A Framework for Adapting Services' Design and Execution to Privacy Regulations

Alexander De Luca, Giuseppe Tropea, Georgios V. Lioudakis, Eleftherios A. Koutsoloukas, Nikolaos L. Dellas, Fabio Testa, Markus Blanchebarbe

***Abstract*—The potential impact of contemporary Information and Communication Technologies on users' privacy rights is regarded as being among their most evident negative effects. In fact, the recent advances in mobile communications, location and sensing technologies as well as data processing, are boosting the deployment of context-aware personalized services and the creation of smart environments, but at the same time, they pose a serious risk on individuals' privacy rights. In order to address this issue, this paper provides a framework for settling the services privacy friendly. The presented approach focuses on specifying a methodology for adapting services to operate on top of a middleware system that incorporates and thus, enforces the privacy regulations, preventing to a great extent the disclosure of personal data to the service providers even if personal data is collected and services are used through pervasive, ubiquitous and wireless devices.**

***Index Terms*—Service Adaptation Methodology, Privacy, Service Composition Methodology**

I. Introduction

More than a century after the first essay identifying that privacy, as a fundamental human right, was endangered by technological advances [1], never before in history the citizens have been more concerned about their personal privacy and the threats by emerging technologies [2].

A general problem with service provision is that the more elaborated a service is, which means that it provides a high quality service, the bigger is the incision in the users' privacy, because it will most probably require a larger amount of personal data. This is even more likely if commercial interests play a decisive role. This means that in sense of privacy protection, a service provider has to be considered as an enemy. Besides, there are numerous cases where the providers' practices contradict to their well stated privacy policies [3].

Thus, this paper provides a framework for adapting services in a way that they become fully compliant with regulatory requirements and provisions. According to the proposed framework, the services are deployed on top of a middleware architecture, which mediates transparently between the user and the service providers' applications, enforcing privacy using technical means. That is, the service provider has no control over the middleware. Especially, the focus is given to the methodology for adapting existing services, as well as for designing new ones with respect to the privacy principles. To that respect, the services are decomposed and executed with the minimum possible set of personal data, while the execution of their critical parts is undertaken by the underlying system, which prevents the disclosure of the considered data. In essence, the service providers are forced to adopt a specific privacy policy, which is based on and reflects the privacy legislation.

The rest of this paper is organized as follows: Section II provides some insights on the considered middleware architecture that enforces privacy protection. Section III describes the concept and the corresponding procedures for modeling the rules that regulate the services' provision using an ontology. Section IV presents the methodology for both new services' authoring and existing services' adaptation with respect to the privacy framework, while Section V provides a characteristic use case example. The paper concludes in Section VI, with a few summarizing remarks.

II. Reference Architecture

The services in the considered framework are adapted to operate over a middleware architecture, namely the D-Core, which acts as a three way privacy mediator between the law, the users and the service providers. This section provides an abstract overview of the system; the reader may refer to [4] or contact the authors for the detailed specification of this work.

A high level components' structure of the middleware architecture is highlighted in Fig. 1. The architecture is based on the concept of the D-Core Box, which constitutes a privacy proxy installed at the service provider's premises but totally controlled by the Privacy Authority. A D-Core Box constitutes the "edge" module of the D-Core infrastructure and the border between the service provider's applications and

Manuscript received January 15, 2007. This work was supported in part by the IST Project DISCREET (IST-FP6-27679).

A. De Luca is with the University of Munich, LMU (Amalienstraße 17, 80333 Munich, Germany; e-mail: alexander.de.luca@ifi.lmu.de).

G. Tropea is with CNIT (Italy, e-mail: gtropea@unict.it).

G. V. Lioudakis, E. A. Koutsoloukas and N. L. Dellas are with the National Technical University of Athens, School of Electrical and Computer Engineering (9 Heroon Polytechniou str., 15773 Athens, Greece; e-mail: {gelioud, lefterisk,ndellas}@telecom.ntua.gr).

F. Testa is with Starbeam S.r.l. (Viale Regione Siciliana 7275 Nord-Ovest, 90146 Palermo, Italy; e-mail: fabio.testa@starbeam.it).

M. Blanchebarbe is CEO of Eyeled GmbH (Science Park 1, 66123 Saarbrücken, Germany; e-mail: blanchebarbe@eyeled.de).

the D-Core. In essence, the services are deployed on top of the D-Core Box and therefore, are adapted to this direction.

The D-Core Box serves as the entry point to a service. Any interaction between a user and a service provider is filtered by the D-Core Box. All data provided by a user as well as all the data collected by a provider without the active participation of a user (e.g. from sensors) are stored inside the D-Core Box, into the encrypted Personal Data Repository (PDR). The storage may be short-time (e.g. immediate service provision) or long-time (e.g. services that require data archives).

In order for any personal data to be disclosed to the provider, the corresponding decision is taken by the Policy Engine (PE) that the D-Core Box incorporates. To that respect, the PE considers the legislation, as specified in the Ontology of Privacy (Section III), as well as the user's privacy preferences. The later are defined by the user and transmitted as metadata together with the data, using a special data structure, the Privacy Lock. That is, no data reach the provider directly; the D-Core Box proxies the traffic and disseminates to the provider only the data specified by the legislation and the user's preferences. Additionally, the D-Core Box embeds all the necessary functional modules for interacting with the user whenever a notification or consent is demanded for any action on personal data, without the provider's participation.

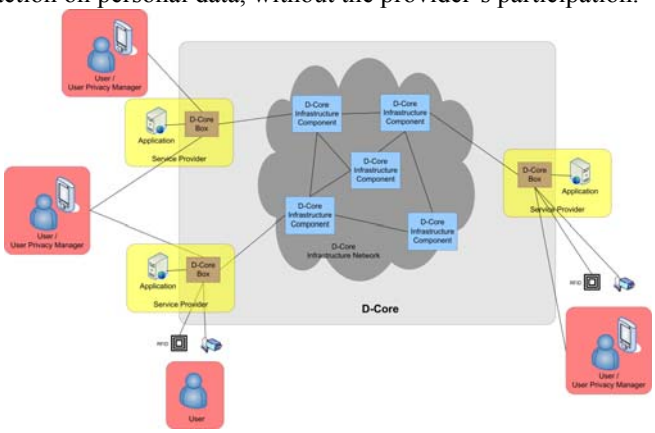

Fig. 1. Middleware architecture.

The D-Core Box's internal structure is complemented by Embedded Operations and Embedded Services modules. They undertake the execution of privacy sensitive data processing tasks and whole services' parts, respectively, in order to further reduce the amount of disclosed data. A typical Embedded Operators' functionality is the filtering of the data precision prior to their disclosure (e.g. the translation of an exact location to more abstract terms, the blurring of human faces in a surveillance video, the transformation of an age to the appropriate age range, etc). Embedded Services concern the internal execution of standard service components that concern identifiable data (e.g. service's charging mediation).

The communication between the service provider and the corresponding D-Core Box is performed by means of a dedicated API. In order for the provider to request and receive personal data for the service provision or make use of Embedded Operators and Services, the corresponding functional methods exposed by the API are called, with the provision of the necessary credentials.

The other architectural components of the framework are the User Privacy Manager (UPM) which is the user-side component and the Infrastructure Components (ICs) that form the Infrastructure Network for the purposes of D-Core Boxes' online monitoring, management and Lawful Interception. The UPM is created in a way, that it can be easily embedded into wireless and thus mobile devices, which is important, since we are dealing with ubiquitous and pervasive computing environments with their characteristic of being everywhere.

III. Ontology of Privacy

The formal modeling of the privacy legislation inside the considered middleware architecture is achieved using a semantic information model that associates personal data and services with explicitly defined regulatory rules. To that respect, the approach taken is to express any related information by means of an ontology, namely the Ontology of Privacy. This section describes this semantic knowledge base.

In order to associate the personal data with specific processing tasks, the identification of the particular type of each personal data item is necessary. Moreover, in order to define the appropriate rules that will regulate the processing of a personal data item with respect to the purpose for which the information is provided by the user or requested by the service provider, a similar taxonomy of the provided services must be present. These taxonomies constitute separate sub-graphs of the ontology, having as root elements, respectively, the `PersonalObject` and `ServiceObject`. Therefore, the Ontology of Privacy provides a detailed vocabulary of personal data types and services' types, structured in a hierarchical way with well defined inheritance rules that enables the system to associate all privacy related decisions to semantically specified notions.

The policies that are needed for regulating the disclosure of personal data to service providers and their consequent processing form a third sub-graph in the Ontology of Privacy, having as root element the `PolicyObject`. Subclasses and instances of `PolicyObject` reflect primarily regulatory requirements like data retention periods, user notifications and user consent requirements. Instances of these policies need to be assigned to pairs of personal data type instances and service instances, so that a reasoner can infer the required policies when a specific service requests a specific data type.

The vision is that the ontology should be as detailed as possible in terms of the types of personal data and services, so that the widest range of services and situations when personal data are involved can be covered. Regarding services, the goal is the creation of a classification similar to and as detailed as the European Common Procurement Vocabulary [5]. Therefore, all possible services should be included there. To that respect, for the creation of the Ontology of Privacy, we consider a negotiation and consultation procedure that involves the Privacy Authority and the service providers along

with the respective Chambers. The outcome of this procedure is the hierarchical detailed definition of the services, the relations between them in terms of features' inheritance and the specification of the rules that govern their provision: necessity of data, retention periods, notifications and consents, data filtering by means of Embedded Operators, services' parts execution assignment to Embedded Services etc. The Ontology of Privacy definition is performed by the Privacy Authority using the ICs and is disseminated to all the D-Core Boxes through the Infrastructure Network.

IV. Service Adaptation

Service adaptation is a rather general term, which simply means the modification of a service as well as the creation of new services to interoperate with the proposed privacy respectful middleware. The proposed methodology can be seen as a step by step manual or guidelines for changing or creating services that run on our proposed middleware. That is, the outcomes of this approach are services that will not be able to harm the users' rights nor their privacy. Therefore, this service adaptation approach is a way of enforcing privacy conformity already during the design of a service.

As Fig. 2 shows, in the design phase of a service, service adaptation takes places after the general decisions about the service have been committed. It is influenced by the middleware and creates the necessary outcome to finally realize the service.

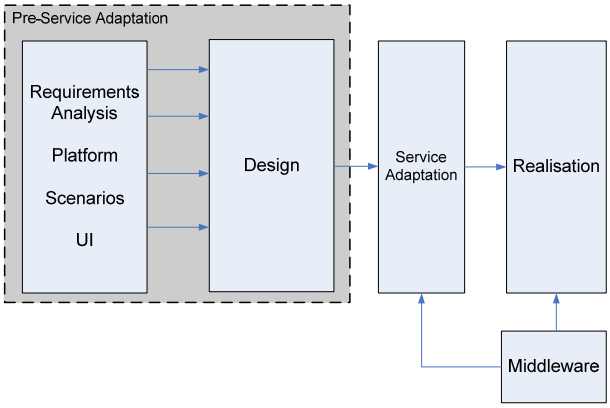

Fig. 2. Service Adaptation classified in the general design process of services.

When talking about adaptation, it is obvious that there must exist some previous state (in this case of a service), which is modified by the adaptation. In our work, there exist two possible starting situations for service adaptation:

- New service design (Service Authoring): In this case, service adaptation starts from a functional description of a service and a design involving functional blocks which only deal with the functionality of the service.
- Legacy service improvement: In this case, the starting point is a running implementation of a service.

To explain the process of service adaptation and how it can ensure the correct use of personal data by a service provider, we will outline the methodology for the creation of new services. In a second step, the differences when adapting existing services will be explained.

A. New Service Authoring

For the creation of privacy respectful services on top of our middleware, two overall steps need to be performed: A Privacy Analysis and the Service Development. These and their sub-phases, as depicted in Fig. 3, will be explained in the following sections.

1) Privacy Analysis

The privacy analysis is a collection of theoretical but mandatory steps to create a basis for the practical part of the adaptation, the service development itself.

a) Ontology Categorization

At first, a service provider has to categorize his service within the previous mentioned Ontology of Privacy. Non technical speaking, this ontology is a human readable version of the whole ontology including the regulatory rules. That is, the service provider will be aware of all the rules (legislation) that apply to his service. As a positive side effect of this categorization, the regulatory and legal work has already been done for the service provider.

In the case that the categorization is not possible – since there was no appropriate service listed in the ontology – the service provider has to contact the Privacy Authority responsible for it. The Privacy Authority has the possibility to modify the ontology and distribute it in the middleware; that is, updating the whole system. After that, the service will be categorized and be ready for development.

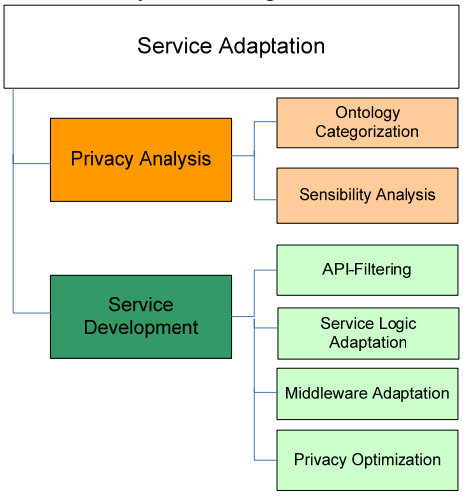

Fig. 3. Service Adaptation Methodology; Main- und sub-phases. The order is top to bottom.

b) Sensibility Analysis

Compared to the previous step, sensibility analysis is a rather general process. Even though a service provider may skip it, it is of great importance. The idea is to provide a knowledge base for service providers, to make them aware that using less sensitive user data than allowed will make them more attractive to users, which can then result in a competitive advantage.

The most important part of the sensibility analysis is a knowledge base of consequences. That is, the consequences occurring if a specific operation (like performing age verification) are done. Mainly, this is the privacy awareness user interfaces, created by the middleware to keep the users aware about how their personal data are handled. The service provider has no influence on them but not using the data.

2) Service Development

With this basic knowledge, the second and last main phase of our service adaptation approach can be performed. It is called service development and includes all steps necessary for the implementation of the service.

a) API-Filtering

As mentioned before, the result of the proper categorization of a service within the ontology will make legal information available to the service provider as well as the data his service is allowed to handle. Still, there is the issue of a service provider dealing with the middleware.

Thus, the API-filtering step has been included into our service adaptation methodology. It is a supportive step, automatically creating a personalized manual for building the specific service. Therefore, the output is a step by step manual of how to adapt the service to the middleware. For example, this includes the APIs that have to be used, the Embedded Operators and Services, etc.

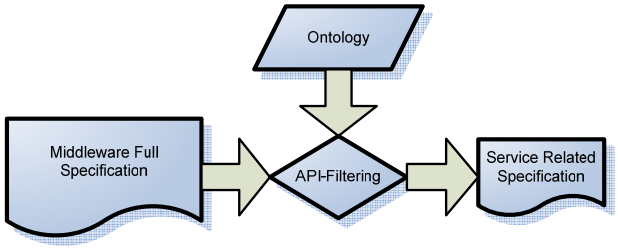

Fig. 4. API-Filtering process. Depending on the categorization of the service within the ontology, the full middleware specification is filtered and only the service related parts are kept for the output specification.

For enabling the service provider to use less detailed personal data, the outcome of the API filtering includes references to Embedded Operators and Services available for this purpose as alternatives or as mandatory tools to use.

The consequences of using specific data instead of an Embedded Operator, where applicable even if not mandatory, are stated in these guidelines as well, to enable the service provider to make the appropriate decision. For instance, this partially is the before mentioned knowledge base about privacy awareness screens, displayed to the user. That is, this is a highly personalized implementation sensibility analysis. The whole process is outlined in Fig. 4.

Using API-Filtering, a service provider does not have to learn and read the whole middleware specification but only the parts needed to implement his services. Additionally, it already provides a step by step implementation manual.

b) Service Logic Adaptation

With the manual resulting from the previous step, the service providers will be able to adapt their service logic to the middleware. This is important, since a huge amount of the service logic is executed within the privacy respectful middleware in order that privacy sensitive data are not disclosed. This means that for example sequence diagrams (UML diagrams in general) will be service/D-Core diagrams. This requires explicit knowledge about the middleware components (e.g. Embedded Operators) at service provider's side. This knowledge is available because of the previous step.

c) Middleware Adaptation

At this point of the development process, all required information for the implementation of the service is available. Now, following the guidelines, the service logic will be implemented. Therefore, this step can be described as a middleware respectful service implementation. Generally speaking, during this step, the source code of the service is written.

d) Privacy Optimization

If, at any time during the development process, the service provider realizes that the service also can be implemented using less information and he is willing to create a more privacy respectful service, privacy optimization takes place. This means, the process will restart from one of the earlier adaptation phases. The service adaptation phase can even restart all over again. Consequently, service adaptation is an iterative privacy development process.

B. Existing Service Adaptation

In the previous sections, the service adaptation phases for the creation of a new service upon the middleware have been described. The second possible starting point is a running implementation of a service that needs to be adapted to the middleware. Usually, such a service does not consider any privacy related functionality. The question is how a service needs to be modified to interoperate with the middleware i.e. to become fully privacy respectful. It is important, to provide a decomposition methodology and the respective guidelines for this purpose. In the most parts, the adaptation of existing services and new services overlap.

The whole first step, which is the privacy analysis, is the same. For the second part, service development, obviously, there are differences. While API-Filtering does not differ, service logic adaptation does. Normally, for an existing service, sequence diagrams and other specifications already exist. This means instead of referring to creation of service/D-Core sequence diagrams, a conversion from the service's existing sequence diagrams to service/D-Core diagrams needs to be performed by the service provider. This means identifying the components that become unusable, since their functionality has to be moved to the middleware and the corresponding Embedded Operators and Services.

Finally, the middleware adaptation is performed in a slightly different way compared to the creation of new

services, since much of the functionality that is not a part of the middleware may already be implemented and needs to be adapted to the new diagrams only.

V. Service Adaptation Example

A new advertisement service is developed by a number of shops grouped under a shopping portal: the users register themselves by entering some categories of goods they are interested in. When a user browses the shop catalogue, objects of interest are proposed by the shop's web site. Our proposed middleware is able to guarantee that the customer enters the shop and buys some garden tool without the operator getting to know his identity and the shop ever knowing that a user interested in "garden tools" was there at that time.

A. Ontology categorization

To cope with this service, inside the Ontology of Privacy, a generic Shop class and a CatalogueBasedService class are provided, under the ServiceObject graph of the ontology.

As stated before, as a first step in adapting the new advertisement service to the framework, a privacy analysis is carried out, so that the aforementioned classes are found to describe the service and grouped together into a more specific CatalogueBasedShopping class which inherits its fundamental PersonalObjects (for example Age, Address, Shopping_interests, ShopItem, ...) and PolicyObjects from its parents.

Furthermore, we assume that law explicitly forbids shopping services to store and use any personal information about customers whatsoever. This is hard-coded inside the ontology graph for our new service.

B. Service development

1) API selection

When the functionality of the service is examined, it is easy to understand that the Payment embedded service and the EmailSending embedded service must be employed, to ensure that credit card information is managed according to the law and further, that the service can send informative emails and protect customers' email addresses from unwanted messages at the same time. We can say that the embedded service Payment is to be used because an information (credit card data) required by the service, is now confined inside the D-Box (as required by the law), and not accessible by the Service Provider. As a rule, embedded services' usage can thus be automatically discovered during the API selection step by modeling all service functionalities and simulating all data flows with and without the D-Box.

This information is provided to the service provider after the API-Filtering has been performed.

2) Service logic adaptation

After the necessary embedded services have been pointed out, the data structures and main functionalities of the service must be written in order to guarantee that:

- personal data structures like database of customer information are kept within the D-Core Box and arranged into the Personal Data Repository;
- the PDR of the D-Box is actively connected to the service database installed at the service provider;
- functionalities (including for example credit card reader at the shop's site and operator's software interface and invoice printing) make use of the Payment and EmailSending embedded services of the D-Box, so that personal data never reaches the service provider, unless data disclosure is allowed by a specific policy.The customer, when browsing the catalogue at home will be able to communicate with the service by means of his browser, but the communication goes through the D-Box proxy and is mediated by the user's User Privacy Manager installed at the customer's PDA or PC.

VI. Conclusion

In this paper, we described a framework for creating privacy respectful services. This is achieved by providing a middleware, which service providers have to use to deploy their services. The advantages are manifold. For the users, a trustable platform is provided, saving their data and informing them about its usage, thus, being privacy aware. Then again, service providers do not have to handle or create privacy mechanisms for the users, since they are already included in the middleware. Therefore, building privacy aware services is reduced to following simple guidelines and assigning the privacy work, like user notifications and consent, to the middleware. This means, one of the main obstacles for designing privacy respectful service, namely, huge additional work, is eliminated by our framework.

Furthermore, our proposed service adaptation approach contains additional mechanisms to ease the service implementation. For example, the API-Filtering provides the complete and exact API that is to be used for implementing a specific service. Thus, it saves production time and as its outcome it provides privacy respectful services.

Acknowledgment

This work is partially supported by the European Union, in the framework of the FP6 – IST Project DISCREET.

References

[1] S. D. Warren and L. D. Brandeis, "The Right to Privacy", *Harvard Law Review*, Vol. IV, No. 5, pp. 193–220, Dec. 1890.
[2] The European Opinion Research Group, "European Union citizens' views about privacy", *Special Eurobarometer 196*, Dec. 2003.
[3] U.S.A. Federal Trade Commission, "Eli Lilly Settles FTC Charges Concerning Security Breach", *File No. 012 3214 In the Matter of Eli Lilly and Company*, Jan. 2002.
[4] C. Kiraly et al., "System Architecture Specification", *IST DICREET Deliverable D2201*, October 2006, available at http://www.ist-discreet.org/Deliverables/D2201.pdf
[5] European Parliament and Council, "Regulation 2195/2002/EC of the European Parliament and of the Council on the Common Procurement Vocabulary (CPV)", *Official Journal of the European Communities*, No. L 340, pp. 1–562, Dec. 2002.
[6] IST Project DISCREET, home page: http://www.ist-discreet.org.