Modelling Secure Cloud Computing Systems from a Security Requirements Perspective

Shaun Shei[1], Christos Kalloniatis[1,2], Haralambos Mouratidis[1], and Aidan Delaney[1]

[1] University of Brighton, School of Computing, Engineering and Mathematics, Secure and Dependable Software Systems (SenSe) Research Cluster, Brighton, UK
{S.Shei,H.Mouratidis,A.J.Delaney}@brighton.ac.uk
[2] Cultural Informatics Laboratory, Department of Cultural Technology and Communication, University of the Aegean, Lesvos, Greece
chkallon@aegean.gr

Abstract. This paper presents a cloud modelling language for defining essential cloud properties, enabling the modelling and reasoning about security issues in cloud environments from a requirements engineering perspective. The relationship between cloud computing and security aspects are described through a meta-model, aligning concepts from cloud computing and security requirements engineering. The central concept of the proposed approach is built around cloud services, where the propagation of relationships from a social perspective, abstract software processes and the foundational infrastructure layer are captured. The proposed concepts are applied on a running example throughout the paper to demonstrate how developers are able to capture and model cloud concepts across multiple conceptual layers, facilitating the understanding of cloud security requirements and the design of security-embedded cloud systems to realise organisational needs.

Keywords: Cloud Computing, Cloud Security, Cloud Security Requirements, Modelling Language, Security Requirements Engineering

1 Introduction

Cloud computing enables the provisioning of a wide range of cloud services, delivered on a self-servicing basis for cloud users based on the concept of abstracting physical and virtual computing resources. This paradigm offers seemingly unlimited scalability, availability and flexibility through a pay-per-use model, where users are able to select and deploy cloud services that satisfy their requirements without worrying about how the cloud service is implemented or delivered. However in order to take advantage of these benefits, the distributed nature of the involved technologies implicitly requires the outsourcing of business processes and data to off-premise, third party providers. Thus the users are required to sacrifice a degree of access and control over their data, relying on third party

providers to ensure that their data is kept secure and available. As the concept of cloud computing evolves from utility services to the foundational focus of business IT infrastructure, there is a clear need for ensuring the security of cloud computing systems and maintaining service-provider transparency.

Cloud computing is an evolving term in which the core characteristics has seen numerous reiterations, definitions and is over-saturated in terms of standard definitions [8]. Producing a concrete meaning, reasoning or realisation of cloud computing is hugely dependent on the sector and discipline; between industry, academia, levels of abstraction and granularity, all parties attempts to provide their own definitions [5]. Despite the perceived immaturity of the concept, specifically concerning security, privacy and jurisdictional issues, organisations are still integrating cloud computing as part of their business strategy [4, 3]. Some have even acknowledged the numerous issues but have opted to use the technology regardless, citing the need to keep up with competitors and stay relevant in today's industry [2, 6].

The primary challenge in cloud computing adoption is the lack of a systematic methodology to facilitate the understanding, reasoning and modelling of non-functional aspects, specifically regarding security issues [18, 10]. Combined with a deploy-first, fix-later approach, this creates scenarios resulting in high losses when deploying business systems to the cloud in terms of financial assets, man-hours and reputation [19]. For example moving from a traditional IT environment towards a cloud environment without adequate planning and understanding of the security issues creates systems that are insecure by design, that is the system will inherit both traditional security issues in addition to cloud specific issues. The result is insecure operational systems riddled with vulnerabilities which requires constant patching and even redesigns, where the underlying cause is the lack of a methodological approach for understanding and addressing security issues during the system life-cycle. Thus there is a lack of a holistic modelling language that captures user security requirements and cloud computing properties within a well-defined contextual environment, which satisfies the demand for understanding and realising the requirements for secure cloud-based systems [19, 4, 7].

In this paper we present a modelling language for defining cloud computing properties, based on capturing and modelling the security requirements of organisational systems and providing case-by-case guidance towards deployment properties in cloud environments. This work is part of an on-going research effort to create a framework for holistically modelling secure cloud computing systems, grounded in security requirements engineering and cloud computing security concepts. The framework consists of the modelling language, a process to systematically apply the concepts to the system-under-design and a tool to facilitate automated security requirement analysis. Our work benefits users involved in the process of securing cloud computing systems, for example organisational stakeholders that wish to migrate aspects of their business system to the cloud, providing guidance for security engineers modelling cloud environments or allowing cloud users to understand the security properties of cloud systems.

Therefore in the running example we introduce the users of our work as organisational stakeholders and cloud security engineers under their employment. Our contributions in this paper are:

- Definition of a cloud service to provide abstract and fine-grained description of cloud systems.
- Concepts required to holistically model a cloud computing environment through a three layer approach which describes properties at the organisation, application and infrastructure level.
- Holistic threat and vulnerability analysis through decomposition and propagation of operationalised security constraints through separate or compositions of conceptual cloud layers.

The rest of the paper is structured as follows. A motivating scenario for migrating hospital processes to the cloud is presented in section 2. The cloud modelling language and cloud computing security concepts are defined in section 3. In section 4 we discuss the respective related work. Finally we conclude the paper in section 5, noting the on-going work and contributions.

2 Health-care Running Example

The health-care industry is one example where business and organisational goals are enacted through a complex environment, involving multiple facets of technology and stakeholders such as the exchange of data through disparate systems, collaboration between geographically dispersed medical personnel and a rapidly growing repository of electronic records and medical images [21]. Thus there is a need to ensure the availability of medical assets, interoperability between collaborating health-care partners and reducing IT upkeep costs. However, due to the sensitive nature of health-care data, there are rigorous federal regulations such as the Health Insurance Portability and Accountability Act (HIPAA) surrounding security and privacy in data storage, processing and transit practises. In traditional health-care systems the administrators, doctors and nurses are responsible for ensuring strict HIPAA compliance. However, one of the primary security concerns when moving health-care processes to the cloud is the disseminated responsibility for compliance with key regulations such as HIPAA, due to the unavoidable loss of control over valuable assets and the reliance on third-parties to ensure secure practises are satisfied. Our running example shown in Fig. 1 describes a scenario where one hospital wishes to partially offload their patient records management system to the cloud, in order to improve the availability and interoperability of the records. This information is captured using organisational goal models with existing security requirements engineering approaches, in this example Secure Tropos [14].

3 Cloud Modelling Language

In this section we discuss the cloud computing paradigm and how we capture the essential characteristics from a security requirements engineering perspec-

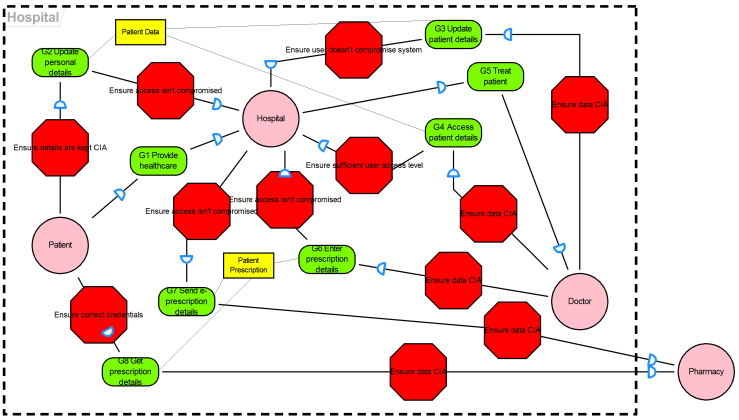

Fig. 1. Simple organisational goal model of hospital processes.

tive. The National Institute of Standards and Technology (NIST) provides the following definition for cloud computing: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.", where the cloud model is composed of "five essential characteristics, three service models, and four deployment models." [1]. We are interested in modelling the service and deployment models, as each service delivery model has a unique set of associated threats and vulnerabilities at the cloud level, while also posing a threat to existing traditional technologies in a cloud environment [11]. The cloud-specific security issues are based on existing work, where our running example demonstrates several cloud threats and vulnerabilities identified by Hashizume et al. [23]. We define our modelling language through established concepts from software security, cloud computing and requirements engineering, combining knowledge from these domains to describe security properties of cloud computing software systems. We follow the Goal-Oriented Requirements Engineering (GORE) approach from the requirements engineering domain [22], where we argue that a cloud service embodies the realisation of a goal.

The proposed cloud meta-model is shown in Fig. 2, illustrating the relationships and attributes of concepts required to describe security in cloud computing through a semi-formal UML notation. The meta-model guides the process of modelling cloud computing systems, through the semi-automated instantiation of concepts and attributes with optional user input to broadly capture scenarios with security in mind. The conceptual model is divided in three groups of concepts. The first group (mainly located on the left side of the meta-model) represents concepts relating to security requirements engineering. The second group (central part of the meta-model) represents concepts for the requirements engineering analysis whereas the third group (right part of the meta-model)

represents the cloud computing concepts. In the following subsections all respective concepts are described based on the aforementioned three groups. For every concept a reference to the running example is provided for better realising the proposed model.

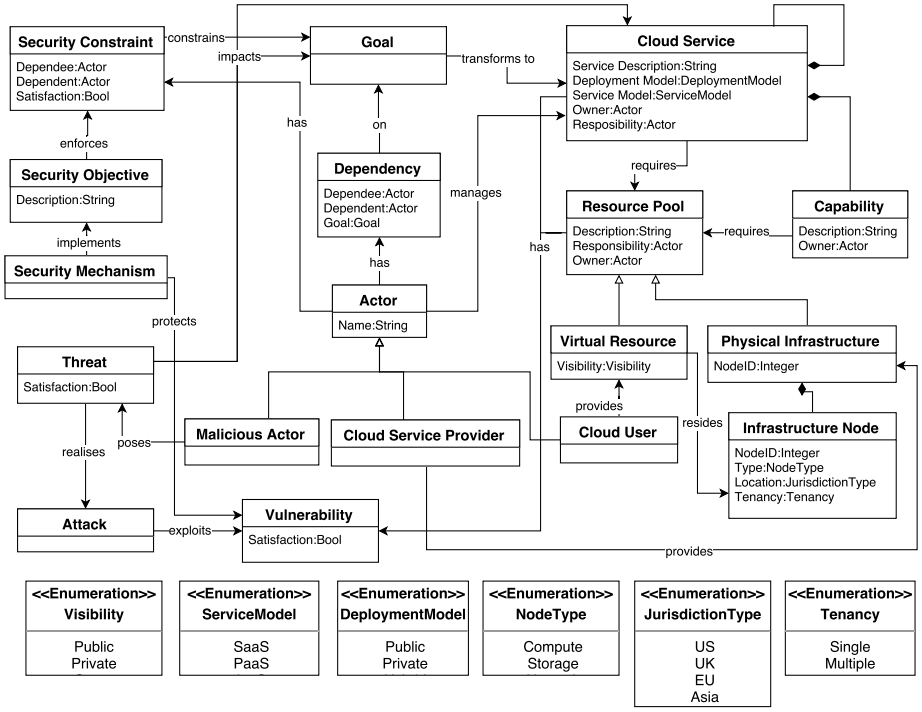

Fig. 2. The Cloud Meta-Model showing the relationships between concepts.

3.1 Security Requirements Engineering Concepts

The proposed concepts are grounded in principles from the requirements engineering and security requirements engineering domain [12–14, 22], in order to facilitate the construction of security-ensured software systems from abstract operational needs. The security-oriented aspects of the modelling language are defined below:

Actor: An actor represents an entity that has intentionality and strategic goals within a software system or around the organisational environment. *A1: Patient*, *A5: CSP*, *A2: Hospital* and *A3: Pharmacy* are examples of actors.

Malicious Actor: This subset of actor represents a stakeholder with malicious intentions, realised through attacks on the system to exploit vulnerabilities and compromise assets. An example of a malicious actor named *A4: Malicious*

Actor is shown in the running example, where they pose the threat *Customer-data manipulation.*

Goal: Goals represent a condition in the world that an actor or system under consideration should achieve. The notion of a goal represents the strategic interests of stakeholders. We convert the concept of goals from requirements engineering to correspond to cloud services, where each goal in an organisational model is mapped to a cloud service. We explain the mapping in detail further on in the cloud computing concepts section below in 3.2.

Threat: A threat embodies the concept of causing harm to an entity, in software security this typically indicates gaining access to, modifying or damaging assets. In the cloud computing context, threats may impact multiple abstract layers. Referring to the running example, the threat *Customer-data manipulation* is posed by a malicious actor, where the threat impacts both *cloud service 1: Patient Details Service* and *cloud service 2: E-prescription Service.* The threat is also realised through the *Cross-site scripting* and *SQL injection* attacks. Threats are unaddressed if at least one vulnerability associated to the threat is not protected by a security mechanism. This is graphically indicated by a exclamation mark inside a red circle, while the satisfaction attribute is flagged false for the instantiated instance of the threat.

Security Constraint: This is a restriction related to security issues, such as the established principles of confidentiality, integrity and availability (CIA). A security constraint is placed from an actor to another actor based around one constrained entity, in this case a goal. This relationship represents the security needs of actors when achieving their goals, which in a cloud computing context indicates security needs that cloud services are required to satisfy. In our running example we have both satisfied and unsatisfied security constraints, respectively indicated visually by a green circle enclosing the letter "s" and a red circle enclosing an exclamation mark. The security constraint *Correct credentials* is satisfied because it is enforced through a security objective, *Ensure only user groups with correct credentials are given access.*

Security Objective: The security objective describes the conditions, criteria and approaches to satisfy security constraints. A high-level description of the security properties provides flexibility when choosing security solutions, as multiple security mechanisms can be implemented to realise the security objective.

Security Mechanism: A security mechanism represents standard security methods for satisfying security objectives, which is described as a high-level solution. In our running example, the security mechanisms *Identity and access management* and *Dynamic credentials* implements the security objective *Ensure only user groups with correct credentials are given access.* It is the security experts responsibility for selecting and realising security mechanisms during the implementation stage, where they decide the most suitable security mechanism through our cloud models.

Vulnerability: This describes a weakness which allows an attacker to reduce a system's information assurance. An example of a vulnerability is *Insecure interface and APIs*, which impacts both *cloud service 1: Patient Details Service*

and *cloud service 2: E-prescription Service*. The vulnerability can be protected by the security mechanism *Web application scanners*, as seen in the running example where the protected vulnerability is visually indicated as satisfied by the letter "s" inside a green circle.

Attack: An attack embodies a specific method of carrying out a threat in order to exploit vulnerabilities in the system. *Cross-site scripting* and *SQL injection* are examples of attacks that exploit the vulnerability *Insecure interface and APIs*, where they are both realised from the threat *Customer-data manipulation*.

3.2 Cloud Computing Concepts

In this section the concepts that are essential for capturing cloud computing properties are presented, centred around the concept of cloud services which are the basic type of resource in a cloud environment.

Cloud Service: A cloud service can be described as a set of six concepts: *capability, actor, resource, relationships, service model, deployment model*. An example of an instantiated cloud service is *cloud service 1: Patient Details Service*, which has the cloud service description *Patient Details Service*, the end-user dependency relationship from the actor *A1: Patient*, the service-provider dependency relationship from the actor *A2: Hospital*, the managed relationship from the actor *A5: CSP*, the requires relationship to virtual resource *Patient Data*, the constraint relationship from the security constraints *SC1: Keep information CIA* and *SC2: Patient access not compromised*, the *SaaS* service model and the *public* deployment model.

Capability: A capability describes, at a high level, an atomic action that is performed by a cloud service to produce a desired outcome. In our running example we have not explicitly modelled capabilities, because the cloud services *cloud service 1: Patient Details Service* and *cloud service 2: E-prescription Service* both provide atomic capabilities. That is if a cloud service only provides an atomic capability, the capability itself is the cloud service and is represented as such. If a cloud service provides two or more capabilities, it is conceptually a composite cloud service where each capability is explicitly represented as an entity belonging to the specific cloud service.

Actor: A cloud service involves direct and indirect stakeholders, disparately distributed throughout the cloud management levels. We define two specialised roles of actors in the cloud, the cloud user and the cloud service provider.

Resource: Assets are represented through resources, which is essential for understanding cloud computing systems and reasoning about security properties. We define two subtypes of resources: *Virtual Resource* to represent information and intangible data and *Physical Infrastructure* to represent tangible assets. Physical Infrastructure is a conceptual container to hold *Infrastructure Nodes*, which abstractly represents physical computing components such as processing servers, data storage and networking connections.

Cloud Service Model: The cloud service provider provides a high-level description of how a cloud service is delivered, which indicates the level of control and the parties responsible for managing computing components. We include

in the model the respective cloud service models; Software-as-a-Service(SaaS), Platform-as-a-Service(PaaS) and Infrastructure-as-a-Service(IaaS) [1]. The cloud service user has the highest level of control in the IaaS model, a lesser degree of control for the PaaS model and little to no control for the SaaS model. For example in the IaaS model the cloud provider is responsible for providing and managing the low-level components such as networking, storage, servers and configuring virtualisation to ensure that users of the service are able to manage the high-level components such as the operating system, application and data.

Cloud Deployment Model: The type of deployment models are also necessary for security reasoning. Thus we include the cloud deployment models as public, private, community and hybrid [1]. The deployment model determines the user group, level of access and accessibility of the cloud service. It also explicitly determines the physical location, ownership and management of computing resources such as infrastructure and data.

Relationships: We propose five types of relationships that are required to capture interactions with cloud services:

- **Dependency:** One actor is dependent on another actor to deliver a cloud service. The depender actor is either a cloud user or an end-user. The dependee actor is a cloud service provider, who themselves can also be a cloud user. A cloud user is an actor that uses an cloud service but has dependents, for example *A2: Hospital* is a cloud service provider and cloud user because they provide the *cloud service 1: Patient Details Service* to the end-user *A1: Patient*, but *A2: Hospital* uses the cloud service provider *A5: CSP* and is dependent on them to provide components of the cloud service.
- **Requires:** One or more resources are required by a cloud service. For example the cloud service *Patient Details Service* requires *Virtual Resource: Patient Data*, indicating that the cloud service requires digital patient records to perform computing processes such as creating, editing and deleting data.
- **Security Constraint:** One or more security constraints are placed on a cloud service. As explained in the security requirements concepts section previously, this represents the security needs of stakeholders which has to be satisfied by the cloud service.
- **Manages:** One or more actors are responsible for managing a cloud service. We use the term manage to represent parties responsible for providing cloud resources, configuring cloud components and ensuring security and jurisdictional requirements are fulfilled. For example *A5: CSP* is responsible for managing the *IaaS*, *PaaS* and *SaaS* layers of the cloud service *Patient Details Service*, while A2: Hospital manages the *SaaS* layer of the same cloud service. This represents that both *A5: CSP* and *A2: Hospital* share the responsibility of ensuring security needs are met and enforced at the *SaaS* level, while only the *A5: CSP* is responsible for the *IaaS* and *PaaS* layers.
- **Impacts:** Threats or vulnerabilities which impact the security properties of a cloud service. These may target a cloud service, or individual capabilities within a cloud service. For example the threat *Customer-data manipulation* and the vulnerability *Insecure interface and APIs* target the cloud services

Patient Details Service and *E-prescription Service*, which indicates that any entities encapsulated in the cloud services are indirectly impacted.

3.3 Cloud Environment Model

Reasoning about security in the cloud computing environment requires a more detailed procedure due to the high complexity and its multi-parameter nature. For assisting prospective users in modelling secure cloud services we have created visual models of the system-under-design during our process based on the cloud meta-model. The graphical notation of cloud security concepts and relationships help facilitate understanding of complex cloud environments, where cloud security engineers are able to holistically model and evaluate security properties of cloud systems based on three conceptual layers at the organisational, application and infrastructure level. Here we explain the cloud environment model through the running example, instantiating concepts from our meta-model to create a holistic cloud view as shown in Fig. 3. The novelty of the approach is based on the three-layer view which assists designers in capturing the necessary concepts for security reasoning holistically.

Organisation Concepts: We describe the stakeholders on the organisation layer in the cloud environment model, identifying the direct and indirect stakeholders as actors through their relationship with cloud services. In our running example we have identified five actors; the *A1: Patient* is an end-user of the *cloud service 1: Patient Details Service* and *cloud service 2: E-prescription Service*, *A2: Hospital* manages cloud service 1 and they are a cloud service provider to *A1: Patient* and a cloud user to *A5: CSP*, *A3: Pharmacy* manages the cloud service 2 and is a cloud service provider to *A1: Patient* and a cloud user to *A5: CSP*, *A5: CSP* is a cloud service provider that manages both clouds services at all three service levels and *A4: Malicious Actor* is a malicious actor which poses a security threat *Customer-data manipulation*.

Application Concepts: This layer represents the abstract concepts for software and applications in the system-under-design, centring around cloud services, components interacting with cloud services and the security impacts. In our running example we model two cloud services, the security issues impacting them, the virtual resources they require and partial solutions for mitigation. The service and deployment models of each cloud service determines the actors that owns the cloud service, actors responsible for managing the cloud service, security issues and propagation of dependencies. For example the cloud service *Patient Details Service* uses a SaaS model and is deployed publicly, determining that the CSP actor *A5: CSP* is responsible for managing components on all three service model layers (SaaS, PaaS, IaaS) while the actor *A2: Hospital* manages the SaaS components. *Customer-data manipulation* is a cloud-specific threat impacting all three service model layers [23], therefore the actors responsible for the cloud services impacted by the threat will be held accountable for deploying security mechanisms in order to mitigate identified threats. In this case the *Customer-data manipulation* threat is realised through attacks *Cross-site scripting* and *SQL injection* which exploit the *Insecure interface and APIs*

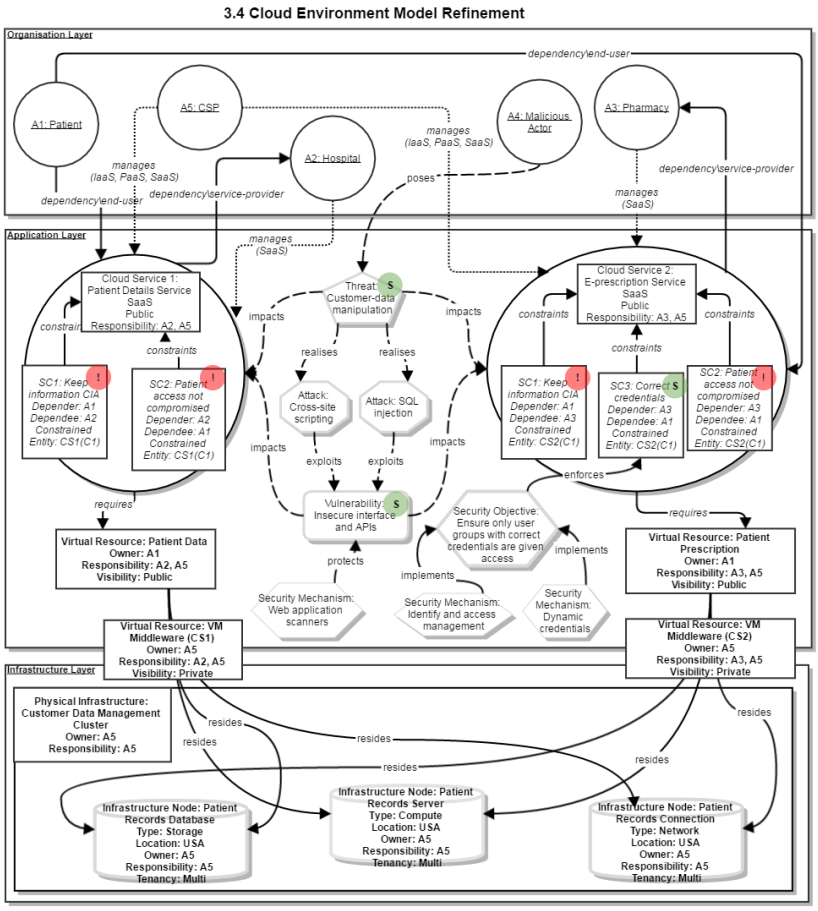

Fig. 3. A holistic cloud view of the health-care running example.

vulnerability, where the cloud security engineer modelling the system has identified a security mechanism *Web application scanners* to protect the vulnerability and thus mitigate the underlying threat.

Infrastructure Concepts: We define this layer to abstractly model physical components required to realise cloud computing services, which we capture as infrastructure nodes belonging to one or more physical infrastructure containers representing IT infrastructure. In our running example, we model a single physical infrastructure to represent one physical IT infrastructure owned and managed by the CSP *A5: CSP*. The compute capabilities are enabled through the abstract notions of a storage, compute and network entity, where they are multi-tenant and geographically located in the USA. From these attributes we can infer jurisdictional legislation such as the USA Patriot Act which applies to all virtual resources residing on infrastructure physically located in the USA,

where multi-tenancy indicates that compute processes are physically shared with one or more unknown cloud service users thus also violating HIPAA compliance. In this scenario the cloud security engineer has a range of options for mitigating these issues, one option is to change the service model of the cloud services to IaaS and provision single-tenancy infrastructure nodes from a CSP geographically located outside the US, thus ensuring dedicated access to cloud computing resources in order to comply with HIPAA regulations.

4 Related Work

Existing research in cloud security is primary focused on mitigating mechanisms and software solutions at the implementation level, which targets software systems that are already implemented and operational [20]. While most work covers multiple security sub-areas, they only target these cloud computing issues in isolation, for example considering security properties in software systems or human factors on a social level but failing to provide direct correlations between the conceptual layers required to fully capture cloud computing issues and indicate impact on security requirements [18, 9]. Li et al. provides a holistic security requirements-eliciting approach towards socio-technical systems [16], however this work lacks expressive power for capturing cloud computing-specific properties which is essential for representing cloud security issues, impact and mitigation. Beckers et al. provides a pattern-based approach for eliciting security requirements and selecting security measures in a cloud computing context [17]. While they provide detailed descriptions of cloud components and properties through their Cloud System Analysis Pattern (CSAP), they do not support propagation of threats or directly model the correlation between security issues and how they are addressed through the instantiation of solutions.

The proposed approach ensures that the system-under-design incorporates security from the early requirements stage, thus addressing underlying vulnerabilities and provides a foundation for implementing security mechanisms and enforcing requirements. We achieve this by building upon existing work in security requirements engineering that lacks the capability to capture or reason about cloud-specific security issues from a holistic point of view [15, 14]. This is achieved through a systematic approach which describes and examines cloud computing properties from three distinct but essential levels of abstraction, aggregating layer-specific details to generate a holistic view of a cloud environment [19]. We identify cloud-specific threats and the impact of attacks within the context of the cloud computing system to elicit security requirements, which is realised through cloud service configurations.

5 Conclusion

Currently there is a lack of a methodology offering systematic support for the process of realising organisational and business needs security through the cloud

computing paradigm. Our work seeks to fill this gap by providing a methodological approach for eliciting secure cloud environment needs from a requirements engineering perspective, enabling developers to realise organisational needs on a cloud computing context with security embedded in the process. We have defined a language to capture cloud computing concepts that enables the modelling of essential cloud properties required to describe cloud services, which we argue represents both abstractly and through a fine-grained perspective, the organisational needs and the relationships required for achieving them. We provide a security-by-design approach using concepts from security requirements engineering, allowing us to model and address cloud security threats and mitigation mechanisms.

We are currently working on a framework to enable the automated transformation of cloud security controls into security patterns, thus providing a pattern library for applying security policies and mechanisms from a security requirements perspective. Initial efforts have been taken to identify patterns from several domains in the Cloud Controls Matrix (CCM) provided by the Cloud Security Alliance (CSA).

References

1. Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.
2. Merrill, T., & Kang, T. (2014). Cloud Computing: Is Your Company Weighing Both Benefits & Risks?. Ace Group.
3. Horwath, C., Chan, W., Leung, E., & Pili, H. (2012). Enterprise Risk Management for Cloud Computing. COSO.[Online].
4. Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). Cloud computing The business perspective. Decision support systems, 51(1), 176–189.
5. Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2008). A break in the clouds: towards a cloud definition. ACM SIGCOMM Computer Communication Review, 39(1), 50–55.
6. Jamshidi, P., Ahmad, A., & Pahl, C. (2013). Cloud migration research: a systematic review. Cloud Computing, IEEE Transactions on, 1(2), 142–157.
7. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50–58.
8. Chen, Y., Paxson, V., & Katz, R. H. (2010). Whats new about cloud computing security. University of California, Berkeley Report No. UCB/EECS-2010-5 January, 20(2010), 2010–5.
9. Iankoulova, I., & Daneva, M. (2012, May). Cloud computing security requirements: A systematic review. In Research Challenges in Information Science (RCIS), 2012 Sixth International Conference on (pp. 1–7). IEEE.
10. Takabi, Hassan, James BD Joshi, and Gail-Joon Ahn. Security and privacy challenges in cloud computing environments. IEEE Security & Privacy 6 (2010): 24–31.
11. Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. Journal of network and computer applications, 34(1), 1–11.
12. Yu, E. (2011). Modelling strategic relationships for process reengineering. Social Modeling for Requirements Engineering, 11, 2011.

13. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology. Autonomous Agents and Multi-Agent Systems, 8(3), 203–236.
14. Mouratidis, H., & Giorgini, P. (2007). Secure tropos: a security-oriented extension of the tropos methodology. International Journal of Software Engineering and Knowledge Engineering, 17(02), 285–309.
15. Fabian, B., Gürses, S., Heisel, M., Santen, T., & Schmidt, H. (2010). A comparison of security requirements engineering methods. Requirements engineering, 15(1), 7–40.
16. Li, T., Horkoff, J., Beckers, K., Paja, E., & Mylopoulos, J. (2015). A holistic approach to security attack modeling and analysis. In Proceedings of the Eighth International i* Workshop (2015, to be published).
17. Beckers, Kristian, et al. A structured method for security requirements elicitation concerning the cloud computing domain. International Journal of Secure Software Engineering (IJSSE) 5.2 (2014): 20–43.
18. Sengupta, S., Kaulgud, V., & Sharma, V. S. (2011, July). Cloud computing security–trends and research directions. In Services (SERVICES), 2011 IEEE World Congress on (pp. 524–531). IEEE.
19. Almorsy, M., Grundy, J., & Mller, I. (2010, November). An analysis of the cloud computing security problem. In Proceedings of APSEC 2010 Cloud Workshop, Sydney, Australia, 30th Nov.
20. Modi, C., Patel, D., Borisaniya, B., Patel, A., & Rajarajan, M. (2013). A survey on security issues and solutions at different layers of Cloud computing. The Journal of Supercomputing, 63(2), 561–592.
21. Ahuja, S. P., Mani, S., & Zambrano, J. (2012). A survey of the state of cloud computing in healthcare. Network and Communication Technologies, 1(2), 12.
22. Van Lamsweerde, A. (2001). Goal-oriented requirements engineering: A guided tour. In Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on (pp. 249–262). IEEE.
23. Hashizume, K., Rosado, D. G., Fernández-Medina, E., & Fernandez, E. B. (2013). An analysis of security issues for cloud computing. Journal of Internet Services and Applications, 4(1), 1–13.