On the way to a configurable testbed to support IoT research

Giuseppe Tricomi*†, Zakaria Benomar*†, Francesco Longo*†, Giovanni Merlino*†, Antonio Puliafito*†
*Dipartimento di Ingegneria, Università di Messina, Italy
Email: {gtricomi,zbenomar,flongo,gmerlino,apuliafito}@unime.it
† CINI: National Interuniversity Consortium for Informatics, Rome, Italy

***Abstract*—The advancement in Edge computing and the fast adoption of Internet of Things (IoT) devices is pushing the research in different fields involving popular topics such as Cyber-Physical Systems, Smart Environments, Digital transformation, Industry 4.0, and much more. Nevertheless, with the great attention of the scientific community and government investments on these topics, at least one question is continuously tackled by researchers: how to realize realistic testbeds able to verify the solutions under analysis without assigning significant shares of their research funds? Nowadays, the scientific communities are working on designing and deploying distributed research infrastructures to support the researchers in their tests and evaluations. In this paper, we present our idea of a platform to provide a distributed infrastructure to support the researcher's work on IoT. For this purpose, containerization, Network, I/O virtualization, and Sensing and Actuation-as-a-Service (SAaaS) are techniques provided by the I/Ocloud paradigm, a solution exploiting the opensource IoT middleware Stack4Things.**

***Index Terms*—Edge Computing, IoT, I/Ocloud, SAaaS, Virtualization, Containerization**

I. Introduction

The last decade has been characterized by great researchers' attention to computing paradigms pushed by the Cloud Computing advent. Furthermore, the improvement of the microchip production processes makes devices more robust, and with the advent of IoT devices, paradigms such as Fog, Edge, and Continuum Computing have become central and relevant in computing research. Edge computing, among the others, represents the cornerstone element of several hot research fields, such as Cyber-Physical Systems, Intelligent Transportation Systems, and Continuous Machine Learning. One of the most significant issues researchers experience in the Edge computing field is setting up of a realistic testbed to test and verify the effectiveness of their studies. The testbed preparation is an activity that requires an appropriate amount of both the researcher's time and capital, reducing the overall productivity. Furthermore, the testbeds are designed to focus on the research (or projects) needs. Even if they can be adapted for other research, the adaptation works require even more energy, time, and money. In the last years, some initiatives promoted by the government are meant to offer exploitable testbeds for the researchers' needs, to name a few: FIWARE [1], Chameleon [2], Fed4FIRE+ [3], IoT Lab [4], F-Interop [5], and SLICES project [6]. These initiatives aim to provide open, accessible, and reliable facilities supporting a wide range of research activities. Fed4FIRE+ offers a federation of Next Generation Internet (NGI) testbeds. Fed4FIRE+ was empowered by the combination of infrastructures specialized in the Internet of Things (IoT) provided by the IoT Lab, and their facilities are exploited by F-Interop, an H2020 European research project integrating and extending several European testbed federations to research and develop online testing tools for the Internet of Things. The latter, the SLICES project, has, in particular, the objective to build the first-ever ESFRI RI to support research in Digital Sciences. Furthermore, SLICES' ambition is to provide a European-wide test platform, providing advanced compute, storage, and network components, interconnected by dedicated high-speed links. This platform will be the preferred collaborative instrument for researchers at the European level to explore and push further the envelope of the future Internet.

At the same time, commercial products are available with their services, such as Amazon AWS[1] and Google[2], to mention the two most prominent players in the market. All the previously cited initiatives are focused on the provisioning of Cloud-based facilities that can satisfy only partially the main requests of an Edge-computing researcher, and much more an Internet of Things (IoT) researcher: testing applications along the Edge and IoT (composed mainly by constrained devices) distributed on an area and, in some cases, interacting with physical devices, directly or through mediation. To clarify this concept, the results obtained by tests realized on ad-hoc or cloud-based testbeds always present limitations due to their intrinsic characteristics. In the former case, by the absence of concurrent access to the testbed resources (multi-tenancy management), and in the latter, the need to emulate the devices or at least the devices connected to them.

Designing a platform built to support IoT-based systems for experimental research is not easy. It has to offer some fundamentals characteristics that make the infrastructure: i) dynamically configurable, ii) programmable, iii) able to run in parallel multiple assignments and tests by preserving the isolation of the execution environments running on it, and iv) lastly but not less relevant, the infrastructure has to provide an environment that is geographically dispersed as it could be the actual application environment (i.e., Smart City and Cyber-

[1]https://aws.amazon.com/?nc2=h_lg
[2]https://cloud.google.com/

ISBN 978-3-903176-48-5

Physical Systems, or Intelligent Transportation Systems).

The first two characteristics are two sides of the same coin. They grant the adaptability of the infrastructure used by a researcher from the infrastructure management point of view (dynamic configuration, e.g., network configuration or features offered) and from the computation tasks performed on the Edge and IoT devices (programmability, e.g., a Machine Learning model or simple function). The third characteristic is necessary because the hypothesis of exclusively using the resources is negligible in a real-world context and has to be included in a research results evaluation. By design, a platform can run multiple tasks with the warranty of isolation of the execution environment (i.e., through containerization). Finally, another characteristic is the geographical distribution of testbed composing elements set up by a researcher. Indeed, an application or a solution working on the IoT is designed to run on a geographical area (wide or not); therefore, issues such as networking latencies or network instability must be considered during testing and evaluation.

This paper presents an overview of our platform, aiming to support the research on IoT. The overview analyzes its main functionalities, belonging to Stack4Things and I/Ocloud solutions. A dissertation about some use-cases applied on the platform is included in the end. We consider this work valuable to support initiatives aiming to create infrastructure to support researchers' works, such as SLICES [6], Fed4FIRE+ [3], IoT Lab [4], and F-Interop [5]. The remainder of the paper is structured as follows: Section II presents the scenario in which this work is conceived and introduces the two solutions based on this work. Sections III and IV present the main aspects that have to be managed by the investigated scenario. Section V presents two use cases in which the utilization of a platform as the one discussed can simplify the testbed setup, increasing our productivity. Consideration and next steps are discussed in Section VI.

II. Background

Cloud Computing is instrumental in expanding the benefits of computing, storage, and networking capabilities to applications. The National Institute of Standards and Technology (NIST) definition of Cloud computing [7] describes a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services). According to the NIST definition, these resources can be rapidly provisioned and released with minimal management effort and without the service provider's involvement. Thereby, the Cloud paradigm produces a two-fold benefit: the users can be free to manage their resources, and on the other side, providers can avoid wasting resources (i.e., the computational power of servers idle or partially used). Suppose the previous sentences describe benefits for a resource provider. In that case, they are still valid for an industry or company that can increase its income by exploiting Cloud computing. The advantages offered by the Cloud have pushed the research to produce advancements in: i) management (e.g., service models: IaaS, PaaS, and SaaS [7]), ii) resource usage accounting [8], iii) resource optimization provisioning [9], [10], and iv) resource aggregation patterns (e.g., Federation or Multi-Cloud approaches [11], to mention a few. In this wave, several products are proposed as Cloud management systems, for business or research purposes, freeware or not, such as AWS Cloud[1], Google Cloud computing[2], Microsoft Azure[3], and OpenStack [12]. The latter, in particular, is a set of open-source software tools for building and managing Cloud computing platforms. OpenStack is a Cloud solution for most commercial, in-house, and hybrid deployments and a fully open-source ecosystem of tools and frameworks allowing the management of virtualized computing/storage resources respecting the Cloud paradigm principles.

The widespread adoption of Internet of Things (IoT) devices feeds the research on Cloud computing techniques extending the scope of resource management towards the network edges. Indeed, the IoT devices work at the edge of network and provide the ubiquity of devices with sensing and actuating capabilities that act as programmable gateways to the physical world. In general, most approaches to fully exploit the IoT ecosystem rely primarily on adopting the Cloud paradigm to provide data-centric solutions such as [13], [14], where the only operations permitted are data manipulation ones. The enhancement of the previous approach offers complete control over an IoT infrastructure and enables the to reprogram of reprogramming it; thereby, the users may opt for vertical solutions to deploy and manage their infrastructure. Commercial solutions are available on the market (e.g., AWS Greengrass[4] or IBM Watson IoT[5]) alongside open-source and research products such as OpenIoT [15] or Stack4Things (S4T) [16].

Nevertheless, a similar solution does not enable the application developers to share an IoT infrastructure. Thus each user has to set up his/her own infrastructure, which is a limitation for adopting IoT applications on a larger scale as the capital expenditure of IoT infrastructure is often non-trivial. Besides, authorizations to deploy IoT nodes in public domains for large-scale deployments can be hard to acquire. In addition to the limitation of sharing the IoT infrastructure, data-centric-oriented solutions are based on sending all generated data toward a datacenter (or a Cloud provider). For instance, an IoT sensing deployment with sensors producing data at a high data rate from a large number of sensors can incur significant operational expenditure in terms of bandwidth [17], storage, and processing cost. In such scenarios, it might be helpful to process the generated data at the edge and transmit only pre-processed information to the Cloud, thus avoiding high bandwidth and storage use. On the other hand, data processing at the network edge is also helpful in satisfying the requirements of typical time-sensitive applications that cannot tolerate delays introduced when relaying on a faraway Cloud.

[3]ttps://azure.microsoft.com/en-us/

[4]https://aws.amazon.com/greengrass/

[5]https://www.ibm.com/cloud/internet-of-things

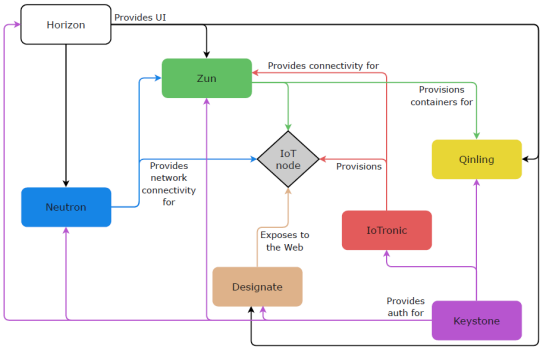

Fig. 1. Stack4Things subsystems.

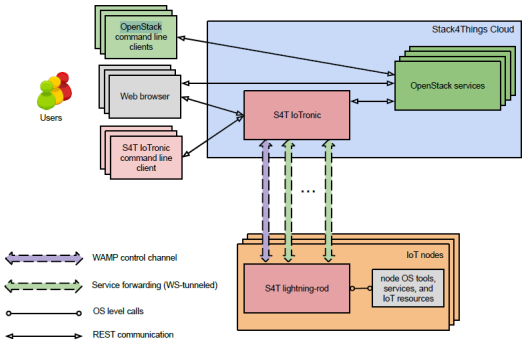

Fig. 2. Stack4Things architecture overview.

A. *Stack4Things*

Stack4Things aims to tackle the issues introduced by the data-centric solution to manage IoTs. It is a middleware open-source belonging to the OpenStack ecosystem to support the management of IoT (or Edge device) deployments; S4T tries to implement suitable capabilities for IoT infrastructure to join an edge-extended IaaS Cloud. Fig. 1 shows the OpenStack subsystems involved in the S4T IoT management, introducing IoTronic, a subsystem meant for the provisioning and configuration of IoT nodes with hosted sensing and actuation resources. Concerning the rest of the OpenStack subsystem used within S4T, the networking service, Neutron, has been enhanced to provide network connectivity for IoT nodes deployed at the network edge. Furthermore, to expose the edge-based IoT nodes resources as Web resources, we used the OpenStack Designate subsystem to associate publicly resolvable domain names with the distributed physical/virtual IoT nodes even when deployed within IPv4 masquerade networks.

Stack4Things middleware has an architecture composed mainly of two parts: a Cloud-side component (IoTronic) and one (or multiple) edge-side component(s) (Lightning-Rod), as shown in Fig. 2.

IoTronic is modeled following the standard design of OpenStack services to ensure its full compatibility with other OpenStack subsystems (e.g., Keystone, Neutron, Designate, Qinling, etc.). By design, the Edge nodes to be managed via S4T are considered (embedded) smart devices capable

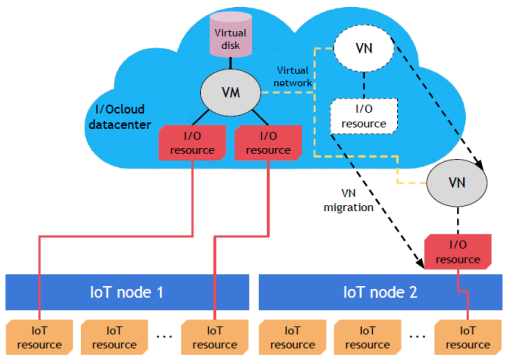

Fig. 3. I/Ocloud operational view

of hosting a minimal Linux distro (e.g., OpenWRT) such as SBCs (e.g., Arduino, Raspberry Pi, and Arancino[6]) that are microprocessor (MPU)-powered. For this reason, Lightning-Rod (LR) is designed as a lean agent modular and fault-tolerant adapt to this category of devices. LR is a key component in the S4T design that links the Edge devices, even when they are deployed behind NATs or strict firewalls, to the S4T IoTronic service. The ability to bypass the networking middleboxes is provided by the adoption of WebSocket (realized with the Web Application Messaging Protocol, WAMP) to set up the full-duplex messaging channel between the Cloud and the devices suitable to route traffic streams (e.g., commands). The communication channels so realized fit perfectly Edge and IoT devices constraints by exploiting two facilities, namely, publish/subscribe (pub/sub) messaging and Remote Procedure Calls (RPCs).

B. *I/Ocloud*

The I/Ocloud approach [18], through the exploitation of the S4T functionalities, aims to offer standardized and generic programming capabilities on top of IoT resources regardless of the underlying infrastructure configurations. In addition, the approach keeps the ability to use the unique characteristics of an IoT-enhanced distributed datacenter, such as the availability of nodes at the edge, which may then be used as computing infrastructure to deal with data (pre)processing. Thereby, this approach aims to achieve a seamless integration between the Cloud and IoT by providing the distributed IoT resources (i.e., sensors and actuators) hosted on nodes deployed at the network edge as virtualized Cloud resources. This integration represents a critical feature of the I/Ocloud: it has to ensure that IoT deployments are engaged as active elements of the Cloud infrastructure while preserving their characteristics. To summarize, it must ensure a well I/O virtualization (virtIO).

As shown in Fig.3, I/Ocloud extends the virtualization concept to the IoT world by abstracting IoT resources and providing them as virtual ones accessible via a developer-friendly interface for an I/O primitive of its physical counterpart. The abstraction mechanism is programmable: it can concern the

[6]https://smartme.io/projects/arancino-cc/

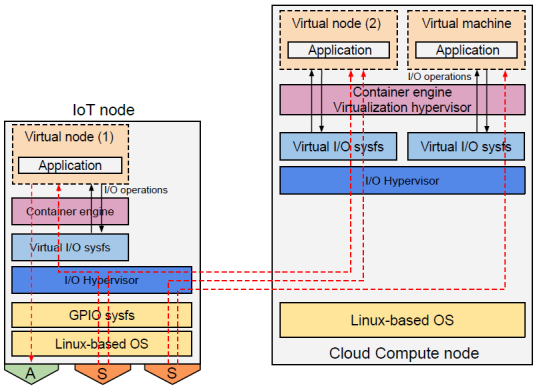

Fig. 4. I/Ocloud virtualization approach.

entire I/O resources of an IoT node or just a subset of the resources, and it can also regroup, logically, IoT resources from different IoT nodes within the same (logical) entity.

The I/O Virtualization, shown in Fig. 4, is based on filesystem virtualization[7] to provide a virtual representation of pins of a physical IoT node while being able to host user-defined logic and providing, at the same time, interactions with the remote physical IoT resources. Technically, an I/Ocloud instance is a self-contained and isolated environment with a user space-defined file system, *sysfs* realized with the exploitation of *FUSE* [19] technology over Remote Procedure Calls (RPCs) to ensure remote interactions with the physical IoT resources.

III. Virtualization and Isolation Facilities

I/Ocloud solution, thanks to the I/O virtualization facilities, can offer users one or more Virtual Node (VN) instances. A VN may be instantiated as an isolated and portable environment (e.g., a lightweight container) either on the Cloud datacenter or a remote Edge (or IoT) node, as it is shown in Fig. 4. Thereby, some benefits are so obtained:

- application logic requesting great computational resources (or relying on data-centric approaches) running on a VM (or on a lightweight container) on the Cloud can accede to multiple resources and data exploiting I/O virtualization;
- application logic that suffers latency issues can be distributed and run over the Edge nodes;
- it is possible to realize and test business logic, both centralized and decentralized, as desired;
- it is possible to migrate or replicate VNs from Cloud to Edge and vice-versa.

I/Ocloud exploits the OS-level virtualization [20], more commonly known as containerization, as a solution for I/O virtualization and VN management. Indeed, this virtualization approach, rather than dedicating a whole OS for each guest VM, enables multiple isolated user-space environments (namely the container) to run on a single host machine while sharing a unique kernel provided by the host. A container has its own dedicated resources, such as file systems, TCP/IP stacks, to name a few. Due to the constrained nature of Edge and IoT devices and a limited demand of computing resources (as well as in terms of storage for the images [21]) the containers are the best solution for the VN virtualization, especially if advanced functionalities (e.g., container migration) may be put in place [22]. Nevertheless, the containerization enables the VN isolation; the I/O Hypervisor grants access to the VNs requesting to interact with the physical resources, as shown in Fig. 4.

[7]Linux-based IoT boards leverage a GPIO pseudo-filesystem to interact with physical pins

IV. Networking Facilities

The virtualization of the IoT nodes and their physical resources is not enough to realize a fully configurable testbed on the Edge and IoT devices; to reach the goal, I/Ocloud necessitates a mechanism to provide network virtualization as well. Networking facilities and network virtualization are critical aspects in such a context where each element may be hidden by networking barriers, as it is the Edge or an IoT deployment. The I/Ocloud view aims to enable users to instantiate personalized networking topologies among any combination of VMs and VNs spanning the datacenter, and Wide Area Networks (WANs) when VNs are deployed at the network edge (see yellow dashed lines in Fig. 3). Networking management is implemented by integrating Neutron facilities and IoTronic-to-Lightning-Rod connectivity. The latter consists of channels from the Cloud to Edge devices by exploiting S4T-bound mechanisms: Websockets and reverse tunneling. Thereby, the I/Ocloud can exploit the Neutron mechanisms to set up virtual LANs by leveraging the Neutron abstractions of *network*, *subnet*, and *port*. In particular, the *network* abstraction is used to isolate the operative context to provide the multi-tenancy management of infrastructure. Furthermore, in our approach, the *binding* hosts (where Neutron ports get created) are deployed inside the same machines hosting the WS tunnel agents (see a red dashed rectangle in Fig. 5). This choice was made on purpose, as the WS agents can set up WS tunnels in order to attach Neutron ports instantiated on those machines to the IoT nodes deployed at the network edge using WebSocket. In simple terms, ports are created/managed on the Cloud and then attached to the IoT nodes deployed at the network edge using WebSocket in a decoupled two-step pattern.

As shown in Fig. 6, which represents the S4T node-side architecture, attaching an IoT node to a virtual user-defined network is relatively straightforward. In particular, a Virtual Interface (VIF) gets instantiated on the IoT node and then attached (using a reverse WebSocket tunnel) to the OpenStack networking platform managed by Neutron on the Cloud.

V. Use Cases

In this section, we want to present some examples of research in which exploiting a platform to create a testbed could have speeded up our work. One of the most suitable use cases to demonstrate the validity of the platform proposed is the Smart City scenario, where a realistic city-scale

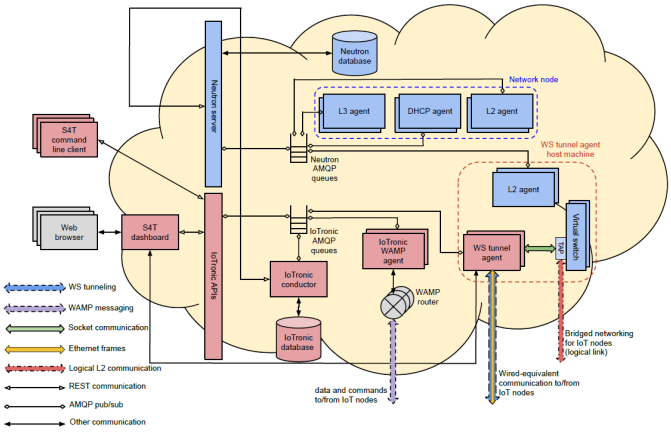

Fig. 5. Integration between S4T and Neutron facilities Cloud-side.

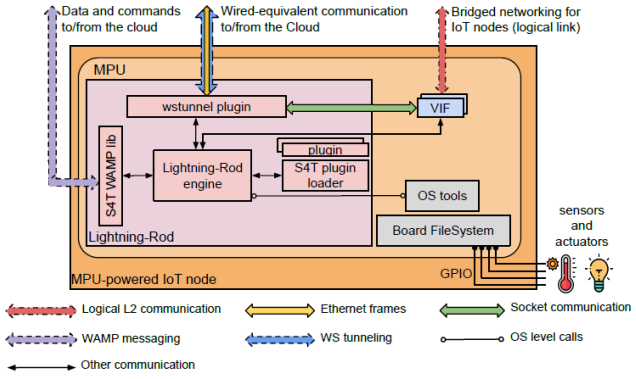

Fig. 6. Integration between S4T and Neutron facilities IoT-side.

testbed is not easy to replicate, if not under some a priori assumptions. A Smart City is an ecosystem of infrastructure and services aiming to bring together society, government, and technology to produce enhanced services (smart mobility, smart environment, etc.). This holistic view calls for an all-encompassing approach to embracing technologies and services, thus providing a broader (or even a global) solution to (smart) city problems. In this light, there is the need for a scalable architecture aiming at reusing, multiplexing, and sharing technologies and services on the urban scale. The goal is to establish a homogeneous ecosystem where multiple applications can scale out to a metropolitan scope, thus underpinning an open and shared Information and Communication Technologies (ICTs) infrastructure made of sensing, actuation, network, processing, and storage resources. As done in the TOO(L)SMART [23] project, five Italian cities have shared an ecosystem-oriented template based on sensing and actuation devices. It can be extended and configured via the control logic running on Cloud. Specifically, on the one hand, the goal is to provide a uniform representation of connected smart objects by abstracting, grouping, and managing them as a unified ecosystem of smart objects to be configured, customized, and contextualized according to the high-level application requirements. On the other hand, a management layer, able to

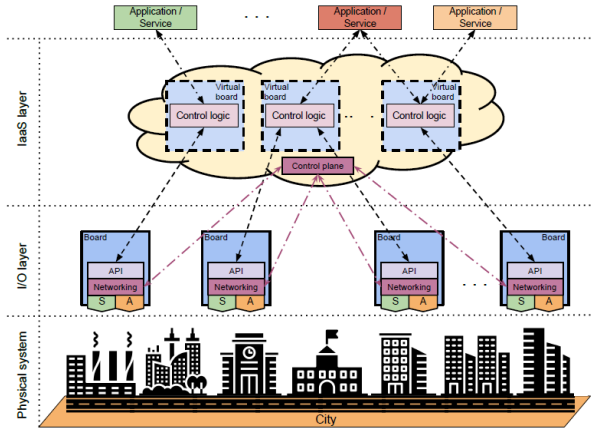

Fig. 7. Software-Defined City.

control the ecosystem dynamics, map such requirements into lower-level ones, and implement and enforce specific policies to satisfies such requirements, is needed. A suitable solution may therefore lie in adopting a software-defined approach, where basic mechanisms provided by the smart cities objects at data plane are used by the control plane to implement policies related to application/end user-level requirements. Thus, we talk about Software Defined Cities (SDCs) [24].

The testbed for this scenario made with the platform in analysis can be realized with data plane level elements running on IoT devices. One or multiple VN are run in the form of containers that can interact with sensors and actuators. Control layer elements instead can execute on clouds in the form of virtual boards (see Fig. 7).

In the same scenario, the research about realizing an infrastructure dynamically configurable upon a Software-Defined City is more tricky than the previous. Fig. 8 depicts the architectural schema in which multiple domains (three environments divided by vertical dashed lines) cooperate through federation. The domain infrastructure elements are separated into three layers (e.g., physical, data, and control layer) with different computation resources. All the domains are federated, and it is not distributed. According to the platform principle, the testbed may be created easily concerning how it was made in our research that has requested the creation of multiple Openstack clouds separately and virtualizing the several elements in the form of VMS. Indeed exploiting the I/Ocloud and Neutron virtual networking, it was possible to create three networks among a group of devices that are separated. The Control Layer services can be exposed by Stack4Things (or with Designate, according to how testing is planned). The inter-domain tunnel may be tested in a realistic scenario considering the pool of IoT devices geographically distributed.

VI. Conclusions and Future Work

This work has presented the main concepts and mechanisms enabling the realization of a platform able to realize on-demand IoT or (Edge computing) testbeds as a tool supporting the researcher during their study. Concerning the other initiatives aiming to provide testbeds and facilities to the researcher,

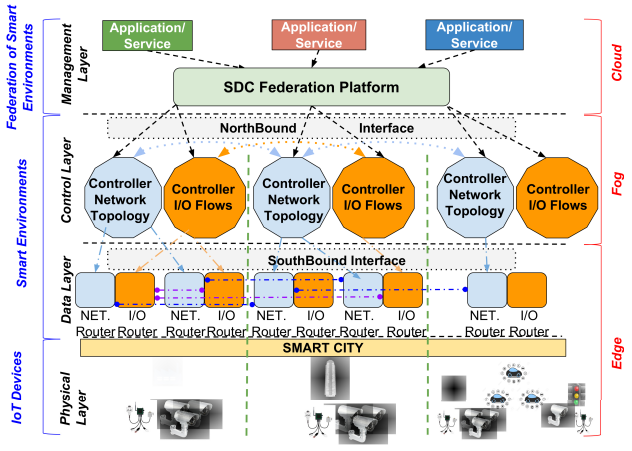

Fig. 8. High Level architecture of Software Defined City. [25]

the presented framework aims to propose not only a way to interact with the IoT testbeds federated spread over a wide area (such as it happens in IoT Lab). Indeed, the presented framework offers programmability of the devices that can be helpful to customize their behavior, abstract functionalities, and decoupling the application logic by the device management logic. All of the above is preliminary work regarding the platform, depicting the main functionalities needed by a platform for experimental studies already available on the solution presented here. Other features that may be valuable for a platform like this (e.g., Function-as-a-Service). However the next step should indeed be the definition of a platform that can provide Testbed-as-a-Service (Taas) for IoT researchers who need an instrument to explore various designs.

Acknowledgment

This work was partially supported by the ESFRI Project SLICES: Scientific Large-scale Infrastructure for Computing/Communication Experimental Studies, and specifically the SLICES-PP project, under grant n. 101079774.

References

[1] F. Cirillo, G. Solmaz, E. L. Berz, M. Bauer, B. Cheng, and E. Kovacs, "A standard-based open source iot platform: Fiware," *IEEE Internet of Things Magazine*, vol. 2, no. 3, pp. 12–18, 2019.

[2] J. Mambretti, J. Chen, and F. Yeh, "Next generation clouds, the chameleon cloud testbed, and software defined networking (sdn)," in *2015 International Conference on Cloud Computing Research and Innovation (ICCCRI)*. IEEE, 2015, pp. 73–79.

[3] FED4FIRE+. [Online]. Available: https://www.fed4fire.eu/

[4] IoTLab. [Online]. Available: https://www.fed4fire.eu/news/new-fed4fire-testbed-iot-lab/

[5] F-INTEROP. [Online]. Available: https://www.f-interop.eu/

[6] SLICES. [Online]. Available: https://slices-ri.eu/

[7] P. Mell and T. Grance, "The nist definition of cloud computing," National Institute of Standards and Technology (NIST), Gaithersburg, MD, Tech. Rep. 800-145, September 2011. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[8] S. Ibrahim, B. He, and H. Jin, "Towards pay-as-you-consume cloud computing," in *2011 IEEE International Conference on Services Computing*. IEEE, 2011, pp. 370–377.

[9] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE transactions on services Computing*, vol. 5, no. 2, pp. 164–177, 2011.

[10] S. Singh and I. Chana, "Cloud resource provisioning: survey, status and future research directions," *Knowledge and Information Systems*, vol. 49, no. 3, pp. 1005–1069, 2016.

[11] G. Tricomi, G. Merlino, A. Panarello, and A. Puliafito, "Optimal selection techniques for cloud service providers," *IEEE Access*, vol. 8, pp. 203 591–203 618, 2020.

[12] OpenStack. [Online]. Available: https://docs.openstack.org/

[13] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by internet of things," *Transactions on emerging telecommunications technologies*, vol. 25, no. 1, pp. 81–93, 2014.

[14] X. Sheng, J. Tang, X. Xiao, and G. Xue, "Sensing as a service: Challenges, solutions and future directions," *IEEE Sensors journal*, vol. 13, no. 10, pp. 3733–3741, 2013.

[15] J. Soldatos, N. Kefalakis, M. Hauswirth, M. Serrano, J.-P. Calbimonte, M. Riahi, K. Aberer, P. P. Jayaraman, A. Zaslavsky, I. P. Žarko *et al.*, "Openiot: Open source internet-of-things in the cloud," in *Interoperability and open-source solutions for the internet of things*. Springer, 2015, pp. 13–25.

[16] G. Merlino, R. Dautov, S. Distefano, and D. Bruneo, "Enabling workload engineering in edge, fog, and cloud computing through openstack-based middleware," *ACM Transactions on Internet Technology (TOIT)*, vol. 19, no. 2, pp. 1–22, 2019.

[17] M. U. Ilyas, M. Ahmad, and S. Saleem, "Internet-of-things-infrastructure-as-a-service: The democratization of access to public internet-of-things infrastructure," *International Journal of Communication Systems*, vol. 33, no. 16, p. e4562, 2020.

[18] D. Bruneo, S. Distefano, F. Longo, G. Merlino, and A. Puliafito, "I/Ocloud: Adding an IoT dimension to cloud infrastructures," *Computer*, vol. 51, no. 1, pp. 57–65, 2018.

[19] B. K. R. Vangoor, V. Tarasov, and E. Zadok, "To {FUSE} or not to {FUSE}: Performance of {User-Space} file systems," in *15th USENIX Conference on File and Storage Technologies (FAST 17)*, 2017, pp. 59–72.

[20] S. Soltesz, H. Pötzl, M. E. Fiuczynski, A. Bavier, and L. Peterson, "Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors," in *Proceedings of the 2Nd ACM SIGOPS/EuroSys european conference on computer systems 2007*, 2007, pp. 275–287.

[21] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. Ott, "Consolidate iot edge computing with lightweight virtualization," *IEEE network*, vol. 32, no. 1, pp. 102–111, 2018.

[22] L. Ma, S. Yi, and Q. Li, "Efficient service handoff across edge servers via docker container migration," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, 2017, pp. 1–13.

[23] *Official site Toolsmart Project*, https://www.torinocitylab.it/it/toolsmart.

[24] G. Merlino, D. Bruneo, F. Longo, A. Puliafito, and S. Distefano, "Software defined cities: A novel paradigm for smart cities through iot clouds," in *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*. IEEE, 2015, pp. 909–916.

[25] G. Tricomi, G. Merlino, F. Longo, S. Distefano, and A.Puliafito, "Software-defined city infrastructure: a control plane for rewireable smart cities," in *2019 IEEE 5th Intl Conf on Smart Computing and Its Associated Workshops (SMARTCOMP)*.