Supporting Privacy by Design Using Privacy Process Patterns

Vasiliki Diamantopoulou, Christos Kalloniatis, Stefanos Gritzalis, Haralambos Mouratidis

Supporting Privacy by Design using Privacy Process Patterns

Vasiliki Diamantopoulou[1], Christos Kalloniatis[21], Stefanos Gritzalis[3], and Haralambos Mouratidis[1]

[1]School of Computing, Engineering and Mathematics
University of Brighton, Brighton, UK
{v.diamantopoulou,h.mouratidis}@brighton.ac.uk
[2]Department of Cultural Technology and Communication
University of the Aegean, Lesvos, Greece
[3]Department of Information and Communication Technologies
University of the Aegean, Samos, Greece
{chkallon,sgritz}@aegean.gr

Abstract. Advances in Information and Communication Technology (ICT) have had significant impact on every-day life and have allowed us to share, store and manipulate information easily and at any time. On the other hand, such situation also raises important privacy concerns. To deal with such concerns, the literature has identified the need to introduce a Privacy by Design (PbD) approach to support the elicitation and analysis of privacy requirements and their implementation through appropriate Privacy Enhancing Technologies. However, and despite all the work presented in the literature, there is still a gap between privacy design and implementation. This paper presents a set of Privacy Process Patterns that can be used to bridge that gap. To demonstrate the practical application of such patterns, we instantiate them in JavaScript Object Notation (JSON), we use them in conjunction with the Privacy Safeguard (PriS) methodology and we apply them to a real case study.

Keywords: Privacy Process Patterns, Requirements Engineering, Information Security Modelling

1 Introduction

Information Privacy is considered as an important challenge for Information and Communication Technology (ICT). With more and more sensitive and confidential information stored, shared and manipulated at digital level [1], both individuals and organisations expect appropriate measures to ensure privacy of such information. However, this is not easy, as privacy is a multifaceted concept with various impact and ways of achievement which depends, amongst other things, on the environments in which it is required to be achieved.

Although the paradigm of Privacy by Design (PbD) has been proposed as a feasible solution to such situation, there are still major challenges that require further research and development. In particular, a challenging task in the

context of PbD is moving from a design (where the privacy requirements of an information system have been elicited) to an implementation that fulfils those requirements. This is problematic for two main reasons. On one hand, there is little expertise on how best to align privacy requirements (from the design stage) to Privacy Enhancing Technologies (PETs) [2] at implementation stage. On the other hand, software engineers, who need to deal with both the design and the implementation stages, lack detailed knowledge of PETs to ensure correct implementation. This paper contributes towards these two challenges by proposing a set of Privacy Process Patterns to enhance detailed knowledge of PETs and a clear alignment between privacy properties (requirements) and PETs. Moreover, we are demonstrating how these patterns can be used as part of a privacy-aware methodology to bridge the gap between design and implementation. To improve the usability of such patterns, we instantiate them with JavaScript Object Notation (JSON), using a template that could be adapted by any programming language. Moreover, we present our patterns in the context of an existing privacy-aware methodology called PriS [3] and we apply our work to a real case study to illustrate practical applicability of the work.

The paper is structured as follows. Section 2 discusses the related work, while Section 3 presents the Privacy Process Patterns. Section 4 describes their implementation and Section 5 illustrates their application to a case study. Finally, Section 6 concludes the paper.

2 Related Work

Patterns have been adopted into software engineering as they encounter each problem in a systematic and structured way. Privacy patters, specifically, have been used as a way to model privacy issues. In [4] privacy patterns are used for web-based activity and especially for conveying privacy policies to end-users during online interactions. Traditional design patterns are described in [5], identifying 45 patterns for the design in ubiquitous computing environments, 15 of which focused on privacy. The authors in [6] propose a pattern language which contains 12 patterns for developing anonymity solutions for various domains, including anonymous messaging, anonymous voting and location anonymity. This work moves on the right direction regarding the modelling of privacy requirements but it fails to combine privacy elicitation concepts for capturing privacy requirements. In [7], six patterns that focus on how to establish boundaries for interaction are presented, focusing on the filtering of personal information in collaborative systems. Finally, the author in [8] presented two privacy patterns, applying this approach to security issues by proposing a set of security patterns to be applied during the software development process.

3 Privacy Process Patterns

Privacy Process Patterns are patterns being applied on privacy related processes in order to specify the way that the respective privacy issues will be realised

through a specific number of steps, including activities and flows connecting them. They assist developers to understand, in a better and more specific way, how to implement the various privacy properties. The use of Privacy Process Patterns is considered as a more robust way for bringing the gap between the design and the implementation phase of a system or module of it.

The proposed pattern structure follows the so-called Alexandrian format [9] which is already accepted and used for the definition of security patterns [10]. This format is efficient enough for the description of the Privacy Process Patterns, matching the fields of each pattern when this is expressed with JSON. Through *definition* field, we give a comprehensive definition of the property. The fields *problems* and *forces* present the goals that need to be fulfilled and the forces that need to be considered when choosing to use this pattern, respectively. The fields *benefits* and *liabilities* present the advantages and the disadvantages that are identified in each privacy property. The field *implementation techniques* covers all the possible techniques that satisfy the respective property. From the range of the proposed implementation techniques, the developers can choose the most appropriate technology based on the privacy process patterns applied on every privacy-related process. Finally, the field of *related patterns* indicates which patterns have similar characteristics with the examined one, which patterns are closely related in terms of functionality and with which other patterns it can be utilised.

This work describes the five basic privacy properties [11], [12], [13], [14] namely *anonymity*, *pseudonymity*, *unlinkability*, *undetectability* and *unobservability*. Our intention is to define a general template for privacy properties that can be used to describe other properties additionally to the five we enlisted above. This is a preliminary work aiming to identify all possible privacy concepts that need to be addressed when designing privacy-aware systems and provide a structured description in order for the developers to take advantage of and manage to handle privacy in a robust way, linking the gap between design and implementation phases. The impact of the selection of respective privacy concepts and the complexity of their applicability is a very interesting topic, but it is not the main focus of this paper. This template comprises a guide for the developers who can understand in a better and more structured way how to implement each privacy concept.

3.1 Anonymity

- *Definition*: Anonymity is a characteristic of information that does not permit a personally identifiable information principal to be identified directly or indirectly. During anonymization, identity information is either erased or substituted
- *Problem*: The user of a service cannot be identified
- *Forces*: Large number of users in the same network is required
- *Benefits*: i) Supports users in accessing services without disclosing their identity, ii) Users are more freely expressed, since freedom from user profiling is

achieved (behaviour of users or other privacy-infringing practices), iii) Freedom from location tracking, iv) Minimal user involvement (they do not have to modify their normal activities for anonymity services)
- *Liabilities*: i) Maintain users' accountability while anonymous, Performance (latency, loss of functionality, bandwidth, etc.), ii) Usability of information (too much data obfuscation can undermine the usefulness of data), iii) Abuse of privacy (malicious users), iv) User count (large anonymity set), v) User friendliness (if the users have to adapt a lot to achieve anonymity, they may start judging where they should have anonymity), vi) Law enforcement (the anonymity might have to be liftable to investigate on crime suspects)
- *Implementation techniques*:
  - Anonymizer products, services and architectures: Browsing pseudonyms [15], Virtual Email Addresses, Trusted third parties, Crowds [16], Onion routing [17], DC-nets [18], Mix-nets (Mix Zone) [19], Hordes [20], GAP [21], Tor [22], Aggregation Gateway [23], Dynamic Location Granularity
  - Track and evident erasers: Spyware detection and removal, Hard disk data eraser, User data confinement pattern, Use of dummies
- *Related patterns*: Pseudonymity, unlinkability

3.2 Pseudonymity

- *Definition*: Pseudonymity is the utilisation of an alias instead of personally identifiable information
- *Problem*: Ensuring that an entity cannot be linked with a real identity during online interactions
- *Forces*: Use authenticated services without disclosing identifiable information
- *Benefits*: i) Supports users in accessing services without disclosing their real identity, ii) Permits the accumulation of reputational capital, iii) The user is still accountable for its actions, iv) A user may have a number of pseudonyms, v) Fills the gap between accountability and anonymity, vi) Hides the identity of the participants, vii) Prevents unforeseen ramifications of the use of online services
- *Liabilities*: i) Maintains users' accountability while pseudonymous, ii) Abuse of privacy (malicious users) iii) Forgery/impersonation, iv) Law enforcement (the anonymity might have to be liftable to investigate on crime suspects), v) Extensive usage of the same pseudonym can weaken it
- *Implementation techniques*:
  - Administrative tools: Identity management, Biometrics [24], Smart cards [25], Permission management
  - Pseudonymizer tools: CRM personalisation [26], Application data management, Obligation management, Mixmaster
- *Related patterns*: Anonymity, authentication

3.3 Unlinkability

- *Definition*: Unlinkability is the use of a resource or a service by a user without a third party being able to link the user with the service

- *Problem*: i) Users' identifiable information is not protected, ii) The strength of unlinkability is depended on the number of nodes belonging to the unlinkability set
- *Forces*: Enforce users' privacy regarding the linkability with the service used
- *Benefits*: i) Protect users' privacy when using a resource or service by not allowing malicious third parties to monitor which services are used by the user, ii) The intentional severing of the relationships (links) between two or more data events and their sources, ensures that a user may make multiple uses of resources or services without others being able to link the uses together, iii) Requires that users and/or subjects are unable to determine whether the same user caused certain specific operations in the system, iv) Minimise risks to the misuse of the privacy-relevant data and to prohibit or restrict profiling
- *Liabilities*: i) Maintain a large unlinkability set, ii) Equal distribution of traffic between the potential senders and the potential recipients, iii) Unidirectional pseudonyms should be preferred because omnidirectional pseudonyms are susceptible to profiling
- *Implementation techniques*:
  - Anonymizer products, services and architectures: Trusted third parties, Surrogate keys, Onion routing, DC-nets, Mix-nets, Hordes, GAP, Tor, Aggregation Gateway
  - Pseudonymizer tools: CRM personalisation, Application data management
  - Track and evident erasers: Spyware detection and removal, Browser cleaning tools [27], Activity traces eraser, Hard disk data eraser, Use of dummies, Identity Federation Do Not Track Pattern
- *Related patterns*: Undetectability, anonymity

3.4 Undetectability

- *Definition*: Undetectability is the inability for a third party to distinguish who is the user (among a set of potential users) using a service
- *Problem*: The strength of undetectability depends on the number of nodes belonging to the undetectability set
- *Forces*: Enforce users' privacy by allowing them to use a service without being detected by a malicious third party
- *Benefits*: i) Protect users' privacy when using a resource or service by not allowing malicious third parties to detect which services are used by the user, ii) The attacker cannot sufficiently detect whether a particular Item of Interest (IOI) exists or not, e.g. steganography, iii) The attacker cannot sufficiently distinguish whether it exists or not
- *Liabilities*: i) Maintain a large undetectability set, ii) Equal distribution of traffic between the potential senders and the potential recipients
- *Implementation techniques*:
  - Administrative tools: Smart cards, Permission management
  - Information tools: Monitoring and audit tools

  - Anonymizer products, services and architectures: Hordes, GAP, Tor
  - Track and evidence erasers: Spyware detection and removal, Browser cleaning tools, Activity traces eraser, Hard disk data eraser, Identity Federation Do Not Track Pattern
  - Encryption tools: Encrypting email [28], Encrypting transactions [29], Encrypting documents
- *Related patterns*: Unlinkability, unobservability

3.5 Unobservability

- *Definition*: Unobservability is the inability of a third party to observe if a user (among a set of potential users) is using a service
- *Problem*: The strength of unobservability set depends on the strength of: i) The sender/recipient anonymity set, ii) The sender/recipient undetectability set
- *Forces*: Users privacy is enforced since they can use a resource or service anonymously and without being detected regarding the service used when the state of IOIs should be indistinguishable from any IOI (of the same type) at all when we want to send messages that are not discernible from e.g. random noise
- *Benefits*: i) Anonymity and Undetectability enforcement per service, ii) Ensures that a user may use a resource or service without others, especially third parties, being able to observe that the resource or service is being used, iii) Requires that users and/or subjects cannot determine whether an operation is being performed
- *Liabilities*: i) Depends on the successful implementation of both anonymity and undetectability, ii) Strong encryption required demanding many resources, iii) Slower communication due to complex calculations
- *Implementation techniques*:
  - Administrative tools: Smart cards, Permission management
  - Anonymizer products, services and architectures: Hordes, GAP, Tor
  - Track and evidence erasers: Spyware detection and removal, Hard disk data eraser, Identity Federation Do Not Track Pattern
- *Related patterns*: Anonymity, undetectability

4 Privacy Process Patterns Implementation

4.1 PriS methodology

The implementation of the aforementioned Privacy Process Patterns follows an abstract approach, enabling them to be applied to any requirements engineering methodology. In order to substantiate the applicability and usefulness of the Privacy Process Patterns that have been presented in Section 3, we opted to apply them on a privacy requirements engineering methodology, i.e. PriS (Privacy Safeguard). This methodology incorporates privacy requirements into the

system design process and has been developed so as to assist designers on eliciting, modelling, designing privacy requirements of the system to be and also to provide guidance to the developers on selecting the appropriate implementation techniques that best fit the organisation's privacy requirements. PriS provides a set of concepts for modelling privacy requirements in the organisation domain and a systematic way-of-working for translating these requirements into system models, adopting the use of Privacy Process Patterns as a way to i) describe the effect of privacy requirements on business processes and ii) facilitate the identification of the system architecture that best supports the privacy-related business processes. PriS methodology comprises the following four activities that are presented below in an abstract way, as the implementation of them will be thoroughly described in Section 5, through a real case study:

1. *Elicit privacy-related goals.* This step concerns the elicitation of the privacy goals that are relevant to a specific organisation. It usually involves a number of stakeholders and decision makers (managers, policy makers, system developers, system users, etc.)
2. *Analyse the impact of privacy goals on organisational processes.* The second step is to analyse the impact of these privacy goals on processes and related support systems
3. *Model affected processes using privacy process patterns.* Having identified the privacy-related processes, the next step is to model them, based on the relevant privacy process patterns
4. *Identify the technique(s) that best support/implement the above process.* The final step is to define the system architecture that best supports the privacy-related process identified in the third step. Again, privacy process patterns are used to identify the proper implementation technique(s) that best support/implement corresponding processes

The proposed framework uses the concept of *goal* as the central and most important concept. Goals are desired state of affairs that need to be attained. Goals concern *stakeholders*, i.e. anyone that has an interest in the system design and usage. Also, goals are generated because of issues. An issue is a statement of a strength, weakness, opportunity or threat that leads to the formation of the goal. Privacy is a highly regulated area in Europe. The protection of users' privacy is stated in many European and national legislations through the form of laws, policies, directives, best practices, etc. [30]. Thus, *legal issues* need to be taken under consideration during the identification of functional and non-functional requirements. Goal identification needs to take under consideration all these elements before further analysis is conducted.

As shown in Figure 1, there are two types of goals in the proposed framework, namely *organisational goals* and *privacy goals*. Organisational goals express the organisation's main objectives that need to be satisfied by the system into consideration. In parallel, privacy goals are introduced because of specific *privacy related properties*. Through the privacy goals, the *realisation* of the identified privacy properties is achieved. Thus, all privacy related properties that need to

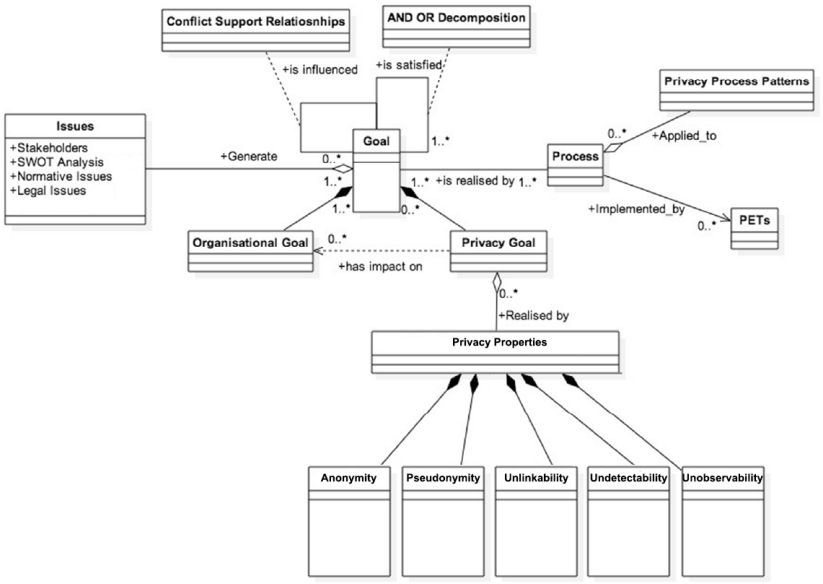

Fig. 1: Conceptual model

be realised, should be addressed as specific privacy goals. Privacy goals may have an *impact* on organisational goals. In general, a privacy goal may cause the improvement/ adaptation of organisational goals or the introduction of new ones. In this way, privacy issues are incorporated into the system's design. Every model has at least one organisational goal, but may have no privacy goals, thus the respective relationships (1..* and 0..*) among the organisational and privacy goal with the generic concept of goal. Goals are realised by *processes*. The relationship between goals and subgoals is many to many, in the sense that one goal can be realised from one or more processes and one process can support the realisation of one or more goals.

4.2 Expression of PriS with JSON Format

Another reason for choosing PriS in order to apply the proposed structure of the Privacy Process Patterns is their expression of its structure in JSON format [31] and the reasoning it facilitates through this format. Prior to PriS, the transmission from the design to the implementation phase was vague; developers did not have a methodology to automate this process, i.e. the selection of most suitable privacy enhancing technologies to apply in their context. PriS extended version is expressed in the JSON format, which is an Open Standard used to transmit data objects consisting of attribute-value pairs [32]. The *attribute* is immutable and corresponds to the concepts of the model. The suggested patterns are generic enough for being used in every Requirements Engineering method.

The JSON format is an example to represent their ability to link the expressed knowledge in a more structured and closer to programmer format in order to bridge the gap between the design the and the implementation phase. One of the most common issues in the RE world is that developers find hard to implement the design outcomes, especially if these are related to system's non-functional requirements. JSON format assists the developers in the realisation of the identified privacy concerns (requirements) and a way that they can be implemented using a structured low level expression language and not generic/abstract software engineering diagrams. Of course, JSON can be replaced by other structured XML-like formats. However, its wide use and dynamic nature of the template inspired us to express our work using this format. The *value* is mutable and corresponds to the values assigned based on the analysis of the respective system. JSON template assist on the direction of simplifying the process for the developers of implementing what had been suggested from the design phase, by expressing the conceptual business process in this format. This format is preferable since it can raise developers' awareness in understanding the outcomes of the aforementioned reasoning. Based on the proposed framework, every identified privacy requirement is expressed in a structured textual format using the JSON format. Through this JSON template, a more formalised expression of the whole set of concepts is achieved, and developers can understand the privacy requirements that need to be satisfied and the processes that need to be altered for addressing the privacy properties.

The PriS JSON template, presented in Fig. 2a, is in accordance with the four activities of PriS presented previously. The object **Privacy requirement** consists of the *Title*, the *Privacy Goal* that it wishes to achieve, the specific *Organisational Goal* that it relates to, the *Process*, and the *Privacy Enhancing Technologies*. The organisational goal consists of its *Title*, its *Parent Goal*, its *Child Goal*, and its *Decomposition Type*. The attribute **Process** indicates which process is affected. It contains the *Title*, the *Parent Process*, the *Child Process* and finally, the *Process Pattern* that needs to be satisfied. From the field of process pattern, we realise which privacy pattern we will implement. Finally, the attribute **Privacy Enhancing Technologies** assists developers on the selection of the set of most appropriate existing privacy enhancing technologies related to the specific privacy properties.

Fig. 2b depicts the Privacy Process Pattern template expressed with JSON format, enhancing PriS methodology. The template follows the same structure as it was described in Section 3, containing the fields *name*, *context*, *problems*, *forces*, *benefits*, *liabilities*, *implementation techniques*, and finally, *related patterns*. This final field contains all the available techniques that *can* satisfy the examined privacy requirement. The difference among this field and the one of *Privacy Enhancing Technologies*, which is included at the general template, is that the first one contains all the *potential solutions*, where the latter picks only the ones that satisfy the specific organisational goal. The template of the privacy process pattern will be included in the general template of the PriS JSON template to enhance the *Process Pattern* attribute.

Fig. 2: PriS and Privacy Process Pattern JSON template

5 Illustration of the Privacy Process Patterns

A real case study, in which PriS methodology has already been implemented to, and can be used to examine the applicability of the proposed Privacy Process Patterns' template, is the one of Aegean Career Unit. Specifically, University of the Aegean has built a software system for its Aegean Career Office. A detailed description of the Career Office System can be found in [33]. The scope of this case study was the identification of all respective concepts based on the PriS framework for conducting privacy-aware analysis based on the system's context and the stakeholders' requirements. The main objective of the Career Office system of the University of the Aegean is boundary management, i.e. helping students to manage the choices and transitions they need to make on exit from their studies in order to proceed effectively to the next step of their life. The Career Office system is described by three main principles that form the three primary organisational goals, namely: a) Provide Career Information, b) Offer Guidance through Events and c) Maintain a lifelong communication with the graduates. In Fig. 3, the goal model of the examined case study is depicted. We analyse only the principle 'Maintain a lifelong communication with the graduates' for simplicity reasons.

In accordance with the first step of PriS, the main privacy requirement identified along with stakeholders, was the following: "Graduates' anonymity should be enforced when collecting the completed questionnaires". For protecting graduates' privacy, it is of major importance to ensure that all types of analysis and produced results don't lead to any form of privacy violation directly or indirectly. Based on the organisation's context, graduates must be ensured that nobody, es-

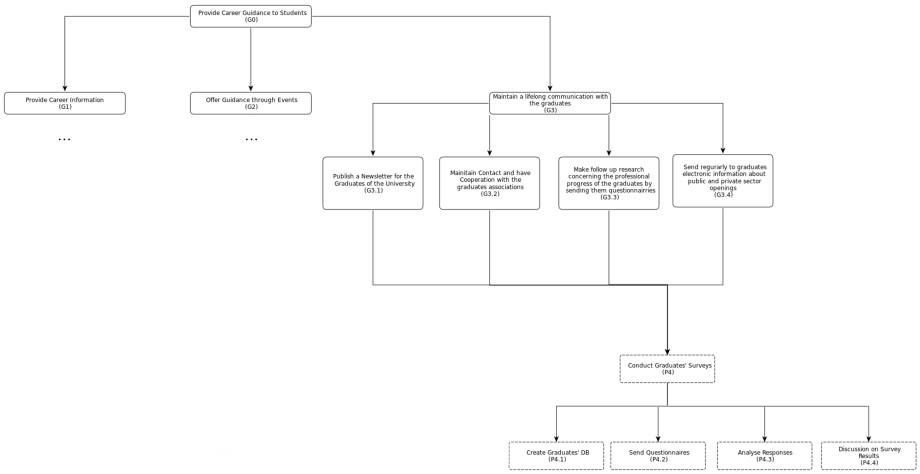

Fig. 3: Goal model

pecially malicious third parties, will be able to reveal the name or other elements that may lead to the identification of the graduate that submits the answered questionnaire; when graduates send information through the career office portal, it must be ensured that others will not be able to reveal any personal identifiable information. Following the identified requirement, the privacy goal that needs to be addressed and fulfilled is the anonymity goal.

Proceeding to the second step of PriS methodology, we need to identify the impact of this goal in the Career Office system, and thus, the identification of the organisational goals and subgoals that deal with the specific requirement is vital. For satisfying the anonymity goal, the main goal, subgoal and process affected are the following:

- Main Goal: Maintain a lifelong communication with the graduates (G3)
- Subgoal: Make follow up research concerning the professional progress of the graduates by sending them questionnaires (G 3.3)
- Main Process: Conduct Graduates Surveys (P4)
- Subprocess: Collect Responses (P 4.3)

The third step of PriS indicates the modelling of the affected processes, using privacy process patterns. For realising the identified privacy goals, the respective processes that implement the privacy-related subgoals were identified. Thus, for the anonymity goal, the respective process that identifies the operationalised subgoal G3.3 is P4 and specifically, the 'P4.3 Collect Responses'. For assisting the realisation of privacy goals on processes, privacy process patterns are introduced. Specifically, for every privacy goal, a respective privacy process pattern may be introduced on to the privacy-aware processes leading to the realisation of the privacy requirements by the respective PET in a more mature and concrete way.

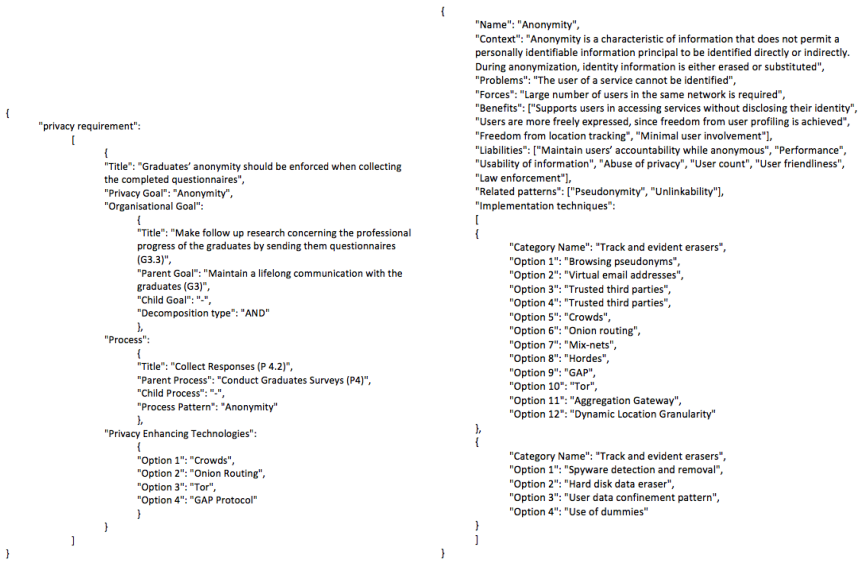

Fig. 4: Instantiation with JSON

By applying the relevant privacy process pattern on the respective privacy-related process, it is easier for the designer to identify the appropriate PETs, leading to the successful satisfaction of the respective goals. In the anonymity pattern, the user initiates a request for using a service to the system. The system checks the request and proceeds with the decision of preserving user's anonymity (in case the type of service requested should satisfy this privacy goal) or executes the identification task which leads the user to the process of providing their real credentials for granting access to use the requested service. Finally, according to PriS, the final step is the identification of the technique(s) that best support/implement the aforementioned procedures, the designer along with the stakeholders and the organisation's developer team decide the most appropriate PET for realising the identified privacy goals. The definition of selection criteria for the most adequate PET is out of the scope of this paper. In the given scenario, from the different options presented in Fig. 4b, our analysis has identified and suggested to the stakeholders the following PETs, presented in Fig. 4a: *Crowds*, *Onion Routing*, *Tor* and *GAP Protocol*.

6 Conclusions

This paper presents a set of privacy process patterns that can be used to bridge the gap between privacy design and implementation, and their instantiation in JSON. These patterns are illustrated using the Career Office system of the

University of the Aegean. Although, due to lack of space we have focused on the definition of five patterns, more patterns can be defined using the same template.

Future work includes the development of a privacy pattern language that will further assist developers in building the gap between design and implementation phase. In addition, we are planning to extend our work to elicit and define privacy patterns in new domains, such as Internet of Things and Cloud Computing.

Acknowledgments. This research was partially supported by the Visual Privacy Management in User Centric Open Environments (VisiOn) project, supported by the EU Horizon 2020 programme, Grant agreement No. 653642.

References

1. Duncan, G.T., Pearson, R.W.: Enhancing access to microdata while protecting confidentiality: Prospects for the future. Statistical Science, vol. 6(3), pp. 219–232. Institute of Mathematical Statistic (1991)
2. Hes R., Borking J.: Privacy Enhancing Technologies: the path to anonymity. ISBN. vol. 90(74087). p12. (1998)
3. Kalloniatis, C., Kavakli, E., Gritzalis, S.: Addressing privacy requirements in system design: the PriS method. In: Requirements Engineering, vol. 13(3) pp. 241–255. Springer (2008)
4. Romanosky, S., Acquisti, A., Hong, J., Cranor, L.F., Friedman, B.: Privacy patterns for online interactions. In: Proceedings of the 2006 conference on Pattern languages of programs, p. 12. ACM (2006)
5. Chung, E. S., Hong, J. I., Lin, J., Prabaker, M. K., Landay, J. A., Liu, A. L.: Development and evaluation of emerging design patterns for ubiquitous computing. In: Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques, pp. 233-242. ACM (2004)
6. Hafiz, M.: A pattern language for developing privacy enhancing technologies. In: Software: Practice and Experience, vol. 43(7), pp. 769–787. Wiley Online Library (2013)
7. Schümmer, T.: The public privacy–patterns for filtering personal information in collaborative systems. In: Proceedings of the Conference on Human Factors in Computing Systems (CHI) (2004)
8. Schumacher, M.: Security Patterns and Security Standards. In: EuroPLoP, pp. 289–300 (2002)
9. Alexander, C.: A pattern language: towns, buildings, construction. Oxford University Press (1977)
10. Mouratidis, H., Weiss, M., Giorgini, P.: Security patterns meet agent oriented software engineering: a complementary solution for developing secure information systems. In: International Conference on Conceptual Modeling, pp. 225–240. Springer (2005)
11. Fischer-Hübner, S.: IT-security and privacy: design and use of privacy-enhancing security mechanisms. Springer-Verlag (2001)
12. Cannon, J. C.: Privacy: What developers and IT professionals should know. Addison-Wesley Professional (2004)
13. Pfitzmann, A., Hansen, M.: A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management (2010)

14. ISO/IEC 29100:2011(E): Information technology - Security techniques - Privacy framework (2011)
15. Gabber, E., Gibbons, P. B., Matias, Y., Mayer, A.: How to make personalized web browsing simple, secure, and anonymous. In: International Conference on Financial Cryptography, pp. 17–31. Springer Berlin Heidelberg (1997)
16. Reiter, M. K., Rubin, A. D.: Crowds: Anonymity for web transactions. In: ACM transactions on information and system security (TISSEC), vol. 1(1) pp. 66–92. ACM (1998)
17. Goldschlag, D., Reed, M., Syverson, P.: Onion routing. In: Communications of the ACM, vol. 42(2), pp. 39–41, ACM (1999) Chicago
18. Chaum, D.: The dining cryptographers problem: Unconditional sender and recipient untraceability. In: Journal of cryptology, vol. 1(1), pp. 65–75. Springer (1988)
19. Chaum, D. L.: Untraceable electronic mail, return addresses, and digital pseudonyms. In: Communications of the ACM, vol. 24(2), pp. 84–90. ACM (1981)
20. Shields, C., Levine, B. N.: A protocol for anonymous communication over the Internet. In: Proceedings of the 7th ACM conference on Computer and communications security, pp. 33–42. ACM (2000)
21. Bennett, K., Grothoff, C.: GAP–practical anonymous networking. In: International Workshop on Privacy Enhancing Technologies, pp. 141–160. Springer Berlin Heidelberg (2003)
22. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. DTIC Document (2004)
23. Akers, T., Ware, B., Zheng, W., Kostet, M., Clark, B.: Service Aggregation Gateway. U.S. Patent Application No. 11/551,066. (2006)
24. Jain, A., Flynn, P., Ross, A. A. (eds.).: Handbook of biometrics. Springer Science & Business Media. (2007)
25. Weis, S. A., Sarma, S. E., Rivest, R. L., Engels, D. W.: Security and privacy aspects of low-cost radio frequency identification systems. In: Security in pervasive computing, pp. 201–212. Springer Berlin Heidelberg (2004)
26. Mulvenna, M.D., Anand, S.S, Büchner, A.G.: Personalization on the Net using Web mining: introduction. In: Communications of the ACM, vol. 43(8), pp. 122–125, ACM (2000)
27. Himmel, M.A., Rodriguez, H.: Method and apparatus for selective caching and cleaning of history pages for web browsers. U.S. Patent No. 6,453,342. (2002)
28. Bacard, A.: Computer Privacy Handbook: A Practical Guide to E-Mail Encryption, Data Protection, and PGP Privacy Software. Peachpit press (1995)
29. Wells, J.R., Felt, E.P.: System and method for message encryption and signing in a transaction processing system. US Patent No. 7,363,495 (2008)
30. European Parliament: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (2016)
31. Crockford, D.: The application/json Media Type for JavaScript Object Notation (JSON). `https://www.ietf.org/rfc/rfc4627.txt` (2006)
32. Mavropoulos, O., Mouratidis, H., Fish, A., Panaousis, E., Kalloniatis, C.: Apparatus: Reasoning About Security Requirements in the Internet of Things. In: International Conference on Advanced Information Systems Engineering, pp. 219–230, Springer (2016)
33. ICTE-PAN: Methodologies and Tools for Building Intelligent Collaboration and Transaction Environments in Public Administration Networks. In: Project Deliverable D 3.1b. University of the Aegean (2005)