Patterns for Security and Privacy in Cloud Ecosystems

Eduardo B. Fernandez
Dept. of Comp, and Elect. Eng. and Comp. Sci
Florida Atlantic University,
Boca Raton, FL, USA
ed@cse.fau.edu

Nobukazu Yoshioka
GRACE Center
National Institute of Informatics
Tokyo, Japan
nobukazu@nii.ac.jp

Hironori Washizaki
Waseda University
Tokyo, Japan
washizaki@waseda.jp

Abstract— **An ecosystem is the expansion of a software product line architecture to include systems outside the product which interact with the product. We model here the architecture of a cloud-based ecosystem, showing security patterns for its main components. We discuss the value of this type of models.**

Index Terms—**Software ecosystems, systems security, security patterns, cloud computing, reference architectures.**

I. Introduction

A cloud-based computing system involves a variety of users and devices connected to it and provides multiple kinds of services. Typically, clouds provide three levels of service: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Consumers can use any of these service levels depending on their objectives.

Clouds usually do not work in isolation but interact with other clouds and with a variety of associated systems. The associated systems are a growing set, where new types of products appear and provide some useful functions for some types of users.

An ecosystem is the expansion of a software product line architecture to include systems outside the product which interact with the product [10]. An ecosystem is advantageous to suppliers who can offer a larger variety of products or services, and to consumers who can find more products to help them reach their business goals. Several companies are developing ecosystems around their products, e.g. Cisco [15].

A convenient representation when building or using cloud ecosystems and similar complex systems is the use of architectural models based on patterns. In particular, security requires a holistic view of a system and a global architecture is of great value to define security aspects. We first describe the ecosystem in the form of a pattern diagram and then describe its components as patterns and reference architectures using UML models. We believe these are the first precise models of cloud ecosystems. Some of the components of this system have been already modeled as patterns but some are missing, we identify here the new patterns we need. We discuss the value of these models with respect to several security objectives. We do not claim completeness, an ecosystem is open-ended and our model is a first step in their architectural representation.

This paper is organized as: Section II describes some background; Section III presents a pattern diagram for the cloud ecosystem. Section IV presents UML models of some patterns for the components. We consider the value of these representations in Section V and we discuss how to expand the ecosystem and some pattern representation issues in Section VI. Section VII considers related work. We end with conclusions in Section VIII.

II. BACKGROUND

A *pattern* is a solution to a recurring problem in a specific context. Patterns can be used to design and analyze complex systems, to capture design decisions, and to evaluate new or existing systems. Patterns can also improve software quality by promoting reusability, scalability, and consistency. Patterns are categorized as *analysis patterns* [25], *design* and *architectural* patterns [12], and *security* patterns [19]. Several classifications exist for security patterns [34]. Pattern solutions are usually described using modeling languages such as UML, maybe combined with formal languages such as OCL. Pattern descriptions usually include class diagrams, sequence diagrams, and state diagrams. *Abstract patterns* can be analysis patterns or *Semantic Analysis patterns* (SAPs), when they describe conceptual business aspects, or *abstract security patterns* (ASPs), which realize one or more security policies able to control (stop or mitigate) a threat or comply with a security-related regulation or institutional policy [20].

A *Reference Architecture* (RA) is a standardized, generic software architecture, with no platform dependencies, valid for a particular domain [3]. An RA should define the fundamental components of a system and the interactions among these units. It may also include use cases. An RA can be built of patterns and then considered itself a pattern.

III. A cloud ecosystem

Figure 1 shows a cloud ecosystem. The Cloud Reference Architecture (Cloud RA) is the main pattern that defines the ecosystem (the *hub*) [22]. This can be converted into a Cloud Security RA (Cloud SRA) by adding security patterns to control its identified threats. The Cloud SRA includes, among others, patterns for Authentication, Authorization, and Logging [19]. The Cloud Compliant Reference Architecture includes patterns that describe regulation rules to the Cloud RA. We just completed a HIPAA-compliant RA [35]. Several regulations, including HIPAA, have provisions for individuals' privacy. We can build also a Cloud RA with stringent controls on privacy.

Threats can be enumerated in several ways and we use an approach based on activities in an activity diagram describing its use cases [19]. Cloud Web Application Firewalls and Security Group Firewalls provide filtering functions that can be provided as services through NFVs or on their own. Rules of regulations can be described by patterns.

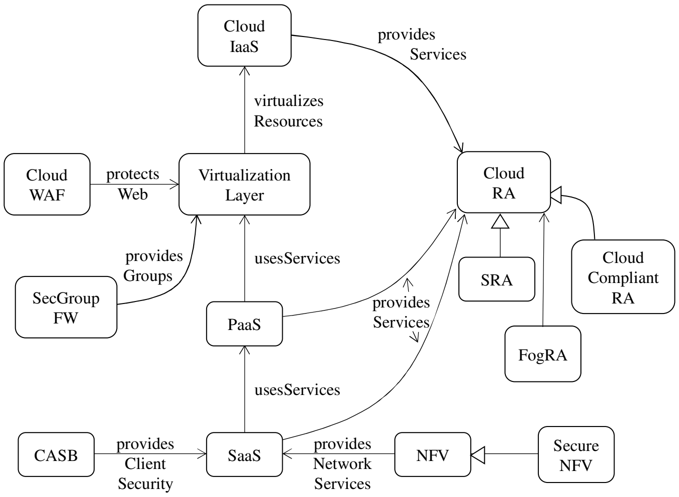

Fig.1. A cloud ecosystem

The service layers of a cloud are themselves compound patterns and we have written patterns for IaaS, PaaS, and SaaS [26]. They provide the services sold by the cloud provider. The figure shows SaaS as built over PaaS, which is the common practice, but it can also be built directly from IaaS (not shown for clarity).

Telecommunication companies have discovered that they can provide services to their customers by building their networks as services rented from some cloud provider [9]. The provision of network functions using virtualization, Network Functions Virtualization (NFV), is a network architecture where node functions such as load balancers, firewalls, IDS, and accelerators are built in software and offered as services. Each Virtualized Network Function (VNF) may use one or more virtual machines or containers running different software. Typically, NFVs come with some security mechanisms but which ones depend on the vendor. To make the model more flexible we show a pattern for NFV without security and a derived pattern for a Secure NFV [24].

Cloud Access Security Brokers (CASBs) are security enforcement points between consumers and service providers that apply security controls to access cloud services, usually SaaS services [23]. Figure 1 shows also a Fog RA discussed in Section V.

There is no pattern yet for the Virtualization Layer, although we defined a pattern for a Virtual Machine Operating System [19]. An important lower-level pattern for this function is an OpenStack pattern, not yet written [31]. Some systems come with an attached set of security mechanisms; however, we prefer to separate the security mechanisms from a basic system and define a secure version of the function, e.g. Secure NFV. The virtualization layer may be put as part of the IaaS layer or not.

IV. MODELS OF THE ECOSYSTEM COMPONENTS

After building a pattern diagram with the components of the ecosystem we need to build detailed models for the components. We show three examples of the patterns in the ecosystem and a part of the SRA. None is shown completely, the idea is to show their main functions, the complete descriptions can be found in their respective references. Pattern descriptions also include sections on Forces, Consequences, Implementation, and Related Patterns. Note that the intents of the patterns have already been mentioned and are repeated because they are a basic part of a pattern description. The idea here is that we can build patterns for every participant in the ecosystem, which provides a unified view of the complete system.

Cloud Access Security Broker

Intent

Cloud Access Security Brokers (CASBs) are security enforcement points between consumers and service providers that apply security controls to the consumer's users to access cloud services, usually SaaS services. They may also control access to internal company resources. Security controls may include authentication (credentials and passwords), authorization policy enforcement, intrusion prevention, antimalware filters, security logging/auditing, and encryption.

Solution

Use an intermediary system, called a CASB, which provides security controls (authentication and authorization), can monitor the use of services by users, and can perform malware detection when users access cloud applications. Additionally, other services such as performance, identity, and search can be provided. Figure 2 shows the class diagram of the CASB. *Consumers* (users) request services through the *Broker*, which in turn gets them from one of the *Service Providers.* The Broker includes a set of security mechanisms such as a *SecurityLogger/Auditor*, an *Authorizer*, an *Authenticator,* an *Encryptor,* and maybe others. Consumers and CASBs can be mutually authenticated. The CASB enforces rights for the consumers when they try to access an application. *Internal Resources (applications)* can also be controlled by the CASB. An *Identity Federation* provides identifiers for consumers and SPs to support authentication. Figure 3 shows the sequence diagram for the use case "Access an application service": a consumer requests a service to the CASB, which invokes an authentication protocol, when authenticated the consumer can access the service if authorized for it; this interaction is logged.

The CASB enforces institution policies in any access as well as protecting against malware. In other words, the CASB is an extended Reference Monitor [19].

Known uses

- Adallom [1]—integrates with the authentication services in SaaS to let institutions monitor the activities of users in any location and any device. This product includes a behavior analysis component to assess the possible risk of specific transactions.
- Bitglass [7]—provides RBAC, encryption, session control, identity, and DRM.
- Cipher Cloud [14]--protection controls include encryption, tokenization, monitoring, data loss prevention, and malware detection.
- Elastica Cloudsoc [17]—provides authentication, authorization, monitoring, and other services. It can interact with third-party APIs.

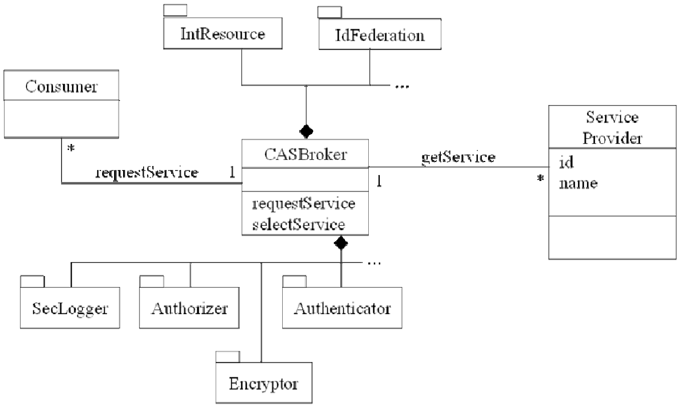

Fig.2 Class diagram of the CASB pattern

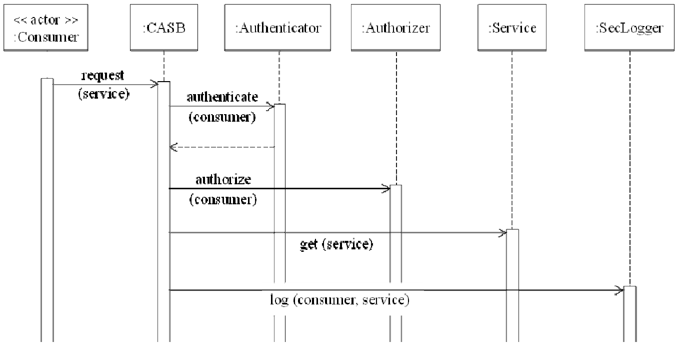

Fig.3 Use case “Access an application service”

Network Function Virtualization

Intent

Network Functions Virtualization (NFV) is an architecture for the construction of services using software building blocks that can be used to assemble network systems. The building blocks, *Virtual Network Functions (VNFs)*, are created from cloud software services.

Solution

Service providers (SPs) can use SaaS services to provide network functions where we simulate the specific functions of a hardware device using software running in one or more virtual machines or containers in the cloud. We can build network services using the PaaS and IaaS services and deploy them as SaaS services. These network services can then be used for the needs of the applications being accessed by the Consumers. The Network Services are provided by a company which in turn rents virtual hardware from the cloud provider.

Structure

Figure 4 shows the class diagram of the NFV pattern. The virtualization layer creates *virtual network services (VNFs)* implemented using one or more *virtual machines.* While the SaaS level is used to provide the services to telecommunication companies (Consumer1…N), the provider of these services would build them using the PaaS of the cloud, utilizing one or more virtual machines to implement each service, or directly from the PaaS services.

Known uses

- Cisco’s ESP uses NFV, SDN, open APIs, and advanced orchestration capabilities to create a flexible and modular platform [16].
- Alcatel-Lucent has a variety of NFV products [2].
- Ericsson has implemented VoLTE (Voice over LTE) services using NFV. They have also extended Neutron, the OpenStack API with appropriate functions [27].

Patterns for HIPAA compliance

We wrote patterns for HIPAA compliance that describe the structure of its rules [21]. That paper described the Privacy Rule and the Security Rule of HIPAA, which we represented as patterns. Privacy in HIPAA and other regulations means protection through Authentication and Authorization of the patient information plus notification of the use of this information. Figure 5 shows another HIPAA rule not included in that paper, the Transactions and Codes Sets rule. A *Patient* treatment results in a set of *Transactions (TX)* performed by different *Covered Entities*. Each transaction is logged by a *SecurityLogger/Auditor* pattern [19], and later all the transactions related to this and other treatments can be audited by an *Overseer* who verifies the proper use of identifiers and codes.

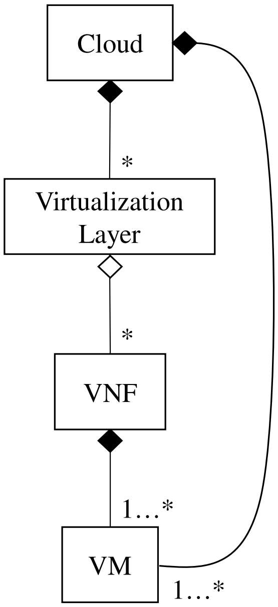

Fig. 4. Class diagram of the NFV pattern

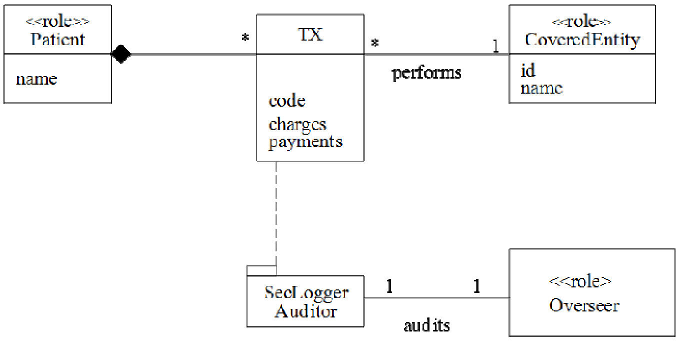

Fig. 5. Class model for Transactions and Code Sets Rule

Security Reference Architecture

We built a Cloud SRA [22]. Analyzing threats and neutralizing them with patterns we arrived to secure units of the SRA. Figure 6 shows a class model for one of its parts, the secure VM image repository system. The *Virtual Machine Image Repository* holds a set of *Virtual Machine Images* (VMI) that can be used to instantiate virtual machines. The *Reference Monitor* uses a *Filter* that scans all VM images before being published or retrieved. The *Authenticator* is an instance of the Authenticator Pattern that allows the Reference Monitor to authenticate the users that access the repository, who can publish or retrieve images if the Authorizer authorizes them. The Reference Monitor pattern enforces the authorization rights defined in the *Authorizer.* The *Security Logger/Auditor* keeps track of accesses to the repository.

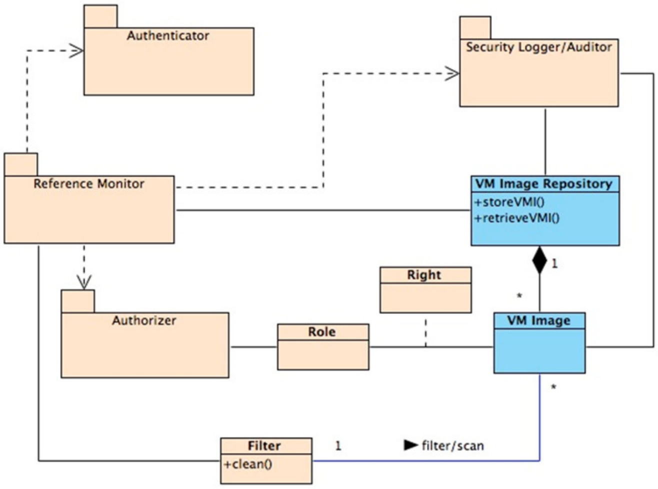

Fig. 6. Secure VMI Repository System

V. VALUE OF THE MODELS

These models have a series of advantages, we discuss a few now. First, *control of heterogeneity*: The involved components come from different vendors or even different countries. This means a variety of standards and protocols. An abstract model can unify this heterogeneity and can provide a way to apply a uniform security model to all the components.

A holistic security view. Several authors, e.g. [11, 18], emphasize the need to develop secure systems in a holistic way. Systems built piecemeal omit important interactions that may result in vulnerabilities. Recently, companies have started to realize of the value of holistic approaches to security [15]. An ecosystem provides such a holistic view by indicating the places where security mechanisms can be attached and their effect in the functional parts of the architecture. We can expand the UML model of the ecosystem by indicating all the points where threats are neutralized with corresponding security patterns. We can trace the propagation of attacks and study where to place defenses for greater effect. Many threats result of the interaction of different units and cannot be discovered by analyzing each unit in isolation. Privacy rules are defined in the clouds but we need to make sure that interactions with the components still respect these rules. *Other quality factors*: Holistic views are very important to combine quality factors such as safety or reliability with security.

Compliance with standards and regulations. An RA can be used to support security standards and regulations, which can be described as policies which in turn can be implemented as patterns and made part of the SRA. The ecosystem helps architects or designers to identify what components of the cloud system are associated with the standard and can be used to comply with the specific rules of the standard. Applications derived from the SRA will automatically comply with the standards or regulations. Relating specific regulations to specific security mechanisms can be used to demonstrate compliance.

Cloud security management. The functions of such a system include determining assets, consideration of regulations, policy definition, and privacy. A model oriented to fulfill the ISO 27000 security regulations is given in [5], which makes use of similar type of models.

VI. New functions

New entities to add include Fog Computing [9], which is a highly virtualized platform that provides compute, storage, and networking services between end devices and Cloud Computing Data Centers. Fog computing systems are key systems for the Internet of Things, they can control for example smart grids or traffic lights [33].

Cloud functions are increasingly using SDN; for example, SDN can be used to implement NFV. No patterns for SDN have appeared yet. We did not include here a standard Cloud Broker, intended to let consumers access several clouds. This is not the same as the CASBs we show in Figure 1.

The metamodel of Figure 7 relates the security concepts we used in the ecosystem, it does not include privacy concepts but these functions can be used to enforce privacy constraints. *Threats* take advantage of *Vulnerabilities* that can exist in any cloud service level. Threats come from analysis of *Use Cases* or from published *Threat Lists*. Each use case has a set of *Roles* that describe the participants in the use case. We can stop a threat by removing the initial vulnerability or by controlling its propagation (by removing other vulnerabilities) through the use of a *Security Pattern*. The security pattern to use can be selected from the countermeasures defined in the *Misuse Pattern* which describes the threat. Threats that lead to misuses are the goals of the attacker and are performed through low-level threats in the Threat List or directly through a use case operation. Use cases include the roles that participate in the use case. Some threats can happen in all service levels. For example, buffer overflow is a language problem and allows escalation of privilege by the attacker operating at any level. Other threats are specific to the level; for example, a financial application can be attacked by taking advantage of lack of proper authentication in remote access to accounts. If the threat takes advantage of a flaw in an application, it may compromise the security of that application. If the threat affects the IaaS level it affects all the cloud computations, and if it happens at the PaaS level it can affect all the applications developed or deployed in the cloud.

VII. Related Work

While there is a good amount of work on ecosystems, e.g. [10], we have found only a few examples of cloud ecosystems. NIST described an ecosystem for its Cloud Reference Architecture [28] and later an ecosystem for its Security Reference Architecture [19]. However, they included only a Broker (not a CASB) to let users access multiple clouds, an Auditor to check compliance with regulations, and a Communications Provider; that is, theirs is a rather meager set of external functions. They describe their models with words and block diagrams, which we consider not precise enough to guide developers and researchers. The Open Group has a web site with their cloud model [30]. This includes a UML model for the main blocks and a table describing the components involved. There are no UML models for the components and they consider the same main components of the NIST model. A blog presents some ideas about models for cloud ecosystems [13]; however, the models are loose and shown as block diagrams. OSGi also has some general ideas about ecosystems [32] but nothing specific about clouds. There is also a considerable amount of work on multiclouds, some of which has common aspects with ours, e.g. [6] and [8], but they emphasize intercloud operations, don't use patterns or consider ecosystem aspects. Ecosystems can also be seen as systems of systems and work on that topic may apply to them.

VIII. Conclusions

Ecosystems have been around for a while, probably starting with the applications that enhanced Windows, and has become important with the thousands of applications for smart phones. Clouds require a variety of related components to be effective and cloud ecosystems are starting to be defined. Some are implicit ecosystems like the combination of clouds and wireless devices. In most wireless cloud papers the wireless devices are treated as different from the clouds and requiring special models and interfaces to interact with them. We believe that a holistic, unified treatment is fundamental to handle the complexity of cloud-based systems. This is especially true for handling security and privacy, a unified approach reduces complexity, one of the most important weaknesses used by attackers and can enable analysis of the propagation of threats and data leaks. The existing models oversimplify this complexity and are not appropriate to study security and reliability aspects, so ours appears to be the first attempt in this direction. The models are already usable in specific domains, e.g. telecommunications, although more details are needed. As part of this work we identified several possible patterns, which define future work. Future work also includes showing how security and privacy constraints propagate across components.

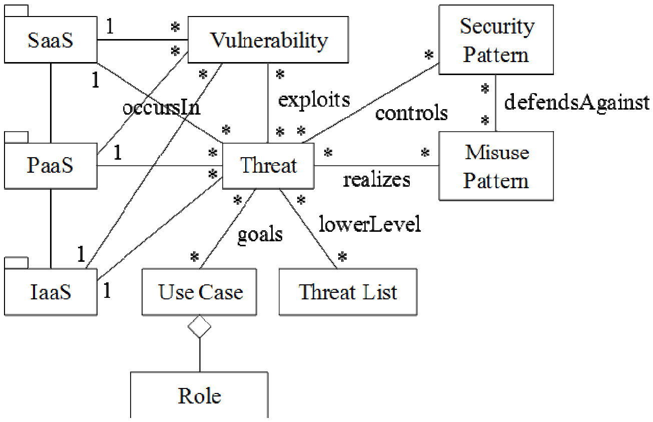

Fig. 7. Metamodel for security concepts

Acknowledgments

We thank the referees for their useful comments that significantly improved this paper. The first author also thanks the National Institute of Informatics of Japan for financing a trip to Tokyo to discuss this work.

References

[1] Adallom, "The case for a cloud access security broker", https://www.adallom.com/wp-content/uploads/2014/12/TheCaseForACASB.pdf

[2] Alcatel-Lucent, http://www.alcatel-lucent.com/solutions/nfv

[3] P. Avgeriou, "Describing, Instantiating and Evaluating a Reference Architecture: A Case Study," *Enterprise Architect Journal*, Fawcette Technical Publications, Jun. 2003.

[4] H. Basilier, M. Darula, and J. Wilke, "Virtualization network services—the telecom cloud", Ericsson Review, March 28, 2014, 2-9.

[5] K. Beckers, I. Coté, S. Fassbender, M. Heisel, and S. Hofbauer, "A pattern-based method for establishing a cloud-specific information security management system", *Requirements Eng*. vol. 18, 2013, 343-395.

[6] David Bernstein and Deepak Vij. Intercloud security considerations. In 2010 IEEE Second International Conference o Cloud Computing Technology and Science (CloudCom 2010), n, pages 537–544. 2010.

[7] Bitglass, "The definitive guide to cloud access security brokers", http://pages.bitglass.com/Cloud-Access-Security-Brokers_PDF.html?aliId=3891489

[8] Jens-Matthias Bohli, Nils Gruschka, Meiko Jensen, Luigi Lo Iacono, and Ninja Marnau. 2013. Security and Privacy-Enhancing Multicloud Architectures. IEEE Trans. Dependable Secur. Comput. 10, 4 (July 2013), 212-224. DOI=10.1109/TDSC.2013.6 http://dx.doi.org/10.1109/TDSC.2013.6

[9] F. Bonomi, R. Milito, J. Zhu.\, S. Addepalli, "Fog computing and its role in the Internet of Things", MCC'12, ACM, Aug. 17, 2012, Helsinki, Finland.

[10] J. Bosch, "From software product lines to software ecosystems", Procs. 13thInt. Software Product Line Conf. (SPLC'09), August 2009, 111-119.

[11] A. Brown, B. Apple, J.B. Michael, and M. Schumann, "Atomic-level security for web applications in a cloud environment", *Computer,* Dec. 2012, IEEE, 80-83.

[12] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerland, and M. Stal, *Pattern-Oriented Software Architecture Volume 1: A System of Patterns*, Volume 1. Wiley, 1996.

[13] D. Chou, "Rise of the cloud ecosystems", http://blogs.msdn.com/b/dachou/archive/2011/03/16/rise-of-the-cloud-ecosystems.aspx

[14] CipherCloud, http://www.ciphercloud.com/2014/09/30/public-cloud-security-demands-cloud-access-security-broker-casb/

[15] Cisco Corp., "Cisco cloud strategy for cloud providers", 2014.

[16] Cisco, Network Functions Virtualization, http://www.cisco.com/c/en/us/solutions/service-provider/network-functions-virtualization-nfv/index.html

[17] Elastica, http://cdn2.hubspot.net/hub/349272/file-1350266572-pdf/2014-07-eBook_Promo/Elastica-Whitepaper-RethinkingSecurityForSaaSCloudapps.pdf

[18] E. B. Fernandez, N. Yoshioka, H. Washizaki, and M. VanHilst, "An approach to model-based development of secure and reliable systems", *Sixth International Conference on Availability, Reliability and Security (ARES 2011)*, August 22-26, Vienna, Austria.

[19] E.B.Fernandez, "Security patterns in practice: Building secure architectures using software patterns", Wiley Series on Software Design Patterns, 2013.

[20] E.B.Fernandez, N.Yoshioka, H.Washizaki, and J.Yoder, "Abstract security patterns for requirements specification and analysis of secure systems", Procs. of the WER 2014 conference, a track of the 17th Ibero-American Conf. on Soft. Eng.(CIbSE 2014), Pucon, Chile, April 2014

[21] E.B.Fernandez and Sergio Mujica "Two patterns for HIPAA regulations", *Procs. of AsianPLoP (Pattern Languages of Programs) 2014*, Tokyo, Japan, March 2014.

[22] E.B.Fernandez, Raul Monge, and Keiko Hashizume, "Building a security reference architecture for cloud systems", *Requirements Engineering,* 2015. DOI: 10.1007/s00766-014-0218-7

[23] E.B. Fernandez, Nobukazu Yoshioka, Hironori Washizaki, "Cloud Access Security Broker (CASB): A pattern for accessing secure cloud services", *Procs. of 4th AsianPLoP (Pattern Languages of Programs) 2015*, Tokyo, Japan, March 2015.

[24] E.B.Fernandez and B. Hamid, "A pattern for Networks Func tions Virtualization", *EuroPLoP 2015*.

[25] M. Fowler, *Analysis patterns – Reusable object models*, Addison-Wesley, 1997.

[26] Keiko Hashizume, E.B.Fernandez, and Maria M. Larrondo-Petrie, "A pattern for Software-as-a-Service in Clouds", RISE'12,Workshop on Redefining and Integrating Security Engineering, part of the *ASE Int. Conf. on Cyber Security*, Washington, DC, December 12-14, 2012.

[27] B. Jellema and M. Vorwerk, "Communications as a cloud ser vice: a new take on Telecoms", *Ericsson Rev*., July 22, 2014, 2-9.

[28] NIST, Cloud Computing Reference Architecture." 2011: http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505

[29] NIST, Cloud Computing Security Reference Architecture." , NIST Special Publication 500-299, http://bigdatawg.nist.gov/_uploadfiles/M0007_v1_3376532289.pdf

[30] The Open Group Cloud Ecosystem Reference Model, http://www.opengroup.org/cloud/cloud/cloud_ecosystem_rm/model.htm

[31] OpenStack, https://www.openstack.org/

[32] OSG, http://www.osgi.org/wiki/uploads/CommunityEvent2012/OSGI%20and%20Cloud%20Computing-%20David%20Bosschaert.pdf

[33] I. Stojmenovic and S. Wen, "The Fog Computing paradigm: Scenarios and security issues", *Procs. of the 2014 Fed. Conf. on Comp. Sci. and Info. Systs, (*ACSIS), 1-8.

[34] M. VanHilst, E.B.Fernandez, and F. Braz, "A multidimensional classification for users of security patterns", *Journal of Research and Practice in Information Technology,* vol. 41, No 2, May 2009, 87-97

[35] D. Yimam and E.B. Fernandez, án approach to build reference architectures", submitted for publication.