future internet

Article

Context Analysis of Cloud Computing Systems Using a Pattern-Based Approach

Ludger Goeke [†], Nazila Gol Mohammadi [*,†] and Maritta Heisel

paluno – The Ruhr Institute for Software Technology, University of Duisburg-Essen, 47157 Duisburg, Germany; ludger.goeke@uni-due.de (L.G.); maritta.heisel@uni-due.de (M.H.)

[*] Correspondence: nazila.golmohammadi@uni-due.de; Tel.: +49-203-379-1929

[†] These authors are ordered alphabetically and contributed equally to this work.

Received: 14 June 2018; Accepted: 29 July 2018; Published: 31 July 2018

Abstract: Cloud computing services bring new capabilities for hosting and offering complex collaborative business operations. However, these advances might bring undesirable side-effects, e.g., introducing new vulnerabilities and threats caused by collaboration and data exchange over the Internet. Hence, users have become more concerned about security and privacy aspects. For secure provisioning of a cloud computing service, security and privacy issues must be addressed by using a risk assessment method. To perform a risk assessment, it is necessary to obtain all relevant information about the context of the considered cloud computing service. The context analysis of a cloud computing service and its underlying system is a difficult task because of the variety of different types of information that have to be considered. This context information includes (i) legal, regulatory and/or contractual requirements that are relevant for a cloud computing service (indirect stakeholders); (ii) relations to other involved cloud computing services; (iii) high-level cloud system components that support the involved cloud computing services; (iv) data that is processed by the cloud computing services; and (v) stakeholders that interact directly with the cloud computing services and/or the underlying cloud system components. We present a pattern for the contextual analysis of cloud computing services and demonstrate the instantiation of our proposed pattern with real-life application examples. Our pattern contains elements that represent the above-mentioned types of contextual information. The elements of our pattern conform to the General Data Protection Regulation. Besides the context analysis, our pattern supports the identification of high-level assets. Additionally, our proposed pattern supports the documentation of the scope and boundaries of a cloud computing service conforming to the requirements of the ISO 27005 standard (information security risk management). The results of our context analysis contribute to the transparency of the achieved security and privacy level of a cloud computing service. This transparency can increase the trust of users in a cloud computing service. We present results of the RestAssured project related to the context analysis regarding cloud computing services and their underlying cloud computing systems. The context analysis is the prerequisite to threat and control identification that are performed later in the risk management process. The focus of this paper is the use of a pattern at the time of design systematic context analysis and scope definition for risk management methods.

Keywords: cloud computing; information security; data protection; context analysis; pattern

1. Introduction

Information and communication technology trends such as cloud computing and Future Internet facilitate the growth of information systems and their integration in our daily life. Cloud computing has enabled significant improvements in efficiency and cost reduction. Cloud computing offers highly flexible and scalable usage of computing resources. However, the sharing of the provided computing resources by multiple users in a multi-tenant model (cf. [1], p. 2) and the provision of computing resources via the Internet causes trust concerns. Because of the difficulty of preventing malicious attacks or the misuse of critical information, users might not trust these systems. For instance, a survey study revealed security and trust concerns as being significant barriers for the acceptance and adoption of cloud computing in companies [2]. Reports (e.g., reference [3]) indicate an increasing number of cyber-crime victims, which has led to a massive deterioration of trust in current cloud computing systems (e.g., with respect to business-critical data). There are some issues in the cloud environment, e.g., the number of parties involved in handling the data of users and the fact that every involved party may have a variety of different goals. Therefore, the problems associated with protecting sensitive data, such as personal data or confidential business data, remain a major concern [4]. Service providers entrusted with handling sensitive data must comply with legislation on data protection as well as with individual users' data protection requirements. If not, they risk high penalties and damage to their reputation. For example, the European Union's General Data Protection Regulation (GDPR) [5] stipulates high fines in cases of non-compliance.

To ensure data protection in a complex and dynamic cloud environment, cloud service providers have to achieve an acceptable level of information security. To this end, cloud service providers should implement and operate an Information Security Management System (ISMS) for their cloud services (cf. [6], p. 21). An ISMS specifies all policies and procedures inside an organization for operating, controlling, ensuring and optimizing information security. The ISO 27001 is a standard that defines the normative requirements for the planning, implementation, maintenance and optimization of an ISMS. One part of an ISMS considers information security risk management. The ISO 27005 standard [7] provides informative guidelines for the realization of information security risk management. Performing a risk assessment on the assets of an organization is a crucial part of information security risk management. It has to be performed during the planning phase of an ISMS. Before performing a risk assessment, the context for the risk management process has to be established. A specific part of the context establishment (in ISO 27005 [7], Section 7.3) considers the specification of the scope and boundaries for information security risk management. The standard states that "the scope of the information risk management process needs to be defined to ensure that all relevant assets are taken into account in the risk assessment" [7]. The importance of this information becomes obvious because the next steps of the ISO 27005 depend upon them, e.g., the identification of threats and controls for the identified assets of an organization. Therefore, inappropriate context establishment can lead to an incomplete asset identification that, in turn, can result in great loss regarding information security. However, due to the sparse description in the standard, executing the specified steps for defining the scope and boundaries is not a trivial task [8].

This paper presents a pattern for analyzing the context of cloud computing services. It supports the identification of legal, regulatory and/or contractual requirements that are relevant for cloud computing services. For this purpose, the pattern provides different types of indirect stakeholders. Furthermore, it represents cloud computing service(s), the processed data and the used cloud computing resources by different types of cloud elements. For the identification of parties that directly interact with the cloud computing service or its used resources, the pattern contains specific types of direct stakeholders. In this context, the stakeholders that represent different types of roles as defined in the GDPR are of great importance. These stakeholders represent data subjects, data controllers and processors as well as data providers and consumers. We describe how the pattern and especially the different stakeholders and cloud elements can be used to support the ISO 27005 activities of defining the scope and boundaries during context establishment and asset identification.

We illustrate our pattern-based approach for context analysis with respect to data protection, using some industrial examples from the RestAssured project (https://restassuredh2020.eu).

The remainder of this paper is structured as follows. Section 2 presents an overview of the background concepts used in our approach. Section 3 explains the proposed pattern for context analysis. Section 4 presents the application of our pattern on industrial use cases. Section 5 explains our tool support for the proposed context analysis. Section 6 discusses related work. Section 7 concludes and sketches future work.

2. Background

In this section, we briefly introduce the fundamental techniques and concepts for our pattern for the context analysis that is described in Section 3.

2.1. Data Protection Goals

Data protection goals are the following: *confidentiality*, *integrity*, *availability*, *unlinkability*, *transparency* and *intervenability* [9]. *Confidentiality* refers to the non-disclosure of data, i.e., keeping it private. *Integrity* refers to avoiding the manipulation and corruption of data. *Availability* means that the data is available for access at any time. *Unlinkability* aims at the separation of privacy-relevant data from any other data, privacy-relevant or not. It should be impossible, or at least infeasible, to find a link between the privacy-relevant data and any other data that does not directly belong to that set of data. This also means that the privacy-relevant data should not be usable in any context other than the one it is intended for. Thus, assuring unlinkability also leads to purpose-binding, in which a set of data is only used for a specific purpose. An example would be the separation of account data like user-names and e-mail addresses from any Personally Identifiable Information (PII), i.e., real name and address. In case of a banking/payment system, the separation of login data and banking data is needed.

Transparency allows the involved parties to understand the privacy-relevant data handling processes and to be able to reconstruct data handling information at any time. It describes the transparency of the system, i.e., there is no black boxing allowed, as this would mean that the involved parties would not be able to understand the handling of the data. Transparency includes planned processing as well as the time after processing. A possible method of proving transparency is the availability of source code. An example is simple open source software. Due to the source code being available, the users of the software can find out what happens to their data. Thus, the software is transparent from a privacy protection point of view. If the software shares the data with some other service that is not an open source, transparency is no longer given.

Intervenability allows all involved parties to interfere with the data processing. This allows corrections in the system and countermeasures in the case something does not work as expected. Intervenability also allows the erasure of data and the withdrawal of consent to any privacy policy.

Note that we carefully distinguish between data protection goals and the measures and mechanisms that are used to achieve them. For example, pseudonymity is a mechanism to achieve the privacy goal of unlinkability. Hence, we do not consider pseudonymity as a goal in our work.

2.2. ISO 27001

The normative ISO 27001 [10] standard specifies “the requirements for establishing, implementing, maintaining and continually improving an information security management system within the context of an organization”. Since these requirements are defined in a generic way, the standard can be applied to organizations of any business area. In Annex A, this standard provides control objectives and controls regarding information security that should be implemented if they are relevant for an ISMS. The ISMS should enable organizations to achieve their information security objectives as well as the requirements regarding the information security of interested parties.

An initial step during the establishment of an ISMS is to determine the scope of the ISMS. Here, the following information is relevant: (1) external and internal issues that are relevant for the ISMS; (2) the information security requirements of interested parties; and (3) “interfaces and dependencies between activities performed by the organization and those that are performed by other organizations" [11].

For the planning phase of an ISMS, the standard specifies that a risk assessment has to be performed on the assets that are contained in the scope of the ISMS. Based on the results of the risk assessment, a process regarding the treatment for identified risks has to be defined and applied (cf. [11], p. 4). In risk treatment, the application of appropriate controls from Annex A to reduce the level of a risk is one option. Furthermore, organizations have to establish their information security objectives and plan how to achieve them. For the establishment of information security objectives, among others, the risk assessment results have to be taken into account.

Regarding the support of an ISMS, the standard defines requirements for the provisioning of resources as well as the competence and awareness of the involved personnel. Further requirements consider the external and internal communication regarding an ISMS as well as the documentation of an ISMS. The operation of the ISMS contains, in addition to the operational planning and controlling of the ISMS, a repetition of the risk assessment and risk treatment in planned intervals.

In the context of maintaining an ISMS, its performance has to be evaluated. For this purpose, the standard defines requirements regarding (1) the monitoring, measurement, analysis and evaluation of an ISMS; (2) the provision of internal audits; and (3) the conducting of reviews for the ISMS by the organization's management. The improvement of an ISMS shall be ensured by appropriate corrective actions regarding occurrences of non-conformity. Continued improvement of the suitability, adequacy and effectiveness of the ISMS (cf. [11], p. 9) is also needed.

Organizations can make transparent to the public that their ISMS conforms to the standard by commissioning an external, accredited auditor for certifying their ISMS. In the certification process, an external audit of the ISMS is performed. During this audit, an external auditor checks among others if the documentation of the ISMS is complete and corresponds to the standard. The implementation, deployment and performed improvements on the ISMS should be consistent with the documentation of the ISMS. Additionally, during the operation of an ISMS, different types of external audits are performed regularly.

2.3. ISO 27005

The ISO 27005 standard provides informative guidelines for conducting an information security risk management in an ISMS that conforms to ISO 27001 [11]. Here, the different guidelines give support for the application of information security risk management in the different phases.

An iteration of the information risk management process starts with a context establishment. For this purpose, the standard provides support by defining the following basic criteria:

- Selection of the risk management approach and assessment if the necessary resources for the risk management are available;
- Specification of risk evaluation criteria for the assessment of information security risks (e.g., value of assets for the business processes of an organization);
- Definition of impact criteria to assess the impact (e.g., monetary loss) if a security property (confidentiality, integrity, availability) of an asset gets compromised;
- Specification of risk acceptance criteria. These criteria are specified as scales for the values of acceptable risk levels regarding the different security properties of assets (confidentiality, integrity, availability).

Furthermore, ISO 27005 provides guidance for the definition of the scope and boundaries. It also supports the establishment of the organization and responsibilities for the risk management process.

After the context establishment, the risk assessment starts with risk identification which includes the following actions:

1. Identification of the assets within the defined scope of the ISMS;
2. Identification of threats for the identified assets;
3. Identification of existing controls that are already implemented;
4. Identification of vulnerabilities regarding the set of identified assets;
5. Identification of the damages and consequences to an organization if a security property of an asset gets compromised.

2.4. Pattern-Based Context Analysis

The context of a system represents a specific part of the system environment. This part is relevant for the definition of a system and understanding of its requirements [12]. A system context includes stakeholders, technical components, processes, events and regulations (e.g., laws) that are relevant for the system [12].

Basically, "a pattern is a solution to a recurring problem in a specific context" [13]. Patterns are used in the following domains: software requirements (e.g., [14]), software design (e.g., [15,16]), information security engineering (e.g., [17,18]), requirements engineering (e.g., [19–21]), test patterns (e.g., [22]), context analysis (e.g., [23]) and more.

The pattern for the definition of a context is a general structural description of a particular type of system including its environment (cf. [23]). The instantiation of a pattern enables the description of a concrete system that represents an implementation of the system type as described by the pattern. A context-pattern consists of (1) a method that describes the usage of a context-pattern; (2) a graphical representation of a context-pattern; and (3) templates with additional information about the elements that are contained in the graphical context-pattern [24].

3. Pattern-Based Context Analysis for Cloud Computing Services

In this section, we explain how to perform a context analysis using the RestAssured-Cloud System Analysis Pattern (ReAs-CSAP).

Section 3.1 presents the format of the ReAs-CSAP. The different elements of the ReAs-CSAP are discussed in Section 3.2. The instantiation of the ReAs-CSAP is discussed in Section 3.3. Section 3.4 describes how to use the information from an instantiated ReAs-CSAP.

3.1. The Format of the ReAs-CSAP

The following sections describe the format of the ReAS-CSAP where we comply with the pattern language format. The structure of the pattern format should always capture the following major parts: context, problem, solution and consequences [25].

3.1.1. Context

Note that this section concerns the context in which the ReAs-CSAP is applied. It does not refer to the context that is analyzed with help of the ReAs-CSAP.

The modelling of the context of a cloud computing service and its supporting cloud computing system represents the basis for a risk assessment (see Figure 1). Here, the context defines the scope and boundaries that are relevant for performing a risk assessment. Furthermore, the context provides necessary information that has to be considered during the risk assessment. By means of context information, the assets that are relevant for the risk analysis are derived. The further steps of the risk assessment are performed on these assets (see Section 2.3).

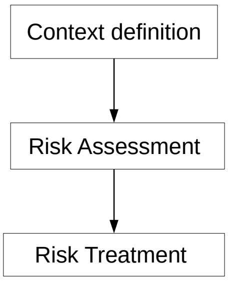


Figure 1. Risk assessment process.

3.1.2. Problem and Forces

As already mentioned in Section 3.1.1, a definition of the context is needed to perform a risk assessment of a system. This context definition has to represent information about the external and the internal contexts of a cloud computing service. With respect to the external context, it is necessary to identify all of the requirements that are indirectly relevant for the provided service and the corresponding system. Here, information related to legal, regulatory and contractual requirements has to be considered. The internal context has to consider the particular cloud computing service and its supporting system itself. The following information has to be specified:

- Relevant legal, regulatory and contractual requirements;
- Relevant internal stakeholders that are interacting with the provided service(s);
- Relevant locations;
- High-level primary assets, such assets represent business processes of an organization and information that is related to the business processes;
- High-level supporting assets, such assets support the execution of primary assets in the business processes by processing, storing and transmitting the related information (e.g., storage, software, network components);
- interfaces (cf. ISO/IEC 27005 [7], p. 12).

To get meaningful results from the risk analysis, it is necessary that the above-mentioned information regarding the external and internal system contexts is complete. It has to be ensured that all different types of context information are considered with respect to their relevance. If the context definition is performed without any further support, there is a risk that particular types of context information are not considered. Furthermore, the relations between different pieces of context information have to be defined, because during the assessment of particular risks, the corresponding interactions have to be considered.

3.1.3. Solution

Our solution is a pattern-based approach for analyzing the context of a cloud computing service. In Section 3.2, we present the solution in more detail.

3.1.4. Consequences

Our pattern allows the definition of the context of a cloud computing service. The pattern approach ensures that no relevant information is overlooked during the context definition. The definition of the pattern syntax by a meta-model enables the analysis and further processing of the information provided by the pattern. This meta-model allows the extension of the pattern by new elements. Thus, the pattern can be adapted to address future needs.

The graphical representation of the pattern facilitates the context definition for users who are not so familiar with that subject. Furthermore, it supports the identification of relationships between the different elements of the pattern.

By including elements that are relevant in the context of the GDPR, our pattern refers to the most important regulation for privacy and data protection in the European Union. To enhance the definition of the scope in conformance to ISO 27001, our pattern can be used in the context of an important widespread security standard.

3.2. Description of the ReAs-CSAP

The *RestAssured-Cloud System Analysis Pattern* provides solutions for the problems described in Section 3.1.2. The ReAs-CSAP provides model elements to define the different types of context information and their relations to cloud computing services. Thus, it can be ensured that neither the type of context information, nor the relevant relations between particular types of context information, are overlooked unintentionally during the context definition.

Because its model elements are defined by a meta-model, the ReAs-CSAP enables a terminology-based context definition. Additionally, the ReAs-CSAP provides a graphical representation of the corresponding model elements. Thus, a good overview of the specific context information is given and intuitive use of the ReAs-CSAP is made possible. Tool support for the definition of the ReAs-CSAP and its instantiation is provided by the *ClouDAT-tool* (see Section 5).

This subsection contains a detailed description of the elements provided by the ReAs-CSAP. The general rules for the instantiation of the ReAs-CSAP are considered in Section 3.3. In Section 3.4, the usage of the context information, represented by the ReAs-CSAP, is discussed. The instantiation of the ReAs-CSAP based on real-life cloud computing services is described in Section 4.

The ReAs-CSAP expands the CSAP from reference [8] with new types of context information that allow the representation of (1) particular information that is relevant for data protection, and (2) information regarding the compliance to laws and regulations for security and privacy information.

The ReAs-CSAP is shown in Figure 2. In the following, we discuss the different elements of the ReAs-CSAP. Here, the ReAs-CSAP elements are associated with the problems of a context definition from Section 3.1.2. Information about the internal context (see Section 3.1.2) is represented by the *direct environment*. The *direct environment* contains the *direct stakeholders* that are relevant for the considered cloud computing services and the *cloud* itself. A cloud is represented graphically by a grey box (see Figure 2). The cloud represents its different parts in the form of *cloud elements*.

Cloud elements specify the particular cloud computing services and the physical resources of the cloud. Graphically, the cloud elements are represented by white boxes inside the cloud (see Figure 2). The associations between the different cloud elements are also represented. Direct stakeholders are able to interact with cloud elements. The logical relationships between direct stakeholders are also considered within the ReAs-CSAP. The different types of direct stakeholders are the following:

Data subject is an identified or identifiable natural person who uses the considered cloud computing service. In this context, personally identifiable information and/or sensitive personal information (SPI) of the *data subject* is processed and/or stored by using the cloud computing service. The *data subject* represents an important role, because assuring the privacy and security of their data is a main goal.

Data controller is a provider of the considered cloud computing service in a form of Infrastructure as a Service (IaaS), Platform as a Service (PaaS) or Software as a Service (SaaS). Data controllers have contracts with the *data subjects* who use the provided cloud computing services. A *data controller* is legally responsible for the compliance of the specified privacy and security requirements for the data subject' data. To ensure this compliance, data controllers make use of data protection concepts and technologies. To make use of such technologies, *data controllers* can register to the *data protection platform*. Depending on the level of the provided cloud computing service, a *data controller* can use either their own IaaS and/or PaaS or an external IaaS and/or PaaS infrastructure provided by particular *cloud providers*.

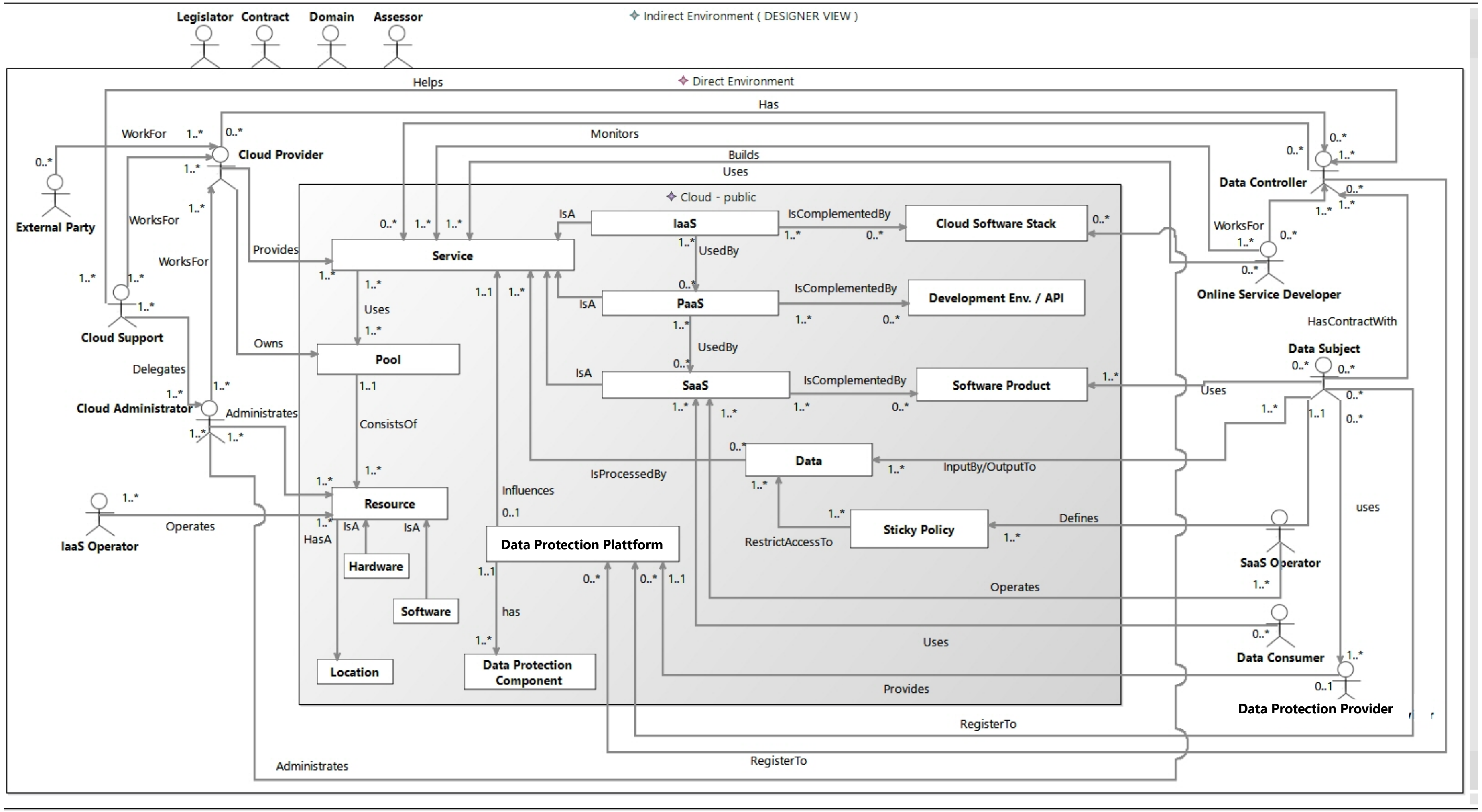


Figure 2. RestAssured-Cloud System Analysis Pattern.

Data consumers are natural or legal persons, public authorities, agencies or any other bodies. Besides being *data subjects*, they are also users of the considered cloud computing service. However, their use case can be different from the use case of the *data subjects*. Within this use case, the accessible data of data subjects is disclosed to data consumers. The rights for *data consumers* with respect to the access to and usage of specific data is defined in the corresponding *sticky policies*.

Cloud providers are legal entities that provide cloud computing services in the form of IaaS and/or PaaS that are used to provide an SaaS. In the case of *IaaS providers*, they also own the resources to provide this type of cloud computing service. *Cloud providers* can have associations with the other types of direct stakeholders who are working for *cloud providers* as follows:

- The optional *cloud support* works for the *cloud provider*. This represents the point of contact for *cloud customers* if they have questions or problems related to the used IaaS and/or PaaS cloud computing services. Possible problems are delegated to the *cloud administrators*.
- *Cloud administrators* work for *cloud providers*. They administrate the resources of the cloud as well as the cloud software stack and handle problems that have been reported by customers.
- *External parties* specify service providers that work for the *cloud provider*. Here, an external party delivers services that are relevant for the cloud or affect the operation of the considered cloud computing service. For example, external parties could be represented by companies for the maintenance of Information Technology (IT)-resources and air conditioning or cleaning services.
- *IaaS operators* perform tasks for the operation of an IaaS.

Online service developer specifies a legal person or organization that has developed the *software product* that is provided by the *data controller* via the corresponding SaaS. *Online service developers* work for the *data controller*. They need profound skills related to the technologies (e.g., Intel Software Guard Extensions (SGX) [26,27]) that are used by the *data protection platform*.

External data protection service providers host the components of the *data protection platform* in an according infrastructure and provide the functionality of these components as a service. This service can be used by *data controllers* that do not want to be in charge of hosting the *data protection components*.

SaaS operators perform tasks for the operation of an SaaS.

Beside the *direct stakeholders*, the *direct environment* also contains the *cloud* which, in turn, contains the different *cloud elements*. The *cloud* can represent a public, private, hybrid or community cloud. In the graphical representation of the ReAs-CSAP, the type of the cloud is displayed in the *cloud* beside the key word "cloud" (see Figure 2). The contained *cloud elements* are as follows:

- High-level primary assets in the form of business processes specified by (cloud computing) *services* (IaaS, PaaS, SaaS);
- High-level primary assets in the form of information specified by *data*;
- High-level supporting assets specified by *resources* (hardware, software, location);
- Data protection concepts and technologies (*sticky policies, data protection platform*).

The different types of *cloud elements* are the following:

Service defines a central point for referencing all provided cloud computing services. Data has an association with service, because the relevant data is processed by all provided cloud computing services. The *data protection platform* influences the processing of the *data* on the PaaS level.

SaaS represents a cloud computing service on the SaaS level. SaaS is complemented by the *software product(s)* whose functionality is provided via the SaaS. SaaS uses a PaaS for the provision of its service.

PaaS specifies a cloud computing service on the PaaS level that is used by the SaaS. PaaS is complemented by the *development environment and Application Programming Interface (API)* that is provided via the PaaS. PaaS uses the resources that are provided by an IaaS. A PaaS is provided by a *cloud provider*.

IaaS defines a cloud computing service on the IaaS level that provides the necessary infrastructure with different types of resources. Examples of such *resources* are hardware for storage and processing power. Resources are used directly by the PaaS and indirectly by the SaaS. IaaS is complemented by the *cloud software stack* and infrastructure resources like *software* and *hardware*. The IaaS and the cloud software stack are administrated by the *cloud administrator*.

Cloud software stack represents the cloud software stack that is necessary to provide the corresponding IaaS.

Development environment and API specifies the development environment and API that is provided by the according PaaS. The provided *development environment and API* are used to develop the *software product* that is provided by the considered SaaS. The API provides functionalities that enable the use of the relevant PaaS resources and IaaS resources by the software product without the need to know any specific technical details concerning these resources.

Software product represents the software that is provided by the *data controller* via the corresponding SaaS. The *software product* is developed by the *online service developer*. For the development, the *development environment and API* of the PaaS are used.

Pool is the central point for referencing all relevant physical resources of the cloud that are necessary for providing the appropriate cloud computing services.

Resource is the central point for referencing all resources in the form of *locations*, *software* and *hardware*.

Location represents all locations that contain cloud resources (e.g., the computing center) or are relevant to the provided cloud computing service in another way (e.g., development site).

Hardware represents necessary cloud hardware resources, e.g., server racks or network components.

Software represents cloud resources in the form of necessary software (e.g., software for managing the cloud or virtualization).

Data specifies the personally identifiable information and/or sensitive personal information of the *data subject* that is processed and/or stored by the according SaaS. The rights for accessing and using this data by *data consumers* are specified by (1) a legal specification by the contract between the data subject and the data controller and (2) a formal specification in the associated *sticky policies*.

Sticky policy defines requirements regarding the access and usage of the *data* of a *data subject*. They are derived from the *contract* between the *data subject* and *data controller*.

Data protection platform provides security and privacy mechanisms. These mechanisms constrain the PaaS level for enforcing the relevant *sticky policies* (e.g., in the transfer, processing and storage). The data protection platform might provide different *data protection components* that implement the appropriate security and privacy mechanisms. The different components can be hosted by either *data controllers* themselves in an according infrastructure or provided by *data protection providers* that provide the functionality of the components as a service.

Data protection component defines the components that implement security and privacy mechanisms for the considered cloud computing service. The *data protection component* could include concrete technologies for data protection, such as attribute-based encryption (e.g., reference [28]), fully homomorphic encryption (e.g., reference [29]) or secure enclaves (e.g., references [26,27]).

The *indirect environment* represents the external context of a cloud computing service. The relevant information about the external context is represented by different types of *indirect stakeholders* that are contained in the *indirect environment*. Here, the different types of *indirect stakeholders* represent the following information:

Legislator: Representation of the laws and regulations of legislators (e.g., Germany or the European Union) that are relevant for the cloud computing service. This type of *indirect stakeholder* is especially important, because it enables the representation of laws regarding data privacy. In the context of the European Union, the *GDPR* is especially relevant. Relevant *legislators* can be derived from the *location* of the considered cloud computing service.

Domain: Specification of domain-specific rules and guidelines that the cloud computing service has to comply with.

Contract: Representation of contractual provisions (e.g., service level agreements with customers) that have to be fulfilled by the cloud computing service. The representation of the contracts between data subjects and data controllers are particularly important because they specify the security and privacy requirements for the data of the data subjects. The *sticky policies* that specify these privacy requirements in a machine-readable way are derived from such *contracts*. It should be ensured that the content of a *contract* conforms to the relevant *legislators* and *domains*.

Assessor: Evaluate the level of security and/or privacy that is provided by the considered cloud computing service. The evaluation can be performed with respect to an existing standard, such as ISO 27001 [11]. Assessors can analyze the results of the performed risk assessment. Here, they can map information about the risk assessment to the real system. For example, they can check the implementation of countermeasures that reduce the levels of corresponding risks.

3.3. Instantiation of the ReAs-CSAP

The ReAs-CSAP specifies the context of a particular cloud computing service that is provided by a data controller. This cloud computing service represents the service that is the target of the risk assessment. It can be an IaaS, a PaaS or an SaaS. The ReAs-CSAP is instantiated according to the cloud computing service that is provided by the data controller. Additionally, all cloud computing services that are used by the cloud computing service of the data controller have to be considered. For example, if the data controller provides a SaaS, then the directly used PaaS and the indirectly used IaaS can also be instantiated.

The instantiation of the ReAs-CSAP has to be performed for every type of data subject that uses the cloud computing service provided by the data controller. Only the instantiated ReAs-CSAP-elements are relevant for the context specification. An instantiated ReAS-CSAP-element can be identified by its name that is followed by its instance type. The instance type is enclosed in angle brackets. If the name of an instantiated ReAs-CSAP-element is equal to its instance type (i.e., the same name as in the pattern), the element is relevant for subsequent risk analysis, but the context analyzer does not consider a concrete implementation of the element. For example, in Figure 3, the *data protection component* is instantiated, but no concrete implementation is considered during context analysis. Such an element is refined to concrete implementations later on.

During the instantiation of the ReAs-CSAP, the following rules should be considered:

- Instances of *legislators* can be derived from the *location(s)* of the cloud system;
- The instances of *contracts* have to conform to the instances of *legislator* that represent particular laws, regulations, etc.;
- Instances of *sticky policies* for the *data subject*'s data are derived from the corresponding instances of *contract*. All contract instances that indicate a legal contract between the data subject and data controller are relevant. These contracts serve as a basis for privacy and security requirements and reference the relevant data of the *data subject*;
- All cloud computing services that are used directly and/or indirectly by the considered cloud computing service have to be instantiated;
- If the *data controller* is not in charge of providing the particular components of the *data protection platform*, the assigned external *data protection provider* should be instantiated.

No scalability issues are to be expected when using the ReAs-CSAP, even in complex cloud scenarios. This is due to the fact that only a small number of instances for the various elements of the ReAs-CSAP will be relevant. Therefore, the number of instantiated stakeholders and cloud elements can be expected to be always manageable.

3.4. Usage of the ReAs-CSAP Information for Risk Assessment

This section describes how the information that is represented in an instantiated ReAs-CSAP is used for analysis.

The primary assets are represented as follows:

- Business processes are represented by instances of the cloud elements with the instance types *service*, *IaaS*, *PaaS* and *SaaS*;
- Informational assets are represented by instances of the cloud elements with the instance type *data*;
- Direct stakeholders are represented by the instance types *data subject* and *data controllers*.

These primary assets are represented on a relatively high level of abstraction. If necessary, these assets should be refined during the risk analysis. For example, a cloud computing service (e.g., SaaS) can be refined into processes and subprocesses that are necessary for providing the considered service. By means of a cloud computing service and its processes, the identification of further informational assets in the form of data can be performed. The data specified in an instance of the ReAs-CSAP can be refined into a set of more specific data during the risk analysis. Supporting assets are represented by cloud elements of the following instance types:

- *Pool*
- *Resources*
- *Hardware*
- *Software*
- *Location*
- *Cloud software stack*
- *Development environment/API*
- *Software product*
- *Sticky policy*.

The supporting assets that are represented in the ReAs-CSAP should also be refined during risk assessment. The cloud elements of the instance types *hardware* and *software* are usually instantiated. In most cases, these instantiations of hardware and software elements consider no concrete implementation during context analysis (only the category type is assigned). However, they can be further refined during the subsequent risk analysis. Here, the respective *hardware* and *software* instances represent a category of supporting assets. The *hardware* instance, for example, could be refined into *hardware* instances that represent servers, network components or personal computers. During the refinement, new *hardware* and *software* instances can be identified by considering the hardware and software that supports the provision of the relevant cloud computing services in the ReAs-CSAP.

The instances of *indirect stakeholders* of the types *legislator*, *domain* and *contract* provide information for the assessment of the values of the assets. This assessment determines the impact in cases where the privacy or security property of an asset is compromised. For example, the compromising of a particular privacy property of data or a data subject can lead to legal penalties according to the GDPR.

4. Industrial Application Examples

This section presents the instantiation of the ReAs-CSAP for three real-life cloud computing services that were studied in the RestAssured project. Section 4.1 explains the instantiation of the ReAs-CSAP for the volunteer service *Ami*. Section 4.2 presents an instance of the ReAs-CSAP for an extension of the *Ami volunteer service* with the *SCANT (Social Care Analysis of Needs Tool) Ami volunteer service*. The online automotive insurance service *Pay-as-you-Drive Insurance* is explained in Section 4.3.

4.1. Context Definition for the Ami Volunteer Service Use Case

The use case of this section is the *Ami volunteer service* which is provided as an SaaS. This SaaS enables the use of the web-based application *Ami* that matches service-providing volunteers (*Ami volunteers*) with people requiring help (*Ami clients*).

Figure 3 shows the instantiation of the ReAs-CSAP for the *Ami volunteer service* that is the target of the risk analysis. Here, the volunteer service is represented by the cloud element instance *volunteer service* of the *SaaS* type. This service provides the functionality of the *Ami web application* that is instantiated by the cloud element *Ami web application* of the type *software product*. The data controller *Oxford Computer Consultants* (*OCC*) (https://www.oxfordcc.co.uk) who provides the volunteer service is specified by the particular instance of the direct stakeholder of the type *data controller*.

The application was developed by *OCC developers* by using the *Azure development environment/API* (https://azure.microsoft.com/en-us/) that is provided by the *PaaS*-cloud element *Azure PaaS*. The provider of this PaaS is *Microsoft* (https://www.microsoft.com/en-us/). This cloud provider also provides the *pool* of *resources* to run the PaaS via the IaaS *Azure IaaS* that is defined by the instantiated *IaaS*-cloud element. The roles of Microsoft as an IaaS and PaaS provider are represented by the same direct stakeholder instance of the type *cloud provider*. The location of the IaaS is instantiated by the cloud element *location* with the country being the *UK* (United Kingdom). The cloud elements of the types *pool*, *resources*, *hardware* and *software* have been instantiated because they are relevant to the context of the *Ami volunteer service*. Refinement of these cloud elements was performed during risk analysis.

The *Ami volunteers* are the *data subjects*. They are defined by an instance of the direct stakeholder of the corresponding type. The *data subjects* provide their personal data via the *Ami web application* to the *Ami volunteer service*. This data is represented by the cloud element instance *volunteer data* of type *data*. The instantiated *sticky policy* cloud element *volunteer data policy* restricts the access and usage of the *volunteer data*. The *volunteer data* is consumed by the *data consumers* in person of the direct stakeholders *Ami clients* and the *volunteer organization* that also represent users of the *Ami volunteer service*. The enforcement of the *volunteer data policies* is realized by the cloud element instance *RestAssured platform*. The *RestAssured platform* is an instance of type *data protection platform*.

The relevant cloud elements that represent the *data protection platform* and its contained *data protection components* are instantiated. The *data protection components* are instantiated with the same name as their instance types in the pattern. This instantiation specifies that *data protection components* are used in the context of the *Ami volunteer service*. However, no further specifications related to the implementation of the actual *data protection components* are made. The actual implementation of the *RestAssured platform* and its *data protection components* is specified during risk analysis by performing a refinement. The functionality of the *RestAssured platform* provides a service by an *external RestAssured provider*. The data controller, *OCC*, and the data subjects, *Ami clients*, have to register to the *RestAssured platform*.

The *Ami volunteer service* is provided and the cloud infrastructure is located in the United Kingdom; hence, the legislations of the United Kingdom and the European Union are relevant. They are instantiated by the indirect stakeholder of type *legislator*. The GDPR is the relevant instance of the *legislator* for the European Union. *Social service* specifies requirements for social services. *Social service* is the instance of the *domain* as an indirect stakeholder. The indirect stakeholder *contract* is instantiated with the *volunteer contract*, representing the contract between a *data subject* and the *data controller*. This contract specifies restrictions for the access to and the usage of the volunteers' data (*volunteer data*) for the data consumers of the *Ami clients* and the *volunteer organization*. The *sticky policies* are derived from this contract.

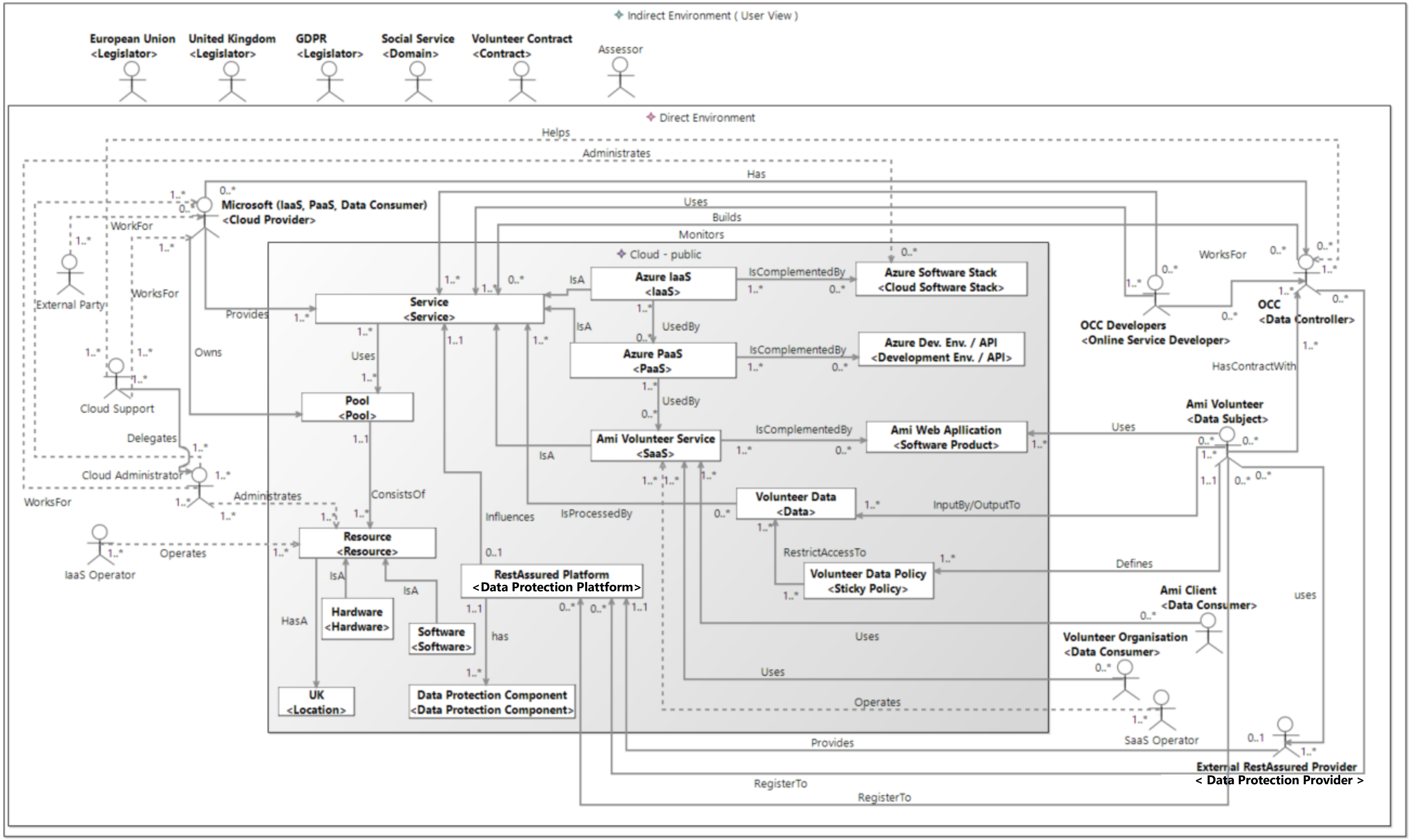

Indirect Environment (User View)
European Union <Legislator>
United Kingdom <Legislator>
GDPR <Legislator>
Social Service <Domain>
Volunteer Contract <Contract>
Assessor
Direct Environment
Helps
Administrates
Has
Uses
Builds
Monitors
Microsoft (IaaS, PaaS, Data Consumer) <Cloud Provider>
WorkFor
External Party
WorksFor
Provides
Owns
Cloud Support
Delegates
Cloud Administrator
Administrates
WorksFor
IaaS Operator
Operates
Cloud - public
Service <Service>
Azure IaaS <IaaS>
Azure Software Stack <Cloud Software Stack>
IsComplementedBy
IsA
UsedBy
Azure PaaS <PaaS>
Azure Dev. Env. / API <Development Env. / API>
Ami Volunteer Service <SaaS>
Ami Web Apllication <Software Product>
Uses
Pool <Pool>
ConsistsOf
Resource <Resource>
HasA
Hardware <Hardware>
Software <Software>
UK <Location>
Influences
IsProcessedBy
Volunteer Data <Data>
InputBy/OutputTo
RestrictAccessTo
Volunteer Data Policy <Sticky Policy>
Defines
RestAssured Platform <Data Protection Plattform>
has
Data Protection Component <Data Protection Component>
OCC Developers <Online Service Developer>
OCC <Data Controller>
WorksFor
HasContractWith
Ami Volunteer <Data Subject>
Ami Client <Data Consumer>
Volunteer Organisation <Data Consumer>
uses
SaaS Operator
Operates
Provides
RegisterTo
RegisterTo
External RestAssured Provider < Data Protection Provider >

Figure 3. Instantiation of the ReAs-CSAP for the *Ami volunteer service* use case.

4.2. Context Definition for the SCANT Ami Volunteer Service Use Case

This section describes the *SCANT Ami Volunteer Service* which represents an extension to the *Ami volunteer service* described in Section 4.1. Compared to the *Ami volunteer service*, the *SCANT Ami volunteer service* is complemented, in addition to the *Ami web application*, by another component, represented by the *SCANT*. *SCANT* enables local authorities to perform queries on the personal data of *Ami clients* for the purpose of data analysis. The goal of this data analysis is to gain information about social issues. The *Ami clients* can restrict or refuse these queries on their personal data by a corresponding sticky policy.

Figure 4 presents the instantiation of the ReAs-CSAP for the *SCANT Ami volunteer service* use case. This use case is an extension of the *Ami volunteer service* (see Section 4.1). Therefore, only the differences compared to the *Ami volunteer service* use case are described.

The *volunteer service* is an instance of an *SaaS* cloud element. It is the target of the risk analysis. Compared to the instantiation of the ReAs-CSAP in Figure 3, the volunteer service is complemented by *SCANT*. *SCANT component* is an instance of the *software product*. The *data subjects* are represented by the *Ami clients*, because the *Ami client data* is generally designated to be accessed by SCANT. In addition to the *volunteer organization*, *local authorities* represent another *data consumer* who wants to get access to the *Ami client data* via SCANT. The access to the *Ami client data* is restricted by the *Ami client data policy* that is derived from the *Ami client contract*. The *Ami client contract* is established between the *Ami clients* and *OCC*. OCC has the role of *data controller*. The SCANT software is developed by the *OCC developers*.

4.3. Context Definition for the Pay-As-You-Drive Insurance Use Case

The *Pay-as-you-Drive Insurance* (PAYD) use case represents a cloud computing service for automotive insurances. PAYD enables insurers to offer innovative, cost effective and usage-based automotive insurance products. This is achieved by recording and analyzing the driving data of the insurance customers. Because this driving data represents the personal data of the insurance customers, the requirements concerning data privacy and data security are very high.

The instantiation of the PAYD use case is shown in Figure 5. PAYD is a SaaS cloud computing service; hence, it is instantiated by a cloud element of type *SaaS*.

The PAYD service is provided by the respective insurers. Accordingly, these insurers take the *data controller* role. The PAYD service is complemented by two *software products*.

The first *software product* is the *PAYD web application* that is developed by *Adaptant*. This web application is used by the *data subjects* who are *insurance customers*. By using the *PAYD web application*, the insurance customers are able to register to an insurance product and get information related to their current insurance conditions. The *insurance customers* provide their personal data as *insurance customer data* to the *PAYD web application*. The access, processing and storage of the *insurance customer data* are restricted by an appropriate *sticky policy*. The *customer data policy* is an instance of a *sticky policy*.

The second *software product* is the *telematic component*. This component is responsible for receiving the *driving data* of the insurance customers. The received driving data is delivered to the cloud infrastructure of the corresponding insurer. If insurance customers give their consent, their driving data can be accessed by *telematics analysts* for the purpose of performance diagnostics. Accordingly, *telematics analysts* represent one type of *data consumer*. The second instance of *data consumer* is the *insurance analysts*. They analyze the *driving data* to get information that is relevant in the context of the provided automotive insurance offers. The *driving data policy* refers to the *driving data*. This *sticky policy* restricts the amount of telematic data that is represented by the *driving data*. This sticky policy also restricts the access, processing and analysis of the *driving data*. The *customer data policy* as well as the *driving data policy* are derived from the *insurance contract* between the *insurer* and the *insurance customers*. The *insurance contract* should conform to the *GDPR*. The *GDPR* is an instance of the indirect stakeholder *legislator*.

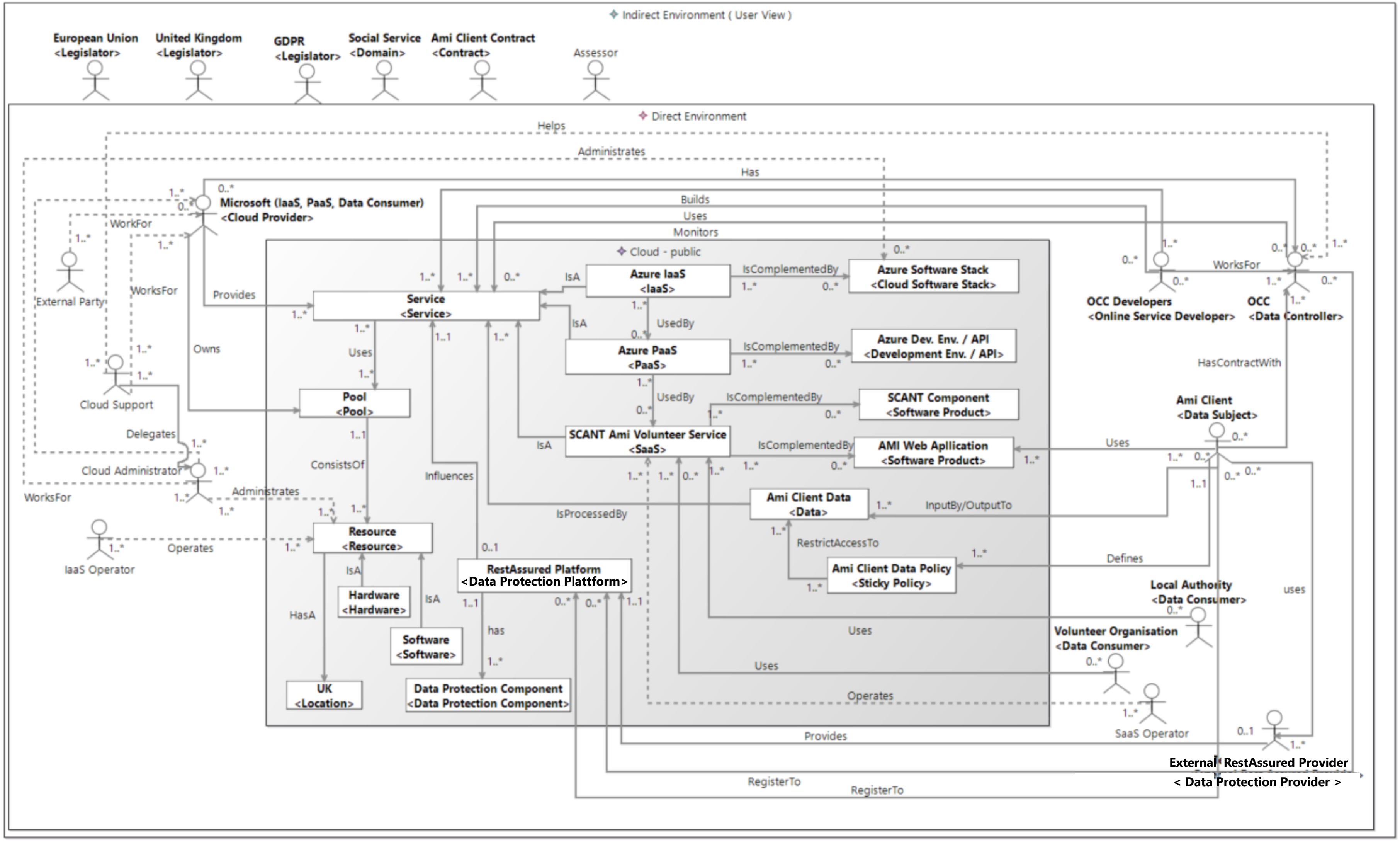

Indirect Environment (User View)
European Union <Legislator>
United Kingdom <Legislator>
GDPR <Legislator>
Social Service <Domain>
Ami Client Contract <Contract>
Assessor
Direct Environment
Helps
Administrates
Has
Builds
Uses
Monitors
Microsoft (IaaS, PaaS, Data Consumer) <Cloud Provider>
WorkFor
WorksFor
External Party
Provides
Owns
Cloud Support
Delegates
Cloud Administrator
Administrates
WorksFor
IaaS Operator
Operates
Cloud - public
Service <Service>
Azure IaaS <IaaS>
Azure Software Stack <Cloud Software Stack>
IsComplementedBy
IsA
UsedBy
Azure PaaS <PaaS>
Azure Dev. Env. / API <Development Env. / API>
SCANT Component <Software Product>
SCANT Ami Volunteer Service <SaaS>
AMI Web Aplication <Software Product>
Pool <Pool>
ConsistsOf
Uses
Influences
IsProcessedBy
Ami Client Data <Data>
RestrictAccessTo
Ami Client Data Policy <Sticky Policy>
InputBy/OutputTo
Defines
Resource <Resource>
Hardware <Hardware>
Software <Software>
HasA
IsA
UK <Location>
RestAssured Platform <Data Protection Plattform>
has
Data Protection Component <Data Protection Component>
OCC Developers <Online Service Developer>
OCC <Data Controller>
HasContractWith
Ami Client <Data Subject>
Local Authority <Data Consumer>
Volunteer Organisation <Data Consumer>
SaaS Operator
Operates
Provides
RegisterTo
External RestAssured Provider < Data Protection Provider >
uses
1..* 0..* 1..1 0..1

Figure 4. Instantiation of the ReAs-CSAP for the use case *Ami volunteer service* with the additional *SCANT component*.

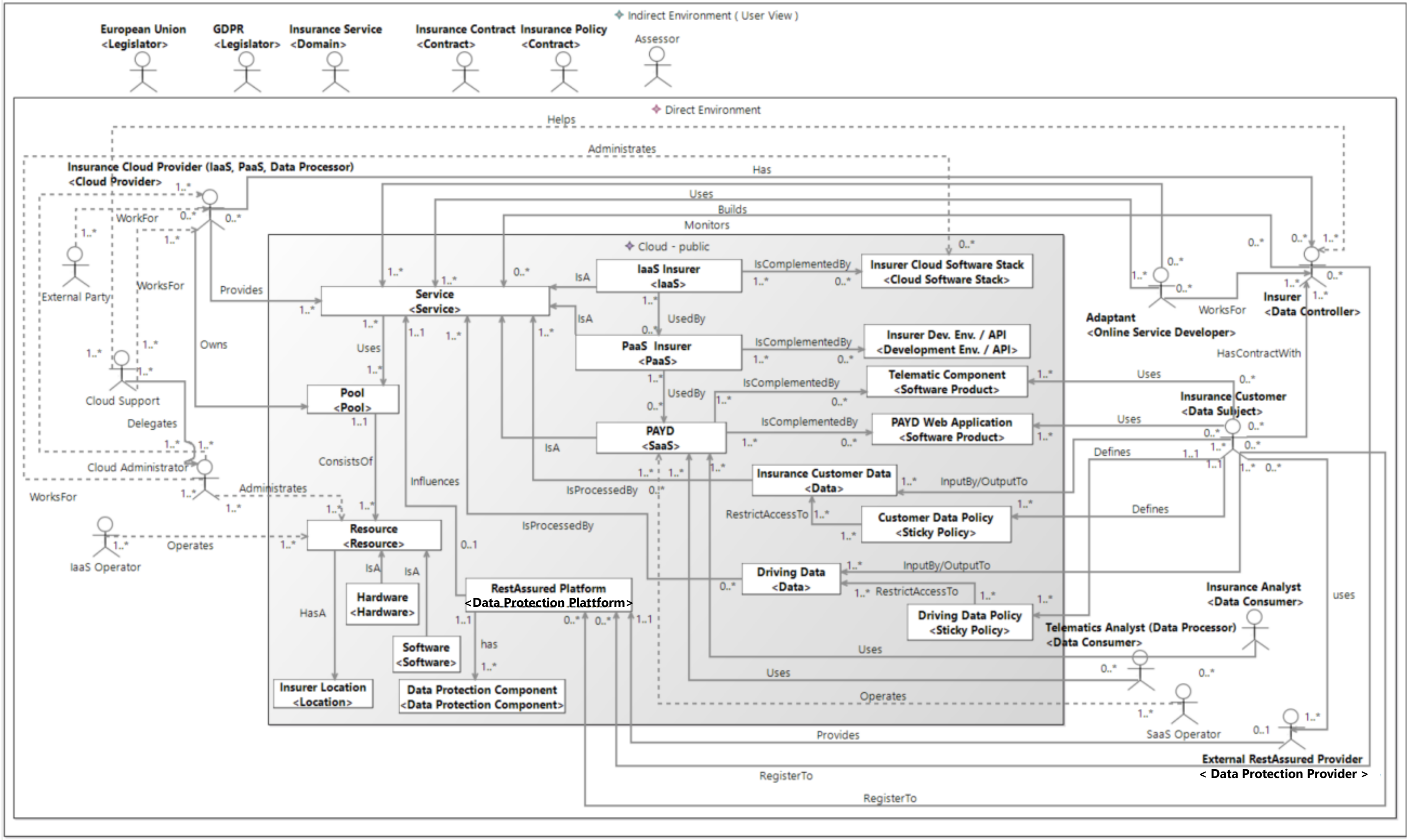

Indirect Environment (User View)
European Union <Legislator>
GDPR <Legislator>
Insurance Service <Domain>
Insurance Contract <Contract>
Insurance Policy <Contract>
Assessor
Direct Environment
Helps
Administrates
Has
Uses
Builds
Monitors
Insurance Cloud Provider (IaaS, PaaS, Data Processor) <Cloud Provider>
WorkFor
WorksFor
External Party
Provides
Owns
Cloud Support
Delegates
Cloud Administrator
Administrates
WorksFor
IaaS Operator
Operates
Cloud - public
Service <Service>
IaaS Insurer <IaaS>
PaaS Insurer <PaaS>
PAYD <SaaS>
IsA
UsedBy
IsComplementedBy
Insurer Cloud Software Stack <Cloud Software Stack>
Insurer Dev. Env. / API <Development Env. / API>
Telematic Component <Software Product>
PAYD Web Application <Software Product>
Insurance Customer Data <Data>
Customer Data Policy <Sticky Policy>
Driving Data <Data>
Driving Data Policy <Sticky Policy>
IsProcessedBy
RestrictAccessTo
InputBy/OutputTo
Pool <Pool>
ConsistsOf
Resource <Resource>
Hardware <Hardware>
Software <Software>
Insurer Location <Location>
HasA
Influences
RestAssured Platform <Data Protection Plattform>
has
Data Protection Component <Data Protection Component>
Adaptant <Online Service Developer>
Insurer <Data Controller>
HasContractWith
Insurance Customer <Data Subject>
Defines
Insurance Analyst <Data Consumer>
Telematics Analyst (Data Processor) <Data Consumer>
SaaS Operator
External RestAssured Provider < Data Protection Provider >
uses
Provides
RegisterTo

Figure 5. Instantiation of the ReAs-CSAP for the use case *PAYD*.

The *PaaS* and *IaaS* that are used in the context for providing the PAYD service are also provided by the respective insurers. Accordingly, the insurers are the *cloud providers* for PaaS and IaaS and own the *resources* of the *pool* that represent the cloud infrastructure. The *cloud provider*, *IaaS*, *PaaS* and its complementing components are instantiated generally for insurers.

The *RestAssured platform* and its *data protection components* are provided by an *external RestAssured provider*.

5. The CSAP Tool

We provide tool support for our pattern, namely with the CSAP tool. This tool provides a graphical editor that supports the creation and instantiation of Cloud System Analysis Patterns (CSAP). To this end, the tool provides two editors, namely the designer editor and the user editor. These editors are explained in the following sections.

5.1. The Designer Editor

The *designer editor* of the CSAP tool enables designers to create specific Cloud System Analysis Patterns. The designers can build a specific cloud system analysis pattern in two different ways:

- By extending the original CSAP [8] (see Figure 6) or;
- By creating an empty CSAP that contains only an *indirect environment*, a *direct environment* and a *cloud*.

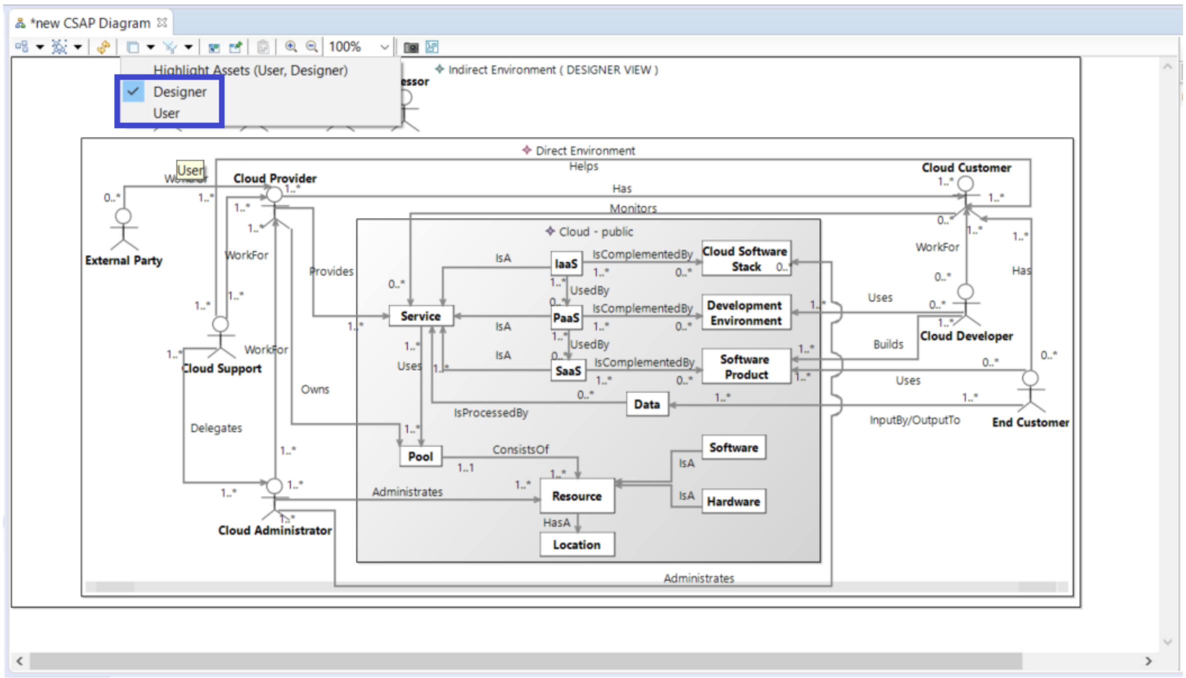
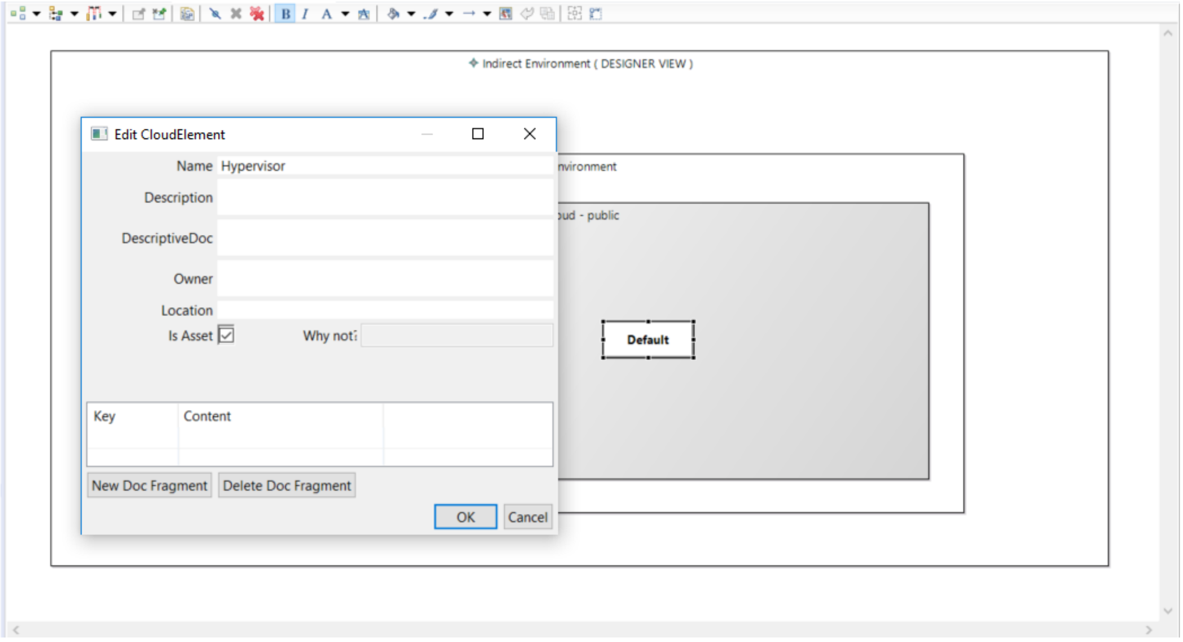


Figure 6. Editor mode of the CSAP tool.

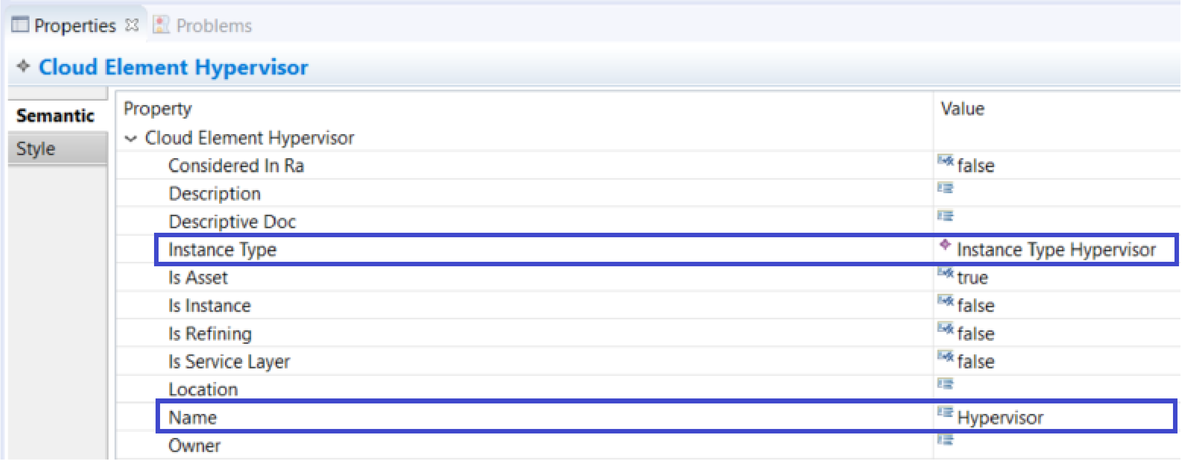
In cases starting with an empty CSAP, designers can create their own CSAP by adding appropriate types of *indirect stakeholders*, *direct stakeholders* and *cloud elements*. Furthermore, associations between *direct stakeholders* and *cloud elements* as well as among direct stakeholders themselves can be created. If the original CSAP is used as a basis, already existing CSAP elements can be modified and/or deleted.

Figure 7 shows the definition of a new *cloud elements* that is added to an empty CSAP. In the designer editor, the name of a CSAP element can be as the same as its instance type (see Figure 8).



Name Hypervisor
Description
DescriptiveDoc
Owner
Location
Is Asset ☑ Why not:

Figure 7. Adding a cloud element and defining its name.



Cloud Element Hypervisor

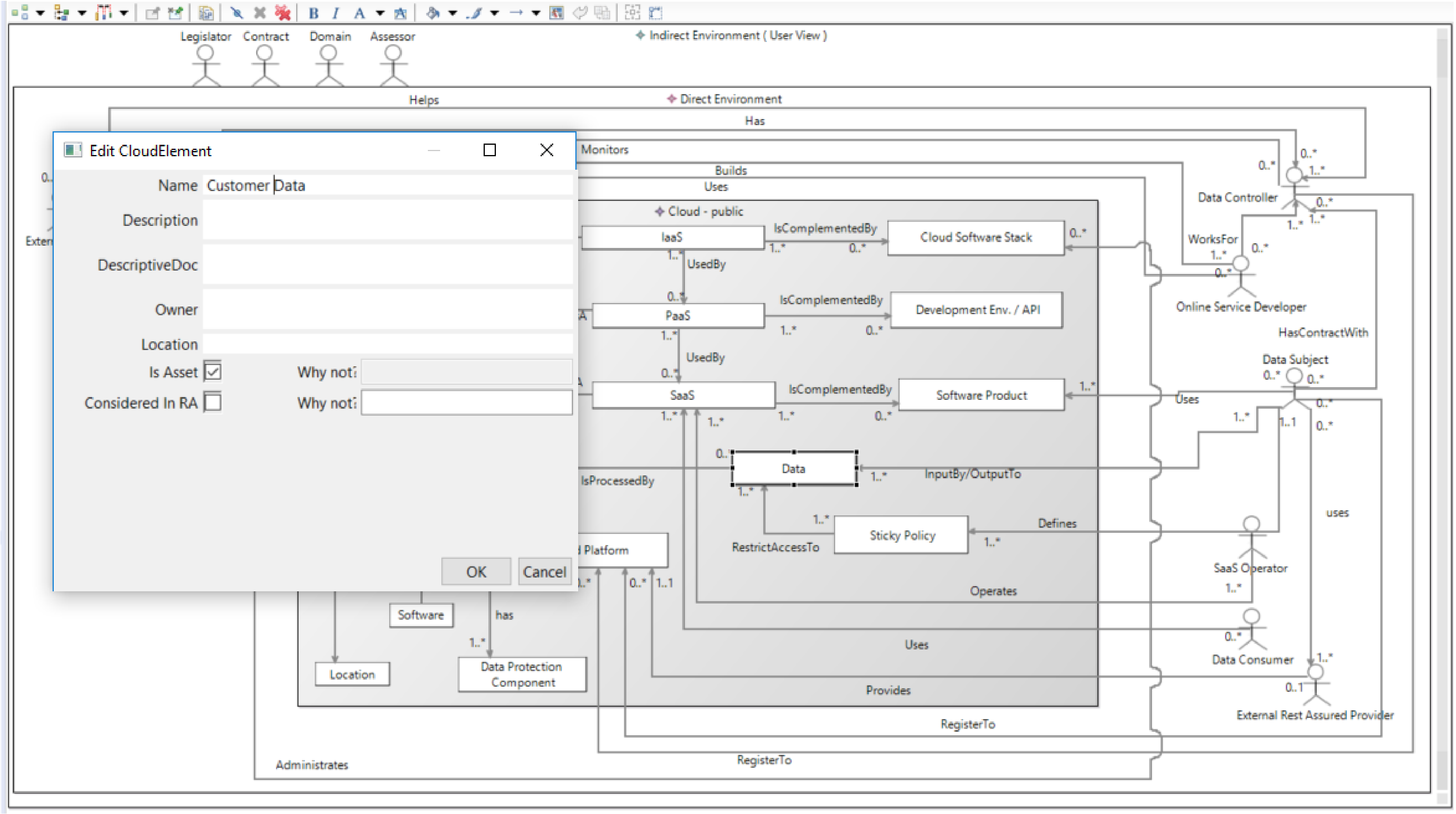
Property	Value
Cloud Element Hypervisor	
Considered In Ra	false
Description	
Descriptive Doc	
Instance Type	Instance Type Hypervisor
Is Asset	true
Is Instance	false
Is Refining	false
Is Service Layer	false
Location	
Name	Hypervisor
Owner	

Figure 8. Equality of the name and instance type of an added cloud element.

5.2. The User Editor

In the *user editor*, any defined pattern can be instantiated. The instantiation of our ReAs-CSAP is described in Section 3.3. Figure 9 shows the instantiation of a *cloud element* of instance type *data* during the instantiation of the ReAs-CSAP. This cloud element instance specifies customers' data and therefore, is named *customer data*.

The instantiation of the *cloud element* is marked by displaying the type name *data* surrounded by angle brackets under the name of the cloud element (see Figure 10). For a subset of the properties of the instantiated cloud element, the corresponding values can be specified in the according dialogue during its instantiation (see Figure 9). For all properties except the *instance type* the corresponding values can be assigned in a property panel (see Figure 11).



Edit CloudElement

Name	Customer Data		
Description			
DescriptiveDoc			
Owner			
Location			
Is Asset	☑	Why not:	
Considered In RA	☐	Why not:	

OK Cancel

Figure 9. Instantiation of a cloud element of the data instance type in a ReAs-CSAP.

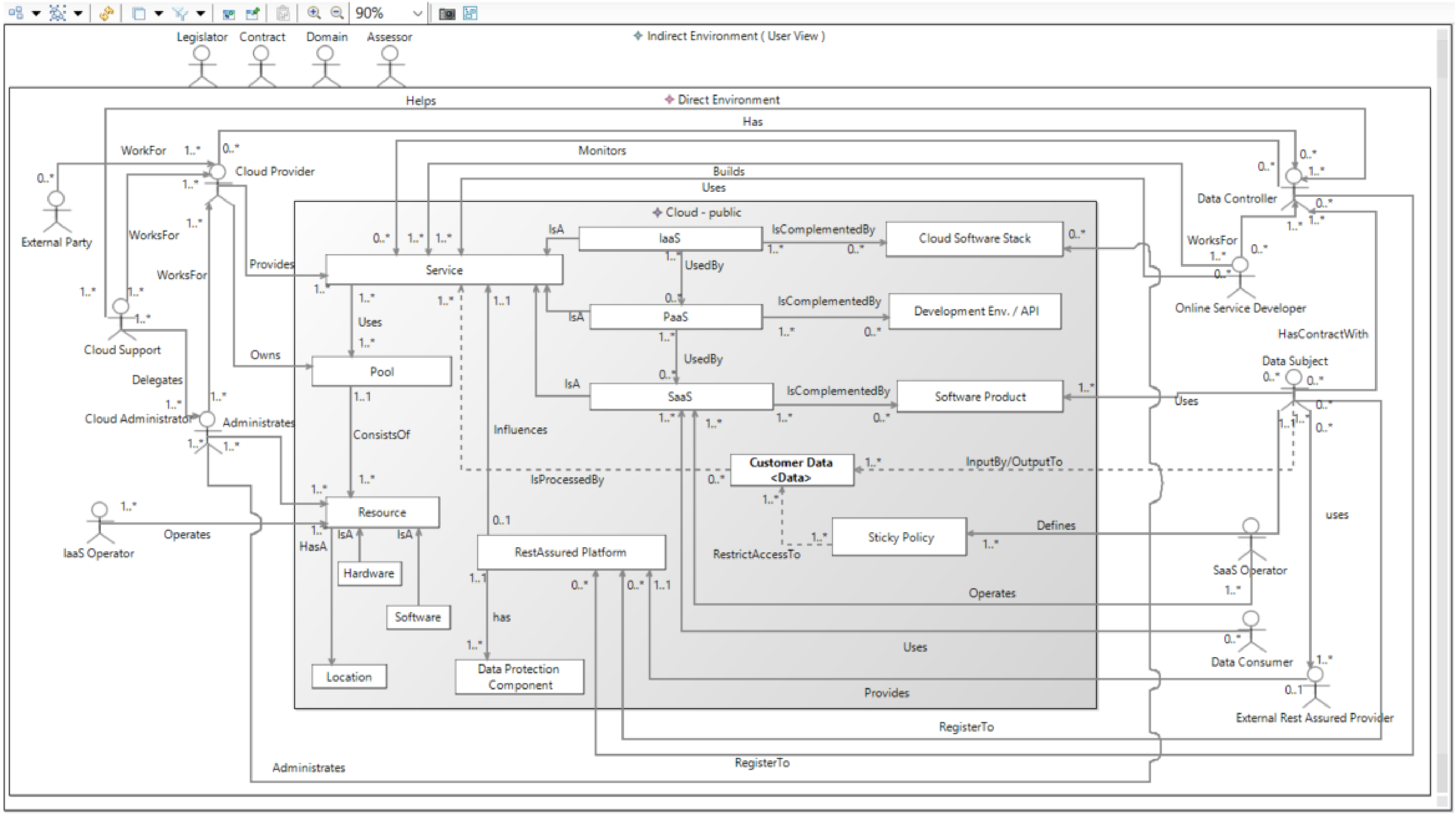
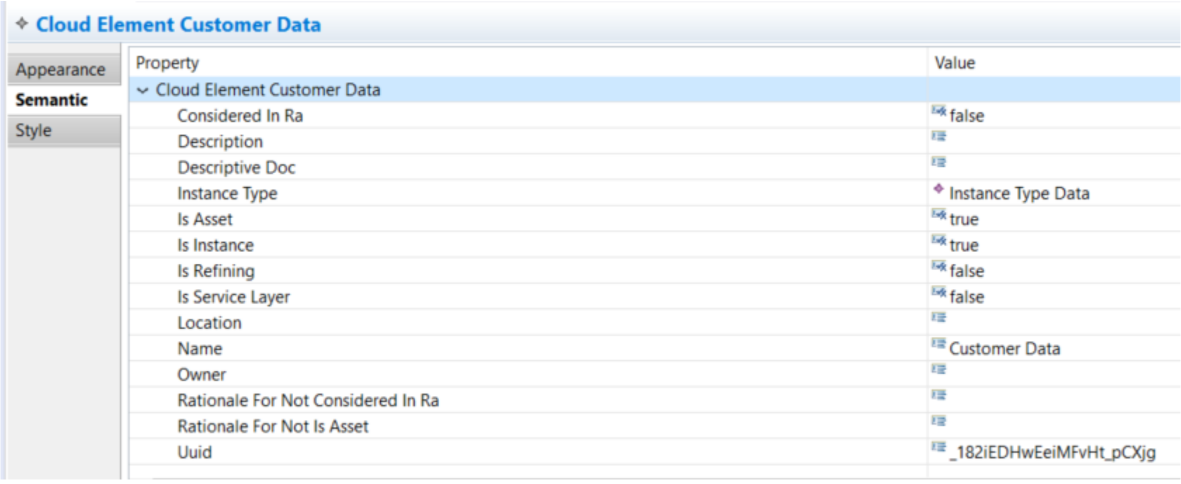


Figure 10. Representation of the instantiated cloud element of the data instance type in a ReAs-CSAP.



Property	Value
Cloud Element Customer Data	
Considered In Ra	false
Description	
Descriptive Doc	
Instance Type	Instance Type Data
Is Asset	true
Is Instance	true
Is Refining	false
Is Service Layer	false
Location	
Name	Customer Data
Owner	
Rationale For Not Considered In Ra	
Rationale For Not Is Asset	
Uuid	_182iEDHwEeiMFvHt_pCXjg

Figure 11. Definition of properties of the instantiated cloud element of the data instance type.

6. Related Work

There are some works in requirement engineering that have dealt with security at the early stages of system design. For instance, Chung provided a non-functional requirement framework [30]. In his work, security is a class of non-functional requirement. To model security aspects in Unified Modeling Language (UML) (http://www.uml.org), use case models have been extended, for example by misuse cases [31] and abuse cases [32]. Abuse frames [33,34] extend Jackson's problem frames [19] to model security aspects. As a supplement to the above-mentioned works, our pattern complements them with a tighter integration of context analysis prior to the execution of risk assessment or requirement analysis methods at design time. This may also guarantee better conformance to standards as well as business and IT alignment.

Fredriksen et al. [35] proposed the CORAS framework that supports a model-based risk management process. Prior to applying this process or any other risk assessment method, our pattern can be applied to establish the scope for risk management in the domain of cloud computing services.

Naudet et al. [36] proposed a security requirements engineering process that consists of the following four steps: context analysis and asset identification, security goal determination, refinement of these goals to security requirements, and countermeasure selection. Complementary to requirements engineering methods like that of reference [36], our pattern-based approach provides a systematic and structured execution of the context analysis. Our pattern can enhance the first step of the security requirements engineering process provided by Naudet et al. [36].

Haley et al. [37] proposed a framework which unifies functional requirements from requirements engineering and assets as well as threats from security engineering. The focus of this work was the transformation of assets and threats into constraints of functional requirements. In the provided framework, the security requirements are identified based on the system context. The system context is specified using Jackson's problem diagrams [19]. In contrast to this usage of problem diagrams, our pattern uses a specific graphical notation for the definition of a system context that is oriented on the UML.

Fenz et al. [38] presented an ontology-based framework for preparing ISO 27001 audits. They provided a rule-based engine which uses a security ontology to determine if the security requirements of a company are fulfilled. This work has no impact on ours, because we focused on different aspects of risk management.

Reusable patterns for security requirements have been studied by several researchers. For instance, Kis defined anti-patterns [39] for the identification of vulnerabilities and security requirements analysis at the business level. This work has no impact on our pattern, because our pattern does not represent an anti-pattern and has a different purpose.

With the focus on software systems, Sindre et al. proposed a reuse-based approach to determine security requirements [40]. Firesmith also provided reusable templates to specify security requirements [41]. In this context, reference [40] as well as reference [41] provide asset-based risk-driven procedures for the identification of security requirements. These procedures contain the identification of valuable assets for which no context for asset identification is defined. Our pattern supports the definition of a context that is a good basis for asset identification. Hence, our pattern can be used complementarily to these methods.

Fernandez et al. [13] provided a unified way of representing all the components of a cloud ecosystem as well as security aspects. Our pattern supports the specification of the indirect and direct environments of a cloud system as well as its stakeholders in addition to the cloud system itself.

In other work, Fernandez et al. [42] provided a method for building a security reference architecture for cloud systems. Furthermore, this paper presented a reference architecture for cloud computing environments using a class diagram. This class diagram represents the cloud services, the cloud and its components as well as cloud consumers and administrators. Our pattern considers the direct and indirect environments of a cloud system as well. Furthermore, all relevant direct stakeholders and their interactions with the cloud and among each other are represented. Specific types of direct stakeholders constitute roles that conform to the GDPR. Furthermore, our pattern can be used for the definition of the scope for risk management with respect to ISO 27005.

Additionally to the security requirement patterns, several design patterns for security design aspects have been developed to specify security measures, such as those in references [17,43,44]. A design pattern addresses a particular category of security concerns. The design patterns are generally used after the requirements engineering phase. In contrast, our work specifically focused on identifying relevant contextual information and defining the scope for a risk assessment, especially during the requirements engineering phase. We focused on analyzing the context and defining the scope for risk analysis and assessment; hence, the above-mentioned patterns can be used complementarily to ours.

Research on reusable patterns for context analysis has also received attention recently. Beckers et al. [45] provided a cloud system analysis pattern for establishing the context of cloud computing systems, expressed in a UML-like notation. In another work, Beckers et al. [8] presented a method for eliciting security requirements for cloud computing systems. They defined a terminology for the Cloud System Analysis Pattern (CSAP) from reference [45] with a meta-model. A concept for textual security requirement patterns that refer to the CSAP was also provided. We built our approach upon this work and presented more specific pattern elements for addressing data protection goals in the context of cloud computing services. For this purpose, our pattern extends the CSAP from reference [8] by adding elements that represent roles as well as concepts and technical components regarding privacy concerns. The added roles conform to the GDPR. Furthermore, we provide information concerning the logical relations of pattern elements that should be considered during the instantiation of the pattern.

In another paper, Beckers et al. [46] provided a catalog of security requirement patterns that are relevant for cloud computing. To order these security requirement patterns, different domains were defined. Furthermore, a structure for the representation of the security requirement patterns was represented. Because our pattern focuses on context definition, it could be used complementarily to this approach.

Lyubimov et al. [47] presented a framework that supports eliciting the requirements for the realization of an ISMS conforming to ISO 27001. Their framework provides representations of different requirements from the ISO 27001 using models. One model considers the requirements with respect to the definition of the ISMS scope in an abstract way. As a specialization, our pattern focuses on the definition of a scope that is specific for cloud computing services. It supports the scope definition on a lower degree of abstraction and provides concrete types of information that should be considered during the scope definition. Thus, our pattern could support the above-mentioned framework in the domain of cloud computing.

Montesino et al. [48] investigated the possible automation of controls that are listed in ISO 27001 and NIST SP800-53 [49] (catalog of security controls from National Institute of Standards and Technology (NIST)). This work considers the planning phase (from ISO 27001) of an ISMS. In contrast, our pattern focuses on the definition of a context. It can be used to define the scope of an ISMS for a cloud computing service during the planning phase of an ISMS. Accordingly, our pattern can be used complementarily to the approach of Montesino et al.

Schmidt [50] presents a threat and risk-driven methodology to security requirements engineering. He derives threat and risk models from the environment of a secure information system. This work is, therefore, complementary to ours.

Ismail et al. [51] provided a framework for analyzing the security transparency of cloud systems. Their framework considers the context of cloud systems from the conceptual view, organizational level and technical level. The concepts for analyzing security transparency on the organizational level were defined by a meta-model. These concepts contain, among others, the identification of assets that belong to the customers of a cloud computing service. As a supplement, our pattern supports the definition of the scope and boundaries of a cloud computing service that serves as a basis for asset identification. Hence, the above-mentioned framework could make use of our pattern.

Surridge et al. [52] presented a risk management method. This method supports the automated identification of threats and controls. Our approach can provide input to such a risk assessment method. Hence, their method can be used complementarily to our context analysis. A combination of both methods has been demonstrated in the context of the RestAssured project. Further detail related to the complementary methodology can be found in reference [53].

7. Conclusions and Future Work

Risk management is a crucial activity in the development of cloud computing services that respect data protection. It is also recognized as one of the most important tasks in the requirement engineering phase. Our proposal improves the context establishment for cloud computing services by using a pattern-based approach. The elements of our pattern define the types of information that have to be considered during such a context establishment. Additionally, our pattern specifies the existing relationships between different elements. An instantiation of our pattern provided input (scope and boundary definition) for security and privacy engineering activities at the earliest stages of development. Our focus was on the integration of context analysis with data protection risk assessment in the development life-cycle of cloud computing service. Here, the instantiation of our pattern was the starting point for the risk assessment process. It defined the scope and boundaries in which the risk assessment is performed. Furthermore, an instance of our pattern supported the identification of high-level assets for cloud computing services. These high-level assets can be refined during the risk assessment to more concrete assets on which the risk assessment is performed.

Our proposed pattern conforms to the GDPR and the ISO 27005 standard. The benefits of our approach are as follows:

- Coverage of necessary types of information for the context analysis;
- Provisioning of a graphical representation of the pattern;
- Tool support for the pattern with two graphical editors;
- Consideration of privacy roles that conform to the General Data Protection Regulation;
- Possible usage for the risk assessment process with respect to ISO 27005;
- Provisioning of state-of-the-art privacy concepts, e.g., sticky policies, secure enclaves;
- The possibility of easily expanding of our pattern.

In future activities, we intend to perform adjustments to our pattern that result from progress in the RestAssured project. Furthermore, we will extend our pattern with further concrete solutions for data protection components, like attribute-based encryption, fully homomorphic encryption, etc. For the next steps, new patterns that refer to other steps of the risk assessment, like the identification of threats and controls, will be developed. We plan to implement a new tool that enables improved exchange of data with the other risk assessment tools. We will further apply our pattern to real-life cloud computing services on service levels other than SaaS. For example, we will consider the use case described by Shojafar et al. [54] for the Pay-as-you-Drive scenario.

Author Contributions: L.G. and N.G. are the main co-authors of the paper. They were both involved in the conceptualization and design of the methodology introduced in this paper. Likewise, they were involved in the preparation of the original draft. Particularly, N.G. took part on the investigation, writing-review, and edition of this paper. Substantial feedback for the improvement of this paper was provided by M.H., who also contributed in the edition process.

Funding: Research leading to these results received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 731678 (RestAssured).

Acknowledgments: We gratefully acknowledge constructive discussions with partners in the RestAssured project. We are also grateful to the reviewers and the editors of the paper for their comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mell, P.; Grance, T. *The NIST Definition of Cloud Computing*; Special Publication 800-145; National Institute of Standards and Technology (NIST): Gaithersburg, ML, USA, 2011.
2. Phaphoom, N.; Wang, X.; Samuel, S.; Helmer, S.; Abrahamsson, P. A survey study on major technical barriers affecting the decision to adopt cloud services. *J. Syst. Softw.* **2015**, *103*, 167–181. [CrossRef]
3. Computer Security Institute. *Computer Crime and Security Survey*; Technical Report; Computer Security Institute: Orlando, FL, USA, 2011.
4. Wei, L.; Zhu, H.; Cao, Z.; Dong, X.; Jia, W.; Chen, Y.; Vasilakos, A.V. Security and privacy for storage and computation in cloud computing. *Inf. Sci.* **2014**, *258*, 371–386. [CrossRef]
5. European Union. Regulation 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Off. J. Eur. Union* **2016**, *L119*, 1–88.
6. Federal Office for Information Security. *Security Recommendations for Cloud Computing Providers*; Federal Office for Information Security: Bonn, Germany, 2011.
7. International Organization for Standardization (ISO); International Electrotechnical Commission (IEC). *Information Technology—Security Techniques—Information Security Risk Management (ISO/IEC 27005:2011)*; IEC & ISO: Geneva, Switzerland, 2011.
8. Beckers, K.; Côté, I.; Goeke, L.; Güler, S.; Heisel, M. *A Structured Method for Security Requirements Elicitation Concerning the Cloud Computing Domain*; IGI Global: Hershey, PA, USA, 2014; Volume 5, pp. 20–43. [CrossRef]
9. Zwingelberg, H.; Hansen, M. *Privacy Protection Goals and Their Implications for eID Systems. IFIP PrimeLife International Summer School on Privacy and Identity Management for Life*; Springer: Berlin, Germany, 2011; pp. 245–260.
10. International Organization for Standardization (ISO); International Electrotechnical Commission (IEC). Available online: https://www.iso.org/standard/54534.html (accessed on 14 June 2018).
11. International Organization for Standardization (ISO); International Electrotechnical Commission (IEC). *Information Technology—Security Techniques—Information Security Management Systems—Requirements (ISO/IEC 27001:2013)*; IEC & ISO: Geneva, Switzerland, 2013.
12. Pohl, K. *Requirements Engineering: Fundamentals, Principles, and Techniques*; Springer: Berlin, Germany, 2010.
13. Fernandez, E.B.; Yoshioka, N.; Washizaki, H.; Syed, M.H. Modeling and Security in Cloud Ecosystems. *Future Internet* **2016**, *8*, 13, doi:10.3390/fi8020013. [CrossRef]
14. Withall, S. *Software Requirement Patterns*; Microsoft: Washington, DC, USA, 2007.
15. Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*; Addison-Wesley: Boston, MA, USA, 1994.
16. Fowler, M. *Patterns of Enterprise Application Architecture*; Addison-Wesley: Boston, MA, USA, 2002.
17. Schumacher, M. *Security Engineering with Patterns: Origins, Theoretical Models, and New Applications*; Springer-Verlag, Inc.: Secaucus, NJ, USA, 2003.

18. Schumacher, M.; Fernandez-Buglioni, E.; Hybertson, D.; Buschmann, F.; Sommerlad, P. *Patterns of Enterprise Application Architecture*; Wiley: Hoboken, NJ, USA, 2006.
19. Jackson, M. *Problem Frames: Analyzing and Structuring Software Development Problems*; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 2001.
20. Tsumaki, T. Requirements Engineering Pattern Structure. In Proceedings of the 11th Asia-Pacific Software Engineering Conference, Busan, Korea, 30 November–3 December 2004.
21. Issa, A.; Al-Ali, A. Use Case Patterns Driven Requirements Engineering. In Proceedings of the 2010 Second International Conference on Computer Research and Development, Kuala Lumpur, Malaysia, 7–10 May 2010.
22. Binder, R. *Testing Object-Oriented Systems: Models, Patterns, and Tools*; Addison-Wesley: Boston, MA, USA, 1999.
23. Context-Patterns, Overview. Available online: http://context-patterns.info/index.html (accessed on 20 July 2018).
24. Context-Patterns, Definition. Available online: http://context-patterns.info/definitions.html (accessed on 20 July 2018).
25. Buschmann, F.; Henney, K.; Schmidt, D. *Pattern-Oriented Software Architecture—Volume 5: On Patterns and Pattern Languages*; Wiley Publishing: Hoboken, NJ, USA, 2007.
26. Intel. Available online: https://software.intel.com/en-us/sgx/details (accessed on 13 July 2018).
27. Anati, I.; Gueron, S.; Johnson, S.P.; Scarlata, V.R. Innovative Technology for CPU Based Attestation and Sealing. In Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy, Tel-Aviv, Israel, 23–24 June 2013; ACM: New York, NY, USA, 2013; Volume 13.
28. Zhang, Y.; Chen, X.; Li, J.; Wong, D.S.; Li, H.; You, I. Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing. *Inf. Sci.* **2017**, *379*, 42–61. [CrossRef]
29. Kogos, K.G.; Filippova, K.S.; Epishkina, A.V. Fully homomorphic encryption schemes: The state of the art. In Proceedings of the IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), St. Petersburg, Russia, 1–3 February 2017; pp. 463–466. [CrossRef]
30. Chung, L. Dealing with Security Requirements During the Development of Information Systems. In Proceedings of the Advanced Information Systems Engineering, CAiSE'93, Paris, France, 8–11 June 1993; pp. 234–251. [CrossRef]
31. Alexander, I. Misuse cases help to elicit non-functional requirements. *Comput. Control Eng.* **2003**, *14*, 40–45. [CrossRef]
32. McDermott, J.; Fox, C. Using Abuse Case Models for Security Requirements Analysis. In Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC'99), Phoenix, AZ, USA, 6–10 December 1999; IEEE Computer Society: Washington, DC, USA, 1999; p. 55.
33. Lin, L.; Nuseibeh, B.; Ince, D.C.; Jackson, M.; Moffett, J.D. *Analysing Security Threats and Vulnerabilities Using Abuse Frames*; The Open University: Milton Keynes, UK, 2003.
34. Lin, L.; Nuseibeh, B.; Ince, D.; Jackson, M. Using abuse frames to bound the scope of security problems. In Proceedings of the 12th IEEE International Requirements Engineering Conference (RE), Kyoto, Japan, 10 September 2004.
35. Fredriksen, R.; Kristiansen, M.; Gran, B.A.; Stølen, K.; Opperud, T.A.; Dimitrakos, T. *The CORAS Framework for a Model-Based Risk Management Process. Computer Safety, Reliability and Security*; Anderson, S., Felici, M., Bologna, S., Eds.; Springer: Heidelber/Berlin, Germany, 2002; pp. 94–105.
36. Naudet, Y.; Mayer, N.; Feltus, C. Towards a Systemic Approach for Information Security Risk Management. In Proceedings of the 11th International Conference on Availability, Reliability and Security, ARES 2016, Salzburg, Austria, 31 August–2 September 2016; pp. 177–186. [CrossRef]
37. Haley, C.B.; Laney, R.C.; Moffett, J.D.; Nuseibeh, B. Security Requirements Engineering: A Framework for Representation and Analysis. *IEEE Trans. Softw. Eng.* **2008**, *34*, 133–153. [CrossRef]
38. Fenz, S.; Goluch, G.; Ekelhart, A.; Riedl, B.; Weippl, E. Information Security Fortification by Ontological Mapping of the ISO/IEC 27001 Standard. In Proceedings of the 13th Pacific Rim International Symposium on Dependable Computing (PRDC), Melbourne, Australia, 17–19 December 2007; pp. 381–388. [CrossRef]
39. Kis, M. Information Security Antipatterns in Software Requirements Engineering. In Proceedings of the 9th Conference on Pattern Language of Programs, Monticello, IL, USA, 8–12 September 2002.

40. Sindre, G.; Firesmith, D.G.; Opdahl, A.L. A Reuse-Based Approach to Determining Security Requirements. In Proceedings of the 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03), Klagenfurt, Austria, 16–17 June 2003; pp. 16–17.
41. Firesmith, D. Specifying Reusable Security Requirements. *J. Object Technol.* **2004**, *3*, 61–75. [CrossRef]
42. Fernandez, E.B.; Monge, R.; Hashizume, K. Building a security reference architecture for cloud systems. *Requir. Eng.* **2016**, *21*, 225–249. [CrossRef]
43. Konrad, S.; Cheng, B.H.; Campbell, L.A.; Wassermann, R. Using Security Patterns to Model and Analyze Security Requirements. In Proceedings of the RE'03 International Workshop on Requirements for High Assurance Systems, Portland, OR, USA, 3–10 May 2003.
44. Li, T.; Horkoff, J.; Mylopoulos, J. *Integrating Security Patterns with Security Requirements Analysis Using Contextual Goal Models. The Practice of Enterprise Modeling*; Frank, U., Loucopoulos, P., Pastor, Ó., Petrounias, I., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 208–223.
45. Beckers, K.; Schmidt, H.; Küster, J.; Faßbender, S. Pattern-Based Support for Context Establishment and Asset Identification of the ISO 27000 in the Field of Cloud Computing. In Proceedings of the Sixth International Conference on Availability, Reliability and Security, ARES 2011, Vienna, Austria, 22–26 August 2011; pp. 327–333. [CrossRef]
46. Beckers, K.; Côté, I.; Goeke, L. A catalog of security requirements patterns for the domain of cloud computing systems. In Proceedings of the Symposium on Applied Computing, SAC, Gyeongju, Korea, 24–28 March 2014; pp. 337–342. [CrossRef]
47. Lyubimov, A.V.; Cheremushkin, D.V.; Andreeva, N.; Shustikov, S. Information Security Integral Engineering Technique and its Application in ISMS Design. In Proceedings of the Sixth International Conference on Availability, Reliability and Security, ARES 2011, Vienna, Austria, 22–26 August 2011; pp. 585–590. [CrossRef]
48. Montesino, R.; Fenz, S. Information Security Automation: How Far Can We Go? In Proceedings of the Sixth International Conference on Availability, Reliability and Security, ARES, Vienna, Austria, 22–26 August 2011; pp. 280–285. [CrossRef]
49. National Institute of Standards and Technology(NIST). *Security and Privacy Controls for Federal Information Systems and Organizations*; NIST: Gaithersburg, MD, USA, 2013.
50. Schmidt, H. Threat- and Risk-Analysis During Early Security Requirements Engineering. In Proceedings of the Fifth International Conference on Availability, Reliability and Security, Krakow, Poland, 15–18 February 2010; pp. 188–195. [CrossRef]
51. Ismail, U.; Islam, S.; Ouedraogo, M.; Weippl, E. A Framework for Security Transparency in Cloud Computing. *Future Internet* **2016**, *8*, 5, doi:10.3390/fi8010005. [CrossRef]
52. Surridge, M.; Nasser, B.I.; Chen, X.; Chakravarthy, A.; Melas, P. Run-Time Risk Management in Adaptive ICT Systems. In Proceedings of the 2013 International Conference on Availability, Reliability and Security, Regensburg, Germany, 2–6 September 2013; pp. 102–110. [CrossRef]
53. Surridge, M.; Wilkinson, T.; Stefanieand Goeke, L.W.; Gol Mohammadi, N. Deliverable D7.1—RestAssured Security and Privacy Engineering Methodology. Available online: https://restassuredh2020.eu/wp-content/uploads/2018/07/D7.1.pdf (accessed on 24 July 2018).
54. Shojafar, M.; Cordeschi, N.; Baccarelli, E. Energy-efficient Adaptive Resource Management for Real-time Vehicular Cloud Services. *IEEE Trans. Cloud Comput.* **2016**, doi:10.1109/TCC.2016.2551747. [CrossRef]