Fog Robotics—Understanding the Research Challenges

Victor Casamayor Pujol and Schahram Dustdar, *Vienna University of Technology, 1040 Vienna, Austria*

Fog robotics is an emerging topic that derives from cloud robotics, but similarly as fog computing, the applications require low latency connections in order to be useful in real life environments. This article presents a new perspective to the topic in which not only robotics takes advantage of the fog computing paradigm, but fog computing is able to leverage the robotics technology in order to enhance its features. This work highlights the benefits obtained by each technology when it is mixed with the other and sketches the relevant topics to research in order to make this partnership possible.

This article discusses a holistic perspective on the relation between the field of networked robotics, precisely fog robotics, and the fog computing paradigm. It will show that their synergies and relations will eventually blur the line that separates them. The main objective of this work is to derive the following required steps in order to take the best from both technologies.

The term fog robotics is appearing in the literature as an improvement to cloud robotics, see Gudi *et al.*[1] or Tanwani *et al.*[2] The research follows the trail of fog computing to improve the next generation of networked robots. In this sense, the benefits for robotic applications are clear: lower latency, enhanced data security, distributed computation, federated learning, distributed sensing, improved interfaces, distributed data storage, energy saving, offloading computations, etc. However, research focuses on creating specific architectures for fog robotic applications, such as in Gudi *et al.*,[3] loosing a great opportunity for generalization.

Thus, from an holistic perspective, similarly as it is done in Deng *et al.*[4] with respect to edge and AI, the research should focus on fog computing architectures that can embrace robots providing the same benefits and, additionally, adding them into the fog computing ecosystem. The latter will also enhance this ecosystem as robots can perform as "super"-nodes being able to do more actions than a typical fog node, such as physically act, sense the environment, move, send information, relay a network or do computations, see Figure 1 for an overview of the benefits of the system.

This article will be structured in four sections. The first will review the work on fog robotics, then, the second section will focus on the benefits that fog robotics provides to robot applications and it will show how these match the requirements for the fog computing paradigm. The third section will expose the benefits that robots can provide to fog computing. Finally, the fourth section will discuss current solutions that could embrace both technologies and the following research steps to achieve the maximum number of synergies from both technologies.

RELATED WORK

In Gudi *et al.*,[5] they first explicitly mention fog robotics as a required evolution from cloud robotics. The same team has two more publications on the topic. In Gudi *et al.*,[1] they show the improvement in network latency between a cloud robotic system and a fog robotic system. To do so, they simulate the experiments of delivery social robots but the precise tasks and the amount of data transferred it is not explicit in the article. In Gudi *et al.*[3] they keep working on the latency, however, this time they send data to real cloud infrastructures to compute it. This work also claims that

1089-7801 © 2021 IEEE
Digital Object Identifier 10.1109/MIC.2021.3060963
Date of current version 29 September 2021.

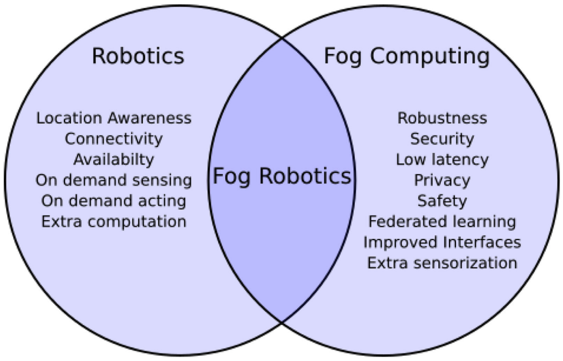

FIGURE 1. Benefits obtained in both technologies from a holistic perspective.

fog robotics requires a different architecture than fog computing, however, during this article it will be shown that their statement does not hold.

In any case, there is previous work that relates fog computing and robotics, such as in Dey and Mukherjee[6] where they offload SLAM computation to the fog through virtual machines. Doing so, they improve computational efficiency and avoid the requirement of a continuous connectivity in a cloud robotics scenario. From a wider perspective, the work in Fernandez *et al.*[7] can be considered as the first stepping stones toward fog robotics as it is based on the cooperation between sensor networks and robots and it highlights some issues that arise now such as the importance of time synchronization.

One of the most studied topics in the literature is the computation offloading for robots, which is directly inherited from cloud robotics, an example can be found in Sun *et al.*[8] However, the topic shifted toward offloading to the fog due to the latency issues related to the cloud connectivity. For instance, in Kattepur *et al.*[9] they estimate computation times for robot, fog, or cloud deployments. The article presents early work in which distributed task division is not considered nor the latencies of the different deployments. Most of the offloading work focuses on the SLAM problem and its derivatives, in Qingqing *et al.*[10] they offload the LIDAR odometry to a fog node, or in Sarker *et al.*[11] they do a similar work as in Dey and Mukherjee,[6] but they actually test it in a real environment creating an architecture with four layers: robot, edge, fog, and cloud layer each with different functionalities. Similarly, in Ichnowski *et al.*[12] fog robotics was used for offloading parallel computation for motion planning.

Some research also focused on using fog robotics to enhance the robots learning capacities such as in Tanwani *et al.*[2], 2019 or Tanwani *et al.*[13], 2020 this work creates a specific fog robotic architecture to improve the robot learning and it also tackles privacy issues by using only the sensitive data in the edge.

Other work uses the fog computing paradigm to improve robot teleoperation capabilities as in Tian *et al.*[14] or in Korovin *et al.*[15] they use the fog computing paradigm to increase the reliability of robot swarms.

Finally, there are a couple of book chapters dedicated to fog robotics. In Song *et al.*,[16] they introduce the topic on fog robotics as a required evolution from cloud robotics and in Yang *et al.*[17] they detail three use cases for fog robotics: SLAM, robot smart factory applications, and robot fleet formation.

Similarly, as it is intended to do with this article, in Shaik *et al.*[18] they analyze the required characteristics and challenges of a fog robotics architecture, however, they focus on industrial robots and they only focus on the benefits for the robots of using the fog computing.

FOG ROBOTICS

This section focuses on the benefits for robots provided by fog robotics and their direct relation with requirements of the general fog computing paradigm.

Low Latency

The main reason for the paradigm shift from cloud robotics towards fog robotics is the promise of low latency. Offloading computations enhance robot systems capabilities. However, high latency prevents robotic systems to work in near real time, which is essential for many reasons such as safety or fluidity.

Similarly, the fog computing paradigm arises due to the increase on the demand of cloud services driven by the huge amount of connected edge devices. These devices congest traffic to the clouds endangering connectivity and latency. The latter, as happens with robots, can prevent edge devices from performing tasks requiring real-time responses.

Privacy

Fog robotics provides better tools to deal with privacy issues. For instance, if the robot has to interact with a person it requires to record its face and voice. Then, it will require to process these data in order to provide the requested feedback. If this information is processed in the fog, the data can be distributed for processing, therefore, if a processing node is compromised the stolen data would not be complete. Also, fog nodes are not able to store all the data that they process which ensures that it will not be available after it is processed. Additionally, if for any reason it requires to be stored in a cloud it can be anonymized before uploading it.

The same trend drives fog computing, for instance, in a smart city context in which traffic cameras are checking if there is an accident. They are also recording people in the street, if this processing is performed in the fog nodes it is easier to ensure that these data are destroyed or anonymized after being processed.

Security

The distributed architecture of fog robotics can apply network security protocols for reaching the network but also between network nodes. Therefore, if a node is compromised the rest of the network can still be safe.

The same idea is directly translated to the fog computing paradigm.

Federated Learning

Deep and reinforced learning are key to provide many functionalities to a system of robots. Fog robotics can take the benefits from federated learning and apply it to robots. Federated learning provides the ability to share learning in distributed systems with different devices and capabilities. This brings new possibilities for robots like sharing the learning or taking advantage of prelearning models.

The fog computing paradigm is one of the enablers and also beneficiary of this federated learning as the edge devices can then train or use deep learning models with their limited capacity.

Improved Interfaces

The benefits for robotic systems come in two different ways. The first is related to latency, interacting with people is a near real time activity which requires heavy computations, therefore, offloading these to the fog and having the results with low latency will provide the expected feedback of these interactions. The second is related to other means of interaction, such as augmented reality (AR). This technology requires the use of devices that might not be part of the robot. But, if they are part of the fog robotics network, the feedback sensitivity and responsiveness obtained can enable this technology for robot interaction. The fog computing paradigm is enabling this technology as, besides of large computational needs, it requires a precise awareness of the location, which can be obtained in a fog computing paradigm.

Extra Sensorization

Another advantage for robots in fog robotics is the possibility of outsourcing some part of its sensorization. Robots might not be able to have all the required sensors equipped for their tasks due to many reasons such as power consumption, computation capabilities, maximum weight, etc. Therefore, being connected to the fog can allow them to use other surrounding sensors in order to enhance or enable a task.

In the fog computing paradigm, this feature is basic. The distribution and heterogeneity of nodes forces that some nodes will require to use sensed data from others, so this type of interactions will be a common trend.

Autonomy

Mobile robots can be very useful in a large amount of situations, however, their main constraint is autonomy which is obviously linked to their battery life. Therefore, extending their battery life is of foremost importance.

On their side, fog computing requires to deal with sensors and nodes that might not be plugged to an energy source, and therefore managing the energy consumption is fundamental for the fog computing paradigm. In this sense, robots can use the same methodologies to take advantage of it and therefore be operative for longer time.

Additionally, advantages already discussed such as offloading computations, carrying less sensors or using less powerful processing units allow robots to save energy.

Safety

Ensuring a safety operation of robots is crucial, and even more important if they operate in an environment with people. Synchronization of events and real-time response are key to this intend.

As it has been already exposed, this is a side effect of the low latency and the real-time network provided by the fog paradigm.

Robustness

Robustness is crucial for robots and avoiding single points of failure is key to ensure it. In the case of failure of a sensor, a processing unit or even a robot, the fog robotics paradigm can allow to easily switch to a device that is correctly working in order to keep with the development of the task.

The fog computing paradigm has the requirement of being resilient to this problem as it cannot be dependent on the availability of all the edge devices connected. Therefore, this property is intrinsic to its architecture as it requires the ability to assign tasks to the best performance devices of the current moment or to start using another device in case of failure or simply unavailability the one used until that moment.

Cooperation

Fog robotics has to enable cooperation between robots, which can be between the same type of robots or different robots. This cooperation will allow more specialized robots with better performances which will enable the completion of complex tasks.

The fog computing paradigm has in its architecture the means for cooperation. Everyone in the network can interact with anyone allowing multiple types of cooperation and coordination strategies.

Scalability

Fog robotics has to allow that a system of robots can grow as a function of the requirements of the task, providing versatility and utility to the system.

Similarly as in the previous point, the fog computing paradigm embraces scalability as the amount of devices connected is huge and, on top, the number can vary due to many different things and therefore, not only scalability but flexibility are key requirements for the fog computing paradigm.

ROBOTS IN FOG COMPUTING

In this section, several characteristics of the robotic systems will be highlighted in order to show their suitability to become fog "super"-nodes as they can provide more features than any typical fog node. In this sense, robots can take advantage of the fog computing paradigm while simultaneously provide the fog with relevant resources.

Mobility

This is one of the main characteristics of robots, although, not all of them can actually move. For those that can, the mobility provides several benefits to the fog computing paradigm.

Location Awareness

Mobility requires knowing its location, and this location awareness is a key characteristic for fog nodes. A fog node has to provide service to the edge devices that are at its surroundings but this includes static devices, such as traffic cameras, and mobile devices, such as smartphones of users walking by. Therefore, robots due to their knowledge of their physical location, can provide this proximity service.

This feature enhances applications by providing more useful information to the user, but also proximity is key to achieve low latency connections.

Connectivity

Edge devices can be located in network isolated areas, and these can require to be connected to the fog from time to time. Similarly, due to a catastrophic event, an area can become isolated from the network. In any of these cases, by sending a robotic system to its proximity it can be possible to directly relay the network so that the edge devices are connected to a larger fog network. This way the edge devices can continue providing its services even during catastrophic events.

Availability

Similarly to the previous paragraph, a robotic system can move toward a specific area in order to provide on-demand specific resources. A fog node cannot have enough resources to provide the services in a peak of demand, so to provide availability to all users of a robotic system can be used temporary to deal with the excess of demand.

Flexibility

Robots are typically designed to perform a single task, however, this task is usually complex. Therefore, robotic systems are equipped with several sensors, actuators, and computing resources. This means that robots can provide flexibility to the fog computing network in three different ways.

Computing

A robot can be used by the fog computing network as a common fog node with a specific availability. While robots are performing their tasks, they are usually using most of their computing resources, so in these situations using them as a fog node might not be possible. However, every certain amount of time they require to return to their charging station to be charged. In this case, they usually do not perform any tasks, and therefore, their computing resources could be lent to the fog.

Sensing

Robots are well equipped in terms of sensors, they have cameras, microphones, accelerometers, etc. Therefore, if they are in the vicinity they can feed the fog with relevant data to improve the quality of the service provided. Similarly, if an unexpected event takes place, it could be possible to send a robot that has a relevant sensor to the area of interest in order to verify what has happened. By doing so, the fog can be informed and proceed in the best possible way.

TABLE 1. Research Challenges.

Technology	Research challenges	
ROS	Capacity for task deployment with QoS constraints	Ensure system scalability
Distributed Orchestration	Research for flat orchestration architectures	Include robust architectures for time sensitive tasks
Control Loops	Ensure response time of all nested control loops	
Computing Elasticity	Provide new elasticity strategies to deal with real-time issues	Provide greater computing elasticity toward not expected events
Machine Learning	Use and develop machine learning techniques with deterministic output for real-time tasks	Develop federated learning techniques that estimate their confidence level

Acting

Equivalently to the previous paragraph, given that the robots are equipped with actuators, in some occasions, they can be able to perform a different task than the one they are designed for. As an example, if a fog node detects a car accident and there is a robot in the proximity that usually collects garbage from a park. The fog could send this robot to the area in order to remove debris from the road enhancing the recovery of the normal circulation.

FOG COMPUTING EMBRACES ROBOTICS

There are existing technologies that can help obtaining the best synergies from both fields. The following will describe them and it will look toward the required changes to leverage this new paradigm. These will guide the following steps in the research to enable this technology, see Table 1 for a summary of the following.

ROS

The robot operating system (ROS, https://www.ros.org/) is the most common middleware used in robotics. A robotic system is based on different processes running inside different pieces of hardware, these processes are called nodes. These are like programs that are executed concurrently on ROS. This middleware also provides a set of communication strategies between nodes through a publish/subscribe bus. The last major development called ROS2 (https://index.ros.org/doc/ros2/) uses DDS, which is a standard technology for data sharing across connected devices. This brings two major advantages, first it allows having a fully distributed system with the capability of discovering new nodes whenever they connect to the network and the opportunity of adding QoS policies to the communication between robotic parts.

From a wider perspective, a fog computing system can be understood as a distributed and dynamic robotic system with several computing processors (fog nodes) and several sensors and actuators (edge devices). Therefore, this middleware can be a basis for the middleware running in the fog, which would help standardizing the communication between the edge and the fog, given that many edge devices are able to run ROS. Additionally, it is possible to create containers with ROS and some nodes to be executed anywhere, for instance using Docker (https://hub.docker.com/_/ros/). However, it is not clear how fast or lightweight the deployment can be in order to meet real-time constraints. Related to this, if ROS is used for communication and also for deployment. It should be deployed only the required process for the task inside ROS, without including the whole ROS system. Also, it is unknown to us how the system can scale in such a large, distributed, and dynamic networks, as it is highlighted in Kronauer *et al.*,[19] the DDS standard is theoretically used for this type of systems but this is not yet the case of ROS.

Distributed Orchestration

In fog computing there are several solutions for task orchestration that derive from cloud computing, which can be used for this new paradigm. Currently, robots are seen as edge devices, so the fog computing is providing service to them, but the possibility of the robot serving the fog is not taken into account. This can force some structural changes in the architecture of these systems, for instance, many of these orchestration systems assume a clear separation between fog nodes and edge devices, but in this new scenario this separation becomes blurry. Additionally, it will be required to include architecture strategies to ensure system robustness to deal with time sensitive tasks, such as hot redundancy or majority voting systems.

Control Loops

Controlling such a complex and large system will require several nested control loops to have

FIGURE 2. Schematic of a street where fog robotics acting. The figure shows fog nodes in purple, people in orange, and robots in blue among other elements of the street.

awareness of the entire system state. At least, there will be the internal control loop for a robotic entity, the control loop for the communication between devices that collaborate with the robot in the task and a larger control loop to ensure that the task is performed with the expected quality. It is a concern to us how all these loops will be able to respond with enough celerity in real-time situations. We anticipate that in some cases it will be required to predict the outcome of these loops before they are completed in order to keep the response time low enough.

Computing Elasticity

Related with the previous paragraph, there is another concept inherited from cloud computing that has to be taken into account: Elasticity, which provides strategies to keep the provided task with the expected QoS and costs.[20] This is related in the sense that the output of the outermost loop will influence the QoS and this will trigger the required strategies. In this new scenario, the number of possible strategies to deal with the issues is greatly increased, as well as the possible and different issues that might arise. Therefore, it will be required to deal with not expected solutions to not expected issues, i.e., that they are not programmed beforehand. In such scenarios, it will not be possible listing everything that can happen with its most appropriate solution. Moreover, these strategies will need to deal with time sensitive tasks.

Machine Learning

Machine learning techniques will be required to solve many of the issues that will be arising during the development of this paradigm. Very popular machine learning solutions like deep learning or reinforcement learning are not always producing deterministic outputs. However, in several situations it will be required to have a deterministic outcome, for instance, when dealing with real-time tasks. Therefore, it will be needed to use machine learning techniques that can provide a high level of trust regarding their output or that their output is directly predictable. Additionally, when developing federated learning techniques they will be required to provide a confidence level, so that the QoS can be ensured.

CONCLUSION

Fog computing is a layer between edge devices and the cloud. From an architectural point of view, it is similar to the edge as it is distributed and at 1 hop from the edge device, which provides low latency connection but also a highly dynamic environment. From a resource perspective point of view, it is a layer closer to the cloud as it is able to perform computations or store data similarly as the cloud does.

Fog robotics is a new field that uses the aforementioned computing paradigm to greatly improve the robotic system features. However, we envision that this improvement of features has to take place in two senses. From the fog computing to the robots, as it has been seen in the related work and in the Fog robotics section. Furthermore, from the robotics field to the fog computing, as it has been explained in section Robots in fog computing.

Figure 2 shows an example on how we see several interconnections between both technologies. There is a robot who is carrying some goods for the person next to it, which is also announcing that they are crossing the street to a fog node. Immediately, this fog node, in purple, alerts the autonomous car that is approaching the pedestrian crossing. It had already detected the person and the robot, but this way the safety is reinforced. Also, there is a robot next to the park that it is currently charging. However, the two shops of the closer building are full of people that are requesting several services and the single fog node in charge of it is offloading some computations to the robot, which will be available until it is fully charged and then it will go to clean the park.

This article highlights the key technologies and, more important, it describes their required changes to enable the synergy between fog robotics and fog computing.

This is the only way both technologies will obtain the best of each other and this is the way we see the future of the networked things.

ACKNOWLEDGMENTS

This work was supported by the "Smart Communities and Technologies (Smart CT)" and it has received funding from the EU's Horizon 2020 Research and Innovation Programme under grant agreement No. 871525. EU web site for Fog Protect: https://fogprotect.eu/.

REFERENCES

1. S. L. K. C. Gudi, S. Ojha, B. Johnston, J. Clark, and M. A. Williams, "Fog robotics for efficient, fluent and robust human–robot interaction," in *Proc. 17th IEEE Int. Symp. Netw. Comput. Appl.*, Nov. 2018.
2. A. K. Tanwani, N. Mor, J. Kubiatowicz, J. E. Gonzalez, and K. Goldberg, "A fog robotics approach to deep robot learning: Application to object recognition and grasp planning in surface decluttering," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2019, pp. 4559–4566.
3. S. L. K. C. Gudi, B. Johnston, and M.-A. Williams, "Fog robotics: A summary, challenges and future scope," Aug. 2019, *arXiv:1908.04935*.
4. S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7457–7469, Aug. 2020.
5. S. L. K. Chand Gudi, S. Ojha, J. Clark, B. Johnston, and M.-A. Williams, "Fog robotics: An introduction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2017, p. 2. [Online]. Available: https://opus.lib.uts.edu.au/handle/10453/120748
6. S. Dey and A. Mukherjee, "Robotic SLAM—A review from fog computing and mobile edge computing perspective," in *Proc. Adjunct Proc. 13th Int. Conf. Mobile Ubiquitous Syst., Comput. Netw. Serv.*, 2016, doi: 10.1145/3004010.3004032
7. J. Capitán Fernandez, J. R. Martinez-de Dios, I. Maza, F. F. Ramon, and A. Ollero, "Ten years of cooperation between mobile robots and sensor networks," *Int. J. Adv. Robot. Syst.*, vol. 12, no. 6, Jun. 2015, Art. no. 70. [Online]. Available: http://journals.sagepub.com/doi/10.5772/60689
8. Y. Sun, X.-S. Zhou, and G. Yang, "Cost aware offloading selection and resource allocation for cloud based multi-robot systems," *IEICE Trans. Inf. Syst.*, vol. E100.D, no. 12, pp. 3022–3026, Dec. 2017. [Online]. Available: https://www.jstage.jst.go.jp/article/transinf/E100.D/12/E100.D 2017EDL8128/ article
9. A. Kattepur, H. K. Rath, and A. Simha, "A-priori estimation of computation times in fog networked robotics," in *Proc IEEE 1st Int. Conf. Edge Comput.*, Sep. 2017, pp. 9–16.
10. L. Qingqing *et al.*, "Edge computing for mobile robots: Multi-robot feature-based lidar odometry with FPGAs," in *Proc. 12th Int. Conf. Mobile Comput. Ubiquitous Netw.*, Nov 2019.
11. V. K. Sarker, J. Pena Queralta, T. N. Gia, H. Tenhunen, and T. Westerlund, "Offloading SLAM for indoor mobile robots with edge-fog-cloud computing," in *Proc. 1st Int.. Conf. Adv. Sci., Eng. Robot. Technol.*, May 2019.
12. J. Ichnowski *et al.*, "Fog robotics algorithms for distributed motion planning using lambda serverless computing," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2020, pp. 4232–4238.
13. A. K. Tanwani, R. Anand, J. E. Gonzalez, and K. Goldberg, "RILaaS: Robot inference and learning as a service," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4423–4430, Jul. 2020.
14. N. Tian *et al.*, "A fog robotic system for dynamic visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2019, pp. 1982–1988.
15. I. Korovin, E. Melnik, and A. Klimenko, "The fog-computing based reliability enhancement in the robot swarm," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11659 LNAI. New York, NY, USA: Springer Verlag, Aug. 2019, pp. 161–169, doi: 10.1007/978-3-030-26118-4 16
16. D. Song, A. K. Tanwani, and K. Goldberg, "Networked-, Cloud- and Fog-Robotics." Siciliano, B. Singapore: Springer Nature MOOCs, 2019, ch. 24.
17. Y. Yang *et al.*, "Fog-enabled multi-robot system," in *Fog-Enabled Intelligent IoT Systems*. New York, NY, USA: Springer, 2020, ch. 4, pp. 99–131, doi: 10.1007/978-3-030-23185-9
18. M. S. Shaik *et al.*, "Enabling fog-based industrial robotics systems," in *Proc. IEEE Int. Conf. Emerg. Technol. Factory Autom.*, Sep. 2020, pp. 61–68.

19. T. Kronauer, J. Pohlmann, M. Matthe, T. Smejkal, and G. Fettweis, “Latency overhead of ROS2 for modular time-critical systems,” Jan. 2021. [Online]. Available: http://arxiv.org/abs/2101.02074
20. S. Dustdar, Y. Guo, B. Satzger, and H. L. Truong, “Principles of elastic processes,” *IEEE Internet Comput.*, vol. 15, no. 5, pp. 66–71, Sep. 2011.

VICTOR CASAMAYOR-PUJOL is a Project Assistant with the Distributed Systems Group, TU Wien, Vienna, Austria. In 2020, he received the Ph.D. degree in ICT by Universitat Pompeu Fabra, Barcelona, Spain. He has a master in Intelligent Interactive Systems by UPF and a specialized master in space systems engineering by ISAE in Toulouse, France. He has also worked in space propulsion at the CNES in Paris, France. His current research interest relates robotics and the computational continuum. He is also interested in AI and their relation with robotics. Contact him at v.casamayor@dsg.tuwien.ac.at.

SCHAHRAM DUSTDAR is an Austrian computer scientist known for his work on distributed systems and elastic computing. He is a Professor of computer science and the Head of the Distributed Systems Group, TU Wien, Vienna, Austria. He became a Fellow of the IEEE for contributions to elastic computing for cloud applications. He is the corresponding author of this article. Contact him at dustdar@dsg.tuwien.ac.at.