An Integrated Framework for Multi-layer Certification-based Assurance

Rajesh Harjani
University of Malaga
Malaga (Spain)
rajesh@lcc.uma.es

Marcos Arjona
University of Malaga
Malaga (Spain)
marcos@lcc.uma.es

Javier Espinar
University of Malaga
Malaga (Spain)
espinar@lcc.uma.es

Antonio Maña
University of Malaga
Malaga (Spain)
amg@lcc.uma.es

Antonio Muñoz
University of Malaga
Malaga (Spain)
amunoz@lcc.uma.es

Hristo Koshutanski
University of Malaga
Malaga (Spain)
hristo@lcc.uma.es

ABSTRACT

Complexity, dynamism and overlays in networks and systems are some of the main challenges we face nowadays when reasoning on systems' assurance and behavior. Security certification has shown to be a solid foundation to provide assurance and trust about system properties. This paper presents a certification framework for composite, layered and evolving systems, such as cloud systems or cyber physical systems. The framework's certification-based methodology defines a solid ground to provide security assurance aspects of these systems. The framework integrates two main domains of research: *(i)* certification, models and mechanisms (based on testing, monitoring, trusted computing, and hybrid evidences) for providing assurance of the system components and attesting properties of the composite systems; and *(ii)* software engineering, process, methodology and tools to enable developers engineer cloud applications with strong awareness and requirements on security assurance of underlying cloud platforms and services.

Keywords

Assurance, Security, Multi-layer Certification, Engineering Process, Monitoring, Testing, Trusted Computing.

1. INTRODUCTION

Most of the current trends and paradigms in computing systems (notably cyber-physical systems, Internet of Things or cloud computing) share a series of characteristics that greatly complicate the tasks of guaranteeing their behavior, especially in terms of security, dependability, privacy, etc. Among these characteristics, we highlight three essential ones:

Dynamism. Systems are not static anymore. At design time, system engineers do not have all the information they would need to design systems that fulfill their requirements, especially the non-functional ones like security, dependability, performance, etc. Moreover, they have to design systems that have both adaptation capabilities (involving short term reaction to better fit the current context and the system state) and evolution capabilities (involving long term reactions to keep the system aligned to its design goals and the external situation). In this situation there is no permanent and complete system implementation that can be used to apply thorough and rigorous code reviewing, testing, formal analysis, and other techniques to verify (ensure its quality and correctness) and validate (ensure fulfillment of requirements) these systems as a whole.

Composition: Most of the new computing paradigms and trends follow a component-based approach. Systems are created by integrating components both statically at development time and dynamically at runtime. These components are frequently coming from different providers, and sometimes remain under the control of such providers instead of the system owner. The evolution of these components is decoupled from the evolution of the systems in which they are used. Composition in these systems happens both vertically (between what we normally call layers) and horizontally (between components at the same layer).

Complexity: We have already mentioned that systems are larger, include more functionalities, require more guarantees, are interconnected to other systems forming Systems-of-Systems, are in continuous evolution, etc. The combination of these characteristics results inevitably in levels of complexity scaling up quickly.

Providing a practical approach to support assurance in complex, composite, layered and evolving systems requires the combination of different elements in a coherent and integrated way. In particular, a practical assurance approach for these types of systems requires at least mechanisms: (i) to provide static assurance based for the system components; (ii) to attest the dynamic state of system components (including the supporting hardware infrastructure); and (iii) to derive properties of the composite system based on the state and properties offered by components. In addition to these, we also believe that any successful approach must be complemented with engineering processes, methods and tools to support developers of such systems to take full advantage of the approach.

Recent extensions and improvements to several existing technologies like certification, trusted computing, monitoring and reconfiguration provide a solid basis to develop an integrated layered assurance framework to support the assurance of complex, composite, layered and evolving systems in practice. In this paper, we present such approach, show how it is applied to cloud computing and discuss the challenges of future application to other types of systems, with a particular focus on cyber-physical systems.

2. INTEGRATED CERTIFICATION-BASED ASSURANCE FRAMEWORK

A common approach in enhancing assurance and reducing risks in the light of such uncertainties is to rely on the certification of the

different components and artifacts that constitute the potentially complex and fast changing nature of our target systems. Therefore, the main goal of the proposed approach is to develop an integrated framework of models, processes and tools supporting the certification-based assurance of security properties for layered computing infrastructures.

Assurance of cloud-based applications and services allows service consumers and providers to ascertain that the service properties provided in the certificates guarantee continuous compliance with their own requirements [1][2][3]. This increases consumers' and providers' confidence that their required level of assurance is being kept, before becoming involved in service design, deployment, and access on cloud.

With this purpose, the framework relies on multiple types of security evidences (e.g., testing, monitoring, trusted computing) used for certificate issuing, and includes relevant mechanisms for generating the evidence supporting a security property and for the secure communication of these evidences between different components within the certification infrastructure [2][3]. This evidence communication is supported by Trusted Computing (TC) [4] mechanisms providing means to establish integrity (authenticity) of evidence, and subsequently verify if the captor integrity holds (can be trusted). Whenever possible, evidence gathering is build upon existing standards and practices (e.g., interaction protocols, representation schemes etc.) regarding the provision of information for the assurance of security in clouds.

Furthermore, the framework supports the generation of hybrid certificates based on the combination of different types of evidences, including testing and monitoring data, and trusted computing platform proofs [5][2]. Hence, it supports decision making in business and societal contexts, which, due to existing legislation, established societal and business practices or individual preferences, might require and accept evidence of specific degrees of formality regarding a security property of a cloud service before this service can be used. This leads to cover security properties to an unprecedented extent and increase the overall confidence in the use of cloud computing.

To address the aforementioned security problems, several partners from European science and industry have joined efforts in the CUMULUS[1] (Certification infrastrUcture for MUlti-Layer cloUd Services) research project to investigate how to improve assurance, security and trustworthiness of multi-layer cloud services facing end users.

In its current implementation, the integrated framework allows service users, service providers and cloud suppliers to work together with certification authorities in order to use security certificates for deriving dynamic assurance evaluations in the ever-changing cloud environment. To achieve this, the proposed approach focuses on the following tasks:

- Definition, development and realization of advanced models for certification-based assurance of security properties based on evidences drawn from service testing and operational monitoring, as well as on trusted computing platform proofs. This facilitates the task of how to address a layered assurance framework given the complexity of interactions of cloud services.
- Development of an interoperable certification infrastructure for generating, maintaining and using certificates according to the different types of the certification models developed, so that to make them available to cloud providers and cloud customers.
- Development of an engineering process supporting the development of (i) cloud services in a way that facilitate their certification through a semi-automatic process and (ii) applications taking advantage of those services.
- Evaluation of the certification framework to ensure its technical soundness and industrial applicability, in particular for SmartCities and eHealth domains.
- Delivery of an interoperable certification solution and contribution to existing standards (e.g., interaction protocols, representation schemes etc.) regarding the provision of information for the assessment of security in clouds.

As a framework that aims to support the certification-based assurance of security properties in clouds at infrastructure, platform and software application layer services, the CUMULUS architecture is structured as an infrastructural overlay of the monitored payload system. The overlay is implemented by components, which provide the hooks to the monitored cloud system. Figure 1 shows the CUMULUS multi-layer certification-based assurance and infrastructure high-level overview including several conceptual layers and artifacts:

- Certification Aware Service/App Engineering Tools: providing means for supporting the engineering of cloud services and applications that can make use of the framework. This is a tool capable of interacting with the Infrastructure, and in particular with the different repositories, in order to take advantage of Certified Services.
- Certification Infrastructure: producing test, monitoring and trusted computing based multi-layer, incremental and hybrid certificates. The Certification Manager will realize this by making use of different Certification Models containing the necessary requirements and guidance to support the generation of certificates.
- Evidence Generation and Communication: for the provision of the certification evidences, it is where the components producing core test, monitoring and trusted computing based evidences are deployed.

The multi-layer certification-based assurance starts from the physical platform's Trusted Platform Module (TPM) [6] where platform integrity measurements are stored. The physical platform assurance (TC assurance) provides the basic building block over which the compositional layered assurance of the higher levels of the cloud system is built upon. Each of the higher levels of a cloud system has it own certification-based assurance provided by test, motoring, TC or hybrid certification models, which provide an assurance building block for next (higher) level of the cloud system and corresponding certification models.

3. CERTIFICATION-BASED ASSURANCE BUILDING BLOCKS

To achieve the proposed goals, namely providing means for evidence generation, communication and combination for assurance, we have developed different certification-based assurance building blocks [2][5][7]. These building blocks are based on testing, monitoring and trusted computing methods.

[1] http://www.cumulus-project.eu

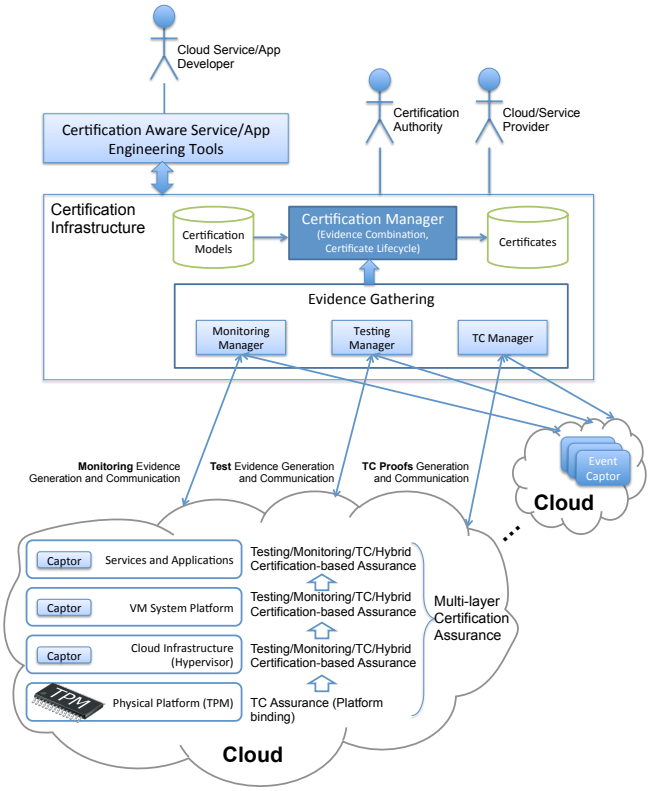

Figure 1. CUMULUS multi-layer certification-based assurance

By using testing mechanisms we can obtain static and dynamic evidences, however monitoring and trusted computing proofs (based on TPM) are clearly focused on collecting dynamic evidences. We make use of both, static and dynamic certificates, as well as TPM remote attestation as the elements for secure evidence communication. For each specific case a certification authority is responsible of the evidence combination and encodes the resulting properties in a certificate. It is important to notice that the combination is not automatic; it is the responsibility of an authority to perform it case by case. The Certification Authority analyses all involved components in the combination with their respective properties, and the resulting one from the combination is also particularly analyzed for it. This analysis can be done by different ways: checking the code, testing, monitoring, etc. The certification models are designed to allow the combination of different properties, making unnecessary the use of external rules to check the validity of the certificates.

Test-based certificates rely upon the results that are extracted of executing tests on the targeted services/software [8][9] as well as on composite services [7]. The Certification Authority guides the tests according to the target of certification in order to know if the software holds a certain property. This kind of certificates can be based on static or dynamic proofs; the static tests are performed offline while the dynamic tests are executed once the software is in a production environment. Therefore, these certificates may include both dynamic and static evidences according to the kind of tests used to extract the proofs. Software testing is performed by testing agents/captors their main task consists of injecting the test cases and collecting the corresponding results to compose the evidences. Testing Captors implement functionalities for both static and dynamic collection of evidences and they can be running on the same system or in another external to the software we want to certify.

Monitoring-based certificates are clearly focused on extracting dynamic evidences [10][11]; the monitoring operations are, by definition, continuous and they have to be performed once the software is deployed and accessible to user [12]. The proofs that will be included in such certificates can cover contextual conditions (e.g., co-tenant software, optimization strategies, network status...) that might not be possible to extract in a pre-production environment. The entities responsible for the monitoring process are the monitoring agents/captors; they capture all events and check if these events are compliant with the assertion included in the certification model [10].

The third type of certificates is based on Trusted Computing mechanisms that are used to provide hardware-based support for securing computing platforms, allowing certification authorities to verify that only authorized code runs on a system [13][14]. A TC mechanism usually is implemented using a TPM chip, which is integrated into the hardware of a platform (such as a PC, a laptop, a PDA, a mobile phone). TPM can be accessed directly via TC commands or via higher layer application interfaces (the Trusted Software Stack, TSS).

We consider two main scenarios for TC-based certification; in these scenarios the trusted computing mechanisms are used to protect both the integrity of the software and the underlying platform (including software and hardware) [14]. In the first scenario, the TC certification model is not used as an independent model but we make use of this building block to provide the trust that is needed for the validation of the Monitoring and Testing based certification. This means that the TC is not employed to directly certify a security property but instead used to increase the trust in other types of certifications. For instance, in a test-based certificate, it can be used to prove that the platform configuration at runtime is the same as the one used during testing in a pre-production environment, or it can also be used to ensure that the agent or monitoring captors have not been modified, meaning that they are extracting the proofs correctly. In the second scenario, the TC certification model will be used as an independent model to certify either platforms or services that are running in distributed systems. We would like to remark that this second use case is a generalization of the first scenario in such a way that allows to protect the integrity of more heterogeneous platforms and services, instead of only the monitoring and testing agents involved in the other CUMULUS certificates.

Each of the three types of certificates complies with the CUMULUS Meta-Model. This Meta-Model has a modular structure to represent: the property vocabulary, the certificate itself (including assertions, evidences, context...) and the certification model.

The combination of different evidences, to cover all components in a heterogeneous distributed system, can be carried out by a certification authority easily since all types of certificates conform to the same Meta-Model. This common structure fosters the combination of different properties to compose new ones avoiding the use of external rules to check the validity of the certificates. In addition, the dynamic testing, monitoring and trusted computing techniques allow these certificates to provide dynamic assurance of the properties they contain. These certificates are able to transit from one state to another, in their lifecycle, based on the dynamic evidences; for example, a valid certificate can be revoked if the monitoring agent captures an incompatible event with the

included assert, or if the trusted computing mechanisms are no longer able to prove the integrity of the software and/or the underlying platform.

4. ENGINEERING PROCESS

On the basis of the insights exposed, both the foundations and the powerful assurance artifacts sustaining the capabilities of this proposal, we need a solid platform ready to combine sources of security knowledge with the means to create assurance building blocks close to the end-users terms and requirements. Such complexity cannot be faced as a whole with a single threaded approach but instead it has to be addressed as a compositional set of engineering tools and activities that converge together into a coordinated and security-enriched process, giving solution to these multidisciplinary issues.

This Service Engineering Process (SEP) [15] sets out different guidelines to drive all the activities during the complete lifecycle of service development. The most prominent virtues and features allow *(i)* to interact with the different security areas and experts to gather security knowledge into machine processable data; *(ii)* to express this information in form of security requirements close to customers, developers and cloud system engineers,*(iii)* to define adequate compositions of techniques, certified services and building blocks to provide validated assurance solutions for those requirements and *(iv)* to support the deployment, usage and integration of these security solutions into cloud services and applications.

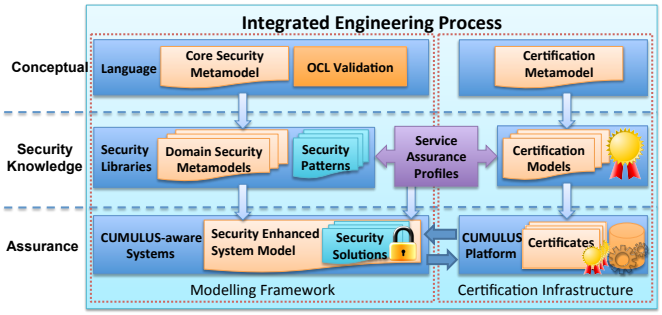

Figure 2. Integrated Engineering Process

Therefore the main objective of the SEP is to support the development of services and applications compliant with predefined certification models, in order to flexibly integrate certified security properties along with the necessary security knowledge to assure the target systems. Figure 2 shows the big picture of the Engineering workflow, defining several layers of abstraction (horizontal) and different elements at same level (vertical). Following, we use this figure as a basis for a brief description of the SEP and its main activities.

As we have stated, the SEP requires a very thoughtful approach to the multiple activities involved in the assurance methodology. Therefore a hierarchical strategy has been defined based on previous research and use cases experiences [16][17] defining three abstraction layers, each one composed of different artifacts:

- Conceptual: The top layer of the process stack lays down the common modeling language to express security knowledge, along with a collection of OCL rules [18] to validate the usage of the language. On the other hand, the certification metamodel and their schematics are conceptually described at this level.
- Security Knowledge: The middle layer uses the concepts of the abstract layer to express the knowledge and experience of security experts into libraries and to elaborate certification models aiming the production of certified services for cloud applications. Both artifacts operate across the Service Assurance Profiles explained later below. The actions of this layer are intended not only to help cloud app developers to integrate security knowledge by design, but also to make the system a better candidate for certification-based assurance.
- Assurance: The bottom layer reflects the final stage of the SEP, where all the security libraries and their solutions are used to transform or reengineer legacy cloud applications into certification-aware systems. This stage implies several transformations based on the security knowledge, and the resulting application will be able to interact with the CUMULUS platform deployed in the cloud infrastructure in order to retrieve or reconfigure certified services. Those services, as explained in the previous section, have been created and accredited by certification authorities, which guarantee the chain of trust.

The layered approach of SEP requires several modules to interact at horizontal level in the same abstraction level but also defines the workflow to move and export resources and knowledge between different vertical layers. This significant amount of activities are not perceived by users because these processes are managed and performed by the different supporting tools of the SEP, depending on their goal and stage in the overall methodology. All of these together make an integrated framework that covers all expected functionalities, dividing the methodology in two different responsibility areas, the Modeling Framework and the Certification Infrastructure. Both connected by Service Assurance Profiles (SAP).

The Modeling Framework has been implemented as a plugin of MagicDraw [19] to support all the designing and modeling activities required in the SEP along with the transformation and validation of the intermediate stages. The results of this software are the sum and composition of the following elements in the three levels of progression:

- The Core Security Metamodel (CSM) and the OCL Validation rules establish the proper and valid definition of UML elements to represent the security knowledge, defining the language, characteristics and mandatory attributes to describe the information for further automatic machine processing.
- Domain Security Metamodels (DSM) and Security Patterns [20] express security knowledge and solutions respectively to fulfill security requirements using the previous language. Security experts create those security libraries gathering security knowledge in compliance with the environment of their applications (company policies, standards, etc.) for a particular domain and they include well-known solutions and solvers in form of security patterns.
- Security Enhanced System Models are obtained once security libraries have been applied to improve the cloud system models with the security measures and solutions, fulfilling the security requirements of the system by means of certification requirements and claiming remote certified services from the CUMULUS platform.

With respect to the certification assurance operations, an interoperable Certification Infrastructure for generating, maintaining and using certificates according to the different types of certification models have been described in Section 3. To

achieve this goal the SEP requires a deployed CUMULUS platform in the cloud infrastructure allowing the interaction with CUMULUS-aware systems to offer and respond to service request by the CUMULUS enhanced systems.

The SAPs are auxiliary artifacts designed to establish the anchorage point between the two categories of security knowledge feeding the engineering process. SAPs aim to link specific details of certification requirements included in the security libraries, with the wide spectrum of certification mechanisms and certified services, most of these with similar goals, certification models, evidences or assurance goals, registered by different certification authorities. Therefore, SAPs have been created to introduce a discrimination of the expected security necessities and preferences, as security experts have the freedom to select the most proper assurance approaches and entities for their security solutions, based on reliability, trustworthiness and efficiency parameters these experienced users have obtained.

5. OUTLOOK AND FUTURE WORK

Cyber Physical Systems (CPSs) – systems created as a federation of smart, cooperative, sensing devices – are starting to play an important role in the everyday life of citizens, connected both to ICT systems and to the physical world. While the notion of sensors gathering data is not new, the sheer amount of new devices, the amount of data they can now gather, their data processing capabilities and the fact that they are all becoming connected to the Internet of Things, enables exciting, new services. The components of a CPS, whether ICT or non-ICT ones, may operate under distributed ownership and control, and within uncontrolled and unprotected physical environments, characterized by changing operational conditions and constraints (e.g., changing temperatures, physical damage, changes to power supply etc.). They may also operate within the remit of different and not always harmonized jurisdictions and transfer data across them. Furthermore, the ICT components of CPSs may have diverse computational features and roles. As a consequence of these factors, CPSs may often:

- Be vulnerable to security attacks and adverse operating context conditions that can compromise the availability and security of some of their components (e.g., local sensors, network components, application level components etc.);
- Generate, make use of and inter-relate massive personal data in ways that can potentially breach legal and privacy requirements;
- Experience frequent and unpredicted changes in the components and infrastructures that they rely on, which can compromise the security, resilience and availability of their operations and/or the service(s) that they offer.

Preserving quality, security and privacy (QSP) properties in CPSs under the above circumstances is a particularly challenging scientific and engineering problem. Hence, engineering CPSs in ways that can simultaneously guarantee all QSP properties of interest becomes a challenging problem requiring an integrated CPS design, development, monitoring, and adaptation approach.

A CPS engineering approach needs also to be aware of and support effectively the composition of software and physical components. This composition is inherently different from traditional software-based/service-based systems compositions. This is because software centric approaches to composition (e.g., software service/components orchestration workflows) are often not appropriate for CPS systems or at least the physical layer of their implementation stack due to their intrinsic complexity that often cannot be supported in the case of embedded systems.

An assurance-oriented engineering approach like the one presented in this paper, that could be applied to the development of CPS would clearly represent an important advance. Based on the current results obtained from the application of our approach to cloud systems, and its capacity to solve several of the problems we have just mentioned, we have started to develop extensions and adaptations to deal with the engineering of CPS. In particular, we have already extended the concept of security pattern to be able to represent solutions for CPS [21]. Ongoing work focuses on the representation of the dual nature of CPS by using layers that allow us to provide different engineering views for the physical and cyber layers of a CPS while maintaining their interrelations.

6. CONCLUSIONS

We have presented a layered assurance framework for certification of security properties of complex, layered and dynamically evolving systems, such as Cloud-based systems. The framework's certification-based assurance methodology provides a solid ground to manage security aspects of these systems. The cornerstones of the framework are the certification models (based on testing, monitoring, TC, and hybrid evidences) which drive certificates lifecycle, and the evidence gathering and composition for certification.

The framework provides to certification authorities a comprehensive tool set to enable effective cloud certification on a number of relevant and important security properties; and to Cloud app developers a comprehensive methodology and tools to engineer cloud applications with strong awareness and requirements specification on security assurance of underlying cloud platforms and services.

The security engineering methodology provides not only a source of specialized structured security knowledge for system engineering, but also compliance to specific certification models used to certify systems' properties. Thus, following the engineering process, a system designer will not only embed system's security aspects by design, but will also embed modular security and assurance of system's components for more effective system certification.

Cyber-physical systems with their complex and composite nature can well leverage on the framework's certification models, mechanisms and methodology to successfully handle security assurance of their (network of) interactive elements. The framework's certification-based assurance is envisaged to provide an important foundation towards a more comprehensive cyber security framework.

7. ACKNOWLEDGMENTS

This work was undertaken as part of the CUMULUS project funded by the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 318580.

REFERENCES

[1] A. Sunyaev, S. Schneider. Cloud Services Certification. *Communications of the ACM, Vol. 56 No. 2*, 2013, Pages 33-36.

[2] G. Spanoudakis, E. Damiani, A. Mana. Certifying Services in Cloud: The Case for a Hybrid, Incremental and Multi-layer Approach. Proc. of 2012 *IEEE 14th International Symposium*

on High-Assurance Systems Engineering (HASE), Omaha, NE, pp. 175-176, 2012.

[3] S. Cimato, E. Damiani, R. Menicocci, F. Zavatarelli. Towards the certification of cloud services. Proc. of *IEEE 2013 International Workshop On Security and Privacy Engineering, Assurance, and Certification (SPEAC 2013),* Santa Clara, CA, pp. 100-105, 2013.

[4] Chris Mitchell, Trusted Computing, Institution of Electrical Engineers, 2005.

[5] S. Katopodis, G. Spanoudakis, K. Mahbub. Towards Hybrid Cloud Service Certification Models. Proc. of 11th *IEEE International Conference on Services Computing*, USA, 2014.

[6] Trusted Platform Module (TPM) Specifications. Trusted Computing Group. http://www.trustedcomputinggroup.org/resources/tpm_main_specification

[7] M. Anisetti, C.A. Ardagna, E. Damiani. Security Certification of Composite Services: A Test-Based Approach. Proc. of the 20th *IEEE International Conference on Web Services (ICWS 2013)*, San Francisco, CA, USA, June--July, 2013.

[8] S. Hanna and M. Munro. An approach for specification-based test case generation for web services. In: Proc. of the *IEEE/ACS International Conference on Computer Systems and Applications (AICCSA 2007*). Amman, Jordan, May 2007.

[9] M. Anisetti, C. A. Ardagna, E. Damiani and F. Saonara. A Test-based Security Certification Scheme for Web Services, *ACM Transactions on the Web, Volume 7 Issue 2*, May 2013.

[10] M. Krotsiani, G. Spanoudakis, K. Mahbub. Incremental Certification of Cloud Services. In *7th Int. Conf. on Emerging Security Information, Systems and Technologies (SECURWARE 2013)*, Barcelona, pp. 72-80, 2013.

[11] H. Foster, G. Spanoudakis and K. Mahbub. Formal Certification and Compliance for Runtime Service Environments. In 9th *IEEE International Conference on Service Computing*. 2012.

[12] Ghezzi C., Guinea S. (2007), Runtime Monitoring in Service Oriented Architectures, In Test and Analysis of Web Services, (eds) Baresi L. & di Nitto E., Springer, 237-264, 2007.

[13] N. Santos, K.P. Gummadi, and R. Rodrigues. Towards trusted cloud computing. In HotCloud, 2009.

[14] A. Muñoz, A. Maña. Bridging the GAP between Software Certification and Trusted Computing for Securing Cloud Computing. In *IEEE International Workshop on Security and Privacy Engineering, Assurance, and Certification (SPEAC 2013)*, June, 2013.

[15] M. Arjona, R. Harjani, A. Muñoz, and A. Maña. An Engineering Process to Address Security Challenges in Cloud Computing, 3rd *ASE International Conference on Cyber Security*, 2014.

[16] J. F. Ruiz, A. Maña, M. Arjona, and J. Paatero. Emergency Systems Modelling using a Security Engineering Process. *3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH)*. SciTePress, 2013.

[17] J.F. Ruiz, A. Rein, M. Arjona, A. Maña, A. Monsifrot, and M. Morvan. Security Engineering and Modelling of Set-Top Boxes, 2012 *ASE/IEEE International Conference on Biomedical Computing (BioMedCom)*.

[18] M. Arjona, C. Dania, M. Egea and A. Maña. Validation of a security metamodel for development of cloud applications. *14th International Workshop on OCL and Textual Modeling Applications and Case Studies (OCL 2014)*. 17th International Conference on Model Driven Engineering Languages and Systems (MODELS'14).

[19] MagicDraw Modelling Tool. http://www.nomagic.com/products/magicdraw.html

[20] M. Arjona, , J.F. Ruiz and A. Maña. Security Patterns for Local Assurance in Cloud Applications. *International Workshop on Engineering Cyber Security and Resilience ECSaR'14*. The Third ASE International Conference on Cyber Security 2014.

[21] A. Maña, E. Damiani, S. Gürguens, G. Spanoudakis. Extensions to Pattern Formats for Cyber Physical Systems. Proceedings of the *31st Conference on Pattern Languages of Programs (PLoP'14)*. Monticello, IL, USA. Sept. 2014.