An edge computing architecture in the Internet of Things

Cristian Martín, Manuel Díaz and Bartolomé Rubio
Department of Languages and Computer Science
University of Málaga, Málaga, Spain
{cmf,mdr,tolo}@lcc.uma.es

***Abstract*—In the last few years, the Internet of Things (IoT) has emerged as the new disruptive technology to change the world. Cloud computing has accompanied this field to overcome its processing and storage limitations. However, this evolution has originated a huge increase in IoT devices and data that will create a bottleneck for current networks, in addition to a lack of low latency in cloud communications. Edge computing has been developed to address this challenge, moving the processing to the edge of the network. In this paper, an edge computing architecture is presented to overcome these challenges. The architecture, based on our previous work on the λ-CoAP architecture, covers the whole vision of an edge computing deployment, from IoT devices, to the edge Smart Gateways and up to a cloud infrastructure.**

I. INTRODUCTION

The Internet of Things (IoT) is the result of combining previous technologies such as Wireless Sensor Networks and Radio Frequency Identification (RFID). This has led to an ecosystem that comprises a wide range of embedded technologies and devices, from unpowered RFID tags to autonomous devices. Nowadays, the IoT is paving the way to sense and actuate over the world [1]. This field is continuously expanding to many areas [2], transforming houses into home automation and improving a citizen's experience of life in the city through digital services related to transportation, cleanliness and traffic.

This unprecedented growth has also been accompanied by vendor lock-in and heterogeneity issues that complicate current IoT growth predictions [3]. The Constrained RESTFul Environments (CoRE) working group belonging the Internet Engineering Task Force (IETF) addressed this issue and released the Constrained Application Protocol (CoAP)[4], with the aim of bringing REST web services to resources constrained devices. CoAP is a RESTful web transfer protocol that it has been adapted to resource constrained devices. CoAP reduces the overhead of heavyweight protocols such as HTTP and TCP, applying RESTful web services over UDP. Packet reordering, header characteristics and handshaking of these protocols represent a strong obstacle in devices with presence in constrained networks and limited capabilities. CoAP provides a client/server model with HTTP-style interactions and the support for asynchronous interactions though its observe operation. This is of particular interest for those applications that require a continuous monitoring of IoT devices. The observe operation allows devices to start sleep cycles to reduce their power consumption, only sending data when available. Sensor and actuators are abstracted and can easily be accessed through the resources defined in each CoAP server, enabling a compatibility with the web that can be reached using a simple mapping proxy.

ISORC 2018. DOI: https://doi.org/10.1109/ISORC.2018.00021

The IoT is also intended to generate large amounts of data, which will keep on growing in the coming years [5]. The limitations of its devices make addressing current paradigms such as big data or deep learning very complicated, as they require large capacities in terms of processing and storage. These paradigms are designed to extract knowledge and generate actions over raw IoT data. In the last few years, the integration of the IoT with disruptive technologies such as cloud computing has provided the capabilities needed in the IoT to address these paradigms [1]. Nevertheless, this has resulted in a increase of latency in IoT communications in some situations (e.g., an actuation after cloud computing processing) and a dependency on many factors such as availability, routing and networks, which entail the use of external services.

Certain IoT applications, such as structural health monitoring [6] enable damage to be detected just by monitoring multiple, simple points of infrastructures. Higher-level concepts can be extracted from lower-level ones, which in addition can help define the higher-level concepts. This process is known as deep learning [7]. Deep learning is fed by multiple sources and large amounts of data from the extraction of complex patterns. This paradigm can require high capacity in terms of processing and storage that can be achieved with cloud computing. However an extreme event such as an earthquake or unanticipated blast loading would require timely action whilst reducing the latency to a minimum. A new paradigm known as edge computing [8], is intended to reduce the response time in IoT communications and reduce the upload bandwidth to the clouds, moving the computation to the edge of the networks. Edge computing can be interchangeable with fog computing, but edge computing is more focused on the things side [8]. For instance, the next generation of cars will generate 1GB of data per second and so real-time processing to make the right decisions will be required [8]. Although the next generation of mobile networks, 5G, is intended to increase the bandwidth and considerably reduce the latency, networks in the car use case like many others, e.g. airplanes, would represent a bottleneck for cloud-based computing. The adoption of an edge computing approach is required in these scenarios to optimize the latency of the communications and avoid network saturation. This does not mean that cloud computing will disappear from IoT applications, but it will be used only when it is required. The contribution of this work is to provide a framework for edge computing based on previous work on the IoT and cloud computing in the λ-CoAP architecture [**?**]. This framework will enable the development of IoT applications using both edge and cloud computing. The resulting applications will take advantage of real-time interactions at the edge and long-term analysis on cloud computing.

The rest of the paper is structured as follows. In Section II our work on λ-CoAP is introduced. Section III presents the general scheme of the proposed architecture. Section IV considers the related work. Finally, the conclusions and future work are presented in Section V.

II. λ-CoAP ARCHITECTURE

Although CoAP can abstract the heterogeneous issues and make devices accessible through the Internet, networking, storing and processing are still limited due to the amounts of data generated by the IoT. According to various estimations, IoT data has grown exponentially with respect to the number of connected devices [5]. This huge expansion together with the environments and applications which require processing and storing large amounts of data should be necessarily accompanied by upper abstractions.

As stated, cloud computing has been used to complement and reduce the limitations of the IoT, a paradigm also known as the Cloud of Things. The need to enable a framework for real-time processing arbitrary functions over arbitrary data and protect large amounts of data from human errors, led Nathan Marz to define the Lambda Architecture (LA) [9]. The LA is a cloud computing architecture composed of various cloud computing components which provides the functionality for processing, analyzing, and consuming large amounts of arbitrary data. The key concept of the LA is the precomputed view, which constrains precomputed results that can be used to reduce the processing time. The LA splits the data flow into three layers:

- **real-time layer**: processes the streaming data and generates real-time precomputed views.
- **batch layer**: generates batch precomputed views though the processing of historical data.
- **serving layer**: offers a way to display and access all generated views.

The three aforementioned layers therefore have the specific task of reducing the latency when processing large amounts of immutable data. In deep learning techniques where large data sets of data are used to extract high-level patterns, the LA significantly reduces the processing time.

Our work on the λ-CoAP architecture provides an IoT and cloud computing architecture, adopting CoAP and the LA. On the one hand, the IoT heterogeneity inside the physical infrastructure is abstracted at the same time as lightweight web services are enabled for accessing them through CoAP. Sensors and actuators can also be created at run-time in IoT devices [10] without a reconfiguration and reboot process. On the other hand, the LA enables the analysis and actuation based on the large amounts of data generated by the IoT, thus reducing the processing latency.

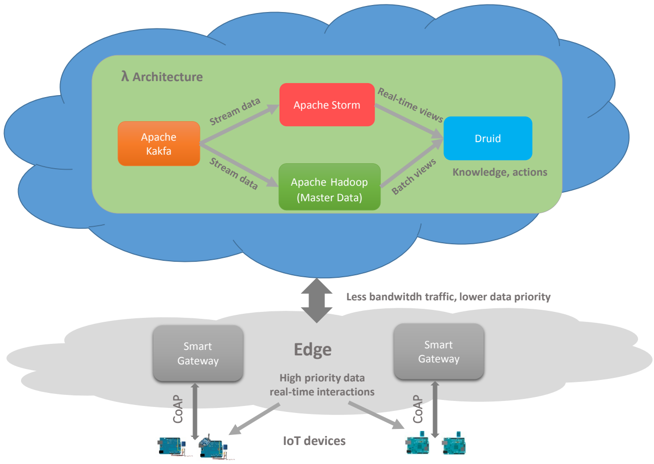

Fig. 1. An overview of the edge computing architecture

The λ-CoAP architecture is complemented with a Smart Gateway to interconnect the LA with the underlying devices and a proxy to enable HTTP interactions. Lastly, a Web user interface (UI) enables the management and visualization of IoT data and devices. Thus far, this architecture has enabled devices to work autonomously, but the whole processing power, which results in actions over the underlying IoT infrastructure and knowledge extraction are provided in the cloud through the LA.

III. AN EDGE COMPUTING ARCHITECTURE

The IoT is generating huge amounts of data that will be difficult to manage by current cloud infrastructures. Edge computing has emerged as a new paradigm to overcome the problems of huge amounts of IoT data, moving the computing to the edge of the network, thus reducing the latency of cloud communications and releasing networks from the bottleneck that would derive from that bandwidth. Nevertheless, cloud infrastructures will not disappear from IoT applications, rather their assets such as high availability and great capacity for processing and storage will complement these applications.

Our work on the λ-CoAP architecture provides an IoT and cloud computing architecture enabling the consumption, processing, and analysis of large amounts of data. Nevertheless, this architecture follows a centralized approach using a cloud infrastructure as a processing engine. As stated, this architecture could create a network bottleneck in some IoT environments, and in addition increase the response latency leading to a limitation when a real-time interaction is required. Therefore this work has been complemented with an edge computing architecture as shown in Fig. 1. The main difference between the λ-CoAP architecture and the edge computing architecture resides in the Smart Gateway, which acquires a greater role.

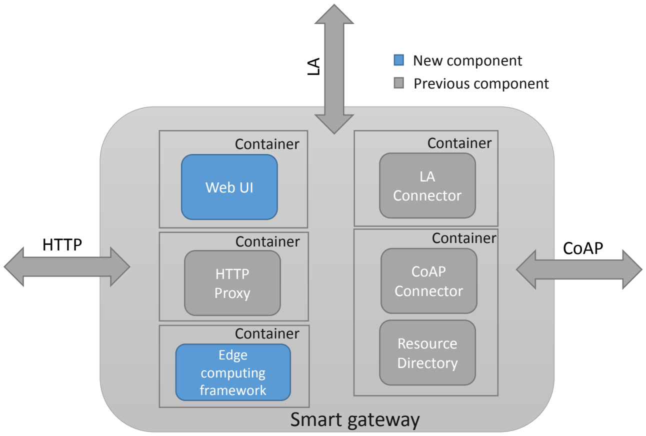

Fig. 2. Smart gateway with lightweight virtualization

Whilst in the λ-CoAP architecture the Smart Gateway acts a bridge to interconnect the cloud platform, the underlying devices and the end users, the Smart Gateway in the edge computing architecture also enables an autonomous device with processing capabilities. With its computation power it can reduce the communication latency and lighten the networks. The edge layer comprises the Smart Gateway deployed in the architecture. The following subsections demonstrate the innovations of this paper on the Smart Gateway's edge nodes.

A. *Lightweight virtualization*

Initially, Smart Gateways in the λ-CoAP architecture had a closed infrastructure composed by a set of components to interconnect the aforementioned participants. Upon a new development or a component modification, it becomes necessary to reconfigure or install these components manually, without component isolation. Lightweight virtualization technologies have garnered a lot of attention in the last few years due to their features: fast processes of building containers, high density of services per container and a high isolation between instances [11]. In contrast to traditional hypervisors, lightweight virtualization technologies implement virtualization of processes through containers in the operating system. This reduces the overhead of the hardware and virtual device virtualization in traditional hypervisors, allowing the deployment of a high density of containers [12]. Therefore, the use of this virtualization technology could help in the management of components in an IoT Gateway.

A lightweight virtualization architecture has been applied to the Smart Gateway. The resulting Smart Gateway with its containers is shown in Fig. 2. Two new components have been added to the Smart Gateway: a Web management UI for managing devices and containers and an edge computing framework to develop the edge logic. This architecture enables the management of processes at run-time and the reuse and backup of containers. Therefore, this virtualization will facilitate the process update, backup and management in the Smart Gateways. In addition, each container is isolated from the rest. Although there are various lightweight virtualization technologies, Docker[1] has been chosen for its compatibility with the target IoT Gateway e.g., Raspberry Pi[2] and the REST API support, which will be integrated into the Web UI. Containers will be fully managed with this API through an integration with the Web UI. Moreover, the API also paves the way to third party integration in order to manage the Smart Gateways.

B. *IoT devices and virtualization management Web UI*

A Web management UI has been incorporated inside the Smart Gateway to manage IoT devices and containers. This enables the management of Smart Gateways without the involvement of a cloud infrastructure. Although, another UI in the cloud is able to coexist with it for global management. Sensors and actuators can be created and managed at run-time in the IoT devices deployed through the Web UI using our previous work [10]. This enables the management of the physical components in IoT devices without necessitating a reconfiguration and stopping their services. In this case, CoAP has been used, but the container-based architecture paves the way towards the integration of new IoT protocols.

A management of containers can also be done in the Smart Gateway through the Docker integration. Containers represent the processes running in the Smart Gateways. The management allows a better control over their current components, facilitating their maintenance and enabling their reuse. As can be seen in Fig. 2, the components are a part of containers and in turn can be managed by this component container. Apart from enabling the management of containers, the Web UI enables the submission of new container images that are automatically installed and available as new containers in the Smart Gateway.

C. *Edge computing framework*

Edge computing is enabled thanks to the inclusion of application logic at the edge of the network, which reduces the communication latency and alleviates the bandwidth with the cloud. This component enables a framework for the application development at the edge, in the Smart Gateway in this case. More specifically, the framework is based on Node-RED[3], a browser-based editor that allows visual data flow programming. A set of components have been created to filter and aggregate IoT data, so that application developers only need focus on the application logic. This component decides what data should be processed immediately or should be sent to the cloud. In the case of an IoT actuation, the communication latency will not depend on the communication with the cloud infrastructure, therefore it should be reduced. Fig. 3 shows an overview of the Node-RED framework. On the left side, application developers can select nodes from the palette to be part of the system, forming data flows. Discovery and IoT interactions are transparently done by the edge computing

---

[1] https://www.docker.com/

[2] https://blog.hypriot.com/

[3] https://nodered.org/

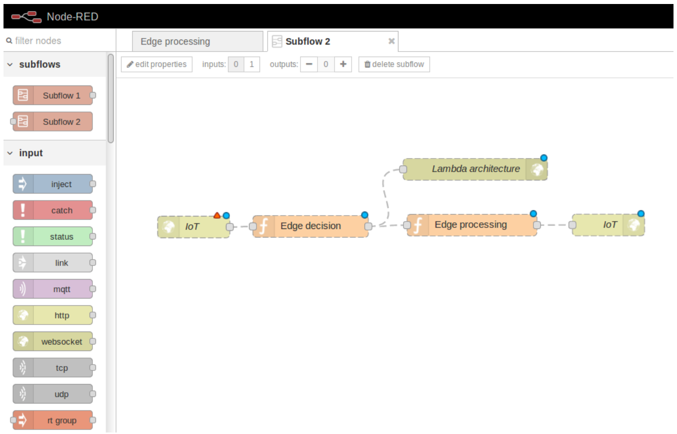

Fig. 3. Node-RED: a visual programming framework

framework, thus application developers just need to focus on the application logic. Moreover, resulting flows can be easily portable between Smart Gateways.

The edge computing framework has a component-based architecture enabling the reuse of components in flows and allowing new components to be created for the application development. The development is done visually with the creation component flows. Visual programming can provide a better user experience, alongside a reduced perceived workload and a higher perceived success.

IV. RELATED WORK

Edge computing can be more efficient than cloud computing for some computing services, especially in the IoT [8]. In [13], the authors propose an edge computing architecture. This architecture only focuses on filtering out noise and performing a little pre-processing to alleviate the bandwidth with the cloud infrastructure. LEGIoT [11] also presents a Smart Gateway with a lightweight virtualization technology. Although this architecture enables an energy-efficient activation of containers, it does not allow a full management of containers and IoT devices like the one presented in this paper. In [14], the authors also propose a Smart Gateway enabling virtualization of sensors and networks, however it does not take into account process management. The architecture presented in this paper covers a broad vision of an edge computing deployment and improves the management of Smart Gateways.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an edge computing architecture for the IoT. This architecture covers the whole scope of an edge computing deployment. This paper has focused on the design of an edge Smart Gateway. A lightweight virtualization technology has been adopted to give a better control and isolation of the running processes. The edge logic can be defined through a visual framework using data flows. This framework incorporates the most common operators to filter and aggregate data, in addition to discovering the underlying IoT infrastructure, so that applications only need focus on the application logic. Lastly, a Web UI enables the management of the virtualized instances and IoT devices. IoT applications such as structural health monitoring could leverage this architecture to reduce their latency and alleviate the bandwidth to the cloud. In the near future, we will finish the development of the Smart Gateway and will evaluate the whole architecture. Moreover, a programming abstraction for cloud and edge computing is planned as a long-term development.

ACKNOWLEDGMENT

This work is funded by the Spanish projects TIC-1572 (MIsTIca: Critical Infrastructures Monitoring based on Wireless Technologies) and TIN2014-52034-R (An MDE Framework for the Design and Integration of Critical Infrastructure Management Systems).

REFERENCES

[1] M. Díaz, C. Martín, and B. Rubio, "State-of-the-art, challenges, and open issues in the integration of internet of things and cloud computing," *Journal of Network and Computer Applications*, vol. 67, pp. 99–117, 2016.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *Communications Surveys & Tutorials, IEEE*, vol. 17, no. 4, pp. 2347–2376, 2015.

[3] J. Rivera and R. van der Meulen, "Gartner says the internet of things installed base will grow to 26 billion units by 2020," *Stamford, conn., December*, vol. 12, 2013.

[4] K. H. Shelby, Zach and C. Bormann, "The constrained application protocol (coap)," Available online: https://tools.ietf.org/html/rfc7252/, (accessed on 13 May 2016).

[5] A. B. Zaslavsky, C. Perera, and D. Georgakopoulos, "Sensing as a service and big data," in *In International Conference on Advances in Cloud Computing (ACC-2012), Bangalore, India*, July 2012, pp. 21–29.

[6] D. Balageas, C.-P. Fritzen, and A. Güemes, *Structural health monitoring*. John Wiley & Sons, 2010, vol. 90.

[7] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[8] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[9] N. Marz and J. Warren, *Big Data: Principles and best practices of scalable realtime data systems*. Manning Publications Co., 2015.

[10] C. Martín, M. Díaz, and B. Rubio, "Run-time deployment and management of coap resources for the internet of things," *International Journal of Distributed Sensor Networks*, vol. 13, no. 3, 2017.

[11] R. Morabito, R. Petrolo, V. Loscri, and N. Mitton, "Legiot: a lightweight edge gateway for the internet of things," *Future Generation Computer Systems*, vol. 81, pp. 1–15, 2018.

[12] R. Morabito, J. Kjällman, and M. Komu, "Hypervisors vs. lightweight virtualization: a performance comparison," in *Cloud Engineering (IC2E), 2015 IEEE International Conference on, Tempe, AZ, USA*. IEEE, 2015, pp. 386–393.

[13] Y. Huang, X. Ma, X. Fan, J. Liu, and W. Gong, "When deep learning meets edge computing," in *Network Protocols (ICNP), 2017 IEEE 25th International Conference on, Toronto, ON, Canada*. IEEE, 2017, pp. 1–2.

[14] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on, Barcelona, Spain*. IEEE, 2014, pp. 464–470.