Patterns- and Security-Requirements-Engineering-based Support for Development and Documentation of Security Standard Compliant ICT Systems *

Kristian Beckers and Maritta Heisel (PhD Supervisor)

paluno - The Ruhr Institute for Software Technology University of Duisburg-Essen
{firstname.lastname}@paluno.uni-due.de

Abstract. Aligning an ICT system with a security standard is a challenging task, because of the sparse support for development and documentation that these standards provide.
We create patterns for the elements of trustworthiness: security, risk management, privacy, and law. The instantiations of these patterns are used to support the development and documentation of ICT systems according to security standards. In addition, we define relations between security standards and security requirements engineering approaches.

Key words: security standards, requirements engineering, security, patterns

1 Motivation and Background

Security is a system property of ICT systems [1, 2] and an acceptable security level has to be achieved for the entire system. Security standards exist that provide relevant methods for achieving this goal. However, aligning ICT systems with security standards is difficult, because the standards provide only sparse support for system development and documentation. For example, assembling an *information security management system (ISMS)* according to the ISO 27001 requires a *scope and boundaries* description among its initial steps. The required input is to consider "characteristics of the business, the organization, its location, assets and technology" [3, p. 4].

Security requirements engineering (SRE) methods, on the other hand, provide structured elicitation and analysis of security requirements. This structured elicitation and analysis of security requirements of SRE methods is useful for numerous security engineering contexts. Therefore, we propose to use SRE methods to support security engineers in the development and documentation of trustworthy ICT systems that are compliant to security standards.

This thesis is inspired by the work of Gamma et. al [4], which manages comprehensible to describe design problems and solutions in a fairly easy way. We

* This research was partially supported by the EU project Network of Excellence on Engineering Secure Future Internet Software Services and Systems (NESSoS, ICT-2009.1.4 Trustworthy ICT, Grant No. 256980).

aim to accomplish the same for design and documentation problems of trustworthy ICT systems. Security engineering "requires cross-disciplinary expertise" [5, p. 3]. Patterns provide the means to collect this expertise and instantiate it to a given security engineering problem. We define trustworthiness as a combination of security, risk management, privacy and compliance attributes. All of these attributes are also required by security standards, e.g., ISO 27001. Hence, we restrict patterns in this work to security, law, privacy, and risk management patterns.

The outcome of this analysis answers the research question, if and to what extent patterns and SRE approaches can support the development of a security standard compliant ICT system. Moreover, it answers the question in what way patterns and SRE methods provide the required documentation for a security standard compliant ICT system and how existing pattern-based and SRE documentation can be re-used for an aforementioned system.

2 Previous Work

ICT systems keep increasing their functionality and distribution in recent years. Unfortunately this increase in complexity of ICT systems leads also to an increase in security problems for instance in cloud computing systems (or short clouds) [6].

We developed a pattern-based approach to support the context establishment and asset identification in the scope of cloud computing systems for the ISO 27005 [7] standard [8]. Our work shows a cloud system analysis pattern and different kinds of stakeholder templates serve to understand and describe a given cloud development problem. We illustrated our support using an online banking cloud scenario, presented in in Fig. 1. Our *cloud system analysis pattern* in Fig. 1 that provides a conceptual view on cloud computing systems and serves to systematically analyse stakeholders and requirements. The notation used to specify the pattern is based on UML[1] notation, i.e. the stick figures represent roles, the boxes represent concepts orientates of the real world, the named lines represent relations (associations) equipped with cardinalities, the unfilled diamond represents a "part-of" relation, and the unfilled triangles represent inheritance.

A *Cloud* is embedded into an environment consisting of two parts, namely the *Direct System Environment* and the *Indirect System Environment*. The *Direct System Environment* contains stakeholders and other systems that directly interact with the *Cloud*, i.e. they are connected by associations. Moreover, associations between stakeholders in the *Direct* and *Indirect System Environment* exist, but not between stakeholders in the *Indirect System Environment* and the cloud. Typically, the *Indirect System Environment* is a significant source for compliance and privacy requirements.

The *Cloud Provider* owns a *Pool* consisting of *Resources*, which are divided into *Hardware* and *Software* resources. The provider offers its resources as *Services*, i.e. *IaaS*, *PaaS*, or *SaaS*. The boxes *Pool* and *Service* in Fig. 1 are hatched,

[1] Unified Modeling Language: `http://www.omg.org/spec/UML/2.3/`

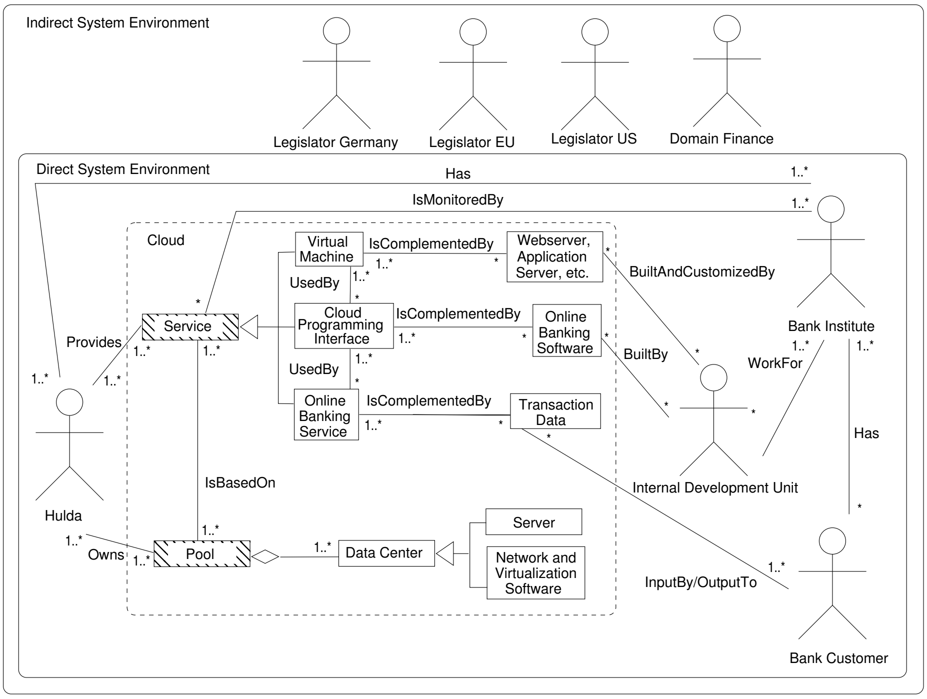

Fig. 1: Cloud System Analysis Pattern

because it is not necessary to instantiate them. Instead, the specialised cloud services such as *IaaS*, *PaaS*, and *SaaS* and specialised *Resources* are instantiated. The *Cloud Developer* represents a software developer assigned by the *Cloud Customer*. The developer prepares and maintains an *IaaS* or *PaaS* offer. The *IaaS* offer is a virtualised hardware, in some cases equipped with a basic operating system. The *Cloud Developer* deploys a set of software named *Cloud Software Stack* (e.g. web servers, applications, databases) into the *IaaS* in order to offer the functionality required to build a *PaaS*. In our pattern *PaaS* consists of an *IaaS*, a *Cloud Software Stack* and a *cloud programming interface (CPI)*, which we subsume as *Software Product*. The *Cloud Customer* hires a *Cloud Developer* to prepare and create *SaaS* offers based on the CPI, finally used by the *End Customers*. *SaaS* processes and stores *Data* in- and output from the *End Customers*. The *Cloud Provider*, *Cloud Customer*, *Cloud Developer*, and *End Customer* are part of the *Direct System Environment*. Hence, we categorise them as *direct stakeholders*. The *Legislator* and the *Domain* (and possibly other stakeholders) are part of the *Indirect System Environment*. Therefore, we categorize them as *indirect stakeholders*.

The cloud system analysis pattern instance in Fig. 1 helps, e.g., identifying assets by considering the instantiated boxes and the associations between the direct stakeholders and the cloud. The associations indicate the flow of information into and out of the cloud and therefore helps to analyze the information assets processed and stored in the cloud. Furthermore, the associations help to find out about the asset owner, as the standard requires.

Identifying relevant compliance regulations for a software system and aligning it to be compliant is a challenging task. Hence, we already developed a pattern-based method for Identifying and analyzing laws [9]. The method makes use of different kinds of patterns, which help to systematically elicit relevant laws.

We also analyzed the ISO 27001 standard to determine what techniques and documentation are necessary and instrumental to develop and document systems according to this standard [10]. Based on these insights, we inspected a number of current SRE approaches to evaluate whether and to what extent these approaches support ISO 27001 system development and documentation. We re-use a conceptual framework (CF) [11] originally developed for comparing SRE methods to relate important terms, techniques, and documentation artifacts of the security requirements engineering methods to the ISO 27001.

3 Future Work

In the future we will extend this approach to support the documentation and development of trustworthy ICT systems, as depicted in Fig. 2. In our approach, we will re-use existing meta models for security standards, e.g., Sunyaev [12] and for risk management standards, e.g., Fenz [13] and combine them into a pattern for security and risk management standards (1). As a next step we will develop relations from these patterns to the CF (2), which allows us to re-use the existing relations to SRE methods (3). We combine the relations 1, 2, and 3 and, thus, we can create transitive relations the SRE methods to multiple security and risk management standards, e.g. ISO 27001 and Common Criteria (4).

However, the privacy and compliance demands of trustworthy ICT systems and security standards, e.g., ISO 27001 and Common Criteria, alike are not yet addressed. Hence, we propose to develop relations between specific patterns for laws (5), risk and security (6), and privacy (7). We will also extend the CF to enable relations to privacy and law extensions of SRE methods. The risk and security patterns shall address issues that are not already covered by an existing SRE method in 3. We will also develop the patterns in 5, 6, and 7, if there are no suitable patterns available yet. As a last step we combine the relations 5, 6, and 7 and, thus, also relate the patterns to multiple security standards, e.g. ISO 27001 and Common Criteria (8).

We choose cloud computing as an example of our work. Hence, we will create more detailed patterns for cloud systems based upon the aforementioned Cloud System Analysis Pattern.

Moreover, aligning clouds to meet compliance regulations is a challenging task, because of a high number of different kinds of stakeholders. We will address this problem by creating specific cloud law analysis patterns as an extension to our existing law pattern approach [9]. Our extension will also make use of results generated by the application of the cloud system analysis pattern.

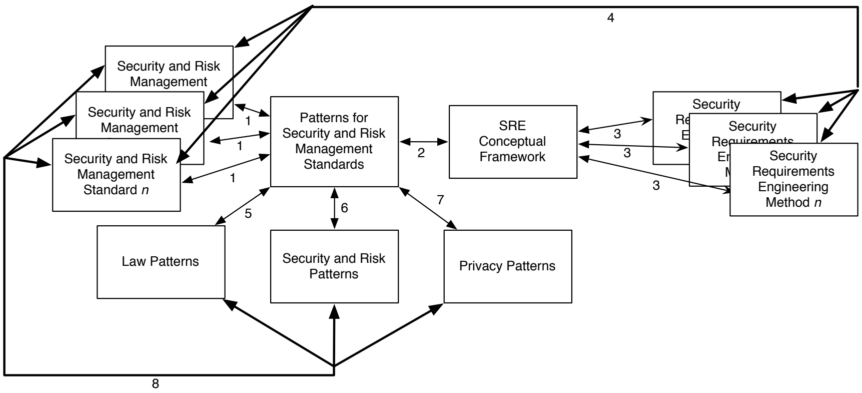
4
Security and Risk Management
Security and Risk Management
Security and Risk Management Standard *n*
1
1
1
Patterns for Security and Risk Management Standards
2
SRE Conceptual Framework
3
3
3
Security
Security Requirements
Security Requirements Engineering Method *n*
5
6
7
Law Patterns
Security and Risk Patterns
Privacy Patterns
8

Fig. 2: Support for Developing and Documenting Trustworthy ICT Systems

We will start working on privacy patterns based upon Nissenbaum's model of informational privacy in terms of contextual privacy [14]. The model considers the context of a given situation, the kind of information and the relation of the information to the context. We will also compare security and risk management patterns using existing surveys, e.g., Heyman et al. [15].

The outcome of our work is a methodology for developing and documenting ICT systems with the goal to be compliant to security standards. We aim at developing a system of patterns supported by security requirements engineering approaches, which can be used to improve the security of an ICT system, as well as to generate a documentation of an ICT system. This documentation can be used as a basis for certification according to a standard.

The patterns in our work will be based upon UML and the problem frame approach by Michael Jackson [16]. In addition, essential parts of the patterns are specified with a formal notation based upon the Z notation [17]. The patterns will be derived from relevant scientific literature, existing pattern libraries, as well as being found in existing implementations of security standards.

We plan to validate our work via using the methodology and the pattern system for an ICT system and a specific security standard. We will compare the resulting documentation against a standard-compliant documentation that is not based on our patterns.

We conclude with a brief summary of the main benefits of our approach:

- A methodology for systematic pattern-based development and documentation of ICT systems
- Complementing patterns with existing SRE approaches in order to completely support the implementation of sections of security standards
- Specific-patterns for laws, privacy, security and risk management to cover all quality requirements of security standards
- Ease the burden of implementing security standards

References

1. Pfleeger, C.P., Pfleeger, S.L.: Security In Computing. 4th edn. Prentice Hall PTR (2007)
2. Anderson, R.: Security EngineerIng. 2nd edn. Wiley (2008)
3. ISO/IEC: Information technology - Security techniques - Information security management systems - Requirements. ISO/IEC 27001, International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) (2005)
4. Gamma, E., Helm, R., Johnson, R., Vlissides, J.M.: Design Patterns: Elements of Reusable Object-Oriented Software. 1 edn. Addison-Wesley Professional (1994)
5. Bishop, M.: Computer Security : art and science. 1st edn. Pearson (2003)
6. Beckers, K., Jürjens, J.: Security and compliance in clouds. In: Information Security Solutions Europe (ISSE 2010). Securing electronic business processes : Highlights of the Information Security Solutions Europe, Vieweg + Teubner (2010) 91–100
7. ISO/IEC: Information technology - security techniques - information security risk management. ISO/IEC 27005, International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) (2008)
8. Beckers, K., Küster, J.C., Faßbender, S., Schmidt, H.: Pattern-based support for context establishment and asset identification of the ISO 27000 in the field of cloud computing. In: Proceedings of the International Conference on Availability, Reliability and Security (ARES), IEEE Computer Society (2011) 327–333
9. Beckers, K., Küster, J.C., Faßbender, S., Schmidt, H.: A pattern-based method for identifying and analysing laws. In: REFSQ. (2012) to be published.
10. Beckers, K., Faßbender, S., Heisel, M., Küster, J.C., Schmidt, H.: Supporting the development and documentation of ISO 27001 information security management systems through security requirements engineering approaches. In: Proceedings of the International Symposium on Engineering Secure Software and Systems (ESSoS). LNCS, Springer (2012) to be published.
11. Fabian, B., Gürses, S., Heisel, M., Santen, T., Schmidt, H.: A comparison of security requirements engineering methods. Requirements Engineering – Special Issue on Security Requirements Engineering **15**(1) (2010) 7–40
12. Sunyaev, A.: Health-Care Telematics in Germany: Design and Application of a Security Analysis Method. 1st edn. Gabler Verlag (2011)
13. Fenz, S., Ekelhart, A., Neubauer, T.: Information security risk management: In which security solutions is it worth investing? Communications of the Association for Information Systems **28**(1) (5 2011) 329–356
14. Nissenbaum, H.: Privacy in Context: Technology, Policy, and the Integrity of Social Life. 1st edn. Stanford (2009)
15. Heyman, T., Scandariato, R., Huygens, C., Joosen, W.: Using security patterns to combine security metrics. In: Proceedings of the International Conference on Availability, Reliability and Security (AReS), IEEE Computer Society (2008) 1156–1163
16. Jackson, M.: Problem Frames. Analyzing and structuring software development problems. Addison-Wesley (2001)
17. ISO/IEC: Information technology – Z formal specification notation – Syntax, type system and semantics. ISO/IEC 13568, International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) (2002)