PrivAPP: AN INTEGRATED APPROACH FOR THE DESIGN OF PRIVACY-AWARE APPLICATIONS

Tania Basso[1], Leonardo Montecchi[2], Regina Moraes[1], Mario Jino[1], Andrea Bondavalli[2]

1 State University of Campinas (UNICAMP), Campinas, Brazil
{taniabasso@ft, regina@ft, jino@dca.fee}.unicamp.br

2 University of Firenze (UNIFI), Firenze, Italy
{lmontecchi,bondavalli}@unifi.it

Abstract. *Nowadays, personal information is collected, stored, and managed through web applications and services. Companies are interested in keeping such information private due to regulation laws and privacy concerns of customers. Also, the reputation of a company can be dependent on privacy protection, i.e., the more a company protects the privacy of its customers the more credibility it gets. This paper proposes an integrated approach which relies on models and design tools to help the analysis, design and development of web applications and services with privacy concerns. Using the approach, these applications can be developed consistently with their privacy policies in order to enforce them, protecting personal information from different sources of privacy violation. The approach is composed of a conceptual model, a reference architecture, and a UML Profile, i.e., an extension of the UML for including privacy protection. The idea is to systematize the privacy concepts in the scope of web applications and services, organizing the privacy domain knowledge and providing features and functionalities that must be addressed to protect the privacy of the users in the design and development of web applications. Validation has been performed by analyzing the ability of the approach to model privacy policies from real web applications, and by applying it to a simple application example of an online bookstore. Results show that privacy protection can be implemented in a model-based approach, bringing values for the stakeholders and being an important contribution towards improving the process of designing web applications in the privacy domain.*

Keywords: privacy; reference architecture; UML Profile; conceptual model; web application.

1. Introduction

Web applications and web services are extremely relevant technologies nowadays, supporting a wide range of services such as e-commerce, e-banking, e-government, and others. Usually, to access these services, the users, customers, and business partners need to provide private information such as addresses, social security IDs, and credit card numbers. Furthermore, modern web sites and service providers can gather, automatically, information related to users' activities as, for example, usage pattern or approximate location. Once this information is made available, how it is actually handled is no longer under the control of its owner. Companies and organizations want to be able to gather, data mine and share this information efficiently and without putting their reputation at risk. Customers want choices and ease of access, without losing their privacy. Thus, relevant concerns arise from both sides with respect to privacy.

In the web applications and services context, privacy refers to privacy of electronic information. A known definition for privacy of information is presented by

Westin [1]: "*privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others*". Furthermore, Wang et al. [2] state that "*privacy usually refers to personal information and the invasion of privacy is usually interpreted as the unauthorized collection, disclosure, or other use of personal information as a direct result of electronic commerce transactions*". More recently, Bertino et al. [3] says that privacy is "*the right of an entity to be secure from unauthorized disclosure of sensible information that is contained in an electronic repository*".

Nowadays, companies and organizations have great interest in protecting the privacy of information manipulated by their web applications and services. The two main reasons are privacy laws and competitive differentials. In the European Union (EU) [4], Canada [5] and Australia [6], for example, regulations for the protection of personal identifiable information have been created, and some of them cross industry sectors. The United States has taken a sectorial approach, enacting separate regulations for health care [7], finance [8] and protection of children's data [9]. In either case the goals are clear - to give better protection to personal information, i.e., the companies and organizations that hold these data have the obligation and responsibility to protect them. Regarding the competitive differentials, the reputation of a company can be strictly dependent on privacy protection, i.e., the more a company protects the privacy of its customers and business partners, the more credibility it gets. Proof of this statement is the least research developed by Truste [10], where 91% of the interviewed Internet users said that they avoid doing business with companies that they do not believe protect their privacy online. Facing this scenario, it is of utmost importance to construct web applications and services that protect privacy.

A recurring problem in constructing web applications and services that protect privacy is the insufficient attention to modeling and documenting privacy features. The lack of integration of privacy requirements in the application design and development makes privacy protection difficult, since privacy mechanisms have to be devised based on both the application and the privacy policy. The lack of privacy reference models impairs the standardization and evolution of systems with respect to privacy concerns. From a more practical perspective, several issues are still open: there are no UML profiles addressing privacy; reference architectures are limited regarding privacy concerns; privacy enforcement mechanisms are not addressed in conceptual models. Existing solutions address only part of these aspects and are not integrated, since they do not derive from the same conceptual model. For example, most solutions do not have a compatible UML profile to be used concomitantly.

To address these problems, we propose PrivAPP, which is an integrated approach to guide the design of privacy-aware applications. Its goal is to provide a better understanding of the privacy domain and facilitate the modeling and development of privacy-aware web applications and services. By *integrated* we mean that the approach is composed of a set of interrelated resources (conceptual model, reference architecture, UML profile), which address common privacy concepts at different stages of the development process. This interrelation provides a better compatibility of the models created during the design process, allowing a more complete and consistent documentation of privacy-aware applications.

The approach is composed of a **Privacy Conceptual Model**, a **Privacy Reference Architecture,** and a **Privacy UML Profile**. Briefly, the Conceptual Model is composed by elements that represent privacy concepts and their relations, in an organized way. Its goal is to specify and organize the privacy domain knowledge. The Reference Architecture describes the features and functionalities that must be addressed

during the development to protect the privacy of the users. The elements of the conceptual model are distributed through the layers that they can be implemented. The goal of the Reference Architecture is to guide the development of concrete architecture models that can facilitate the development of privacy-aware technology. Finally, the Privacy UML Profile extends the UML language [11] to incorporate privacy concepts. With this extension, UML diagrams can be used for the development process of privacy-aware applications and services. It is used to document the existence of elements of the conceptual model in the architecture, in order to reduce ambiguities in the solution.

The proposed approach systematizes the privacy concepts in the scope of web applications and services, providing a model of the domain concepts that are required for modeling views of the system where privacy management and protection are applied. Furthermore, it can be used in a modular way, i.e., it is possible to use, for example, only the Reference Architecture or only the Privacy UML Profile, depending on the requirements or the needs. The focus of the approach is not the requirements elicitation, but rather documenting how privacy policies can be enforced in the software architecture.

Concerning evaluation, we developed an application example where the proposed approach is applied in the design (and, subsequently, implementation) of data privacy protection features for a web application that represents an online bookstore. Also, we evaluated our approach through an empirical study, where a set of privacy policies from relevant companies were analyzed, verifying the cases in which the elements from PrivAPP are able to model these policies and solutions for enforcing them. The results of these evaluations were promising: on one hand, it was possible to implement privacy protection using a model-based approach in a simple application example; on the other hand, we verified that most of the concepts included in the policies were represented in PrivAPP's conceptual model.

The paper is organized as follows. Section 2 introduces some background and relevant related work in the scope of web applications and services. Section 3 presents the proposed approach. We describe thoroughly its three components: the Privacy Conceptual Model, the Reference Architecture and the UML Profile. Section 4 presents the application example cited above. The concrete architectures, UML diagrams, and overview of the implementation are shown in this section. In Section 5 the evaluation process for the proposed approach is presented. Finally, Section 6 presents the key conclusions about this work.

2. Background and Related Work

To establish the background for privacy and web applications we sought for related works, focusing on those that provide some privacy models and UML extensions in this context. The result of this study is presented in this section. We start describing how web applications handle privacy nowadays. During this description we highlighted some elements we believe it is important to represent in our approach. Then we discuss some privacy reference models that address privacy concerns. The few existing solutions reinforce the novelty of PrivAPP. Next, we introduce the concept of reference architecture, its importance and its goals as part of the proposed approach. We also present some related work regarding privacy concrete architectures. Finally, we introduce the importance of extending the UML with a profile, its role in PrivAPP, and some related works, indicating the novelty of the profile we propose.

2.1. Privacy Context and Concepts

Nowadays, to acquire some product or service through companies' online web sites, the most common approach we face is a presentation of a privacy policy, usually before sending our personal information. A **privacy policy** is a document that explains how an organization handles any customer, client or employee information gathered in its operations [12]. Usually, they specify personally identifiable information (PII) that is gathered (such as name, address, credit card number, etc.) as well as information about the activities the users perform on Internet, (such as visited websites, search strings, etc.). The policies also usually explain if data may be left on a user's computer (through cookies or similar technologies), disclosed, shared with or sold to third parties (i.e., other partners, companies and organizations) and, if so, for what purpose. The privacy policies should be written based on privacy laws, principles and regulations, addressing requirements across geographical boundaries and legal jurisdictions.

Policies can be seen as a set of statements**,** where a **statement** is a portion of text related to the same topic and having a complete meaning, describing one of the policy rules. At this point it is important to mention that, in our view, privacy-related policies can be organized in a hierarchy: highest-level policies are described in natural language; lowest-level policies are specified in machine-readable format, and used by the application itself to e.g., perform access control. Reproducing high-level statements in machine-readable statements is a very difficult task due to the semantics involved. The lower the level, the greater is the loss of semantics. The authors of [13] identified seven layers in which privacy policies are implemented: legal, business, process, application, information, system, device, and network. The scope of the work in this paper ranges from the legal to the application layer, i.e., we want to deal with higher-level policies in order to keep their semantics.

According to common privacy principles [14][15], statements describe the processes of **collection**, **usage**, **retention** and **disclosure** of private data. **Private data** can be classified in two types: **personal information**, which refers to information that the user provides to the system (e.g. name, address, credit card number, etc.) and **usage information**, which refers to data the system collects (e.g. links accessed, user's actual location, search strings, etc.).

A statement may specify: which data would be obtained (**collection**), how the private data will be used and for which purposes (**usage**), the period the data will be retained and what will be done with this data after the stated period (**retention**), which data will be disclosed and to whom, i.e., the **recipient** of the data (**disclosure**). Usually the recipients are third parties, i.e., other partners, companies and organizations, external to the organization which holds the data. It is important to mention that, according to the referred privacy principles, personal data should not be disclosed, made available or used for purposes other than those specified, except with the consent of the **data subject** (i.e., the owner of the data) or by the authority of law. Also, the **purposes** for which personal data are collected should be specified not later than at the time of data collection. Less frequently, statements can express some **condition**, describing prerequisites to be met before any action can be executed.

In practice, nowadays, the privacy policy is displayed and gives the users or visitors only the option to agree or disagree with this whole policy. If they do not agree, they cannot perform the desired task. Most of the times it is not possible to users to express their privacy preferences in a more complete way, expressing their **consent**, i.e., agreeing or not with parts of the privacy policy (statements) and not with the whole one.

We cannot talk about privacy preferences without mentioning the two main works that address this issue: P3P (Platform for Privacy Preferences Project) [16] and EPAL (Enterprise Privacy Authorization Language) [17]. P3P is a standard that allows websites to declare, in a standard format, the intended use of the information they collect about users, such as what data is collected, who can access those data and for what purposes, and for how long the data will be stored. Similarly, the EPAL allows enterprises to formalize their privacy practices into policies that define the categories of users and data, the actions performed on the data, the business purposes associated with the access requests, and obligations incurred on access. The biggest problems with this two technologies is that, even though they provide standard means for enterprises to define privacy promises to their users, they do not provide any mechanism to ensure or provide evidence that these promises are consistent with the internal data processing.

The **enforcement** of a privacy policy is not a trivial task. It is necessary to adopt resources that can be used in order to enforce the privacy policy statements, respecting the data subjects' preferences. Privacy is still a very abstract concept, with abstract conceptual elements. Privacy policies are defined using textual natural language, which makes their enforcement difficult. Currently, the way over used to enforce policies is to rely only on existing security-related resources as, for example, access control, auditing, cryptography, among others. However, these resources are not enough to protect privacy as a whole, because, in our viewpoint, privacy goes beyond security: it is possible to have poor privacy and good security practices. We believe the key is to combine different methods to overcome their individual limitations and our proposed approach goes in this direction.

2.2. Privacy Conceptual and Reference Models

As far as we know, there is no other integrated approach to deal with privacy, which provides enforcement elements in an architectural level and UML resources for documentation. However, a few efforts for designing reference models for data privacy exist. Cherdantseva and Hilton [18] present a Reference Model of Information Assurance & Security, which endeavors to address the recent trends in the IAS (Information Assurance & Security) evolution. The model incorporates four dimensions: Information System Security Life Cycle, Information Taxonomy, Security Goals and Security Countermeasures. The goal is to provide the understanding and communication among stakeholders through informal visual representation. Although security is strictly related to privacy, the focus of the paper is in data security (with security goals as accountability, authenticity, availability, confidentiality, integrity, non-repudiation) and considers few privacy aspects, only mentioning that system should obey privacy legislation and it should enable individuals to control, where feasible, their personal information.

The work of Sathiyamurthy [19] defined a conceptual model that they called an "holistic privacy archetype", which provides a pragmatic approach for the business to manage and stay abreast of growing regulatory and fiduciary requirements. The model is divided in three main layers (business process layer, strategy and governance layer, and operational layer) and was applied to a financial business model to describe its capabilities. However, due to being more enterprise-focused (business-processes oriented), the model neglects more specific characteristics of web services and applications as, for example, privacy policies definition and management.

The Privacy Management Reference Model and Methodology (PMRM) [20] is an OASIS specification that provides a conceptual model and a methodology for

understanding and analyzing privacy policies and their privacy management requirements. Also, it allows selecting technical services that must be implemented to support privacy controls. The model is based on a non-normative working set of operational privacy definitions and the privacy requirements are defined through use cases. Although this is a recent privacy reference model, it considers only intrinsic characteristics (core) of privacy, i.e., it did not directly incorporate privacy requirements related to different sources of privacy violation, in a broader privacy context. Also, it is generic and do not specify resources for enforcing privacy policies.

Other approach that can be considered is in the field of trustworthiness-by-design. Mohammadi et al. [21] proposes enhancing a broad spectrum of general software development methodologies to incorporate the consideration of trustworthiness, which would include privacy. However, trustworthiness by design approaches involve different quality attributes (security, reliability, dependability, etc.) and the scope is more focused on processes than models. Also, to the best of our knowledge, there is no approach of trustworthiness by design that relates privacy policies to ways of enforce these policies in the software architecture. We believe our approach could complement trustworthiness-by-design approaches, by providing a methodology to document privacy aspects during the software modeling and development phases, with a focus on the software architecture.

Even if the approach we propose employs some security resources, its focus is on privacy concerns. Key elements of privacy domain (as privacy policy, user's preferences) are represented, and the *enforcement* elements are based on different sources of privacy violation, providing a set of solutions that can be used to protect privacy. Also, it considers the scope of web applications and services, where sharing private information in this ubiquitous environment deserve special attention. Some elements in our approach deal with this problem.

2.3. Privacy Requirements Elicitation

Studies are being conducted on the research field of security and privacy requirements elicitation. The KAOS framework [22] states system goals as a strategy to derive complete and consistent requirements through a formal refinement process. Its model is a set of interrelated goal diagrams that have been put together for tackling a particular problem. It describes the problem to be solved and the constraints that must be fulfilled by any solution provider. Whereas KAOS primarily focuses on functional requirements our work allows modelers to incorporate non-functional requirements, focusing on privacy, and suggests some solutions for enforcing these requirements.

Mouratidis et al. [23] proposes the Secure Tropos, which is an extension of the previous Tropos methodology [24] for requirements engineering, with new concepts to cover security modelling. It includes security constraints (restrictions related to security issues), secure dependency (describes one or more security constraints that must be fulfilled for a dependency to be satisfied objectives), and secure entity (represents any secure goal/task/resource of the system). These concepts produce different kinds of diagrams, which are used as input to the later activities. However, Secure Tropos focuses explicitly on security issues and extensions for addressing privacy issues are required.

The PriS method [25] elicits privacy requirements in the software design phase. Privacy requirements are modeled as organizational goals. Furthermore, privacy process patterns are used to identify system architectures, which support the privacy requirements. The PriS method starts with a conceptual model, which also considers

enterprise goals, stakeholders, privacy goals, and processes. It is based upon a goal-oriented requirements engineering approach. Our work focuses on privacy policies and architectural layers as a foundation for the privacy analysis, while the PriS method uses organizational goals as a starting point.

The focus of the approach we propose in this paper is not on privacy requirements elicitation. However, PrivAPP can help in this process. Results of using the approach can be employed to help stakeholders understand the privacy domain, and the documentation specifying how privacy policies can be enforced can be used as input of the process. Furthermore, the previous works we mentioned propose their own processes and diagrams. The use of UML language can collaborate with these models, unifying these diagrams and avoiding spending time learning new and different symbols.

2.4. Reference and Software Architectures

The software architecture constitutes the backbone of any successful software system. Decisions made at the architectural level directly enable, facilitate, or interfere with the achievement of business goals as well as functional and quality requirements. A reference architecture refers to a special type of software architecture that captures the essence of the architectures of a set of software systems in a given domain, i.e., an abstraction of concrete architectures. The purpose of reference architectures is to serve as guidance for the development, standardization, and evolution of systems in that domain, as well as guarantee the interoperability between systems and between components of systems [26].

The two main contributions regarding privacy reference architectures which are related to the scope of our work are the standard ISO/IEC 29101 [27] and the work of Shin *et al.* [28]. The ISO/IEC 29101 [27] describes best practices for the technical implementation of privacy requirements. The standard covers the various stages of data life cycle management and the required privacy functionalities for protecting data, as well as the definition of the roles and responsibilities of all the involved parties. Similarly, Shin *et al.* [28] present a privacy reference architecture as a security model for the management of personal information in its lifecycle. They divide the lifecycle of personal information into four stages and introduce the steps of the personal information processing performed at each stage. The architecture is based on the three types of actors involved in PII processing: principal, controller and processing.

Although the two works introduced above [27][28] integrate privacy considerations into technical design, their enforcement components are limited, considering few sources of potential privacy violation of the system. The goal of PrivAPP's reference architecture is, as well as the whole approach, to consider privacy enforcement for different sources of privacy violation. Furthermore, its three-layer architectural style and its privacy layer are more related to the context of web applications and services.

Besides privacy reference architectures, there are two important concrete software architectures that deserve to be mentioned: the IBM Tivoli Privacy Manager [29] and the HP Privacy-Aware Access Control architecture [30]. IBM Tivoli Privacy Manager [29] provides an enterprise-wide system that enables a company to use PII it collects according to the principles of Fair Information Practices [31] and to monitor and enforce its compliance with those principles. More focused on access control context, the HP Privacy-Aware Access Control architecture [30] helps the enforcement of privacy policies for personal data stored by enterprises. Both these architectures

[29][30] were used in the evaluation process of the privacy reference architecture that is part of our approach. A detailed discussion on how our reference architecture relates to them is reported in [32].

2.5. UML and Profiles

A UML profile is an extension of the UML metamodel containing specializations for a specific domain, platform, or purpose; a UML profile is defined through the Profile Diagram. Profiles are defined using *stereotypes*, *attributes*, and *constraints*. *Stereotypes* are the main construct in a profile; a stereotype is an extension of an existing UML metaclass, possibly defining a set of additional *attributes* (i.e., properties). A stereotype can also be an extension of another stereotype. When a stereotype is *applied* to an instance of the metaclass it extends, values can be specified for its attributes. A UML profile may also define additional *constraints*, i.e., statements that need to be satisfied for a model to be well-formed according to the profile.

There are UML Profiles for different domains. In [34], for example, a model-driven development approach was introduced to the development of access control policies for distributed systems. Hsu [35] defines an UML profile for different Web 2.0 applications, including Web 2.0 mashups and Web 2.0-based context-aware applications. However, to the best of our knowledge, there are no UML Profiles for the privacy domain. The works which are more closely related to our proposed UML profiles are for the security domain. Yet they are far from representing privacy domain concepts. Cirit and Buzluca [36] proposed a UML Profile for Role-Based Access Control (RBAC), with which access control specifications can be modeled graphically together with problem domain specifications from the beginning of the design phase, making it possible to extend security integration over entire development process. The work of Jürjens [37] presents UMLsec, a profile that allows expressing security relevant information within UML diagrams. The profile encapsulates the knowledge of recurring security requirements of distributed object-oriented systems, such as secrecy, fair exchange, and secure communication link. However, even if security and privacy are strictly related concepts, the two above work are limited to security concerns only. For our approach we need UML extensions specifically addressing privacy concerns.

2.6. Summary of gaps and contributions

In the previous subsections we pointed out some limitations of existing solutions, emphasizing the need of enforcement elements. So, we propose PrivAPP with aims to fulfill specific research gaps, summarized below.

- *Conceptual models do not address privacy enforcement mechanisms*. In PrivApp we propose a conceptual model with enforcement elements, providing suggestions to document how the system will enforce privacy policies, (i.e., how it will accomplish the privacy promises).
- *Reference architectures are limited regarding privacy concerns*. There is a need of linking privacy policy and the software architecture, in a way that elements of the architecture that are responsible for enforcing the statements of the policy are identified. In the PrivAPP architecture we defined an orthogonal layer to address concepts directly related to the privacy domain, including privacy enforcement elements for different sources of privacy violation.

- *There are no UML profiles addressing privacy*. UML is one of the most popular ways to document software architectures, and the de-facto standard in both industry and academia. Still, there is a need of an UML profile to address privacy concerns. PrivAPP provides a privacy UML profile for documenting UML models from a privacy perspective, allowing the designer to identify privacy enforcement mechanisms directly in the software design, and link them to statements of the associated privacy policy.

Besides addressing the above individual research gaps, our solution is integrated, i.e., the elements of the approach were conceived in a comprehensive way, providing a reference to address privacy concepts from their definition/understanding (conceptual model), to their implementation in a software architecture (reference architecture), to the modeling in actual diagrams (UML profile). With this, models and documents are more consistent and compatible, allowing the designer to keep track of privacy concerns addressed in the privacy policy and of enforcement solutions described in the architecture.

It would be difficult to achieve this integration by combining existing previous solutions; moreover, many of these solutions are not focused on privacy protection in the context of web applications and services (e.g. [18] is more focused on security; [19] is more focused on business). On the other hand, the proposed PrivAPP approach does not prevent other techniques to be applied in conjunction, as privacy is a complex property that needs to be protected at different levels using complementary approaches.

3. The Proposed Approach: PrivAPP

The approach we propose systematizes the privacy concepts in the scope of web applications. It helps providing a better understanding of the privacy domain and, consequently, facilitates research, modeling and development of privacy-aware technology. Figure 1 outlines PrivAPP as a reference model and how it fits in the development process.

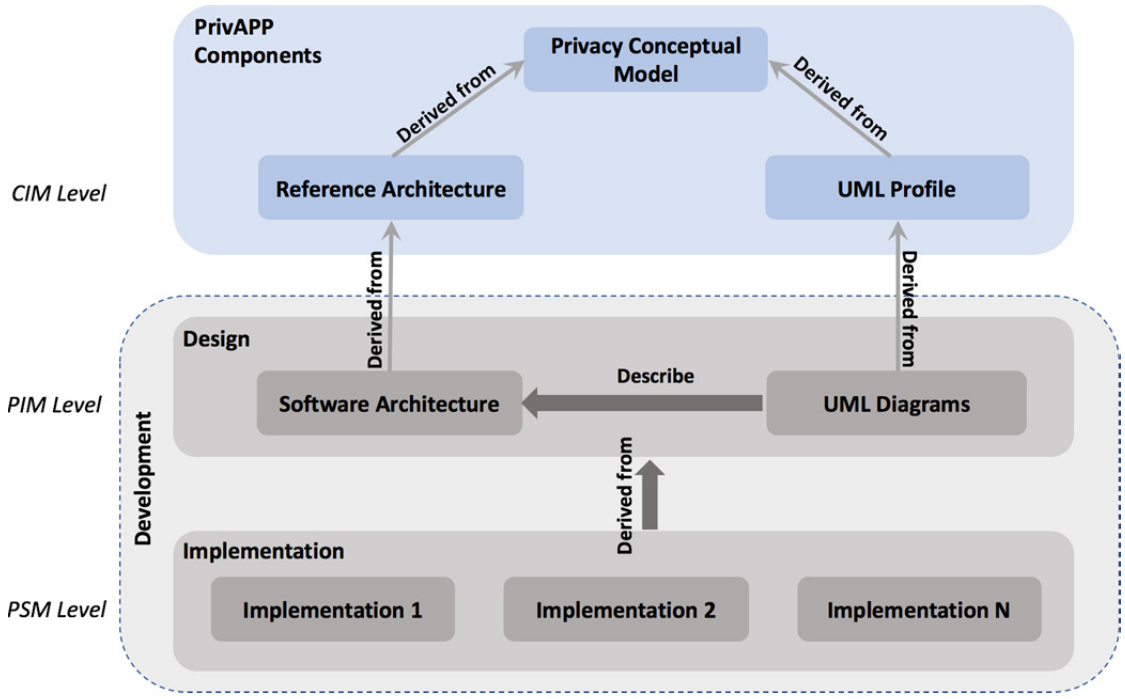

Figure 1. The proposed privacy approach and its application.

In Figure 1, the **Privacy Conceptual Model**, the **UML Profile** and the **Reference Architecture** are the three components of the proposed approach. They are described in details in the next subsections. Besides addressing the research gaps described in Section 2.6, the motivations for structuring the PrivAPP approach in these three constituents are the following:
(i) *The conceptual model is a key element in the design of UML profiles* [38] Also, it is the starting point for representing significant relationships in the privacy domain. It served as basis for the conception of our UML Profile and Reference Architecture, i.e., these both resources are *derived from* the conceptual model.
(ii) *Many companies have been adopting reference architectures and achieving positive results* [39]. Reference Architectures capture the accumulated architectural knowledge of several previous works. It reflects experiences captured in architecture principles and best practices. This condensed know-how provides guidance to later developments and key engineering decisions, preventing the reoccurrence of bad experiences.
(iii) The Profile must describe the reference architecture [40]. The reference architecture defines a logical architecture, in which general modeling concepts are identified and described in a UML Profile. This profile reflects a reference architecture that should be enforced when modeling privacy-aware applications and services. In PrivAPP, as the reference architecture and the UML profile are based on the same conceptual model, the integration is easier and more consistent.

The **development** process, represented in Figure 1, shows how the approach can support the design and implementation of privacy-aware applications. In the **design** phase, a concrete software architecture is specified, *derived from* the PrivAPP Reference Architecture. UML diagrams are used to *describe* this concrete software architecture.
Such diagrams include annotations for modeling privacy policy statements and enforcement resources, *derived from* the PrivAPP UML Profile.

These models will guide the **implementation** of the product, including solutions that help protecting privacy of personal information in the target application. So, the implementations of these solutions can be *derived from* the models created in the approach. Based on other requirements, technical constraints, or specific choices, different implementations can be derived from the same design.

Figure 1 also shows how the proposed PrivAPP approach relates to the Model Driven Architecture (MDA) from OMG. The Model Driven Architecture [33] is an approach for the development of software systems following a model-driven approach, consisting in a set of standards and guidelines. MDA foresees modeling at three different layers of abstraction, and progressive refinement with the aid of model transformation. The *Computation Independent Model (CIM)* is a business model or domain model; the Privacy Reference Architecture, the Conceptual Model, and the UML Profile all reside at this level. The *Platform Independent Model (PIM)* is a logical model of the software that abstracts from implementation concerns; the concrete software architecture resides at this level. In the architecture, privacy concerns are described by diagrams that use the UML Profile. The *Platform Specific Model (PSM)* is a concrete model of how the system is actually implemented; at this level we have the implementation(s) of the system for different platforms, which are derived (either manually or automatically) from artifacts at the PIM level.

It should be noted that the PrivAPP components (CIM level) are defined only once, and they are the main contribution of this paper. Conversely, the development and

implementation phases (PIM and PSM levels) will be executed at different times for different applications. The PrivAPP components are described in the following.

3.1. The Privacy Conceptual Model

The Privacy Conceptual Model is a model of the domain concepts that are required for modeling views of the system where privacy management and protection are applied. It is an extension of the metamodel introduced in our previous work [45], to specify more elements that can be used for policy enforcement. It is important to mention that the model considers users to inform their privacy preferences about their personal information, agreeing or not with the policies or part of them. This feature allows users to make more thoughtful choices about the use of their personal information online. The goal of the conceptual model regarding enforcement elements is to help the selection of enforcement mechanisms, according to statements of policies and users preferences.

The proposed elements were created based mainly on two main sources: privacy principles and reference models. Regarding the privacy principles we adopted, more specifically, for this conceptual model, the privacy principles described by the fair information practices developed by the Organization for Economic Cooperation and Development (OECD) [41] and the Global Privacy Standard [42].The guidelines created by the OECD were adopted because they have been used as the model for most of the privacy legislation throughout the world. The Global Privacy Standard was selected because it attempts to develop a single privacy instrument, i.e., a set of universal privacy principles. Regarding the reference models, we based mainly on the ISO/IEC 29100 [43], the ISO/IEC 29101 [44], the OASIS (Organization for the Advancement of Structured Information Standards) Privacy Management Reference Model [20].

The identified privacy elements and their relations are organized in a conceptual model, which is presented in Figure 2.

In Figure 2, the *Privacy Policy* element represents the artifact that must be defined and presented to the user. A *Privacy Policy* element can be defined by means of its attributes: *id* (identification of the policy), *name* (name of the policy) and *dateCreation* (the date that the policy was created).

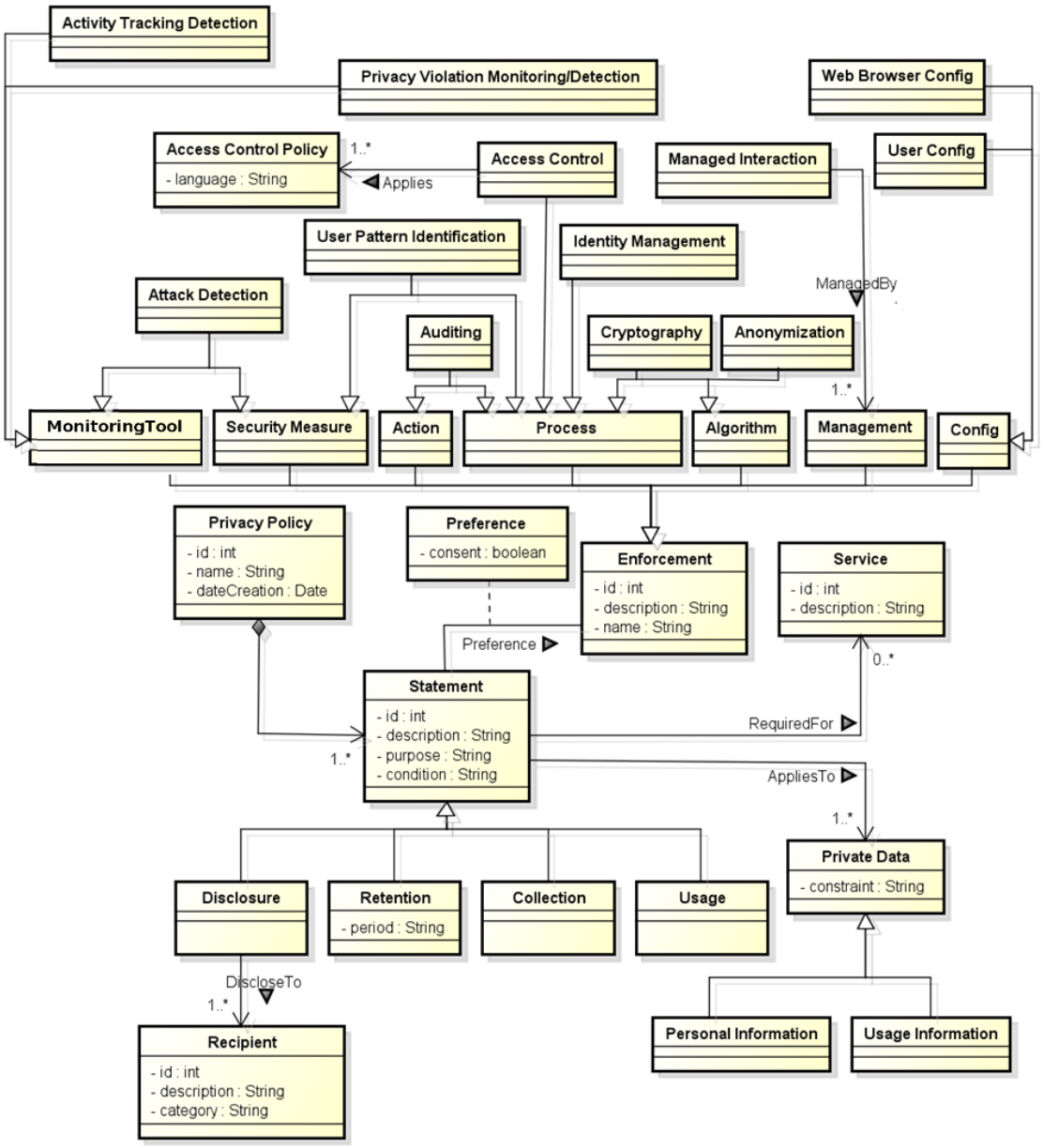

Figure 2. The approach's privacy conceptual model.

The *Privacy Policy* element is composed by one or more *Statements*. The *Statement* element represents the description of one of the rules that are specified in the privacy policy. The attributes that identify the statements are: *id* (identification of the statement); *description* (description of the rule), *purpose* (the purpose for which the data is collected or managed, e.g. research and development, or contacting visitors for marketing of services or products); *condition* (e.g.: "before collecting, using or disclosing personal information from a child, an operator must obtain verifiable parental consent from the child's parent").

In addition to the generic *Statement* element, there can be four main specialized types: *Disclosure* (specifies which data will be disclosed and to whom), *Retention* (specifies the period the data will be retained), *Collection* (specifies what information, i.e., what private data will be collected) and *Usage* (specifies how the private data will be used). Based on the statements, users inform their privacy preferences, i.e., they decide to agree or not to statements.

The *Disclosure* element is related to the *Recipient* one. *Recipient* represents who will access the data to be disclosed. Its attributes are *id* (identification of the recipient), *description* (a textual description), and *category* (used to classify the recipient according to a given taxonomy, e.g., internal or external groups, individual or organization, etc.).

The *Service* element represents the services offered by the company, which are related to statements, i.e., the services that a person can use if he/she provide his/her private information to the applications' company. The *Service's* attributes are *id* (identification of the service) and *description* (description of the actions and results provided by the service). There is an association between a *Statement st* and a *Service sv* if the utilization of *sv* is subordinated to the acceptance of *st* by the user.

A *Statement* has also a relation with the *Private Data* element, which represents the data to be collected and managed by the application. The *Private Data* can be of two types: *Personal Identifiable Information* (data that identify a person, e.g. name, address, phone number, e-mail, etc.) and *Usage Information* (data collected when the data subject use services, e.g. links accessed, current location, search strings, etc.). The association between a *Statement* and a *Private Data* element keeps track of which private data each statement applies to.

Besides the *Statement*, another key element of the privacy conceptual model is the *Enforcement* element. This element represents the resources that can be used in order to enforce the privacy policy statements, respecting the data subjects' preferences. The attributes of the *Enforcement* element are *id*, *name*, and *description*, which represent, respectively, the identification, the name and the description of the resource to be used. *Statements* can be associated with the resources that are adopted for its enforcement (*Enforcement* elements). The association is performed through the *Preference* relation; such relation has an attribute, *consent* (*true* or *false*, meaning that the user consents or not to the statement), which is used to specify in which case such enforcement resource need to be applied. Each statement may be associated to one or more *Enforcement* elements.

The *Enforcement* can be represented as *Monitoring Tool* (e.g. tracking activities tool, intrusion detection tool), *Security Measure* (e.g. security packages updates, use of antiviruses and firewalls), *Action* (e.g. allow access, deny access, anonymize data, remove from storage devices, logging actions, encrypt data), *Algorithm* (e.g. k-anonymity – for anonymizing data, RSA – for encrypting data), *Process* (e.g. identity management, access control, auditing), *Management* (management of privacy policies), *Config* (e.g. web browser security configurations, changes in default configurations). We specialized the *Enforcement* element with elements that we consider most relevant to the purpose. Obviously, the model is highly representative but not exhaustive; yet it is extensible enough to include more *Enforcement* elements as necessary.

Activity Tracking Detection is a tool that verifies if the system user has his/her activities tracked. *Privacy Violation Detection* is a tool that verifies if the user's privacy is violated sometime. *Attack Detection* is a tool which verifies if the system suffers an attack and, as it is related to the security of the system, it can also be considered a security measure.

The *User Pattern Identification* element is a process that analyzes stored users behaviors and uses them as a security resource against malicious users. Usually it consists of observing and collecting data over time periods and then applying analysis methodologies to identify different user patterns. Obviously, as it gathers PII, this process must be used only for security purposes.

Auditing refers to auditing resources that web application must use to monitor and identify possible privacy violation sources. These resources should monitor all the

system elements, as databases, servers, application, services calls, etc. This can be done automatically, though processes, or with the support of auditors actions.

Identity Management is a set of processes and technologies to manage, simplify and protect against unauthorized access. *Access Control* is also a process with a set of rules by which users are authenticated and by which the access to applications and other information services is granted or denied. The *Access Control Policy* represents the document that specifies roles and the information each role can access. The *language* attribute refers to the language the access control policy is defined (e.g., XML).

Cryptography element represents the process used to cypher information and avoid unauthorized access. It can be done through the use of algorithms as, for example, the RSA. The *Anonymization* element represents the process used to avoid disclosure of stored confidential information that is retrieved even by means of data analysis. The k-anonymity is a representative algorithm to support this process.

The *Management* element refers to the management of privacy policies in a ubiquitous environment, where data are transferred to different third-parties components or services. When this transfer happens, it is necessary to be sure that the policies are correspondent in order not to violate the main privacy policy (the privacy policy of the main application, i.e., the one whose statements the user agreed with). So, it is necessary to verify this correspondence and, if the policies are not correspondent, some actions must be taken (as, for example, adaptations in the policies). The *Managed Interaction* represents interfaces between different parts of the system using different privacy policies. As such interfaces may involve violation of privacy policies, they should be correctly managed. Thus, a managed interface has a relation with a *Management* element that is in charge of managing/protecting the communication, verifying the policies of entities interacting with the system.

Finally, the *Web Browser Config* represents configurations outside the system (i.e., each user must configure its own web browser) in order to protect their privacy, especially when they do not want to be tracked. *User Config* represents the configurations users can do in the own system or web page in order to refuse some services as, for example, advertisements or cookies and similar. Both these elements (*Web Browser Config* and *User Config*) were added to the conceptual model in result of the approach evaluation, described in Section 5.

3.2. The Privacy Reference Architecture

The Privacy Reference Architecture (PRA) is part of the proposed approach and it is based on the Privacy Conceptual Model, i.e., the elements of the conceptual model are distributed through the layers so that they can be implemented. It presents elements in a higher level of abstraction, describing features and functionalities that must be addressed during the development of web applications in order to protect the privacy of the users' information. From the PRA, it is possible to derive concrete architecture models that facilitate the development of privacy-aware technology.

The PRA was constructed using the ProSA-RA approach (Process based on Software Architecture - Reference Architecture [46], a systematic and iterative process for specification, design and evaluation of reference architectures). Besides the ProSA-RA, multiple sources of information have been considered as basis to its definition: (i) the conceptual model; (ii) software architectures with privacy concerns available in the literature; (iii) legislation, standards and norms for developing privacy-aware applications; (iv) solutions, frameworks and tools for privacy information protection; (v) privacy violation taxonomies. These sources were selected as they present, in a

broad context, current privacy problems and possible resources to protect information against those problems, which are two key issues to support the design of the privacy reference architecture.

The PRA is based on a three-layers architectural style: *Presentation*, *Application* and *Persistence*. Also, for privacy protection, we introduced a logical *Privacy* layer orthogonal to the *Presentation*, *Application* and the *Persistence* ones. It is represented in Figure 3.

In Figure 3, the **Presentation Layer** refers to the user interface. It allows the user to interact with the application. The *Web Browser* element, on the client side, refers to security configurations that must be set to protect the personal information from activity tracking. This tracking consists of identifying user activities on the Web without consent and may therefore represent privacy violation.

The **Application Layer** represents the application logic, with functionalities inherent to the organization's business model. For each application there are two elements: *Privacy Policy Statements* and *User Preferences.* The *Privacy Policy Statements* element refers to the fact that the web application must provide the privacy policy document to its customers and business partners, which is composed of a set of statements.

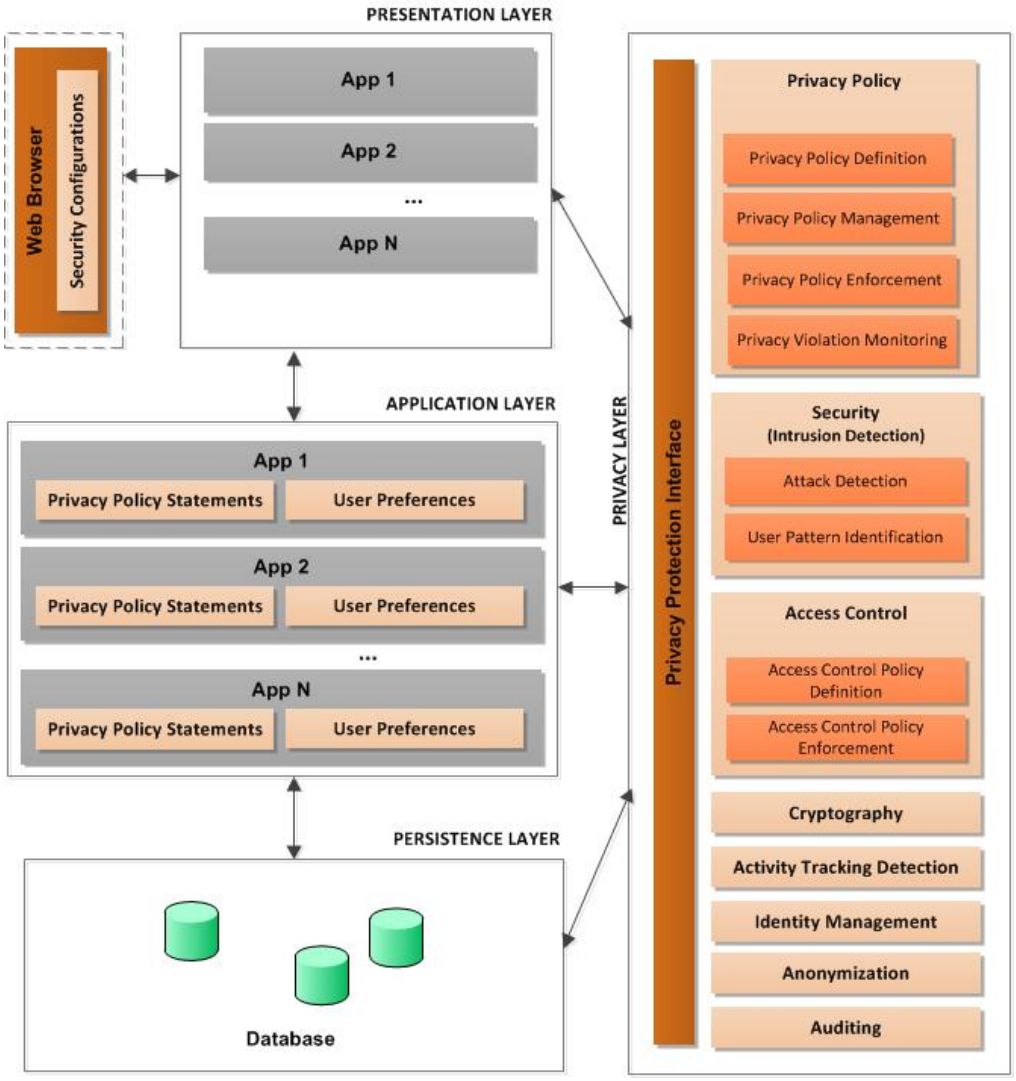

Figure 3. The Privacy Reference Architecture.

The *User Preferences* element refers to the need for the web application to permit users to state their privacy preferences regarding personal information, agreeing or not with the presented policies (or part of them).

The **Persistence Layer** is responsible for the storage of information. The *Database* element represents the storage resources and the functionalities the web application may use, such as the DBMS (Database Management System) and other technologies that support the data management and recovery. Many different applications can access the same database. This data sharing requires ways of ensuring that private information for one application is not accessed by other unrelated applications.

Still in Figure 3, the **Privacy Layer** includes most of the concepts directly related to the privacy domain. This layer is an orthogonal concept and can be accessed by the other three layers (presentation, application, persistence) directly, through well-defined interfaces. The *Privacy Protection Interface* represents the interfaces that allow access to the services or components for privacy protection of personal information. These services and components are cross-cutting concerns, representing functionalities that are independent of the application and may be encapsulated as transversal elements or aspects. So, privacy services or components can be implemented separately and used as aspects or even libraries to be integrated in the other layers. The decision of which layer will implement which privacy service/component must be taken in the modeling phase.

The *Privacy Layer* has a set of eight elements: (i) *Privacy Policy*, (ii) *Security/Intrusion Detection,* (iii) *Activity Tracking Detection*, (iv) *Access Control*, (v) *Cryptography*, (vi) *Identity Management*, (vii) *Auditing* and *(viii) Anonymization*. A short description of each is provided next.

(i) **Privacy Policy**. This element is necessary to represent the privacy policy document, which establishes how private data must be handled. It is responsible for defining, enforcing and managing privacy policies. The *Privacy Policy Definition* element is responsible for privacy policies to be defined and presented to the user. Also, based on the policies, users should be able to state their privacy preferences.

Besides defining privacy policies, web applications must ensure that such policies are enforced, i.e., that the agreement signed in the privacy policy is fulfilled. Such important requirement is assured by the *Privacy Policy Enforcement* element.

The *Privacy Policy Management* element represents the management of privacy policies between third-parties (i.e., independent web applications or services that interact with the main application). This is an important element because different applications and services can have different privacy policies and the way information is exchanged between them must comply with these policies. Also, this element is responsible for updates in the privacy policies, which should be managed. The updates in the privacy policy must be notified to the users and new preferences about these updates must be recorded and enforced.

The *Privacy Violation Monitoring* element refers to solutions that can be used to detect privacy violation. It is included in the reference architecture because having solutions that continuously monitor access to personal data and detect misuse or abnormal behavior are important in the process of protecting privacy.

(ii) **Security/Intrusion Detection**. Security and Privacy in web applications are closely related, since security breaches can result, in some cases, in misappropriation and misuse of information by malicious users. The detection of attacks is extremely important as it allows actions to be taken to avoid the privacy violation. This feature is represented by the *Attack Detection* element.

Another way for malicious users to access the application is by using valid credentials, usually obtained through identity theft. To help avoiding these malicious users, behavioral tendency analysis can be used, as far as the web application collects the users' behaviors. So, the application must analyze these stored behaviors and use them as a security resource. Malicious users can potentially show a behavior that is different from the one of the legitimate users and when such difference is detected, the application may ask for some new identification, to confirm the user identity and, thus, reinforce the security. This resource is represented by the *User Pattern Identification* element.

(iii) **Activity Tracking Detection**. Activity tracking consists of identifying user activities on the Web without consent and building a profile, which potentially represents a privacy violation. Similarly to *Privacy Violation Monitoring*, this element was included in the reference architecture because it is important that the web application uses resources for detecting improperly tracking, allowing actions to be taken and, consequently, protecting against such violation.

(iv) **Access Control**. Access Control is a set of rules by which users are authenticated and by which the access to applications and other resources is granted or denied. It is very common nowadays and an important resource to protect private information, so, this element is necessary to the reference architecture.

The web application must allow access control policies to be defined, specifying the roles and the information each role can access. The definition of this policy is represented in the *Access Control Policy Definition* element.

Besides defining access control policies, the application must enforce them, assuring that only authorized users will access particular private information. The *Access Control Policy Enforcement* element refers to these enforcement resources.

(v) **Cryptography**. This element is necessary to protect the personal private information during the transmission through the web. Cryptography provides confidentiality (only the authorized receiver can read the message), integrity (the receiver will be able to identify whether the message has been changed along the way), authentication (the receiver can identify if the sender is the same person who should have sent), and non-repudiation (it should not be possible for the sender to deny having sent the message).

(vi) **Identity Management**. If the web application uses some digital representation of the known information about a specific individual or organization, it must use a digital identity management resource. This resource consists of a set of processes, tools, social contracts and supporting infrastructure to create, maintain, and terminate a digital identity. It enables secure access to an expanding set of systems and applications.

(vii) **Auditing**. Auditing is used to evaluate internal controls in an automated information system and to verify the results of phases and processing systems. This element is needed for auditing resources that the web application must use to monitor and identify possible privacy violation sources. These resources should monitor all the system elements, such as databases, servers, application, etc.

(viii) **Anonymization**. This element is necessary in the reference architecture because it specifies that the web application must provide techniques to avoid disclosure of confidential information that is retrieved even by means of data analysis. This must be done especially when dealing with statistical databases, which are used mainly to produce statistics on various populations and may contain confidential data regarding individuals.

It is important to mention that the architecture is generic and should be instantiated considering the specific properties of the target web application (if necessary). Our goal is to provide a general view about the elements that the application can adopt to avoid the violation of the privacy of personal information.

3.3. The Privacy UML Profile

The Privacy UML Profile was also constructed based on the Privacy Conceptual Model, i.e., it is used to document the elements of the conceptual model in order to reduce ambiguities in the solution. The profile described in this paper extends our previous work in [45]. It defines new modeling elements, the enforcement ones, bringing specific concepts related to privacy protection to the UML language. Our Privacy UML Profile can be used for modeling views of the system that include privacy protection concepts. Its main purpose is to structure concepts of privacy definition and enforcement, and support the documentation of privacy specifications of web applications. Models created using the profile are meant to be used both during the development phase of a web application, as well as after its deployment. During the development, models created using the profile help developers to keep track of privacy requirements and how they are implemented. After the deployment, the same model can provide the users with a structured description of how the application will handle its private information.

As we recall from Section 2.5, UML profiles are defined using *stereotypes*, *attributes*, and *constraints*. The elements of our extended Privacy UML Profile are listed in Table 1, in alphabetical order.

Table 1. The Privacy UML Profile.

Stereotype	Base Metaclass or Stereotype	Attributes
<<AccessControl>>	Process	
<<AccessControlPolicy>>	Artifact	
<<Action>>	*Enforcement*	
<<ActivityTrackingDetection>>	MonitoringTool	
<<Algorithm>>	*Enforcement*	
<<Anonymization>>	Algorithm, Process	
<<AttackDetection>>	SecurityMeasue, Tool	
<<Auditing>>	Action, Process	
<<Collection>>	*Statement*	
<<Config>>	*Enforcement*	
<<Cryptography>>	Algorithm, Process	
<<Disclosure>>	*Statement*	
<<Enforcement>> *(abstract)*	Class	id (int) name (string) description (string)
<<IdentityManagement>>	Process	
<<ManagedInteraction>>	Port	
<<Management>>	*Enforcement*	
<<PersonalInformation>>	*PrivateData*	
<<Preference>>	Association	consent (Boolean)
<<PrivacyPolicy>>	Artifact	id (int), name (string), dateCreation (date), constraint (string)
<<PrivacyViolationDetection>>	MonitoringTool	
<<PrivateData>> *(abstract)*	Property, Class	

<<Process>>	*Enforcement*	
<<Recipient>>	Actor	id (int), description (string) category (string)
<<Retention>>	*Statement*	period (string)
<<SecutiyMeasure>>	*Enforcement*	
<<Service>>	Component	id (int) description (string)
<<Statement>>	Class	id (int), description (string), purpose (string), condition (string)
<<MonitoringTool>>	*Enforcement*	
<<Usage>>	*Statement*	
<<UsageInformation>>	*PrivateData*	
<<UserPatternIdentification>>	SecurityMeasure, Process	

In Table 1, the conceptual elements from the privacy conceptual model (see Figure 2) are mapped to UML *stereotypes*, and listed in the first column; for completeness, also abstract stereotypes are included in the table. The base element of each stereotype (UML *metaclass* or another stereotype) is listed in the second column. It should be noted that a stereotype may also extend another newly introduced stereotype. Finally, stereotype *attributes* are listed on the last column.

The <<PrivacyPolicy>> stereotype extends the *Artifact* metaclass, which represents the specification of a physical piece of information that is used or produced by a software development process, or by deployment and operation of a system [11]. Also, it extends the *Class* metaclass. In UML, a *Class* describes a set of objects that share the same specifications of features, constraints, and semantics [11].

The <<Statement>> stereotype extends the *Class* metaclass. In UML profiling, *Class* is often selected as a "default" base metaclass, and it is typically adopted for stereotypes that do not represent software elements as well. The <<Statement>> stereotype is further extended by stereotypes that characterize the nature of the statement of the privacy policy: <<Disclosure>>, <<Retention>>, <<Collection>> and <<Usage>>.

The <<PrivateData>> abstract stereotype extends both the *Property* and the *Class* metaclasses. The *Property* metaclass is a structural feature which represents an attribute [11], i.e., a portion of data; the *Class* in this context is seen as an aggregation of multiple elements of information. <<PersonalInformation>> and <<UsageInformation>> are used to mark data that is regarded to as personal information or usage information, respectively, and they extend <<PrivateData>>.

The <<Enforcement>> stereotype and its descendants represent resources and solutions that are used to enforce the statements described in the privacy policy. Ideally, the profile should allow the modeler to relate enforcement solutions directly to elements in the model of the software architecture. Depending on the context, an enforcement solution (e.g., an algorithm) may be described by either a structural (e.g., a *Component*) or a behavioral feature (e.g., an *Activity*). In order to be able to cover both cases, our <<Enforcement>> stereotype extends the *Class* metaclass, which is a common ancestor of both the *Component* and *Behavior* UML metaclasses [11]. The <<Enforcement>> stereotype is then extended to better categorize the nature of the enforcement solution.

The <<Preference>> stereotype extends the *Association* metaclass, which specifies a semantic relationship that can occur between typed instances, in our case elements of the <<Statement>> and <<Enforcement>> elements. Such association

relates an enforcement solution with a statement for which it is needed, also detailing for which kind of user preference (*opt-in, opt-out*) is actually needed.

The <<Service>> stereotype extends the *Component* and *Port* metaclasses. A *Component* describes a modular part of a system that encapsulates its contents, i.e., without focusing on its internal implementation, but only on the service(s) it provides. The <<ManagedInteraction>> stereotype extends the *Port* metaclass; a *Port* may be used to specify in more details the services a classifier provides (requires) to (from) its environment. When a port is stereotyped with the <<ManagedInteraction>> stereotype it is identified as port requiring special care in handling the information flow from/to other services. A <<Management>> element should take care of such interaction.

Finally, the <<Recipient>> stereotype extends the *Actor* metaclass. This metaclass specifies a role played by a user or any other system that interacts with the subject.

The *constraints* needed to express our domain concepts are limited to relationship multiplicities (see Figure 2); no additional constraints are included in the profile.

4. Application example

An application example was developed to include data privacy protection in a web application. The goal is to apply the proposed approach in practice, although on a small example, to have an indication of its practical feasibility and contribution. To do this we followed some steps: (i) we selected an application without privacy protection resources; (ii) we established a privacy policy for this application; (iii) we created, based on the main statements of the application's privacy policy, the UML diagrams that describe the policy and show how the application must enforce its statements; (iv) we created a software architecture including the privacy protection elements in the original web application; (v) we implemented the solution designed by the diagrams and architecture.

The requirements we consider are based on the privacy policy, the web application architecture, and functional requirements; they are expressed through the application of PrivAPP. The process led us to the implementation of an access control mechanism which allows users to express their privacy preferences and, according to these preferences, the requested information are permitted or denied. Moreover, this mechanism is integrated in the relational database system, contributing to security against possible attacks to the web application or the network.

4.1. The bookstore application

The web application we used in the application example is a Java implementation of TPC-W [47]. TPC-W is a benchmark for web-based transactional systems where several clients access the website to browse, search, and process orders. The typical workload that it supports consists of shopping sessions. Each session emulates the behavior of a customer connected to the server and generally consists of a sequence of interactions: search, browse, add to shopping cart, make purchases, and so on. In this study, we adapted the TPC-W through an implementation of a retail online book store, which simulates the sale of books through the Internet. By purpose, the application is devoid of any data privacy protection. So, for sake of security and privacy, we did not use real user's data. The diagram in Figure 4 shows a high-level view of the TPC-W architecture, based on Garcia and Garcia [48].

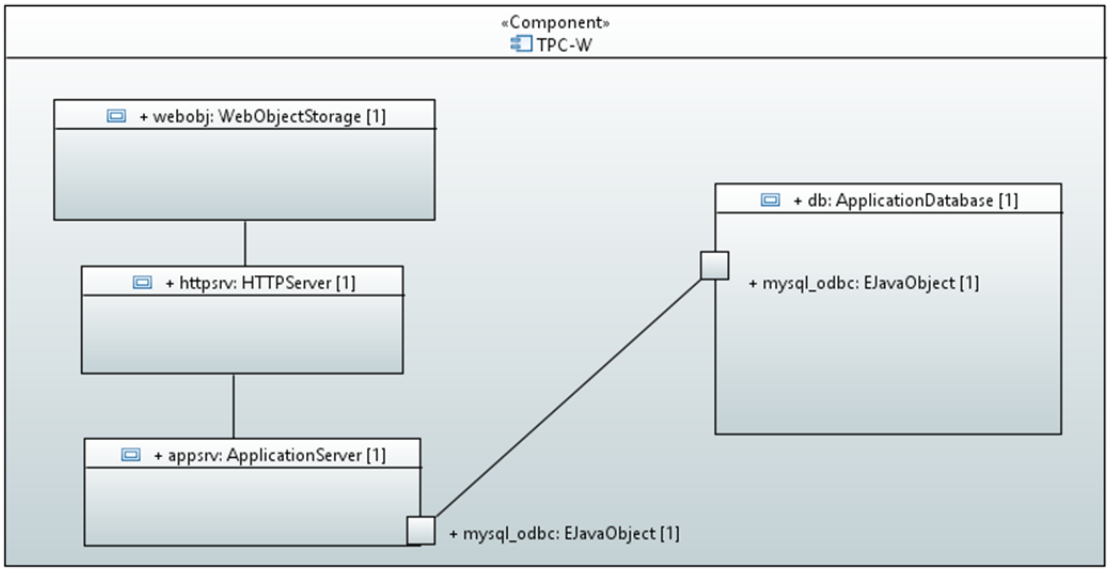

Figure 4. TPC-W's architecture diagram.

Like all e-commerce benchmarks, TPC-W has a client-server architecture. The client computers function as remote browser emulators to simulate the workload real customers would generate. In Figure 4, the system includes an HTTP server with Web object storage, an application server, and an application database. This system communicates with the clients through a dedicated network.

The TPC-W component of our major interest is the Application Server. It is in this server that the bookstore implementation runs. Figure 5 details this Server, showing its components.

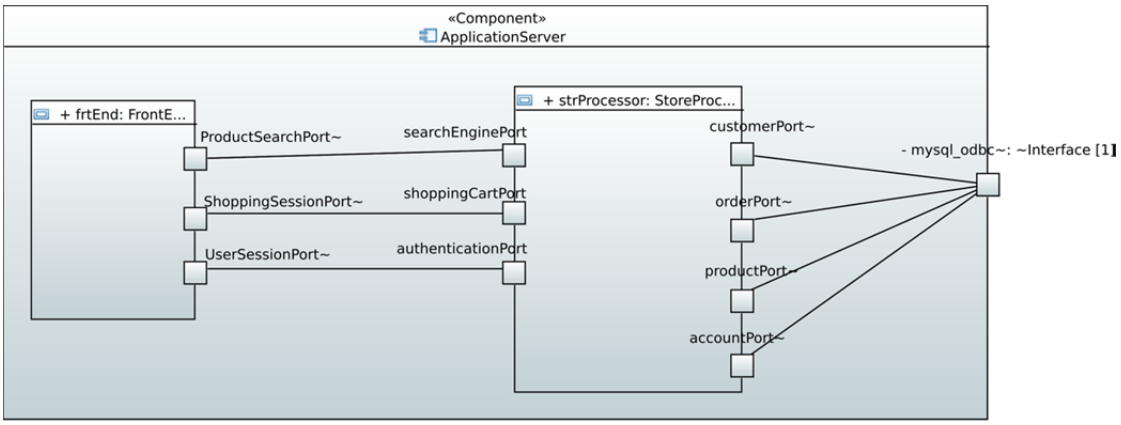

Figure 5. TPC-W's Application Server detailment.

In Figure 5, the *frtEnd* component is the one to represent the Front End of our implementation. It represents the presentation layer, with the interface between the user and the application. The *strProcessor* component is the Store Processor of the implementation, i.e., the procedures necessary to purchase the books online. They exchange information through their interface (ports), where the symbol "~" means that the interface is required, or provided otherwise. Next we detail each of these components.

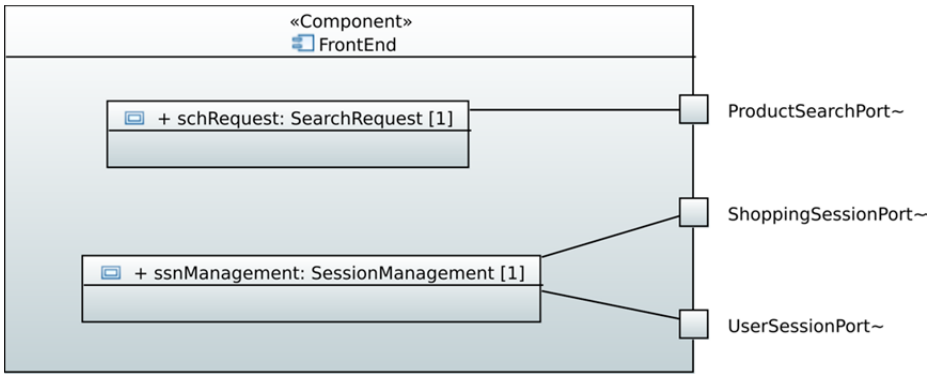

Figure 6. *Front End* component.

In Figure 6, the *FrontEnd* component is composed by the *schRequest* and the *ssnManagement* components. The *schRequest* is responsible for the interface where customers and visitors can search books on the bookstore. The *ProductSearchPort* exchange search queries with the component responsible for processing them. The *ssnManagement* is the interface where sessions can be created in two ways: (i) the visitor adds books to shopping cart without registering (shopping session); (ii) a registered user authenticates in the application to use it (user session). The *ShoppingSessionPort* and *UserSessionPort* are, respectively, the ports on which information related to the sessions is exchanged.

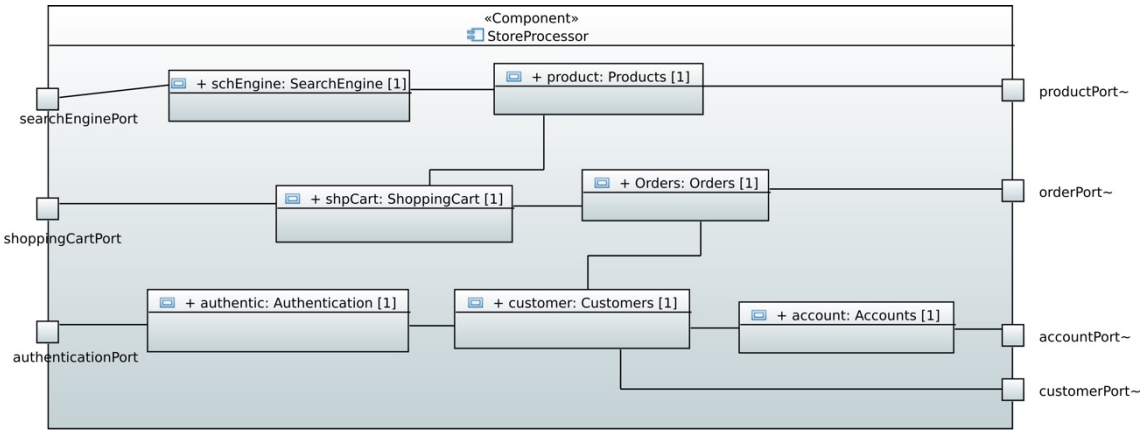

Figure 7. *Store Processor* component.

Figure 7 shows the *StoreProcessor* and its seven components. The *schEngine* is responsible for processing the search strings the user requested. It communicates with the *Product* component, which manages the inclusion, exclusion and updates of books. The *ShoppingCart* component is responsible for management of items to be bought, while the *Orders* processes the orders and the payments. To make a purchase, the visitor must register first. So, the *Customer* component communicates with to *Orders* and is responsible for managing the customers' records. Also, a customer's account is managed by the *Account* component. Once the visitor is registered and an account associated, an authentication process is necessary. This is done through the *Authentication* component.

The ports *productPort, orderPort, accountPort* and *customerPort* are those from where the respective components interact with the database, i.e., they use the interface provided by the database.

To implement privacy protection in this application it is first necessary to define a privacy policy, since it is the artifact that guides all privacy control process. As the focus in this paper is not on policy definition, we adopted the Amazon's privacy policy [49]. This adoption was done because Amazon is a very popular online book store; its privacy policy is therefore representative of this segment, or at least is affecting a considerable portion of users of this segment. Obviously, we cannot use the whole policy because our application is simpler than Amazon one. Also, representing all the statements would be unfeasible for this work. Thus, we selected 5 statements that are closely related to the functioning of our application; such statements are described in Table 2.

It is important to mention that, for guaranteeing privacy protection, in this work we interpreted fuzzy statements in their worst-case meaning, e.g., if a statement says "*we usually keep the copy*" we interpreted it as "*we do keep the copy*".

Table 2. Selected statements from the privacy policy for enforcement of privacy protection [49].

Statement	Description
ST1	"*We work to protect the security of your information during transmission by using Secure Sockets Layer (SSL) software, which encrypts information you input*."
ST2	"*You can add or update certain information on pages such as those referenced in the "Which Information Can I Access?" section. When you update information, we usually keep a copy of the prior version for our records."*
ST3	"*Cookies are unique identifiers that we transfer to your device to enable our systems to recognize your device and to provide features such as 1-Click purchasing, Recommended for You, personalized advertisements on other Web sites (e.g., Amazon Associates with content served by Amazon.com and Web sites using Checkout by Amazon payment service), and storage of items in your Shopping Cart between visits"*
ST4	"*Affiliated Businesses We Do Not Control: We work closely with affiliated businesses. In some cases, such as Marketplace sellers, these businesses operate stores at Amazon.com or sell offerings to you at Amazon.com. In other cases, we operate stores, provide services, or sell product lines jointly with these businesses. Click here for some examples of co-branded and joint offerings. You can tell when a third party is involved in your transactions, and we share customer information related to those transactions with that third party*."
ST5	"*Third-Party Service Providers: We employ other companies and individuals to perform functions on our behalf. Examples include fulfilling orders, delivering packages, sending postal mail and e-mail, removing repetitive information from customer lists, analyzing data, providing marketing assistance, providing search results and links (including paid listings and links), processing credit card payments, and providing customer service. They have access to personal information needed to perform their functions, but may not use it for other purposes*."

4.2. Applying the PrivAPP

As we have already said, the online book store is, by purpose, devoid of any data privacy protection. Our goal is to include privacy protection in this application. To do this, we modeled privacy concerns and the elements needed to enforce privacy.

We first created in the architecture a logical group of measures that helps in privacy protection. This logical group is defined as <<aspect>> because Aspect Oriented technology is rooted back to the separation of concerns by which different concerns of the software system can be designed and reasoned about in isolation from each other [50]. These aspects can be used in (i.e., crosscut) different components of the application, so, they can be used in both *ApplicationServer* and *DatabaseServer* of the original bookstore application. This logical grouping is represented by the *PrivacyManagement* component, in Figure 8.

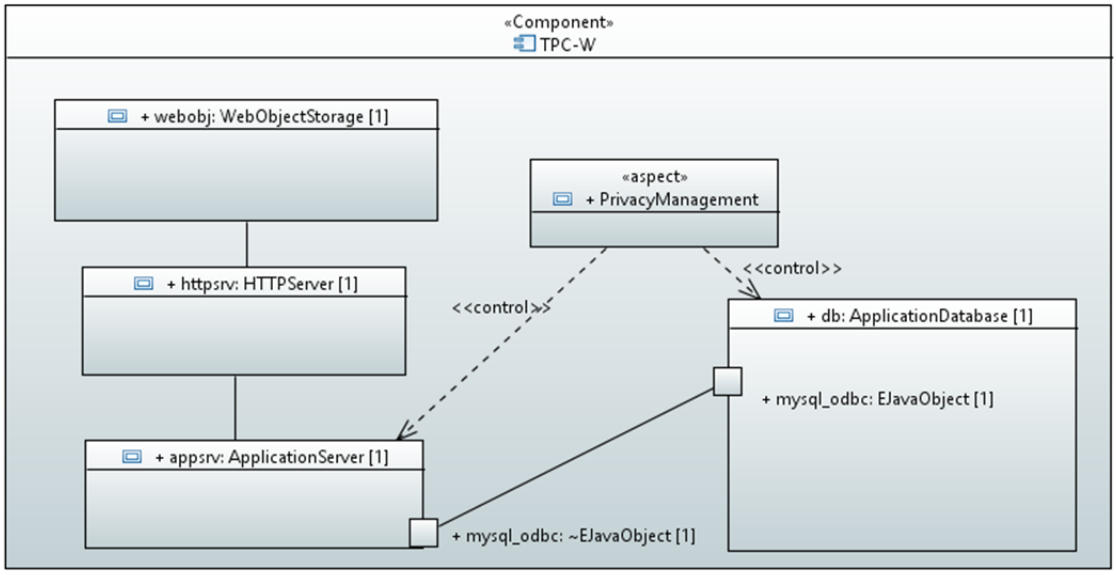

Figure 8. Including the *PrivacyManagement* component, responsible for privacy protection, in the original web application architecture.

After defining the logical group, we started defining the UML models because they help to better understand the privacy policy statements and the resources that can be used to enforce these statements.

At this point it is important to mention that the Privacy UML Profile and the Privacy Reference Architecture are elements from the proposed approach that can be used individually or in parallel, in a complementary manner (see Figure 1). In this application example, we used them in a complementary manner. So, while constructing the UML diagrams, we used the Privacy Reference Architecture to identify the enforcement elements that are more adequate to each statement and keep track of them in the UML diagrams.

For the sake of organization, we split the UML diagrams in two parts, represented in Figures 9 and 10. The privacy policy statements are the ones described in Table 2.

In Figure 9, the Statements *ST1*, *ST2* and *ST3* are shown, modeled respectively as <<*Statement*>>, <<*Retentiton*>> and <<*Collection*>> elements. *ST1* and *ST3* are related to *PrivateData*, which is identified both as <<*PersonalInformation*>> and <<*UsageInformation*>>. The statement *ST3* is related with 3 types of services provided

by the application: *OneClickPurchase, PersonalizedAdvertisement* and *StorageOfItems*. The ST2 applies to *OutdatedData*, which represents <<*PersonalInformation*>>. Furthermore, each statement is related to a set of enforcement elements by preference relations (<<*Preference*>>, *consent=true* or *consent=false*), based on whether that enforcement measure is required in case of consent of disagreement of the user. *ST1* is related to the *SSL*, which is a <<*Cryptography*>> element; *ST2* is related to *RemoveData*, which is an <<*Action*>> element; *ST3* is related to *DisableCookies*, which is a <<*WebBrowserConfig*>> element.

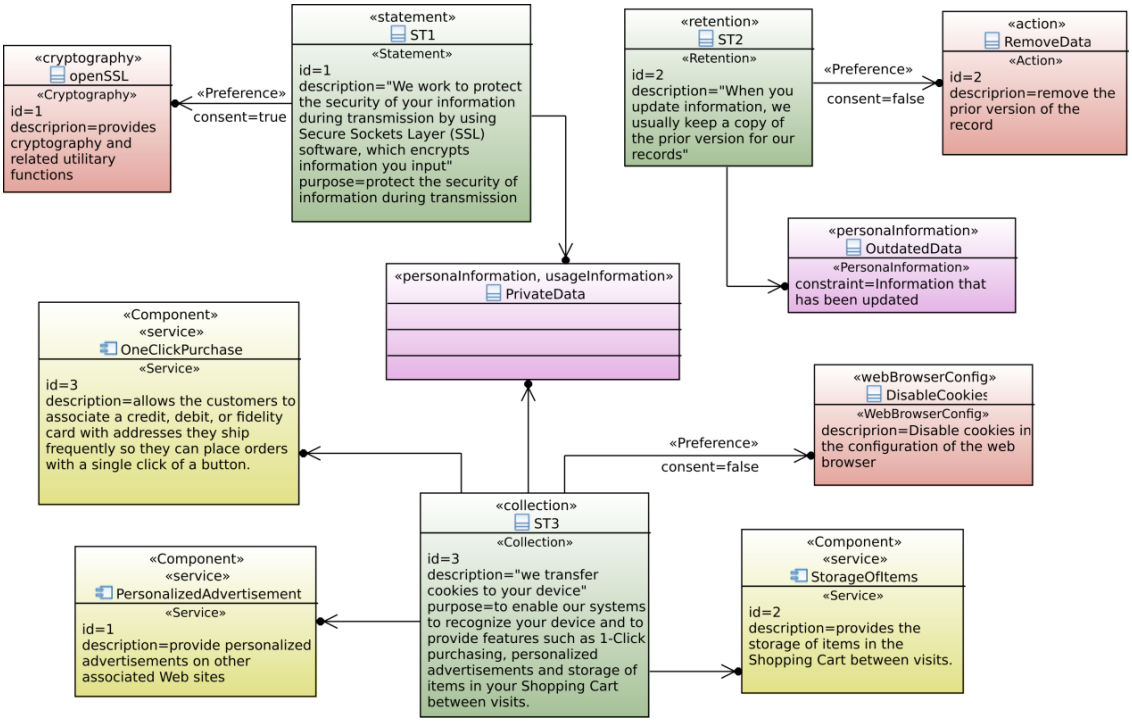

Figure 9. Statements ST1, ST2 and ST3 representation, with their related elements.

The same happens in Figure 10: statements *ST4* and *ST5* are modeled as <<*Disclosure*>> elements. They are related, respectively, with the *AffiliateBusinessOperations* and *BasicFunctions* services (<<*Service*>>) and the *AffiliatedBusinesses* and *ThirdPartyServiceProviders* recipients (<<*Recipient*>>), as well as the *PrivateData* (represented by <<*PersonalInformation, UsageInformaiton*>>). Both statements are related to their own user preference (<<*Preference*>>, *consent=false*) and the enforcement is given by the *AccessControlMechanism,* an <<*AccessControl*>> element, related to its <<*AcccessControlPolicy*>>, which we called *ACPolicy1*. Also for the Statement ST5, if the user agrees with the policy (<<*Preference*>>, *consent=true),* three enforcement elements need to be applied: <<*Management*>>, <<Anonymization>> and <<*Auditing*>>. The <<*Management*>> element, which we called *CheckThirdPartiesPolicies,* is responsible for checking the compatibility of the original application's privacy policy and the third party service provider's privacy policy. If they are correspondent, the services (*BasicFunctions*) can be provided. The <<Anonymization>> element (*AnonymizationMechanism*) is responsible for anonymizing private data before disclosing them to data analysis. The <<*Auditing*>> element (*AuditThridPartiesPurposes*) is responsible for periodically verify if the third parties are using the personal information shared with them according to the specified purposes.

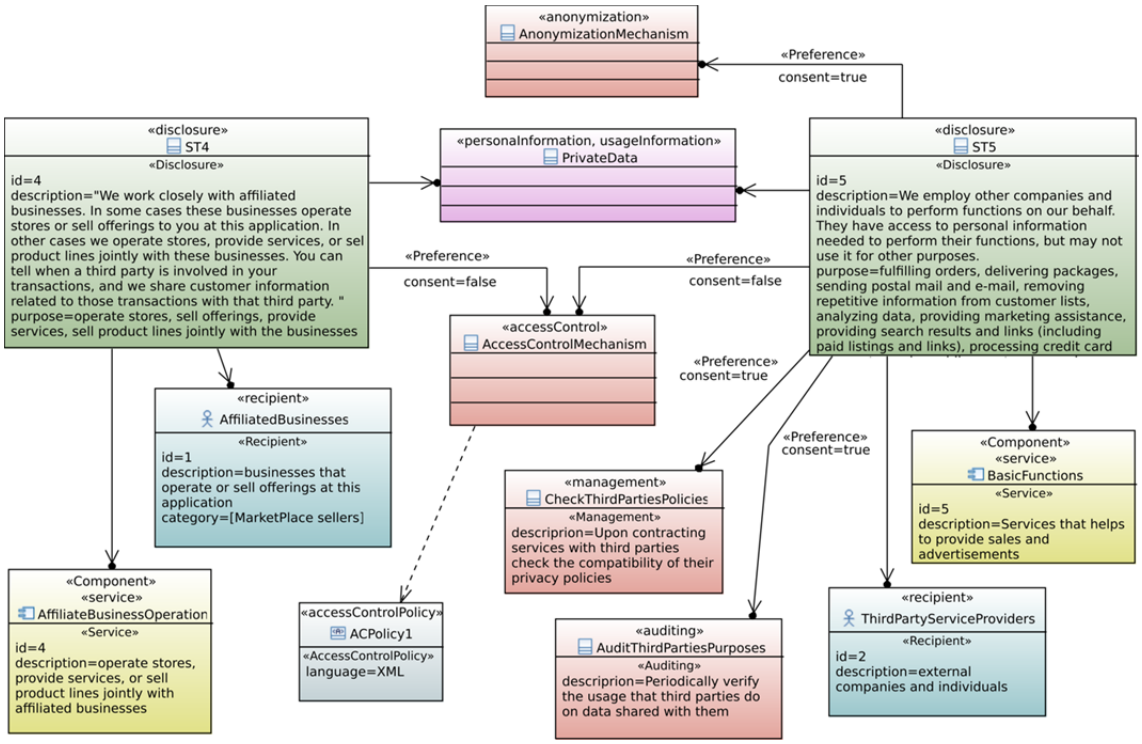

Figure 10. Statements ST4 and ST5 representation, with their related elements.

According to the Privacy UML Profile, a complete model should also include a <<*PrivacyPolicy*>> element, having a containment relation with all the statement elements included in the model. For this application example, <<*PrivacyPolicy*>> contains the 5 statements shown in the figures 9 and 10. A diagram representing all the statements aggregated to the <<*PrivacyPolicy*>> is presented in Figure 11.

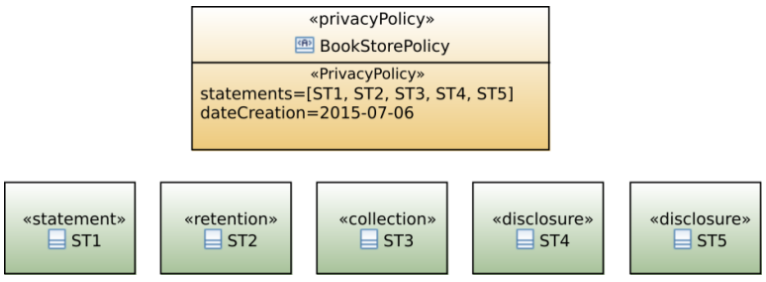

Figure 11. Privacy Policy element and respective Statements.

After defining the UML diagrams for the policy, we created a software architecture that represents the application with privacy protection. So, we identified the components corresponding to enforcement elements adopted in the UML diagrams in the reference architecture. They are: User Preferences, Web Browser (Security Configurations), *Privacy Policy Enforcement, Cryptography, Access Control Policy Definition, Anonymization* and *Access Control Policy Enforcement* (see Figure 3). The enforcement elements can be grouped into the *PrivacyManagement* component (see Figure 8) as subcomponents to be used in the software architecture. Figure 12 shows this group.

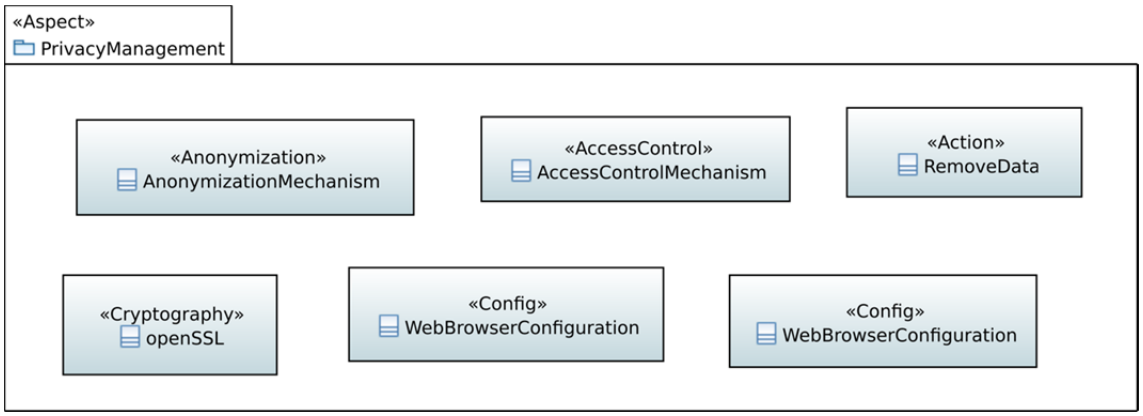

Figure 12. Enforcement components.

From the selected enforcement elements, in this application example we detail the implementation of the <<accessControl>> and the <<Anonymization>>, with the *AccessControlMechanism* and Anonymization Mechanism components respectively, because they are more interesting, since (i) <<cryptography>>, with *OpenSSL*, is off-the-shelf; (ii) <<config>>, with *WebBrowserConfiguration*, does not belong to the application, but rather to the user environment; (iii) <<action>>, with *RemoveData* is a simple implementation.

Basically, an access control mechanism has some access control policies and a mechanism that, based on these policies, gets the requested information and allows or denies these information to the requester. We modified the original TPC-W architecture to include the access control mechanism, in order to help protecting privacy according to the privacy policy (statements *ST4* and *ST5*).

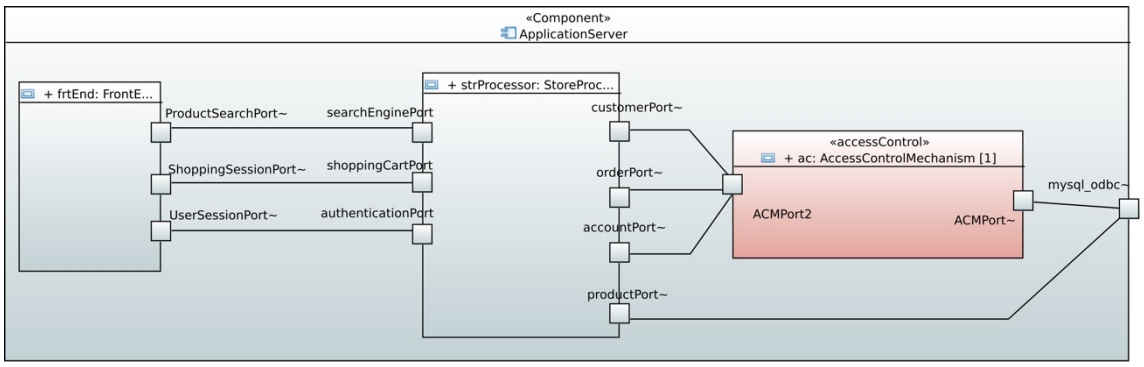

Figure 13. TPC-W's Application Server with the addition of the Access Control Mechanism

As illustrated in Figure 13, the *AccessControlMechanism* component was added to the original TPC-W's *ApplicationServer*. The *ApplicationServer* was presented previously in Figure 5 and now, in Figure 13, the *orderPort, customerPort* and *accountPort* ports are connected to the *ACMPort2*, which is the provided interface of the access control component. The idea is to control the access from third parties to information that includes orders, customers and account data. The *productPort* port is not connected to the access control because the products to be sold in the bookstore have free access to customers and visitors, i.e., the access control is not necessary for this information.

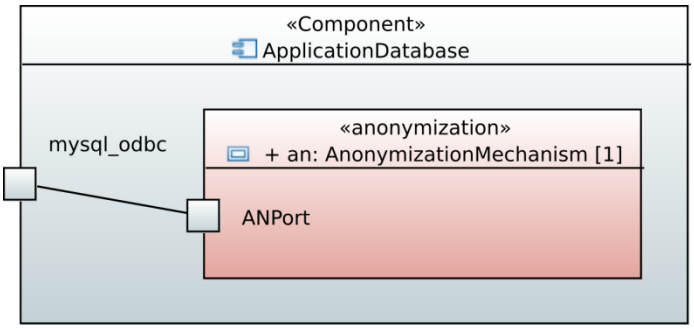

Figure 14. TPC-W's Database Server with the addition of the Anonymization Mechanism

In Figure 14 we show the the *AnonymizationMechanism* component, which was added to the original TPC-W's *ApplicationDatabase.* In this case, the *ANPort* port is connected to the provided interface of the respective TPC-W's component. As anonymization is usually performed in the ETL (Extract, Transform, Load) process, we decided to implement the anonymization in the database server because it anonymizes the data before disclosing it to the application.

During these decisions, our Privacy UML Profile and its stereotypes allow designers and developers to keep track of where the <<AccessControl>> and <<Anonymization>> enforcement is implemented within the software architecture and, in turn, where the privacy policy statements requiring these elements (ST4 and ST5 in our example) are being enforced.

One difference of the access control mechanism that we represented in the software architecture is the users' privacy preferences management. The mechanism must allow users to express their privacy preferences, related to each piece of their personal information, and this must be respected, i.e., the access to private information must be controlled according to these preferences. Thus, still detailing the software architecture, we detailed the components of the access control. This is reported in Figure 15.

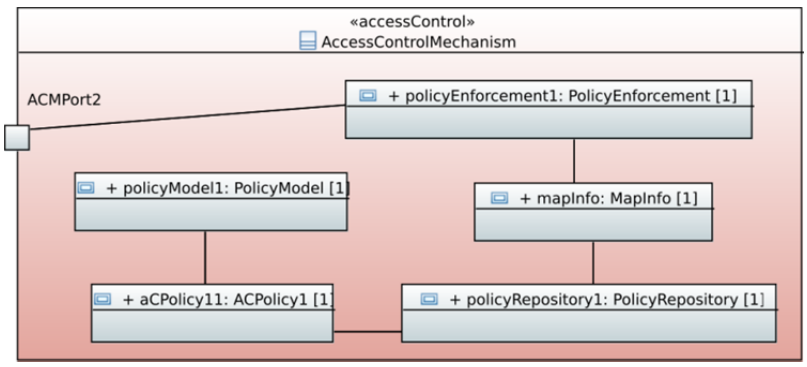

Figure 15. Access control mechanism detailing.

Briefly, In Figure 15, the components of the access control mechanisms are: (i) *policyModel1*: represent the model or set of models to be used in order to create the access control policies; (ii) *aCPolicy11*: represent the access control policies. This component is responsible for helping creating these policies; (iii) *PolicyRepository1*: responsible for maintaining the access control policies; (iv) *mapInfo0*: as the access control mechanism we represent is to be implemented in the database application, the policies information must be managed so that their information can be manipulated by it. The users' preferences must be considered and also managed by this component; (v)

policyEnforcement1: this component analyzes the access control policy information and the users' preferences and, according to them, it allows or denies the access to the requested information.

Based on the privacy UML Profile diagrams and the privacy software architecture, a database framework for access control was implemented. This framework allows users to express their preferences in a more complete way, where the privacy preference of each piece of their personal information can be defined. It also provides a mechanism to enforce the access control policies, guaranteeing that the user's privacy preferences will be fulfilled and, thus, contributing to privacy protection. This mechanism is integrated in the relational database system, which helps improving private information security. An experimental evaluation was conducted, in terms of performance of the implemented solution. The results showed that, although some performance impact was identified, it can be considered acceptable front of the importance of protecting privacy information, especially considering users privacy preferences in detail and giving these users more flexibility while dealing with their personal information online. The details of the implementation can be found in [51].

5. Approach Evaluation

Besides the application example described in the previous section, which helped demonstrating the feasibility of the approach, we wanted to identify if the conceptual model behind PrivAPP is able to model existing privacy policies of some popular web applications. This can give an indication of PrivAPP potential success regarding privacy protection and help us to identify possible improvement directions in order to improve its completeness.

For this evaluation we used an empirical approach, which consists in selecting some privacy policies from relevant companies and analyzing them, verifying if elements from PrivAPP can help enforcing these policies. We split the privacy policy in statements and, for each of them, evaluate two aspects: first, if the statement can be represented by PrivAPP, and user preferences expressed about it. Second, if PrivAPP is able to describe elements that help enforcing the statement.

The details of the evaluation are described in the next sections.

5.1. Evaluation Setup

As the population of e-commerce websites in the world is inestimable, we established, empirically, the target of 20 privacy policies to be analyzed. Obviously, this number is arbitrary, and we cannot generalize the results of this evaluation to the universe of e-commerce companies for which privacy is very important. However, this list includes web stores with huge number of customers and sales. With respect to other smaller websites, we can argue that the selected ones should have more complex and refined privacy policies, because they are exposed to a larger set and frequency of privacy issues.

Therefore, we assumed that, for this study, the adopted set of policies provides a representative sample in the e-commerce domain, and can help up to evaluate if the approach is suitable for this domain.

We used two main criteria to select the companies and their respective privacy policies for our e-commerce sample set:

(i) Laws and regulations. As the privacy policies are based on privacy principles, laws and regulations, we decided to select companies from different

countries, including Brazil, USA and countries in the European Union. Ideally, the difference between regulations could reveal inconsistencies or the need for adding new elements in the model.

(ii) Size and market segment. We selected companies that are "top-of-mind" regarding the volume of sales and consumer preference. We based on lists of top Internet companies such as [52], which lists the top 50 online retail according to the revenues of online sales in fiscal year 2012, and [53], which lists the 250 major companies of the Brazilian retail in 2015 according to their gross revenues. The selected companies are online web stores that sell several kinds of products, including electronics, tourism, cosmetics, furniture, etc. We did not select Amazon [49] for this evaluation, because its privacy policy has already been used as an application example for the profile application (Section 4). The result of the selection is shown in Table 3.

Table 3. Selected companies and respective privacy policies to support the evaluation of PrivAPP

Company	Market Segment	Brazil	Other countries	Origin	Policy size	Statements
Americanas	Wide variety of products such as books; games; Cine & Photo; Mobile Phones; Electronics, etc.	Yes	no	Brazil	S	5
Casas Bahia	Home appliances, electronics, furniture and housewares.	Yes	no	Brazil	M	15
CVC	Tourism products and services.	Yes	no	Brazil	M	17
WalMart	Wide variety of products such as electronics, home appliances, computers, mobile phones; etc.	Yes	yes	USA	L	22
Dafiti	Shoes, clothes, accessories, sports products, perfumes, beauty products and decorative items	Yes	yes	Brazil	S	6
Cia dos Livros	Books	Yes	no	Brazil	S	8
Decolar	Tourism products and services	Yes	yes	USA	M	17
Aliexpress	Wide variety of products such as clothing, accessories, cars, motorcycles, cell phones, electronics, etc.	Yes	yes	China	L	26
Brigette's Boutique	Cosmetics, makeup, hair products	No	yes	USA	M	16
E-bay	E-commerce solutions to help individuals and companies to buy and sell products via Internet	Yes	yes	USA	L	45
Submarino	Wide variety of products such as books; games; mobile phones; electronics; watches, etc.	Yes	no	Brazil	S	4
DealeXtreme	Wide variety of products such as electronics, phones, electrical tools, car accessories, etc.	Yes	yes	China	L	32
Drugstore	Health, beauty, vision, and pharmacy products.	No	yes	USA	L	22
Mercado livre	E-commerce solutions to help individuals and companies to buy and sell products via Internet.	Yes	yes	Argentina	L	43
OLX / Bom Negocio	E-commerce solutions to help individuals and companies to buy and sell products via Internet.	Yes	yes	Argentina	M	20
Topshop	Clothes, shoes, bags and accessories, makeup.	No	Yes	United Kingdom	M	14
Media Markt	Wide variety of products such as flat-screen TVs, tablets, smartphones, coffee makers, etc.	No	yes	Germany	M	19
Worten	Home appliances, consumer electronics and entertainment.	No	yes	Portugal	S	7
Selfridges	Clothes, bags, makeup, cosmetics, perfumes, home appliances, mobile phones, tablets, wines, etc.	No	yes	United Kingdom	M	16
Carrefour	Supermarket, gas stations, drugstores and financial services.	Yes	yes	France	S	9

In Table 3, the selected companies and respective market segment are described. As most of this research was performed in Brazil, we adopted the perspective of a Brazilian user. So, the *Brazil* and *Other Countries* columns describe, respectively, where the companies operate. We can observe that 5 companies operate only in Brazil (Americanas, Casas Bahia, CVC, Cia Dos Livros, Submarino), 9 companies operate in Brazil and other countries (Walmart, Dafiti, Decolar, Aliexpress, E-bay, DealeXtreme, Mercado Livre, OLX/Bom Negócio, Carrefour) and 6 companies do not operate in Brazil (Brigette's Boutique, Drugstore, Topshop, Media Markt, Worten, Selfridges).

From the *Origin* column we can observe that 6 companies were originated in Brazil (Americanas, Casas Bahia, CVC, Dafiti, Cia dos Livros, Submarino), 5 were originated in USA (Walmart, Decolar, Brigette's Boutique, E-bay, Drugstore), 2 were originated in China (Aliexpress and DealeXtreme), 2 in Argentina (Mercado Livre, OLX/Bom Negócio) and 5 were originated in the European Union (Topshop, Media Markt, Worten, Selfridges, Carrefour). All this information was found in the *about us* links in the respective companies' websites.

We also classified the privacy policy size of each company (*policy size* column). We considered small (S) the policies that presented 10 or less statements, medium (M) the policies that presented 20 or less statements and large (L) the ones with more than 20 statements. Then, we have 6 small policies, 8 medium and 6 large ones. In total, we analyzed 351 statements.

5.2. Analysis and Results

Table 4 reports a mapping between the elements of our conceptual model, and the privacy policies of the 20 selected companies. Just for better organization, we split the approach's elements in two groups: the *fundamental elements* and the *enforcement elements*. The numbers represent the frequency with which each element is associated to the privacy policy. Although the privacy policies are publicly available, in the following the companies are not explicitly associated with any result, in order to assure neutrality and also because they usually do not permit the publication of the results of this type of evaluation. Therefore, the companies will be referred, from this point on, as 1 to 20, without any special order. We assume that all of them really comply with their privacy promises. Some discussions about the results are in the following.

Fundamental Elements

The most frequent element found, from the *Fundamental Elements* set, is the *Statement* (see last column in Table 4, with the totals). This element refers to statements that are generic, i.e., none of its specializations applies. Examples of *Statements* are: "*The User guarantees the truthfulness and accuracy of the personal data he/she provide to XXXX and assumes the corresponding responsibility*"; *"This online privacy policy applies only to information collected through our website and not to information collected offline.",* where XXXX is the name of the company. The high frequency of this element is because privacy policies typically present more statements than the ones strictly related to the management of private data (collection, retention, usage, disclosure).

The second most frequent element is the *Collection*. All the policies we analyzed have at least one statement that refers to data collection. These statements can specify the collection of personal identifiable information, users' activities (e.g. the links they click or the sites they access), users' system information (IP address,

operating system, web browser) or even generic data. Example of collection statements: *"Information including, but not limited to, user name, address, phone number, fax number and email address ("Registration Information") may be collected at the time of user registration on the XXXX."; "We record and retain details of users' activities on the XXXX. […]"*. 95% of the policies state that the collection of users' activities and system information is done through cookies, web beacons and similar technologies.

Table 4. PrivAPP's elements corresponding to company's privacy policies.

	Element	Companies																				Total
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	
Fundamental elements	Privacy Policy Definition	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	20
	Statement	2	10	4	12	3	2	5	8	5	19	2	11	7	6	16	4	5	4	17	7	149
	Disclosure	1		1	1	1	2	4	8	3	13	1	1	2	3	1	1	2	1	7	3	56
	Retention		1	3	2			2			2		6	2		1	1		1	3		24
	Collection	2	4	4	5	1	3	5	9	6	6	1	7	10	3	2	1	2	9	8	2	90
	Usage		1	2	3	1	1	4	3	1	4		6	1	2	2	1	1	2	9	8	52
	Recipient	1		1	1	1	1	4	8	3	13	1	1	2	2	1	1	2	1	3	3	50
	Service	1	1	2	3	1	1	2	2	1	6	1		3	2	2		1	2	2	1	34
	Private Data		1	1	1		1	1	1		1	1	1	2			1	1		1		14
	Usage Information	1	1	1	1	1		1		1	1	1	2	1	2	1		1	5	1	1	23
	Personal Identifiable Information			1			1	1	3	1	2		1	1	1	1			1	1	1	16
	Preference	1	2	3	3	1	1	7	3	5	11	1	4	7	6	3	1	1	8	9	6	83
Enforcement elements	Enforcement												1									1
	MonitoringTool																					0
	Activity Tracking Detection	1	1	1	2	1	1	2	1	1	2	1	3	4	2	1				3	1	28
	Privacy Violation Detection	1			1					1	1	1							1	1	1	8
	Security Measure		1	2	3	1			2	2	2		1	1		1				2		18
	Attack Detection		1	1	2				1		1		1	1						2		10
	User Pattern Identification	1	1	1	1		1	1	1			1								2		10
	Auditing	1	2	2	3	1	2		1	2	7	1	6	4	4	2	2	1	1	11	2	55
	Action		1	1	2	1		2	1		9		1	4	3	4	3	1		8	2	43
	Process																					0
	Access Control	2		2	2			2	3	3	6	1	4	2	2				1	3	2	35
	Access Control Policy	2		2	2			2	3	3	6	1	4	2	2				1	3	2	35
	Identity Management												1							1		2
	Cryptography		1	1	1	1			1	1	1		1	1		1				2		12
	Anonymization		1		1	1			2									1				6
	Algorithm																					0
	Config	1	2	1	1	2	1	4	2	7	9	1	6	10	3	5	1	3	9	8	5	81
	Management	1	1		3		1	4	3	2	2	1	2	4	2	1		1				28
	Managed Interaction				2			3	2	1	2		1		1							12
	Total	20	34	38	59	19	20	57	69	50	127	18	73	72	47	46	18	24	48	108	48	

Preference is the third most frequent element found from the *Fundamental Elements* set. We considered as *Preference* the statements that offer the user the option

to choose, i.e., to agree or disagree (opt-in or opt-out) with the referred statement. Example: *"We may also send you from time to time (by email or post) information about products and services and details of promotions and special offers from XXXX"; A Cookie is […]. We use Cookies to keep track of your current shopping session […]"*. For all these statements classified as *Preference*, the users could say that they opt-out, i.e., they do not want to receive e-mails with advertisements or to have their activities tracked. In these cases, the companies need to take some actions in order to respect these preferences (and the enforcement elements of our approach can help in this direction). Expressing the preference for the statements would be very useful to provide more flexibility to the users, allowing them to make more thoughtful choices about the collection and the use of their personal information online. Consequently, it would provide more respect to the user privacy, increasing the credibility of the company.

Enforcement Elements

Config is the *Enforcement* element that has been most widely used in our analysis, with 81 occurrences. We associated the *Config* element with statements that need two types of configurations to enforce the policy, even in the cases where users express their preferences: *web browser configurations* and *user configurations*. *Web browser configurations* is from the Reference Architecture [32] and, although it is a configuration outside the system (i.e. each user must configure its own web browser), we believe this is an important resource that must be made explicit at least in the privacy policy. Guiding the user to configure their web browser would help to respect their privacy, especially when they do not want to receive cookies or be tracked. *User configurations* is an instance of *Config* that we created to represent the enforcement of statements where the system itself allows users to refuse some services as, for example, advertisements or cookies and similar. Example of statements for user configurations: *"If your personally identifiable information changes, or if you no longer desire our service, you may correct, update, request deletion, or deactivate it by making the change on the "your account" page or by e-mailing us at privacy@XXXX.com".*

Auditing is the second most frequent enforcement suggestion, with 55 occurrences. As the privacy policies must be in accordance with the laws and regulations, many times the statements refer to the use of private data in order to comply with them. So, it is necessary to evaluate if the data is really being used for these legal purposes. Some statements that need auditing to be enforced are, for example: "*Your Data may be retained beyond the expiry of its purpose if that is required by law, such as a provision of a statute, or a court order such as a search warrant or subpoena, or a warning by a law enforcement agency that delivery of a court order is imminent*". Also, we used the Auditing for the enforcement of statements that disclose the private data with specific goals, as *"As part of the customer data management, the data collected will be transmitted to third parties, the transport companies, for the exclusive purpose of the realization of the services or products purchased by the user."*. Auditing can be a complex and expensive resource, but we believe this is necessary especially in the cases that involve laws.

Action is an element that represents mechanisms that the system could implement to help protecting privacy. It can have a wide variety of instances and some of that we created are: *notify policy changes* (to notify the users in case of changes in the privacy policy and, if necessary, ask them to express their new preferences); *inform user about automatic collection* (to inform the user when cookies are sent or other mechanisms will track the activities, allowing the user to express their preference,

agreeing or not); *do not send text message* (when statements say that text messages will be sent to the cell phone and the user disagree with it).

The *Access Control* and respective *Access Control Policy* elements have also been widely used. We adopted these elements in the cases where the statements cite that only qualified and authorized staffs are allowed to access personal data and in the cases where they cite the disclosure of the private data to third-parties. It is evident that controlling the disclosure goes far beyond of just access control and we just use the most adequate element from the reference model. *Auditing* could be a good complementary solution to be added in this process.

Summary of results

We wanted to verify if the proposed approach can fit some privacy policies with necessary information for helping privacy protection and, also, to identify if the approach and their respective models needed some improvement. In Table 4 we can observe, in the last column, that almost all elements were found in the policies, except *MonitoringTool*, *Process* and *Algorithm*. Although we did not found any specific corresponding element for these ones, we used their specializations (*Activity Tracking Detection, Privacy Violation Detection, Access Control, Identity Management, Cryptography, Anonymization*). So, the approach still offers some generic elements that could be used for statements referring to resources that we can associate to them and that are not too specific as their specializations. The fact of the generic elements have been rarely used is an indication that the specialization of these elements is, at least for the analyzed policies, adequate.

Also in Table 4, we can observe that, from the 351 analyzed statements, we used 384 enforcement suggestions. This number is due to the fact that some statements required more than one enforcement measure. The enforcements solutions considered in our approach were applicable to all the companies, varying from the minimum of 6 (companies 6 and 16) to the maximum of 48 (company 10) suggestions per company. This analysis indicates that, for the set of analyzed policies, the identified enforcement elements can support the fulfillment of the policies.

In performing this analysis, we did not find any new element that could be added to the approach. However, we noted that we could refine the *Config* element in two separate instances, that we were using with high frequency: *Web browser configurations* and *User configurations* (their descriptions are in the Section 3.1). These two specializations, which were not present in the first version of the PrivAPP conceptual model, have been added as a side-result of this evaluation, as a refinement of the initial model.

Finally, although the *Statement* element also has a high frequency of occurrence in this study, we could not identify additional representative groups that could result in other specializations with respect to the ones already in PrivAPP conceptual model, i.e., statements modeled with the generic *Statement* were quite different from each other.

6. Conclusions

In this paper we propose PrivAPP, which is an integrated approach to guide the design of privacy-aware applications. The main goal of PrivAPP is to contribute in improving the current lack of privacy protection in the scope of web applications and services.

In the proposed approach, the Privacy Conceptual Model shows the privacy elements and their relations in an organized way, systematizing privacy concepts in the domain of web applications. With this, we have a model of the domain concepts for modeling views of the system where privacy management and protection are applied. The Reference Architecture provides a detailed description of the functionalities that have to be addressed in the implementation of web applications and services, helping to protect personal information privacy. The UML Profile is used to describe the privacy policy that is applied by an application, and to keep track of which elements are in charge of enforcing it, e.g., for tracking of privacy requirements or for documentation purposes. The direct relation between the Reference Architecture and the UML profile allows privacy-related components to be immediately identified within the software architecture, and their role highlighted with domain-specific stereotypes.

Based on the application example and the evaluation process we performed, we can state that PrivAPP introduces some benefits for software developers and business professionals: first, the elements which compose the approach serve as a guideline for the design of concrete architectures that support web applications and services with privacy protection features. The models derived from the approach, i.e., UML diagrams and software architecture, provide resources for the documentation of privacy specifications of web applications, helping to structure particular concepts of privacy. They facilitate the understanding of the privacy domain by the stakeholders and are useful for them to communicate and to support the discussions on the general analysis of privacy resources when dealing with web applications. Consequently, these models support a faster development of privacy issues, by letting the programmers free from the task to decide which technology to use in order to enforce the privacy policy.

An application example was performed, applying the proposed approach in the construction of a web application with privacy protection. The implementation of a solution regarding some elements of the approach allowed a partial observation of its capability to be applied in practice. Also, an evaluation of PrivAPP was performed through an empirical study, where a set of privacy policies from relevant companies were analyzed, verifying to what extent the elements from PrivAPP were able to describe and help enforcing these policies. Although the evaluation process is limited to a set of 20 policies, they include some of the most representative e-commerce websites from different countries and different market segments.

The results indicate that the PrivAPP approach is suitable for managing privacy concerns and documenting enforcement solutions in the development of web applications. In an era where digital information has immense value and privacy is a *must have*, we believe that the proposed approach helps to improve the process of designing web applications in the privacy domain, integrating privacy-related information in the development process of a web application.

As future work we intend to apply the approach to larger web applications with privacy requirements, and to investigate the degree of adaptability of the approach on cloud environments, providing extensions and adaptations, if necessary.

Acknowledgments

This work has been partially supported by the project DEVASSES (DEsign, Verification and VAlidation of large-scale, dynamic Service SystEmS), funded by the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no PIRSES-GA-2013-612569. We also thank the project EU-BRA BIGSEA (*Europe – Brazil Collaboration of BIG*

Data Scientific Research through Cloud-Centric Applications), funded by *European Commission Information Society and Media* (grant agreement n. 690116), *Ministério de Ciência, Tecnologia e Inovação - MCT e Rede Nacional de Ensino e Pesquisa – RNP.*

References

[1] Westin, A. Privacy and Freedom. Bodley Head, 1987.

[2] Wang H, Lee M, Wang C. Consumer privacy concerns about Internet marketing. In Commun. ACM, 1998; **41**(3): 63-70.

[3] Bertino E, Lin D, Jiang W. A Survey of Quantification of Privacy Preserving Data Mining Algorithms. Privacy-Preserving Data Mining,2008, vol. 34, C. C. Aggarwal, P. S. Yu, and A. K. Elmagarmid, Orgs. Springer US;183–205.

[4] European Union. The European Union Directive 95/46/EC: On The Protection Of Individuals With Regard To The Processing Of Personal Data And On The Free Movement Of Such Data, February 20 1995. www.privacy.org/pi/intl_orgs/ec/eudp.html. [14 February 2016].

[5] Canada. The Personal Information Protection and Electronics Document Act: Bill C6. www.parl.gc.ca/36/2/parlbus/chambus/house/bills/government/C-6/C-6_3/C-%6_cover-E.html. [14 February 2016].

[6] Australia. Privacy Act 1988, 1988. www.privacy.gov.au/act/index.html. [14 February 2016].

[7] HIPAA States. The Health Insurance Portability and Accountability Act of 1996 (HIPAA), October 1998. www.hcfa.gov/hipaa/hipaahm.html. [17 March 2016].

[8] United States. Gramm-Leach-Bliley Act: Financial Privacy and Pretexting, November 12 1999. www.ftc.gov/privacy/glbact/glboutline.htm. [17 March 2016].

[9] COPPA. Children's Online Privacy Protection Act of 1998 (COPPA), October 1998. www.cdt.org/legislation/105th/privacy/coppa.html. [17 March 2016].

[10] Truste. "TRUSTe Privacy Index. 2015 Consumer Confidence Edition". htttps://www.truste.com/resources/privacy-research/us-consumer-confidence-index-2015/. [17 March 2016].

[11] Object Management Group. OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1. OMG Document {formal/2011-08-06}. Aug. 2011.

[12] WhatIs. "Definition - Privacy Policy". http://whatis.techtarget.com/definition/privacy-policy. [23 March 2016].

[13] Mont M C, Pearson S, Creese S, Goldsmith M, Papanikolaou N. A Conceptual Model for Privacy Policies with Consent and Revocation Requirements. Privacy and Identity Management for Life. IFIP Advances in Information and Communication Technology 2011; **352**: 258-270.

[14] Object Management Group. UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms (OMG QoS&FT), Version 1.1. OMG Document {formal/2008-04-05}. Apr. 2008

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

[16] Cranor L, Dobbs B, Egelman S, Hogben G, Humphrey J, Langheinrich M, Marchiori M, Presler-Marshall M, Reagle J M, Schunter M, Stampley D A, Wenning R. The Platform for Privacy Preferences 1.1 (P3P1.1) Specification. World Wide Web Consortium NOTEP3P11-20061113, 2006.

[17] Ashley P, Hada S, Karjoth G, Powers C, Schunter M. Enterprise Privacy Authorization Language (EPAL 1.2), 2003. http://www.zurich.ibm.com/security/enterprise-rivacy/epal/Specification/index.html. [23 March 2016].

[18] CherdantsevaY, Hilton J. A Reference Model of Information Assurance & Security. Eighth International Conference on Availability, Reliability and Security (ARES), 2013;546-555.

[19] Sathiyamurthy S. The Struggle for Privacy and the Survival of the Secured in the IT Ecosystem. ISACA Journal, 2011; **2**:1-7.

[20] OASIS. "Privacy Management Reference Model and Methodology (PMRM) Version 1.0",2012. http://docs.oasis-open.org/pmrm/PMRM/v1.0/csd01/PMRM-v1.0-csd01.pdf. [23 March 2016].

[21] Mohammadi N G, Bandyszak T, Paulus S, Meland P H, Weyer T, Pohl K. Extending development methodologies with trustworthiness-by-design for socio-technical systems. Trust, 2014; 206-207.

[22] Dardenne A, Fickas S, Lamsweerde A V. Goal-Directed Requirements Acquisition, Science of Computer Programming, 1993; **20**(1-2): 3-50.

[23] Mouratidis H, Giorgini P. Secure Tropos: a Security-Oriented Extension of the Tropos Methodology. International Journal of Software Engineering and Knowledge Engineering 2007; **17**(02): 285-309.

[24] Giunchiglia F, Mylopoulos J, Perini A. The Tropos Software Development Methodology: Processes, Models and Diagrams. Lecture Notes in Computer Science, 2003; **2585**:162–173.

[25] Kalloniatis C, Kavakli E, Gritzalis S. Addressing privacy requirements in system design: the PriS method. Requirements Engineering, 2008; **13**(3):241-255.

[26] Nakagawa E Y, Oquendo F, Becker M. RAModel: A Reference Model for Reference Architectures. Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012; 297-301.

[27] ISO/IEC 29101. International Standard - Information technology - Security Techniques - Privacy architecture framework. First Edition, 2013.

[28] Shin Y N, Chun W B H, Jung S, Chun M G. Privacy Reference Architecture for Personal Information Life Cycle. Advanced Communication and Networking, T Kim T, Adeli H, Robles R J, Balitanas M. Orgs. Springer Berlin Heidelberg, 2011; 76–85.

[29] Bücker A, Haase B, Moore D, Keller M, Kobinger O, Wu H-F. IBM Tivoli Privacy Manager. Solution Design and Best Practices. IBM Redbooks, 2003.

[30] Mont M C, Thyne R, Bramhall P. Privacy Enforcement with HP Select Access for Regulatory Compliance. Hewlett-Packard Company, 2005.

[31] Federal Trade Commission. Privacy Online: Fair Information Practices in the Electronic Marketplace: A Federal Trade Commission Report to Congress, 2000. https://www.ftc.gov/reports/privacy-online-fair-information-practices-electronic-marketplace-federal-trade-commission. [18 March 2016].

[32] Basso T, Moraes R, Jino M, Vieira M. Requirements, Design and Evaluation of a Privacy Reference Architecture for Web Applications and Services. Proceedings of the 30th ACM/SIGAPP Symposium on Applied Computing, Salamanca, Spain, 2015.

[33] Object Management Group, "Model Driven Architecture (MDA) – MDA Guide rev. 2.0", OMG Document ormsc/2014-06-01. June 2014.

[34] Fink T, Koch M, Pauls K. An MDA approach to Access Control Specifications Using MOF and UML Profiles. Electronic Notes in Theoretical Computer Science 2006; **142**:161–179.

[35] Hsu I-C. Extending UML to model Web 2.0-based context-aware applications. Software Practice and Experience, 2012; **42** (10):1211–1227.

[36] Cirit Ç, Buzluca F. A UML profile for role-based access control. Proceedings of the 2nd International conference on Security of information and networks (SIN'09), ACM, New York, USA, 2009; 83-92.

[37] Jürjens J. UMLsec: Extending UML for Secure Systems Development. In Proceedings of the 5th International Conference on The Unified Modeling Language (UML '02), Jean-Marc Jézéquel, Heinrich Hußmann, and Stephen Cook (Eds.). Springer-Verlag, London, UK, 2002;412-425.

[38] Zoughbi, G, Briand L, Labiche Y. Modeling safety and airworthiness (RTCA DO-178B) information: conceptual model and UML profile. Softw Syst Model, 2011; **10**(3): 337-367.

[39] Zhu L, Staples M, Tosic V. On Creating Industry-Wide Reference Architectures. Proceedings of the 12th International IEEE Enterprise Distributed Object Computing Conference, Munich, Germany, 2008; 24-30.

[40] Khosrowpour M. Issues & Trends of Information Technology Management in Contemporary Organizations. Information Resources Management Association International Conference, Seattle, Washington, USA, 2002; 497-499.

[41] Organization for Economic Co-operation and Development: OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, 2013. http://www.oecd.org/sti/ieconomy/2013-oecd-privacy-guidelines.pdf [26 August 2014].

[42] Cavoukian A. Creation of a Global Privacy Standard, 2006. https://www.ipc.on.ca/images/resources/gps.pdf. [23 March 2015].

[43] ISO/IEC 29100. International Standard - Information technology - Security Techniques - Privacy framework. First Edition, 2011.

[44] ISO/IEC 29101. International Standard - Information technology - Security Techniques - Privacy architecture framework. First Edition, 2013.

[45] Basso T, Montecchi L, Moraes R, Jino M, Bondavalli A. Towards a UML Profile for Privacy-Aware Applications. 15th IEEE International Conference on Computer and Information Technology (CIT-2015), Liverpool, UK., 2015.

[46] Nakagawa E Y, Guessi M, Maldonado J C, Feitosa D, Oquendo F. Consolidating a Process for the Design, Representation, and Evaluation of Reference Architectures. Proceedings of the 2014 IEEE/IFIP Conference on Software Architecture (WICSA '14). IEEE Computer Society, Washington, DC, USA, 2014; 143-152.

[47] TPC-W. Transaction Processing Performance Council. http://www.tpc.org/tpcw/. [18 March 2016].

[48] Garcia, D.F., Garcia. J. TPC-W e-commerce benchmark evaluation. Computer, 2003; **36** (2):42-48.

[49] Amazon.com Privacy Notice. http://www.amazon.com/gp/help/customer/display.html/ref=footer_privacy?ie=UTF8&nodeId=468496. [18 March 2016].

[50] Aldawud O, Elrad T, Bader A. UML Profile for Aspect-Oriented Software Development. The Third International Workshop on Aspect Oriented Modeling, Boston, USA, 2003.

[51] Basso T, Piardi L, Moraes R, Jino M, Antunes N, Vieira M. A Database Framework for Expressing and Enforcing Personal Privacy Preferences. In: XVI Workshop of Tests and Fault Tolerance (WTF2015), Vitória, Brazil, 2015.

[52] G1. Brazil has two of the top 50 online retail, research shows. G1 Economy and Bbusiness. http://g1.globo.com/economia/negocios/noticia/2014/01/brasil-tem-2-entre-50-maiores-do-varejo-online-mostra-pesquisa.html. [18 March 2016].

[53] SBVC. Ranking sbvc - the 250 major companies of the Brazilian retail 2015. http://www.sbvc.com.br/wp-content/uploads/2015/09/edicao-39016cfe079db1bfb359ca72fcba3fd8.pdf. [18 March 2016].

[54] J. Siegel, “Using OMG’s Model Driven Architecture (MDA) to Integrate Web Services”, 2002. http://www.omg.org/mda/mda_files/MDA-WS-integrate-WP.pdf. Last accessed 12/07/2017.