

# A comprehensive architecture for an IoRT-aware Business Process outsourcing into Fog and Cloud computing

Najla Fattouch<sup>a</sup>, Imen Ben Lahmar<sup>b</sup> and Khouloud Boukadi<sup>a</sup>

<sup>a</sup>University of Sfax, Mir@cl Laboratory, Sfax, Tunisia

<sup>b</sup>University of Sfax, ReDCAD Laboratory, Sfax, Tunisia

## Abstract

The Internet of Robotic Things (IoRT) is an aggregation of the Internet of Things (IoT) and robotic technologies. Thanks to its benefits, IoRT technology has seen significant growth, and it has swept various fields such as Business Process (BP), where it has given birth to a new business generation known as IoRT-aware Business Process (*IoRT-aware BP*). These processes necessitate costly execution infrastructures, which encourage their outsourcing. Business Process Outsourcing (BPO) is among the most common and appealing business strategies that aim to reduce enterprise operating costs. In this setting, the business managers seek to outsource their IoRT-aware BPs to the Fog and Cloud paradigms. Enterprises can avail of Fog and Cloud at their infrastructures and services during the execution of their process. Nonetheless, outsourcing an IoRT-aware BP to the Fog and Cloud is not a trivial task due to the diversity of the tasks requirements, the capacities of Fog/ Cloud devices, and the outsourcing techniques. In addition, most of the existing works are limited to outsourcing of classical BPs based on the Virtual Machine (VM) execution despite the proliferation of container technique that empowers the Fog/ Cloud computing through an horizontal scalability. Therefore, we aim through this paper to propose an architecture for optimal outsourcing of an IoRT-aware BP into the Fog and Cloud paradigms. Our architecture considers not only the VM execution but also the container one. Furthermore, our proposal aims to respond better to the process requirements by considering the available resources in Fog and Cloud infrastructures.

## Keywords

IoRT-aware BP, Fog computing, Cloud computing, Business Process Outsourcing

## 1. Introduction

In recent years, IoT technology and robotic are gaining more and more attention. Consequently, several researchers seek to benefit from both by combining them into one technology called *Internet of Robotics Things (IoRT)*. The IoRT is defined as a mixture of IoT and robot technologies that can imitate the human intervention where an IoT device and robots achieve human tasks [1]. The IoRT has several benefits, which make it among the proliferated technologies. The ability to bridge the gap between the physical and virtual worlds is among the significant IoRT advantages as it facilitates people's lives. Besides, it is characterized by its ability to automate tasks via an imitation of human intervention. Moreover, the IoRT technology can sweep different fields (e.g., health, education, etc.). In this setting, many business managers are looking to integrate the IoRT technology into their Business Processes (BPs) to decrease human intervention, gain productivity, and speed up production. In this regard, [2] defines this incorporation as a merger of the IoRT within the classic process to add


---

Tunisian Algerian Conference on Applied Computing (TACC 2021), December 18–20, 2021, Tabarka, Tunisia

✉ fattouchnajla@gmail.com (N. Fattouch); imen.benlahmar@isims.usf.tn (I.B. Lahmar); khouloud.boukadi@fsegs.usf.tn (K. Boukadi)



© 2021 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

dimension to a complicated situation. Besides, [3] defines the incorporation as an enterprise digitization via the automation of its process tasks. We get inspired by these definitions to propose the IoRT-aware BP notation to designate integration of the IoT and robots concepts within the traditional BPs. The IoT allows the connection of anything with the internet and provides sensing and actuating methods. However, a robot may represent either a machine that can move and execute tasks generally accomplished by users in a manual manner [4] or a software program (Bot) that can perform a script code [5]. In general, an IoRT-aware BP may necessitate costly execution infrastructures due to the high amount of data to be transferred in network. Towards this issue it is possible to reduce the enterprise operating costs by outsourcing partially or totally to an external provider [6]. This is called a Business Process Outsourcing (BPO).

The major benefit of outsourcing is to allow enterprises to get rid of the burden of the whole process's execution. Thus, they will concentrate on the core process rather than all process parts. Moreover, outsourcing guarantees that a task is accomplished faster and with a better quality thanks to rid the burden of human errors. As well as, it reduces the enterprise's costs (e.g., operational cost, recruitment cost, etc.). In literature, BPs are usually outsourced to the Cloud infrastructure due to the high availability and capacities of its datacenters [7]. Recently, a new trend came up based on outsourcing to Fog computing [8]. Fog computing is defined as an extension of the Cloud paradigm on its core to the edge of the network [9]. Thus, a Fog infrastructure provide an effective and increased assistance through a shift to the edge, or near devices of some Cloud-based applications [10]. Therefore, several researchers seek to avail from Fog and Cloud infrastructures on outsourcing their business process.

Our extensive literature exercise revealed that most of the current works deal with the outsourcing of traditional BPs. In [11], authors proposed an architecture to execute a BPs model including IoT and robot concepts on Fog and Cloud computing. However, the authors consider a robot as an IoT device. Nonetheless, a robot as a machine is quite different from a classical IoT device as it can move and execute tasks generally accomplished by users in a manual manner [4], while, the IoT devices (e.g., smartphones, smartwatches, etc.) need human intervention to achieve their movement. Moreover, most of these works are limited to the classic outsourcing solution based on virtual machine (VM) execution. A VM is defined as a computer file that behaves like a real machine computer. However, running a VM into a Fog/ Cloud infrastructure is often expensive rather than a container solution which represents a promising method. Besides, it is hard to manage a VM compared to a container [12]. In addition, most of the existing outsourcing approaches fail to consider the tasks requirements execution.

Therefore, this paper deals with outsourcing an IoRT-aware BP on Fog and Cloud computing via a suggestion of an architecture seeking to find optimal outsourcing of an IoRT-aware BP model into Fog and Cloud paradigms. Towards this objective, it considers the resource requirements for each process task. Besides, the proposed architecture attempts to deploy processes by using the virtualization method as well as containers that respond better to the limited capacities of Fog devices.

The remainder of this paper is organized as follows: Section 2, depicts the works that deal with the outsourcing of the traditional BP, the BP that incorporates the IoT, and the IoRT-aware BP. Section 3, details our proposed approach. Finally, Section 4 concludes the paper with an overview of our future directions.

## 2. Related work

Several works have considered the outsourcing of a BP into Fog and Cloud computing. In what follows, we overview the recent works that deal with the outsourcing of the traditional BP, the BP that integrates the IoT technology known as an IoT-aware BP and the IoRT-aware BP.

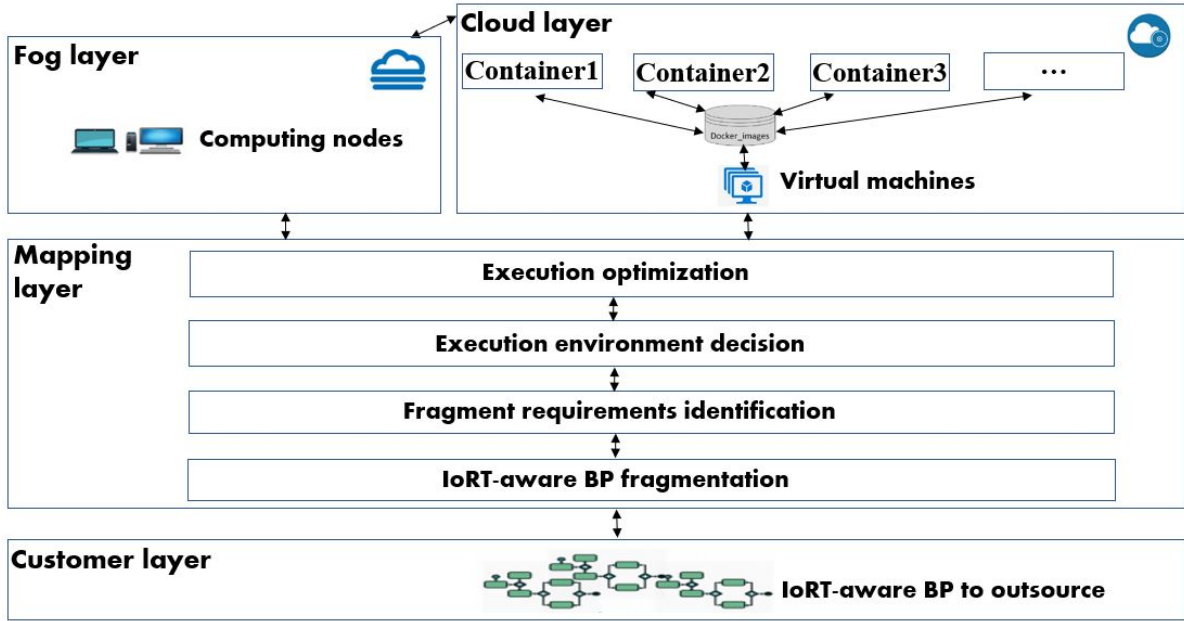
However, most of these works concentrate on the outsourcing of the traditional BPs into Fog and Cloud computing. In [13, 14], the authors target the outsourcing of a traditional BP into Cloud infrastructure based on containers. Towards this objective, they propose an architecture for an optimal deployment of the BPs. The suggested architecture is composed by three main components *BP Simulator*, *BP optimizer*, and *VM Scheduler*. However, the authors are limited to the Cloud infrastructures to outsource the process tasks without taking into account the Fog paradigm despite its relevance. In addition, their proposal does not consider the requirements of the process tasks during its execution. Another initiative, in [15], the authors propose an approach that helps the BP designers to find an optimal planning for the execution of the BP on the Cloud. In this regard, the authors propose an extension of the CloudSim which is a Cloud simulator to simulate the Cloud resources to deploy a BP model. However, their approach is limited to the outsourcing to the Cloud computing while the Fog computing represents an attractive paradigm that suit better the latency-sensitive applications. Furthermore, their proposal is limited to the classic outsourcing based on the VMs execution.

Among the recent works that deal with the outsourcing of an IoT-aware BP into Fog and/ or Cloud paradigms, we cite [16]. Their proposed architecture supports the distributed communication and the real-time stream in the intention to handle and enhance the IoT data. However, this architecture does not provide an optimal resource allocation solution. Another initiative presented in [17] where the authors propose a framework, named FogBusto to facilitate the end-to-end integration of the IoT on the Fog edge/ Cloud. This framework offers an independent platform to assume the execution and the interaction between IoT applications. It is composed of three main layers: *IoT Devices*, *Fog infrastructure* and *Cloud infrastructure*. Nonetheless, the proposed framework is limited to the outsourcing of the applications that integrates the IoT devices. Besides, the authors in [18, 19] propose, firstly, an orchestration algorithm based on the decision tree for the offloading of the IoT tasks on the Fog/ Cloud resources. Secondly, they propose an architecture to accomplish the synchronization between the IoT devices and the containers as a deployed application. Nonetheless, these proposals target the outsourcing of applications that incorporate the IoT devices without considering their BPs.

Back to the literature, few of the existed works have proposed solutions to execute a BP that incorporates the IoT and the robot concepts. Among these works, we cite [2] that designs and develops a global architecture to recognize robot actions that are integrated into the traditional process during a rescue attempt in real time. In this regard, the authors are using, on the one hand, a BPMN ad-hoc process model, and, on the other hand, a rescue robot out fitted with IoT sensors. The proposed architecture can be enhanced through the integration of some outsourcing methods to the Fog or Cloud computing. In addition, in [11] authors proposed an architecture to execute a BPs model including IoT and robot concepts on Fog and Cloud computing. However, the authors consider a robot as an IoT device. Nonetheless, a robot as a machine is quite different from a classical IoT device as it can move and execute tasks generally accomplished by users in a manual manner.

## 3. Outsourcing Architecture of an IoRT-aware Business Process

In this section, we present our proposed architecture for executing an IoRT-aware BP model on both Fog and Cloud paradigms. The goal is to suggest an effective matching of an IoRT-aware BP activ-



**Figure 1:** Overview architecture for an IoRT-aware BP outsourcing on Fog and Cloud.

ities with the available Fog/ Cloud resources. Nonetheless, this matching is not a trivial task due to the diverse requirements of the IoRT-aware BP tasks and each Fog/ Cloud resource’s specificities. Therefore, we propose in this paper an architecture for an optimal outsourcing of the process tasks on the Fog/ Cloud computing. As shown in Figure 1, our architecture consists of four layers, namely *Customer layer*, *Mapping layer*, *Fog layer* and *Cloud layer*.

In what follows, we detail the different architecture layers.

### 3.1. Customer layer

The entry point of the suggested architecture, is the customer request for outsourcing its IoRT-aware BP. This request includes:

1. A BP model is made of activities linked by workflow patterns like AND, XOR, etc. The process is modeled in BPMN, the de facto standard for BPs. We used the ISO/IEC 20924: 2018 [20] standard to specify the IoT concepts. To identify robot concepts, we relied on ISO/DIS 8373 : 2020, a pioneer in robotics standards [4].
2. The specificity and the QoS requirements for each process activity. The specificity presents the activity characteristics such as its complexity, repetitiveness, activity type, and so forth. The QoS requirements denote the resource requirements in terms of CPU, memory, bandwidth, etc.

### 3.2. Mapping layer

The *Mapping layer* represents the core layer of the proposed architecture. It is composed of a set of steps targeting the optimal process outsourcing. These steps are presented in what follows.

### 3.2.1. IoRT-aware BP fragmentation

This step allows the fragmentation of an IoRT-aware BP to a set of Single-Entry Single-Exit (SESE) blocks that are identified via a well known decomposition technique called Refined Process Structure Tree (RPST). The RPST decomposes a process to a set of blocks called SESE that have a single input and a single output. According to [21], an SESE is a sub-process that has one input and output and should preserve as much as possible the workflows between process and the sub process tasks.

We assume in our approach that an SESE is supported by a VM where its tasks are supported by some containers.

In our architecture, we considered the set of sub-process activities blocks of three types:

1. `Traditional bloc` is a bloc that contains traditional tasks used in the IoRT-aware BP process.
2. `IoT bloc` is a bloc that contains sensing and acted tasks.
3. `Script bloc` is a bloc that contains script tasks.

During the specification of the used blocks, we are limited to the above ones. The robot machines are exempted from the outsourcing as they will remain within the enterprises.

### 3.2.2. Fragment requirements identification

In the previous step, the BP model is fragmented into a set of SESE blocks. Then, there is a crucial need to estimate the cost for executing these SESE either on Fog or on Cloud infrastructure. The SESE requirements should be identified based on the requirements of its related tasks. These requirements may include required CPU, memory, and bandwidth, etc.

### 3.2.3. Execution environment decision

The propounded architecture targets the Fog/ Cloud paradigms where the SESE blocks of an IoRT-aware BP model will be executed. Indeed, among the significant challenges of this step is identifying which environment will support the execution of a SESE bloc. To address this challenge, we define a set of criteria that seems relevant for determining the execution environment of a SESE bloc. Hence, for each SESE, we consider the following criteria:

1. `Frequency` refers to the number of occurrences of sending data, from a task to the Fog and the Cloud devices, per unit of time. The SESE frequency refers to the higher task frequency among the SESE task's frequency. This criteria allows to know how many times the task sending data happens within a certain time period.
2. `Complexity` refers to the complexity of the set of tasks (T) that constitutes a SESE bloc. In the literature, there are several methods to estimate the process complexity value. The complexity estimation method cited in [22] is among the most used and attractive method. Hence, we are based on [22] to assess the complexity of a SESE. This method entails adding 10 points to the complexity score for each used application in the process/sub-process. Besides, it awards 2 point for each used screen in the process/sub-process. Finally, the method adds 2 points to the complexity score for each used condition. The authors assume that the complexity is categorized as a *Low Complexity* if its value is less than 30, as a *Medium Complexity* if its value is between 30 to 60 else the complexity is classified as a *high Complexity*. Based on the previously detailed complexity, we designate the SESE complexity by  $comp_S$  (see equation 1):

$$CompS = \left[ \sum [(used\_application * 10) + (used\_screen * 2) + (used\_condition * 2)] \right] / T \quad (1)$$

3. *Data\_Volume* presents the amount of data that the SESE tasks produce and transfer to Fog or Cloud. It equals to the sum of the amount of data of the SESE tasks. In our architecture, we assume that if the *Data\_Volume* is huge, it will be sent to a Cloud paradigm rather than to a Fog one where the Cloud computing has a high capability to handle an amount volume of data regarding Fog computing.
4. *Repetitiveness* presents the number of times of a task execution. The repetitiveness of a SESE refers to the maximum number of repetitions of the tasks that constitute the SESE. We assume that whether SESE has a high repetitiveness value, it's recommended to be executed on a near devices (Fog) else it executes to a Cloud.

Given the variety of decision criteria and decision alternatives, a Multi-Criteria Decision Making (MCDM) technique is required to decide the execution environment of the SESE. An MCMD technique provides meaningful decision-making for selecting the best alternative via a combination of multiple and different criteria. In the literature, there are several MCDM techniques such as SAW, AHP, MAUT, etc. In general, each decision-making technique follows three main steps. It starts with the specification of the relevant criteria. Then, it must attach a numerical value to the specified criteria to measure its relevance regarding the other criteria. Finally, it determines the ranking of each criterion [23]. In our work, we opted for using a fuzzy MCDM technique named fuzzy logic to manage the fuzzy nature of the proposed decision criteria like *Data\_Volume* and *Repetitiveness*. According to [24], the fuzzy logic approach has three main phases which are fuzzification, inference, and defuzzification. The Fuzzification phase converts the value of each input variable to a fuzzy value via a membership function. In the inference phase, a set of fuzzy rules is defined in natural language where these rules define the output value based on the input values. Finally, in the defuzzification phase, converts the output resulting value into a crisp values. In the literature, there are many defuzzification strategies, such as center of gravity (widely used), maximum, and the center average.

### 3.2.4. Execution optimization

After selecting the execution environment for each SESE, we aim in this step to propose an optimal resource allocation for an IoRT-aware BP on Fog and Cloud paradigms using both VM and container technologies. For this purpose, we rely on the Particle Swarm Optimization (PSO) as a meta-heuristic algorithm used in resolving the optimisation problems. PSO was introduced by Kennedy and Eberhart in 1995 [25]. It was inspired from animals behaviors such as flock of birds seeking a food source. The PSO knew an increased use thanks to its advantages such as its simplicity, robustness, and effectiveness. [26]. Moreover, the PSO necessitates less computational effort compared to other meta-heuristic algorithms such as the genetic algorithm. The PSO's principal idea is to look for an optimal solution by sharing information amongst members of a group generally called *population*. The population is initialized by a set of solutions generally called *Particle (P)* that are initialized randomly. It is crucial to denote, that we assume in our proposed architecture to fix the number of both process tasks and Fog/ Cloud devices to form the initial population. In the future work, we aim to deal with scalability of process tasks Fog devices. Each particle, has two main parameters that are the position noted by  $x$  which is presented in equation 1 and the velocity noted by  $v$  as it presented in equation 2. Among the main PSO parameters, we note the *pbest* that refers to the best position of a particle ( $P_i$ ), while the best position of the whole population noted by *gbest*.

Besides, during the applying of a PSO algorithm, the fitness function of a particle is specified according to the studied problem.

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$



T1	T2	T3
2	3	1
1	1	1

**Figure 2:** PSO particle structure for an IoRT-aware BP outsourcing.

$$v_i^{k+1} = wv_i^k + c_1r_1(pbest_i - x_i^k) + c_2r_2(gbest_i - x_i^k) \quad (3)$$

Where,  $c_1$ ,  $c_2$  present the acceleration coefficients. The  $r_1$  and  $r_2$  are two random numbers within  $[0, 1]$ .  $W$  presents the inertia weight. Finally,  $x_i(K)$  and  $x_i(K + 1)$  denote the current position of a particle  $i$  at the iteration  $k$  and  $k + 1$ .

For the optimal allocation of Fog and Cloud resources for executing an IoRT-aware BP, we suggest presenting a particle as a set of SESE bloc tasks, the Fog/ Cloud resource used for each SESE bloc tasks, and the VM used for each SESE. Let's consider as an example a SESE bloc with three tasks where the first task (T1) is allocated to container 2, the second task (T2) is allocated to container 3, and the third one (T3) to container 1. VM1 supports all these containers. Figure 2 depicts the particle structure for this example. The particle structure presented via a matrix of three lines and the number of columns is as much as the number of the SESE tasks. The first line presents the SESE tasks, the second gives the container identifier used for each task, and the last one determines the VM identifier that supports the SESE.

### 3.3. Cloud layer

The Cloud layer presents the layer that has a vast computing and storage capacities. In our architecture, we interest to the Cloud VMs as well as its containers. Both of them have a set of resources that should fit with the SESE requirements (e.g., CPU, memory, Bandwidth, etc). To respond better to these requirements, we assume that a VM will be used for the execution of SESE blocs, while the SESE tasks are executed by the containers.

### 3.4. Fog layer

The Fog layer is an intermediate layer between customer and Cloud layers. It is constituted by a set of computing nodes such as the smartphone, tablet, etc. These nodes are characterized by their ability to communicate with each other. In our architecture, we assume that each used computing node can support one or more VM. Moreover, each used VM is characterized by its QoS requirements such as the CPU value, the consumed memory, and the needed bandwidth. Furthermore, we assume that each used VM can be a support for one or more containers. The computing nodes of the Fog layer can communicate with the Cloud layer to assure persistent storage of their massive data.

## 4. Conclusion

An IoRT-aware BP merges the IoRT technology within the traditional process. The process has several advantages, such as assuming the Machine-to-Machine (M2M) communication between devices without human intervention. An IoRT-aware BP may be executed locally or outsourced partially or totally to an external provider known as Business Process Outsourcing (BPO). Therefore, the business

managers seek to benefit from the BPO strategy to reduce their processes costs. Besides, the BPO allows enterprises to get rid of the burden of executing the whole process. Thus, they will concentrate on the core process rather than all process parts. Nonetheless, outsourcing an IoRT-aware BP should carry about the diversity of resources and requirements of each process task. Towards these issues, we propose in this paper an architecture for the outsourcing of an IoRT-aware BP into Fog and Cloud infrastructures. We aim to profit from both virtualization and creation of containers methods during the execution of the process tasks on Fog and Cloud paradigms during our proposal.

As a future endeavor, we aim to apply our architecture in a real case where we aim to use a real Fog and Cloud resources on which we execute the IoRT-aware BP tasks. In addition, the resource allocation based on machine learning is among the attractive area for us to assume an optimal outsourcing of an IoRT-aware BP.

## References

- [1] S. N. Krishna, A. Vijayalakshmi, Internet of robotic things: An analysis, *IUP Journal of Computer Sciences* 15 (2021) 7–11.
- [2] A. Rebmann, J. Rehse, M. Pinter, M. Schnaubelt, K. Daun, P. Fettke, Iot-based activity recognition for process assistance in human-robot disaster response, in: D. Fahland, C. Ghidini, J. Becker, M. Dumas (Eds.), *Business Process Management Forum - BPM Forum 2020*, Seville, Spain, September 13-18, 2020, Proceedings, volume 392 of *Lecture Notes in Business Information Processing*, Springer, 2020, pp. 71–87.
- [3] Y. Masuda, A. Zimmermann, S. Shirasaka, O. Nakamura, Internet of robotic things with digital platforms: Digitization of robotics enterprise, in: *Human Centred Intelligent Systems*, Springer, 2021, pp. 381–391.
- [4] T. Haidegger, M. E. Barreto, P. J. S. Gonçalves, M. K. Habib, S. K. V. Ragavan, H. Li, A. Vaccarella, R. Perrone, E. Prestes, Applied ontologies and standards for service robots, *Robotics Auton. Syst.* 61 (2013) 1215–1223. URL: <https://doi.org/10.1016/j.robot.2013.05.008>. doi:10.1016/j.robot.2013.05.008.
- [5] A. Egger, A. H. ter Hofstede, W. Kratsch, S. J. Leemans, M. Röglinger, M. T. Wynn, Bot log mining: Using logs from robotic process automation for process mining, in: *International Conference on Conceptual Modeling*, Springer, 2020, pp. 51–61.
- [6] L. Fadila, Z. Karim, B. Djamel, Modelling business processes for outsourcing into the fog and cloud computing, in: *Proceedings of the 8th International Symposium on Data-driven Process Discovery and Analysis (SIMPDA)*, 2018, pp. 18–31.
- [7] N. Y. Chong, H. Bakht, An analysis of threats and challenges in the deployment of cloud computing, *International Journal of Computing and Network Technology* 6 (2018).
- [8] D. Battulga, D. Miorandi, C. Tedeschi, Speck: Composition of stream processing applications over fog environments, in: M. Matos, F. Greve (Eds.), *Distributed Applications and Interoperable Systems - 21st IFIP WG 6.1 International Conference, DAIS 2021*, Held as Part of the 16th International Federated Conference on Distributed Computing Techniques, DisCoTec 2021, Valletta, Malta, June 14-18, 2021, Proceedings, volume 12718 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 38–54.
- [9] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, P. A. Polakos, A comprehensive survey on fog computing: State-of-the-art and research challenges, *IEEE Commun. Surv. Tutorials* 20 (2018) 416–464. URL: <https://doi.org/10.1109/COMST.2017.2771153>. doi:10.1109/COMST.2017.2771153.



- [10] S. S. Gill, R. C. Arya, G. S. Wander, R. Buyya, Fog-based smart healthcare as a big data and cloud service for heart patients using iot, in: *International Conference on Intelligent Data Communication Technologies and Internet of Things*, Springer, 2018, pp. 1376–1383.
- [11] O. Bibani, S. Yangui, R. H. Glitho, W. Gaaloul, N. B. Hadj-Alouane, M. J. Morrow, P. A. Polakos, A demo of a paas for iot applications provisioning in hybrid cloud/fog environment, in: *IEEE International Symposium on Local and Metropolitan Area Networks, LANMAN 2016*, Rome, Italy, June 13-15, 2016, IEEE, 2016, pp. 1–2.
- [12] A. M. Potdar, D. Narayan, S. Kengond, M. M. Mulla, Performance evaluation of docker container and virtual machine, *Procedia Computer Science* 171 (2020) 1419–1428.
- [13] K. Boukadi, R. Grati, M. Rekik, H. B. Abdallah, From vm to container: A linear program for outsourcing a business process to cloud containers, in: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, Springer, 2017, pp. 488–504.
- [14] K. Boukadi, R. Grati, M. Rekik, H. Ben-Abdallah, Business process outsourcing to cloud containers: How to find the optimal deployment?, *Future Gener. Comput. Syst.* 97 (2019) 397–408. URL: <https://doi.org/10.1016/j.future.2019.02.069>. doi:10.1016/j.future.2019.02.069.
- [15] R. Ben Halima, S. Kallel, M. Ahmed Nacer, W. Gaaloul, Optimal business process deployment cost in cloud resources, *The Journal of Supercomputing* 77 (2021) 1579–1611.
- [16] A. Kallel, M. Rekik, M. Khemakhem, Iot-fog-cloud based architecture for smart systems: Prototypes of autism and COVID-19 monitoring systems, *Softw. Pract. Exp.* 51 (2021) 91–116. URL: <https://doi.org/10.1002/spe.2924>. doi:10.1002/spe.2924.
- [17] S. Tuli, M. R. Mahmud, S. Tuli, R. Buyya, Fogbus: A blockchain-based lightweight framework for edge and fog computing, *J. Syst. Softw.* 154 (2019) 22–36. URL: <https://doi.org/10.1016/j.jss.2019.04.050>. doi:10.1016/j.jss.2019.04.050.
- [18] C. Mechalikh, H. Taktak, F. Moussa, A fuzzy decision tree based tasks orchestration algorithm for edge computing environments, in: L. Barolli, F. Amato, F. Moscato, T. Enokido, M. Takizawa (Eds.), *Advanced Information Networking and Applications - Proceedings of the 34th International Conference on Advanced Information Networking and Applications, AINA-2020*, Caserta, Italy, 15-17 April, volume 1151 of *Advances in Intelligent Systems and Computing*, Springer, 2020, pp. 193–203.
- [19] C. Mechalikh, H. Taktak, F. Moussa, Towards a scalable and qos-aware load balancing platform for edge computing environments, in: *17th International Conference on High Performance Computing & Simulation, HPCS 2019*, Dublin, Ireland, July 15-19, 2019, IEEE, 2019, pp. 684–691.
- [20] ISO, Information technology – internet of things (iot) – vocabulary-iso/iec 20924, 2018.
- [21] J. Munoz-Gama, J. Carmona, W. M. Van Der Aalst, Single-entry single-exit decomposed conformance checking, *Information Systems* 46 (2014) 102–122.
- [22] D. H. Timbadia, P. J. Shah, S. Sudhanvan, S. Agrawal, Robotic process automation through advance process analysis model, in: *2020 International Conference on Inventive Computation Technologies (ICICT)*, IEEE, 2020, pp. 953–959.
- [23] E. Triantaphyllou, Multi-criteria decision making methods, in: *Multi-criteria decision making methods: A comparative study*, Springer, 2000, pp. 5–21.
- [24] L. A. Zadeh, Fuzzy sets, in: *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*, 1996, pp. 394–432.
- [25] J. Kennedy, R. Eberhart, Particle swarm optimization, in: *Proceedings of ICNN'95-international conference on neural networks*, volume 4, IEEE, 1995, pp. 1942–1948.
- [26] S. Shabir, R. Singla, A comparative study of genetic algorithm and the particle swarm optimization, *Int. J. Electr. Eng* 9 (2016) 215–223.