Towards a Unified Secure Cloud Service Development and Deployment Life-cycle

Aleksandar Hudic, Philipp M. Radl,
Thomas Lorunser
Austrian Institute of technology
Digital Safety and Security
Vienna, Austria
Email: firstname.lastname@ait.ac.at

Matthias Flittner, Roland Bless
Karlsruhe Institute of Technology (KIT)
Institute of Telematics
Karlsruhe, Germany
Email: firstname.lastname@kit.edu

***Abstract*—Designing and developing cloud services is a challenging task that includes requirements engineering, secure service deployment, maintenance, assurance that proper actions have been taken to support security and, in addition, considering legal aspects. This is unfortunately not possible by taking current methods and techniques into consideration. Therefore, we require a systematic and comprehensive approach for building such services that starts the integration of security concerns from early stages of design and development, and continuous to refines and integrate them in the deployment phase. In this paper we therefore propose a solution that integrates security requirements engineering and continuous refinement in a comprehensive security development and deployment life-cycle for cloud services and applications. Our approach is focused on iterative refinement of the security-based requirements during both software engineering (development phase) and software maintenance (deployment phase).**

***Keywords*-security, cloud, development life-cycle, requirements engineering, security policies, assessment, assurance, monitoring**

I. Introduction

The still ongoing revolution in the usage and consumption of IT resources driven by the cloud paradigm is dramatically changing the ICT landscape. Essentially, the cloud paradigm utilizes the service delivery model to facilitate outsourcing on all possible layers. While some see it as a novel technological concept, others only consider it as an evolutionary step of ICT technologies [1]. Nevertheless, cloud computing plays an important role by providing significant economical and operational advancements, by utilizing interoperability, scalability, on demand service provisioning on global scale with minimum management effort [2].

Prior to being hosted in a cloud, services need to be designed and tailored to maximally leverage the cloud characteristics (on-demand provisioning, ubiquitous network access, resource pooling, rapid elasticity and measured services) [3], by taking design principles and requirements into consideration. A common approach for engineering software products is the *Software Development Life Cycle* (SDLC) [4] also supported by ISO 12207 Standard for software life-cycle processes [5].

The complexity of software, especially in dynamic and volatile environments such as cloud, is hard to predict in early stage of development since it depends on the evolvement and refinement of the initial requirements. During the early stage of software design and development initial high level objectives and requirements that the software needs to fulfill are defined. The requirements are considered the foundation of the software development process very often require to be refined during both development and deployment life-cycle of a product. Therefore, it is necessary to integrate an iterative requirements engineering process that will implement continuous evolvement of both software and requirements during the whole life-cycle of a product [6].

On top of the aforementioned challenges towards designing and developing software products, security is another important requirement nowadays that is often being treated independently of the development process and considered in later stages. Furthermore, its consideration is often premature and security objectives are often traded for usability aspects. However, in our work we emphasize that security engineering should be an integral part of the whole engineering process and carefully considered in each step of a product's life-cycle. In particular, we propose a secure software development life-cycle for cloud services that covers both development and deployment phases of the product life cycle in a cloud environment. Furthermore, our model aligns design, development and deployment with standards, best practices and guidelines as an iterative process that also integrates security requirement engineering and refinement [7], [8].

This work is structured as follows. Section II offers a detailed literature overview of methodologies for secure service development and operation from early design to late deployment and maintenance. Next, in Section III we show the results of a survey that analyzed the relevance of cloud characteristics and design concerns during the development stage. Furthermore, we also analyze and propose a generic security requirements engineering process that we will integrate later in our secure cloud service life-cycle. In Section IV we present our approach for iterative cloud service design and development that integrates security in each step through both development and deployment phase for developing and deploying secure cloud services. Finally, in Section V we conclude our work and outline our future work.

II. State of the Art

Most recent studies indicate that despite the attractive benefits, lack of security (e.g. lack of transparency, data privacy, trust, data lock-In, data loss) still remains a major obstacle for deploying services into the cloud. The research community is keen on analyzing and addressing security challenges in the cloud environment [9], [10], [11], [12], [13], [14], [15]. This is especially challenging when it comes to critical infrastructure services that require high attention when it comes to security [9], [15]. Nevertheless, there is a part of the research community that proposes solution and methods for service deployment in a cloud. Khajeh et. al [16], [17] proposed, in their work, the migration of enterprise IT services to the cloud with regards to context of financial and socio-technical enterprise issues and the decision making process for service migration by taking into consideration cost modeling and risk assessment. Furthermore, Kaisler and Money, in their work [18], evaluate the compatibility of the service migration approach with the cloud computing paradigm by addressing acquisition, implementation, security, usage reporting, valuation and legislative challenges during the process. In their work, Fehling et. al. [19] elaborate and advise best practices for addressing web based service migration challenges with regards to migration patterns.

Requirements engineering plays an important role in secure service development, because it identifies crucial security considerations that have to be taken into account during the early stage of development till the deployment and maintenance to ensure the complete service life-cycle. Therefore, in their work Haley et.al. [20] present a comprehensive framework for security requirement analysis and elicitation based on context analysis. Security requirements are defined as constraints on system's functional requirements based upon system or service security goals. Furthermore, Mellado et.al. [21], [22] leverage the Common Criteria approach for to utilizing requirements engineering with respect to security concerns in early development stage, formally referred as Security Requirements Engineering Process (SREP). The SREP is an asset-based and risk-driven model that elicitates security requirements through iterative micro-processes (e.g. identifying, prioritization and categorizing requirements, vulnerabilities and threats, assessing risks, and identifying security objectives). The work of Hesse et.al. [23] outlines an approach that combines heuristics, monitoring and decision documentation to perform semiautomatic security requirements engineering, whereby heuristics monitoring is used to mitigate the manual effort.

Maintaining and ensuring that security of our systems and services are at the proper level we require solutions (e.g., security assessment, security monitoring, auditing) that will perform the validation or assessment of security in our ICT systems. One of the most prominent approaches nowadays, Common Criteria [24] provides an efficient and systematic approach for security assessment that offers a certain level of confidence that predefined set of security requirements (i.e. functional or assurance requirements) have been met by the evaluated product formally refereed as Target of Evaluation(ToE). The pioneering work of security assessment in ICT environments was developed by the US Department of Defense under the name Trusted Computer System Evaluation Criteria (TCSEC) [25], commonly referred to as the "The Orange Book". The approach classifies the assessment across three fundamental categories minimal, discretionary, mandatory and validated protection where each category contains a proposed set of security controls that are being validated (e.g. policies, access control models, audit trails, roles, processes, etc.). Furthermore, TCSEC is unfortunately mostly focusing on confidentiality and towards a high-level evaluation of systems and services. Analogous to the TCSEC approach, the Information Technology Security Evaluation Criteria (ITSEC) [26] also builds the evaluation on security controls. However, the users are able to encounter the preferred set of security requirements tailored to their product which the ITSEC formally refers to as Target of Evaluation (ToE). In addition, the ITSEC while still assigning levels it differentiates between functional and assurance levels.

With respect to the above state of the art our work overcomes the gap of a holistic integration of security engineering during both development and deployment phase, i.e., it offers continuous security integration that is based upon iterative security requirement engineering. Our security requirements engineering process provides security requirements in various abstract forms (objectives, requirements and properties) to support secure service design, monitoring and assessment through development and deployment phases. Our solution present an uniform solution that is aligned with the approaches proposed by Wagner et.al. [27] and Hudic et.al. [28].

III. Information Security Requirements Analysis and Engineering

Engineering and elicitation of requirements for supporting software design and development is a cumbersome and time consuming process, especially when it comes to security. Often, due to the very strict deadlines that are dictated by time to market, security is not considered as a primary concern in service design or development. Such products are easily being prone to security flaws because of some minor mistake during design stage [29]. Hence, design and development of secure services becomes additionally challenging when it comes to deploying those services in cloud environments, because there are many challenges entailed with the features offered by the cloud that have to be carefully taken into consideration for both development and production phases [30].

Design, followed by development, of a service life cycle commonly starts with defining very high level objectives that have to be engineered into the development process of a service and maintained until a service is finally deployed. To support such analogy it is often necessary that through an iterative refinement process security requirements are continuously improved before being integrated in a partic-

ular stage. By taking these into consideration, we propose and develop an approach for continuous integration, refinement and maintenance of security related requirements during both development and deployment phase. Prior to building our secure cloud service life cycle we analyze relevance and depict a conceptual model for the security requirements engineering process for cloud applications and services, shown in Figure 3.

A. Requirements Analysis

In order to support our requirement engineering model we conducted a survey among 31 academic and 46 industry professionals where we investigated the relevance of cloud characteristics and design concerns during the development phase. The first objective of our survey was to investigate if the participants take under consideration, in early stage of design and development, the following criteria: risk assessment, SLA management, architectural patterns, service life-cycle, autonomic security management, forensics and auditing, and international standards. The results, depicted in Figure 1, show that generally industry participants have higher interests in the above mentioned objectives than the academic participants with the exception of autonomic security management that was slightly below (80%) for industry participants. Although generally lower than industry, the academic participants focus their attention towards risk assessment (82%), forensics and auditing (75%), international standards (67%) and autonomic security management (75%).

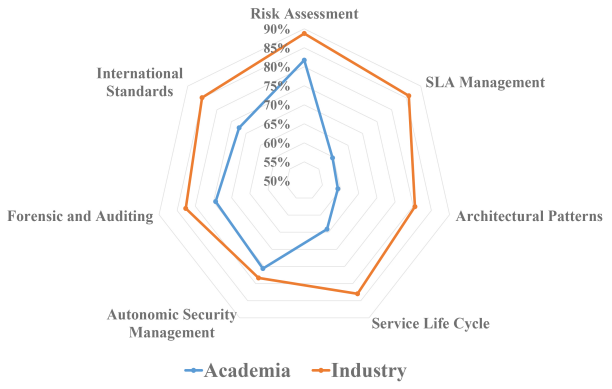

Figure 1. Relevant design concerns

The second objective in our survey is focused to investigate how the NIST cloud computing characteristics [3] are embraced when designing, developing and deploying services. The results, depicted in Figure 2, show high interest for both industry and academic participants for considering cloud characteristics when designing, developing and deploying their services. The results, shown in Figure 2, indicate also in case of NIST cloud characteristics that generally interest by industry participants is higher than for academic participants. However, the geolocation was the only characteristic where the industry had a slightly lower interest than the academic participant, and at the same time characteristic with the lowest interest results for industry participants. This is most probably due to the higher interest in industry for usability over security when it comes to developing products.

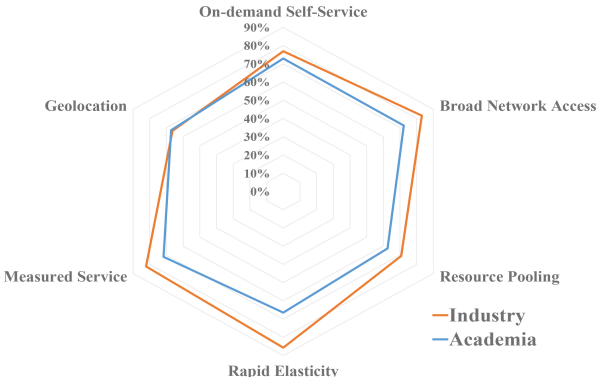

Figure 2. Relevance of NIST cloud characteristics during service design and development phase

B. Requirements Engineering Model

We envisioned the security related requirements engineering and refinement, shown in Figure 3, as multi stage iteration process that starts with the initial context analysis for a particular use case scenario, i.e., application or service being developed or deployed in cloud environment. The output of the context analysis are high level business objectives, functional requirements, and security objectives that are aligned with standards, guidelines and best practices. Next step in our process performs an analysis and refinement of the high level requirements and objectives based upon risk and vulnerability assessment. Hence, the output of service risk and vulnerability assessment is taken in to consideration when performing refinement of security objectives. The result of security objective refinement are concise security requirements that are used for secure service design, development and deployment. After the service is deployed in production, validating security during runtime is supported by the refined version of security requirements that are defined as security validation elements, i.e., security properties. Both security requirements and properties must be aligned with standards, guidelines and best practices. Furthermore, both security properties and security requirements are used for defining security related policies for maintaining, auditing, designing and assessing services during both development and production phases.

IV. Secure Cloud Service Life-cycle

A software service life-cycle comprises several stages ranging from design, development, deployment to maintenance. To improve the overall software quality and ensure its security it is crucial to consider security aspects in all stages. Commonly, service design starts with high level business and security requirements. They are afterwards translated to functional and non-functional requirements and used throughout the development phase to implement software functionalities. Whenever a particular service is being deployed or migrated to an environment such as

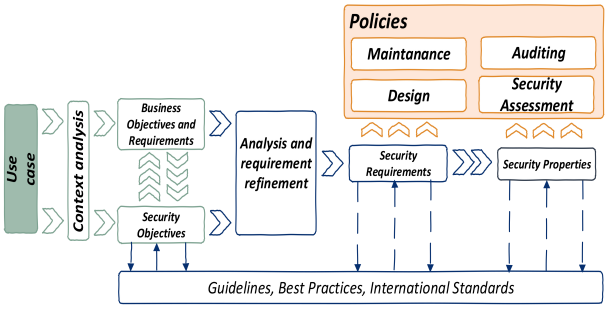

Figure 3. Iterative process for building security policies and security requirements engineering

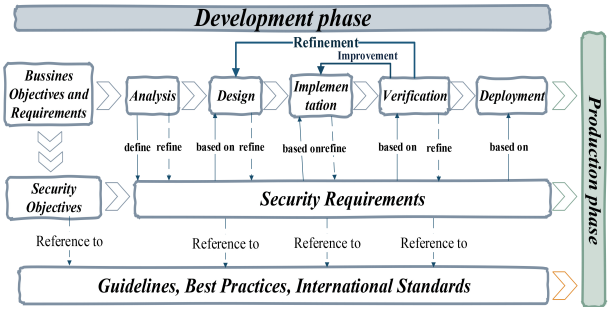

Figure 4. Development phase of secure cloud-service development life-cycle

cloud we need to define additional requirements together with a deployment strategy [27]. Furthermore, we would also like to have a guarantee that the security requirements are fulfilled at design, implementation and maintenance of a cloud service. In order to check compliance to defined security requirements, measurable assessment criteria have to be defined which can be used in validation process. In summary, we see that the protection of software through its entire life-cycle requires elicitation and refinement of security requirements in all stages which can become challenging when targeting heterogeneous and distributed infrastructures. In the case of cloud usage a new abstraction level has to be applied to support mapping between different stages into an unified approach.

We propose a two-phase secure cloud service life-cycle that integrates requirements engineering and iterative refinement with respect to security, through each stage of both phases. The first phase is called *Development phase* and covers the sequential set of steps where a service is being designed and developed. Secondly, the *Production phase* is where a deployed service is validated against those security requirements that have been defined in development phase.

As mentioned before, consistent integration of security concerns throughout each step of both phases is vital for designing and operating secure systems and services. Therefore, in each phase of our proposed life-cycle we conduct an iterative security requirements engineering process to align the requirements to the needs of a particular step (design, development, maintenance, assessment or monitoring) and standards, best practices, or guidelines [31], [32]. Brining together the cloud secure development life-cycle [27] with the cloud assurance assessment framework [28] and cloud inspector [33] offers a unique and smooth way to continuously integrate security in service life-cycle from development phase to production phase. We perform continuous refinement of security in terms of properties that through the life-cycle yield bot security functional and non-functional requirements depending on the phase corresponding step.

A. Cloud Service Development

Security by design approach is a vital part for designing secure software and preparing it for deployment in security demanding environments such as cloud. We introduce an enhanced version of Secure Cloud Service Development Life-cycle model from Wagner et.at. [27], that supports our model for integrating security across all stages of the cloud service life-cycle from design, development, deployment preparation and migration, till production.

In our approach we narrow down the focus of the secure development life-cycle process by taking the following objectives into consideration:

- Integration, engineering and continuous refinement of security requirements in each stage of software (design, development, testing, deployment and maintenance) for cloud based architectures.
- Secure software development for a cloud environments from scratch.
- Software migration from a legacy system to the cloud (adoption for cloud).
- Software migration from private to public clouds and vice versa.
- Iterative security requirements engineering during both development and production life-cycle phases.

As mentioned before, we align our approach with Wagner et.al. [27] that is focused on building a guideline for secure service migration in cloud by integrating the security engineering process into the cloud secure development life-cycle. We extend the approach by adding an additional initial step to addresses high level security requirements according to the high level business objectives supported by the continuous security requirement refinements according to standards, guidelines and best practices. Furthermore, our approach highlights the following six stages for the secure software development phase:

a) High level security objectives analysis: This preliminary step consolidates high level business objectives with security related standards, best practices and guidelines to set the initial security objectives for secure service design, development and deployment.

b) Analysis: This step performs analysis of a service with respect to cloud requirements. The IT services that are intended to be developed and cloudified, i.e. hosted in the cloud, are analyzed to prove their eligibility of deployment in cloud environments. Furthermore, the initial set of security requirements is specified and potential threats to the particular use case are identified. Ideally, if security requirements for the IT service are predefined,

they are taken into account and, if needed, adjusted to the circumstances occurring in each subsequent stage. In particular, this also involves security requirements indirectly resulting from the cloudification or development of a particular service. For example, this might refer to requirements for providing credible digital evidence on the providers security-related conduct for the potential case of legal conflicts, or to cloud-specific requirements from data protection law. Moreover, this step also proposes an analysis of the implications that the development or cloudification of the IT service has on the organization and the business.

c) Design: In the design step, the software architecture for the to be migrated or developed IT service is designed in line with the security requirements specified in the analysis step. If required, refinements of the security requirements are performed for precisely aligning the security requirements towards the particular use case.

d) Implementation: The foundations defined in design step are used to implement part or complete service in line with the NIST cloud characteristic. Here we can also take into consideration the NIST cloud characteristic analysis performed in Section III-A. Also in this step additional security property refinements can be performed if required.

e) Verification: In this step, software is tested against the predefined set of security requirements before being deployed or migrated in to the cloud. In addition, in case of cloudifying IT service the readiness of the organization shall be verified (e.g. special disaster recovery strategies, trainings, or revisions of SLAs might be required). If the verification does not succeed, either further implementation effort needs to be taken and/or the design needs to be revised. Additionally, risk assessment is performed in order project the risks involved based upon potential threats and vulnerabilities at this particular stage.

f) Deployment: In the final step of development phase the IT service is deployed to the cloud environment by taking into account the security requirements related to platform configuration.

The result of the development phase is a service that integrates best practices with respect to security aligned with most prominent standards and guidelines nowadays. Furthermore, an early stage security requirements engineering yields from security objectives a more concise set of security requirements, that iteratively through the above mentioned steps lead to more robust and secure design of our services. These requirements are also used to perform the selection of the most eligible cloud provider that can fulfill the needs of a particular customer to host his service and as the input for the production phase where the security of a particular service should be maintained with respect to security requirements.

B. Cloud Service Deployment

The integration of the development life-cycle, that we introduced in Section IV-A, with the assurance assessment framework and CloudInspector offers a unique way to verify the implementation of the initial security objectives

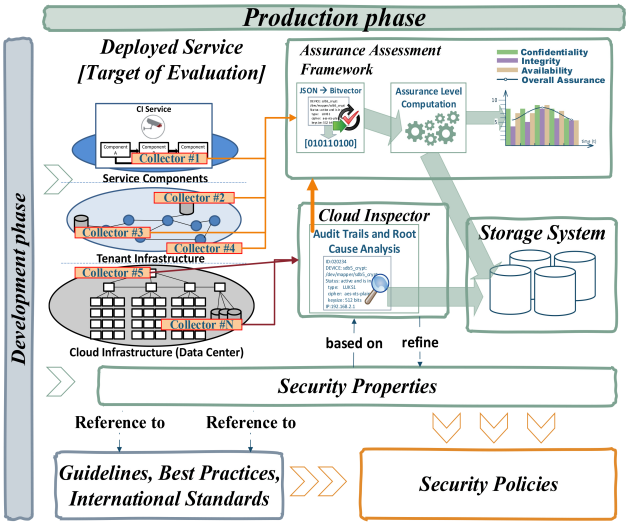

Figure 5. Production phase of secure cloud-service development life-cycle

from the early stage of development to the production stage. At the same time, these properties are being refined iteratively through stages to meet security related best practices, standards and guidelines. The second phase of our secure cloud-service life-cycle model, the production phase, is focused on maintaining and monitoring security concerns based upon the security requirements defined in the development phase.

The development phase, shown Figure 4, enumerates the foundational security requirements and measurable properties of the system that can be used in both assurance assessment framework and CloudInspector, shown in Figure 5. The main difference of those assurance framework and CloudInspector is the service model. CloudInspector is generally envisioned as a module used at infrastructure level to provide on demand audits, whereby assurance framework is covering all cloud levels in a continuous manner. The security requirements have to be aligned with common security guidelines, best practices, and international security related standards. Nevertheless, these security requirements require additional adjustments to be used in the production phase by assurance assessment framework and CloudInspector.

1) Cloud Assurance: The security requirements defined by the development phase are focused to identify functionalities of a service rather than validation and measurement elements. Hence, a requirement engineering process is necessary to transforms development phase security requirements into non-functional production phase security requirement that we refer to as *security properties*. Furthermore, the security properties are then used by the security assurance framework developed by Hudic et. al. [28] to acquire security related information from the infrastructure where the evaluated services are being deployed. Since our focus is to evaluate large complex ICT infrastructures such as cloud the assurance assessment framework is of a particular benefit for our secure cloud

service life-cycle. As mentioned, the assurance assessment framework performs security based evaluation of complex multi-layered infrastructures with respect to a specific predefined set of security properties, i.e., quantitative security analysis of a particular entity hosted in the cloud. The foundation of the quantitative security assessment criteria is aligned with the Common Criteria, a comprehensive and systematic approach developed by Department of Defense for performing security assessment. Furthermore, our assurance security assessment approach uses the following three elements to systematically organize the security assurance assessment of particular system or a service: Target of Evaluation (ToE), Group of Evaluation (GoE) and Component of Evaluation (CoE). These entities are used for creating holistic service abstraction and, at the same time, offer flexibility, granularity and precision for continuously assessing or validating security concerns across a wide area of components, groups of components, and even entire ICT infrastructures. Nevertheless, to perform consistent security assessment framework requires a predefined set of security properties, used to validate security across individual components of interest (CoE). These security properties need to be concisely defined in order to be used by the assurance assessment framework security validation based upon their conditions. Therefore, we use the security requirements from the development phase to engineer the security properties for our validation process. To acquire security related information across the infrastructure where a particular service is deployed we use *Collectors* that according to the precise definitions of individual security property harvest information across the whole infrastructure and deliver it to assurance framework to compute the assurance level. Additionally, both security requirements and properties are used to define policies for maintaining or assessing security, and even guidance for secure service development and design. The collected security related information across the observed infrastructure is then compiled to security levels, which we refer to as assurance levels, and classified across three assurance classes (confidentiality, integrity, availability).

2) Cloud Inspector: As important as monitoring of the assurance level with the security assurance framework is to verify if the cloud provider fulfills contractually agreed security policies (such as geo-location of virtual machines, dedicated host requirements, or physical host anti-affinity) during runtime. The current best practice for that task is certification (e.g., [31], [34], [35]) of cloud provider practices only in large intervals. This does not permit continuous transparency of fulfillment of security policies during runtime. Additionally, in case of data protection, the law states in many countries that a processor of personal data has to be actively controlled. Therefore, we propose to use an independent Transparency-as-a-Service solution such as *CloudInspector* [33] to overcome the lack of transparency in cloud computing.

The *CloudInspector* solution consists of two functionalities: on-demand auditing and continuous evidence gathering. *CloudInspector* enables tenants to continuously control contractual agreements (security policies or properties) during the cloud deployment phase (runtime). Additionally, *CloudInspector* continuously collects meta data about current cloud behavior. In case of a dispute in court the collected meta data at best could be used to determine the root cause of a failure (i.e. negligence of cloud provider). To do so, meta data about cloud behavior must be collected beforehand during deployment phase. The *CloudInspector* solution therefore gathers information within the cloud environment independently of the cloud management platform. Due to the independence the *CloudInspector* solution is able to unveil misconfiguration or malfunction of the cloud management platform. The Transparency-as-a-Service solution consists of two elements: Transparency Controller Module (TCM) and Transparency Enhancement Module (TEM). Whereas the TCM provides an audit request interface for tenants. The TEMs are distributed monitoring agents on each physical host within the cloud environment. The TCM serves as interface for tenants, processes on-demand audit inquiries and coordinates audit requests to as well as responses from the distributed TEMs. The TCM offers per tenant access via web-based or RESTful interfaces. A TCM transforms incoming audit inquiries of tenants into internal audit requests. Audit requests are used for on-demand real-time auditing. These audit requests are sent via audit channels to the TEMs.

After all audit results of the corresponding TEMs are received, the TCM evaluates them and prepares an audit response for the tenant. The TEM monitors physical and virtual resources residing on a physical host and uses data sources that are largely independently of the cloud management platform. Only a single TEM per physical host (compute, network, storage) is necessary. A TEM gathers *audit data* on-demand (i.e., due to an audit request) or continuously according to individual tenant policies. Furthermore, a TEM consists of four basic types of components, namely *Collectors*, *Manager*, *Analysis*, and *Logging*. Collectors are connected to different cloud management platform independent audit sources, e.g., information may be gathered from the hardware, the operating system, event logs, the virtualization library or other physical components. For example, a list of active virtual machines can be obtained by using an operating system command and/or using an API from the hypervisor. Each collector is instructed by the Analysis (i.e., on-demand auditing) or Logging component (i.e., continuous evidence gathering) to gather specific audit data. This can happen regularly by polling certain values or it can be event based so that other components only have to act on such events. The Manager component detects changes of tenant assigned virtual resources (e.g., creation, deletion, migration, start, stop). Depending on those observations the manager joins or leaves an audit channel. The manager component receives audit requests from a TCM via audit channels and sends back audit results directly. If an audit request is received the manager instructs the analysis component to process the audit request (on-demand auditing)

or triggers the logging component to continuously record related events (continuous evidence gathering).

The Analysis component processes incoming audit calls from the manager. It either performs a lookup in recently locally recorded data or performs an on-demand check. It may trigger one or several collector components to gather audit data and preprocesses information according to the specific audit request. Finally, it returns the corresponding audit data to the manager component. The logging component allows tenants to initiate policy-based continuous evidence gathering in order to create audit trails. Based on tenant policies this component may continuously trigger collectors or record certain events to gather relevant audit data. For instance, it could log the operating system, cloud platform and hypervisor versions twice a day or any kind of management access, hardware failures, or reboots at any time when they occur. The usage of *CloudInspector* during the deployment phase guarantees that tenants are able to verify if the cloud provider fulfills contractually agreements during runtime and that in case of a dispute in court significant evidence about cloud behavior will be available.

C. Application scenario

To demonstrate the application of our life-cycle on a real world scenario we highlight major steps of process for building, deploying and maintaining a video surveillance service for critical infrastructure, such as public safety, deployed in a cloud environment. The initial objectives of our video surveillance service is to identify potentially malicious behavior, especially in high security areas. Therefore the video surveillance systems has to be able to process video recordings in real time and identify potentially malicious individuals. Since the prior objective of the video surveillance software is the facial recognition functionality that performs authentication of individuals. The high level requirements of our service are:

- identifying and authorizing individuals,
- providing high availability of video surveillance service without downtime,
- protecting the confidentiality and integrity of video records.

The analysis phase in this case would identify functional security requirements in line with the above objectives: secure service auditing, restricting access to sensitive video recordings, ensuring high availability and protecting confidentiality of video recordings. Further, more detailed, analysis outlines a more concise set of security requirements in line with the ISO 27001/27002 and NIST 800-53 standards that offer a comprehensive list of security requirements with comprehensive description. Additionally, as part of the analysis step we performed a risk assessment based on potential threats that can occur (e.g., malicious insider that could temper or destroy video surveillance records, broken disk containing unencrypted video records being lost, hosting data under legislative domain with invasive privileges to access information, etc). The result of the risk assessment and threat analysis extends security requirements used to develop our secure video surveillance services. In this early stage of design, implementation and verification standards such as ISO 27001/27002 are used to identify best practices for implementing security controls that can ensure confidentiality and integrity of video records. Therefore, during these three process steps we perform additional service refinements and improvements with respect to security requirements.

Furthermore, in the implementation phase of a service we take into consideration the cloud characteristics, depending on the relevance to the use case just as in our analysis in Figure 2, in order to leverage them properly and ensuring high availability of the service when being deployed in cloud. Prior to performing and planning deployment of the cloud service, additional verification and risk assessment analysis is performed to identify unexpected deviations in design or development, and any new potential threats.

Next, we have to consider secure deployment or migration of our service to the cloud by considering the CloudSDLCv1 from Wagner et.al. [27]. Once the service is deployed, we have to further consider measurable metrics to perform the security validation and auditing of the deployed service. In this part we conduct additional security requirement engineering that yields a set of security properties from the security requirements defined in the development phase. They are used for the validation of security goals via the security assurance framework developed by Hudic et. al. [28]. The security assurance assessment framework abstracts the service over individual components to perform independent security validation that is afterwards taken into consideration when performing holistic security analysis, as shown by Figure 6. The right hand side of the Figure 6 shows a general tree model of abstracted video surveillance service where set of security properties is validated across each component. The assurance framework identifies per component, e.g., components 1, 3, 4, 5 and 6, individual security properties that do not fulfill their security control requirement, marked red in right hand side of the Figure 6. The underlying infrastructure information is being delivered by the CloudInspector on demand. This gives us the ability to easily deploy our assurance framework on any given public cloud provider that integrates CloudInspector as independent solution for acquiring infrastructure information. Furthermore, the security assurance assessment framework offers the ability to the end users for defining variety of security validation policies for validating the security concerns of their cloud providers.

In the same way, the CloudInspector solution is used during deployment phase to actively control the cloud provider, to collect evidence for a potential debate in court and to support security assurance assessment framework with infrastructure related information. From data protection perspective recorded video footages of individuals are personal data. If for example a tenant uses this cloud-based video surveillance service to process such personal data of

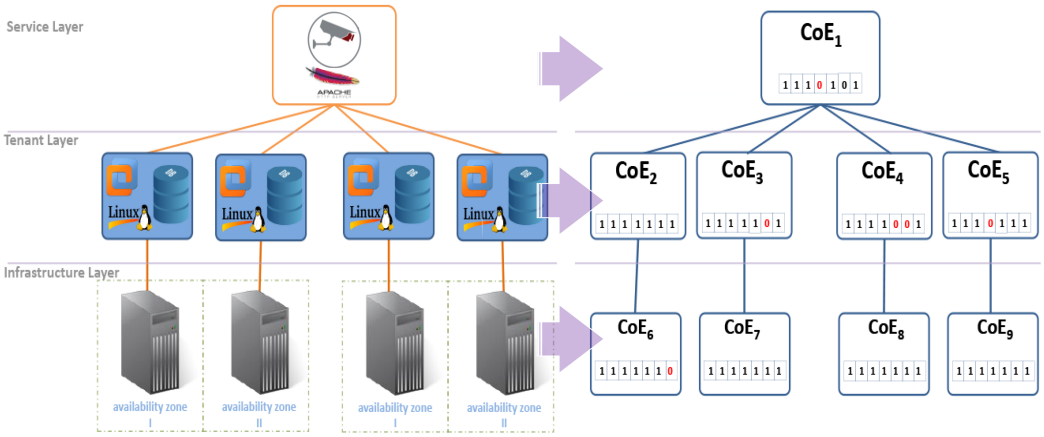

Figure 6. Illustration of holistic abstraction of video surveillance service for security assurance assessment process.

individuals, he has to actively control the cloud provider that he processes the data as ordered (only in datacenters in specific countries or on dedicated physical hosts). Additionally, with CloudInspector actively controls if the cloud provider fulfills contractual agreements. From a civil law perspective continuously collected and stored meta data about cloud behavior is very useful. CloudInspector will provide this meta data in case of a dispute in court, so that this meta data can be used during a root cause analysis (which maybe unveil cloud provider negligence).

As shown by our application scenario bringing together secure development life cycle with security assurance assessment framework and CloudInspector enhances the service security by design and increases the transparency with respect to securely hosting services in cloud. If a particular cloud provider integrates the CloudInspector for investigating infrastructure related security concerns that additionally support legal restrictions, we are able to easily deploy assurance assessment framework without interfering with internal cloud processes and exposing internal infrastructure sensitive information and derive security assessment.

V. Conclusion and Future Work

We present in our work a uniform methodology for developing and deploying secure cloud services as a life-cycle that implements continuous integration and refinement of security requirements. Our methodology is built as a uniform and sequential process that integrates design, implementation, testing, deployment, maintenance, assessment and monitoring of cloud services. These sequential steps are divided in two phases, Development phase that covers design, implementation, testing, deployment, and Production phase that covers maintenance, assessment and monitoring. In both phases, each of the mentioned steps integrates security as its essential objective of service evolution. Furthermore, to ensure that security requirements have not just been properly integrates in to a service but that the deployed environment offers the same, we have integrated security assurance assessment framework and CloudInspector to monitor key security aspects of both deployed service and infrastructure on which the service is being deployed.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Comp. Syst.*, vol. 25, no. 6, pp. 599–616, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.future.2008.12.001

[3] P. Mell and T. Grance, "The nist definition of cloud computing," 2011.

[4] I. Jacobson, G. Booch, J. Rumbaugh, J. Rumbaugh, and G. Booch, *The unified software development process*. Addison-Wesley Reading, 1999, vol. 1.

[5] I. O. for Standardization and I. E. Commission, "Systems and software engineering – software life cycle processes," International Organization for Standardization, Standard, July 2008.

[6] P. Forbrig, "Continuous requirements engineering and human-centered agile software development," in *Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, Research Method Track, and Poster Track co-located with the 22nd International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2016), Gothenburg, Sweden, March 14, 2016.*, 2016.

[7] C. B. Haley, R. C. Laney, J. D. Moffett, and B. Nuseibeh, "Security requirements engineering: A framework for representation and analysis," *IEEE Trans. Software Eng.*, vol. 34, no. 1, pp. 133–153, 2008. [Online]. Available: http://dx.doi.org/10.1109/TSE.2007.70754

[8] D. Mellado, E. Fernández-Medina, and M. Piattini, "A common criteria based security requirements engineering process for the development of secure information systems," *Computer Standards & Interfaces*, vol. 29, no. 2, pp. 244–253, 2007. [Online]. Available: http://dx.doi.org/10.1016/j.csi.2006.04.002

[9] C. Alcaraz and S. Zeadally, "Critical infrastructure protection: Requirements and challenges for the 21st century," *International Journal of Critical Infrastructure Protection*, no. 0, pp. –, 2014.

[10] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.

[11] B. Kandukuri, V. Paturi, and A. Rakshit, "Cloud Security Issues," in *Services Computing, 2009. SCC '09. IEEE International Conference on*, Sept 2009, Security, pp. 517–520.

[12] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.

[13] R. Piggin, "Are industrial control systems ready for the cloud?" *International Journal of Critical Infrastructure Protection*, no. 0, pp. –, 2014.

[14] M. Nanavati, P. Colp, B. Aiello, and A. Warfield, "Cloud Security: A Gathering Storm," *Commun. ACM*, vol. 57, no. 5, pp. 70–79, May 2014. [Online]. Available: http://doi.acm.org/10.1145/2593686

[15] M. Y. A. Younis and K. Kifayat, "Secure cloud computing for critical infrastructure: A survey," *Liverpool John Moores University, United Kingdom, Tech. Rep*, 2013.

[16] A. Khajeh-Hosseini, D. Greenwood, and I. Sommerville, "Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, July 2010, Service migration, pp. 450–457.

[17] A. Khajeh-Hosseini, I. Sommerville, J. Bogaerts, and P. Teregowda, "Decision Support Tools for Cloud Migration in the Enterprise," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, July 2011, Service migration, pp. 541–548.

[18] S. Kaisler and W. Money, "Service Migration in a Cloud Architecture," in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, Jan 2011, Service migration, pp. 1–10.

[19] C. Fehling, F. Leymann, S. Ruehl, M. Rudek, and S. Verclas, "Service Migration Patterns – Decision Support and Best Practices for the Migration of Existing Service-Based Applications to Cloud Environments," in *Service-Oriented Computing and Applications (SOCA), 2013 IEEE 6th International Conference on*, Dec 2013, Service migration, pp. 9–16.

[20] C. Haley, R. Laney, J. Moffett, and B. Nuseibeh, "Security requirements engineering: A framework for representation and analysis," *IEEE Transactions on Software Engineering*, vol. 34, no. 1, pp. 133–153, Jan 2008.

[21] D. Mellado, E. Fernández-Medina, and M. Piattini, "A common criteria based security requirements engineering process for the development of secure information systems," *Computer standards & interfaces*, vol. 29, no. 2, pp. 244–253, 2007.

[22] D. Mellado, E. Fernandez-Medina, and M. Piattini, "Applying a security requirements engineering process," in *Computer Security–ESORICS 2006*. Springer, 2006, pp. 192–206.

[23] T. M. Hesse, S. Grtner, T. Roehm, B. Paech, K. Schneider, and B. Bruegge, "Semiautomatic security requirements engineering and evolution using decision documentation, heuristics, and user monitoring," in *Evolving Security and Privacy Requirements Engineering (ESPRE), 2014 IEEE 1st Workshop on*, Aug 2014, pp. 1–6.

[24] D. S. Herrmann, *Using the Common Criteria for IT security evaluation*. CRC Press, 2002.

[25] D. C. Latham, "Department of defense trusted computer system evaluation criteria," *Department of Defense*, 1986.

[26] K. Rannenberg, "Recent development in information technology security evaluation-the need for evaluation criteria for multilateral security." in *Security and control of information technology in society*, 1993, pp. 113–128.

[27] C. Wagner, A. Hudic, S. Maksuti, M. Tauber, and F. Pallas, "Impact of critical infrastructure requirements on service migration guidelines to the cloud," in *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, Aug 2015, pp. 1–8.

[28] A. Hudic, M. Tauber, T. Lorunser, M. Krotsiani, G. Spanoudakis, A. Mauthe, and E. R. Weippl, "A multilayer and multitenant cloud assurance evaluation methodology," in *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*, Dec 2014, pp. 386–393.

[29] M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. R. Weippl, "Dark clouds on the horizon: Using cloud storage as attack vector and online slack space," in *20th USENIX Security Symposium, San Francisco, CA, USA, August 8-12, 2011, Proceedings*, 2011.

[30] H. Takabi, J. B. D. Joshi, and G. Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 24–31, 2010. [Online]. Available: http://dx.doi.org/10.1109/MSP.2010.186

[31] I. O. for Standardization and I. E. Commission, "Information technology – Security techniques – Information security management systems – Requirements)," International Organization for Standardization, Standard, July 2005.

[32] Nist and E. Aroms, *NIST Special Publication 800-53 Revision 3 Recommended Security Controls for Federal Information Systems and Organizations*. Paramount, CA: CreateSpace, 2012.

[33] M. Flittner, S. Balaban, and R. Bless, "CloudInspector: A Transparency-as-a-Service Solution for Legal Issues in Cloud Computing," *IEEE International Conference on Cloud Engineering Workshop*, 2016.

[34] "Payment Card Industry (PCI) Data Security Standard," Nov. 2013, https://www.pcisecuritystandards.org/documents/PCI_DSS_v3.pdf.

[35] I. Sedinic and Z. Lovric, "Influence of established information security governance and infrastructure on future security certifications," in *Information Communication Technology Electronics Microelectronics (MIPRO), 2013 36th International Convention on*, May 2013, pp. 1111–1115.