Cloud Ecosystems support for Internet of Things and DevOps using Patterns

Madiha H. Syed
Dept. of Comp. and Elect.Eng.and Computer Science
Florida Atlantic University,
Boca Raton, FL, USA
msyed2014@fau.edu

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

***Abstract*— An ecosystem is the expansion of a software product line architecture to include systems outside the product which interact with the product. A cloud ecosystem includes interdependent and communicating components. Not only the cloud ecosystem itself keeps evolving, but it also affects the way in which we develop and deploy software. We model the architecture of a cloud ecosystem as a set of patterns, showing partial descriptions for some of them. We then discuss the role of this evolving platform in facilitating the Internet of Things and the new DevOps framework for developing software.**

Keywords-Software ecosystems; Internet of Things; cloud computing; fog computing; software containers; DevOps; reference architectures.

I. Introduction

Cloud computing has become a popular execution platform by offering demand-based computing service. It has proved its worth for large and small businesses alike. The cloud has brought about a global shift in how computing systems are used. The cloud computing paradigm itself is also evolving. New services and components are being added at a fast pace.

A cloud is not a single system, it comprises a multitude of systems, components, services, and applications, which allow it to provide valuable benefits. A cloud has a number of interdependent and associated systems, which are rapidly evolving. In addition to their standard uses, these systems are providing support for new frameworks for application development and deployment schemes.

An ecosystem is the expansion of a software product line architecture to include systems outside the product which interact with the product [1]. Developing ecosystems for complex systems helps service providers as well as consumers. Cloud computing is an excellent example of such complex interconnected systems. A cloud, its associated systems, providers, consumers, brokers, software, and infrastructure are all related and make up the cloud ecosystem.

The complexity of the cloud ecosystem is also increasing as new functions or technologies become available. Growth in dimension and diversity of this ecosystem is contributing to the evolution of intelligent and interactive environments like IoT. All this can be of tremendous value but we need to properly handle this complexity in order to better utilize the full potential of the system.

Pattern-based architectural models have been found useful in representing such complex systems. Architectural models provide holistic and unified views of the system. This can be useful for understanding the system as well as effectively securing it. A cloud ecosystem has been described in the form of a pattern diagram in [2]; that paper also described a few components in the form of patterns. We will now add to the cloud ecosystem newly identified components, described as patterns. We will discuss two new types of systems which are driving the changes in cloud ecosystems like IoT. We also discuss how this evolution is changing the way we handle software development and deployment. This work is just a step in the architectural representation of the cloud ecosystem. It cannot be considered as complete or final as it is bound to change with the growth of the ecosystem. Our objective now is not to implement such a system but to describe the architecture of existing systems.

This paper is organized as: Section II presents background; Section III shows a pattern diagram for the cloud ecosystem. Section IV presents two new patterns (Containers and Fog Computing). We discuss the IoT in Section V. Section VI describes DevOps and how new components and cloud orchestration are influencing the development of new software frameworks. Section VII lists related work; it is followed by conclusions in Section VIII.

II. Background

Architectural models using patterns can be used to describe ecosystems and their components. A pattern is a solution to a recurrent 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 [3]. They encapsulate the experience and knowledge of designers, provide a larger unit of reuse, and a communication vocabulary for designers.

Usually, a template with predefined sections is used to describe patterns. This systematic approach facilitates the work of both writers and users of the pattern. We use the POSA (Pattern-oriented software architecture) template [3]. Pattern descriptions can include formal languages in addition to modelling techniques such as UML diagrams.

In addition to patterns we will also use Reference Architectures (RAs) to describe the cloud ecosystem. An RA is a generic and abstract software architecture that applies to a particular domain and does not contain implementation details. It specifies the components of the system, their individual functionalities and their mutual interaction [5]. An RA can be considered as a compound pattern and its components described as patterns.

III. A Cloud Ecosystem

Figure 1 shows a pattern diagram for a cloud ecosystem, previously described in [2]. We have added new components

978-1-4673-9948-7/16 $31.00 © 2016 IEEE
DOI 10.1109/IoTDI.2015.31

in color to reflect changes in the ecosystem. The core pattern of this ecosystem is the Cloud Reference Architecture (Cloud RA) [6]. Addition of security patterns to a Cloud RA converts it into a Cloud Security RA (Cloud SRA). The Cloud SRA includes security patterns for Authentication, Authorization, and Logging, among others, which can help control known threats to the cloud [4]. Patterns that describe how regulations apply to the Cloud RA are included in the Cloud Compliant RA [7].

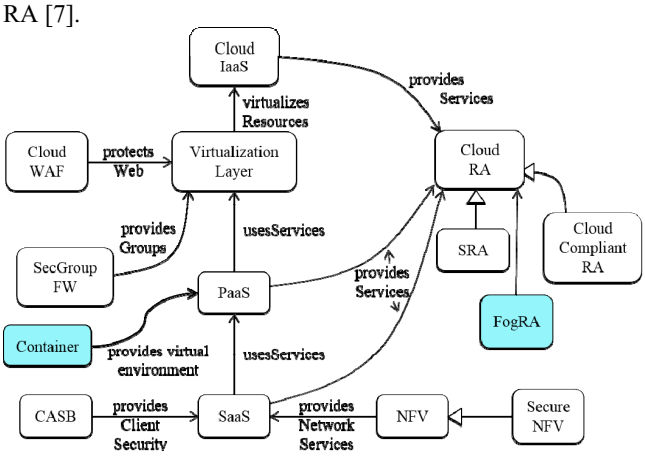

Figure 1. A cloud ecosystem

Other components of the ecosystem, shown in Figure 1, include the service layers of a cloud, IaaS, PaaS and SaaS [4]. Network Functions Virtualization (NFV) is a network architecture where network functions are provisioned in software using virtualization [8].

Other security-related components are also represented as patterns. Filtering functions are provided by Cloud Web Application Firewalls (Cloud WAF) and Security Group Firewalls (SecGroup FW) [9]. 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 [10].

New entities that have been added to this ecosystem include containers and fog computing. A *Container* provides an execution environment for applications sharing a host operating system (OS), binaries, and libraries with other containers with strong isolation between them [11]. Fog Computing is an intermediate platform that provides computation, storage, and networking services between end devices and the cloud [12]. It has emerged as an extension of the cloud for supporting IoT [13].

IV. Models of The Ecosystem Components

A pattern diagram for cloud ecosystem gives a holistic and unified view and the next step is to define detailed models for each of its components. Some of these components have already been described as patterns, here we include patterns for two components which were not present in our earlier work [2]. Complete patterns can be found in their respective references, just main functions of the components are included here. Pattern descriptions also include Forces, Consequences, Implementation, and Related Patterns sections. The idea here is that we can build patterns for every participant in the ecosystem, which provides a unified view of the complete system.

4.1 Software Container

4.1.1. Intent

A Software Container provides an execution environment for applications sharing a host OS, binaries, and libraries with other containers with strong isolation between them. Containers are lightweight, portable, extensible, reliable, and secure.

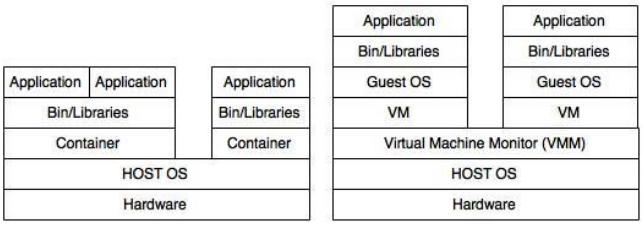

Figure 2. "Two containers sharing one OS" vs "A Virtual Machine" (above)

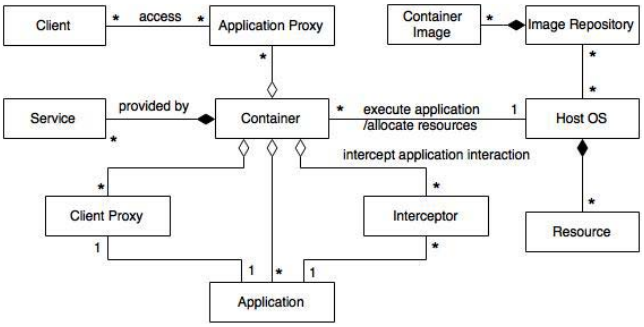

Figure 3. Class diagram of the Container pattern

4.1.2 Solution

A Software Container provides a runtime environment that can support the isolated execution of applications on a shared Host OS. Figure 2 shows a comparison between Containers and Virtual Machines. Figure 3 shows the class diagram for this pattern. A *Container* controls a set of *Applications* sharing a *Host OS* that provides a set of *Resources*. An *Interceptor* mediates the services provided to the application by the container. Applications hosted in containers can be accessed remotely through *Proxies*, where the Container acts as a broker. The client interacts with the *Application Proxy*, which represents the application. The application interacts with the *Client Proxy*, which represents the client. The Container provides *Services* to the applications. Containers facilitate application distribution, especially for DevOps teams [15]. Docker [32] is a popular example of software containers.

4.2 Fog Computing

4.2.1 Intent

Fog Computing is a virtualized platform that stands between cloud computing systems and Internet devices, providing to these computation, storage, and networking services and allowing a cloud to control and communicate with these devices. Fog can offer low latency, location awareness, efficient use of bandwidth and storage services.

4.2.2 Solution

Introduce a platform to provide cloud computing-like services closer to the devices to be monitored or controlled. This is called Fog Computing. It provides computation,

storage, and networking services between end user devices and cloud providers. Data is processed locally for better response. Aggregated data and other relevant information can be forwarded to the cloud for analysis.

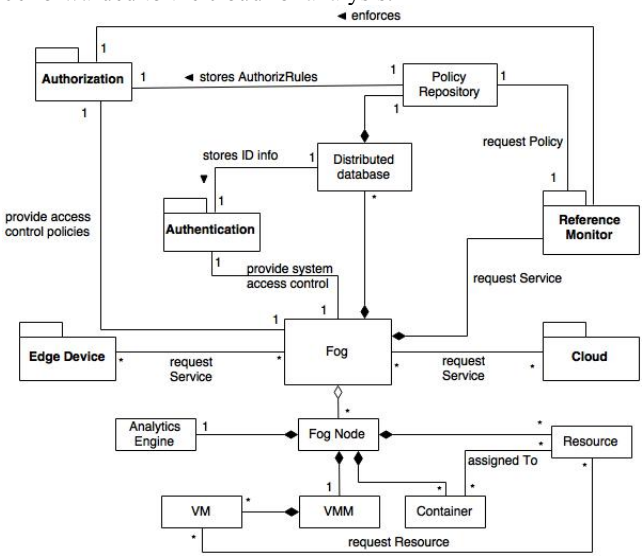

Figure 4. Class diagram of the Fog Computing pattern

Figure 4 shows the class diagram for this pattern. The *Fog* is a collection of several distributed tiny clouds called *Fog Nodes*. They can be resource-rich servers, routers, access points, mobile devices, etc. Edge devices are the devices located at the edge of the networks which need to be monitored or controlled. A Fog Node has *resources* including hardware (compute, networking and storage) capabilities. These nodes provide local real-time analytics using an *Analytics Engine*. Applications can be hosted in the fog nodes using virtualization, Virtual Machine Monitor (*VMM*), Virtual Machines (*VM*) and/or *Containers*. A *distributed database* stores both application data and necessary metadata for service orchestration. It also has information about capabilities of nodes, information about the state of fog nodes and services, policies for security, filtering, and configuration. A *Reference monitor* is used to perform policy-based access enforcement. New policies can be added to the *Policy Repository.* Data is transferred between fog nodes and the various components of the Fog. The Fog also provides *Authentication* and *Authorization* services. In addition, services like filtering, data aggregation, logging, etc., can be provided. Cisco provides fog computing platforms [16].

V. Internet of Things

Cloud computing has established itself as a promising computing paradigm. However, the recent popularity of IoT has made the limitations of clouds more apparent. IoT applications require low latency, mobility support, location awareness and support for geo-distribution [13], this can be provided by fog computing. It is important to realize that fog computing complements rather than replaces the cloud.

The IoT has brought about an explosive proliferation of endpoints. According to an estimate, nearly 50 billion heterogeneous devices will be connected to the internet by 2020 [18]. Storage and processing costs are declining and the devices are getting smaller and less expensive.

There is also an increase in the intelligence of these devices, which are producing huge amounts of data. 90% of the world's data was created in the last 2 years, and it is increasing exponentially. We need to manage the large number of devices and process the data produced by them. Fog computing offers one of such solutions. Cisco has fog computing solutions. Amazon Web Services (AWS) just launched its Amazon IoT cloud service for IoT [19]. IBM also has IoT solutions for analytics [20].

The IoT has found its application in various domains. A comprehensive survey of enabling technologies, protocols, and architecture for smart cities is provided in [21]. Examples include smart traffic control, smart grid, wireless sensor networks, precision agriculture, intelligent buildings, health care, industrial automation, oil and gas, etc. Numerous papers discuss these IoT applications in connection to fog computing [13]. Ours is the first pattern for fog computing [19].

Containers are also providing more lightweight, portable virtualization solutions that will offer support for IoT applications. Containers are being used for rapid development, testing, deploying, and updating IoT applications [22].

VI. Dev-Ops

The cloud has had an impact on all facets of the computing world including software development. DevOps is a conceptual framework which can be considered as a type of Agile Software Development. DevOps stands for "development" and "operations" and it calls for increased communication, collaboration and smooth integration between developers and operations teams. It enables continuous development and frequent releases of software to the end user [23], which can lead to earlier problem detection and solution delivery.

DevOps is driven by the wide availability of virtualized and cloud infrastructure. Cloud-Computing-enabled orchestration of services enables development and operation teams to use code for automatically managing infrastructure. Automating the process of software delivery and infrastructure changes enables frequent iterations. Many major cloud service providers, like IBM and Amazon, are providing DevOps solutions as part of their platforms [24][25].

Software containers are part of cloud ecosystems and are supporting DevOps teams for the isolation of services. They make application distribution much easier by providing lightweight, isolated execution environments for applications. A few patterns have been identified and described for software orchestration on clouds [15].

Emerging technologies like Cloud Computing, IoT and DevOps are mutually influencing each other. DevOps is being enabled by the cloud. IoT applications require the integration of development, IT operations and quality assurance which can be achieved by practicing DevOps. However, DevOps teams need to address concerns like auditability, traceability, compatibility, interoperability, testing and quality assurance, not to mention security [26].

IV. VII. Related Work

Ecosystems have been around for quite some time now but few examples of cloud ecosystems can be found. NIST has

defined cloud ecosystems in their Cloud RA [27], and Security RA [28]. These RAs include a set of external functions but they do not include new components for virtualization (like containers) or platforms (like fog computing). In addition, the models are described using block diagrams and text, rather imprecise vehicles. The Open Group has its own Cloud Ecosystem Reference Model [29]. It includes a UML model for the main blocks and the involved components are described in the form of a table. It does not mention any components that can support IoT or virtualization components. Block-diagram-based models for cloud ecosystems are also discussed in [30]. OSGi discusses ecosystems [32] but nothing specifically for clouds. Some common aspects to our ecosystem can be found in the work on multiclouds [31]; however, this work is focused on intercloud operations, and does not use patterns.

VIII. Conclusions

The cloud has long reached that stage where a large number of interconnected components exist while new ones are being added. We need to provide a holistic and unified view of the system to its users, developers, and researchers. Cloud ecosystems are still new and we are beginning to define them. This holistic, unified treatment is fundamental to handle the complexity of cloud-based systems. Such an ecosystem can help us control heterogeneity, provide a holistic security view, as well as take care of quality and compliance issues. Most of the work related to cloud ecosystems presents very simplified models. New computing platforms (like fog computing), development practices and software frameworks (such as DevOps) are already popular. These advances are complementing and influencing each other as technologies mature. Detailed cloud ecosystems will help both IoT system developers and DevOps practicing teams. The ecosystem presented in this work should be considered open-ended. The model presented here is just a step for achieving precise architectural representation of the cloud ecosystem. Many components still have to be defined as patterns, including security and threat patterns. In addition, new components will continue be added as cloud computing evolves. In our future work we are considering the mappings of the authorization models from clouds to fogs to devices.

References

[1] J. Bosch, “From software product lines to software ecosystems”, Procs. 13th Int. Software Product Line Conf. (SPLC’09), pp. 111-119.

[2] E.B. Fernandez, N. Yoshioka and H. Washizaki, “Patterns for Security and Privacy in Cloud Ecosystems”, 2nd Int. Workshop on Evolving Sec. and Privacy Req. Eng.(ESPRE’15), Jan 2015.

[3] 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.

[4] E.B. Fernandez, Security patterns in practice: Building secure architectures using software patterns, Wiley, May 2013.

[5] P. Avgeriou, “Describing, Instantiating and Evaluating a Reference Architecture: A Case Study,” *J. Enterprise Architect*, Fawcette Tech. Publications, Jun. 2003.

[6] E.B. Fernandez, R. Monge, and K. Hashizume, “Building a security reference architecture for cloud systems”, J. Req. Eng., 2015. pp. 1-25. doi:10.1007/s00766-014-0218-7.

[7] D.Yimam and E.B.Fernandez, “Building Compliance and Security Reference Architectures for Cloud Systems”, in press.

[8] E.B.Fernandez and B. Hamid, “A pattern for Networks Functions Virtualization”, EuroPLoP 2015.

[9] E.B.Fernandez, N.Yoshioka and H.Washizaki, “Patterns for cloud firewalls”, AsianPLoP 2014.

[10] E.B. Fernandez, N. Yoshioka and H. Washizaki, “Cloud Access Security Broker (CASB): A pattern for accessing secure cloud services”, AsianPLoP 2015.

[11] M. H. Syed and E. B. Fernandez, “The Software Container pattern”, PLoP 2015, Pittsburgh, PA,, Oct. 24-26, 2015.

[12] M. H. Syed, E. B. Fernandez and M. Ilyas, “A pattern for Fog Computing”, in press.

[13] F. Bonomi, R. Milito, P. Natarajan and J. Zhu, “Fog Computing: A Platform for Internet of Things and Analytics”, Big Data and Internet of Things: A Roadmap for Smart Environments, Springer, Studies in Computational Intelligence , vol. 546 , pp.169 -186 , 2014.

[14] I. Stojmenovic and S. Wen, “The Fog Computing paradigm: Scenarios and security issues”, Procs. Fed. Conf. on Comp. Sci. and Info. Systs, (ACSIS’14), pp. 1-8. 2014.

[15] T.Sousa, F.Correia, H.Ferreira,"DevOps patterns for software orchestration on public and private clouds“, PLoP 2015.

[16] Cisco, 2015. [Online]. Available: http://www.cisco.com/web/solutions/trends/iot/cisco-fog-computing-with-iox.pdf. [Accessed: 31- Jun- 2015].

[17] D.F.Willis, A.Dasgupta, S.Banerjee, “Paradrop: a multitenant platform for dynamically installed third party services on home gateways”. ACM SIGCOMM workshop Dist.Cloud Comp.2014.

[18] Cisco, "Internet of Things (IoT)", 2015. [Online]. Available: http://www.cisco.com/web/solutions/trends/iot/iot-products.html. [Accessed: 31- Jun- 2015].

[19] Amazon Web Services, "AWS IoT", 2015. [Online]. Available: https://aws.amazon.com/iot/. [Accessed: 02- Nov- 2015].

[20] IBM IoT Solutions, 2015. [Online]. Available: http://www.ibm.com/analytics/us/en/internet-of-things/. [Accessed: 01- Nov- 2015].

[21] A.Zanella, N.Bui, A.Castellani, L.Vangelista, M.Zorzi, "Internet of Things for Smart Cities," IEEE J. Internet of Things,1,1,pp.22-32, 2014

[22] IBM DeveloperWorks, “Rapidly develop Internet of Things apps with Docker Containers”, 2015 [Online]. Available: https://www.ibm.com/developerworks/library/iot-docker-containers/[Accessed:05-Nov-2015].

[23] S. Sharma and B. Coyne, “DevOps for Dummies”, IBM DevOps fundamentals, John Wiley & Sons, 2015

[24] IBM DevOps, 2015. [Online]. Available: http://www.ibm.com/ibm/devops/us/en/. [Accessed: 01- Nov- 2015].

[25] AWS, 2015. [Online]. Available: https://aws.amazon.com/campaigns/emea-devops/ . [Accessed: 06- Nov- 2015].

[26] NIST Cloud RA, 2015. [Online]. Available: http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505[Accessed: 05- Nov- 2015].

[27] NIST Cloud SRA, 2015. [Online]. Available: http://bigdatawg.nist.gov/_uploadfiles/M0007_v1_3376532289.pdf [Accessed: 05- Nov- 2015].

[28] The Open Group Cloud Ecosystem Ref. Model, 2015. [Online]. Available:http://www.opengroup.org/cloud/cloud/cloud_ecosystem_rm/model.htm. [Accessed: 05- Nov- 2015].

[29] D. Chou, “Rise of the cloud ecosystems”, 2015. [Online]. Available: http://blogs.msdn.com/b/dachou/archive/2011/03/16/rise-of-the-cloud-ecosystems.aspx. [Accessed: 15- Oct- 2015].

[30] J.M. Bohli, N. Gruschka, M. Jensen, L. L. Iacono, and N. Marnau.“Security and Privacy-Enhancing Multicloud Architectures”, IEEE Trans. Dependable Secur. Comput. 10, 4 (July 2013), 212-224. http://dx.doi.org/10.1109/TDSC.2013.6

[31] OSGi, 2015. [Online]. Available: https://en.wikipedia.org/wiki/OSGi. [Accessed: 15- Oct- 2015].

[32] Docker, 2015. [Online]. Available: http://www.docker.com/. [Accessed: 05-May-2015].