**. Each of these diagrams shows the same Create Contract Operation occurring in the middle of different scenarios.

5.3. End Contract Operation

The End Contract Operation begins when a message is received from the service consumer that states they refuse to change their privacy policy. This refusePolicy message is sent in response to an attempt by the PS to negotiate a new privacy policy. Once this message is received, the PS terminates the attempted negotiations and discards the current privacy policies and WER documents it has gathered. An example of this operation is shown in **Figure 7**.

5.4. Convert Policies Operation

If a service provider requires the use of a third party ser-

vice to accomplish a goal, it must become a consumer in a new consumer-provider relationship. This operation follows the successful contract creation between the service provider and an original service consumer. At some point during the interactions between this service consumer and provider, the provider requires the use of a third party service. The original service provider will begin a Privacy Contract Agreement with the third party provider, while alerting the PS that a conversion of contracts is required. The PS will then generate a new policy for the provider-turned-consumer based on the original agreement. When the PS receives the policy from the third party provider, the generated policy is treated as a consumer policy and compared to the third party provider policy as if it were a typical consumer-provider scenario. An example of the Convert Policies Operation is shown in **Figure 8**.

5.5. Handle Challenge Operation

This operation will return the contract and associated policies when a challenge message is received from either party. A challenge occurs only after a contract has been agreed upon and some interactions between the service consumer and provider have taken place. A request Challenge message can then be sent by either party to the PS if that party believes information is being used or gathered in violation of the signed privacy contract. The PS will reply with a return Documents message which contains the original contract. An example of the Handle Challenge Operation is shown in **Figure 9**.

6. Discussion

Service-Oriented Architecture is a desirable software system structure due to the strengths and abilities of its interoperable services. Services in an SOA environment are loosely coupled to their underlying technologies, requiring no one standard be followed. These services are autonomous and are made available over computer networks by their providers in order to be used, reused or combined in any way their consumer prefers. Services coordinate by passing information from one service to the next, allowing services from many different sources to combine to meet any problem or need. This communication among services, while provides the backbone of an SOA, raises many important privacy concerns. With many services possibly passing information between themselves, a consumer can quickly become unaware of how each service is using their personally identifiable information. Businesses have found that in order for any emerging technology to gain widespread success, it must be trusted by the general population of consumers. Applying privacy policies designed for Web pages to an SOA environment is not enough to provide adequate privacy while maintaining the usability of services. A new approach that considers the interactive characteristics of services must be created to enhance consumer privacy while minimizing consumer interruptions.

The major contribution of this paper to the field of SOA privacy is the creation of a Privacy Protection Framework for SOA (PPFSOA). The PPFSOA embraces the use of a privacy policy suitable for any SOA environment. This privacy policy is able to describe privacy rules that range from very specific to very general. A comparison of privacy policies within the PPFSOA is carried out in order to create a privacy contract. These comparisons are designed to be impartial, transitive and require minimal effort from the consumer. The Privacy Service in the PPFSOA is created like any other service in an SOA environment. This means the PS is an autonomous, loosely coupled service that can be published and discovered in a repository and reused by many different consumers and providers. The PPFSOA includes the steps required for the PS to interact with the service provider and service consumer, acting as an intermediary between the two.

Other contributions were also made through the creation of a privacy policy suitable for protecting privacy in an SOA environment. The elements that make up this privacy policy were fully defined, outlining the boundaries of what the elements could and could not be.

Work in the field of SOA privacy is still young and progressing. Within real world SOA environments, privacy protection is often ignored. Those few attempts at addressing issues of privacy in an industrial SOA setting are often done so unsatisfactory or left incomplete. This is exemplified in the security approach proposed by IBM, in which a privacy standard is referenced that as of yet does not exist [15]. Activity in the academic world has begun to address the issues related to SOA privacy, but has also progressed slowly. The PPFSOA described in this paper moves the field forward by taking a serious look at the unique privacy concerns of SOA and not settling for a solution that was originally designed for different environment. The PPFSOA attempts to take the onus of protecting private information off the individual consumer. In the cases where the consumer is required to perform actions, guidelines and information are provided to assist the consumer in making informed decisions.

The PPFSOA described in this paper provides the ability to create privacy contracts. Unfortunately, enforcement of these privacy contracts is not guaranteed with this framework. Providers that choose to disregard the contracts they have signed with consumers risk the loss of use and profits that will occur when such transgressions are exposed. In order to provide greater accountability, governing bodies will eventually be required to monitor providers to ensure their agreements are fol-

lowed. Legislation addressing privacy in an SOA environment would assist any governing body by providing tools of enforcement, such as punishment for infractions and the ability to audit systems.

7. Conclusion and Future Work

The PPFSOA allows for privacy elements to be quantified in order for privacy rule comparisons to be possible. This framework has the inherent issue of requiring the service provider, consumer and broker to all follow the same standard approach. This issue is addressed by the fact that no privacy standard for SOA currently exists. The PPFSOA was created to fill this need and the end goal of the PPFSOA is to become a privacy standard for SOA.

The relatively young age of privacy protection for Web services means there are still points to be addressed in future work. The PPFSOA that has been suggested here needs to be tested in a full business environment. This evaluation would provide further evidence of the framework's performance and ability to protect privacy, while giving valuable insights to areas that can be improved or optimized.

The creation of a unique Enterprise Service Bus (ESB), which also acts as a service, is planned. The ESB service will work as an interface between the many service consumers and the PPFSOA. When the PPFSOA is placed within a larger security framework, the ESB will also act as an interface between the Privacy Service and other security services. The ESB will be tasked with the job of routing messages from services to their correct destination. When in public use, the demand for privacy will quickly overwhelm any single PS. To overcome this issue of traffic, the ESB will also monitor the number of requests through the use of an intelligent engine and replicate the PS as many times as required to meet the current demand. The ability to dynamically replicate the PS will provide assurance that each SOAP message sent is processed within an appropriate amount of time. Replication will also improve the accessibility and performance of the PPFSOA, as its intelligent engine will predict the number of Privacy Services that are required to process the current number of SOAP messages. This will ensure a PS is always available for any consumer who wishes access to one.

Different consumers will undoubtedly describe the same type of PII in different ways. For example, consumers who wish to protect their last name may use the term "last name", "family name" or "surname". All three terms are commonly used and all refer to the same piece of PII. Due to this situation, the PS must recognize these terms are equal. Similarly, different goals in the purpose element may be expressed in different ways. For example, "mapping" and "directions" may refer to the same act of generating a path from one location to another. In either case, the PS will require a Vocabulary Processing Engine which will contain a repository of terms and use ontology matching to determine which terms correspond to one another. This will be particularly important when processing a consumer's WER and comparing it to the types of information the provider states they wish to col- lect.

An intelligent core should be added to the PS. An intelligent core would allow the service to make better decisions when collecting attributes, converting policies and resolving conflicts. The intelligent core would also provide the PS with the ability to assist the provider in making decisions about its privacy policy. In a regular case, only the consumer is negotiated with when a conflict in privacy policies occurs. This is done to avoid overwhelming a provider with the possibility of multiple conflicts with many consumers occurring simultaneously. The use of an intelligent core could make decisions as to when it is appropriate to reverse this procedure and alert the provider. Take for example a situation where a single element in the provider's policy is causing more than 50% of its consumers to reject the transaction, thereby causing the provider to lose a great deal of revenue. The PS could alert the provider to this situation and allow the provider to determine if they wish to rework their policy to meet this demand.

The privacy requirements presented here also specify that upon termination of a privacy contract, a provider should produce a list of all third parties with whom it has shared the consumer's information. This list is required to inform the consumer which parties now have access to its data in order to fulfill the principle of Openness. Future work will investigate what role the PS will play in the creation and distribution of this list.

REFERENCES

[1] J. Epstein, S. Matsumoto and G. McGraw, "Software Security and SOA: Danger, Will Robinson," *IEEE of Security & Privacy*, Vol. 4, No. 1, 2006, pp. 80-83. doi:10.1109/MSP.2006.23

[2] H. F. EL Yamany, M. A. M. Capretz and D. S. Allison, "Intelligent Security and Access Control Framework for Service-Oriented Architecture," *Journal of Information and Software Technology*, Vol. 52, No. 2, 2010, pp. 220-236. doi:10.1016/j.infsof.2009.10.005

[3] R. Kanneganti, P. Chodavarapu, "SOA Security," Manning Publications Co., Greenwich, 2008. http://www.manning.com/kanneganti/

[4] G. Yee, "Privacy Protection for E-Services," IGI Publishing, Hershey, 2006. doi:10.4018/978-1-59140-914-4

[5] T. Shan and W. Hua, "Service-Oriented Solution Framework for Internet Banking," *International Journal of Web Services Research*, Vol. 3, No. 1, 2006, pp. 29-48. doi:10.4018/jwsr.2006010102

[6] J. Reagle and L. Cranor, "The Platform for Privacy Preferences," *Communications of the ACM*, Vol. 32, No. 2, 1999, pp. 48-55. doi:10.1145/293411.293455

[7] R. Dodge, C. Carver and A. Ferguson, "Phishing for User Security Awareness," *Computers & Security*, Vol. 26, No. 1, 2007, pp. 73-80. doi:10.1016/j.cose.2006.10.009

[8] B. Schneier, "Secrets and Lies: Digital Security in a Networked World," Wiley Publishing, Toronto, 2000.

[9] D. Allison, H. EL Yamany and M. Capretz, "Metamodel for Privacy Policies within Service-Oriented Architecture," *The Proceeding of the* 5*th IEEE International Workshop on Software Engineering for Secure Systems in Conjunction with the* 31*st IEEE International Conference of Software Engineering*, Vancouver, 19 May 2009, pp. 40-46. doi:10.1109/IWSESS.2009.5068457

[10] D. Allison, H. EL Yamany and M. Capretz, "A Privacy Service for Comparison of Privacy and Trust Policies within Service-Oriented Architecture," In: M. Gupta, J. Walp, R. Sharman, Eds., *Threats, Countermeasures, and Advances in Applied Information Security*, IGI Global, New York, 2012, pp. 249-266. doi:10.4018/978-1-4666-0978-5.ch013

[11] A. Cavoukian and T. Hamilton, "The Privacy Payoff: How Successful Businesses Build Customer Trust," McGraw-Hill Ryerson Limited, Whitby, 2002.

[12] G. Yee and L. Korba, "Semi-Automated Derivation and Use of Personal Privacy Policies in E-Business," *International Journal of E-Business Research*, Vol. 1, No. 1, 2005, pp. 54-69. doi:10.4018/jebr.2005010104

[13] N. Guermouche, S. Benbernou, E. Coquery and M. S. Hacid, "Privacy-Aware Web Service Protocol Replaceability," *Proceedings of the IEEE International Conference on Web Services*, Salt Lake City, 9-13 July 2007, pp. 1048-1055. doi:10.1109/ICWS.2007.143

[14] T. Erl, "Service-Oriented Architecture: Concepts, Technology and Design," Prentice Hall PTR, Upper Saddle River, 2005.

[15] A. Buecker, P. Ashley, M. Borrett, M. Lu, S. Muppidi and N. Readshaw, "Understanding Service-Oriented Architecture Security Design and Implementation" 2nd Edition, IBM Redbook, IBM Corp., 2007. http://www.redbooks.ibm.com/abstracts/SG247310.html

[16] T. Moses, "eXtensible Access Control Markup Language Version 2.0," Advancing Open Standards for the Information Society, 2005. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

[17] R. Cover, "IBM Releases Updated Enterprise Privacy Authorization Language Specification," Advancing Open Standards for the Information Society, 2003. http://xml.coverpages.org/ni2003-07-09-a.html

[18] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall and J. Reagle, "The Platform for Privacy Preferences 1.0 Specification," W3C Recommendation, 2002. http://www.w3.org/TR/P3P/

[19] Organisation for Economic Co-Operation and Development, "OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data," 1980. http://www.oecd.org/document/18/0,3343,en_2649_34255_1815186_1_1_1_1,00.html

[20] P. Beatty, I. Reay, S. Dick and J. Miller, "P3P Adoption on E-Commerce Web Sites," *IEEE Internet Computing*, Vol. 11, No. 2, 2007, pp. 65-71. doi:10.1109/MIC.2007.45

[21] V. Cheng, P. Hung and D. Chiu, "Enabling Web Services Policy Negotiation with Privacy Preserved Using XACML," *Proceedings of the* 40*th Hawaii International Conference on System Sciences*, Waikoloa, 3-6 January 2007, p. 33. doi:10.1109/HICSS.2007.207

[22] M. Lorch, S. Proctor, R. Lepro, D. Kafura and S. Shah, "First Experiences Using XACML for Access Control in Distributed Systems," *The Proceeding of the* 2003 *ACM Workshop on XML Security*, Fairfax, 31 October 2003, pp. 25-37. doi:10.1145/968559.968563

[23] A. Anderson, "A Comparison of Two Privacy Policy Languages: EPAL and XACML," Sun Microsystems, 2005. http://labs.oracle.com/techrep/2005/smli_tr-2005-147/TRCompareEPALandXACML.html

[24] A. Anderson, "Web Services Profile of XACML Version 1.0," Advancing Open Standards for the Information Society, 2007. http://www.oasis-open.org/committees/download.php/24951/xacml-3.0-profile-webservices-spec-v1-wd-10-en.pdf

[25] A. Anderson, "The Relationship between XACML and P3P Privacy Policies," Sun Microsystems, 2004. http://labs.oracle.com/projects/xacml/XACML_P3P_Relationship.html

[26] S. Dürbeck, R. Schillinger and J. Kolter, "Security Requirements for a Semantic Service-Oriented Architecture," *The Proceeding of the* 2*nd International Conference on Availability, Reliability and Security*, Vienna, 10-13 April 2007, pp. 366-373. doi:10.1109/ARES.2007.138

[27] P. Ashley, S. Hada, G. Karjoth, C. Powers and M. Schunter, "Enterprise Privacy Architecture Language," W3C Member Submission, 2003. http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/

[28] Canadian Standards Association, "Model Code for the Protection of Personal Information (Q830-96)," March 1996. http://www.csa.ca/cm/ca/en/privacy-code/publications/view-privacy-code

[29] C. Bennett, "Arguments for the Standardization of Privacy Protection Policy: Canadian Initiatives and American and International Responses," *Government Information Quarterly*, Vol. 1, No. 4, 1997, pp. 351-362.

[30] G. Yee, "Estimating the Privacy Protection Capability of a Web Service Provider," *International Journal on Web Services Research*, Vol. 6, No. 2, pp. 20-41. doi:10.4018/jwsr.2009092202

[31] Office of Security Management and Safeguards, "Further Amendment to EO 12958, as Amended, Classified National Security Information," 2003. http://nodis3.gsfc.nasa.gov/displayEO.cfm?id=EO_13292

[32] D. Bell and L. L. Padula, "Secure Computer Systems: Mathematical Foundations," The Mitre Corporation Tech-

nical Report 2547, Vol. 1, Mitre Corporation Corporation, 1 March 1973.

[33] P. Massa and P. Avesani, "Trust-Aware Recommender Systems," *Proceedings of the* 2007 *ACM Conference on Recommender Systems*, Minneapolis, 19-20 October 2007, pp. 17-24. doi:10.1145/1297231.1297235

[34] Office of Public Sector Information, "The Privacy and Electronic Communications (EC Directive) Regulations 2003," 2003. http://www.legislation.gov.uk/uksi/2003/2426/contents/made

[35] Treasury Board of Canada Secretariat, "Canadian Privacy Legislation and Policy," 2003. http://www.tbs-sct.gc.ca/pgol-pged/piatp-pfefvp/course2/mod1/mod1-3-eng.asp

[36] R. Thibadeau, "A Critique of P3P: Privacy on the Web," The eCommerce Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, 2000.

[37] L. Clement, A. Hately, C. von Riegen and T. Rogers, "UDDI Version 3.0.2," Advancing Open Standards for the Information Society, 2004. http://www.uddi.org/pubs/uddi_v3.htm

[38] M. Bartel, J. Boyer, B. Fox, B. LaMacchia and E. Simon, "XML Signature Syntax and Processing (Second Edition)," W3C, 2008. http://www.w3.org/TR/xmldsig-core/

