AORTA: Advanced Offloading for Real-time Applications

Ali Balador[1], Johan Eker[1,4], Raihan Ul Islam[1], Raquel Mini[1], Klas Nilsson[2]
Mohammad Ashjaei[3], Saad Mubeen[3], Hans Hansson[3], Karl-Erik Årzen[4]
[1]Ericsson Research, Kista, Sweden
[2]Cognibotics AB, Lund, Sweden
[3]Mälardalens University, Västerås, Sweden
[4]Lund University, Lund, Sweden

***Abstract*—We are currently witnessing the second wave of cloud services that go beyond web storefronts and IT systems, aiming for digitalization of industrial systems. Automation and time-sensitive systems are now taking their first steps toward the cloud. The AORTA project aims to facilitate this transition by providing key technology components needed for real-time services running in the cloud. The ambition is to support a future robotics ecosystem that enables a new level of flexible productivity in industrial production. AORTA will develop technologies that allow offloading of real-time services/functions to the edge and cloud. We will build upon recent advances in 5G, cloud, and networking technologies. The AORTA framework will support a fluid compute model where functionality will be dynamically deployed locally, in the edge, or in the cloud and support integration and real-time performance irrespective of where it executes. Results of the project will be demonstrated in a real-world robotics manufacturing and construction scenarios operating via a 5G network with real-time edge and large-scale cloud service. The AORTA technologies will provide opportunities for automation enterprises and system integrators by adding real-time capabilities needed to evolve beyond the currently closed ecosystem. They will also add value to telecom providers and operators that may host these new automation services in addition to their current portfolio.**

***Index Terms*—Edge computing, Offloading, Real-time applications.**

I. Introduction

The cloud computing paradigm provides a promising solution also for computation-intensive industrial applications, ranging from infrastructure monitoring, smart manufacturing to collaborative robots, to mention a few [1], [2]. Although cloud computing provides enhanced storage and computing capacity, it can cause high and unpredictable latencies [3], [4]. Edge computing is a distributed computing paradigm and ecosystem, where the execution environment (e.g., the compute and storage resources) is closer to the location of the source of data compared to the traditional cloud computing paradigm [5]. Edge computing also enables offloading of certain functionality from a resource-constrained device to edge or cloud servers. By offloading, it is possible to improve the performance of the local device, e.g., reduce power, and extend its capabilities by connecting advanced services only possible to run in the cloud. Another advantage of offloading is the reduction in the cost of local devices since they can be equipped with low-cost less powerful hardware [6], [7].

The reduced latency between the physical devices and the edge-hosted application instance has the potential to support time-critical industrial use cases. However, to ensure deterministic performance and support timing predictability of real-time applications that utilize the edge-cloud continuum, more enhancements are required in edge and cloud computing (the computation domain), as well as in the communication between embedded devices and edge/cloud, and communication among the servers within edge and cloud (the communication domain). Enhancements in these domains include flexible distributed execution environments that provide real-time performance, modelling and timing analysis of real-time applications while considering dynamic adaptation and offloading, as well as dynamic orchestration and resource management of the applications in the edge-cloud continuum.

Recent works show that the benefits of edge and cloud computing can be leveraged if the computation and communication domains are deployed in an integrated fashion and are managed in a coordinated way [3], [8]. Building upon the existing edge and cloud computing concepts, one major need is a real-time cloud, with capabilities to design and offer deterministic and timing predictable computing services for deploying reliable real-time applications on the edge-cloud. This must cover how the computation domain and the deterministic communication network should be integrated by considering possible combinations of different types of 3GPP Non-Public Networks (NPN) [9], deterministic networks like Time Sensitive Networking (TSN) [10], [11], and Deterministic Networking Architecture (DetNet) [12] together with cloud and edge deployment models, such as solutions based on Kubernetes, Webassembly[1], and Julia[2].

With the intention of ensuring deterministic performance and supporting timing predictability of real-time applications that utilize the edge-cloud continuum, we will develop the AORTA (Advanced Offloading for Real-Time Applications)

[1]https://webassembly.org/
[2]https://julialang.org/

framework. AORTA allows offloading application components that require real-time services and functionalities, as well as integration of them with services in the edge-cloud. The ambition is to support, for example, advanced robotics or manufacturing applications in utilizing non-local services in a predictable fashion. Using recent advances in predictable communication and compute technologies (such as TSN, Kubernetes, and 5G), we will build a new real-time computing platform consisting of a portable real-time container that supports dynamic code migration for offloading. Together with enhanced robot programming principles, interactive programming and optimization of motions will be supported both technically and business-wise, thereby forming a basis for a new eco-system.

The rest of this paper is organized as follows: Section II presents existing technologies that provide predictable communication and compute solutions. The overall concept of AORTA framework is presented in Section III. Section IV describes the use-case scenario. The main envisioned results of AORTA framework are presented in Section V. Finally, Section VI presents the conclusions.

II. Background

A. Programmable, Light Containerisation

Existing solutions for executing untrusted code on shared hardware such as virtual machines and containers are not optimized in respect to code size and performance. This is because an operating system, libraries or configuration files must be provided together with the code for execution. Large program overheads are often caused by supporting libraries, configuration files, etc. As a more lightweight model, AORTA aims at using a standardized bytecode format - where only the program byte code itself is transferred, originally written in the developers language of choice.

Lightweight and portable execution environments have been identified as a crucial enabler for higher flexibility and dynamism of application deployment in distributed networks [13], [14]. In AORTA, we experiment with WebAssembly technologies to fill this gap. WebAssembly [15] is an *open* binary instruction format for a stack-based virtual machine, designed to support existing programming languages in a web browser environment.

In the browser, WebAssembly is generally faster than JavaScript due to its more compact format and manual memory management. The virtual machine is, however, not bound to browsers, but can be used standalone on various platforms. A deployment flow is where a high-level language such as C, C++, Rust, Python and so on can be compiled to WebAssembly (using clang) and executed on various hardware or software platforms. Webassembly allows applications to execute within a wide range of devices and operating systems.

B. Modelling and Resource Analysis

Several works have addressed modelling and analysis of resources and predictability in edge-cloud computing systems. A recent survey [16] presents existing offloading approaches with various modelling techniques, e.g., mathematical modelling and simulation-based modelling. Moreover, an overview of task offloading approaches on edge-cloud computing has been presented in [17]. These surveys indicate that there are still significant gaps in achieving an optimal resource utilization in edge-cloud computing using task offloading techniques. In particular, the proposed offloading techniques have limitations to be used for time-critical applications where strict timing requirements are imposed by the applications.

In the context of real-time applications with short latency requirements some of the recent proposals include an architecture with low-latency support for smart mobile devices [18], multi-task offloading for mobile devices [19], and dynamic task offloading [20], to name a few. However, many of the existing works on task offloading models focus on optimizing metrics such as reducing end-to-end delay, but fail to consider communication costs between tasks. In addition to these, several works address utilization of specific technologies for real-time edge-cloud computing [21]. Among them we can mention, RT-Kubernetes [22] that supports dynamic adaptation in the edge-cloud computing based on Kubernetes technology, orchestration of services based on Kubernetes [23], and real-time containers to support soft-real-time applications with run-time adaptation of containers [24]. The above-mentioned works present mathematical models for tasks and applications mainly customized for computation.

Given the above overview, a comprehensive model that can represent tasks, real-time constraints, communication among tasks, relation to applications, and resource utilization is missing. Such a model can help in understanding such relations in complex systems, while providing a base for timing and resource analysis of services in the edge-cloud computing.

C. Control over the Cloud

Advanced controllers, such as Model-Predictive Controllers (MPC) [25], are examples of applications that might need more compute resources than what is available in the local device, e.g., in the local robot controller. In MPC, an optimization problem is solved in each sampling period, which can be quite costly in terms of computational resources. Therefore, offloading the optimization solver to the edge/cloud and, hence, closing the control loop over the edge-cloud, is an interesting approach. This also creates real-time requirements on the end-to-end latency from the local device to the edge-cloud and back to the device again, i.e., the latency involves both the communication and the computation latency.

The MPC approach is flexible. It is possible to compensate for latency variations in different ways up to a certain extent, and to dynamically modify the compute requirements by formulating a smaller optimization problem, thus trading control performance for resource requirements. This opens the possibility for a negotiation-based approach where the application can adapt its resource requirements when they cannot be met by the infrastructure. MPC-based control over the cloud is the topic that is discussed by authors in [26].

Moreover, there are other applications that fit into the proposed framework. There has been a large amount of work done on deep neural network-based classification of, e.g., images captured using local devices such as mobile phones where the training and inference is performed in the edge-cloud. Compare to previous MPC based application, the latency requirements are typically not so stringent and often it is only in one way, i.e., from the device to the edge-cloud direction for this kind of applications. Therefore, based on computation capabilities of edge and cloud applications requirements can be negotiated in our proposed framework.

III. AORTA: Offloading for Real-time Applications

The goal of AORTA is to develop a framework that allows offloading of real-time services and functionalities to the edge and cloud, as well as their integration with services in the edge and cloud. The ambition is to support, for example, advanced robotics or manufacturing applications in utilizing non-local services in a predictable fashion. We will build upon recent advances in predictable communication and compute technologies, such as TSN, Kubernetes and 5G. A new real-time computing platform consisting of a portable real-time virtual machine that supports dynamic code migration for offloading will be developed.

The anticipated framework consists of two interacting parts: the modelling and scheduling part, and the application part. The first part consists of a new technique to allow holistic modelling of real-time applications that utilize the edge-cloud continuum. The modelling technique will be expressive enough to model the application's resource information such as end-to-end timing, network bandwidth, memory, and the applications' criticality levels. This will be combined with a resource verification framework for these applications. This part is also envisioned to support continuous monitoring of the application's resource usage together with run-time analysis of the applications' timing behaviours while considering on-the-fly adaptation and offloading. Based on this, parts of the application or the entire application can be dynamically offloaded to the most suitable compute platform in the embedded devices to edge-cloud continuum. The application part of the framework is responsible for expressing the real-time requirements of the applications and informing the scheduling part about this. We will develop an adaptive negotiation-based approach where the applications can adapt their resource requirements when they cannot be met by the edge-cloud infrastructure and where the infrastructure can adapt the resources provided when requirements and/or the total amount of available resources change. For the control applications this implies that new cloud-aware control system architectures are needed [26].

With the developed framework we will be able to (1) support the evolution of a third-party ecosystem of real-time components, and (2) support collaborative robotics use cases in new applications areas. The results will be demonstrated and evaluated in the context of industrial and construction robotics in two dimensions:

1) Ecosystem evolution: The ambition is to support a future ecosystem around robotics that enables a new level of flexible productivity in manufacturing. To achieve true flexibility, it must be possible to combine control, sensor fusion, learning, and vision components with real-time requirements to meet demands on both productivity and flexibility. Some of the components could be deployed in the embedded device, whereas others must be deployed in the edge or cloud due to resource demands. The developed framework will support a fluid model where components may be dynamically deployed locally, at the edge or in the cloud and support integration among the components and supporting real-time performance irrespective of where they execute. Figure 1 shows a robot cell connected to edge and cloud service. In this case, the robot control functions can be moved to the edge and cloud depending on dynamic requirements and available resources to improve operation and performance. The outcome of AORTA will allow for the offloading as well as hosting of modules such that third party technology providers (of equipment or operations) contribute to an evolved ecosystem that involves both large enterprises and SMEs.

2) New collaborative robotics: Applying robotics in new collaborative use cases with physical human interaction is challenging, in particular from a communication and compute resource perspective. The individual robots must each be guaranteed sufficient resources so that the total task can be achieved. In Industry 4.0 and 5.0, cloud-based digital twin technology and human interaction are key. However, the subsystems with strict timing requirements are still running locally at the embedded device and manual operation during production is neglected. Online upgrades and dynamic software configuration are not considered for the real-time parts of the system. The outcome of AORTA will enable ad-hoc production cells where humans and robots collaborate safely and efficiently, i.e., with strict timing.

IV. Use Cases

The most relevant use-cases are defined from core limitations that are experienced within two industrial application areas: manufacturing and construction. Robot usage in manufacturing is standard, at least in large scale series production. In smaller companies and for products that vary due to short series or customization, there are still many challenges.

Since competitive robot systems have control systems that are optimized for the embedded online computations needed for high-performance motion, additional functions for motion optimization and processing of sensor data requires additional computing power. This is however better provided on-demand for cost-efficient utilization of resources (including maintenance and upgrades). Offloading from the robot controller to the cloud is then an attractive alternative, but current networking and factory practices do not allow the needed solutions.

Defining the robot tasks is done by means of teaching/programming the robot in some language, which so far has been vendor specific due to the the way motions and

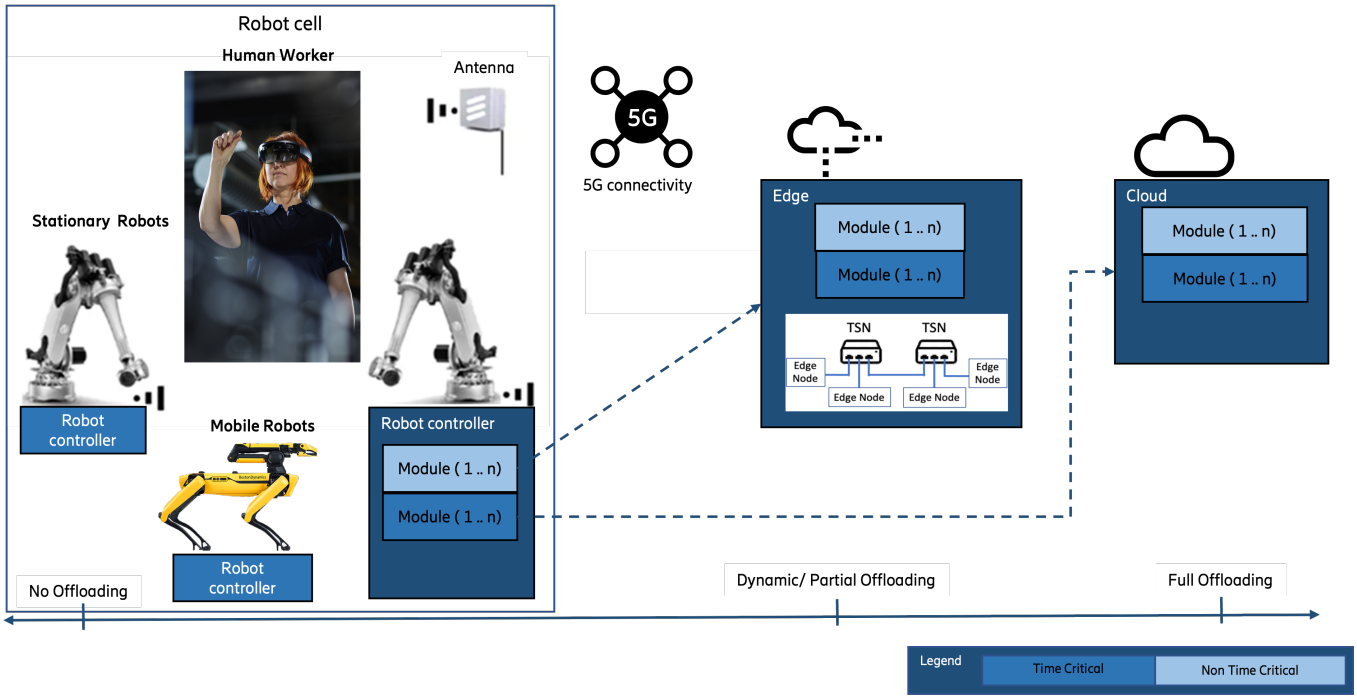

Fig. 1. A use case scenario of dynamic offload of functionality during operator switches from configuration mode to run time mode.

real-time sensing is supported. In AORTA, we have a new take on motion programming by developing new programming principles based on the language Julia. This will additionally provide a language candidate for edge and cloud computing, for instance with a WebAssembly backend, and will allow third parties to (technically and business-wise) provide new solutions for human-robot interaction.

Construction work-places are in need of robot assistance, in a safe way and such that robots can collaborate with humans, for instance allowing an experienced worker to guide or adjust motions for improved outcome. This can happen outdoors, or in nearby indoor setups for so called prefab production.

There are two main differences compared to the situation in manufacturing: (1) tasks and environments are less known and less structured, and hence more sensing and sensor processing in combination with robot world modelling are needed. This requires software that is better maintained as cloud services; and (2) there is no stationary IT infrastructure based on wires and wireless networks, and there is no time/desire/resources to setup such networking for each new construction site. Instead, data networks based on 4G/5G/6G could be used, but when reliability and predictability are required the mobile networks need some tailoring to be applicable on construction sites.

In most situations where robots still do not deliver the desired assistance to human labor, there are computing needs that are best fulfilled by a cloud-edge approach. However, real-time predictability is often required, and for the needed agility of the communication, a telecom approach is clearly more suitable than (standard rigid) local factory networks. Both (worker) safety and (information) security are considered within the AORTA developments. This is, however, less demanding than mission-critical solutions since robots in production are allowed to slow down or even stop when some part of the system is not working as intended.

From a business perspective, future agile but predictable networking and computing will create new opportunities for smaller tech providers (such as Cognibotics) to enhance production systems (in manufacturing and construction) by means of new solutions from innovative telecom providers (such as Ericsson).

V. Discussions and Envisioned Results

The goal of AORTA is to support a future ecosystem around robotics by developing technologies that allow offloading of real-time services and functionality to the edge and cloud. This enables a new level of flexible productivity in the manufacturing industry. In terms of the economic impact, the activities proposed in AORTA can ensure deterministic and predictable communication across verticals including, but not limited to, Industry 4.0/Industry 5.0, automotive, augmented virtual reality, and healthcare. The potential for innovation-based growth and leveraging investments from AORTA results is extensive and can assist in generating autonomy within EU in emerging 6G technologies associated with applications of end-to-end deterministic communication. The implementation of AORTA is divided in three main parts as illustrated in Figure 2. The following sections describe these parts as well as the envisioned results in each part.

A. Holistic Modelling and Resource Analysis

In AORTA, we aim at developing 1) new techniques and a framework for holistic modelling, 2) on-the-fly adaptation and offloading, and 3) resource verification of real-time applications that utilize the edge-cloud continuum. The developed techniques will allow holistic modelling of real-time applications that utilize edge-cloud continuum. The modelling technique is envisioned to support timing predictable networks for these applications such as TSN and URLLC via 5G. The technique will be expressive enough to model the application's resource requirements such as end-to-end timing, network bandwidth, memory, and criticality levels. Furthermore, the envisioned resource verification framework will be comprehensive enough to support pre-runtime as well as runtime analysis of the applications' timing behaviour while considering on-the-fly adaptation and offloading to edge-cloud continuum.

The framework will support on-the-fly adaptation by continuously monitor the application's resource requirements. Based on the dynamic updates in the resource requirements, parts of the application (software components/tasksets) or entire application can be dynamically offloaded to the most suitable compute platform in the embedded nodes to edge-cloud continuum while considering the application's real-time constraints. In addition, the framework will also consider the networks' resources.

B. Control Algorithm and Architecture

In AORTA, we aim at developing dynamic and distributed edge and cloud-aware control systems, including dynamic negotiation with the edge-cloud infrastructure about resource requirements, resource provisioning, and system performance, in a scalable intent-based fashion. The proposed control systems will support dynamic integration with third party components to form ad-hoc distributed systems. The components shall preferably be adaptive and able to dynamically express requirements on, e.g., CPU cycles, deadlines, etc. The components may be loosely coupled during configuration but provide tight coupling with timing guarantees during run-time.

We will also develop edge and cloud negotiation and offloading methods for scenarios involving multiple collaborative robots where the resource requirements among the robots are strongly connected. A combination of stationary robots and mobile robots will be considered that includes human interaction. The work will address services that are safety-critical and require graceful degradation in case of loss of connectivity. This includes dynamic scenarios where offloaded functionality becomes unavailable and local services need to take over.

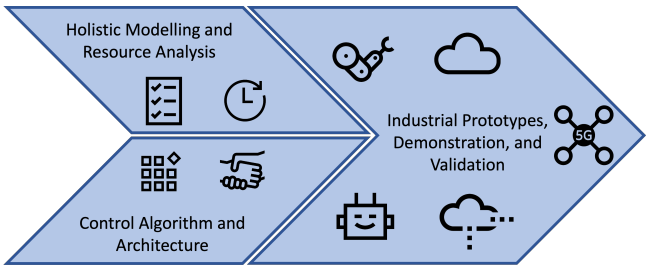

Fig. 2. Implementation parts of AORTA.

C. Industrial Prototypes, Demonstration, and Validation

We aim at prototyping and showing the potential of the AORTA results in the context of robotics and automation. Technical requirements for the AORTA technologies will gradually be refined into specifications for AORTA components. The application scenarios will be developed in the virtual robot environment. The use case will serve to evaluate and validate the new techniques, methods, and the AORTA framework. An industrial interactive robot control and programming platform will be used as base for the final validation and demonstration of the AORTA framework.

VI. Conclusions

The development of future computing and communication platforms for distributed computing and telecom systems should take into account resource awareness/management, scalability (which includes technical considerations such as networking, as well as business considerations spanning from enterprises to small companies). These features need to be seamlessly operating in the cloud, in the upcoming real-time cloud, on the edge, locally in interactive systems, and in embedded devices. In robotics and automation, the present Industry 4.0 with emphasis on efficient usage of interconnected machines and their digital twins, needs to be enhanced towards Industry 5.0 that better reflects needs coming from human interaction and interactive motion adjustment/programming. The presented AORTA initiative brings together these two areas, to form the technical foundations of a 6G standard that is open to new types of business along a variety of value chains.

Specifically, upcoming 6G networks will be a game-changer that enable wide-spread use of robots for both automated and interactive (collaborative) tasks. Robotic systems can be interactively taught in a more intuitive and safe manner than possible today, also on distributed work-places without prior local installation of a fixed factory network. During automated operation, monitoring and optimization of motions take place. In both these modes of operation, new languages, computing, communication, resource-modeling and performance optimization will be supported by advanced offloading to the real-time cloud. From robot-related demonstrations within two years, AORTA will pave the way for the next generation of systems with interconnected devices, with apparent upcoming opportunities for many other products too.

Acknowledgement

This work was partially supported by the project AORTA (Advanced Offloading for Real-Time Applications) that has received funding from Swedish Innovation Agency (VINNOVA) under grant agreement No 2022-03039.

REFERENCES

[1] R. Chaâri, F. Ellouze, A. Koubâa, B. Qureshi, N. Pereira, H. Youssef, and E. Tovar, "Cyber-physical systems clouds: A survey," *Computer Networks*, vol. 108, pp. 260–278, 2016.

[2] B. Costa, J. Bachiega, L. R. de Carvalho, and A. P. F. Araujo, "Orchestration in fog computing: A comprehensive survey," *ACM Comput. Surv.*, vol. 55, jan 2022.

[3] C.-H. Hong and B. Varghese, "Resource management in fog/edge computing: A survey on architectures, infrastructure, and algorithms," *ACM Comput. Surv.*, vol. 52, sep 2019.

[4] S. Mubeen, P. Nikolaidis, A. Didic, H. Pei-Breivold, K. Sandström, and M. Behnam, "Delay Mitigation in Offloaded Cloud Controllers in Industrial IoT," *IEEE Access*, vol. 5, pp. 4418–4430, 2017.

[5] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019.

[6] T. Nylander, M. Thelander Andrén, K.-E. Årzén, and M. Maggio, "Cloud application predictability through integrated load-balancing and service time control," in *2018 IEEE International Conference on Autonomic Computing (ICAC)*, pp. 51–60, 2018.

[7] A. Ullah, H. Dagdeviren, R. Ariyattu, J. DesLauriers, T. Kiss, and J. Bowden, "Micado-edge: Towards an application-level orchestrator for the cloud-to-edge computing continuum," *Journal of Grid Computing*, vol. 19, 12 2021.

[8] M. Ashjaei, S. Mubeen, M. Daneshtalab, V. Casamayor, and G. Nelissen, "Towards a predictable and cognitive edge-cloud architecture for industrial systems," in *Real-time And intelliGent Edge computing workshop*, July 2022.

[9] 3GPPP Specifications, https://www.3gpp.org/specifications.

[10] IEEE 802.1 Time-Sensitive Networking (TSN) Task Group, https://1.ieee802.org/tsn.

[11] M. Ashjaei, L. Lo Bello, M. Daneshtalab, G. Patti, S. Saponara, and S. Mubeen, "Time-sensitive networking in automotive embedded systems: State of the art and research opportunities," *Journal of Systems Architecture*, vol. 117, 2021.

[12] N. Finn, P. Thubert, B. Varga, and J. Farkas, "Deterministic Networking Architecture." RFC 8655, Oct. 2019.

[13] G. Wikström *et al.*, "6g – connecting a cyber-physical world: A research outlook towards 2030," *Ericsson, White paper*, Feb. 2022.

[14] A. Sefidcon, W. John, M. Opsenica, and B. Skubic, "The network compute fabric – advancing digital transformation with ever-present service continuity," *Ericsson Technology Review*, June 2021.

[15] A. Haas *et al.*, "Bringing the web up to speed with webassembly," in *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2017, (New York, NY, USA), p. 185–200, Association for Computing Machinery, 2017.

[16] K. Gasmi, S. Dilek, S. Tosun, and S. Ozdemir, "A survey on computation offloading and service placement in fog computing-based iot," *The Journal of Supercomputing*, vol. 78, pp. 1–32, 02 2022.

[17] M. Akhlaqi and Z. Hanapi, "Task offloading paradigm in mobile edge computing-current issues, adopted approaches, and future directions," vol. 212, p. 103568, 03 2023.

[18] B. Osibo, Z. Jin, B. Marah, C. Zhang, and Y. Jin, "An edge computational offloading architecture for ultra-low latency in smart mobile devices," *Wireless Networks*, vol. 28, 07 2022.

[19] H. Zhang, Y. Yang, X. Huang, C. Fang, and P. Zhang, "Ultra-low latency multi-task offloading in mobile edge computing," *IEEE Access*, vol. 9, pp. 32569–32581, 2021.

[20] C.-F. Liu, M. Bennis, M. Debbah, and H. V. Poor, "Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing," *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 4132–4150, 2019.

[21] V. Struhár, M. Behnam, M. Ashjaei, and A. V. Papadopoulos, "Real-Time Containers: A Survey," in *2nd Workshop on Fog Computing and the IoT (Fog-IoT 2020)* (A. Cervin and Y. Yang, eds.), vol. 80 of *OpenAccess Series in Informatics (OASIcs)*, (Dagstuhl, Germany), pp. 7:1–7:9, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020.

[22] S. Fiori, L. Abeni, and T. Cucinotta, "Rt-kubernetes: Containerized real-time cloud computing," in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, p. 36–39, Association for Computing Machinery, 2022.

[23] B. Blieninger, A. Dietz, and U. Baumgarten, "Mark8s-a management approach for automotive real-time kubernetes containers in the mobile edge cloud," *RAGE 2022*, p. 10.

[24] V. Struhár, S. Craciunas, M. Ashjaei, M. Behnam, and A. Papadopoulos, "React: Enabling real-time container orchestration," in *26th IEEE International Conference on Emerging Technologies and Factory Automation*, September 2021.

[25] J. Rawlings, D. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017.

[26] P. Skarin, *Control over the Cloud: Offloading, Elastic Computing, and Predictive Control*. PhD thesis, Department of Automatic Control, Nov. 2021.