Requirements Eng (2013) 18:343–395
DOI 10.1007/s00766-013-0174-7

A pattern-based method for establishing a cloud-specific information security management system

Establishing information security management systems for clouds considering security, privacy, and legal compliance

Kristian Beckers · Isabelle Côté · Stephan Faßbender · Maritta Heisel · Stefan Hofbauer

Received: 5 January 2013 / Accepted: 11 May 2013 / Published online: 4 June 2013

Abstract Assembling an information security management system (ISMS) according to the ISO 27001 standard is difficult, because the standard provides only very sparse support for system development and documentation. Assembling an ISMS consists of several difficult tasks, e.g., asset identification, threat and risk analysis and security reasoning. Moreover, the standard demands consideration of laws and regulations, as well as privacy concerns. These demands present multi-disciplinary challenges for security engineers. Cloud computing provides scalable IT resources and the challenges of establishing an ISMS increases, because of the significant number of stakeholders and technologies involved and the distribution of clouds among many countries. We analyzed the ISO 27001 demands for these multi-disciplinary challenges and cloud computing systems. Based on these insights, we provide a method that relies upon existing requirements engineering methods and patterns for several security tasks, e.g., context descriptions, threat analysis and policy definition. These can ease the effort of establishing an ISMS and can produce the necessary documentation for an ISO 27001 compliant ISMS. We illustrate our approach using the example of an online bank.

Keywords ISO 27001 · Security policies · Control selection · Threat analysis · Information security management system · Asset identification privacy · Legal compliance · Cloud computing security

1 Introduction

The possibility of quickly acquiring or disposing of resources such as storage and memory provides a great attraction to a variety of customers. Cloud computing systems (or simply clouds) provide the means for this kind of acquisition or deposition. However, potential customers are still reserved when it comes to using cloud resources. In 2009, a study was conducted by the International Data Corporation[1] about this issue. It pointed out that security is a significant barrier for the acceptance of clouds in companies. The lack of trust in cloud security lies within the nature of clouds: storing and managing critical data and executing sensitive IT processes are performed beyond the company's/customer's control. To gain the customer's trust and to illustrate that security is taken seriously, cloud providers have to certify their services with respect to security. One way of doing that is to turn to standards that put security at the center of interest. Examples for such standards are the ISO 27000 standards family and the common criteria [1]

K. Beckers (✉) · S. Faßbender · M. Heisel
Paluno, The Ruhr Institute for Software Technology, University of Duisburg-Essen, Oststrasse 99, 47057 Duisburg, Germany
e-mail: kristian.beckers@paluno.uni-due.de

S. Faßbender
e-mail: stephan.fassbender@paluno.uni-due.de

M. Heisel
e-mail: maritta.heisel@paluno.uni-due.de

I. Côté
ITESYS Institute for Technical Systems GmbH,
Emil-Figge-Str. 76, 44227 Dortmund, Germany
e-mail: Isabelle.Cote@itesys.de

S. Hofbauer
Network Integration Services Department,
Amadeus Data Processing GmbH,
Berghamer Straße 6, 85435 Erding, Germany
e-mail: stefan.hofbauer@amadeus.com

[1] http://www-304.ibm.com/isv/library/pdfs/cloud_idc.pdf.

(CC). The CC is a document-driven standard. It is necessary to specify a *target of evaluation* (TOE), which can be a security system or a security product. The TOE must be described completely. Whenever a part of the TOE changes, it is necessary to recertify the system or product. These two points constitute a problem when dealing with clouds. A cloud consists of a significant amount of hard and software parts. Describing the TOE may therefore be a challenging task. Furthermore, due to the fact that resources of clouds can be dynamically scaled, the TOE changes with every scaling. Thus, a recertification would be triggered each time the customer initiates a change in the resource usage. We therefore conclude that performing a CC certification for a complete cloud system is rather not practical. We are currently not aware of any company that completed a CC evaluation for a whole cloud computing system. However, it can be used to certify specific parts within a cloud. For example, the hypervisor is a prominent candidate for a CC evaluation. The ISO 27001 standard—in contrast to the CC—is process driven. This applies well to the service concept of a cloud. Several well-known companies have adopted this approach such as Microsoft,[2,3] Amazon,[4] Google,[5,6] and Salesforce.[7] The aim of the ISO 27001 standard is to establish an information security management system (ISMS). To use this standard for cloud computing systems is in accordance with the German Federal Office for Information Security (BSI).[8] The current version of the standard does not take cloud-specific security issues into consideration. The BSI recommends to consider cloud-specific threats when dealing with cloud systems. The Cloud Security Alliance (CSA) [2] and Gartner [3] have identified several of these threats. We take their findings and use them in our work. Assembling an ISMS according to the ISO 27001 standard is a non-trivial task. This is supported by the fact that descriptions for system development and documentation are rather sparse. For example, the required input for the *scope and boundaries* description is to consider "characteristics of the business, the organization, its location, assets and technology" [4, p. 4]. No further information beyond that is given.

We present our *PAttern-based method for establishing a Cloud-specific information Security management system* (*PACTS*). We analyzed the activities demanded by the standard to build an ISMS and present patterns for these incorporating existing security requirements approaches, where applicable. We also provide a structured method that shows how the different elements described above have to be applied in order to create the required ISMS documentation admissible for certification. We use existing research on context descriptions for clouds in our method in order to provide a domain-specific approach. The patterns define stakeholders and technological artifacts that are used in the context description and all subsequent patterns and models, e.g., security policies. Furthermore, we provide relations from these patterns to cloud-specific lists of threats proposed by the CSA [2] and Gartner [3]. In addition, our approach provides a structured refinement of the cloud system's and stakeholder's information to assess the threats for a particular instantiation of our cloud pattern. Our method uses this information for risk assessment and security control selection according to the ISO 27001 standard. Moreover, the ISO 27001 standard demands consideration of privacy and legal compliance. We integrated existing pattern-based research for compliance and privacy requirements into our approach in order to satisfy these demands.

The main contributions of our PACTS method are:

- A structured method to build an ISMS considering security, compliance, and privacy
- Detailed sub-methods for each step of PACTS
- Patterns- and templates to support the documentation of management commitment, scope descriptions, asset documentation, and defining security policies
- Re-use of these patterns- and templates for different projects via instantiation
- Integration of our patterns and templates into existing methods for the identification of relevant laws, eliciting and verifying privacy requirements, and risk management
- Supporting the ISO 27001 documentation demands

Running example We illustrate our approach by the example of a bank providing an online banking service to their customers. The bank uses a cloud for providing the service. We consider transaction services for two particular kinds of customers, namely a bank customer and a VIP bank customer that have specific service level agreements with the bank regarding the availability of the service. These customers' data such as account number, balance, and transaction log history are stored in the cloud. The bank authorizes its software department to design and build the cloud-specific software according to the interface and platform specification of the envisioned cloud provider.

[2] http://blogs.msdn.com/b/windowsazure/archive/2011/12/19/windows-azure-achieves-is0-27001-certification-from-the-british-standards-institute.aspx.

[3] http://www.windowsazure.com/en-us/support/trust-center/compliance/.

[4] http://aws.amazon.com/security/.

[5] http://googleenterprise.blogspot.com.br/2012/05/google-apps-receives-iso-27001.html.

[6] http://www.computerweekly.com/news/2240150882/Google-Apps-for-Business-wins-ISO-27001-certification.

[7] http://www.salesforce.com/platform/cloud-infrastructure/security.jsp.

[8] http://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Minimum_information/SecurityRecommendationsCloudComputingProviders.pdf.

The cloud provider orders its data center administration to configure the cloud accordingly and the support team of the cloud provider helps the bank with problems.

The remainder of the work is organized as follows: Sect. 2 presents background on clouds and the ISO 27001 standard. We explain our method in Sect. 3. The method consists of steps for management commitment (see Sect. 4), ISMS scope definition (see Sect. 5), asset identification (see Sect. 6), threat analysis (see Sect. 7), risk management (see Sect. 8), control reasoning (see Sect. 9), and creating ISMS design specifications (see Sect. 10). We propose an addition to our method, which is also explained in Sect. 3, that adds tasks for legal compliance (see Sect. 11) and considering privacy (see Sect. 12). Section 13 presents related work and Sect. 14 concludes and gives directions for future research.

2 Background

We illustrate the general idea of cloud computing in Sect. 2.1, and our cloud system analysis pattern in Sect. 2.2. We introduce the ISO 27000 series of standards in Sect. 2.3 and the 27001 standard in particular in Sect. 2.4.

2.1 Cloud computing

The term *cloud computing* describes a technology as well as a business model [5]. According to the *National Institute of Standards and Technology* (*NIST*) cloud computing systems can be defined by the following properties [6]: the cloud customer can acquire resources of the cloud provider over *broad network access* and *on-demand* and pays only for the used capabilities. Resources, i.e., storage, processing, memory, network bandwidth, and virtual machines, are combined into a so-called *pool*. Thus, the resources can be virtually and dynamically assigned and reassigned to adjust the customers' variable load and to optimize the resource utilization for the provider.

The virtualization causes a location independence: the customers generally have no control or knowledge of the exact location of the provided resources. Another benefit is that the resources can be quickly scaled up and down for customers and appear to be unlimited, which is called *rapid elasticity*. The pay-per-use model includes guarantees such as availability or security for resources via customized *service level agreements* (*SLA*) [7].

The architecture of a cloud computing system consists of different service layers and allows different business models: on the layer closest to the physical resources, the *Infrastructure as a Service* (*IaaS*) provides pure resources, for example virtual machines, where customers can deploy arbitrary software including an operating system. Data storage interfaces provide the ability to access distributed databases on remote locations in the cloud. On the *Platform as a Service* (*PaaS*) layer, customers use an API to deploy their own applications using programing languages and tools supported by the provider. On the *Software as a Service* (*SaaS*) layer, customers use applications offered by the cloud provider that are running on the cloud infrastructure. Furthermore, cloud providers require a layer that monitors their customers' resource usage, e.g., for billing purposes and service assurances. Buyya et al. [8] introduce this layer as a middleware in their cloud model. Cloud computing offers different *deployment scenarios*: *private clouds* are operated solely for an organization, *public clouds* are made available to the general public or a large industry group and are owned by a third party selling cloud services. In between these scenarios are *hybrid clouds* where users complement internal IT resources upon demand with resources from an external vendor [5].

2.2 Cloud system analysis pattern

We propose patterns for a structured domain knowledge elicitation. Depending on the kind of domain knowledge that we have to elicit for a software engineering process, we always have certain elements that require consideration. For this work we use a specific *context elicitation pattern*, the so-called *cloud system analysis pattern* [9]. We base our approach on Jackson's work on Problem Frames [10] that considers requirements engineering from the point of view of a machine in its environment. The machine is the software to be build and requirements are the effect the machine is supposed to have on the environment. Any given environment considers certain elements, e.g., stakeholders or technical elements. Jackson [10], who describes Problem Frames as follows: "A problem frame is a kind of pattern. It defines an intuitively identifiable problem class in terms of its context and the characteristics of its domains, interfaces and requirement.". We were also inspired by Fowler [11], who developed patterns for the analysis phase of a given software engineering process. His patterns describe organizational structures and processes, e.g., accounting, planning, and trading.

Our patterns for the analysis phase differ from patterns concerning solutions for the design phase of software engineering like the Gang of Four patterns [12] or the security patterns by Schumacher et al. [13]. The reason is that we provide a means for a structured elicitation of domain knowledge for cloud computing systems. We do not provide solutions for the implementation phase of clouds.

We present a short introduction of our so-called *Cloud System Analysis Pattern (or short: Cloud Pattern)* [9] in

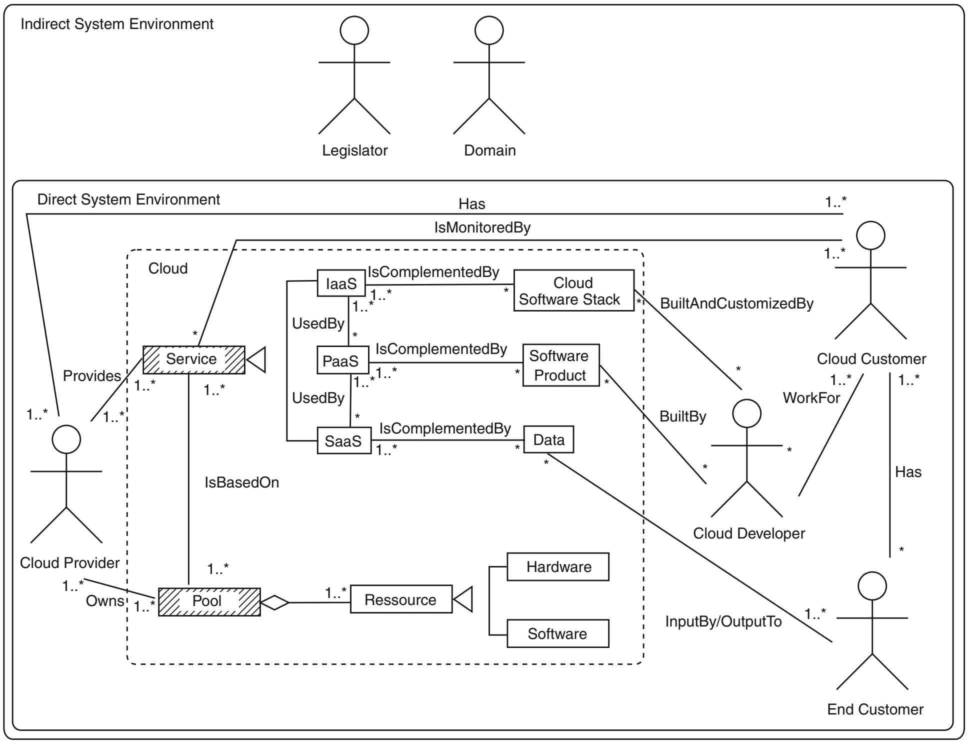


Fig. 1 Cloud system analysis pattern taken from [9]

the following. We created the pattern for cloud-specific context establishment and asset identification compliant to the ISO 27000 series of standards. A *Cloud* (see Fig. 1) 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 to the cloud 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 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 cloud concepts and it is not necessary to instantiate them. Instead, the specialized cloud services such as *IaaS, PaaS*, and *SaaS* and specialized *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 virtualized hardware, in some cases it is 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* input 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 categorize 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*. We also provide templates for each stakeholder that describe their attributes in detail (see Sect. 5).

2.3 The ISO 27000 series of standards

The ISO 27000 series of standards addresses information security matters and the subsequent ISMS. This is a system independent of vendors, technologies or the size/type of organization that is part of the management system of an organization [14].

The central standard in the series is the ISO 27001 that defines the requirements for an ISMS. A certification of an implementation of the ISO 27001 process is possible. All the other standards of the series are specifications of this standard and describe parts or usage scenarios of the ISMS in detail [15].

The ISO 27000 standard [15] divides the standards of the ISO 27000 series of standards into four categories. The

ISO 27000 standard itself defines the terminology of the series, the ISO 27001 states the general requirements for an ISMS. General guidelines specify parts of the ISMS e.g., the ISO 27005 specifies risk management. Sector-specific guidelines describe how an ISMS is to be implemented in a specific kind of organization, e.g., ISO 27011 concerns telecommunication organizations.

The ISO 27007 describes the auditing and certification of the ISO 27001 standard, while the ISO 27006 lists the certification body requirements. Organizations can get accreditation for certifying ISO 27001 realizations.

The remaining standards of the series describe a specific topic in relation to the ISMS. For instance, ISO 27010 describes how to combine different ISMS within one company, ISO 27031 describes business continuity management. However, even though numerous standards in the ISO 27000 series exist, that specify parts of the ISO 27001, it is not mandatory to use these specifications. The standard also allows to use different specifications, as long as they fulfill the requirements of the ISO 27001 [16]. Hence, we focus in our work on the ISO 27001 standard.

The ISO considers also to publish a standard ISO 27017 to provide guidance for implementing an ISMS for clouds and the ISO 27018 to provide privacy guidelines for clouds. Both standards will be released in a draft status soon. However, neither of them will replace the ISO 27001 as the normative standard of the ISO 27000 series of standards.[9]

2.4 The ISO 27001 standard

The ISO 27001 defines the requirements for establishing and maintaining an ISMS [4]. In particular, the standard describes the process of creating a model of the entire business risks of a given organization and specific requirements for the implementation of security controls. The ISO 27001 standard is structured according to the “Plan-Do-Check-Act” (PDCA) model, the so-called *ISO 27001 process* [4]. In the *Plan* phase an ISMS is established, in the *Do* phase the ISMS is implemented and operated, in the *Check* phase the ISMS is monitored and reviewed, and in the *Act* phase the ISMS is maintained and improved. In the *Plan* phase, the *scope and boundaries* of the ISMS, its *interested parties, environment, assets*, and all the *technology* involved are defined. In this phase also the ISMS *policies, risk assessments, evaluations*, and *controls* are defined. Controls in the ISO 27001 are measures to *modify risk*. The ISO 27001 standard demands the creation of a set of documents and the certification of an ISO 27001 compliant ISMS is based upon these documents.

Changes in the organization or technology also have to comply with the documented ISMS requirements. Furthermore, the standard demands periodic audits toward the effectiveness of an ISMS. These audits are also conducted using documented ISMS requirements. In addition, the ISO 27001 standard demands that management decisions, providing support for establishing and maintaining an ISMS, are documented as well. This support has to be documented via management decisions. This has to be proven as part of a detailed documentation of how each decision was reached and how many resources are committed to implement this decision.

3 Overview of our PACTS method

Evaluating business benefits against privacy, security, and compliance concerns of clouds is difficult, because implementation and operational details are often not transparent to cloud customers or end customers. These stakeholders entrust their data to a cloud provider, which leads to concerns regarding data integrity, recovery and location, as well as legal issues [3].

We address these concerns by proposing our PACTS method for creating a cloud-specific ISMS compliant to the ISO 27001 standard with a particular focus on legal compliance and privacy. PACTS considers either the *cloud provider* or the *cloud customer* as possible stakeholders, who build an ISMS. The reason is that these are organizations that should earn the trust of their customers via certifying an ISMS.

Our *cloud system analysis pattern* (see Sect. 2.2) provides a basic structure of a cloud computing architecture, which considers the relations between stakeholders and the cloud. The pattern can be instantiated for any given cloud scenario and if required extended with little effort. The pattern provides a basis for cloud-specific asset identification, threat analysis, risk management, and control selection. For example, several threats are already mapped to the cloud pattern and can be analyzed based upon the patterns instantiation. The instantiated pattern is also the input for our identification of relevant laws and analysis of privacy requirements.

Our cloud pattern reduces the effort for creating a description of a cloud. We can simply instantiate the pattern in order to get a description. The benefit of basing our method on the cloud pattern is also that knowledge collected using the pattern can be re-used for different instantiations of the pattern. For example, assets identified using the pattern can be instantiated for different projects, e.g., the *Data* in the cloud pattern have been identified as an asset. Hence, all instantiations of *Data* are assets as well. In additions, experiences from using our method can

[9] http://www.iso27001security.com/html/27017.html.

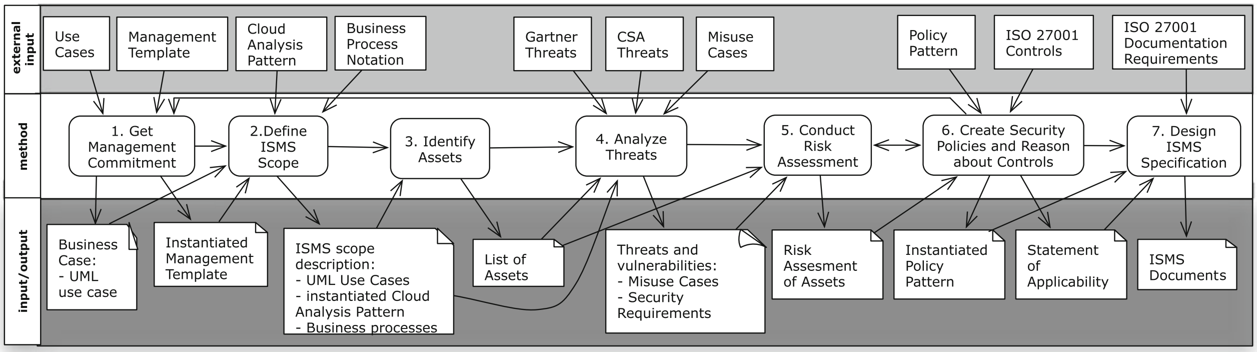

external input
Use Cases
Management Template
Cloud Analysis Pattern
Business Process Notation
Gartner Threats
CSA Threats
Misuse Cases
Policy Pattern
ISO 27001 Controls
ISO 27001 Documentation Requirements
method
1. Get Management Commitment
2.Define ISMS Scope
3. Identify Assets
4. Analyze Threats
5. Conduct Risk Assessment
6. Create Security Policies and Reason about Controls
7. Design ISMS Specification
input/output
Business Case: - UML use case
Instantiated Management Template
ISMS scope description: - UML Use Cases - instantiated Cloud Analysis Pattern - Business processes
List of Assets
Threats and vulnerabilities: - Misuse Cases - Security Requirements
Risk Assesment of Assets
Instantiated Policy Pattern
Statement of Applicability
ISMS Documents

Fig. 2 The steps of our PACTS Method concerning security

also lead to an improved pattern, e.g., the pattern can be extended with further stakeholders.

We present an overview of our method for establishing a cloud-specific ISMS in this section. In the remainder of the section we provide detailed descriptions of each step of our PACTS method. We begin by describing the steps concerning security, depicted in Fig. 2.

Step 1: Get management commitment The precondition for building an ISMS is that the management commits to it. Thus, we dedicate the first step of our method to get management commitment for the ISMS and the provision of adequate resources to establish it. We describe the characteristics of the business via UML use case diagrams [17]. The use cases are accompanied by our management templates, which have to be instantiate with relevant information for building the ISMS, e.g., high-level security goals, cloud-specific management concerns, and resource management.

Step 2: Define ISMS scope The scope for building the ISMS shall be described using the initial use cases. These are refined using our cloud system analysis pattern for structural description of the cloud scenario and a business process notation for behavioral description. In our examples, we choose UML activity diagrams [17] as business process notations.

Step 3: Identify assets The entire ISMS scope description is the input for the asset identification. We identify all items of value to the cloud stakeholders and by iterating over the relations from cloud stakeholders to cloud elements in the cloud system analysis pattern and activity diagrams. This results in a list of assets and the stakeholders that own them as an output of this step of the method.

Step 4: Analyze Threats We conduct a threat analysis using the list of threats released by the CSA [2], an industrial consortium that investigated practical security issues with clouds and the threats that Gartner [3] considers. We propose to identify threats to the previously identified assets using our cloud system analysis pattern. This activity includes an investigation of vulnerabilities of cloud components. We use the identified threats as an input for misuse cases. The results of the misuse cases are specific threats and security requirements.

Step 5: Conduct Risk Assessment The assets, threats, vulnerabilities, and security requirements serve as input for our risk assessment. We conduct an asset-based method that uses the previously elicited knowledge to derive likelihood and consequences scales, as well as acceptable risk levels. This information is used to determine, which cloud threats cause unacceptable risks.

Step 6: Create Security Policies and Reason about Controls Controls in the ISO 270001 standard reduce risks to assets. The reasoning about controls considers the risks to each assets and supports the decision if a control is needed or not. For each asset, we propose to compile a list that states why a control in the normative ANNEX A of the ISO 27001 should or should not be applied to that asset. We instantiate our policy pattern to ease this activity. The policy patterns help to define precisely which elements of the cloud pattern the control refers to and the security goal the control shall achieve. If the decision is made that a control has to be introduced, we go back to the previous step of our method in order to adjust the risk assessment for that particular asset. This information is in turn used to check if the control already results in an acceptable risk level or if it has to be modified or another control should be introduced. The resulting information is used to compile the so-called *Statement of Applicability*, which is a mandatory document for reasoning about the ISO 27001 controls.

We also have to check carefully if our use cases defined in Step 1 can lead to acceptable risk levels using reasonable and affordable controls. Hence, we also have to consider changing the use cases in case acceptable risk levels cannot be achieved with reasonable efforts.

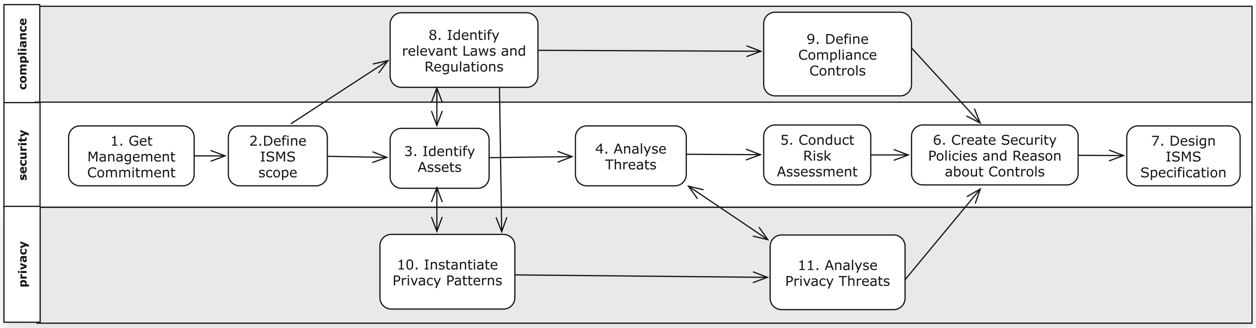

compliance
security
privacy
8. Identify relevant Laws and Regulations
9. Define Compliance Controls
1. Get Management Commitment
2.Define ISMS scope
3. Identify Assets
4. Analyse Threats
5. Conduct Risk Assessment
6. Create Security Policies and Reason about Controls
7. Design ISMS Specification
10. Instantiate Privacy Patterns
11. Analyse Privacy Threats

Fig. 3 The steps of PACTS concerning compliance and privacy

Step 7: Design ISMS specification The final step of our method concerns the ISO 27001 specification, an implementable description of the ISMS. We consider the ISO 27001 documentation demands and use the information elicited and documented in the previous steps of our method. This information is mapped to the required document types. These documents are also the basis for a certification of an ISO 27001 compliant ISMS. The ISO 27001 standard has several demands for quality requirements beyond security, namely compliance and privacy. We refer to Sects. 11 and 12 for definitions and detailed discussion of these terms. We provide support for eliciting and analyzing these requirements as part of our approach. We show how compliance and privacy concerns are addressed in Fig. 3. The consideration of compliance issues for cloud computing systems is also a key recommendation of Gartner's [3] analysis.

In our PACTS method compliance identifies relevant laws and regulation and defines corresponding requirements. The relation between compliance and privacy is that the compliance part identifies relevant laws that concern privacy. The privacy part of our method uses these laws as input.

Step 8. Identify relevant laws and regulations We use the information from the asset identification. Namely we identify relevant laws and regulations with this information. This activity also has to identify assets in terms of laws and regulations, which can be related to assets in terms of security. We discuss the differences in Sect. 11. Moreover, laws and regulations can regulate privacy concerns. This information is used during the *instantiate privacy patterns* step of our method.

Step 9. Define compliance controls Once laws and regulations are identified, they have to be translated into ISO 27001 compliance controls. This translation is difficult, because in some cases laws or regulations demand reasoning about a specific concern or they demand a specific functionality. We discuss this issue also in Sect. 11. In addition, compliance controls can have relations to other ISO 27001 controls. For example, a law could demand a specific control in a certain situation, while the risk assessment results would not.

The ISO 27001 standard demands also the consideration of privacy in the informative ANNEX B. We propose the following steps to address this concern.

Step 10. Instantiate privacy patterns We use textual privacy patterns based upon the ISMS scope definition and relevant laws and regulations. These patterns can be instantiated and they give rise to initial privacy requirements. In addition, the identified assets for security can be considered, because if these contain personal information they can also support instantiating further privacy patterns.

Step 11. Analyze privacy threats We use a privacy threat analysis based on the information flow between requirements. We analyze the flow of personal information based on the previously instantiated privacy patterns and functional requirements of the cloud scenario. We also refine the initial privacy requirements. The information flow among the requirements shows, which stakeholders have potentially access to which personal information. Afterward, software engineers have to check if the requirements have to be modified in order to be privacy preserving.

In the following, we begin each section with an instantiation of the template presented in Table 1. The rows of the template state the relations of the topic of the section to the standard, e.g., threat analysis, and the

Table 1 Template for relation to the ISO 27001 standard

Significance for the ISO 27001 standard	Define the general relation of the section to the ISO 27001 standard
Related section(s) of the standard	State the related sections of the standard

Table 2 Management commitment demands within the ISO 27001 standard

Significance for the ISO 27001 standard	The ISO 27001 standard demands documentation of management commitment for the establishment of an ISMS
Related section(s) of the standard	Section 5.1 management commitment concerns proof the management shall provide for establishing an ISMS objectives, plans, responsibilities and accepting risks. Section 5.2 resource management concerns the provision of resources for establishing the ISMS and the training of the members of the organization for security awareness and competence

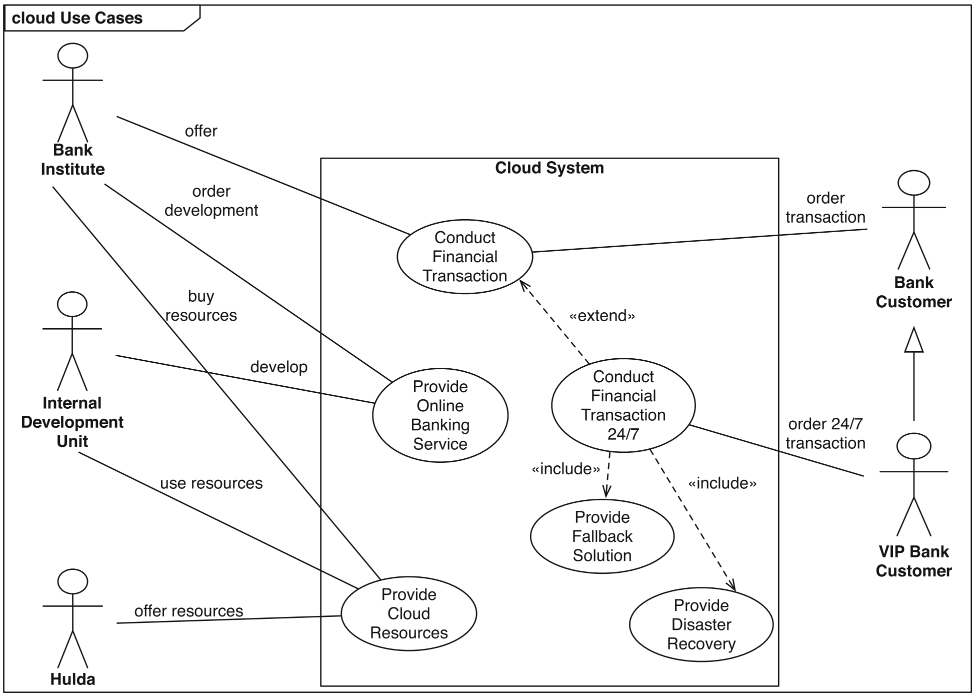


Fig. 4 Use Case for a cloud-based Online banking system

standard's relevant section for that topic. We assume that users of our method may look for support for establishing a specific ISO 27001 section instead of establishing an entire ISMS. Hence, the information in the last column can be used to identify the part of our method that supports a specific ISO 27001 section.

We use several artifacts from previous research as part of this work. We explicitly state the background and contributions in the beginning of these sections.

4 PACTS Step 1: get management commitment

The ISO 27001 standard dedicates its entire Sect. 5 to the importance of management commitment for implementing an ISMS, shown in Table 2. ISO 27001 Sect. 5 contains subsections for management comment proofs and provisioning of sufficient resources.

The management commitment for implementing an ISMS according to the ISO 27001 standard is of utmost importance, because without the commitment of sufficient personal and resources the ISMS implementation is doomed to fail. In addition, the publicly available examples of ISMS documentations, e.g., the so-called *ISMS toolkit*[10] defines this also as the first step of establishing an ISO 27001 compliant ISMS.

The management commitment is based upon business cases concerning a cloud scenario. We defined a set of UML use cases [17] to illustrate business cases for our running example, depicted in Fig. 4. The *Bank Institute* offers a service to *Conduct Financial Transactions*. The institute orders the development of the *Online Banking Service* by the *Internal Development Unit*. The cloud provider *Hulda* provides the cloud resources to implement the service. The *Bank Customer Conducts Financial Transactions*. The *VIP Bank Customer* conducts a specific kind of financial transaction, a so-called *24/7 Financial Transaction* that has a guaranteed availability on all days of the week of 99.9999 %. This is guaranteed by a SLA. This

[10] http://www.iso27001security.com/html/iso27k_toolkit.html.

Table 3 Template for management approval of the ISMS

Management commitment	
ISMS security goal	State the referenced security goal from the policy pattern
Establish responsibilities	State which person is responsible for the overall ISMS establishment
Communicate importance of security	Define the actions taken to communicate the importance of security
Criteria for risk acceptance	Define worst case scenarios
Conduct ISMS audits	Define Audit responsibilities
ISMS management reviews	Define ISMS management audit responsibilities
Cloud-specific management commitment	
Decide a cloud deployment scenario	Define responsibilities for deciding to use a public, private or hybrid cloud deployment scenario
Check security assurances of cloud provider	Define responsibilities for analyzing the security assurances of the cloud provider
Conduct on site reviews of cloud provider	Define responsibilities for on site auditing of the cloud provider's data center(s)
Conduct neutral security assessment of cloud provider	Define who has to contract an external security team to validate the security audit of the cloud provider. If no external security team is contracted this management decision has to be justified
Resource management	
Provided resources for the ISMS	List the provided resources for establishing, implementing, operating, monitoring, reviewing, maintaining and improving an ISMS and reason why the listed resources are sufficient
Security supports business needs	List the resources that allow security support without interfering business needs
Competent personal	List the provided resources for establishing, implementing, operating, monitoring, reviewing, maintaining and improving an ISMS
Provide training	List the training programs initiated to ensure acceptable security levels
Effectiveness evaluation	List the measures taken to check the effectiveness of the measures
Records of education, training, skills, experience and qualification	Define responsibilities for documentation of education, training, skills, experience and qualification of personal with regard to security

service uses a specific fallback solution and disaster recovery.

We provide a template for management approval of the ISMS, presented in Table 3. The template structure is inspired by Sect. 5 of the ISO 27001 standard. The first row in the template lists the management concerns and the second the natural person that is responsible for this concern or the tasks or resources required to address the concern.

The template consists of three parts: *Management Commitment* states the responsible persons for the overall establishment of the ISMS and vital concerns toward its success, e.g., criteria for risk acceptance. *Cloud-specific Management Commitment* defines responsible persons for cloud-specific concerns like deployment scenarios and audit management. *Resource Management* states the required resources for establishing an ISMS. We use our running example with the template to show an integrity goal in Table 4.

Cloud computing relies on the evaluation of security controls at the site of the cloud provider [3]. We need to assign responsibilities for checking the security assurances of possible cloud providers and for on site evaluations of these providers. Moreover, a decision has to be made if a neutral third party performs the security assessment or if this is done with internal personal.

5 PACTS Step 2: define ISMS scope

The relevance of the scope definition of the ISO 27001 standards is already mentioned on page 1 of the standard. The ISMS establishment description in Sect. 4 of the ISO 27001 standard contains numerous references to the scope description and thus, highlights its importance. We listed all appearances in Table 5.

We introduced our *cloud system analysis pattern* in Sect. 2.2. The pattern supports the scope definition for the ISO 27000 series of standards. The contribution in this section is an updated cloud system analysis pattern and templates. We also devise a technique to refine the pattern and add further details. For example, behavioral descriptions using UML activity diagrams. We recently developed tool support[11] recently for our pattern, which was not part of previous publications. We also updated the cloud pattern with further elements and included behavioral descriptions in it.

5.1 The extended cloud pattern

We show the updated version of our *Cloud System Analysis Pattern* in Fig. 5. We included a *Location* element for the *Resource*s to define in which countries the cloud hardware

[11] http://www.uml4pf.org/cloudtool/cloudSystemAnalysisTool.zip.

Table 4 Instantiated template for management approval of the ISMS

Management commitment	
ISMS security goal	The transaction data of bank customers shall be kept confidential The integrity of the transaction data shall be preserved The online banking shall be available 24/7
Establish responsibilities	The responsible person from the bank institute, e.g., Mr. Jones
Communicate importance of security	The employees of the bank institute receive an education about the consequences to the bank caused by a loss of integrity
Criteria for risk acceptance	The bank wants to avoid bankruptcy
Conduct ISMS Audits	Mr. Jones is responsible for building the ISMS; hence, he should not be responsible for hiring or conducting the audits. Mr. Smith is responsible for conducting internal and external audits
ISMS management reviews	Neither Mr. Smith nor Mr. Jones should be responsible for the management reviews, because they are part of it. Instead this tasks is assigned to Mr. Shell
Cloud-specific management commitment	
Decide a cloud deployment scenario	Mr. Schneider is responsible for deciding to use a public, private or hybrid cloud scenario
Check security assurances of cloud provider	Mr. Schneider is responsible for this task, because he is the resident security expert
Conduct on site reviews of cloud provider	Mr. Schneider is also responsible for this task
Conduct neutral security assessment of cloud provider	Mr. Smith is responsible for the selection of an external security evaluator for the selected cloud provider
Resource management	
Provided resources for the ISMS	The ISMS requires external parties to conduct the checking of the file integrity of transaction information using e.g., [18] by the cloud provider. The resources for these integrity checks have to be provided
Security supports business needs	The integrity checking of the files should not make the transactions impossible or decrease the transaction time significantly
Competent personal	List all resources necessary for conducting integrity checks. These are financial resources for hiring security experts to conduct integrity checks
Provide training	The training program in this case is for auditing the cloud provider and the external party that conducts the integrity checks. The bank institute requires skilled parties to conduct these audits
Effectiveness evaluation	Have an audit that checks all taken measures. In this case, audit training programs and personal. A specific audit for that case has to be taken
Records of education, training, skills, experience and qualification	Mr. Jones is responsible for fulfilling documentation demands, e.g., which external party was hired and the reasons for hiring this particular party

Table 5 Relevance of the scope definition within the ISO 27001 standard

Significance for the ISO 27001 Standard	The ISMS scope definition of the ISO 27001 standard is a vital step for its successful implementation, because all subsequent steps use it as an input
Related section(s) of the standard	Section 4 describes the Information security management system and in particular in Sect. 4.2—establishing and managing the ISMS states the scope definition. Section 4.2.1 a demands to "Define the scope and boundaries of the ISMS in terms of the characteristics of the business, the organization, its location, assets and technology, and including details of and justification for any exclusions from the scope" [4, p. 4]. Sect. 4.2.1 d concerns risk identification and the section recommends to consider the scope definition for identifying assets. Section 4.2.3 demands management reviews of the ISMS that also includes to check for possible changes in the scope of the ISMS. Section 4.3 lists the documentation demands of the standard and Sect. 4.3.1 d requires a documentation of the scope of the ISMS

and software is located. This supports the ISO 27001 demands for including locations in the context description found in ISO 27001 Sect. 4.2.1 a. Moreover, in order to identify relevant laws we need the location information, as well (Fig. 6).

Moreover, we added two cloud-specific software components to the pattern (see Fig. 5). The first component is a *Cloud database*. This provides the scalability in terms of data storage. Cloud databases differ from traditional SQL databases. The main differences are that cloud databases allow inconsistencies in the data storage for a short period of time and a decreased control over the data in the cloud due to data distribution. The latter manifests itself in the fact that cloud providers are often not able to provide detailed information on the location of their customers' data [19]. An example for a cloud database is Google's Bigtable [20]. An open question

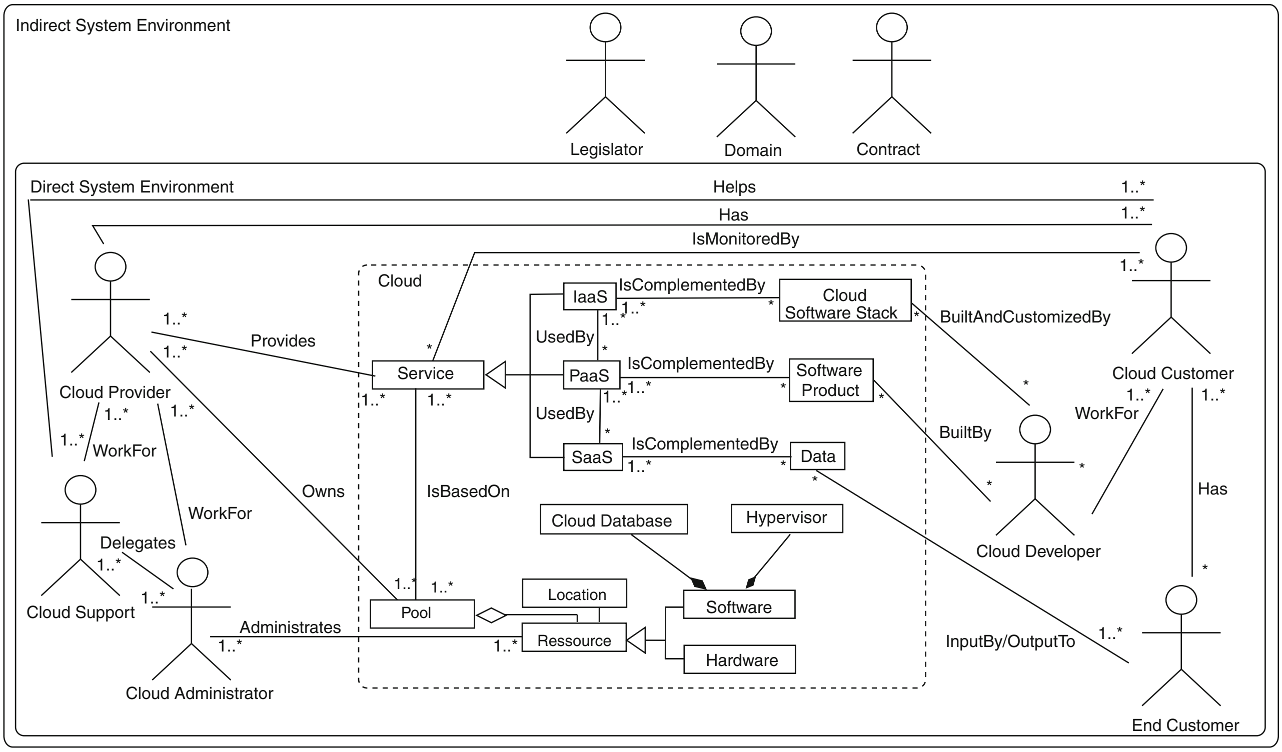


Fig. 5 Extended cloud system analysis pattern

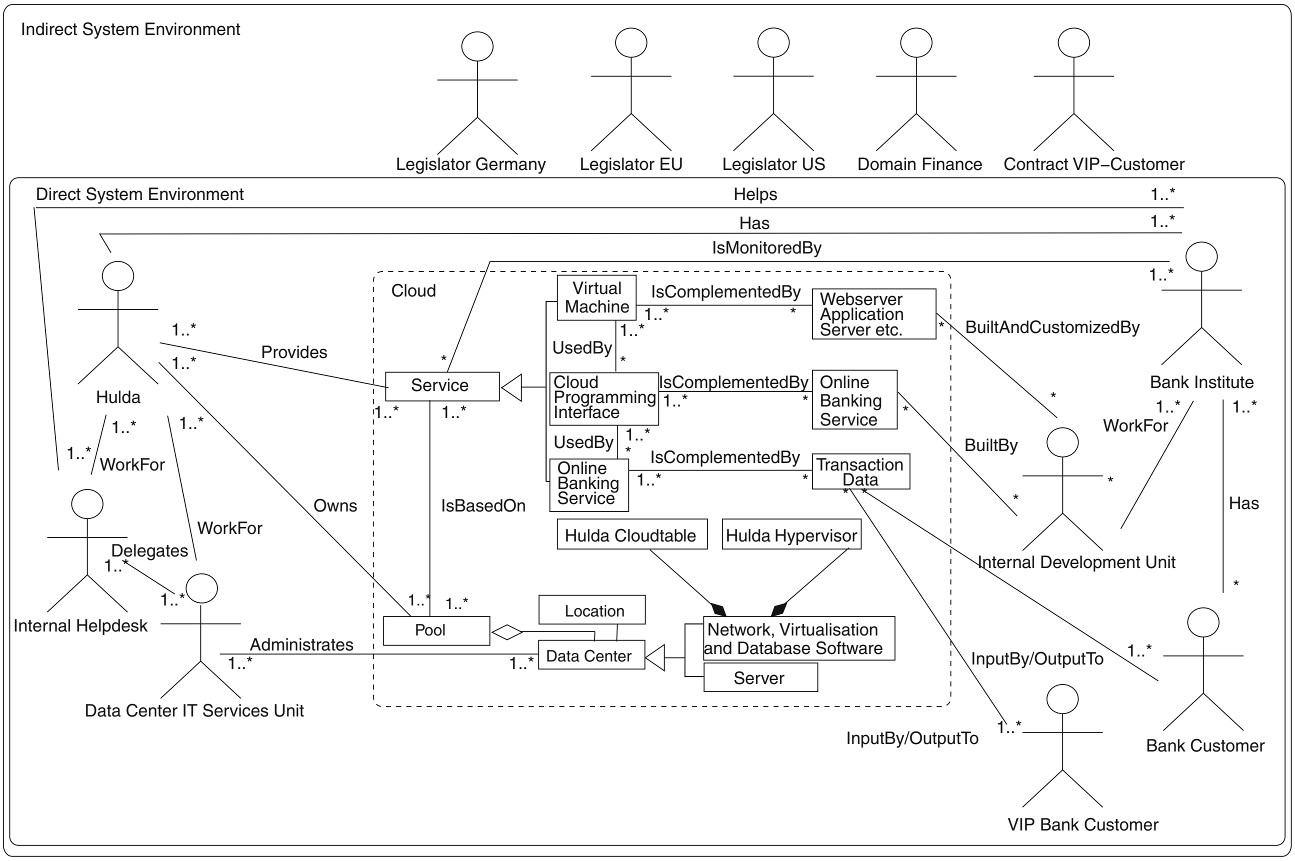


Fig. 6 Example instantiation of our extended cloud system analysis pattern

Table 6 Direct stakeholder template—updated version from [9]

Name	State the identifier of the stakeholder or group of stakeholders, e.g., company name or group of end customers
Description	Describe the stakeholder informally, e.g., if the stakeholder is a natural or a legal person
Relations to the cloud	Describe the input and output represented as relation (line from this stakeholder to the cloud) between the stakeholder and the cloud, e.g., the kind of data or software
Cloud deployment scenarios	State the deployment scenarios the cloud stakeholder demands: public, private or hybrid. Also state the reason for the particular deployment scenario
Location	State the country the stakeholder works in
Motivation	State the motivation of the stakeholder for using the cloud based on the previous considered relations to the cloud, e.g., business goals such as profit increase
Relations to other direct stakeholders	For each relation (line from this stakeholder to another direct stakeholder), name the kind of dependency between the stakeholders, e.g., indirectly influenced by customer-demand
Assets	State the assets of the stakeholder that are already known
Compliance	State relevant laws and regulations for the cloud scenario that are already known

Table 7 Indirect stakeholder template—updated version from [9]

Name	See direct stakeholder template
Description	See direct stakeholder template
Relations to other stakeholders	For each relation from this stakeholder to another direct or indirect stakeholder (no line explicitly shown), name the kind of dependency between the stakeholders, e.g., protected by, controlled by law, implement laws
Motivation	State the motivation of the stakeholder for having any reason of considering the cloud for its work or the motivation for having any kind of relation to stakeholders of the direct or indirect environment, e.g., protect privacy of citizens or implement concrete laws of an economic community
Compliance	Identify relevant laws as well as regulations based on the indirect stakeholders. Specify and identify the ones relevant for the stakeholder at hand, e.g., HIPAA

is how a cloud provider can prove that data have been deleted [21]. Cloud databases are particular relevant to the threat *Data Loss or Leakage* in Sect. 7.1.

The second component we add to our cloud pattern is the *Hypervisor*. According to Scarfone et al. [22] a hypervisor *controls the flow of instructions between the guest OS and the physical hardware, such as CPU, disk storage, memory, and network interface cards. The hypervisor can partition the systems' resources and isolate the guest OS so that each has access to only its own resources, as well as possible access to shared resources such as files on the host OS. Also, each guest OS can be completely encapsulated, making it portable. Some hypervisors run on top of another OS, which is known as the host operating system*. Therefore, the hypervisor is of particular interest considering the threat *Shared Technology Issues* (see Sect. 7.1).

Furthermore, we added two stakeholders to the *Direct System Environment*. The *Cloud Support Helps* the *Cloud Customer* when using the cloud and *WorksFor* the *CloudProvider*. The stakeholder is relevant, e.g., for the threat *Malicious Insiders* (see Sect. 7.1). We also introduce the stakeholder *Cloud Administrator* who *Administrates* the cloud's *Resources* and *WorksFor* the *CloudProvider*. This stakeholder is relevant, e.g., for the threat *Account or Service Hijacking* mentioned in Sect. 7.1.

Cloud Stakeholder Templates We supplement the cloud system analysis pattern by templates to systematically gather domain knowledge about the direct and indirect system environments based upon the stakeholders' relations to the cloud and other stakeholders. We accompany this cloud system analysis pattern by templates to systematically gather domain knowledge about the direct and indirect system environments based on the stakeholders' relations to the cloud and to other stakeholders. We updated the templates with location, cloud deployment scenarios, and privacy concerns with respect to a previous publication [9].

The first template serves to describe stakeholders contained in the direct system environment, shown in Table 6. The second template describes the stakeholders contained in the indirect system environment (see Table 7). In addition to our method, we have a hierarchical structure of models, which lets us analyze the cloud system at different decomposition levels or views. The use case diagrams in Sect. 4 are the initial model and the *Cloud System Analysis Pattern* is the first refinement level. Beyond the *Cloud System Analysis Pattern* there is no strict rule on the kind of diagrams to include or their scope. It depends on the size and technology involved in the cloud. Hence, our method

Table 8 Indirect stakeholder template: legislator Germany (cf. [9])

Name	*Legislator Germany*
Description	The *Legislator Germany* represents all German laws relevant for this cloud scenario
Motivation	The German laws try to control the risks of companies (*Hulda* and *Bank Institute*) and to protect the privacy of the *Bank Customers* by regulating disclosure of personal data
Relations to other stakeholders	Controlled by law: The laws have to be obeyed by all stakeholders of the *Direct System Environment*
Compliance	The following regulations might be considered
	Privacy protection: e.g., BDSG
	Risk management: e.g., AktG

Table 9 Direct stakeholder template: bank customer (cf. [9])

Name	*Bank customer*
Description	The *Bank Customer* uses the online banking service of the *Bank Institute*
Motivation	The *Bank Customer* wants low cost and secure financial transactions via the bank‘s cloud computing offer
Relations to the cloud	*InputBy/OutputTo*: *InputByfinancial data, data related to a person*, which is required for billing of the *Bank Institute* and maintenance of the cloud
Cloud deployment scenarios	The bank considers a public cloud, because it offers significant savings in terms of money
Location	The *Bank Customer* is located in Germany
Relations to other direct stakeholders	*Has*: *Bank Institute* as SaaS provider
Assets	Financial data and all data related to the person
Compliance	The following laws might be of relevance:
	Privacy protection: BDSG Sects. 3, 4, 9, and 11
	Risk management: AktG Section 91, Section 93

scales, because we can attach diagrams for different decomposition levels of the cloud or views to the *Cloud System Analysis Pattern*.

In our running example, we are interested in adding behavioral information about business processes to the pattern. Hence, we use UML [17] activity diagrams to show this particular view of the cloud system.

5.2 Instantiate the extended cloud pattern with our running example

We instantiate our cloud pattern using the example of the online banking service introduced previously. We consider the use cases introduced in Sect. 5 and in particular the financial institute (bank) and its bank customers. The financial institute is located in Germany. The bank plans to hire an internal software development unit to develop software for online banking in the cloud and a customized operating system (OS) for the developed online banking software. Hence, the bank plans to outsource the affected IT processes to the cloud to reduce costs and scale up their system for a larger amount of customers. Customer data such as transaction log history are stored in the cloud database, and transactions like money transfers are processed in the cloud.

The cloud developer creates SaaS systems for the bank institute via developing in a PaaS environment and customizing an OS for a given IaaS offer. The bank authorizes its internal software department, to design and build the cloud-specific software according to the interface and platform specification of the cloud provider. The main goal of the cloud provider, in our example a company called *Hulda*, is to maximize profit by maximizing the workload of the cloud. Therefore, subgoals are to increase the number of customers and their usage of the cloud, i.e., the amount of data as well as the number and frequency of calculation activities they outsource into the cloud. Fulfilling security requirements is only an indirect goal to acquire customers and convince them to increase the subset of processes they outsource. The bank customer is a person, juristic or natural, who has an account at the bank, which enables him to do financial transactions via the banking offers. In our scenario, this financial transactions can be done via the web service the bank offers using the cloud.

Basically, the online banking cloud service is embedded in 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

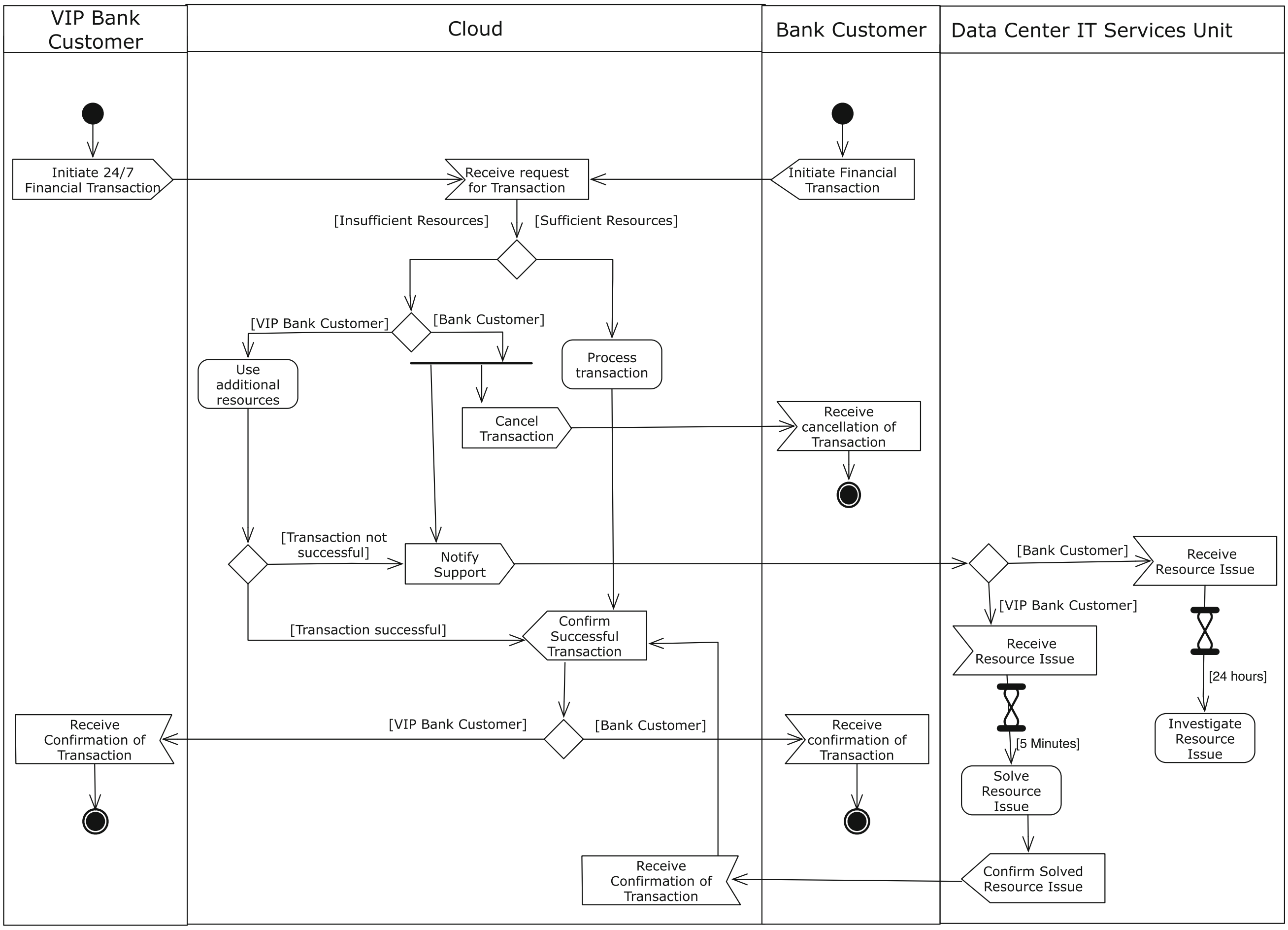

VIP Bank Customer
Cloud
Bank Customer
Data Center IT Services Unit
Initiate 24/7 Financial Transaction
Receive request for Transaction
Initiate Financial Transaction
[Insufficient Resources]
[Sufficient Resources]
[VIP Bank Customer]
[Bank Customer]
Use additional resources
Process transaction
Cancel Transaction
Receive cancellation of Transaction
[Transaction not successful]
Notify Support
[Bank Customer]
Receive Resource Issue
[VIP Bank Customer]
Receive Resource Issue
[Transaction successful]
Confirm Successful Transaction
[24 hours]
[5 Minutes]
Investigate Resource Issue
Solve Resource Issue
Receive Confirmation of Transaction
[VIP Bank Customer]
[Bank Customer]
Receive confirmation of Transaction
Receive Confirmation of Transaction
Confirm Solved Resource Issue

Fig. 7 Activity diagram describing the process of conducting a financial transaction

cloud through associations, e.g., the *Bank Customer*. Moreover, associations between stakeholders in the *Direct* and *Indirect System Environment* exist, but not between stakeholders in the *Indirect System Environment* and the cloud. For example, the *Legislator Germany* is part of the *Indirect System Environment*. Typically, the *Indirect System Environment* is a significant source for compliance and privacy requirements.

We derive the indirect stakeholders required for this scenario based on the instantiation of the *Direct System Environment*. The *Cloud* is located in Germany and the USA. This is the reason for the *indirect stakeholders Legislator* Germany and *Legislator* US. Germany is a member of the European Union resulting in an additional set of regulations. They are described by the *Legislator* EU that represents a set of EU regulations. The financial institute has also several contractual obligations, one of which is the *Contract VIP Customer* that defines the 24/7 availability of the online banking system. As examples, we present one stakeholder template instance for an indirect stakeholder (see Table 8) and one for a direct stakeholder (see Table 9).

The *VIP Bank Customer* relates in a similar manner to the cloud as the *Bank Customer*, because the pattern focuses on structural information. We show an activity diagram for conducting a financial transaction in Fig. 7, which illustrates the difference in behavior between those customers. We have two different kinds of end customers, the *Bank Customer* and the *VIP Bank Customer*. The *VIP Bank Customer* is entitled to a 24/7 financial transaction service, while the *Bank Customer* is only entitled to a normal financial transaction service. The difference is that the normal transaction services gives only very limited guarantees to the availability of the service. The 24/7 transaction service provides the guarantee that a transaction can be conducted at 99.9999 % of the time and that any occurring problem is fixed within 5 min.

The process depicted in Fig. 7 begins with either the *Bank Customer* or the *VIP Bank Customer* initiating a financial transaction. We focus on the transaction of the *Bank Customer*. The transaction request is sent to the cloud and executed if sufficient resources exist. The *Financial Institute* rented only a limited amount of resources in the cloud and scaling these resources causes an increase in the

Table 10 Demands for asset identification of the ISO 27001 standard

Significance for the ISO 27001 standard	The design goal of the ISO 27001 ISMS is to protect assets with adequate security controls and this is stated already on page 1 of the standard
Related section(s) of the standard	Section 4 describes the Information security management system and in particular in Sect. 4.2—establishing and managing the ISMS states the scope definition. Section 4.2.1 a demands the definition of assets. Section 4.2.1 b concerns the definition of ISMS security policies demands that the policy shall consider assets. Section 4.2.1 d that concerns risk identification uses the scope definition to identify assets, to analyze threats to assets, and to analyze the impacts of losses to these assets. Section 4.2.1 e concerns risk analysis, which also clearly define to analyze assets and to conduct a vulnerability analysis regarding assets in light of the controls currently implemented

costs. Thus, these increases in resources will not happen for single *Bank Customer*s. Only if sufficient numbers are present the *Financial Institute* will increase the resources. This is fundamentally different from the *VIP Bank Customer*s, who pay for the scaling of resources. Hence, if resources are not sufficient for conducting a financial transaction of a *Bank Customer*, the system notifies the *Bank Customer* that the transaction is not possible at this time and suggests to try again later. In addition, the cloud sends a message to the *Data Center IT Services Unit*. The unit investigates within 24 h if the cloud resources need to be increased, due to a significant amount of requests. Otherwise, the unit does nothing.

Should there not be sufficient resources to conduct the financial transaction of the *VIP Bank Customer*, the cloud service automatically uses further resources to conduct the transaction. In addition, should any problem occur during the transmission the *Data Center IT Services Unit* has to solve the problem within 5 minutes. The *VIP Bank Customer* is informed about the successful transaction afterward.

This concludes our ISMS scope description in our example. We propose to use multiple processes and further refinements in scope descriptions, but for space reasons we limit ourselves to one. In addition, process descriptions can also benefit from documentation standards for IT Services, e.g., ITIL [23]. These provide example processes for typical tasks regarding IT management.

6 PACTS Step 3: identify assets

The ISO 27001 standard lists the protection of assets with adequate security goals already on its first page. In Table 10 we state several references of the standard toward asset identification. These references occur in particular during the scope definition, policy definition and risk estimation phases of the standard.

This section is inspired by the idea of using the relations in the cloud system analysis pattern to identify assets presented in [9]. We improved the content of the aforementioned publication with a structured method in this work. We enhance the asset identification by already identifying assets using simply the cloud pattern. These assets can be re-used for different projects. In addition, we check the instantiated cloud pattern for further assets. Hence, the contribution of this section is a structured method for asset identification using the cloud system analysis pattern and its instantiation.

Figure 9 depicts an overview of this method. We explain its steps in the following (Fig. 8).

Instantiate asset template The ISO 27001 standard defines an asset [4, p. 2] as follows: “anything that has value to the organization”. The organization in our case is either the *cloud provider* or the *cloud customer*.

We identify assets in the cloud pattern by analyzing the associations (the lines) from all stakeholders toward the cloud. We check if the cloud elements at the end of the associations have potentially value to the stakeholders and, thus, are assets. If they are assets, we check if associations between these and further cloud elements lead to further assets. Let us take the end customer to illustrate this step (see Fig. 8). The association from the stakeholder to the cloud leads to *Data*. These data have potential value to the end customer, because the *Data* should be processed or stored by the cloud. It is not likely that the end customer invests money into processing or storing *Data*, which have no value to her. Hence, we identify *Data* as an asset. The relation between *Data* and *SaaS* is also investigated and the *SaaS* is used by the end customer, but not of particular value. We assume that the *end customer* can also use other offers.

The identified assets are stored in Table 11. We explain in the following the technique for identifying assets using the cloud pattern. The benefit of identifying asset in the pattern is that the identified assets can be re-used for each instantiation of the pattern.

It also contains the information of a so-called *asset provider*. This is a stakeholder that either owns an asset or creates an asset. For example, the *cloud developer* built the *software product*. The last column of the table contains a

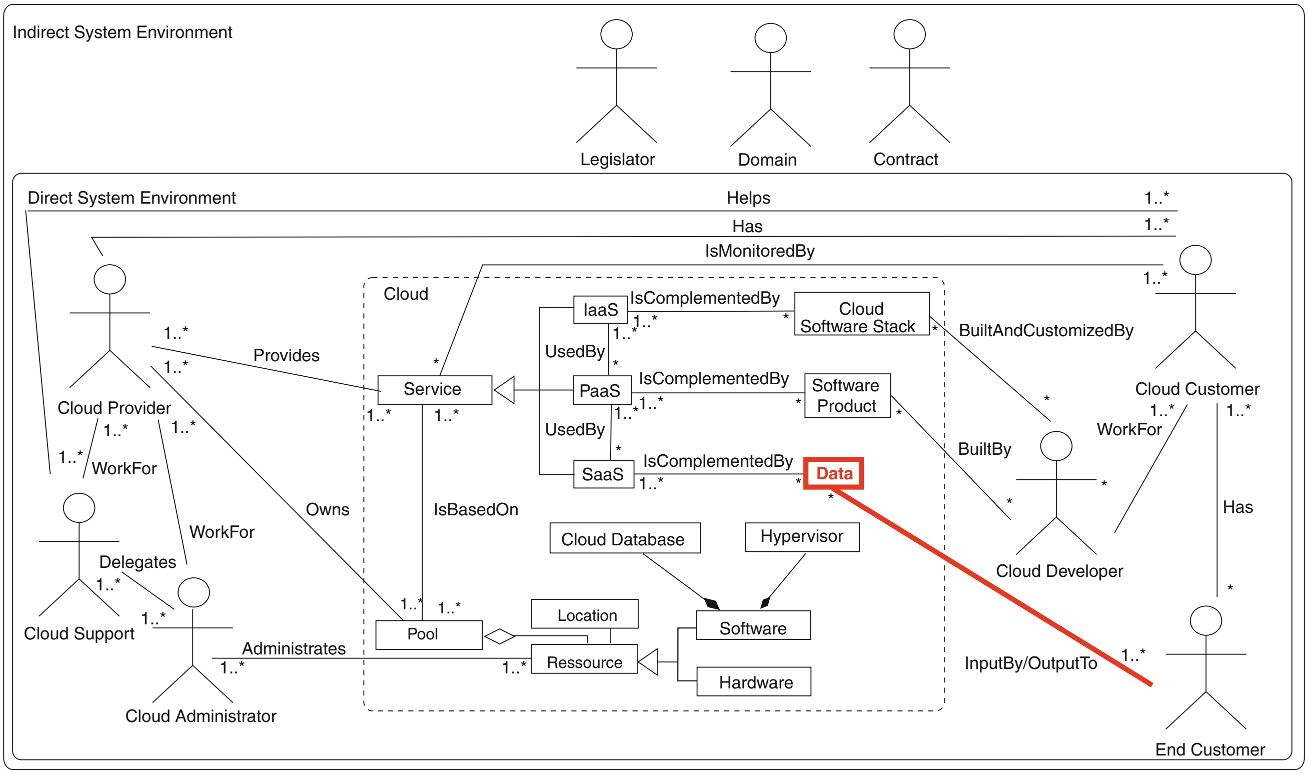


Fig. 8 Asset identification for the end customer

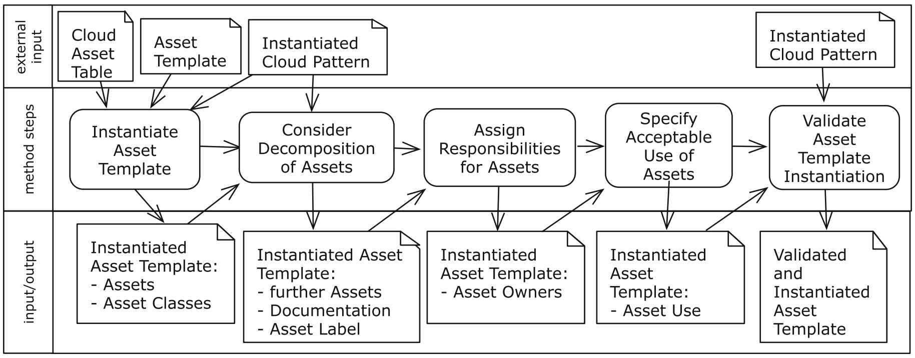

external input
method steps
input/output
Cloud Asset Table
Asset Template
Instantiated Cloud Pattern
Instantiated Cloud Pattern
Instantiate Asset Template
Consider Decomposition of Assets
Assign Responsibilities for Assets
Specify Acceptable Use of Assets
Validate Asset Template Instantiation
Instantiated Asset Template:
- Assets
- Asset Classes
Instantiated Asset Template:
- further Assets
- Documentation
- Asset Label
Instantiated Asset Template:
- Asset Owners
Instantiated Asset Template:
- Asset Use
Validated and Instantiated Asset Template

Fig. 9 A method for cloud-specific asset identification

reasoning that the asset has value to a stakeholder and harm to it would effect this stakeholder. The stakeholder could, e.g., suffer financial loss in case her assets are financial data. In addition, the data could also contain personal information and leaking it could also harm the stakeholder.

An example for considering not only direct relations to the *cloud* is the *cloud provider* (see Fig. 10), who owns the *pool*. Harm to the *pool* could result in the bankruptcy of the *cloud provider*. The pool consists of *resources*, which are also assets. We include the *Hardware, Software, Cloud Database*, and *Hypervisor* in a similar manner.

We use an asset template in our method, shown in Table 12. Our asset template collects all the information required for assets by the ISO 27001 standard [4, p. 15]. In the first step, we instantiate our cloud asset table (see Table 11) and enter the names of the assets into the first column of our asset template. We explain the instantiation of the remaining columns in the following.

Consider decomposition of assets The second step of our method is the decomposition of the assets, which are already listed in the instantiated asset template. The *transaction data* of the *VIP Bank Customer* (see Fig. 11). We present as an example the details of the *transaction data* as a UML class diagram, see Fig. 12.

Assets should be decomposed if the decomposition reveals further information for classifying or describing the assets. For example, the *Webserver, Application Server, etc.*

Table 11 Cloud asset table

Asset	Asset provider	Asset reasoning
Cloud software stack	Cloud developer	The *cloud software stack* is the basis for the *software product* of the *cloud customer*. Harm to it can effect the functionality of the *software product* and cause financial harm to, as well as harm to the reputation of, the *cloud customer*
Software product	Cloud developer	The *software product* is essential to the business of the *cloud developer* and harm to can cause financial harm to the *cloud customer*. Harm to this asset can also cause harm to the reputation of the *cloud customer*
Data	End customer	Harm to the asset can possibly cause financial loss and privacy violation to the *end customer*. The harm depends on the kind of data
Resources	Cloud provider	The resources are the essential infrastructure of the *cloud* and harm to these can cause bankruptcy of the *cloud provider*
Hardware	Cloud provider	The resources are the essential infrastructure of the *cloud* and harm to these can cause bankruptcy of the *cloud provider*
Software	Cloud provider	The resources are the essential infrastructure of the *cloud* and harm to these can cause bankruptcy of the *cloud provider*.
Hypervisor	Cloud provider	The resources are the essential infrastructure of the *cloud* and harm to these can cause bankruptcy of the *cloud provider*
Cloud database	Cloud provider	The resources are the essential infrastructure of the *cloud* and harm to these can cause bankruptcy of the *cloud provider*

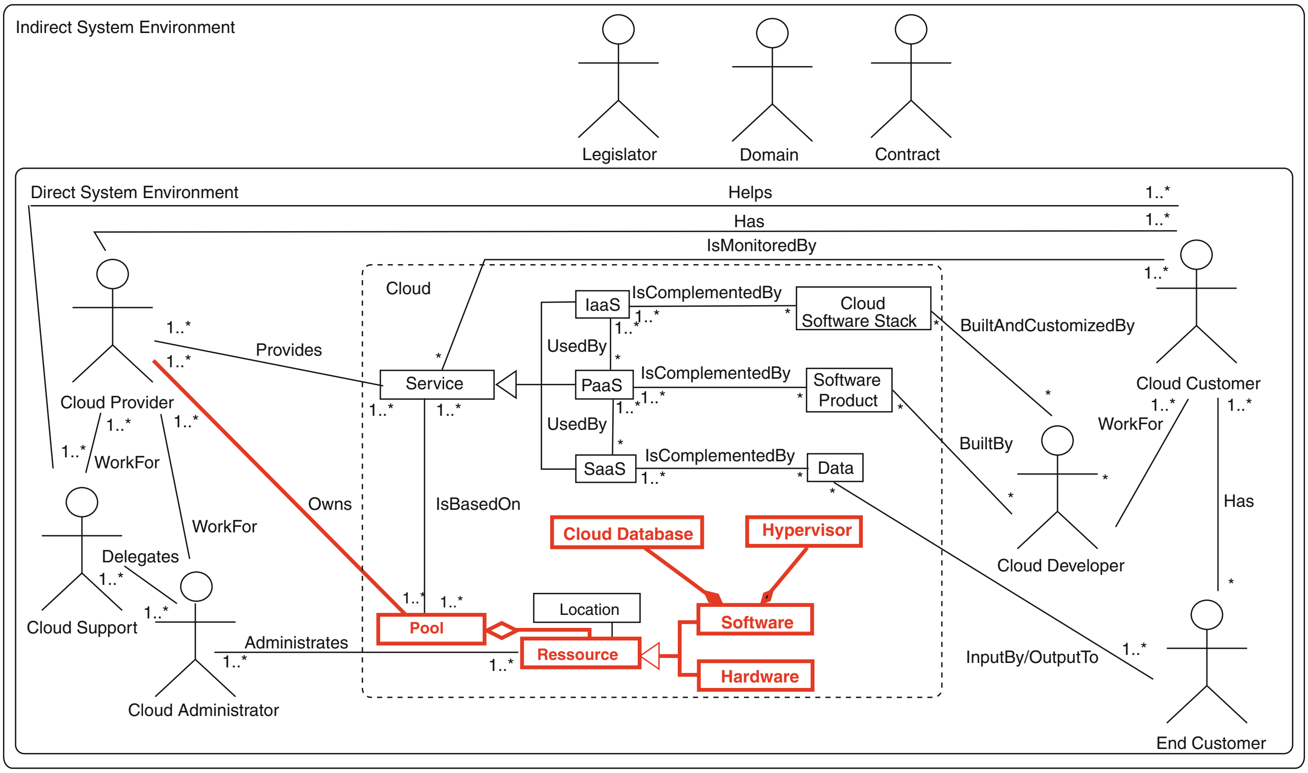


Fig. 10 Asset identification for the cloud provider

is decomposed into two web servers for redundancy, and also several different types of application servers. All of these have to be listed in the asset template. For simplicities' sake, we do not refine assets further for this example.

Assign responsibilities for assets This step of our method concerns the assignment of responsibilities for assets and relates to the second column of our asset template. The column lists so-called *asset owners*. The standard defines

Table 12 Instantiated asset template

Asset	Asset owner	Asset use	Asset class	Asset label
Webserver, application server, etc	Mr. Smith	External activity diagram	Software	AS_SO_100
Online banking service	Mr. Jones	External activity diagram	Software	AS_SO_110
Transaction data	Mr. Jones	See Fig. 14	Data	AS_DA_120
Data center	Mr. Mintz	External activity diagram	Physical	AS_PH_100
Network, virtualization and database software	Mr. Lock	External activity diagram	Software	AS_SO_130
Hulda cloudtable	Mr. Lock	External activity diagram	Software	AS_SO_140
Hulda hypervisor	Mr. Lock	external activity diagram	Software	AS_SO_150
Server	Mr. Mintz	external activity diagram	Hardware	AS_HA_100

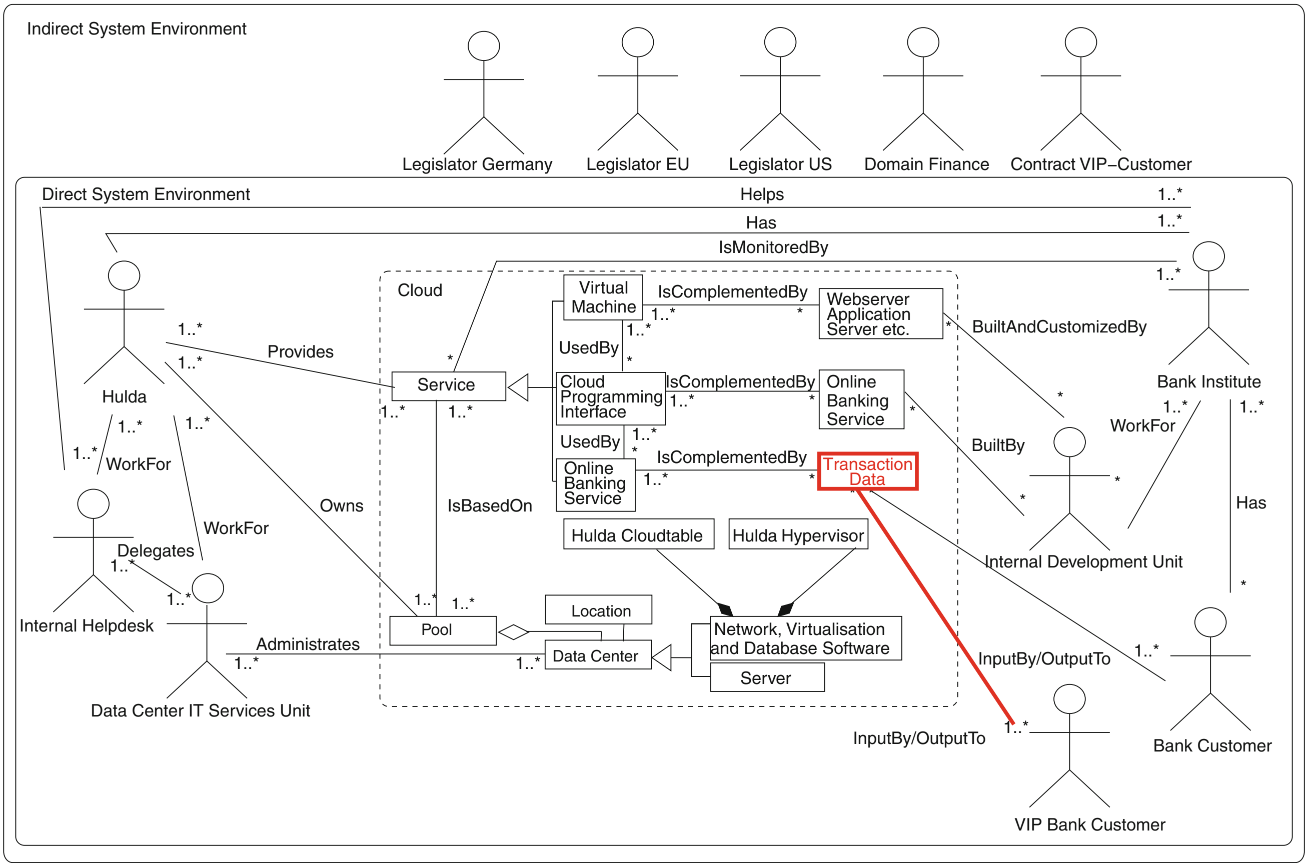


Fig. 11 Asset identification of the VIP bank customer

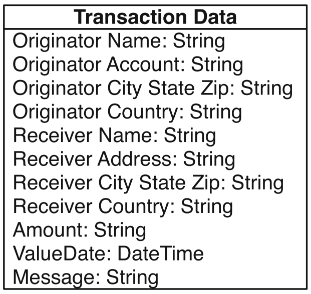

Transaction Data
Originator Name: String
Originator Account: String
Originator City State Zip: String
Originator Country: String
Receiver Name: String
Receiver Address: String
Receiver City State Zip: String
Receiver Country: String
Amount: String
ValueDate: DateTime
Message: String

Fig. 12 Decomposition of the asset transaction data

the term as follows: “The term owner identifies an individual or entity that has approved management responsibility for controlling the production, development, maintenance, use and security of the assets. The term owner does not mean that the person actually has property rights to the asset” [4, p. 15].

Specify acceptable use of assets The third column states the acceptable use of that asset, because using it outside of the specification can result in harm to that asset. We propose to specify the acceptable use of assets in UML activity

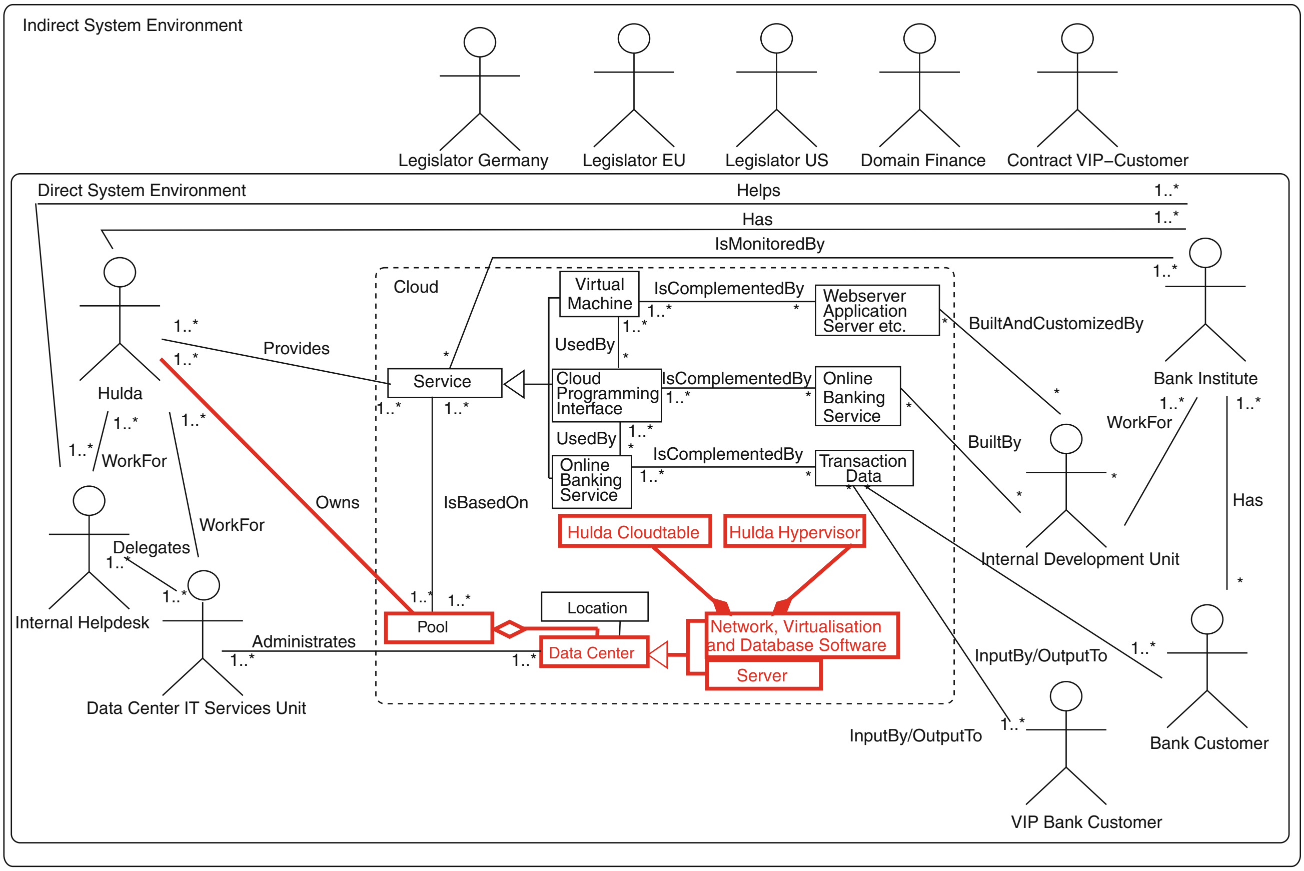


Fig. 13 Asset identification for the cloud provider Hulda

diagrams. For example, Fig. 14 specifies the acceptable use of the asset *transaction data*. The data are send to the cloud, where it is processed and stored for 30 days. Afterward it is deleted. If the deletion fails, the *Data Center IT Services Unit* gets a message from the cloud and executes the deletion of the data. For space reasons, we only show one diagram. Assets also have to be classified in order have a unique identifier per asset. We propose a classification into *Hardware, Software, Data* and *Physical*. The fourth column states the classification of an asset and the last column its unique label. We propose a labeling schema that uses the first two letters of the word asset, followed by the first two letters of the asset class and an increasing number.

Validate asset template instantiation We propose to check the instantiation of the asset template via several validation conditions. We have identified the following conditions so far:

- Check the instantiated cloud pattern for assets. For example, we present the asset identification for *Hulda* in Fig. 13. This is in particular relevant if the cloud pattern has been extended with further cloud elements or stakeholders.
- Check all cloud elements for not yet considered assets. This validation condition should check for completeness.
- Check if the instantiated asset template has empty fields. After the conclusion of this step all fields of the instantiated asset template should be filled. If this is not the case the information for the missing fields has to be elicited.
- Check if assets are considered more than once. The asset template does not contain a check for duplicate entries. The security expert shall check if two assets are in fact the same and remove the duplicates.
- Do some assets require further refinement? This validation condition answers the question if the decomposition is complete. The security expert shall ask themselves if an asset contains further assets or if the decomposition is complete.

7 PACTS Step 4: analyze threats

The ISO 27001 standard demands a threat analysis in order to determine and analyze risks to identified assets (see Table 13). The background of this section are the cloud

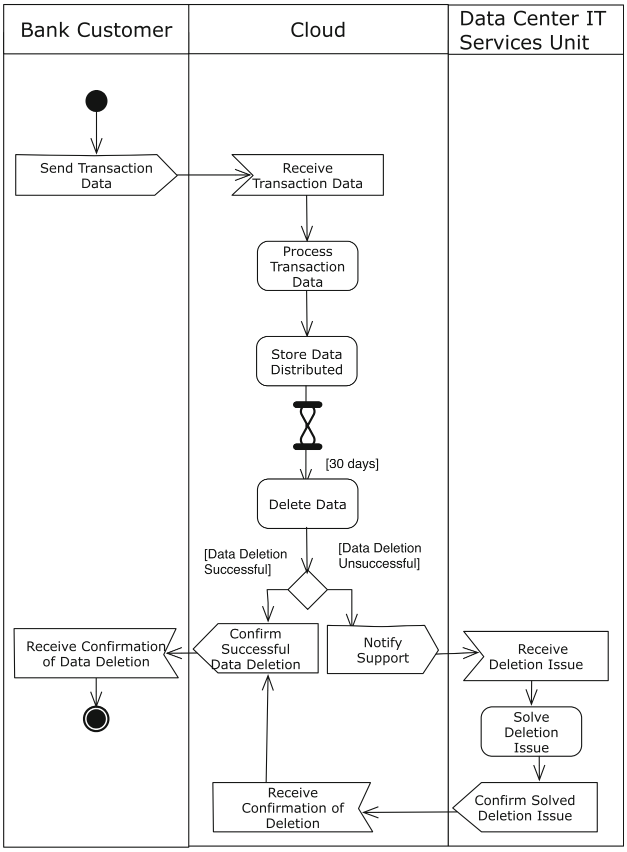

Bank Customer
Cloud
Data Center IT Services Unit
Send Transaction Data
Receive Transaction Data
Process Transaction Data
Store Data Distributed
[30 days]
Delete Data
[Data Deletion Successful]
[Data Deletion Unsuccessful]
Receive Confirmation of Data Deletion
Confirm Successful Data Deletion
Notify Support
Receive Deletion Issue
Solve Deletion Issue
Receive Confirmation of Deletion
Confirm Solved Deletion Issue

Fig. 14 Description of use of transaction data

security issues from CSA and Gartner. We combine these threats with our cloud pattern and use the results of misuse cases to elicit security requirements. Our contribution in this section is a cloud-specific threat analysis method.

7.1 Cloud security alliance: top threats to cloud computing

The CSA presents a list of seven threats for clouds and their relations to IaaS, PaaS, and SaaS [2]. We use this particular list of cloud threats, because it summarizes the experience in the field of cloud computing from the point of view of a large industrial consortium. In the following, we present a short summary of these threats:

Abuse and nefarious use of cloud computing Nefarious can mean criminal or treacherous.[12] This threat describes the abuse of the scalable cloud resources, e.g., storage or network capacity. For example, the resources can be used by spammers or malicious code authors.

This threat refers to the cloud service models: IaaS and PaaS.

Insecure interfaces and APIs Clouds provide interfaces for provisioning, management, orchestration, and management of services. Security functions, e.g., authentication, access control, and encryption rely upon these. Hence, malicious use of these interfaces has to be prevented. An example for the malicious usage of interfaces is the eavesdropping during clear-text transmission of content.

This threat refers to the cloud service models: IaaS, PaaS, and SaaS

Malicious insiders The cloud provider controls access to the cloud. A cloud customer or end customer has very limited transparency considering data access permissions provided to cloud employees. Hence, the threat of malicious insiders, which are employees of the cloud provider, scales with the resources and offered services in the cloud. An example for a specific problem is policy compliance. Cloud customers or end customers have no visibility into the hiring or monitoring of the cloud providers' employees.

This threat refers to the cloud service models: IaaS, PaaS, and SaaS.

Shared technology issues The different stakeholders in the cloud use the same physical resources, e.g., CPUs and GPUs. These are shared using so-called *Hypervisor*s, which provide isolation properties for these physical resources. Side channel attacks on these *Hypervisor*s can provide a stakeholder with inappropriate levels of control of the underlying cloud infrastructure.

This threat refers to the cloud service models: IaaS.

Data loss or leakage The threats to data in a cloud scales with the amount of data stored in it. Deletion or alteration of data without a backup is an example. Moreover, cloud databases store data distributed. The links to records in these cloud databases can be destroyed, which results in unrecoverable data. In addition, the loss of an encoding key renders data useless.

This threat refers to the cloud service models: IaaS, PaaS, and SaaS.

Account or service hijacking Clouds provide numerous services and credentials, and passwords are often reused. Thus, compromised credentials provide access to a large set of data about activities and transactions of stakeholders. Thus, the attacker can exploit the reputation of a cloud customer and launch a large-scale attack on its end customers. The cloud customer's reputation can lead to directed phishing and farming attacks at its end customers.

This threat refers to the cloud service models: IaaS, PaaS, and SaaS.

Unknown risk profile Cloud customers and end customers do not own cloud resources. Hence, cloud providers can apply the so-called *security by obscurity* policy. Thus, the

[12] According to http://thesaurus.com/browse/Nefarious?s=t.

Table 13 Threat analysis demands within the ISO 27001 standard

Significance for the ISO 27001 standard	The ISO 27001 standard concerns threat analysis in several sections for determining the risks to assets
Related section(s) of the standard	Section 4.2.1 d demands a threat analysis for assets for the purpose of identifying risks and the vulnerabilities that might be exploited by those threats. Section 4.2.1 e concerns risk analysis and evaluation and demands to determine likelihoods and consequences for threats. Section 4.2.4 d concerns there view process of the ISMS and also demands a threat identification. Section 7.2 that concerns the management review of the ISMS also demands a threat analysis

cloud customers and end customers do not know the exact specifications of the security mechanisms used in the cloud. This results in an unknown exposure of assets and increases the difficulty of creating a risk profile for a cloud scenario.

This threat refers to the cloud service models: IaaS, PaaS, and SaaS.

7.2 Gartner's cloud security risks assessment

Gartner assesses the security risks of cloud computing and defined a list of recommendations for cloud customers that help to evaluate cloud providers. The difference between risk and threats is that high-level risks cause loss to a stakeholder, while threats exploit vulnerabilities and can be used to realize attacks [24]. Hence, in order to map risks to threats we have to link them to possible vulnerabilities. In the following, we present Gartner's evaluation criteria and relate each criteria to the CSA threats. In addition, we formulate a new threat for each criteria that could not be mapped to an original CSA threat.

Privileged user access Sensitive data processing outside the organization or by non-employees leads to an uncertain level of risk, because the security controls of the organization are bypassed. The cloud costumer depends on the controls of the cloud provider for upholding security assurances for sensitive data. The cloud provider should release information about hiring and oversight of all personal that has access to the sensitive data and controls concerning access to the data.

Relation to CSA threats: Malicious Insiders, Unknown Risk Profile.

Compliance Regulations hold the cloud customer responsible for the security of their organization's and customer's data. Hence, cloud customers should demand security certifications, which include documentation of controls, as well as security audits.

Relation to CSA threat: Unknown Risk Profile.

Data location Several privacy regulations demand that personal information stays in certain geographical regions. The cloud customer has to know if the cloud provider upholds privacy regulations and can restrict personal information from flowing into restricted geographical regions.

Relation to new threat: Unrestricted Flow of Personal Information.

Data segregation Data have to be transferred to the cloud and stored in it. Encryption is one solution for protecting the data. In transit most cloud providers use SSL, but for storing data in the cloud, not all cloud providers use encryption. The cloud customer has to check, which kind of encryption is used and who tested and analyzed its implementation. In addition, the cloud provider has to provide the information of who has the key for decrypting the data.

Relation to CSA threats: Data Loss or Leakage, Insecure Interfaces and APIs.

Availability The cloud customer has to check the commitments regarding availability of the cloud provider. These commitments have to be in contractual form of service level agreements. The contracts have to contain written penalties for the cloud provider.

Relation to new threat: Insufficient Service Level Agreements.

Recovery The cloud customer requires insurances for data recovery in case of total disaster to the cloud. The cloud provider has to provide the specification of the backup systems and describe in detail if data replication is done and if a complete or partial data recovery after disaster is possible.

Relation to CSA threat: Data Loss or Leakage.

Investigative support Logging in clouds is difficult, because of changing hosts and data centers. Hence, the cloud customer can have difficulties to prove wrong doings or even to conduct investigations. The cloud customer requires the cloud providers written commitment to provide the means for investigations and evidence storing.

Relation to new threat: Impossible Investigations

Viability The long term viability of the cloud has to be evaluated. What happens to the data of the cloud customer if the cloud provider goes broke or is acquired? The cloud provider has to provide assurance that the data and applications can still be accessed after these events.

Table 14 Relations between cloud threats and security/privacy goals

Threats	Security goals					Privacy
	Confidentiality	Integrity	Availability	Authorization	Non-repudiation	
Abuse of cloud computing	effect	effect	effect	**cause**	effect	effect
Insecure interfaces and API	**cause**	**cause**	effect	**cause**	effect	effect
Malicious insiders	effect	effect	effect	**cause**	effect	effect
Shared technology issues	**cause**	**cause**	effect	**cause**	effect	effect
Data loss or leakage	**cause**	**cause**	**cause**	effect	effect	effect
Account or service hijacking	**cause**	effect	effect	**cause**	effect	effect
Unknown risk profile	effect	Effect	effect	**cause**	**cause**	effect
Unrestricted flow of PI	effect	effect	effect	effect	effect	**cause**
Insufficient SLAs	effect	**cause**	**cause**	effect	effect	effect
Impossible investigations	effect	effect	effect	effect	**cause**	effect

PI personal information, *SLA* service level agreement

Relation to CSA threat: Data Loss or Leakage.

Support in reducing risk The cloud customer has to evaluate the level of information and support provided by the cloud provider to safely and reliably use the cloud. Does the cloud provider support definitions of policies and attack prevention?

Relation to CSA threat: Unknown Risk Profile.

We propose to add the following threats to the CSA list based upon the missing relations between Gartner's cloud security risks and CSA threats.

Unrestricted flow of personal information Personal information is protected by regulations in several countries. If the cloud has no means to control the flow of personal information and to determine the location of it, these laws can be violated. The access to personal information outside its acceptable region also violates these laws.

Associated with Gartner Risk: Data Location.

Insufficient service level agreements Missing availability of the cloud and with it the cloud customer's data and application can cause financial loss and even put the cloud customer out of business. The reason is that the financial loss of the cloud customer, which results from the missing availability of the cloud, is not compensated by the cloud provider.

Associated with Gartner Risk: Availability.

Impossible investigations Logging in clouds is difficult to implement, because of the complexity of clouds. Hence, the cloud customer can be without the means to investigate possible wrong doings or prove crimes in the cloud.

Associated with Gartner Risk: Investigative Support.

We show in the following section how to relate these threats to our cloud pattern as a preparation to use both in a structured method.

7.3 Relations between threats and the cloud pattern

The mentioned threats may have a cause or an effect on security goals such as *integrity* or *non-repudiation*. In Table 14 we summarized the cause-effect relations between the threats and security goals. Note that we abbreviated the following threats: *Abuse and Nefarious Use of Cloud Computing* to *Abuse of Cloud Computing, Unrestricted Flow of Personal Information* to *Unrestricted Flow of PI*, and *Insufficient Service Level Agreements* to *Insufficient SLAs* for the remainder of this work.

The entry "effect" states that a threat has an impact on one or more security goal(s), e.g., the threat *Insecure Interfaces and API* has an effect on the security goals *Confidentiality, Integrity, Availability*, and *Non-Repudiation*. We also consider *Privacy* in this work, as discussed in Sect. 3. The entry "cause" means the pattern element may be involved in causing the threat, e.g., the *Cloud Developer* may cause the threat *Malicious Insiders* to occur (see third row in Table 15). Moreover, the threats *Shared Technology Issues* and *Data Loss or Leakage* have no *cause* entry, because the cause of these threats lie with the use of a particular technology, e.g., hypervisors. For example, the *Abuse of Cloud Computing* threat is caused by an *Authentication* problem. Hence, all cloud elements and stakeholders relevant for the cloud *Authentication* have to be a analyzed in detail. We conduct this analysis using our cloud pattern (see Sect. 5).

We summarize our findings in two tables: in the first table (Table 16) we consider the cloud view and relate the threats to the different elements found in this environment (see Fig. 18). In addition, we provide an overview on the affected layers. The second table (Table 15) shows how the threats manifest themselves in the direct system environment. Throughout the tables, we use the following elements to describe the identified relations: An "x" indicates

Table 15 Cloud threats: direct system environment view

Threats/cloud pattern	Cloud provider	Cloud customer	Cloud developer	End customer	Cloud administrator	Cloud support
Abuse of cloud computing	x	cause	cause	x	x	x
Insecure interfaces and API	x	x	cause	x		
Malicious insiders	x/cause	x	cause	x	cause	cause
Shared technology issues	x	x	cause	x	cause	
Data loss or leakage	cause	x	cause	x	cause	cause
Account or service hijacking		x	cause	x	cause	cause
Unknown risk profile	cause	x	cause	x	cause	
Unrestricted flow of PI		x	cause	x	cause	cause
Insufficient SLAs	cause	x		x		
Impossible investigations	x/cause	x	cause	x	cause	

PI personal information, *SLA* service level agreement

Table 16 Cloud threats: cloud view

Threats/cloud pattern	IaaS	PaaS	SaaS	CloudSoftwareStack	Software product	Data	Resources	Hardware	Software
Abuse of cloud computing	x	x	x	x	x	x[a]			
Insecure interfaces and API	x/cause	x/cause	x/cause	x/cause	x/cause				x/cause
Malicious insiders	x	x	x	x/cause		x	x	x	x
Shared technology issues	x	x	x	x/cause	x/cause		x	x	x/cause
Data loss or leakage	x	x	x	x/cause	x/cause	x			x/cause
Account or service hijacking	x	x	x	x/cause	x/cause	x			
Unknown risk profile	x	x	x	x/cause	x/cause	x	x/cause	x	x/cause
Unrestricted flow of PI	x	x	x	x/cause	x/cause	x	x/cause	x	x/cause
Insufficient SLAs	x	x	x	x	x	x			
Impossible investigations	x	x	x	x /cause	x/cause	x	x/cause	x	x

PI personal information, *SLA* service level agreement

[a] Considering browser based attacks

that an element of the pattern is somehow affected by the respective threat, e.g., the *Cloud Provider* is somehow affected by the threat *Abuse of Cloud Computing* (see first row in Table 15). However, to determine what exactly is affected it is necessary to conduct a further examination on the treated element. Some cells in Tables 15 and 16 do not have any of the above-mentioned entries assigned. Nevertheless, a free cell does not imply that there is no relation between threats and elements in the cloud pattern. It simply means that we did not come across any relation so far.

7.4 Cloud threat patterns

Our threat analysis is two-fold: First, we execute the threat analysis on the cloud pattern and create a *cloud threat table* that lists all stakeholders, cloud elements and possible misuse actions. Second, we consider the instantiated cloud pattern in a structured method (see Sect. 7.5), which relies on the information in the *cloud threat table*.

For a thorough analysis, all of the threats mentioned in Sect. 7.1 as well as Sect. 7.2 have to be considered and the relations to the cloud pattern. We iterate over the cloud threats (see Sect. 7.1) sequentially. Only after one threat has been successfully investigated, the analysis of the next one begins. To investigate the threats caused by cloud elements, we use Table 16. We select the entries in the table that share a *cause*-relationship and highlight the corresponding cloud elements in the cloud pattern.

The investigation of the threats caused by cloud stakeholders is based on Table 15. Similar to the cloud elements, we also highlight the cloud stakeholders that cause the threat in the cloud pattern. We check if further cloud elements or stakeholders exist, which can also give rise to the threat. For this purpose, we explore the relations

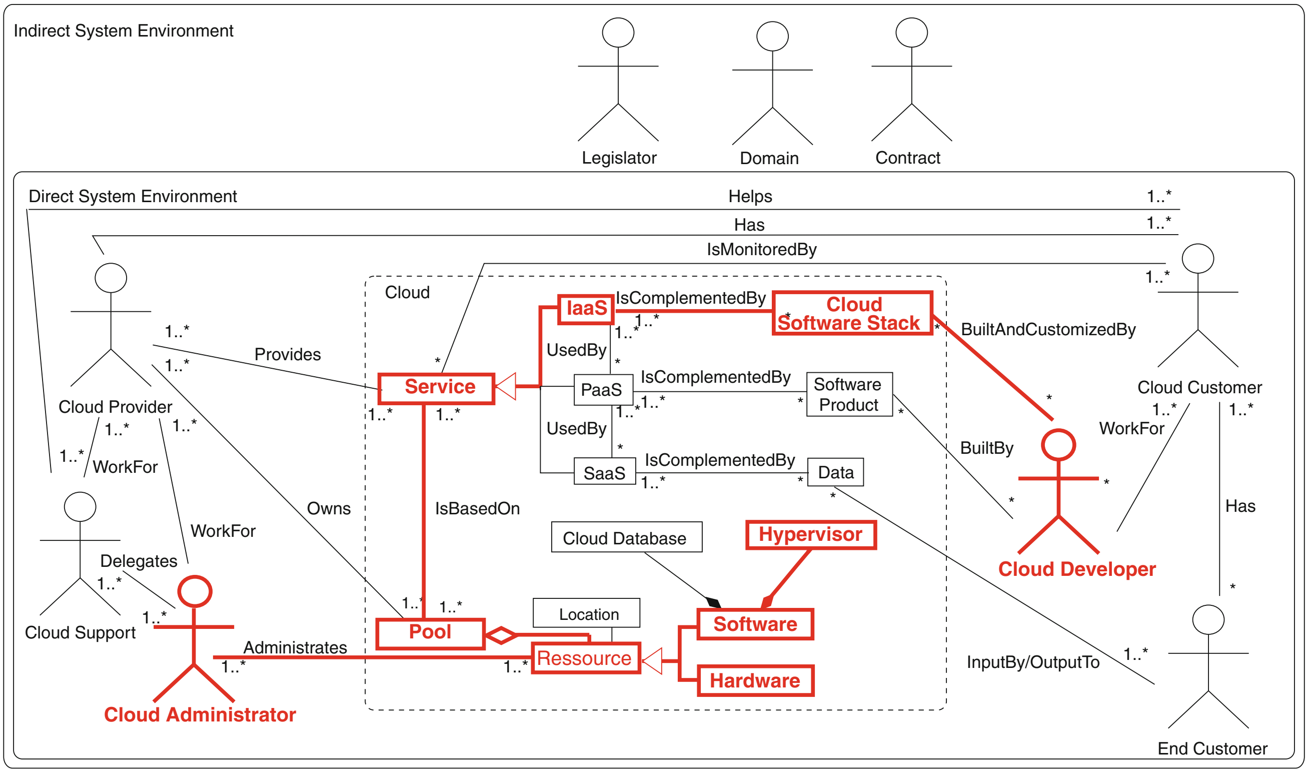


Fig. 15 Pattern-based cloud computing threat analysis for shared resources

between cloud elements. We start with the highlighted cloud elements and reason about the cloud elements that have relations (lines) to it. If a cloud elements contributes to a threat it is also highlighted in the cloud pattern. This is possible because we assume that cloud threats are caused by elements that have direct relations to each other. This continues until a cloud element does not contribute to a threat or no further cloud elements exist.

We limit ourselves to one threat to show the applicability of our technique. We choose to illustrate our approach using the threat *Shared Technology Issues*. We use Table 16 to identify cloud components that cause the threat. For our selected threat we have two *cause*-relationships, namely the *Cloud Software Stack* and *Software*. Table 15 states that the *Cloud Developer* and the *Cloud Administrator* are a cause for this threat. We mark the found elements in bold and red, see Fig. 15. We investigate the relationships from the highlighted stakeholders and cloud elements. Following the relations from the *Cloud Software Stack* to other cloud elements, we get to the *Virtual Machine* leading us to the *Service*. Both cloud elements support the sharing of resources. Hence, both cloud elements give rise to the threat. The *PaaS* cloud element only uses shared resources. Thus, it is not highlighted.

The threat we investigate is caused by managing the sharing of IT resources, e.g., the slicing of a physical hard drive into sections that virtual machines can use. The *Cloud Programming Interface* uses shared resources, but it does not participate in managing the sharing of IT resources. This cloud element just uses the IT resources it gets assigned by the *IaaS* layer or its instance of a *Virtual Machine*. Thus, we also exclude the *Cloud Programming Interface* as a cause for this particular threat.

The *Virtual Machine* has also a relation to the *Service*, which has one to the *Pool* that contains *Resources* and *Locations*. In this case, *Locations* do not give rise to the *Shared Technology Threat*, because the location of the technology is not related to the process of sharing IT resources. However, the *Resource* is relevant to the threat, because it contains the technology that allows the sharing of IT resources in the cloud. This technology is further refined into *Software* and *Hardware*. The *Hardware* is relevant, because it is the *Resource* that is shared and the *Software* is also shared and also orchestrates the sharing. The *Hypervisor* is a particular software for sharing resources and thus also highlighted. The *cloud database* is not relevant, because even though it provides resources it is not involved in sharing these.

The *Cloud Customer* is not involved in the technical realization of the *software product*. Thus, the stakeholder is not marked. The same argument holds for the *Cloud Provider*.

Table 17 Cloud threat table

Threat	Cloud element	Stakeholder	Threat actions
Abuse of cloud computing	–	Cloud Developer, Cloud Administrator, Cloud Support, Cloud Customer	Conduct cybercrime, execute treacherous IT attack
Insecure interfaces and API	IaaS, PaaS, SaaS, cloud software stack, software product	Cloud developer	Ignore security functions, corrupt interface, create backdoor
Malicious insiders	Cloud Software Stack, Resource, Software, Hypervisor, Software Product	Cloud Developer, Cloud Provider, Cloud Administrator, End Customer, Cloud Customer, Cloud Support	Neglect employee monitoring, missing background checks, hide information about employees
Shared technology issues	Cloud Software Stack, IaaS, Resource, Software, Hardware, Hypervisor	Cloud developer, cloud administrator	Side channel attacks, misconfiguration
Data loss or leakage	Cloud Software Stack, Resource, Software, Cloud Database, Software Product, IaaS, PaaS	Cloud Provider, Cloud Developer, Cloud Administrator, Cloud Support	Change data, loose data, not conducting backups
Account or service hijacking	Cloud Software Stack, Resource, Software, Cloud Database, Software Product, IaaS, PaaS, Saa	Cloud Developer, Cloud Administrator, Cloud Support	Attack end customer, steal credentials
Unknown risk profile	Cloud Software Stack, Resource, Software, Hardware, Cloud Database, Hypervisor, Software Product, IaaS, PaaS, SaaS	Cloud Provider, Cloud Developer, Cloud Administrator	Restrict security information, incomplete information gathering
Unrestricted flow of PI	Cloud Software Stack, Resource, Software, Cloud Database, Software Product, IaaS, PaaS, SaaS	Cloud Developer, Cloud Administrator	Ignore personal information, conduct global data distribution
Insufficient SLAs	–	Cloud provider	Write incomplete SLAs, write insignificant penalties
Impossible investigations	Cloud Software Stack, Software Product, Resources, Hardware, Software, Cloud Database, Hypervisor	Cloud Provider, Cloud Developer, Cloud Administrator	Implement incomplete logging, erase logging data

We executed the proposed techniques for all threats and show the resulting cloud threat table (see Table 17). The table lists the threat in the first column, the cloud elements and stakeholders that can cause the threat in the second and third column. The fourth column lists threat actions, which can help to build misuse cases. For example, a misuse case for the shared resources threat considers the threat action *misconfiguration*. The table is constructed using just the pattern and can be instantiated. Thus, the table can be re-used for different projects.

7.5 A method for pattern-based threat analysis for clouds

We present our method in Fig. 16 and explain each step in the following.

Create misuse cases We instantiate the columns *stakeholder* and *cloud element* of the cloud threat table. For example, the *cloud provider* is instantiated with *Hulda*. We use the misuse case notation [25] for the first step and in particular a textual representation of misuse cases as introduced by Deng et al. [26]. We iterate over all threats in our instantiated cloud threat table (see Table 17) and write misuse cases using the instantiated cloud pattern. We also consider the identified assets (see Sect. 6) in the misuse cases. For example, we consider the following misuse case for the *Malicious Insiders* threat and the asset *online banking service*:

1. The *bank institute* neglects to monitor the *internal development unit* during the development of the *online banking service*. The *bank institute* also did not conduct background checks of the team hired for the

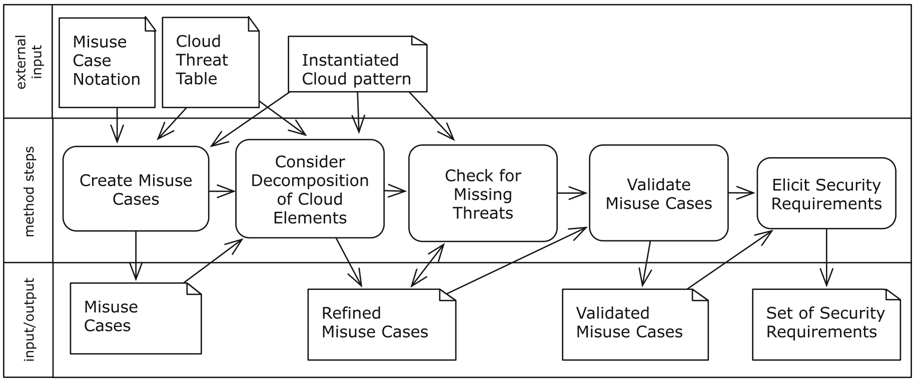


Fig. 16 A method for pattern-based threat analysis for clouds

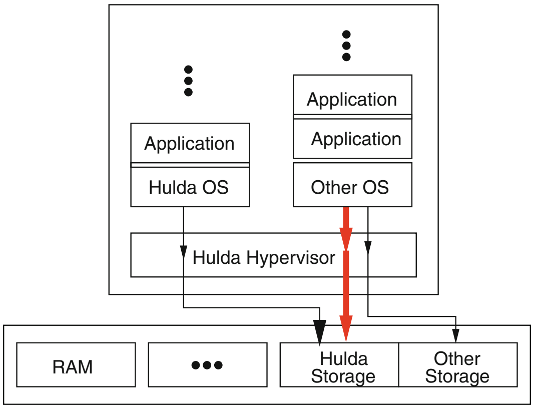


Fig. 17 Hypervisor threat analysis

development. This results in multiple black mail attacks on bank customers perpetrated by members of the *internal development unit*.

Consider decomposition of cloud elements We propose to analyze the initial set of misuse cases for further threats or threats that require refinement. If cloud components are involved in the threats these should be decomposed to provide the information for a more detailed threat analysis. For example, we consider the misuse case:

2. Integrity of *Transaction Data* might be compromised by a shared technology attack exploiting a vulnerability.

We decompose the *Hulda Hypervisor*, depicted in Fig. 17, because it organizes the sharing of resources. Hence, a threat analysis of this element is of particular interest. In our example, the *Hulda Hypervisor* separates a virtual machine running the operating system *Hulda OS* and another virtual machine that runs *Other OS*. The *Hulda Hypervisor* arranges that the *Hulda OS* can only access the *Hulda Storage* and the *Other OS* can access only the *Other Storage*. A possible threat is that, due to, e.g., a configuration mistake of the *Hulda Hypervisor*, the *Other OS* can access the *Hulda Storage*. In particular the transaction data of the *bank customer* and the *VIP bank customer* are threatened.

The *Data Center IT Services Unit* can be the stakeholder that causes a mistake in the configuration in the *Hulda Hypervisor*.

This information is considered in a refinement of the initial use case:

2. Integrity of *Transaction Data* might be compromised by a shared technology attack exploiting a vulnerability on the *Hulda Hypervisor* in order to gain access from one OS to another
3. The *Data Center IT Services Unit* does not configure the *Hulda Hypervisor* to ensure isolation of *Bank Customer*s and *VIP Bank Customer*s using the *Online Banking Service*.

Check for missing threats The instantiated cloud pattern has to be analyzed for missing or incomplete threats. We use the marked cloud pattern introduced in Sect. 7.4 and check if the information in the instantiated cloud pattern leads to further cloud elements that have to be considered. For example, *Hulda* uses the *Hulda Cloudtable* and *Hulda Hypervisor*. These components are implemented in such a way that the *cloud table* optimizes itself using the configuration information in the *Hulda Hypervisor* and also has the ability to adapt the *Hulda Hypervisor* configuration regarding sharing of database resources. Hence, we have to include the *Hulda Cloudtable* in the threat analysis for the shared resources threat (see Fig. 18).

Validate misuse cases We propose to check the misuse cases via several validation conditions. We identified the following:

- Are the misuse cases addressing the elements of the instantiated cloud pattern?
- Do all misuse cases consider at least one asset?
- Is a cloud stakeholder the cause and the victim of a threat?
- Did we consider decomposition of cloud components in sufficient degree?

Table 18 From misuse cases to security requirements for our running example

Misuse case	Security requirement
1. The *bank institute* neglects to monitor the *internal development unit* during the development of the *online banking service*.The *bank institute* also did not conduct background checks of the team hired for the development. This results in multiple black mail attacks on bank customers perpetrated by members of the *internal development unit*	Conduct background checks of the members of the *internal development unit* and hire external auditors to monitor the work of the *internal development unit*
2. Integrity of *Transaction Data* might be compromised by a shared technology attack exploiting a vulnerability on the*Hulda Hypervisor* in order to gain access from one OS to another OS	Ensure the integrity of the *Transaction Data* is not harmed by side channel attacks caused by the *Hulda Hypervisor*
3. The *Data Center IT Services Unit* does not configure the *Hulda Hypervisor* to ensure isolation of *Bank Customer*s and *VIP Bank Customer*s using the *Online Banking Service*	The *Hulda Hypervisor* has to be configured such that isolation of all users of the *Online Banking Service* is ensured
…	…

Table 19 Risk management demands of the ISO 27001 standard

Significance for the ISO 27001 standard	The ISO 27001 standard states that managing risk by implementing security controls as a main goal of the process the standard creates. The standard mentions this already on page 1
Related section(s) of the standard	Section 4.2.1 b states that the ISMS policy has to align with the risk management. Section 4.2.1 c demands a risk assessment that includes criteria for accepting risks and identify the acceptable risk levels. Section 4.2.1 d concerns risk identification and Sect. 4.2.1 e demands risk analysis and evaluation. Section 4.2.1 f concerns risk treatment and Section 4.2.1 g is about controls for risk treatment. Section 4.2.1 h demands management approval for acceptable levels of risk. Risk is also mentioned in several chapters of the Do, Check, and Act phases

- Is an attacker in the misuse case causing harm to himself?
- Do all misuse cases consider at least one cloud component and one stakeholder?

Elicit security requirements We use the information about threats collected in the previous steps and create at least one misuse case for each threat. Afterward the threat is used as a basis for eliciting a security requirement.

According to [24], a security requirement is typically a confidentiality, integrity or availability requirement. It refers to a particular piece of information, the *asset*, that should be protected, and it indicates the *counter-stakeholder* against whom the requirement is directed. A *stakeholder* is an individual, a group, or an organization that has an interest in the system under construction. Furthermore, the *circumstances* of a security requirement describe application conditions of functionality, temporal, spatial aspects, or the social relationships between stakeholders. Hence, circumstances have relations to functional requirements, stakeholders, etc., which shall be considered in the system-to-be. We use the elicited threats as inputs for misuse cases. In this step we also consider the use cases introduced in Sect. 4. These are textual representations of attacker's actions for threat identification. We use them to derive security requirements and check for missing threats. We propose a table as introduced by Deng et al. [26] that lists misuse cases and their corresponding security requirement for this step. In contrast to the work of Deng et al. we do not consider solutions in this step. We discuss them in Sect. 9 in relation to ISO 27001 security controls. We present exemplary misuse cases and security requirements in Table 18.

The elicitation of security requirements concludes our threat analysis. Security requirements are of importance, because they allow a statement if they are fulfilled or not. Hence, if all security requirements of a cloud scenario can be fulfilled, we can state that the security level of a cloud system is sufficient.

The security requirements elicited are for the establishment of an ISMS, which is a process. Several security requirements should lead to implementations in software e.g., the integrity checks of transactional data in the second security requirements (see Table 18). These requirements need to be refined with technical details e.g., the software that conducts the integrity checks.

8 PACTS Step 5: conduct risk assessment

Risk management is mentioned in numerous sections of the ISO 27001 standard. In the approach, risk is used to asses if an asset requires an additional control or not. We provide a

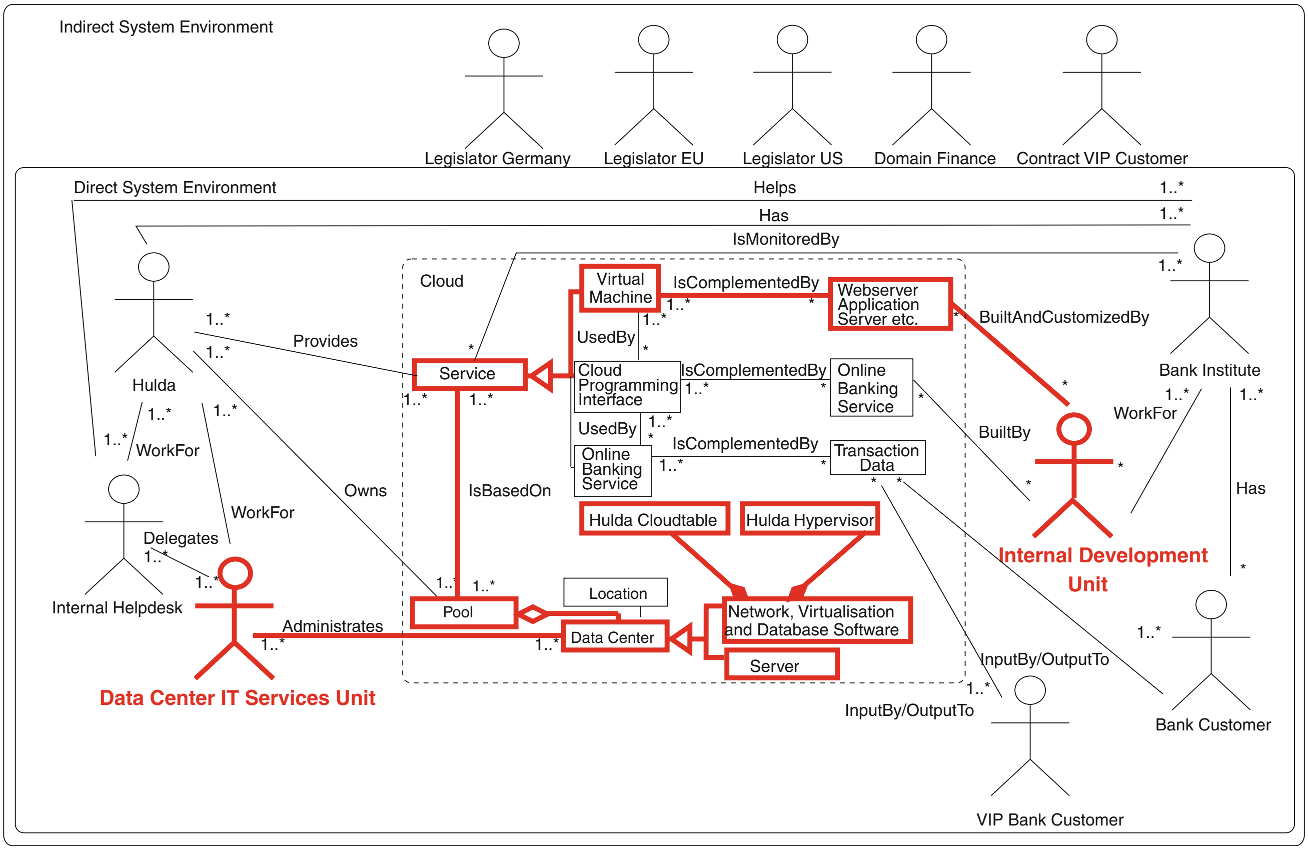


Fig. 18 Pattern-based cloud computing threat analysis for online banking service

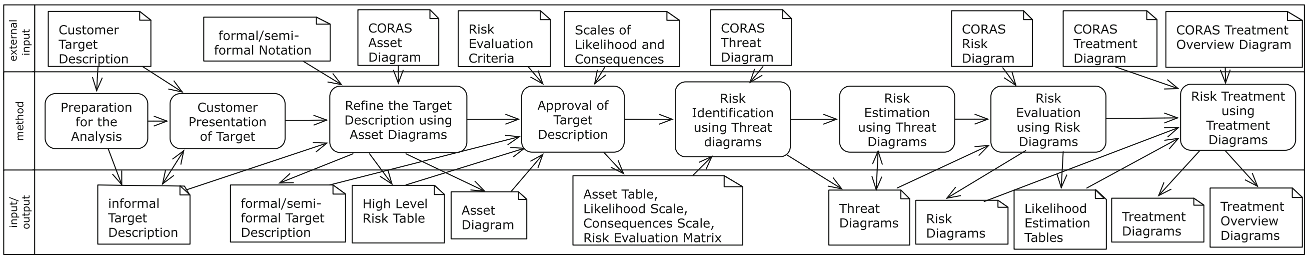

external input
method
input/output
Customer Target Description
formal/semi-formal Notation
CORAS Asset Diagram
Risk Evaluation Criteria
Scales of Likelihood and Consequences
CORAS Threat Diagram
CORAS Risk Diagram
CORAS Treatment Diagram
CORAS Treatment Overview Diagram
Preparation for the Analysis
Customer Presentation of Target
Refine the Target Description using Asset Diagrams
Approval of Target Description
Risk Identification using Threat diagrams
Risk Estimation using Threat Diagrams
Risk Evaluation using Risk Diagrams
Risk Treatment using Treatment Diagrams
informal Target Description
formal/semi-formal Target Description
High Level Risk Table
Asset Diagram
Asset Table, Likelihood Scale, Consequences Scale, Risk Evaluation Matrix
Threat Diagrams
Risk Diagrams
Likelihood Estimation Tables
Treatment Diagrams
Treatment Overview Diagrams

Fig. 19 The CORAS approach

list of every mentioning in the standard in Table 19. In addition, we use the CORAS [27] approach for risk management in our PACTS method, because it is asset-based and we are able to integrate it in our work. Hence, CORAS is background and the integration into our work is the contribution of this section.

The CORAS approach, depicted in Fig. 19, starts with a preparation step. This step considers an initial target description from the customer. The risk experts process the description into an informal description of the target for risk management. This informal description is refined in a presentation of the target to the customer and afterward used as an input for the semi/formal target description. In addition, the risk experts create a CORAS asset diagram and a high-level risk table for these assets. All of these artefacts are presented to the customer and, after approval of the customer, likelihood and consequence scales are defined. The information collected so far is used to create CORAS threat diagrams, which are used for risk identification, risk estimation, and risk evaluation. The results are

Table 20 Qualitative likelihood scale for clouds

Likelihood value	Description
Certain	A high number of similar incidents have been recorded; has been experienced a very high number of times by several users
Likely	A significant number of similar incidents have been recorded; has been experienced a significant number of times by several users
Possible	Several similar incidents on record; has been experienced more than once by the same actor
Unlikely	Only very few similar events on record; has been experienced by few actors
Rare	Never experienced by most actors throughout the total lifetime of the cloud

Table 21 Qualitative consequence scale for the cloud's pool

Consequence	Generic interpretation
Catastrophic	Permanent blackout (e.g., by loss of electricity); can put the cloud provider *Hulda* out of business
Major	Large-scale outage for a short time; can cause significant monetary reparations for the cloud provider *Hulda*
Moderate	Several outages of small scale for a short time; can cause monetary reparations for the cloud provider *Hulda*
Minor	Few outages for a short time; tolerable if easy to recover from and if very rare
Insignificant	A single outage; no impact on operations of the cloud provider *Hulda*

threat diagrams and likelihood estimation tables. In the final step of the analysis, the risk treatment, all the information collected so far is used to create a risk diagram that describes the proposed risk treatments. For simplicity, we do not show a complete walkthrough of the CORAS approach. Instead, we explain how the approach interfaces with our method.

Preparation for the analysis We consider the description of use cases presented in Sect. 4 in this step. These are the initial descriptions of cloud use cases that also describe the target of the risk management.

Customer presentation of target We use our cloud analysis pattern, templates, and business processes as presented in Sect. 5 as customer target description that consider the documentation of the previous step as input.

Refine the target description using asset diagrams The asset identification presented in Sect. 6 is used as an input for the CORAS asset diagrams. Moreover, the high-level security goals presented in Sect. 4 in policy patterns can be input for high-level risks, which are derived from these security goals.

Approval of target description Risk evaluation criteria and likelihood scales have to be defined using CORAS. We explain in the following how to set up likelihood and consequences scales. Risk assessment can be conducted either quantitative or qualitative. Quantitative risk assessment demands that the likelihood and consequences scales contain numeric values. These have to express in which time frame a risk is likely to occur and what the consequences are in, e.g., number of affected assets. Should these numbers not be available, likelihood and consequences tables can contain a qualitative scale that does not contain numbers. This qualitative scale is a starting point for risk assessment. We present an example of a qualitative likelihood scale in Table 20. For each of the direct assets we define a separate consequences table. We provide one example of a consequence scale for the pool considering availability in Table 21.

We present a risk evaluation matrix in Table 22. The red (or dark gray) parts indicate combinations of consequence and likelihood that result in unacceptable risks. The green (or light gray) combinations result in an acceptable risk.

Risk identification using threat diagrams We use our threat analysis presented in Sect. 7 as input for CORAS threat diagrams. These represent all threats to assets.

Table 22 Risk evaluation matrix for the cloud's pool

		Consequence				
		Insignificant	Minor	Moderate	Major	Catastrophic
Frequency	Rare	🟩	🟩	🟩	🟩	🟥
	Unlikely	🟩	🟩	🟩	🟥	🟥
	Possible	🟩	🟩	🟥	🟥	🟥
	Likely	🟩	🟥	🟥	🟥	🟥
	Certain	🟥	🟥	🟥	🟥	🟥

Table 23 Security policy demands of the ISO 27001 standard

Significance for the ISO 27001 standard	The ISO 27001 standard concerns high-level ISMS policies during the establishment of the ISMS to guide the focus of security and security policies as controls that define in detail what a specific security controls should achieve
Related section(s) of the standard	Section 4.2.1 b concerns the definition of ISMS policies

Table 24 Controls of the ISO 27001 standard

Control name	Control objective	Important demands
A.5 Security policy	Provide directions for information security	Documentation and review requirements
A.6 Organization of information security	Manage security within the organization and with external parties	Clear management commitment, responsibilities, coordination, and independent consultation and review
A.7 Asset management	Achieve and ensure appropriate protection levels for assets	Identify assets, assign responsibilities for assets, classify assets, define and document rules for treatment of assets
A.8 Human resources security	Provide security training for employees, communicate responsibilities, provide structured exit procedures	Specify role and terms of employment, define responsibilities and provide security education and training, define disciplinary process, define termination responsibilities, return of assets and removal of rights
A.9 Physical and environmental security	Prevent unauthorized physical access, damage and interference to secure areas and equipment	Establish security parameter, physical controls for access to secure rooms. Equipment shall be protected e.g., from power failure and the support for the equipment shall be ensured e.g., protect cable connection from interference
A.10 Communications and operations management	Ensure secure operations of information processing, especially for service delivery from third parties, ensure availability, integrity, and confidentiality of information processing	Guidelines for processes, e.g., segregation of duties, and specific demand that ensure the goals e.g., back up and monitoring of processes
A.11 Access control	Control the access to information	Ensure access control on information systems, networks, operating systems etc
A.12 Information systems acquisition, development and maintenance	Embed security in information systems and prevent misuse of information	Specific measure are demanded e.g., security requirements analysis, input/output data validation, use of cryptography, prevent information leakage, etc
A.13 Information security incident management	Identify security events and weaknesses associated with information security and provide timely corrective action, ensure a consistent and effective approach	Ensure a reporting for security events and security weaknesses, learn from information security incidents
A.14 Business continuity management	Protect critical business processes from effects of information system failures and ensure their timely resumption	Include security and risk management in the business continuity management process, reassess and test the business continuity plans
A.15 Compliance	Ensure compliance with laws, regulations, contractual obligations, security requirements, organizational security policies, and standards, consider system audits	Identify relevant laws, regulations, contractual obligations, etc. and also data and privacy protection measures, check the compliance to these laws, regulations, contractual obligations, etc. and use also audits to check compliance

Risk estimation using threat diagrams For all threats to assets the risks have to be evaluated using the tables created in the *Approval of Target Description* step of CORAS.

Risk evaluation using risk diagrams The risks are refined in this step in more detail, e.g., considering the relation between harm of different assets. The step also considers discussions with the customer if the assets that have acceptable risks and do not need further risk management.

Risk treatment using treatment diagrams For all assets that have unacceptable risks controls have to be implemented or improved to get these assets to acceptable risk levels. The approach presented in Sect. 9, which instantiates policy patterns and selects appropriate security controls can support this step.

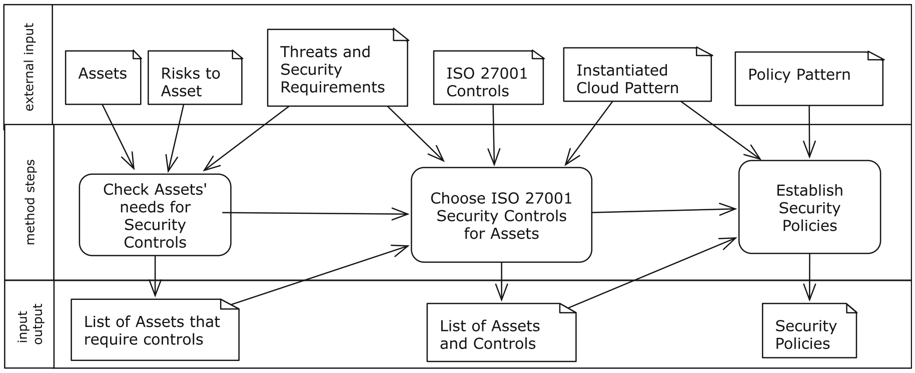

external input
Assets
Risks to Asset
Threats and Security Requirements
ISO 27001 Controls
Instantiated Cloud Pattern
Policy Pattern
method steps
Check Assets' needs for Security Controls
Choose ISO 27001 Security Controls for Assets
Establish Security Policies
input output
List of Assets that require controls
List of Assets and Controls
Security Policies

Fig. 20 A method for establishing ISO 27001 policies

9 PACTS Step 6: create security policies and reason about controls

An important part of assembling an ISMS is to define security policies (see Table 23). They define security goals and state high-level solutions. Security policies are the basis for selecting and defining ISO 27001 controls, which are solutions for security goals.

A security policies has to be in accordance with business requirements and relevant laws and regulations as well as with contractual obligations regarding security. The policy has to be in written form and contain management approval. The policy document is also published and communicated to all stakeholders, which the policy concerns. The policy shall also be reviewed at planned intervals and adapted to changes in the organization and its environment [4, p. 13].

The ISO 27001 standard defines also ISMS policies that are: “considered as a superset of the information security policy. These policies can be described in one document” [4, p. 4]. In particular the ISMS policy contains an alignment with risk management, risk evaluation criteria, and management approval. The security goals and high-level actions have to be aligned with these activities.

We provide a pattern-based method to formulate security policies and select ISO 27001 controls to address the security concerns in the policies. Our approach also addresses changes of policies and an integration with risk management.

9.1 Controls in the ISO 27001 standard

The Annex A of the ISO 27001 standard describes the normative controls of the standard. We present a short overview of these in Table 24. The numbering of the controls starts with A.5 and ends with A.15. The reason for not starting the numbering with A.1 is that the control numbering shall align with the controls listed in the ISO 27002 standard. This standard provides guidelines on how to implement the controls and also further controls, but it is not normative.

9.2 A method for establishing ISO 27001 policies

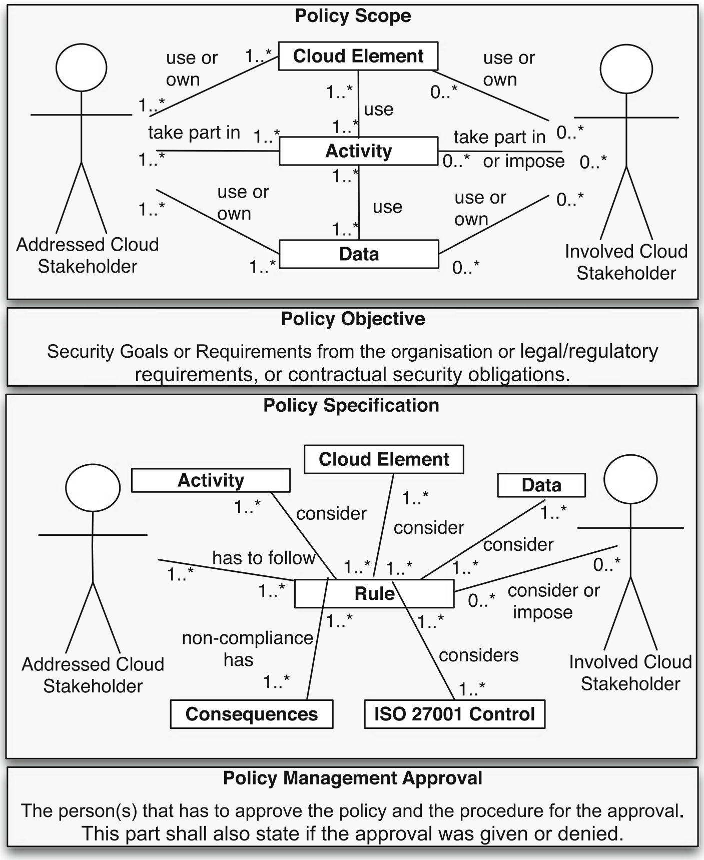
We propose a method for establishing ISO 27001 security policies and reasoning about ISO 27001 security controls (see Fig. 20). In the following, we describe its different steps.

Check assets’ needs for security controls We check the risk levels of each asset in this step. During the risk management, acceptable risk levels were defined. If the risk level of an asset is above this threshold a security control is required to decrease the risk level. The output of this step is a list with all assets that do not have an acceptable risk level.

Choose ISO 27001 security controls for assets Security controls shall reduce the risk that threats can harm assets. We use the list of assets created in the previous step and iterate over all the assets. We have elicited threats and security requirements for assets in Sect. 7. We consider the security requirements for an asset and iterate over them, as well. We reason, which of the ISO 27001 controls (see Table 24) can fulfill a security requirement. In particular, the column *Control Objective* contains relevant terms, which we can try to relate to the security requirement. We also reason why the remaining controls do not fulfill this requirement. This results in a list that explains, which controls are relevant for each security requirement and which are not. We use this list as a ISO 27001 specific document, the so-called *statement of applicability* (*SOA*). The SOA has to contain a reasoning about the selection of ISO 27001 controls.

Establish security policies We specify security policies using our policy pattern (see Fig. 21), which we explain in the following. The pattern describes the structure of a security policy considering ISO 27001 controls and our

Fig. 21 Policy pattern



cloud pattern. The pattern is composed of four main parts, described in the following.

The *Policy Scope* contains at least one *Addressed Cloud Stakeholder*, which is a *Cloud Stakeholder* from the cloud pattern, who is addressed in the policy. The *Addressed Cloud Stakeholder*s *take part in* an *Activity* that *uses Data* or *Cloud Element*s. Data can be any data, while a cloud element refers to an element of the cloud in the cloud system analysis pattern (see Sect. 5). The *Addressed Cloud Stakeholder*s also *use or own Data or Cloud Element*. The pattern can also contain *Involved Cloud Stakeholder*s that can *take part in* or *use or own* a *Data or Cloud Element*. The security expert has to check if a *Cloud Stakeholder* is an *Addressed Cloud Stakeholder* or an *Involved Cloud Stakeholder*. The policy pattern has to be instantiated for each scenario.

The *Policy Objective* states the goal and security requirements of the policy. Security requirements are refinements of security goals [24]. These can originate from organizational goals, laws and regulations or business contracts. These goals are formulated in natural language and shall only refer to the elements of the *Policy Scope*.

The *Policy Specification* addresses an *Addressed Cloud Stakeholder* that *has to follow Rules*. These *Rules* are *consider* at least one *ISO 27001 Control*. The *non-compliance* of these *Rules* has *Consequences*. *Rules* also *consider* at least one *Activity* and *Data* or *Cloud Element*. A *Rule* can also consider *Involved Cloud Stakeholders*. The *Responsible Cloud Stakeholder* is of type *Absorbed Cloud Stakeholder* and the elements *Activity, Cloud Element, Data* and *Involved Cloud Stakeholder* have to occur in the *Policy Scope*.

The *Policy Management Approval* considers the demands of the ISO 27001 standard for management commitment (see e.g., ISO 27001 Sect. 4.2.1 b). The *Policy Management Approval* contains a statement of who is responsible for approving the policy as well as the procedure for approval. The *Policy Management Approval* also contains the date of the approval.

We analyzed the controls in Table 24 for relations to the cloud pattern and in order to ease the identification of possible stakeholders for the instantiation of our policy pattern.

We present the results in Table 25. The column *control name* states the name of each control, the column

Table 25 Controls of the ISO 27001 standard and their relations to the cloud system analysis pattern

Control name	Addressed stakeholder	Action	Cloud element
A.5 Security policy	All	All	All
A.6 Organization of information security	All	Security management activity e.g., clear management commitment	None
A.7 Asset management	Cloud provider, cloud customer, end customer	Activities regarding identify, classify and protect assets	Data, software product, cloud software stack etc
A.8 Human resources security	Cloud Support, Cloud Administrator, Cloud Developer	Activities regarding training, responsibility assignment, designing and implementing exit procedures etc	None
A.9 Physical and environmental security	Cloud provider, cloud administrator	Activities regarding concerning physical access and prevention of damage/interference of hardware	All that are physical e.g., hardware
A.10 Communications and operations management	Cloud provider, cloud administrator, cloud support, cloud developer, cloud customer	Activities regarding Guidelines for processes, e.g., segregation of duties	All
A.11 Access Control	Cloud support, cloud developer	Activities regarding implement and monitor access to information	All that are software
A.12 Information systems acquisition, development and maintenance	Cloud administrator, cloud software developer	Activities regarding eliciting of security requirements and vulnerability detection e.g., penetration testing and specific measures e.g., cryptography	All that are software
A.13 Information security incident management	Cloud administrator, cloud support, cloud developer	Activities regarding reporting security events and issues, ensuring a consistent and effective response, learning from security incidents,	All
A.14 Business continuity management	Cloud administrator, cloud support, cloud developer	Activities regarding business continuity management for business processes e.g., security and risk management	All
A.15 Compliance	Cloud administrator, cloud support, cloud developer cloud provider, cloud customer, cloud end customer	Activities regarding identifying laws, regulations and contractual obligations. Privacy protection, monitor compliance to the laws regulations and contractual obligations, compliance audits	All

addressed stakeholder lists possible stakeholder as candidates from the cloud system analysis pattern for *addressed stakeholders* in our policy pattern. The column *action* lists activities related to a control. These shall support the selection of controls. The column *cloud element* lists potentially relevant cloud elements for these controls.

The policy patterns ease the effort of writing policies. For example, the provided elements in the patterns help not to miss an important element of a policy. Moreover, several parts of the policy pattern use elements of the cloud system analysis pattern and descriptions created in previous sections. This should improve the consistency between the different documents for the ISO 27001 standard.

Moreover, the instantiated patterns are a vital part of the ISMS specification, because they define the decisions for the protection of each asset within the cloud scenario. They are also the basis for refining high-level ISO 27001 controls into concrete security solutions.

9.3 Application of our ISO 27001 policy method to our running example

Check assets' needs for security controls We have conducted the risk analysis of the asset *Hulda Hypervisor* and the resulting risk level for the *Hulda Hypervisor* is unacceptable. In our example, the unacceptable risk level is caused by the *Shared Technology* threat (see Sect. 7).

We consider the following security requirement in our example, which addresses this threat: The configuration data of the *Hulda Hypervisor* has to be protected from unauthorized changes.

Choose ISO 27001 security controls for assets We consider all the controls listed in Table 24 for our security requirement. We argue that all controls are relevant and finish this step (see Table 26). If we would have found out that a control is not relevant a solid argument would have to be presented why this is the case.

Table 26 Control reasoning for the ISO 27001 standard for our running example

Control name	Control reasoning
A.5 Security policy	We require a clearly defined policy for access to the *Hulda Hypervisor*' configuration data. We address this control by instantiating a policy pattern
A.6 Organization of information security	Organizational demands for defining processes of how change requests regarding the*Hulda Hypervisor* have to be satisfied. For example, an end customer might wish to join two online banking accounts and the hypervisor shall merge the related resources. In this case, the end customer needs a way to request these changes
A.7 Asset management	We have already addressed this control in Sect. 6 and the asset *Hulda Hypervisor* is documented and responsibilities have been assigned to it
A.8 Human resources security	We require security training for the *Data Center IT Services Unit*, which should be the only stakeholder allowed to change the configuration data of the*Hulda Hypervisor*. The *Data Center IT Services Unit* has to be trained in the access control mechanisms of the *Hulda Hypervisor* and how to communicate with other cloud stakeholders about change requests
A.9 Physical and environmental security	The *Hulda Hypervisor* is software, but the software runs on a physical hardware server. We have to protect this server and need a control for it. For example, the server room shall be in a closed room with an emergency power supply. The emergency power supply shall keep the servers working during power shortages. In addition, the room has to be locked in order to prevent unauthorized access to the server
A.10 Communications and operations management	We have to define processes for accessing the *Hulda Hypervisor* as well as maintaining and repairing it
A.11 access control	The used mechanism to ensure access control have to be chosen and described in detail. For example the mechanisms defined in the ANSI standard for role based access control. [28] could be used for this purpose. Moreover, for clouds the XACML [29] standard for extensible and xml-based access control can be relevant
A.12 Information systems acquisition, development and maintenance	We use this control to check our chosen security mechanisms for access control, e.g., via penetration testing [30]
A.13 Information security incident management	For each possible security incident a process has to be defined, so that the *Data Center IT Services Unit* can react accordingly. The possible incidents can be derived from the threats and security requirements elicited in Sect. 7
A.14 Business continuity management	The selection of the cloud provider *Hulda* was done carefully by a bidding process among different providers. The availability values and response times for incidents were compared with multiple providers. *Hulda* offered the best values and could convince with a qualified 24 × 7 support team in place, which the customer can call, when there is a major incident or outage. Furthermore, *Hulda* could provide disaster recovery on the hardware side, e.g., VMware HA [31] and VMware Vmotion [32].*Hulda* also provided proof of the existence of a fallback data center
A.15 Compliance	We have to identify relevant laws and regulations using the method presented in Sect. 11

Establish security policies We instantiate our policy patterns (see Fig. 21) for each of the security requirements identified during the threat analysis. We show an example instantiation of our policy pattern in Fig. 22. We focus on the *Data Center IT Services Unit*. In this policy the involved stakeholders are the *Bank Customer* and the *VIP Bank Customer*. The policy addresses the requirement that the configuration of the *Hulda Hypervisor* has to be protected. The policy specification states that only the *Data Center IT Services Unit* is allowed to change the configuration of the *Hulda Hypervisor*. In our example, a re-configuration of the *Hulda Hypervisor* is done on behalf of the *Bank Customer* and the *VIP Bank Customer*. Our example policy considers the control A5 in particular, therefore, it is printed in bold. The other controls have to be implemented in subsequent steps. The policy was approved by Mr. Jones, who is responsible for establishing the ISMS (see Sect. 4).

9.4 Consistency checks

We propose to check instantiated policy pattern via several validation conditions. We identified the following so far:

- All cloud stakeholders that are referred to in the policy pattern have to occur in the cloud pattern.
- All controls in the policy pattern have to occur in ISO 27001 Annex A or a reason has to be given why none of them is applicable.
- All cloud elements that are referred to in the policy pattern have to occur in the cloud pattern.

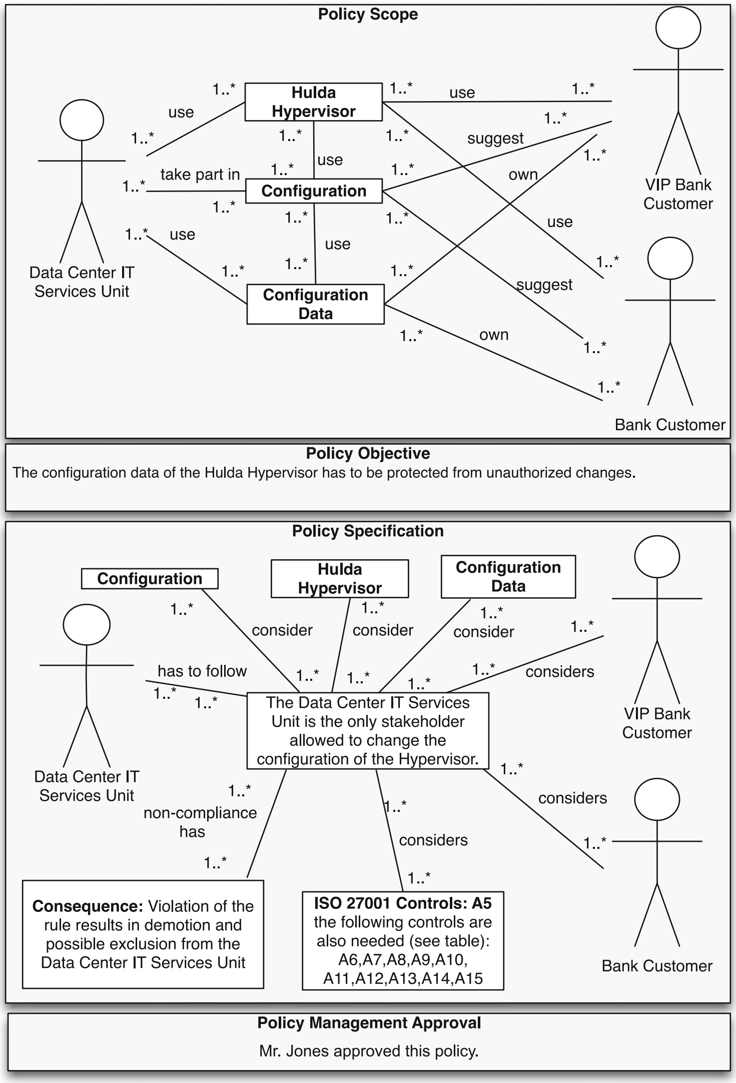

Policy Scope
Hulda Hypervisor
Configuration
Configuration Data
Data Center IT Services Unit
VIP Bank Customer
Bank Customer
use
take part in
use
use
use
use
suggest
suggest
own
own
use
1..*
Policy Objective
The configuration data of the Hulda Hypervisor has to be protected from unauthorized changes.
Policy Specification
Configuration
Hulda Hypervisor
Configuration Data
consider
consider
consider
considers
considers
considers
has to follow
non-compliance has
The Data Center IT Services Unit is the only stakeholder allowed to change the configuration of the Hypervisor.
Consequence: Violation of the rule results in demotion and possible exclusion from the Data Center IT Services Unit
ISO 27001 Controls: A5 the following controls are also needed (see table): A6,A7,A8,A9,A10, A11,A12,A13,A14,A15
Data Center IT Services Unit
VIP Bank Customer
Bank Customer
1..*
Policy Management Approval
Mr. Jones approved this policy.

Fig. 22 Example instance of our policy pattern

- Each policy has to refer to at least one asset and the asset has to be referred to in at least one threat.
- The policy scope, policy objective, and policy specification have to refer to the same asset and the same stakeholders.

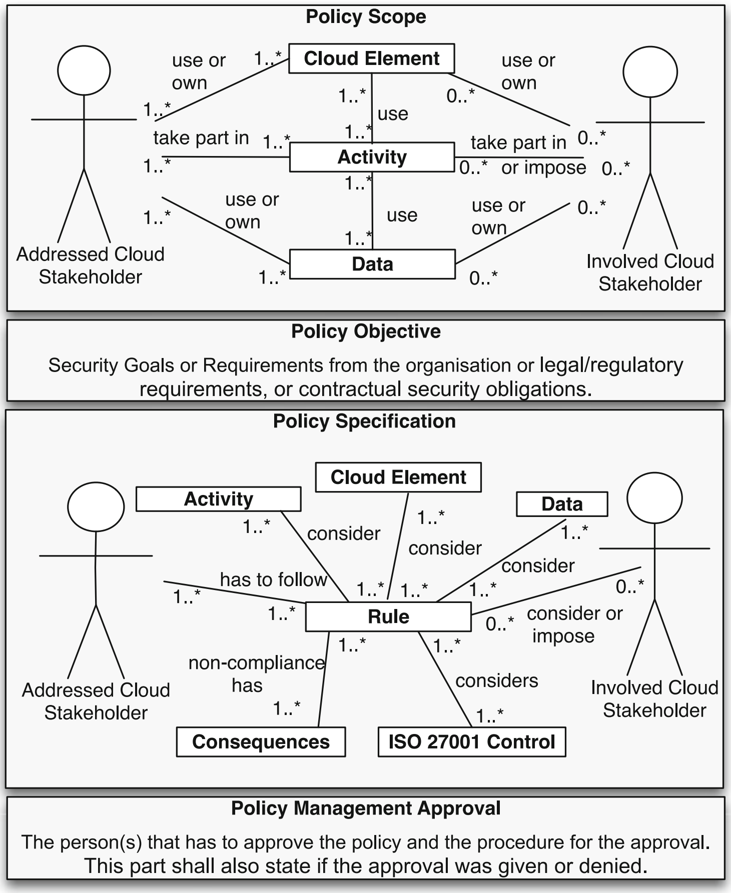
9.5 Policy change pattern

We also present our policy change pattern (see Fig. 23), which allows to propose changes for an existing policy pattern. The change pattern has several parts. The *Change Scope* describes the scope of the policy after the change. *Reasons for the Change* define the security goals or requirements that cause the change. The *Policy Specification* describes the security policy specification after the change and the *Change Detail* describes the difference between the new and the old policy. The *Change Approval* states who has to approve the change and how it has to be approved.

10 PACTS Step 7: design isms specification

The ISO 27001 standard demands a documentation of the ISMS (see Table 27). Sect. 4.3.1 demands a documentation of several parts of the ISMS. A list is provided in the column *ISO 27001 Documentation* of Table 28. These “documents and records may be in any form or type of medium.” [4, p. 8]. Hence, we propose a mapping in Table 28 of the generated artifacts from our method to the documentation demands of the ISO 27001 standard. The column *Support from our Method* states the artifact that

Fig. 23 Policy change pattern



Security Goals or Requirements from the organisation or legal/regulatory requirements, or contractual security obligations.

The person(s) that has to approve the policy and the procedure for the approval. This part shall also state if the approval was given or denied.

Table 27 Documentation demands of the ISO 27001 standard

Significance for the ISO 27001 standard	The ISO 27001 standard requires documentation in any form or medium to ensure the ISMS produces satisfying results
Related section(s) of the standard	Section 4.3 lists documentation demands of the standard. The section contains subsections for general documentation concerns, control of documents and control of records. Records provide evidence and conformity with the ISMS requirements and proof of the effective operation of the ISMS

Table 28 Support of our method for ISO 27001 documentation demands

ISO 27001 documentation	Support from our method	Sections
ISMS policies and objectives	Management approval and policy pattern	4
Scope and boundaries of the ISMS	Cloud system analysis pattern	5
Procedures and controls	Documentation of security controls	9
The risk assessment methodology	Our risk methodology	8
Risk assessment report	Asset identification, threat analysis, and risk assessment	6, 7, 8
Risk treatment plan	Risk assessment and control selection	8, 9
Information security procedures	Control documentation and policy pattern	9
Control and protection of records	Security solution concerning the control *A.10.7.4 security of system documentation*	9
Statement of applicability	Reasoning about controls	9

Table 29 Legal compliance in the ISO 27001 Standard

Significance for the ISO 27001 standard	The consideration of legal obligations is already mentioned in the first paragraph on page 1 of the standard. This states that the compliance to the standard does not confer immunity for legal obligation. Thus, legal obligations have to be known in order to be able to follow this demand
Related section(s) of the standard	Sect. 4.2.1 b concerns the definition of an ISMS policy that includes the consideration of legal compliance. Section 4.2.1 g concerns the selection of controls, which demand also legal and regulatory compliance

relates to a specific part of the ISMS. The column *Sections* states the section of our work that describes how to create the artifacts mentioned in the column *Support from our Method*.

We describe our mapping in the following (see Table 28). We use our policy pattern and the attached management commitment templates to document the ISMS policies and objectives. The scope and boundaries of the ISMS are documented using our cloud system analysis pattern and the procedures and controls are documented as part of our chosen security controls. The risk methodology is our described risk approach and the risk assessment report uses the asset identification, threat analysis, and risk analysis approaches. The risk treatment plan contains the risk estimation for each asset and the established controls to reduce the risk to acceptable levels.

The information security procedures are documented using our policy pattern and the attached tables. The control and protection of records is the documentation of the selected security solution for the control *A.10.7.4 Security of system documentation*, which describes the protection of the ISMS documentation against unauthorized access. This control has to be referenced in one of our policy pattern and a solution has to be implemented for it.

The *statement of applicability* defines the implemented controls for each asset. It also contains a reasoning for the controls not selected for this asset (see Sect. 9). The results of the legal compliance (see Sect. 11) analysis and privacy analysis (see Sect. 12) are of cross-concern and considered in all documents.

11 Considering legal compliance in the PACTS method

The ISO 27001 standard mentions the importance of considering law already on page 1 of the standard. We listed all sections that demand legal consideration in Table 29. The section on how to establish an ISMS explicitly states that legal obligations have to be considered during the definition of the ISMS policy and the selection of controls. The standard also has an explicit control for legal compliance. The controls for security policy and human resource security state the explicit consideration of laws when applying the control. In a previous work, we developed a pattern-based approach for identifying relevant laws for a software engineering project [33].We presented the *law pattern* for structuring laws. We structured requirements in a similar pattern, the *law identification pattern*. This allowed us to use a matching algorithm from requirements to laws, because requirements and laws were stored in a similar structure. Subsequently, we integrated this work with our cloud pattern. We used the information in the pattern to instantiate the law identification pattern. This work also contains a detailed description of how to map laws to requirements as part of a structured method for identifying relevant laws [34].

We conducted a literature review regarding clouds and laws. Sect. 11.1 contains the results of this review and shows in particular the difficulty of identifying relevant laws. We describe how our law identification process works in Sect. 11.2 We also propose a novel modification of the law identification process, which integrates our law identification method with our PACTS method. In addition, we also contribute tool support[13] for our law method. The tool allows to instantiate law patterns, law identification patterns, and supports the matching between these patterns. The pattern instantiation is possible in

In Sect. 11.3 our novel process is applied to the running example. Section 11.4 describes how to derive controls from the results of the law identification process.

11.1 Overview on compliance issues of clouds

A PriceWaterhouseCoopers study from 2010[14] reveals that identifying compliance requirements is a significant challenge for compliance management in clouds. Compliance requirements are requirements derived from relevant laws or regulations. In particular, we identified the following cloud compliance issues:

Identification of relevant Laws The identification of relevant laws and the elicitation of legal requirements for a software system is essential in order to be *compliant*. This

[13] http://www.uml4pf.org/law/lawtool.html.

[14] http://www.pwc.de/en/prozessoptimierung/trotz-einiger-bedenken-der-virtuellen-datenverarbeitung-gehoert-die-zukunft.jhtml.

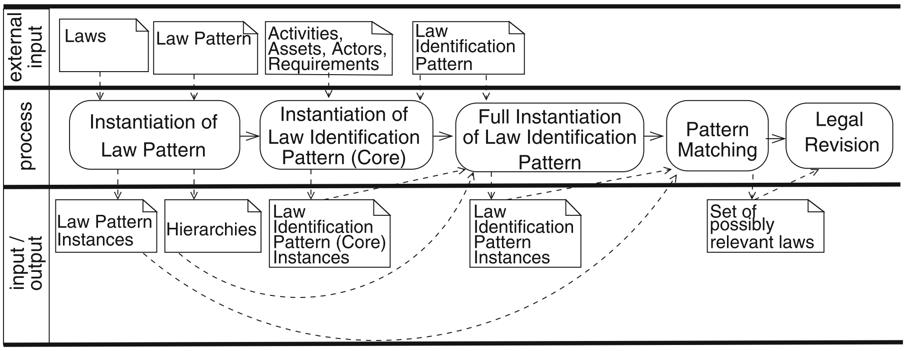

external input
process
input / output
Laws
Law Pattern
Activities, Assets, Actors, Requirements
Law Identification Pattern
Instantiation of Law Pattern
Instantiation of Law Identification Pattern (Core)
Full Instantiation of Law Identification Pattern
Pattern Matching
Legal Revision
Law Pattern Instances
Hierarchies
Law Identification Pattern (Core) Instances
Law Identification Pattern Instances
Set of possibly relevant laws

Fig. 24 Law identification process (cf. [34])

is considered to be difficult, because it is a cross-disciplinary task in laws and software and systems engineering [35]. This task has a significant complexity for clouds, due to the amount of different stakeholders, functionalities and locations a high number of different laws have to be considered.

Data location and deletion Cloud providers often are not able to provide detailed information on the location of their customers' data [19]. This is relevant e.g., to obey privacy laws, which we explain at the end of this section. In some case the information is available, but not disclosed to the cloud customer [19]. Another open question is how a cloud provider can prove that data have been deleted [21], which is also relevant for compliance to privacy laws.

Choosing a legislation Cloud providers and customers are often located in different countries. In this case, the laws of the cloud provider's country are usually relevant. However, cloud providers and customers can agree on using the laws of one country for their cloud business. Furthermore, contracts have to fill the gap between the chosen law and the law that is not chosen [36]. The chosen law is binding to all cloud stakeholders. Thus, understanding the law of the different countries and making an informed choice is essential and presents a significant challenge.

Contractual obligations Contracts are also used to define the ramifications of violations of the clouds' SLAs. In addition, contracts have to fill the gap between the agreed law for a cloud system and national law of the cloud stakeholders [21]. Detecting these gaps completely is a challenge, because of the lack of methods to support this task.

Subcontractor issues The previous issues multiply in complexity, when the cloud provider can use subcontractors, e.g., from another country. Moreover, it is hardly possible for cloud customers to detect that their data have been processed by a third party [21].

Audibility The use of distributed computing environments, spread all over the globe, provides a challenge for auditing demands [21].

The cloud pattern can be used to elicit relevant information for *Contractual Obligations, Subcontractor Issues, Audibility, Data Location and Deletion*, because the relevant direct and indirect cloud stakeholders and cloud elements including their location are part of our extended cloud pattern. Moreover, our pattern is extensible and further attributes, stakeholders or cloud elements can be included with little effort.

In this work, we focus on identifying relevant laws and regulations that affect IT risk management and privacy. Thus, we tackle in particular the *Identification of relevant Laws* issue. The *Choosing a Legislation* can also be supported by our main focus, because one can identify relevant laws for a cloud project using our method. The resulting requirements from these laws can provide the means for an informed decision regarding a legislation.

In our running example, we chose the German law as the binding law. However, we believe that our law identification and analysis method is also valid for laws of other nations. In order to give an idea of the number of laws, regulations, and standards, that would have to be considered, we present the following list, which could be extended even further:

> Law on Monitoring and Transparency in Businesses (KonTraG), Stock Corporation Act (AktG), German banking act (KWG), Securities Trading Act (WpHG), Minimum Requirements for Risk Management (MaRisk), Commercial Code (HGB), Tax Code (AO), State Data Protection Acts (LDSG), Telemedia Act (TMG), Federal Data Protection Act (BDSG).

From this short survey one can recognize, that even for our small running example, a huge number of laws might become relevant. This fact emphasizes the need for an engineering method for the identification of relevant laws and their analysis. For simplicity, we focus in our running

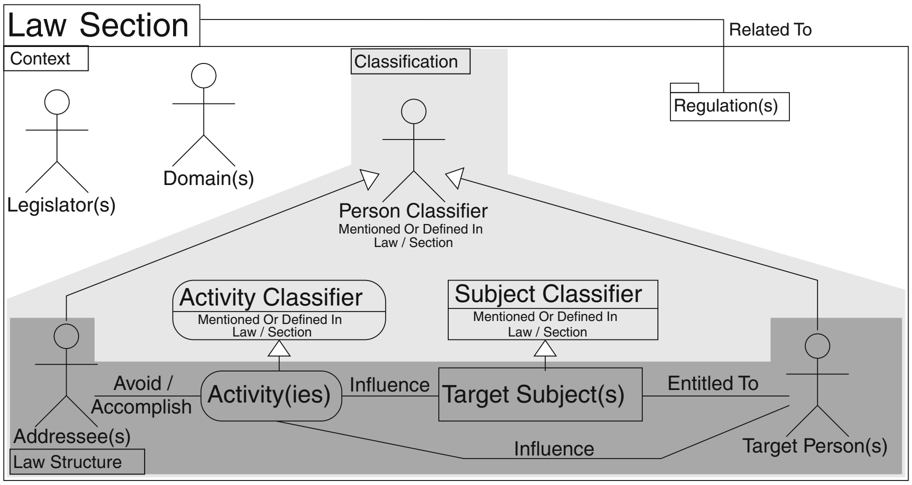


Fig. 25 Law pattern (cf. [34])

example on relevant compliance regulations for privacy. We only explain the laws and regulations that we use in the example.

In 1995, the European Union (EU) adopted the *Directive 95/46/EC* on the processing of personal data that represent the minimum privacy standards that have to be included in every national law. Germany implements the European Privacy Directive in the *Federal Data Protection Act (BDSG)*. According to *Section* 1 *BDSG* all private and public bodies that automatically process, store, and use personal data have to comply with the BDSG. IT systems have increased the feasibility of unwanted disclosure, because storage capacity and speed of computers allow to store, search and correlate data. *Section* 9 *Sentence 1 BDSG* states different requirements that have to be fulfilled by technical and organizational measurements for protecting personal data, e.g., physical and virtual access control to data and the separation of storing and processing data collected for different purposes. Furthermore, it must be verifiable whether personal data have been deleted and by whom and that data have only been processed with the permission of the customer.

Moreover, the EU law as well as *Section* 4*b BDSG* forbid sharing data with companies or governments in countries that have weaker privacy laws. For exchange with companies in the United States (US), there exists the *Safe Harbour* agreement. But under the *US Patriot Act*, officials could access information about citizens of other countries, if that information is physically located within the US or accessible electronically. The priority of the Patriot Act has never been explicitly tested in court, but is a risk for bringing privacy-critical data into the cloud where data centers can be technically distributed world-wide. As cloud computing is considered as contracted data processing, the cloud customer is responsible to adhere to the complete BDSG, according to *Section* 11 *BDSG*. The law further defines the contract between customer and outsourcing provider. For example after ending the contract all data have to be deleted.

11.2 PACTS Step 8: identify relevant laws and regulations

In the context of our PACTS method, the overall requirements engineering process (see [34]) is not of central relevance. Hence, we present here only a brief description of the steps of our law identification method (Fig. 24).

Instantiation of law pattern We propose to store laws in a specific structure, which helps us to search for them. The structure is our so-called *law pattern*. The structure is derived from legal literature and discussed in detail in [33].

Figure 25 shows our law pattern. The pattern consists of three parts and to better distinguish them, we color-coded them: the dark gray part represents the *Law Structure*, the light gray part depicts the *Classification* to consider the specialization of the elements contained in the *Law Structure* in related laws or sections, and the white part considers the *Context*.

Our pattern contains three parts. The first is the law structure comprised of the following elements. An *activity* describes actions that an addressee has to follow or avoid to be compliant. A *target subject* describes impersonal subjects that are objectives of an activity. Subjects can be material, such as a product, or immaterial, such as information. A *target person* is directly influenced by an activity of an addressee or has a relation to a target subject. A *legal consequence* defines the consequence for an addressee, e.g., the punishment when violating the law.

The second part is the so-called *classification*. We organize some elements of the pattern in hierarchies, because laws usually address kinds of activities or persons. In order to determine the kind of a specific person or

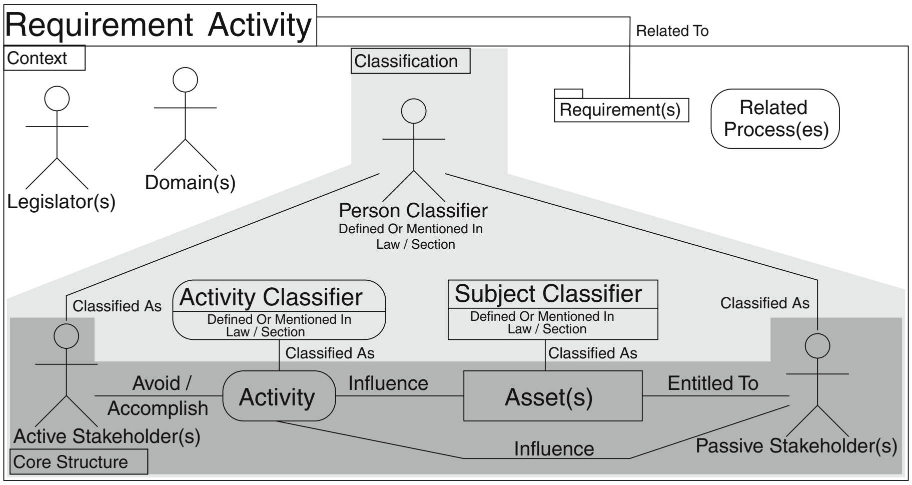


Fig. 26 Law identification pattern (cf. [34])

activity, we consider hierarchies. In particular, *Person Classifier, Activity Classifier*, and *Subject Classifier* use *hierarchies*.

The *Context* part of the law pattern contains the *Legislator(s)* defining the jurisdiction, and the *Domain(s)* clarifying for which domain the law was established.

Instantiation of law identification pattern (core) We consider requirements, activities, and assets for the instantiation of our law pattern. Important besides the functional requirements, activities, and assets themselves, are the relations between them and other already elicited information. First of all, a *Requirement* can be related to other *Requirements* and dictates a certain behavior. A behavior can be a certain *Activity* or a whole *Process*. A *Process* consists of different *Activities*. An *Activity* involves an *Active Stakeholder* and in some cases an *Asset*. Additionally, an *Activity* influences a *Passive Stakeholder* either in a direct way or indirect through an *Assets*. In addition, *Assets* can be related to each other, e.g., one *Asset* is part of an other *Asset*. All these relations have to be discovered and documented, as well.

Figure 26 shows an instance of our law identification pattern. The structure is similar to the law pattern in Fig. 25 to allow a matching of instances of both patterns. In contrast to the legal vocabulary used in the *Law Structure* of our law pattern, the wording for the elements in the dark gray colored *Core Structure* of our law identification pattern is based on terms known from requirements engineering. For example, the element *Asset(s)* in our law identification pattern represents the element *Target Subject(s)* in our law pattern. In this step, we instantiate the core structure of our law identification pattern, which is the dark gray area in Fig. 26. The core structure can be instantiated by software engineers alone.

Full instantiation of law identification pattern In this step, the terms and notions of the software requirements have to be mapped to legal terms. In practical terms, we instantiate the light gray area in Fig. 26. This results in a classification of the software engineering terms to legal terms. In this step, legal experts and software engineers have to collaborate to bridge the gap between technical and legal wording.

Pattern matching In this step, we conduct a matching between the instances of our law pattern and our law identification pattern. This results in a set of laws, which might be of relevance for the software. The resulting laws are only possibly relevant, because we use an overestimation in our method.

Law revision In this last step, a lawyer decides, which of the identified laws are relevant.

The relevant laws are then mapped to requirements as we proposed in [34]. The mapping checks if a requirement contains the elements of a security requirement namely stakeholders, assets, counter-stakeholder, and circumstances. If all these elements exist, we translate the law into a security requirement. If this is not the case, we have to determine if the law can be translated into another kind of quality requirement or a functional requirement. At last, a conflict analysis should be conducted and the development of a consolidated set of all types of requirements as proposed by [37].

The original law identification method is designed to be embedded into a full requirements engineering process (see Sect. 11.2). At this level using this method results in a detailed set of relevant law sections and their relation to requirements. Hence, the impact of laws on the system-to-be can be easily derived and reflected. But, there are scenarios in which using an RE process is not feasible and, hence, detailed requirements are not available. For example, when only a preliminary ISO 27001 complaint ISMS documentation is the aim or legacy systems are part of the ISMS.

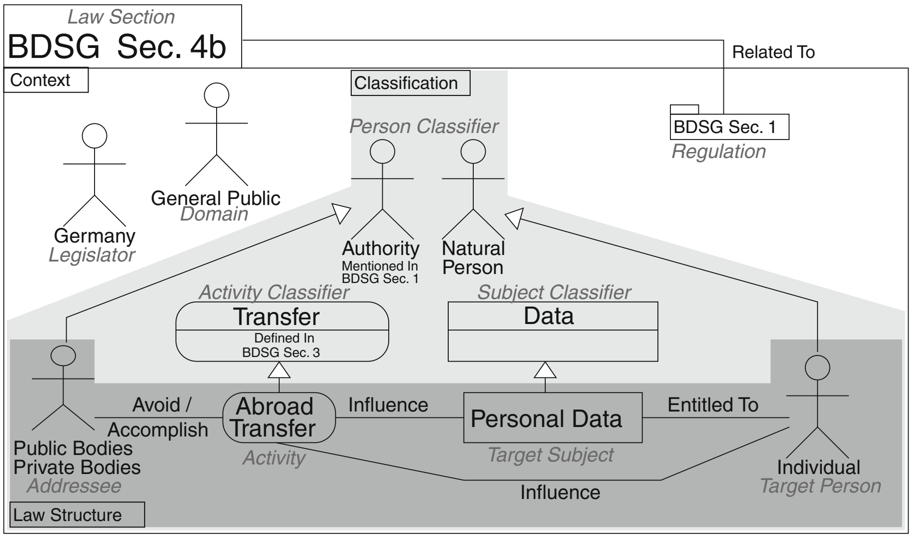


Fig. 27 BDSG Sect. 4b (cf. [34])

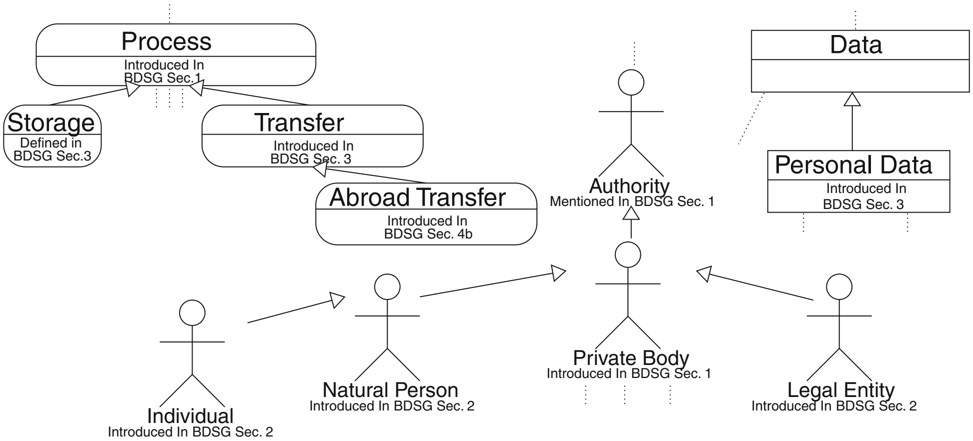


Fig. 28 Hierarchies for person (*bottom*), subject (*upper right*), and activity (*upper left*) (cf. [34])

In this section, we present a modified version of our law identification method that relies on the artifacts created in our PACTS method. Note, that the only step modified is the *Instantiation of Law Identification Pattern (Core)*, which we describe in the following. The remaining steps have to be executed as described in Sect. 11.2.

Instantiation of law identification pattern (core) We instantiate the core part of our law identification pattern, which is the dark gray area in Fig. 26. We use the instances of the following artifacts of our PACTS method for the instantiation of the core part of our law identification pattern:

- our cloud pattern (see Sect. 5)
- our cloud asset table (see Sect. 6)
- the modeled processes (see Sects. 5 and 6)

We consider all direct stakeholders of our cloud pattern and select one at a time. The next part is an iteration over each of their assets by considering our asset templates. We select all processes in which an asset appears, as well. For each process, we select those activities that are related to the asset and also executed by the selected stakeholder. We use this information to instantiate the core part of a law identification pattern for each activity.

We use the following mapping between elements of our law identification pattern and the artifacts of our PACTS method:

Active stakeholder The active stakeholder of core structure corresponds to the selected direct stakeholder.
Activity The activity of the core structure corresponds to the currently selected activity.
Asset The asset of the core structure corresponds to the currently selected asset.
Passive stakeholder The passive stakeholder of the core structure corresponds to the asset provider of the currently selected asset.

11.3 Example

We present an example of the method described in Sect. 11.2. We apply the method to our running example in the following.

Instantiation of law pattern We illustrate our approach by an example based on Sect. 4b of the German *Federal Data Protection Act* (*BDSG*). Thus, we fill our database with all laws of the BDSG in order to be able to discover dependent laws. Then, we use our *law pattern*. The resulting law pattern instance is depicted in Fig. 27. The light gray words close to an element of an instance refer to the type of the element in the original pattern. We consider the transfer of data outside of Europe. For example in Fig. 27 the light gray words *Activity Classifier* near *Abroad Transfer* indicate that the *Abroad Transfer* element is an instantiation of the *Activity Classifier* element in the original pattern.

We instantiate BDSG Sect. 4b, which refers to further sections of the BDSG, e.g., *BDSG Sect. 1*, which we also instantiate. This is noted in the *Context* part of our pattern (white area in Fig. 27). The *Legislator(s)* and *Domain(s)* can be instantiated according to the considered legislators (e.g., *Germany* and *General Public* in the *Context* part.). We instantiate *Activity* with *Abroad Transfer*. *Addressee, Target Subject*, and *Target Person* are instantiated using the related Sect. 1 BDSG. Finally, we instantiate the *Classification* part (light gray area in Fig. 27). We depict the hierarchies for the law in Fig. 28, which is discussed in detail in [34]. This hierarchies show, e.g., that a *Natural Person* is a specialization of *Authority*. The hierarchies in Fig. 28 are updated with *Transfer*, defined in Sect. 3 BDSG, with a specialization *Abroad Transfer*.

Instantiation of law identification pattern (*Core*) After instantiating the BDSG law, we consider the information of our ISMS. For our example, we select *Hulda* as *direct stakeholder* from the cloud analysis pattern instance (Fig. 6). For Hulda, we select the corresponding *asset template instances*, which are described in Sect. 6. From these instances, we select the *Transaction Data* entry (Table 30). From the entry we see that the *asset* is *Transaction Data*. For this asset, we select *VIP Bank Customer, and Bank Customer* as *Asset Provider* from the cloud asset table (Table 31). The related *process* is described as activity diagram (Fig. 14). This process describes several *activities*, which are executed by the cloud offered by *Hulda*, and which are related to the transaction data. One of those activities is *Store Data Distributed*. We select this activity and instantiate the law identification pattern core (Fig. 29).

Table 30 Asset template instance (excerpt from Table 12)

Asset	Asset owner	Asset use	Asset class	Asset label
…				
Transaction data	Mr. Jones	See Fig. 14	Data	AS_DA_120
…				

Considering our mapping from Sect. 11.2, we instantiate the *Active Stakeholder* with our direct stakeholder *Hulda*. For the *Activity* we instantiate *Store Data Distributed*. The *Asset* is the *Transaction Data*. And the *Passive Stakeholder* are our asset provider *VIP Bank Customer* and *Bank Customer*.

Full instantiation of law identification pattern We use processes, activities and assets documented as described in Sect. 11.2 for fully instantiating the law identification pattern. Figure 29 presents an example. We instantiate the legislators *Germany, US, EU*, the domain *Finance*, as well as the process *Offering Transaction Data Processing*. The activity *Store Data Distributed* is classified as *Abroad Transfer* and *Storage, Transaction Data* are classified as *Personal Data*, (*VIP*) *Bank Customer* is classified as *Individual*, and *Hulda* is classified as *Legal Entity* based on a discussion between the legal experts and software engineers.

Pattern matching The matching of the classification area of the law identification pattern instance (light gray area) and the law structure and classification part of the law pattern instance (light and dark gray areas) reveals relevant laws. This law identification uses the previously documented hierarchies. The matching is successful for *Abroad Transfer, Personal Data*, and *Individual*. Hence, we have a match between our law identification pattern instance shown in Fig. 29 and our law pattern instance of Section 4b BDSG, depicted in Fig. 27. The matching for *Hulda*, who is classified as *Legal Entity* uses the hierarchy depicted in

Table 31 Cloud asset table (excerpt from Table 11) Instance

Asset	Asset provider	Asset reasoning
…		
Transaction data	VIP bank customer, bank customer	Harm to the asset can possibly cause financial loss and privacy violation to the *VIP Bank Customer, Bank Customer*. The harm depends on the kind of data
…		

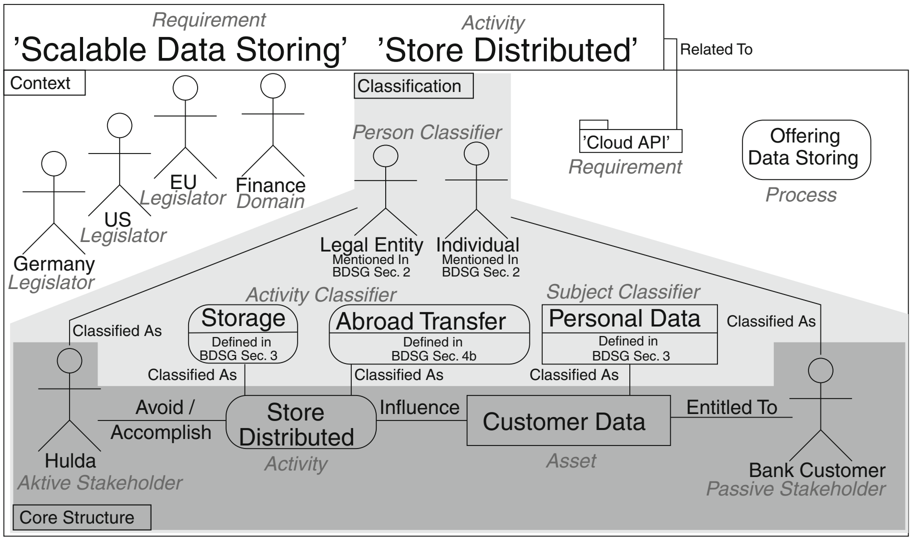


Fig. 29 Law identification pattern instance

Table 32 Privacy in the ISO 27001 standard

Significance for the ISO 27001 standard	Relation to Sect. 4.2.1 of the plan phase and several phases of the Do, Check, Act phases of the standard according to ANNEX B
Related section(s) of the standard	ANNEX B—OECD principles

Fig. 28. This reveals that *Legal Entity* is a specialization of *Private Bodies*, which results in identifying Section 4b BDSG as relevant.

Law revision The resulting set of relevant laws has to be validated by lawyers. The lawyers confirm that the BDSG is indeed relevant for the scenario.

11.4 PACTS Step 9: define compliance controls

We use the results of our PACTS step 8 to define ISO 27001 controls. The control *A.15.1 Compliance with Legal Requirements*. Hence, we formulate legal requirements and add these to the set of controls of the ISMS. We have to check if a legal requirement also has a relation to other controls using our ISO 27001 control overview (see Sect. 9 and in particular Table 24). For instance, the appendix to BDSG Sect. 9 demands specific methods, e.g., access control, which has a relation to control *A.11 Access Control*. Thus, a particular instance of this control also has to be added to the set of selected ISMS controls.

12 Considering privacy in the pacts method

The ISO 27001 considers privacy in ANNEX B (see Table 32), which, in turn, refers to Sect. 4.2.1 of the plan phase of the standard. ANNEX B references *The Fair Information Practice Principles* (—or short FIPs) of the OECD [38]. These are widely accepted privacy regulations. They state that a person's informed consent is required for the data that are collected, collection should be limited for the task it is required for and erased as soon as this is not the case anymore. The collector of the data shall keep the data secure and shall be held accountable for any violation of these principles.

In previous work, we developed textual patterns to formulate privacy requirements [39] and proposed a computer-aided privacy threat analysis [40]. Both methods are based on problem frames, which we introduce in Sect. 12.1. We combine our previous work into one structured method for privacy requirements and threat analysis. We also adapt this method to integrate it with our PACTS method in Sect. 12.2. Section 12.5 contains an example of the privacy steps of our PACTS method.

12.1 Overview on problem frames

We use a requirements engineering method inspired by Jackson [10]. Requirements can only be guaranteed for a

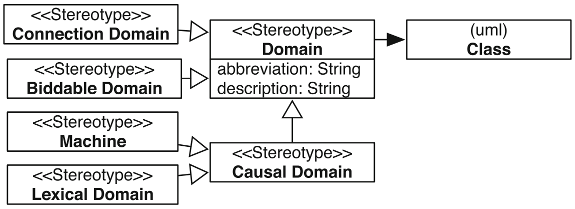


Fig. 30 Inheritance structure of different domain types

Table 33 Mapping cloud pattern to problem frames

Cloud pattern	Problem frames
Indirect environment	
Legislator	Considered as requirements in problem diagrams
Domain	Considered as requirements in problem diagrams
Contract	Considered as requirements in problem diagrams
Direct environment	
Cloud provider	Biddable domain
Cloud support	Biddable domain
Cloud administrator	Biddable domain
Cloud customer	Biddable domain
Cloud developer	Biddable domain
End customer	Biddable domain
Cloud: service	
IaaS	Causal domain
PaaS	Causal domain
SaaS	Causal domain
Cloud software stack	Causal domain or machine domain
Software product	Causal domain or machine domain
Data	Lexical domain
Cloud: pool	
Resource	Causal domain or machine domain
Location	Part of the description of a causal domain
Software	Causal domain or machine domain
Hardware	Causal domain or machine domain
Hypervisor	Causal domain or machine domain
Cloud Database	Causal domain or machine domain
Stakeholder templates	
Name	Name of the biddable domain
Description	Used in the description of the domain
Motivation	Used in the description of the domain
Relations to other stakeholders	Basis for one or more phenomenon
Relations to the cloud	Basis for one or more phenomenon
Relations to other direct stakeholders	Basis for one or more phenomenon
Assets	Causal domain or lexical domain
Compliance and privacy	Considered as requirements in problem diagrams

certain context. Therefore, it is important to describe the *environment*, because we build a system to improve something in the world. The environment in which the system to be built (called *machine*) will operate is represented by a *context diagram* and problem decomposition in *problem diagrams*.

We use the UML4PF framework and its tool support for our method, which uses the UML4PF profile to create a context diagram and problem diagrams. A detailed description can be found in [41].

Stereotypes give a specific meaning to the elements of a UML diagram they are attached to, and they are represented by labels surrounded by double angle brackets (see Fig. 30). The class with the stereotype *machine* represents the thing to be developed (e.g., the software). The classes with some domain stereotype, e.g., *CausalDomain* or *BiddableDomain* represent *problem domains* that already exist in the application environment. Domains are connected by interfaces consisting of *shared phenomena*. Shared phenomena may be events, operation calls, messages, and the like. They are observable by at least two domains, but controlled by only one domain, as indicated by an exclamation mark. Jackson distinguishes the domain types, as depicted in Fig. 30. *CausalDomain*s comply with some physical laws, *LexicalDomain*s that are data representations, and *BiddableDomain*s that are usually people. In the UML4PF profile *Domains* have *names* and *abbreviations*, which are used to define interfaces. Hence, the class *Domain* has the attributes *name* and *abbreviation* of type string.

Requirements engineering with problem frames proceeds as follows: first, the environment in which the machine will operate is represented by a *context diagram*. Like a frame diagram, a context diagram consists of domains and interfaces. However, a context diagram contains no requirements. Then, the problem is decomposed into subproblems. If ever possible, the decomposition is done in such a way that the subproblems fit to given problem frames. To fit a subproblem to a problem frame, one must instantiate its frame diagram, i.e., provide instances for its domains, phenomena, and interfaces. The instantiated frame diagram is called a *problem diagram*.

12.2 A method for considering privacy in an ISMS

We refine the steps 10 to 12 of PACTS and combine them into a privacy method (Fig. 31). We describe the steps of our privacy method for privacy requirements elicitation and threat analysis in the following. We state the steps of PACTS and underneath the steps of our privacy method that refine them.

12.3 Pacts Step 10: instantiate privacy patterns

Describe the environment We use the information in the cloud pattern with our privacy methods. Hence, we present a mapping of our cloud pattern and the templates in Table 33. We first map the stakeholders of the indirect environment to domains. These are part of the indirect environment, which is not considered in the context diagram of Jackson's approach, but in the problem frames.

The stakeholders of the direct environment are all considered as biddable domains that already appear in the context diagram. We divided the cloud into cloud elements that are part of the service and elements that are part of the pool. The IaaS, PaaS, and SaaS layers are considered as an integral part of the cloud and, hence, only used and not part of the machine to be built, while the cloud software stack, the software product can be part of the machine to be build. Data maps to a lexical domain. All elements that are part of the pool can be either a causal domain or part of the machine, except for the location(s) of the resource. We combined the elements of the templates for direct and indirect stakeholders. The name, description, and motivation are used in the attributes of domains. Relations between stakeholders or between stakeholders and the cloud give rise to phenomena between domains. Assets are lexical domains if they are only data, if they also have a physical representation these are causal domains. For example, an address would be a lexical domain and a hard drive containing an address is a causal domain. Compliance and privacy demands are mapped to requirements in problem diagrams.

The first step of our method uses the mapping table and instantiated cloud patterns. This results in several problem frame models, namely a context diagram and several problem frames. These are accompanied by textual requirements.

Instantiate privacy patterns In the second step, we use textual patterns for privacy requirements introduced in previous work [39]. Privacy requirements are difficult to elicit for any given software engineering project that processes personal information. The problem is that these systems require personal data in order to achieve their functional requirements and privacy mechanisms that constrain the processing of personal information in such a way that the requirement still states a useful functionality.

We present privacy patterns for anonymity, pseudonymity, unlinkability and unobservability in accordance with the definitions of the ISO 15408 standard—CC for Information Technology Security Evaluation (or short CC) [1]. Our privacy patterns have a textual representation that can be instantiated using problem frame models. We also show predicate patterns that can validate the instantiation of our privacy patterns. We presented an exhaustive discussion about privacy terminology in [39] and present in this work only the CC definition of these terms.

The privacy specification in the CC defines four privacy goals. These goals can be refined into privacy requirements for a given software system. *Anonymity* means that a subject is not identifiable within a set of subjects, the anonymity set. *Unlinkability* of two or more items of interest (IOI) means that within a system the attacker cannot sufficiently distinguish whether these IOIs are related or not. *Unobservability* of an IOI means that an IOI is not detectable by any subject uninvolved in it and anonymity of the subject(s) involved in the IOI even against the other subject(s) involved in that IOI. A pseudonym is an identifier of a subject other than one of the subject's real names. Using pseudonyms means *pseudonymity*.

We explain specific privacy domain types, which we use in the remainder of our method:

- A *Stakeholder* is a *BiddableDomain* (and in some special cases also a *CausalDomain*) with some relation to stored or transmitted personal information. It is not necessary that a stakeholder has an interface to the machine.
- A *CounterStakeholder* is a *BiddableDomain* that describes all subjects (with their equipment) who can compromise the privacy of a *Stakeholder* at the machine. We do not use the term attacker here, because the word attacker hints malicious intend. Privacy of stakeholders can also be violated by accident.
- *PersonalInformation* is a *CausalDomain* or *LexicalDomain* that represents personal information about a *Stakeholder*. The difference between these domains is that a *LexicalDomain* describes just the stored information, while a *CausalDomain* also includes the physical medium the data is stored upon, e.g., a hard drive.
- *StoredPersonalInformation* is *PersonalInformation*, which is stored in a fixed physical location, e.g., a hard drive in the U.S.
- *TransmittedPersonalInformation* is *PersonalInformation*, which is transmitted in between physical locations, e.g., data in a network that spans from Germany to the U.S.
- *InformationAboutPersonalInformation* is a *CausalDomain* or *LexicalDomain* that represents information about *PersonalInformation*, e.g., the physical location of the name and address of a stakeholder.

We introduce four privacy patterns in the following. The terms in *italic* can be instantiated. These patterns are a subset of the patterns presented in [39].

Anonymity

> Preserve anonymity of *Stakeholder*s and prevent disclosure of their identity by *CounterStakeholder*s.

Pseudonymity A *Pseudonym* is a *LexicalDomain* used as an identifier of a *Stakeholder* without revealing *PersonalInformation*. An *Authorized User* is a *Stakeholder* who is allowed to know the identity of the *Stakeholder* the *Pseudonym* belongs to.

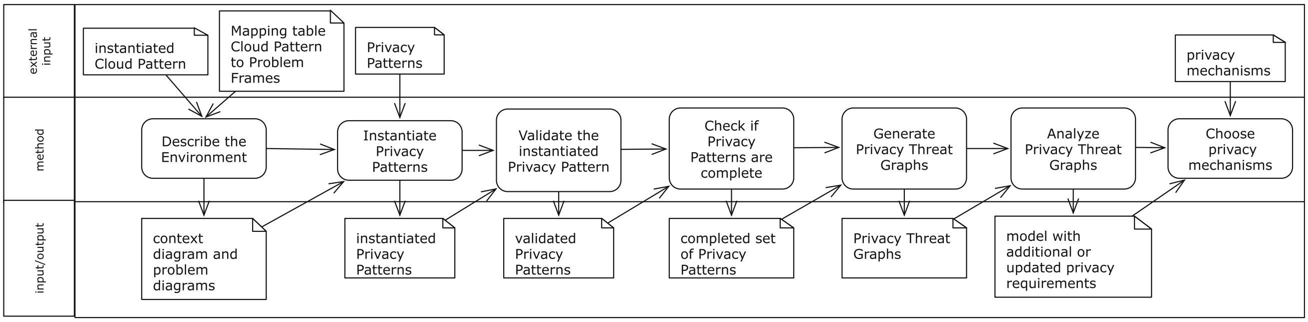

external input
method
input/output
instantiated Cloud Pattern
Mapping table Cloud Pattern to Problem Frames
Privacy Patterns
privacy mechanisms
Describe the Environment
Instantiate Privacy Patterns
Validate the instantiated Privacy Pattern
Check if Privacy Patterns are complete
Generate Privacy Threat Graphs
Analyze Privacy Threat Graphs
Choose privacy mechanisms
context diagram and problem diagrams
instantiated Privacy Patterns
validated Privacy Patterns
completed set of Privacy Patterns
Privacy Threat Graphs
model with additional or updated privacy requirements

Fig. 31 A method for considering privacy using the cloud pattern and problem frames

> Preserve pseudonymity of *Stakeholder*s via preventing *CounterStakeholder*s from relating *Pseudonym*s to their *Stakeholder*s.

Unlinkability A *ConstrainedDomain* is a *CausalDomain* or a *ConnectionDomain* that is constrained by a functional or privacy requirement.

> Preserve unlinkability of two or more *ConstrainedDomain*s for *Stakeholder*s and prevent *CounterStakeholder*s of disclosing that the *ConstrainedDomain*s have a relation to the *Stakeholder*.

Unobservability

> Preserve unobservability of a *ConstrainedDomain* that is used by *Stakeholder*s and prevent *CounterStakeholder*s from recognizing that the *ConstrainedDomain* exists.

Validate the instantiated privacy patterns In addition, for each privacy pattern a predicate pattern exists that can be used to validate its instantiation. This validation is based upon the problem frame models and it validates that the pattern is instantiated with the correct privacy domain types. We introduce our predicate patterns in the following.

Validate the instances of the anonymity pattern with:

$$anon_{cs} : BiddableDomain \times \mathbb{P}\, BiddableDomain \rightarrow Bool$$

The suffix "cs" indicates that this predicate describes a requirement considering a certain **C**ounter**S**takeholder. The definition of anonymity from Pfitzmann and Hansen [42] states that a stakeholder shall not be identifiable from a set of stakeholders. This is the so-called *anonymity set*, which is represented in our pattern by a *Stakeholder*, a subclass of the *Biddable Domain*. Hence, all the persons which can be instantiated as a specific *Stakeholder* form the anonymity set.

Validate the instances of the pseudonymity pattern with:

$$\begin{aligned} pseudo_{cs} : {} & LexicalDomain \times BiddableDomain \\ & \times \mathbb{P}\, BiddableDomain \\ & \rightarrow Bool \end{aligned}$$

Validate the instances of the unobservability pattern with:

$$\begin{aligned} unobserv_{cs} : {} & \mathbb{P}\, CausalDomain \times BiddableDomain \\ & \times \mathbb{P}\, BiddableDomain \\ & \rightarrow Bool \end{aligned}$$

Validate the instances of the unlinkability pattern with:

$$\begin{aligned} unlink_{cs} : {} & \mathbb{P}\, CausalDomain \times BiddableDomain \\ & \times \mathbb{P}\, BiddableDomain \\ & \rightarrow Bool \end{aligned}$$

Check if privacy patterns are complete The predicate patterns can also be used to identify incomplete privacy requirements. In incomplete privacy pattern instantiation domain types are missing, which results in incomplete requirements. Hence, we check for each requirement that all the textual gaps are instantiated and if gaps can be instantiated with sets of domains are completely instantiated.

For example, requirements that have to be instantiated with a *Stakeholder* have to name an instance of a *Biddable Domain* from the context diagram, e.g., *Bank Customer*s. Several privacy patterns require instantiation with sets of *Biddable Domain*s. For example, a privacy pattern might not only be directed toward *Bank Customer*s, but also toward the *Bank Institute*. In this case, we can reason for all Biddable Domains in the context diagram if they are a *CounterStakeholder* or not.

In addition, a requirement might be missing, e.g., because of an incomplete threat analysis. In order to execute this check, all personal information in the cloud has to be elicited. For each *CausalDomain* or *LexicalDomain* we have to check if these are *StoredPersonalInformation*, *TransmittedPersonalInformation* or *InformationAboutPersonalInformation*. If this is the case, we check if these were considered in the privacy threat analysis. If this is not the case, we have to re-do the privacy threat analysis and start our process from the beginning.

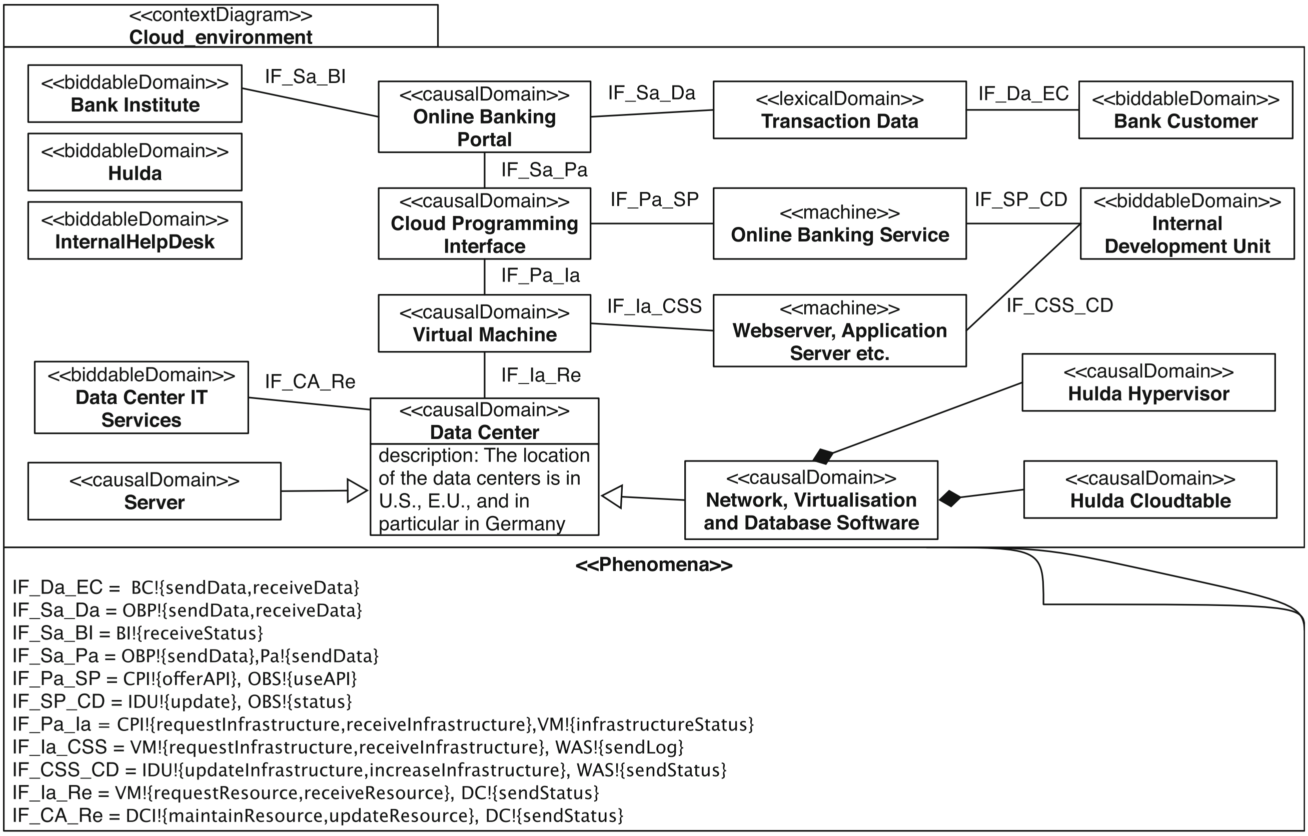

<<contextDiagram>>
Cloud_environment
<<biddableDomain>> Bank Institute
IF_Sa_BI
<<causalDomain>> Online Banking Portal
IF_Sa_Da
<<lexicalDomain>> Transaction Data
IF_Da_EC
<<biddableDomain>> Bank Customer
<<biddableDomain>> Hulda
<<biddableDomain>> InternalHelpDesk
IF_Sa_Pa
<<causalDomain>> Cloud Programming Interface
IF_Pa_SP
<<machine>> Online Banking Service
IF_SP_CD
<<biddableDomain>> Internal Development Unit
IF_Pa_Ia
<<causalDomain>> Virtual Machine
IF_Ia_CSS
<<machine>> Webserver, Application Server etc.
IF_CSS_CD
<<biddableDomain>> Data Center IT Services
IF_CA_Re
IF_Ia_Re
<<causalDomain>> Data Center
description: The location of the data centers is in U.S., E.U., and in particular in Germany
<<causalDomain>> Hulda Hypervisor
<<causalDomain>> Server
<<causalDomain>> Network, Virtualisation and Database Software
<<causalDomain>> Hulda Cloudtable
<<Phenomena>>
IF_Da_EC = BC!{sendData,receiveData}
IF_Sa_Da = OBP!{sendData,receiveData}
IF_Sa_BI = BI!{receiveStatus}
IF_Sa_Pa = OBP!{sendData},Pa!{sendData}
IF_Pa_SP = CPI!{offerAPI}, OBS!{useAPI}
IF_SP_CD = IDU!{update}, OBS!{status}
IF_Pa_Ia = CPI!{requestInfrastructure,receiveInfrastructure},VM!{infrastructureStatus}
IF_Ia_CSS = VM!{requestInfrastructure,receiveInfrastructure}, WAS!{sendLog}
IF_CSS_CD = IDU!{updateInfrastructure,increaseInfrastructure}, WAS!{sendStatus}
IF_Ia_Re = VM!{requestResource,receiveResource}, DC!{sendStatus}
IF_CA_Re = DCI!{maintainResource,updateResource}, DC!{sendStatus}

Fig. 32 Context diagram of the instantiated cloud analysis pattern

12.4 Pacts Step 11: analyze privacy threats

Generate privacy threat graphs The problem-based privacy analysis (ProPAn) [40] is a method including tool support[15] for identifying privacy threats during the requirements analysis of software systems using problem frame models. Our approach does not rely entirely on the privacy analyst to detect privacy threats, but allows a computer-aided privacy threat identification that is derived from the relations between stakeholders, technology, and personal information in the system-to-be. We defined a UML-profile for privacy requirements and a reasoning technique that identifies stakeholders, whose personal information are stored or transmitted in the system-to-be and stakeholders from whom we have to protect this personal information. For this purpose, we have tool support that uses problem frame diagrams to create a privacy threat graph. This graph uses the information from the instantiated privacy patterns documents the information flow of personal information in the system-to-be. In our method, graphs are labeled and directed and the set of vertices is a subset of the domains occurring in the model. The edges are annotated with problem diagrams and point from one domain to another. Hence, the graph can be used to evaluate the information flow between the domains. A formal definition is provided in previous work [40].

Analyze privacy threat graphs The analysis of the privacy threat graph reveals if privacy requirements need to be added or updated in the model. The results of the analysis lead to a refined set of privacy requirements.

Choose privacy mechanism The last step of our approach is to *choose a privacy mechanism* that solves the problem. For example, to achieve pseudonymity a privacy enhancing identity management systems [43] can be chosen.

12.5 Example of our privacy method

We used the mapping in Table 33 to create a context diagram according to Jackson, depicted in Fig. 32. In this work, we focus on the interactions with the *machine*. We created a number of phenomena to describe these interactions in more detail. The application used in this example considers that the *EndCustomer* sends and receives *Data* to the cloud via the *SaaS* and the *CloudCustomer* receives status messages from the *SaaS*. This service, in turn, sends the data to the *PaaS* domain, which provides access to the data via an API to the *SoftwareProduct*. The

[15] http://www.uni-due.de/swe/apf12.shtml.

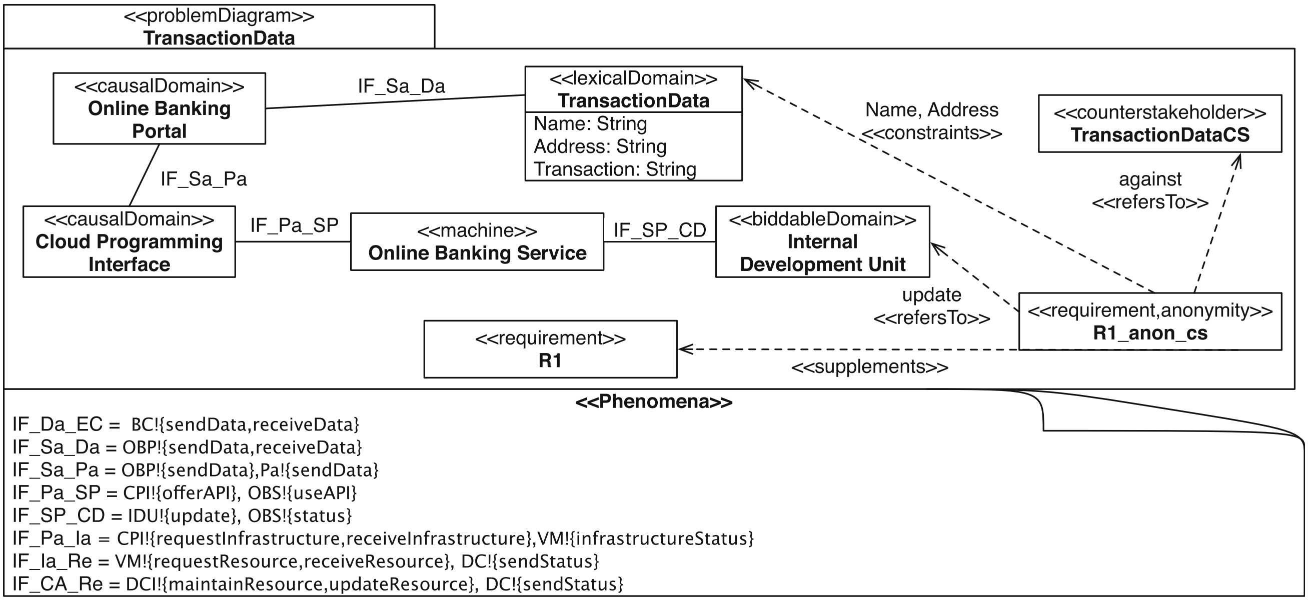

<<problemDiagram>>
TransactionData
<<causalDomain>>
Online Banking Portal
IF_Sa_Da
<<lexicalDomain>>
TransactionData
Name: String
Address: String
Transaction: String
Name, Address
<<constraints>>
<<counterstakeholder>>
TransactionDataCS
against
<<refersTo>>
IF_Sa_Pa
<<causalDomain>>
Cloud Programming Interface
IF_Pa_SP
<<machine>>
Online Banking Service
IF_SP_CD
<<biddableDomain>>
Internal Development Unit
update
<<refersTo>>
<<requirement,anonymity>>
R1_anon_cs
<<requirement>>
R1
<<supplements>>
<<Phenomena>>
IF_Da_EC = BC!{sendData,receiveData}
IF_Sa_Da = OBP!{sendData,receiveData}
IF_Sa_Pa = OBP!{sendData},Pa!{sendData}
IF_Pa_SP = CPI!{offerAPI}, OBS!{useAPI}
IF_SP_CD = IDU!{update}, OBS!{status}
IF_Pa_Ia = CPI!{requestInfrastructure,receiveInfrastructure},VM!{infrastructureStatus}
IF_Ia_Re = VM!{requestResource,receiveResource}, DC!{sendStatus}
IF_CA_Re = DCI!{maintainResource,updateResource}, DC!{sendStatus}

Fig. 33 Problem frame for anonymity constrain

SoftwareProduct receives updates from the *CloudDeveloper* and sends status messages to the *CloudDeveloper*. The *PaaS* domain requests infrastructure from the *IaaS* domain, which, in turn, requests these from the *CloudSoftwareStack*. The *CloudDeveloper* increases and updates the infrastructure of the *CloudSoftwareStack*. The *IaaS* domain requests resources in order to provide the infrastructure from the *Resource*, which is maintained and updated by the *CloudAdministrator*. The description of the *Resource* has to contain the location of the resource.

We included the biddable domains of the *CloudCustomer, CloudProvider*, and *CloudSupport* into the context diagram, but these have no relations to the cloud in the context diagram, because these do not have interactions with the cloud directly that could be mapped to a phenomena. We instantiated the context diagram according to the instantiated Cloud Pattern in Fig. 6, depicted in Fig. 32. The context diagram is accompanied by textual requirements. For simplicity, we show three requirements at this place.

- **R1** The *Bank Customer* can store *Transaction Data* using the *Online Banking Portal*.
- **R2** All *Transaction Data* from the *Online Banking Service* is stored in the *Hulda Cloudtable*.
- **R3** The *Data Center IT Services Unit* ensures the availability of the *Hulda Cloudtable*.

We draw a problem diagram for each requirement in order to refine it. We present a problem frame for **R1** in Fig. 33.

Instantiate privacy patterns We aim to protect the privacy of the *Bank Customer* and choose to instantiate an anonymity privacy pattern in order to do so. The reason is that the identity of the *Bank Customers* is personal information and should not be revealed. Anonymity offers a way to achieve this. This decision is also done without thinking about further functional requirements and should represent a naive decision. The experience of the authors is that the decision about privacy requirements is often taken in a simplistic manner. Hence, we instantiate our privacy pattern for anonymity, which is shown in the following.

> Preserve anonymity of *Bank Customers* and prevent disclosure of their identity by the *Internal Development Unit*.

The problem frame shown in Fig. 33 presents the anonymity requirement *R1_anon_cs* that supplements the requirement **R1** and constrains the *TransactionData*, specifically the *Name* and *Address* attributes of it. The requirement protects against the counterstakeholder *TransactionDataCS*. The *Internal Development Unit*

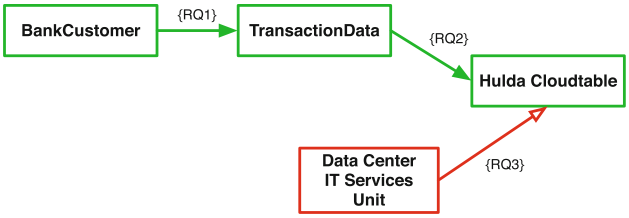


Fig. 34 Privacy threat graph

conducts an update as part of the phenomenon *IF_SP_CD*. The requirements refer to this phenomenon.

Validate the instantiated privacy patterns We execute the validation check for the anonymity pattern. The result is true, because the instantiated stakeholder and the counterstakeholder are both instantiated and are biddable domains.

Check if privacy patterns are complete The iteration over all the possible biddable domains reveals that the biddable domain *Data Center IT Services Unit* is a possible counterstakeholder and it is also integrate into the privacy pattern and the problem frame. The reason is that the *Data Center IT Services Unit* can identify the identity of *Bank Customers*.

Generate privacy threat graphs We use the ProPAn tool to generate a privacy threat graph for our scenario to visualize, which domains have a relation to each other. The relations are labeled with the requirements that introduce them. For simplicities sake, we do not show the entire privacy threat graph here, but for the scenario presented here we expect at least 30 requirements and at least 90 relations between domains from our previous experience when using ProPAn. The effort for modeling the 30 problem frames can be calculated by 1 h per problem frame. This results in at least 30 person hours.

Analyze privacy threat graphs We present a privacy threat graph for the *Bank Customer* considering the requirements **R1, R2**,**R3** in Fig. 34 and the counter-stakeholder *Data Center IT Services Unit*. The edges from the counterstakeholder are red (light gray), bold edges with white arrowheads. We have two possibilities to solve the privacy threats identified by the threat graph. We can consider the edges starting from the counter-stakeholder and restrict the information the counter-stakeholder can access. We consider the other edges of the threat graph, as well. This results in a restriction of the information flow between the domains.

We can see that the counter-stakeholder *Data Center IT Services Unit* may gain information from the lexical domain *Transaction Data*, while it is being stored in the *Hulda Cloudtable*. In order to fulfill the anonymity requirement, the *Name* and *Address* attributes have to be erased. If these are not erased, anonymity cannot be achieved. However, this data is required for the *Bank Institute* to conduct banking business with the *Bank Customer*. A possible solution is to apply pseudonymity instead of anonymity. In this case, the *Name* and *Address* attributes are replaced with other values and a later mapping is possible.

Choose privacy mechanism After identifying privacy threats we have to reason, which technical privacy enhancement technique to us. We refer to the works of Deng et al. [26] that represent an extensive list of these mechanisms.

13 Related work

We present related work regarding the ISO 27001 standard and clouds as well as legal compliance and privacy.

13.1 ISO 27001 and cloud computing

Calder [14] and Kersten et al. [44] provide advice for an ISO 27001 realization. In addition, Klipper [16] focuses on risk management according to ISO 27005. The author also includes an overview of the ISO 27000 series of standards. However, none of these works consider to use security requirements engineering methods.

Cheremushkin et al. [45, 46] present a UML-based meta-model for several terms of the ISO 27000, e.g., assets. These meta-models can be instantiated and, thus, support the refinement process. However, the authors do not present a holistic approach to information security. The work mostly constructs models around specific terms in isolation. The CF of Fabian et al. [24], on the other hand, presents a holistic framework for information security.

Montesino et al. [47] investigate possible automation of controls that are listed in the ISO 27001 and ISO 27002. Their work can complement our own.

Fenz et al. [48] introduce an ontology-based framework for preparing ISO/IEC 27001 audits. They provide a rule-based engine which uses a security-ontology to determine if security requirements of a company are fulfilled.

Auty et al. [49] base their work on the risk controls in the ISO 27001 [4] and 27002 [50] standards. The authors discuss if these controls are adequate for cloud computing or if adjustments have to be considered. The discussion ranges from social to technical threats. This approach can complement our own by proposing updates for the ISO 27001 and ISO 27002 controls based upon the findings of Auty et al.

Shaikh and Haider [51] map the existing research for cloud security threats to the categories context, problem description, technique used, and proposed models or tools. The authors conclude that privacy and data loss are the threats that cause the most concerns. This approach differs from our own, because the authors aim to identify the most severe threats rather than providing a threat analysis method.

Greenwood and Sommerville [52] propose to use responsibility modeling to identify threats for cloud computing. The method operates on the same abstraction levels as goal based notations like I*. The method models agents in the system, resources, responsibilities and the relations among them. Resources can be information or physical systems. The threat analysis investigates the responsibilities and describes conditions in which case a threat to this responsibility can occur. Our method is based upon a cloud pattern, which can be re-used for different projects. The work of Greenwood and Summerville requires a new model for each project.

Grobauer et al. [53] investigate risks for cloud providers and users. The authors base their work upon the ISO 27005 [54] standards for high-level risk criteria, and they use the risk taxonomy of the Open Group for a refinement of these criteria. The authors map these risks to a cloud architecture description from IBM. The authors describe risks, e.g., for the cloud management interface. The work can complement our own by integrating the risks into our threat analysis.

13.2 Legal compliance and privacy

Breaux et al. [55, 56] present a framework that covers analyzing the structure of laws using a natural language pattern. This pattern helps to translate laws into a more structured restricted natural language and then into a first order logic. The idea of using first order logic in the context of regulations is not a new one. For example Bench-Capon et al. [57] made use of first order logic to model regulations and related matters. In contrast to our work, the authors of those approaches assume that the relevant laws are already known and thus do not support identifying legal texts.

Siena et al. [58] describe the differences between legal concepts and requirements. They model the regulations using an ontology, which is quite similar to the natural language patterns described in the approaches mentioned before. The ontology is based in the Hohfeld taxonomy [59], which describes the means and relations between the different means of legal texts in a very generic way. Thus, Hohfeld does not structure a certain law at all but aims at the different meanings of laws. So the resulting process in [58] to align legal concepts to requirements and the given concepts are quite high level and cannot directly applied to a scenario. In a second work Siena et al. [60] try to bridge the gap between the requirements engineering process and compliance using a goal-oriented approach. In contrast to our approach they do not identify relevant laws and do not intertwine compliance regulations with already elicited requirements.

Álvarez et al. [61] describe reusable legal requirements in natural language, and based on the Spanish adaption of the EU directive 95/46/CE concerning personal data protection. We believe that the work by Álvarez et al. complements our work, i.e., applying our law identification method can precede using their security requirements templates.

Deng et al. [26] present a threat tree for privacy based upon the threat categories: linkability, identifiability, non-repudiation, detectability, information disclosure, content unawareness, and policy/consent non-compliance. These threats are modeled for the elements of an information flow model, which has data flow, data store, processes and entities as components. Privacy threats are described for each of these components. Hence, privacy threat identification for an existing data flow model is simplified, because for each data flow element in a model only the threats shown in the tree need to be considered. The work differs from our own, because the privacy threat identification has to be carried out manually.

The PriS method [62] 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, while our work uses a problem-based approach as a foundation. The difference is that our work focuses on a description of the environment as a foundation for the privacy analysis, while the PriS method uses organizational goals as a starting point. In addition, the PriS method has to be carried out manually.

Hafiz [63] describes four privacy design patterns for the network level of software systems. These patterns solely focus on anonymity and unlinkability of senders and receivers of network messages from protocols, e.g., http. The patterns are specified in several categories. Among them are intent, motivation, context, problem and solution, as well as forces, design issues and consequences. This work focuses on privacy issues on the network layer and can complement our work in this area.

14 Conclusion

The decision whether a cloud service is chosen by a costumer relies, among other reasons, on how trustworthy the cloud system is. One way to establish this trust is to demonstrate that security, privacy, and compliance are taken seriously by the cloud provider. This is usually achieved by providing certified services. A well-known standard for such a certification is the ISO 27001 standard. However, establishing an ISMS as required by this standard is a non-trivial task. Furthermore, the standard does not take the special needs of cloud computing into consideration, yet. With the work presented in this article we intend to close the aforementioned gap. We do so by providing a structured pattern-based method to establish an

ISMS according to the ISO 27001 standard. It has been tailored to suit the demands for the cloud computing domain. We introduce specific patterns for clouds to elicit the context of the envisioned ISMS. The approach further allows to refine the initially elicited context with behavior descriptions. It also provides the means for documenting management commitment, threat and risk analysis, as well as a pattern-based definition of security policies compliant to the ISO 27001 standard. We enhance the approach by providing validation conditions that can be used to check the instantiated context as well as policy patterns. It is, for example, possible to check whether a given responsible stakeholder in the policy pattern is also present in the context pattern. Moreover, we take the standard's demand to consider legal compliance and privacy into account.

In summary, the benefits of our approach are:

- A structured method for establishing a cloud-specific ISMS compliant to ISO 27001.
- Detailed steps for asset identification, threat analysis, risk management and security reasoning.
- The pattern-based method provides the means for consistency checks e.g., for the instantiation of the pattern.
- Consideration of legal compliance via steps for identifying laws and regulations.
- Support for formulating and validating privacy requirements and conducting a privacy threat analysis.
- A systematic support to generate the required ISMS documentation in compliance to the standard.
- Integration of proven existing methods e.g., CORAS and Misuse Cases.
- Integrating requirements engineering for security, legal compliance, and privacy to construct a holistic ISMS.

In the future, we plan to provide a UML model for the cloud and policy patterns and implement the consistency checks using the Object Constraint Language [64]. We currently work on providing tool support for generating documents from our instantiated patterns in accordance with the ISO 27001 standard.

Acknowledgments 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) and the Ministry of Innovation, Science, Research and Technology of the German State of North Rhine-Westphalia and EFRE (Grant No. 300266902 and Grant No. 300267002).

References

1. ISO/IEC (2009) Common criteria for information technology security evaluation. ISO/IEC 15408, International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC)
2. Cloud Security Alliance (CSA) (2010) Top threats to cloud computing v1.0. http://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf
3. Gartner (2008) Assessing the security risks of cloud computing. http://www.gartner.com/id=685308
4. ISO/IEC (2005) Information technology—Security techniques—Information security management systems—Requirements. ISO/IEC 27001, International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC)
5. Armbrust M, Fox A, Griffith R, Joseph AD, Katz RH, Konwinski A, Lee G, Patterson DA, Rabkin A, Stoica I, Zaharia M (2009) Above the clouds: A berkeley view of cloud computing. Technical report, EECS Department, University of California, Berkeley
6. Mell P, Grance T (2009) The NIST definition of cloud computing. Working Paper of the National Institute of Standards and Technology (NIST)
7. Vaquero LM, Rodero-Merino L, Caceres J, Lindner M (2008) A break in the clouds: Towards a cloud definition. Special Interest Group Data Commun (SIGCOMM) Comput Commun Rev 39(1):50–55
8. Buyya R, Ranjan R, Calheiros RN (2009) Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities. In: Proceedings of the international conference von high performance computing and simulation (HPCS). IEEE Computer Society
9. Beckers K, Küster JC, Faßbender S, Schmidt H (2011) 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, pp 327–333
10. Jackson M (2001) Problem frames: analyzing and structuring software development problems. Addison-Wesley, Reading, MA
11. Fowler M (1996) Analysis patterns: reusable object models. Addison-Wesley, Reading, MA
12. Gamma E, Helm R, Johnson R, Vlissides J (1994) Design patterns: elements of reusable object-oriented software. Addison-Wesley, Reading, MA
13. Schumacher M, Fernandez-Buglioni E, Hybertson D, Buschmann F, Sommerlad P (2006) Security patterns: integrating security and systems engineering. Wiley, New York
14. Calder A (2009) Implementing Information Security based on ISO 27001/ISO 27002: A Management Guide. Haren Van Publishing
15. ISO/IEC (2009) Information technology—Security techniques—Information security management systems—Overview and Vocabulary. ISO/IEC 27000, International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC)
16. Klipper S (2010) Information security risk management mit ISO/IEC 27005: Risikomanagement mit ISO/IEC 27001, 27005 und 31010. Vieweg+ Teubner
17. UML Revision Task Force. OMG Unified Modeling Language (UML), Superstructure. http://www.omg.org/spec/UML/2.3/Superstructure/PDF
18. IETF (1997) Hmac: keyed-hashing for message authentication. IETF rfc 2104, Internet Engineering Task Force (IETF)
19. Jansen WA (2011) Cloud hooks: Security and privacy issues in cloud computing. In: HICSS. IEEE Computer Society, pp 1–10
20. Chang F, Dean J, Ghemawat S (2006) Bigtable: A distributed storage system for structured data. Technical report, Google
21. Chow R, Golle P, Jakobsson M, Shi E, Staddon J, Masuoka R, Molina J (2009) Controlling data in the cloud: outsourcing

computation without outsourcing control. In: CCSW. ACM, pp 85–90

22. Scarfone KA, Souppaya MP, Hoffman P (2011) Sp 800-125. guide to security for full virtualization technologies. Technical report, NIST, Gaithersburg, MD, USA
23. Government H (2012) It infrastructure library (ITIL). http://www.itil-officialsite.com/home/home.aspx
24. Fabian B, Gürses S, Heisel M, Santen T, Schmidt H (2010) A comparison of security requirements engineering methods. Requir Eng 15(1):7–40
25. Opdahl AL, Sindre G (2009) Experimental comparison of attack trees and misuse cases for security threat identification. Inf Softw Technol 51:916–932
26. Deng M, Wuyts K, Scandariato R, Preneel B, Joosen W (2011) A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requir Eng 16:3–32
27. Lund MS, Solhaug B, Stølen K (2010) Model-driven risk analysis: the CORAS approach, 1st edn. Springer, Berlin
28. American National Standards Institute (ANSI) (2004) American national standard for information technology—role based access control. Ansi incits, pp 359–2004, ANSI
29. OASIS (2005) extensible Access Control Markup Language TC v2.0 (XACML). OASIS. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf
30. McGraw G (2006) Software security: building security in. Addison-Wesley, Reading, MA
31. VMWARE. Vmware ha. http://www.vmware.com/de/products/datacenter-virtualization/vsphere/high-availability.html
32. VMWARE. Vmware vmotion. http://www.vmware.com/files/pdf/VMware-VMotion-DS-EN.pdf
33. Beckers K, Faßbender S, Küster JC, Schmidt H (2012) A pattern-based method for identifying and analyzing laws. In: Proceedings of the international working conference on requirements engineering: foundation for software quality (REFSQ). In: LNCS. Springer, pp 256–262
34. Beckers K, Faßbender S, Schmidt H (2012) An integrated method for pattern-based elicitation of legal requirements applied to a cloud computing example. In: Proceedings of the international conference on availability, reliability and security (ARES)—2nd international workshop on resilience and it-risk in social infrastructures (RISI 2012). IEEE Computer Society, pp 463–472
35. Biagioli C, Mariani P, Tiscornia D (1987) Esplex: a rule and conceptual model for representing statutes. In: ICAIL. ACM, pp 240–251
36. Duisberg A (2011) Gelöste und ungelöste Rechtsfragen im IT-Outsourcing und Cloud Computing. In: Picot A, Götz T, Hertz U (eds) Trust in IT, Springer, Berlin, pp 49–70
37. Gürses SF, Santen T (2006) Contextualizing security goals: a method for multilateral security requirements elicitation. In: Dittmann J (ed.), Sicherheit 2006: Sicherheit—Schutz und Zuverlässigkeit, Beiträge der 3. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.v. (GI), pp 20–22. Februar 2006 in Magdeburg, vol 77 of LNI., pp 42–53. GI
38. OECD (1980) OECD guidelines on the protection of privacy and transborder flows of personal data. Technical report, Organisation for Economic Co-operation and Development (OECD)
39. Beckers K, Heisel M (2012) A foundation for requirements analysis of privacy preserving software. In: Proceedings of the International Cross Domain Conference and Workshop (CD-ARES 2012). Lecture Notes in Computer Science, Springer, pp 93–107
40. Beckers K, Faßbender S, Heisel M, Meis R (2012) A problem-based approach for computer aided privacy threat identification. In: Privacy Forum 2012. Lecture Notes in Computer Science, Springer. Accepted for Publication
41. Côté I, Hatebur D, Heisel M, Schmidt H (2011) UML4PF—a tool for problem-oriented requirements analysis. In: Proceedings of the international conference on requirements engineering (RE), IEEE Computer Society, pp 349–350
42. Pfitzmann A, Hansen M (2011) A terminology for talking about privacy by data minimization: Anonymity, unlinkability, unobservability, pseudonymity, and identity management—version v0.34. Technical report, TU Dresden and ULD Kiel
43. Clauß S, Kesdogan D, Kölsch T (2005) Privacy enhancing identity management: protection against re-identification and profiling. In: Proceedings of the 2005 workshop on Digital identity management. DIM '05, ACM, pp 84–93
44. Kersten H, Reuter J, Schröder KW (2011) IT-Sicherheits management nach ISO 27001 und Grundschutz. Vieweg+Teubner
45. Cheremushkin DV, Lyubimov AV (2010) An application of integral engineering technique to information security standards analysis and refinement. In: Proceedings of the international conference on Security of information and networks. SIN '10, ACM, pp 12–18
46. Lyubimov A, Cheremushkin D, Andreeva N, Shustikov S (2011) Information security integral engineering technique and its application in isms design. In: Proceedings of the international conference on availability, reliability and security (ARES), IEEE Computer Society, pp 585–590
47. Montesino R, Fenz S (2011) Information security automation: how far can we go? In: Proceedings of the international conference on availability, reliability and security (ARES), IEEE Computer Society, pp 280–285
48. Fenz S, Goluch G, Ekelhart A, Riedl B, Weippl E (2007) Information security fortification by ontological mapping of the ISO/IEC 27001 standard. In: Proceedings of the international symposium on dependable computing, IEEE Computer Society, pp 381–388
49. Auty M, Creese S, Goldsmith M, Hopkins P (2010) Inadequacies of current risk controls for the cloud. In: Proceedings of the 2010 IEEE second international conference on cloud computing technology and science. CLOUDCOM '10, IEEE Computer Society, pp 659–666
50. ISO/IEC (2005) Information technology - Security techniques—code of practice for information security management. ISO/IEC 27002, International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC)
51. Shaikh F, Haider S (2011) Security threats in cloud computing. In: Internet technology and secured transactions (ICITST), 2011 international conference for, pp 214 –219
52. Greenwood D, Sommerville I (2011) Responsibility modeling for identifying sociotechnical threats to the dependability of coalitions of systems. In: System of systems engineering (SoSE), 2011 6th international conference on, pp 173 –178
53. Grobauer B, Walloschek T, Stocker E (2011) Understanding cloud computing vulnerabilities. Secur Priv, IEEE 9(2):50–57
54. ISO/IEC (2008) Information technology—security techniques—information security risk management. ISO/IEC 27005, International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC)
55. Breaux TD, Vail MW, Antón AI (2006) Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. In: RE, IEEE Computer Society, pp 46–55
56. Breaux TD, Antón AI (2008) Analyzing regulatory rules for privacy and security requirements. IEEE Trans Softw Eng 34(1):5–20
57. Bench-Capon T, Robinson G, Routen T, Sergot M (1987) Logic programming for large scale applications in law: a formalization of supplementary benefit legislation. In: ICAIL, ACM, pp 190–198
58. Siena A, Perini A, Susi A (2008) From laws to requirements. In: RELAW, IEEE Computer Society, pp 6–10

59. Hohfeld WN (1917) Fundamental legal conceptions as applied in judicial reasoning. Yale Law J 26(8):710–770
60. Siena A, Perini A, Susi A, Mylopoulos J (2009) A meta-model for modelling law-compliant requirements. In: Proceedings of the international workshop on requirements engineering and law (RELAW), IEEE Computer Society, pp 45–51
61. Álvarez JAT, Olmos A, Piattini M (2002) Legal requirements reuse: a critical success factor for requirements quality and personal data protection. In: Proceedings of the international conference on requirements engineering (RE), IEEE Computer Society, pp 95–103
62. Kalloniatis C, Kavakli E, Gritzalis S (2008) Addressing privacy requirements in system design: the PriS method. Requir Eng 13:241–255
63. Hafiz M (2006) A collection of privacy design patterns. In: Proceedings of the 2006 conference on pattern languages of programs. PLoP '06, ACM, pp 7:1–7:13
64. UML Revision Task Force (2010) OMG object constraint language: reference