



A Service Bus Architecture for Highly Distributed and Dynamic IoT Environments

Marco Garofalo, Zakaria Benomar, Francesco Longo, Giovanni Merlino,
Nikolaos Georgantas, Antonio Puliafito

► To cite this version:

Marco Garofalo, Zakaria Benomar, Francesco Longo, Giovanni Merlino, Nikolaos Georgantas, et al..
A Service Bus Architecture for Highly Distributed and Dynamic IoT Environments. 9th Italian
Conference on ICT for Smart Cities and Communities, Sep 2023, Salerno, Italy. hal-04405204

HAL Id: hal-04405204

<https://hal.science/hal-04405204>

Submitted on 19 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Service Bus Architecture for Highly Distributed and Dynamic IoT Environments

Marco Garofalo^{*}, Zakaria Benomar[†], Francesco Longo^{*‡}
Giovanni Merlino^{*‡}, Nikolaos Georgantas[†], Antonio Puliafito^{*‡}

^{*}Department of Engineering, University of Messina, Italy

Emails: marco.garofalo@studenti.unime.it {flongo,gmerlino,apulafito}@unime.it

[†] National Institute for Research in Digital Science and Technology (INRIA), Paris, France

Emails: {zakaria.benomar, nikolaos.georgantas}@inria.fr

[‡] CINI: National Interuniversity Consortium for Informatics, Rome, Italy

Abstract—The Internet of Things (IoT) impacts numerous application domains nowadays, including smart cities, smart factories, and intelligent transportation, to name a few. However, leveraging the IoT within software applications raises tremendous challenges from the networking up to the application layer due to the ultra-large-scale, extreme heterogeneity, and dynamics of IoT. This paper explores how the service-oriented architecture paradigm may be revisited to address communication challenges posed by the IoT for the development of distributed applications in the IoT realm.

Index Terms—Smart Cities, Cloud computing, Interoperability.

I. INTRODUCTION

The proliferation of the Internet of Things (IoT) has revolutionized the way we interact with technology, embedding intelligence into various aspects of our lives. From smart homes and connected vehicles to industrial automation and healthcare systems, IoT has enabled seamless connectivity and automation, promising enhanced efficiency, productivity, and improved quality of life. As the IoT ecosystem continues to expand, the need for efficient and reliable communication mechanisms becomes increasingly crucial.

Traditional centralized communication approaches, which rely on a centralized server to facilitate data exchange between IoT devices, face significant challenges when confronted with the scale and complexity of IoT deployments. Issues such as latency, scalability, network congestion, and vulnerability to single points of failure hinder the optimal functioning of IoT applications. To overcome such challenges, Machine-to-Machine (M2M) communications using a service-oriented approach (i.e., SOA) in IoT is a relevant approach. The M2M communication model is a promising paradigm providing a decentralized approach for pairs and systems communication. By leveraging the capabilities of M2M communications and SOA in the IoT field, devices and IoT systems/services can establish direct connections, enabling faster and more efficient data transmission, reducing latency, and enhancing overall system responsiveness, compatibility, and management.

Taking as an example the smart city ecosystem, we are led to deal with a heterogeneous IoT environment. In such a large scale deployment, we may leverage tens, hundreds

or even thousands of devices belonging to different broadcast domains while using distinct communication protocols (e.g., HTTP, MQTT, CoAP, etc.). Indeed, we can not think about a smart city deployment on which all the devices belong to the same network/sub-network and use the same protocol. Such constraints lead to the adoption of IoT-based systems in very specific, scale-limited and statically defined scenarios (e.g., Local Area Network (LAN)-based deployments).

To deal with the aforementioned challenges, we propose a system meant to ensure seamless interaction between devices, regardless of their localization, underlying networking configuration (e.g., being behind a NAT), and the used communication protocols. For this purpose, we integrate two systems namely IoTronic (a component of the Stack4Things middleware [1] and the Data eXchange Mediator Synthesizer (DeXMS) [2]. In particular, IoTronic is used as an IoT Infrastructure Manager (IIM) to deal with all the infrastructure-related duties (e.g., remote management and programming). On the other hand, DeXMS takes care of mediating communications between IoT devices/systems that use different protocols.

II. SYSTEMS DESCRIPTION AND INTEGRATION

To enhance the IoT smart environments, we integrated DeXMS and the OpenStack-based system Stack4Things. Fig. 1 depicts the architecture of the two OpenStack subsystems Neutron (blue subsystem) and IoTronic (red subsystem) dealing with networking and IoT infrastructure management, respectively, and DeXMS (purple subsystem) that deals with

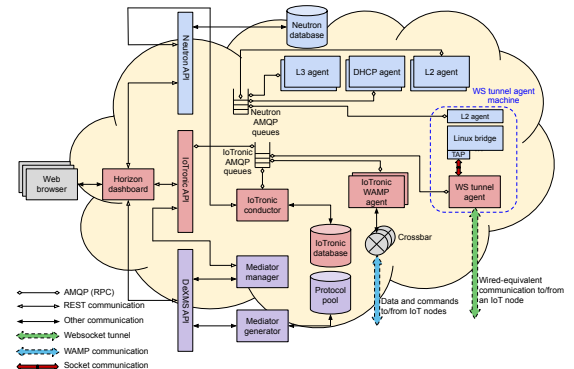


Fig. 1: IoTronic, Neutron and DeXMS integration design.

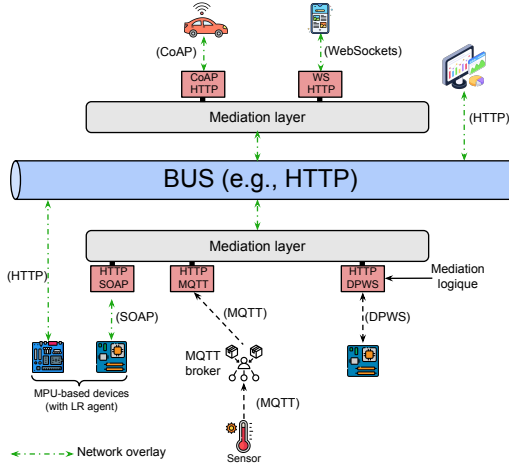


Fig. 2: An HTTP mediation bus for distributed IoT systems

protocols mediation. Fig. 2 conceptually shows an instance of the system bus regrouping a set of geographically distributed devices and systems. The architecture depicted in Fig. 2 defines a common middleware technology in a horizontal layer between IoT devices, systems, and IoT applications. The Cloud-based bus is shown as the horizontal blue cylinder. The vertical green dashed lines represent the OpenStack overlay network that covers both the Cloud and the Edge deployments (i.e., VMs, containers, and remote IoT nodes as reported in [3]). We mention that each Cloud tenant may create his/her bus connecting a set of distributed equipment following the as-a-Service (aaS) paradigm. All the devices and systems in the figure (including the DeXMS mediators) are part of the same overlay. The mediation layer which is reported in the figure deals with protocols' adaptation. The overlay links (i.e., green dashed lines) can be thought of as "connections" that are the sources and/or destinations of messages flowing over the bus. Regarding the Edge networking network service, the Neutron ports (i.e., the kernel virtual network TAP interfaces) get instantiated on the data center, precisely, on the machines hosting the WebSockets (WS) tunnel agents (i.e., the blue bashed component in Fig. 1) and then attached to the Neutron virtual switching facilities managed by the Neutron L2 agent (i.e., the *Linux bridge* in Fig. 1) [3]. It is worth mentioning that in this networking design, the switching facilities (i.e., virtual switches) are running on the cloud thus making the system highly scalable for particular hefty configuration requirements. Another advantage of the networking solution is the support of mobile devices/systems. Even if a device that belongs to a user-defined overlay is a mobile one and may experience connectivity changes due to vertical handovers (switching between different connectivity options such as WiFi, 3/4/5G) and thus, an underlay networking configuration change (i.e., IP address modification), the device will remain part of the same virtual network while keeping the same overlay IP address as before the connectivity interruption [3].

For protocols' adaptation, DeXMS is the pillar of our mediation layer. In the scenario in Fig. 2, a set of protocols can be translated to the bus protocol (in this case, HTTP). Technically, to instantiate a DeXMS mediator, we use Docker containers.

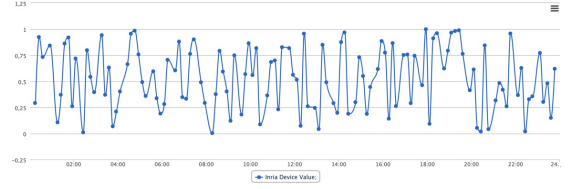


Fig. 3: The HTTP-based visualization service.

We devote containers' provisioning to the OpenStack Zun subsystem to be compliant with the other services we use (e.g., Neutron) and in particular, the virtual networking solution we described earlier [3].

III. TESTBED IMPLEMENTATION

We report in this section the description of a preliminary testbed being used to test the system. The implementation uses an IoT device (a Raspberry Pi) deployed in Paris, France, and managed using S4T. The device is configured to send data periodically (every 30 seconds) to an MQTT broker. To expose the IoT device-generated data through the HTTP bus, a DeXMS mediator is deployed. This mediator is managed through S4T and it is configured to be on a user-defined Neutron overlay network. As a client, we use an HTTP-based service deployed on a Raspberry Pi deployed in Messina, Italy. This later HTTP service (which is part of the same network overlay) can the MQTT data using GET requests in the destination of the DeXMS mediator. The service is configured to get data every 30 seconds, it calculates the average value in an interval of 10 minutes and then adds the value to a graph that can be visualized, in real-time, by users. Fig 3 shows the graph of the data being collected and calculated during 24 hours (23/06/2023). We highlight that the presented solution has been also tested with other protocols (e.g., MQTT to HTTP to CoAP and MQTT to HTTP to SOAP) to be used in a French project called Cloud Platform for Smart City (CP4SC).

IV. CONCLUSION

The management of heterogeneous interconnected things is a novel challenge that has to be addressed to lay down the foundation of a fresh way to manage and live in a (Smart) City. We believe that our system can address a set of challenges in IoT. As a future extension of this work, the reported system will be extensively tested with real-world scenarios related to the CP4SC project. In particular, we will use and adapt this solution to deal with a large-scale Intelligent Transportation System (ITS) as well as a use case related to digital twins.

REFERENCES

- [1] G. Merlino, D. Bruneo, S. Distefano, F. Longo, and A. Puliafito, "Stack4things: Integrating iot with openstack in a smart city context," in *2014 International Conference on Smart Computing Workshops*. IEEE, 2014, pp. 21–28.
- [2] G. Bouloukakis, N. Georgantas, P. Ntumba, and V. Issarny, "Automated synthesis of mediators for middleware-layer protocol interoperability in the iot," *Future Generation Computer Systems*, vol. 101, pp. 1271–1294, 2019.
- [3] Z. Benomar, F. Longo, G. Merlino, and A. Puliafito, "Cloud-based network virtualization in iot with openstack," *ACM Transactions on Internet Technology (TOIT)*, vol. 22, no. 1, pp. 1–26, 2021.