

# Cloud of Things

Marcus Vinícius Paulino Gomes

Instituto Superior Técnico

**Abstract.** Abstract should have 200 words at maximum.

## 1 Introduction

## 2 Objectives

### 3 Related Work

Cloud and Internet of Things are emerging computing paradigms that features distinctively different computing resources and system architecture. As its popularity has been growing across the academics and the industry, researchers and developers are spending a lot of effort to investigate how to integrate these technologies in order to take advantage of the benefits provided by both of them.

IoT applications often encapsulate several relatively complex protocols and involves different software components. Moreover, they require a significant investment in infrastructure, besides that the system administrators and users spend time with large client and server installations, setups, or software updates. As most of the computing resources are allocated on the Internet on servers in Cloud computing, integrating these paradigms in a Cloud-based model results in a solution with more flexibility of implementation, high scalability and high availability, and with a reduced upfront investment.

In RFID-based IoT applications, Guinard et al. [3] point out that the deployment of RFID applications are cost-intensive mostly because they involve the deployment of often rather large and heterogeneous distributed systems. As a consequence, these systems are often only suitable for big corporations and large implementations and do not fit the limited resources of small to mid-size businesses and small scale applications both in terms of required skill-set and costs. To address this problem, Guinard et al. proposes a Cloud-based solution that integrates virtualization technologies and the architecture of the Web and its services. They applied the Utility Computing blueprint to the software stack required by RFID-based IoT applications using the AWS platform and the EC2 service. The Elastic Cloud Computing (EC2) service allows the creation and management of virtual machines (Amazon Machine Images, or AMIs) that can then be deployed on demand onto a pool of machines hosted, managed and configured by Amazon. A benefit of this approach is that the server-side hardware maintenance is delegated to the cloud provider which is often more cost-efficient for smaller businesses. Furthermore, it also offers better scaling capabilities as the company using the EPC Cloud AMI, can deploy additional and more powerful instances regarding to the amount of requests.

Distefano [2] et al. proposed a high-level modular architecture to implement the Cloud of Things. According to Distefano et al. things not only can be discovered and aggregated, but also provided as a service, dynamically, applying the Cloud provisioning model to satisfy the agreed user requirements and therefore establishing Things as a Service providers. The *Things as a Service* (TaaS) paradigm envisages new scenarios and innovative, pervasive, value-added applications, disclosing the Cloud of Things world to customers and providers as well, thus enabling an open marketplace of "things". To address this issues, an ad-hoc infrastructure is required to deal with the management of sensing an actuation, mashed up resources provided by heterogeneous Clouds, and things, by exploiting well know ontologies and semantic approaches shared by and adopted by

users, customers and providers to detect, identify, map and transform mashed up resources. The proposed architecture provides blocks to deal with all the related issues, while aiming to provide things according to a service oriented paradigm.

CloudThings [5] is an architecture that uses a common approach to integrate Internet of Things and Cloud Computing. The proposed architecture is an online platform which accommodates IaaS, Paas, and SaaS and allows system integrators and solution providers to leverage a complete IoT application infrastructure for developing, operating and composing IoT applications and services. The applications consists of three major modules, the CloudThings service platform, that is a set of Cloud services (IaaS), allowing users to run any applications on Cloud hardware. This service platform dramatically simplifies the application development, eliminates need for infrastructure development, shortens time to market, and reduces management and maintenance costs. The CloudThings Developer Suite is a set of Cloud service tools (PaaS) for application development, such as Web service API's, which provide complete development and deployment capabilities to developers. The CloudThings Operating Portal is a set of Cloud services (SaaS) that support deployment and handle or support specialized processing services.

The effort put in the research to integrate the paradigms of Cloud Computing and Internet of Things resulted in a essential contribution, but there are several issues regarding to the integration between Cloud Computing and Internet of Things that must be addressed. In particular, due of the heterogeneity of the IoT applications environments, its hard for solution providers to efficiently deploy and configure applications for a large number of users. Thus, automation for the management tasks required by IoT applications is a key issue to be explored.

TOSCA (Topology and Orchestration Specification for Cloud Applications) [4] is proposed in order to improve the reusability of service management processes and automate IoT application deployment in heterogeneous environments. TOSCA is a new cloud standard to formally describe the internal topology of application components and the deployment process of IoT applications. The structure and management of IT services is specified by a meta-model, which consists of a *Topology Template*, that is responsible to describe the structure of a service, then there are the *Artifacts*, that describes the files, scripts and software components necessary to be deployed in order to run the application, and finally the *Plans*, that defined the management process of creating, deploying and terminating a service. The correct topology and management procedure can be inferred by a TOSCA environment just by interpreting te topology template, this is know as "declarative" approach. Plans realize an "imperative" approach that explicitly specifies how each management process should be done. The topology templates, plans and artifacts of an application are packaged in a Cloud Service Archive (.csar file) and deployed in a TOSCA environment, which is able to inter-

pret the models and perform specified management operation. These .csar files are portable across different cloud providers, which is a great benefit in terms of deployment flexibility. As a newly established standard to counter growing complexity and isolation in cloud applications environments, TOSCA is gaining momentum in industrial adoption as well academic interests.

Breitenbücher et al. [1] proposed to combine the two flavors of management supported by TOSCA, *declarative processing* and *imperative processing*, in order to create a standards-based approach to generate provisioning plans based on TOSCA topology models. The combination of both flavors would enable applications developers to benefit from automatically provisioning logic based on declarative processing and individual customization opportunities provided by adapting imperative plans. These provisioning plans are workflows that can be executed fully automatically and may be customized by application developers after generation. The approach enables to benefit from strengths of both flavors that leads to economical advantages when developing applications with TOSCA.

Recently a growing number of organizations are developing Orchestrators, Design Tools and Cloud Managers based on TOSCA. Juju is an Open Source TOSCA Orchestrator that can deploy workloads across public, private clouds, and directly onto bare metal. HP Cloud Service Automation is cloud management solution that supports declaratives services design that are aligned with TOSCA modeling principles. IBM Cloud Orchestrator provides integrated tooling to create TOSCA applications, deploy them with custom policies, monitoring and scale them in cloud deployments.

## 4 Solution Architecture



## 5 Evaluation

## 6 Conclusion

## References

1. Breitenbücher, U., Binz, T., Képes, K., Kopp, O., Leymann, F., Wettinger, J.: Combining declarative and imperative cloud application provisioning based on toasca. IC2E. IEEE (2014)
2. Distefano, S., Merlino, G., Puliafito, A.: Enabling the cloud of things. In: Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on. pp. 858–863. IEEE (2012)
3. Guinard, D., Floerkemeier, C., Sarma, S.: Cloud computing, rest and mashups to simplify rfid application development and deployment. In: Proceedings of the Second International Workshop on Web of Things. p. 9. ACM (2011)
4. Li, F., Vogler, M., Claeßens, M., Dustdar, S.: Towards automated iot application deployment by a cloud-based approach. In: Service-Oriented Computing and Applications (SOCA), 2013 IEEE 6th International Conference on. pp. 61–68. IEEE (2013)
5. Zhou, J., Leppanen, T., Harjula, E., Ylianttila, M., Ojala, T., Yu, C., Jin, H.: Cloudthings: A common architecture for integrating the internet of things with cloud computing. In: Computer Supported Cooperative Work in Design (CSCWD), 2013 IEEE 17th International Conference on. pp. 651–657. IEEE (2013)