

# Distributed Robotic Systems in the Edge-Cloud Continuum with ROS 2: a Review on Novel Architectures and Technology Readiness

Jiaqiang Zhang<sup>†</sup>, Farhad Keramat<sup>†</sup>, Xianjia Yu<sup>†</sup>, Daniel Montero Hernández<sup>†</sup>,  
Jorge Peña Queraltá<sup>†</sup>, Tomi Westerlund<sup>†</sup>

<sup>†</sup>Turku Intelligent Embedded and Robotic Systems (TIERS) Lab, University of Turku, Finland.  
Emails: <sup>1</sup>{jizhan, fakera, xianjia.yu, daanmh, jopequ, tovewe}@utu.fi

**Abstract**—Robotic systems are more connected, networked, and distributed than ever. New architectures that comply with the *de facto* robotics middleware standard, ROS 2, have recently emerged to fill the gap in terms of hybrid systems deployed from edge to cloud. This paper reviews new architectures and technologies that enable containerized robotic applications to seamlessly run at the edge or in the cloud. We also overview systems that include solutions from extension to ROS 2 tooling to the integration of Kubernetes and ROS 2. Another important trend is robot learning, and how new simulators and cloud simulations are enabling, e.g., large-scale reinforcement learning or distributed federated learning solutions. This has also enabled deeper integration of continuous interaction and continuous deployment (CI/CD) pipelines for robotic systems development, going beyond standard software unit tests with simulated tests to build and validate code automatically. We discuss the current technology readiness and list the potential new application scenarios that are becoming available. Finally, we discuss the current challenges in distributed robotic systems and list open research questions in the field.

**Index Terms**—Robotics; Distributed Robotic Systems; Edge-Cloud Continuum; Cloud robotics; Edge computing; ROS 2; Containerization; Computational offloading;

## I. INTRODUCTION

With robotic systems being increasingly connected, there is a growing intersection between the IoT and robotics domains [1]. Specifically, much potential can be found at the intersection of autonomous robots with the edge and cloud domains [2], [3]. This is leading to a myriad of new architectures, technologies and proposals for robots to leverage the edge-cloud computing continuum to enhance autonomy [4], drive multi-robot cooperation and human-robot collaboration [5], [6], [7], and provide additional computational capabilities [8].

The benefits of wider usage of edge and cloud computing paradigms and solutions to robotic systems are indeed enormous; however, widespread adoption requires further integration into the existing tools and frameworks, and particularly with the Robotic Operating System (ROS), the *de facto* standard middleware for the design and development of robotic systems. Therefore, even though there is an extensive literature in the areas of edge, fog, and cloud robotic systems, only recently more general frameworks.

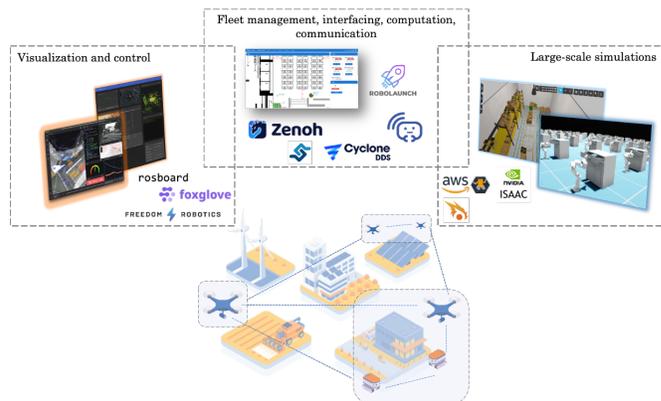


Fig. 1: Illustration of a subset of edge-fog-cloud robotic platforms and tools, and the potential areas where cloud and edge computing can play a key role for autonomous robots. From visualization and teleoperation to simulation and online learning, and including fleet management or computational offloading, connectivity can be leveraged to build more robust and capable distributed robotic systems.

From the perspective of users, we are witnessing a new wave of remote visualization and teleoperation tools that are cloud-native, mobile-ready, and ROS-based [9], [10]. At the same time, a rapidly increasing number of simulation tools for both desktop and cloud are enabling next-generation embodied AI and robot learning algorithms [11], [12]. These simulators are leveraging physics and game engines leading to more photorealistic and larger-scale simulations [13], [14], [15]. In the cloud, simulations are also being connected to continuous integration and development pipelines (CI/CD) [16], and to the Metaverse and digital twin platforms [17]. In parallel to all of these, new ROS extensions allow for easily switching between edge and cloud ROS deployments [18], [19], and a myriad of platforms for fleet management and multi-robot cooperation are emerging [10].

Within the above context, this review paper covers novel design approaches, architectures, and technologies that leverage the increasing connectivity and ubiquity of robots. In contrast to other surveys and reviews in the literature, we look closely at the compliance of such systems and architectures

with ROS. We delve into the basic concepts behind these technologies and frameworks (containerization, computational offloading, and security, among others), and into the specific technologies driving the state-of-the-art ROS integrations in the edge-cloud continuum. We then describe different use cases in the literature that demonstrate the potential of these technologies across industries and application areas.

The remainder of this document is organized as follows. Section II introduces background concepts related to ROS 2, DDS, containerization, DLTs, and computational offloading orchestration. Section III then reviews novel architectures and technologies for autonomous robots in the edge-cloud continuum, with Section IV focusing on application scenarios and use cases for the different technologies. Section V discusses open research questions and Section VI concludes the work.

## II. BACKGROUND

Through this section, we introduce the key technologies and concepts behind today's autonomous and distributed robotic systems, but also those with the potential to play a more important role in the next generation of technologies (e.g., DLTs).

### A. ROS 2 and DDS

The Robot Operating System (ROS) is the *de-facto* standard in autonomous robotic systems. ROS 2 is a revamped version of the original ROS middleware that has been under development for over half a decade. ROS 2 design principles include (i) distribution, (ii) abstraction, (iii) asynchrony, and (iv) modularity [20]. All of these, but specially distribution, modularity, and asynchrony, are tightly related to the new reality of connected and decentralized robotic systems, moving from single robots to fleets, and moving from single-host computing to computing architectures naturally distributed across the edge-cloud continuum. At the same time, the increasing connectivity and distribution bring new challenges, with ROS 2 design requirements including (i) built-in security, and the ability to support (ii) embedded systems, (iii) diverse networking conditions, and (iv) real-time computing. A final design requirement is (v) product readiness. Again, these extend the natural requirements of robust system architectures in the edge-cloud continuum, with a need for built-in security and resilience against alternating network conditions, as well as support for heterogeneous computing hardware.

A typical ROS 2 system is composed of a number of distributed processes or *nodes*, ranging from sensor drivers to algorithm implementations, high-level decision-making, or external interfacing and control. ROS 2 nodes communicate to each other through available communication patterns: (i) topics, an asynchronous message passing framework with a pub/sub system; (ii) services, synchronous request-response patterns; and (iii) actions, asynchronous interfaces with a request-feedback-response architecture that fits particularly well physical actions requiring time and physical interaction with a robot's environment. Additionally, ROS 2 provides a

series of abstraction layers, with the communication middleware relying on implementations for the industry-standard DDS protocol [21]. In the different architectures reviewed in this paper, we will look at how some use-cases require the use of additional communication middlewares or data streaming possibilities. This is owing to the data-intensive DDS middleware and the high-density and uncompressed raw data provided by some sensors (e.g., high-quality images or three-dimensional point cloud) not allowing for offloading computation from edge to cloud directly.

### B. Containerization

Virtualization technologies are some of the core foundations of cloud computing. Virtual machines (VMs) have been used as the backbone of cloud computing to provide virtualized operating systems, until the outbreak of container technology [22]. Containers offer a lightweight virtualization solution where the application can run on the environment with all the dependencies required. Docker and Kubernetes have revolutionized virtual techniques and cloud computing, and both are leading container orchestration tools nowadays [23], [24]. Docker is a commercial containerization platform and runtime where researchers and developers can build, deploy, and run containers. Kubernetes, which was originally developed by Google, is a platform to orchestrate containers [25].

### C. Computational Offloading

In its most basic form, it can be argued that a cloud robotics framework must allow for either (i) remote teleoperation of robots, (ii) running part of the computation required for autonomous operation in the cloud, or (iii) both of those features. This leads to computational offloading being one of the main studied areas in the literature of cloud robotics [26]. At the same time, the standardization of the abstract computing distribution paradigm that edge computing defines brought forward by the European Telecommunications Standards Institute (ETSI) with multi-access edge computing (MEC) is built around computational offloading possibilities to the radio access network (RAN) [27]. Indeed, with the advent of 5G, 6G and, beyond mobile networking solutions, low-latency and high-throughput computational offloading is made possible by using computational resources of servers deployed across the RAN layer. The "multi-access" term puts an emphasis on the multi-tenant infrastructure, meaning that such computing infrastructure is more limited than in the cloud and therefore it is more important to manage, or *orchestrate*, how it is shared between services. In next-generation robotic systems, computational offloading has the potential to become a more broad and complex area as computation is distributed across the edge, between edge and cloud, but also anywhere in between (e.g., the RAN layer) within a heterogeneous computing continuum. Key parameters taken into account in computational offloading orchestration include real-time latency and bandwidth [28].

#### D. Security

SROS2 provides a set of tools, and libraries for providing security, authentication, and access control for ROS 2 entities [29]. As ROS 2 is built on top of DDS, the DDS security specification is utilized to implement the security tools for ROS 2 at the communication level. OMG's DDS security specification [30] enforces security models in five Service Plugin Interfaces (SPI). SROS2 implements only three of these SPIs, which are authentication, access control, and cryptography. The authentication SPI enables ROS 2 entities to establish identities through the use of Public Key Infrastructure (PKI). The access control SPI restricts access to ROS 2 publish/subscribe communication channels only to predefined authenticated identities. The cryptography SPI encompasses cryptography functions such as encryption, decryption, digital signatures, and hash for the other SPIs.

#### E. Distributed Ledger Technologies

Consensus has always been an essential problem in distributed systems [31]. Therefore, decentralized consensus plays a key role in building towards robust solutions for the edge-cloud continuum. While earlier research on consensus was restricted to crash tolerance or a limited number of participants [32], with the advent of distributed ledger techniques (DLT), research on this domain has advanced [33]. DLTs provide an immutable ledger of the messages transmitted between the participant nodes. The shared immutable ledger facilitates consensus and provides a secure medium for the interaction of the participants in a distributed system. By using a DLT, trust is removed from third parties, resulting in more decentralized systems. Apart from decentralization, DLTs also offer security features, making them important components of distributed system development [34]. Distributed robotics systems are among the areas where DLTs are being used due to their need for decentralized agreement solutions and secure communication channels [35]. Blockchain solutions have already been applied to different robotic use-cases in the literature, but newer technologies such as those using directed acyclic graph (DAG) architectures are paving the way for more scalable and lightweight DLTs that have significant potential in distributed robotic systems [36], [37], [38], [39], [40].

Among the key technologies used in the literature are Ethereum [41], Hyperledger Fabric [42], and IOTA [43]. An overview of consensus algorithms and use cases within the more general edge-cloud domain can be found in [44]. A subset of those consensus algorithms is reviewed within the context of robot swarm systems in [45].

### III. TECHNOLOGY READINESS

This section covers the latest advances in cloud robotics with ROS 2, DLT integrations to ROS 2, containerization and edge-to-cloud system architectures. In Table I, we list some of the state-of-the-art robotics platforms or tools applied on the cloud and edge.

#### A. ROS 2 in the Cloud

Cloud computing has been playing an increasingly important role in the last few years in robotics, owing in part to the increase in the number of embedded devices and the availability of high-speed networking solutions [54], [55].

As an example, Amazon AWS Robomaker [46] is a cloud-based platform for developing, testing, deploying intelligent robotics applications, and running simulations. Robotics developers can scale and automate simulation workloads, run large-scale and parallel simulations with Amazon AWS's technologies on cloud computing, big data processing, and web services. Furthermore, AWS RoboMaker can run automated test within a CI/CD pipeline, train reinforcement models, and test multi-robot system.

An alternative ecosystem is NVIDIA Omniverse. The NVIDIA Isaac Sim [15] powered by Omniverse, provides photorealistic, physically accurate environments for developing, testing, and management of AI-based robots with sim-to-real capabilities. Regarding training perception models, Isaac Sim can generate synthetic data with Isaac Replicator allowing for adjustments in texture, colors, illumination, and location. It is available both locally and on the cloud for scalability. Notably, NVIDIA Isaac Sim is compatible with ROS and ROS2 and supports a variety of sensor modalities. ROS developers can move simulations from the most popular simulator in robotics, Gazebo, to Isaac Sim by exploiting the connectors between Gazebo and Isaac Sim. In addition, Isaac Sim is capable of real-time collaboration with teams from around the world for the development of AI robots. NVIDIA Isaac Sim can execute vision-based deep learning tasks including object detection, segmentation, spatial detection, Reinforcement Learning tasks provided by NVIDIA Isaac Gym, and other similar tasks.

#### B. Edge-to-Cloud ROS 2 Integrations

While cloud computing has become increasingly popular, computing has been extended from the cloud side to the edge side to decrease latency and network congestion. In addition to the cloud and edge paradigms, fog computing provides computing resources with high quality and low latency at the edge-side network, aiming to bring processing capabilities closer to the consumers, prevent overuse of cloud resources, and reduce operating costs [56].

One of the most prominent platforms supporting the edge-to-cloud deployment of ROS 2 nodes is FogROS2 [19]. FogROS2 extends the previous version of the platform, FogROS [18], tightly integrating with the ROS 2 launch and communication system. It allows for existing ROS 2 nodes to be deployed either in the cloud or at the edge, and then adapts the communication patterns accordingly to ensure real-time data delivery. The FogROS2 framework relies on a VPN connection to secure the ROS 2 DDS communication while allowing for the auto-discovery mechanisms in DDS to seamlessly bridge edge and cloud nodes. To allow for image processing on the cloud side for data generated at the edge, FogROS2 relies on H.264, a compression method used to

TABLE I: Selection of platforms, technologies, and tools in the robotics domain for the edge-cloud continuum.

Technology	Computing	Role/Type	Open-Source	Description
<b>AWS RoboMaker</b> [46]	Cloud	Simulation, deployment	Partly	Automated tests, scriptable simulations, end-to-end integration.
<b>Formant</b> [47]	Cloud	Fleet management	Partly	Platform for visualization and fleet management
<b>Robolaunch</b> [48]	Cloud	Simulation, deployment	Partly	End-to-end infrastructure for development and simulation
<b>ROBOTCORE</b> [49]	Edge	Hardware acceleration	Yes	Reference hardware architecture for edge acceleration.
<b>ROSBoard</b> [50]	Edge	Visualization	Yes	Lightweight web server for visualization and teleoperation.
<b>FogRos2</b> [19]	Edge-Cloud	Deployment	Yes	Integrated to ROS 2 launch system
<b>Foxglove</b> [51]	Edge-Cloud	Data recording, visualization	Partly	MCAP format alternative to standard rosbag in ROS 2
<b>NVIDIA Omniverse</b> [12]	Edge-Cloud	Ecosystem	Partly	Simulation, fleet management, HW acceleration, and others.
<b>Open-RMF</b> [10]	Edge-Cloud	Fleet management	Yes	Multi-tenant, interoperable, fleet management operations.
<b>Vulcanexus</b> [52]	Edge-Cloud	ROS 2+DDS Platform	Partly	ROS 2 tool set platform with cloud integrations.
<b>Zenoh</b> [53]	Edge-Cloud	Communication middleware	Yes	Discovery-efficient pub-sub communications.

reduce the bandwidth and latency and compress successive images in video format rather than individually (the default ROS compression). The H.264 encoding is done before publishing to the corresponding topic; the H.264 decoding is done before the subscribers receive the information. Also, a browser-based 3D visualizer, Foxglove, is used for remote monitoring and visualization. According to the paper, FogROS2 supports AWS and GCP. FogROS2 is evaluated on 3 example robot applications where cloud computing accelerates processing greatly compared to the computing capabilities of the robot and shows a significant performance benefit to using cloud computing, with the additional improvement from transparent video compression.

Other works in the literature have focused instead on improving the containerization of ROS nodes to enable more widespread adoption. For example, Bakshi et al. present a discussion on key considerations for the integration of ROS applications with container-based architectures [57]. Using Fault Tree Analysis, the authors identify the limitation of existing container-based virtualization architectures, the lack of persistent storage for stateful applications at the fog layer, which can be overcome by applying storage containers at the edge side.

### C. ROS 2 in Containers

Containers and DDS have been applied to solve the challenges caused by the growing number of edge devices and demand for low latency. Some container-based methodology for drones has been explored. At first, the containers are only used on the cloud side. In the paper [54], the authors propose an open-source drone-cloud platform based on Service-Oriented Architecture (SOA) and ROS. A Kubernetes cluster is deployed as cloud services on the first layer of a three-layer framework. The second and the third layer are ROS/ROS 2 and the simulator or flight controller.

Containerized services are used to assure the robustness of autonomous systems. In [58], the authors use containerized simulation to assure cloud-to-UAV communication. Xfce desktop environment and Virtual Network Computing (VNC) are used in the UAV simulation. The authors build a docker image with Dockerfile and run it on a Kubernetes cluster. In order to deploy the applications automatically, they use a Kubernetes cluster with three nodes to manage the docker images.

With a lightweight container platform such as K3S, edge-side applications can also be containerized. For instance, Lumpp et al. [55] propose a design methodology that containerizes and orchestrates ROS applications for heterogeneous and hierarchical hardware architectures. An extension of the Docker and K3S which is a lightweight Kubernetes distribution was used to build a Cloud-Edge platform that could deploy ROS-based robotic applications.

### D. DLT Integrations

The vast majority of experimental applications and use-cases in the literature involving DLTs and autonomous robotic systems are implemented through the integration of the Ethereum blockchain and ROS 1. The first few works in the literature integrating blockchain for robots were mostly related to the swarm robotics domain [35], [59], [60]. For example, in [35], multiple potential integrations and use cases are introduced for a variety of robots and blockchain uses (e.g., as a marketplace and cryptocurrency platform, or as a collaborative decision-making platform). A specific application of byzantine fault-tolerant consensus algorithms in blockchain solutions is the detection of byzantine agents within a robot swarm or multi-robot system. An early implementation of such an approach in [59] shows the potential of Ethereum smart contracts for identifying agents that are fabricating data or altering their sensor data in a cooperative mission. Other approaches in the literature include an architecture for building a shared *reputation* measure based on the quality or compliance of sensor data [60], as well as different approaches to collaborative decision-making processes [61].

More recently, however, new architectures have emerged that use next generation permissioned blockchain frameworks and DAG-based architectures such as IOTA's. For example, a framework for integrating Hyperledger Fabric Smart contracts with ROS 2 is presented in [36]. The high throughput of permissible blockchain frameworks makes them suitable for robotic systems, as shown by this integration. Furthermore, the paper discusses that Hyperledger Fabric has the potential to be used to provide identity management, immutable data logs, and secure communication channels for distributed robotic systems without drastically decreasing performance, and even near-real-time applications could be supported. Similarly, integration with the recently launched IOTA Smart Contract

Platform in [38] demonstrates the first ROS and blockchain integration that is tolerant to network partitions. The partition tolerance, in addition to byzantine tolerance, ensures the platform is usable in real-world networks where consistent connectivity cannot always be ensured.

Furthermore, a framework that exploits the functionality and speed offered by Hyperledger Fabric Smart Contracts to aid with the selection of Edge Nodes for Computational Offloading. The design and implementation of a system that analyzes the latency and resources in a network of nodes through blockchain smart contracts in [62] demonstrate that the technology can serve dependable distributed applications, and with the addition of execution of containerized applications, could perform operations autonomously.

### E. Security

A methodology for utilizing SROS2 on robotic distributed systems is presented by [29]. This methodology is iterative, starting with threat modeling, identity authentication, generating security requirements, and monitoring deployment. The iteration continues until all required communication channels have been secured. [63] have conducted experiments on evaluating how enabling ROS2 security affects the performance metrics such as latency and throughput of the ROS2 communication channel. The experiments are also carried out on wired and wireless networks, as well as Virtual Private Networks (VPN). The results of the experiment could be used to determine how the ROS2 security parameters should be configured to achieve the desired performance metrics. They have also performed a security analysis on the implementation of SROS2, and they have claimed that the implementation did not conform with the DDS security specification. SROS2 is still under development and the next versions would resolve the implementation compromises.

In addition, the use of DLTs, multiple works in the literature have been focusing on securing ROS, ROS 2, and distributed robotic systems in general. In [64], the authors introduce an approach to *secret* robot missions leveraging Merkle trees for data validation, a structure typically used in blockchain frameworks. The paper presents an architecture for robot swarms to achieve collaboration and cooperative missions without revealing to each other the content of the sub-tasks that are individually performed, but still allowing their validation by other robots. These missions are extended to more generic applications in [65], where the authors focus on instruction graphs that can be iteratively validated but are unknown when a mission starts.

### F. Distributed Machine Learning Frameworks

With the improvement of ML techniques and the capability to leverage hardware acceleration over the past decade, however, it is still challenging for computation-constrained edge devices to fully exploit the benefits of ML and other artificial intelligence algorithms. The modern era has witnessed the incredible proliferation of robotic edge devices as part of the Internet of Things. This situation demands high-level

collaboration among those robots situated in various locations across the globe.

A distributed machine learning framework was brought forward to utilize the multiple processors to accelerate the DL training process by parallel the computation and the data [66], [67]. A substantial quantity of literature is devoted to distributed DL with the purpose of achieving tighter collaboration between cloud and edge computing [68], [69]. This equilibrium between the two paradigms is projected to become more ubiquitous as the IoT age matures. Data security and user privacy are immediate problems that arise with the implementation of distributed DL across the cloud and edge. Through [70], [71], [72], various research directions have emerged with the goal of making distributed learning processes more scalable, secure, and privacy-preserving. Additionally, other research focuses on employing distributed DL for processing and learning from sensitive data such as health data [73] or public institutions [74].

In the distributed multi-robot system, Federated Learning (FL) has garnered attention [75]. Instead of collecting sensor data from numerous edge devices, federated learning maintains the data locally by sharing the trained models at the edge in a decentralized manner in order to create a global model. By doing so, FL ensures the system's security and privacy while maintaining its distributed nature. Due to the heterogeneous computing and communication capabilities of edge devices, fog learning was proposed to dynamically distribute ML model training throughout the continuum of nodes, from edge devices to cloud servers [76].

## IV. APPLICATION SCENARIOS

This section covers example applications of cloud robotics, and ROS 2 integration to DLTs and edge-cloud systems.

### A. Edge-to-Cloud ROS Applications

Containers have shown great potential in Edge-Cloud ROS 2 integrations. Docker and Kubernetes are both widely used container orchestration tools. In [77], the authors propose a hybrid cloud-based robot grabbing system, which is deployed on a Kubernetes cluster provided by Alibaba where each ROS node runs in separate docker containers. Alternatively, in [78], the authors propose a collaborative robotics testbed platform based on Docker. The testbed platform is deployed on a K3S.

Some blockchain-based applications can also apply to an Edge-Cloud system. For example, in [79], the authors propose a decentralized volunteer edge cloud architecture for real-time IoT applications. Kubernetes and KubeEdge are used to integrate smart contracts into the off-chain Edge-Cloud system.

Other works have discussed the potential of ROS2 for the next generation of robot swarms [80], and in blockchain technology for managing the interaction of robots within heterogeneous swarms of robots [28].

Looking into the application of one of the most mature frameworks for cloud support, FogROS [19], the authors of the paper show the potential for offloading image processing

applications, which can then leverage more GPU or even FPGA resources in the cloud. This ties together with the recent efforts in driving standardization for hardware acceleration in ROS [49], [81].

### B. ROS 2 and Distributed Learning

Distributed learning augmented by ROS2 can be utilized in a variety of robotic and autonomous system applications. Federated Learning has been demonstrated to be an effective approach for training and deploying vision-based obstacle avoidance models, both in the real world and when simulating photorealistic data [82]. Compared to the centralized technique for obstacle avoidance in navigation tasks, it can achieve comparable performance [83]. There are multiple research works on enabling Deep Reinforcement Learning in a distributed manner with the help of ROS2 for both the simulation and real-world application [84], [85], [86].

### C. ROS 2 and DLT Use Cases

The integration of DLTs to ROS-based systems has immediate applications in the area of byzantine agent detection [59], but specially within the more general of collaborative decision making. In that direction, the literature includes examples for cooperative mapping [38], where fabricated or noisy data is eliminated effectively by assessing the compliance and disparity between the maps submitted by different robots. An extension of that work in [39] shows that smart contracts combined with vision-based change and anomaly detection can also lead to more varied byzantine agent detection methods.

## V. DISCUSSION AND OPEN RESEARCH QUESTIONS

Throughout the paper, we have reviewed recent advances in cloud and edge robotics. The number of frameworks, tools, and platforms that have been recently released and are currently under development is staggering. Indeed, robots are no longer independent entities but are instead deployed in fleets and benefit from computing resources across the edge cloud continuum. While much has been improved, there is a number of open research challenges that both the IoT and robotics communities will be potentially solving in the near future.

Regarding open research questions, a number of technologies and methods have the potential to play a key role in solving some of the following open research questions:

*Live migration:* how to effectively address the live migration of containerized ROS 2 nodes between edge and cloud to ensure minimal loss of data? Compared to existing methods, live migration in robotic applications requires addressing real-time requirements and latency constraints, leading to safety concerns if a container migration affects autonomy [18], [19].

*Interoperability:* the Open-RMF standard is driving multi-fleet interoperability [10]; however, what other standards and ROS integrations are needed to drive the adoption of Open-RMF?

*Elasticity:* going beyond live migration, how can decentralized workload distribution be adapted to dynamically changing robotic systems, changes in their network topology, and

changes in their operational requirements? Elastic computing techniques can bring more robustness and resilience to multi-robot systems, optimizing the computational resources, but multiple challenges remain as these systems have a higher degree of dynamism than more typical cloud or edge clusters [2].

*Decentralization:* in this survey, we have covered decentralized decision-making through DLTs. The research in the area is however still incipient. There is significant potential to combine the decentralized consensus approaches in DLTs with the already distributed nature of ROS 2.

*Robustness and enhanced autonomy:* the computing resources available across edge and cloud can enable higher degrees of autonomy and robustness, for example with larger deep learning models in the cloud. However, this also needs additional fallback mechanisms to ensure that autonomy does not depend on stable connectivity to a remote host.

## VI. CONCLUSION

In this paper, we have given a description of novel design approaches, architectures, and technologies that leverage the increasing connectivity and ubiquity of robots. We have reviewed a selection of technologies, tools and frameworks that integrate with ROS 2 and allow for robots to better exploit the computational resources of the cloud-edge continuum. These technologies are introduced with a series of background concepts, and we look into application scenarios and use cases. We also discuss some of the current challenges and potential directions for future research.

## ACKNOWLEDGMENT

This research work is supported by the Academy of Finland's RoboMesh project (Grant No. 336061) and AeroPolis project (Grant No. 348480), and by the R3Swarms project funded by the Secure Systems Research Center (SSRC), Technology Innovation Institute (TII).

## REFERENCES

- [1] Pieter Simoons, Mauro Dragone, and Alessandro Saffiotti. The internet of robotic things: A review of concept, added value and applications. *International Journal of Advanced Robotic Systems*, 15(1):1729881418759424, 2018.
- [2] Jorge Peña Queralta, Li Qingqing, Tuan Nguyen Gia, Hong-Linh Truong, and Tomi Westerlund. End-to-end design for self-reconfigurable heterogeneous robotic swarms. In *International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2020.
- [3] Olimpiya Saha and Prithviraj Dasgupta. A comprehensive survey of recent trends in cloud robotics architectures and applications. *Robotics*, 7(3):47, 2018.
- [4] Jorge Peña Queralta, Li Qingqing, Zhuo Zou, , and Tomi Westerlund. Enhancing autonomy with blockchain and multi-access edge computing in distributed robotic systems. In *The Fifth International Conference on Fog and Mobile Edge Computing (FMEC 2020)*. IEEE, 2020.
- [5] Peng Huang, Liekang Zeng, Xu Chen, Ke Luo, Zhi Zhou, and Shuai Yu. Edge robotics: Edge-computing-accelerated multi-robot simultaneous localization and mapping. *IEEE Internet of Things Journal*, 2022.
- [6] Eduard Fosch-Villaronga and Christopher Millard. Cloud robotics law and regulation: Challenges in the governance of complex and dynamic cyber-physical ecosystems. *Robotics and autonomous systems*, 119:77–91, 2019.

- [7] Wuhui Chen, Yuichi Yaguchi, Keitaro Naruse, Yutaka Watanobe, Keita Nakamura, and Jun Ogawa. A study of robotic cooperation in cloud robotics: Architecture and challenges. *IEEE Access*, 6:36662–36682, 2018.
- [8] Wang Huamin, Ding Bo, and Jie Xu. Cloud robotics: a distributed computing view. In *Symposium on Real-Time and Hybrid Systems*, pages 231–245. Springer, 2018.
- [9] Viraj Dawarka and Girish Bekaroo. Building and evaluating cloud robotic systems: A systematic review. *Robotics and Computer-Integrated Manufacturing*, 73:102240, 2022.
- [10] Open Robotics. Open robotics middleware framework (open-rmf). <https://github.com/open-rmf/rmf>. [Online].
- [11] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, pages 621–635. Springer, 2018.
- [12] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.
- [13] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [14] Honglin Yuan and Remco C Veltkamp. Presim: A 3d photo-realistic environment simulator for visual ai. *IEEE Robotics and Automation Letters*, 6(2):2501–2508, 2021.
- [15] Nvidia. Nvidia isaac sim. <https://developer.nvidia.com/isaac-sim>. [Online] - Last access: 2022-10-19.
- [16] Giovanni Toffetti and Thomas Michael Bohnert. Cloud robotics with ros. In *Robot operating system (ROS)*, pages 119–146. Springer, 2020.
- [17] Xiao Li, Baris Can Yalcin, Olga-Orsalia Christidi-Loumpasfiski, Carol Martinez Luna, Maxime Hubert Delisle, Gonzalo Rodriguez, James Zheng, and Miguel Angel Olivares Mendez. Exploring nvidia omniverse for future space resources missions. 2022.
- [18] Kaiyuan Eric Chen, Yafei Liang, Nikhil Jha, Jeffrey Ichnowski, Michael Danielczuk, Joseph Gonzalez, John Kubiawicz, and Ken Goldberg. Fogros: An adaptive framework for automating fog robotics deployment. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 2035–2042. IEEE, 2021.
- [19] Jeffrey Ichnowski, Kaiyuan Chen, Karthik Dharmarajan, Simeon Adebola, Michael Danielczuk, Victor Mayoral-Vilches, Hugo Zhan, Derek Xu, Ramtin Ghassemi, John Kubiawicz, et al. Fogros 2: An adaptive and extensible platform for cloud and fog robotics using ros 2. *arXiv preprint arXiv:2205.09778*, 2022.
- [20] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66):eabm6074, 2022.
- [21] Stefan Profanter, Ayhun Tekat, Kirill Dorofeev, Markus Rickert, and Alois Knoll. Opc ua versus ros, dds, and mqtt: performance evaluation of industry 4.0 protocols. In *2019 IEEE International Conference on Industrial Technology (ICIT)*, pages 955–962. IEEE, 2019.
- [22] Claus Pahl. Containerization and the paas cloud. *IEEE Cloud Computing*, 2(3):24–31, 2015.
- [23] Thomas Uphill, John Arundel, Neependra Khare, Hideto Saito, Hui-Chuan Chloe Lee, and Ke-Jou Carol Hsu. *DevOps: Puppet, Docker, and Kubernetes*. Packt Publishing Ltd, 2017.
- [24] Steven Armstrong. *DevOps for Networking*. Packt Publishing Ltd, 2016.
- [25] Jay Shah and Dushyant Dubaria. Building modern clouds: using docker, kubernetes & google cloud platform. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0184–0189. IEEE, 2019.
- [26] Mahbuba Afrin, Jiong Jin, Akhlaqur Rahman, Ashfaqur Rahman, Jiafu Wan, and Ekram Hossain. Resource allocation and service provisioning in multi-agent cloud robotics: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(2):842–870, 2021.
- [27] S Kekki et al. Mec in 5g networks. *ETSI white paper*, 28:1–28, 2018.
- [28] Jorge Peña Queralta and Tomi Westerlund. Blockchain-powered collaboration in heterogeneous swarms of robots. *Preprint*, 2020. Presented at the 2019 Symposium on Blockchain for Robotics and AI Systems, MIT Media Lab.
- [29] Victor Mayoral-Vilches, Ruffin White, Gianluca Caiazza, and Mikael Arguedas. Sros2: Usable cyber security tools for ros 2. *arXiv e-prints*, pages arXiv–2208, 2022.
- [30] Object Management Group (OMG) Std. *DDS Security*, 9 2017. Rev. 1.1.
- [31] Michael J Fischer. The consensus problem in unreliable distributed systems (a brief survey). In *International conference on fundamentals of computation theory*, pages 127–140. Springer, 1983.
- [32] Bernadette Charron-Bost. Agreement problems in fault-tolerant distributed systems. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 10–32. Springer, 2001.
- [33] Shehar Bano, Alberto Sonnino, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, and George Danezis. Sok: Consensus in the age of blockchains. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pages 183–198, 2019.
- [34] Maria Gorbunova, Pavel Masek, Mikhail Komarov, and Aleksandr Ometov. Distributed ledger technology: State-of-the-art and current challenges. *Computer Science and Information Systems*, 19(1):65–85, 2022.
- [35] Eduardo Castelló Ferrer. The blockchain: a new framework for robotic swarm systems. In *Proceedings of the future technologies conference*, pages 1037–1058. Springer, 2018.
- [36] Salma Salimi, Jorge Peña Queralta, and Tomi Westerlund. Towards managing industrial robot fleets with hyperledger fabric blockchain and ROS 2. In *IEEE/SICE International Symposium on System Integration*. IEEE, 2023.
- [37] Salma Salimi, Paola Torrico Morón, Jorge Peña Queralta, and Tomi Westerlund. Secure heterogeneous multi-robot collaboration and docking with hyperledger fabric blockchain. In *IEEE 8th World Forum on Internet of Things*. IEEE, 2022.
- [38] Farhad Keramat, Jorge Peña Queralta, and Tomi Westerlund. Partition-tolerant and byzantine-tolerant decision-making for distributed robotic systems with iota and ROS 2. *arXiv:2208.13467*, 2022.
- [39] Sahar Salimpour, Farhad Keramat, Jorge Peña Queralta, and Tomi Westerlund. Decentralized vision-based byzantine agent detection in multi-robot systems with iota smart contracts. *arXiv:2210.03441*, 2022.
- [40] Paola Torrico Morón, Salma Salimi, Jorge Peña Queralta, and Tomi Westerlund. UWB role allocation with distributed ledger technologies for scalable relative localization in multi-robot systems. In *IEEE International Symposium on ROSE 2022*. IEEE, 2022.
- [41] Chris Dannen. *Introducing Ethereum and solidity*, volume 1. Springer, 2017.
- [42] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the thirteenth EuroSys conference*, pages 1–15, 2018.
- [43] Paulo C Bartolomeu, Emanuel Vieira, and Joaquim Ferreira. Iota feasibility and perspectives for enabling vehicular applications. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–7. IEEE, 2018.
- [44] Jorge Peña Queralta and Tomi Westerlund. Blockchain for mobile edge computing: Consensus mechanisms and scalability. *Mobile Edge Computing (Book Chapter)*, 2021.
- [45] Volker Strobel, Eduardo Castelló Ferrer, and Marco Dorigo. Blockchain technology secures robot swarms: A comparison of consensus protocols and their resilience to byzantine robots. *Frontiers in Robotics and AI*, 7:54, 2020.
- [46] AWS. Aws robomaker. <https://aws.amazon.com/robomaker/>. [Online].
- [47] Formant. Formant. <https://formant.io/>. [Online].
- [48] Robolaunch. Robolaunch. <https://www.robolaunch.io/>. [Online].
- [49] Víctor Mayoral-Vilches, Sabrina M Neuman, Brian Plancher, and Vijay Janapa Reddi. Robotcore: An open architecture for hardware acceleration in ros 2. *arXiv preprint arXiv:2205.03929*, 2022.
- [50] Dheera Venkatraman. Rosboard. <https://github.com/dheera/rosboard/>. [Online].
- [51] Foxglove. Foxglove. <https://foxglove.dev/>. [Online].
- [52] Vulcanexus. Vulcanexus. <https://vulcanexus.org/>. [Online].
- [53] Eclipse. Zenoh. <https://zenoh.io/>. [Online].
- [54] Golizheh Mehrooz, Emad Ebeid, and Peter Schneider-Kamp. System design of an open-source cloud-based framework for internet of drones application. In *2019 22nd Euromicro Conference on Digital System Design (DSD)*, pages 572–579. IEEE, 2019.

- [55] Francesco Lumpp, Marco Panato, Franco Fummi, and Nicola Bombieri. A container-based design methodology for robotic applications on kubernetes edge-cloud architectures. In *2021 Forum on specification & Design Languages (FDL)*, pages 01–08. IEEE, 2021.
- [56] H Sabireen and V Neelanarayanan. A review on fog computing: Architecture, fog with iot, algorithms and research challenges. *Ict Express*, 7(2):162–176, 2021.
- [57] Zeinab Bakhshi, Guillermo Rodriguez-Navas, and Hans Hansson. Fault-tolerant permanent storage for container-based fog architectures. In *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*, volume 1, pages 722–729. IEEE, 2021.
- [58] Lea Matlekovic, Filip Juric, and Peter Schneider-Kamp. Microservices for autonomous uav inspection with uav simulation as a service. *Simulation Modelling Practice and Theory*, 119:102548, 2022.
- [59] Volker Strobel, Eduardo Castelló Ferrer, and Marco Dorigo. Managing byzantine robots via blockchain technology in a swarm robotics collective decision making scenario. 2018.
- [60] M Dorigo et al. Blockchain technology for robot swarms: A shared knowledge and reputation management system for collective estimation. In *Swarm Intelligence: 11th International Conference, ANTS 2018, Rome, Italy, October 29–31, 2018, Proceedings*, volume 11172, page 425. Springer, 2018.
- [61] Pranav K Singh, Roshan Singh, Sunit K Nandi, Kayhan Z Ghafoor, Danda B Rawat, and Sukumar Nandi. An efficient blockchain-based approach for cooperative decision making in swarm robotics. *Internet Technology Letters*, 3(1):e140, 2020.
- [62] Daniel Montero Hernández, Jorge Peña Queralta, and Tomi Westerlund. Distributed ledger technologies for managing heterogenous computing systems at the edge, 2022.
- [63] Jongkil Kim, Jonathon M Smereka, Calvin Cheung, Surya Nepal, and Marthie Grobler. Security and performance considerations in ros 2: A balancing act. *arXiv preprint arXiv:1809.09566*, 2018.
- [64] Eduardo Castelló Ferrer, Thomas Hardjono, Alex Pentland, and Marco Dorigo. Secure and secret cooperation in robot swarms. *Science Robotics*, 6(56):eabf1538, 2021.
- [65] Jorge Peña Queralta, Li Qingqing, Eduardo Castelló Ferrer, and Tomi Westerlund. Secure encoded instruction graphs for end-to-end data validation in autonomous robots. *IEEE Internet of Things Journal*, 9(18):18028 – 18040, 2022.
- [66] Eric P Xing, Qirong Ho, Wei Dai, Jin Kyu Kim, Jinliang Wei, Seunghak Lee, Xun Zheng, Pengtao Xie, Abhimanu Kumar, and Yaoliang Yu. Petuum: A new platform for distributed machine learning on big data. *IEEE Transactions on Big Data*, 1(2):49–67, 2015.
- [67] Zhenheng Tang, Shaohuai Shi, Xiaowen Chu, Wei Wang, and Bo Li. Communication-efficient distributed deep learning: A comprehensive survey. *arXiv preprint arXiv:2003.06307*, 2020.
- [68] Huaming Wu, Ziru Zhang, Chang Guan, Katinka Wolter, and Minxian Xu. Collaborate edge and cloud computing with distributed deep learning for smart city internet of things. *IEEE Internet of Things Journal*, 7(9):8099–8110, 2020.
- [69] Haotian Jiang, James Starkman, Yu-Ju Lee, Huan Chen, Xiaoye Qian, and Ming-Chun Huang. Distributed deep learning optimized system over the cloud and smart phone devices. *IEEE Transactions on Mobile Computing*, 20(1):147–161, 2019.
- [70] Shaohuai Shi, Xianhao Zhou, Shutao Song, Xingyao Wang, Zilin Zhu, Xue Huang, Xinan Jiang, Feihu Zhou, Zhenyu Guo, Liqiang Xie, et al. Towards scalable distributed training of deep learning on public cloud clusters. *Proceedings of Machine Learning and Systems*, 3, 2021.
- [71] Yiran Li, Hongwei Li, Guowen Xu, Tao Xiang, Xiaoming Huang, and Rongxing Lu. Toward secure and privacy-preserving distributed deep learning in fog-cloud computing. *IEEE Internet of Things Journal*, 7(12):11460–11472, 2020.
- [72] Davit Buniatyán. Hyper: Distributed cloud processing for large-scale deep learning tasks. In *2019 Computer Science and Information Technologies (CSIT)*, pages 27–32. IEEE, 2019.
- [73] Praneeth Vepakomma, Otkrist Gupta, Abhimanyu Dubey, and Ramesh Raskar. Reducing leakage in distributed deep learning for sensitive health data. *arXiv preprint arXiv:1812.00564*, 2019.
- [74] Niranjan Balachandar, Ken Chang, Jayashree Kalpathy-Cramer, and Daniel L Rubin. Accounting for data variability in multi-institutional distributed deep learning for medical imaging. *Journal of the American Medical Informatics Association*, 27(5):700–708, 2020.
- [75] Yu Xianjia, Jorge Peña Queralta, Jukka Heikkinen, and Tomi Westerlund. Federated learning in robotic and autonomous systems. In *Procedia Computer Science*. Elsevier, 2021. 18th International Conference on Mobile Systems and Pervasive Computing (MobiSPC-2021).
- [76] Seyyedali Hosseinalipour, Christopher G Brinton, Vaneet Aggarwal, Huaiyu Dai, and Mung Chiang. From federated to fog learning: Distributed machine learning over heterogeneous wireless networks. *IEEE Communications Magazine*, 58(12):41–47, 2020.
- [77] Sheng Haiyan, Wei Shimin, Yu Xiuli, and Tang Ling. Research on robot grabbing system based on hybrid cloud. *The Journal of China Universities of Posts and Telecommunications*, 28(6):48, 2021.
- [78] Tim Farnham, Simon Jones, Adnan Aijaz, Yichao Jin, Ioannis Mavromatis, Usman Raza, Anthony Portelli, Aleksandar Stanoev, and Mahesh Sooriyabandara. Umbrella collaborative robotics testbed and iot platform. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–7. IEEE, 2021.
- [79] Ming-Tuo Zhou, Feng-Guo Shen, Tian-Feng Ren, and Xin-Yu Feng. Blockchain-based volunteer edge cloud for iot applications. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pages 1–6. IEEE, 2021.
- [80] Jorge Peña Queralta, Yu Xianjia, Li Qingqing, and Tomi Westerlund. Towards large-scale scalable MAV swarms with ROS2 and UWB-based situated communication. In *R2T2: Robotics Research for Tomorrow's Technology, organized by TII*, 2021. Best paper award.
- [81] Víctor Mayoral-Vilches and Giulio Corradi. Adaptive computing in robotics, leveraging ros 2 to enable software-defined hardware for fpgas. *arXiv preprint arXiv:2109.03276*, 2021.
- [82] Yu Xianjia, Jorge Peña Queralta, and Tomi Westerlund. Federated learning for vision-based obstacle avoidance in the internet of robotic things. 2022.
- [83] Yu Xianjia, Jorge Pe na Queralta, and Tomi Westerlund. Towards lifelong federated learning in autonomous mobile robots with continuous sim-to-real transfer. In *Procedia Computer Science*. Elsevier, 2022. Presented at EUSPN/ICTH 2022.
- [84] Matteo Lucchi, Friedemann Zindler, Stephan Mühlbacher-Karrer, and Horst Pichler. robo-gym—an open source toolkit for distributed deep reinforcement learning on real and simulated robots. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5364–5371. IEEE, 2020.
- [85] Nestor Gonzalez Lopez, Yue Leire Erro Nuin, Elias Barba Moral, Lander Usategui San Juan, Alejandro Solano Rueda, Víctor Mayoral Vilches, and Risto Kojcev. gym-gazebo2, a toolkit for reinforcement learning using ros 2 and gazebo. *arXiv preprint arXiv:1903.06278*, 2019.
- [86] Daniel Pinheiro Leal, Midori Sugaya, Hideharu Amano, and Takeshi Ohkawa. Fpga acceleration of ros2-based reinforcement learning agents. In *2020 Eighth International Symposium on Computing and Networking Workshops (CANDARW)*, pages 106–112. IEEE, 2020.