A survey on Edge Computing in Operating Systems

Sanjith BS
SCORE, Vellore Institute of Technology, Vellore

Abstract

Edge computing has emerged as one of the leading trends in modern computing, which shifts the load of cloud services to the edge of the network. To achieve this, it needs a reliable OS. Hence, it is the integration of edge computing with operating systems (OS) that needs to be studied. This survey defines edge computing, its requirement, challenges, applications and how it is different from cloud computing. The next section focuses on the evolution of OS and how it is an integral part of edge computing by laying out its roles and responsibilities, the challenges it faces and how it is currently being used in the market. The following section studies the architectural design of OS required. This paper looks at the related work, how edge computing affects OS design, and future work in this space. This paper surveys key developments in operating systems supporting edge computing, mentioning current OS solutions, challenges in this field, and future directions.

Keywords

Edge, Edge Computing, Operating Systems, Real-time Processing, Resource Management, Distributed Computing, Internet of Things (IoT), Edge layer, Cloud computing, Optimisation, Network, Computing Architecture.

1. Introduction

With development of newer technologies and the exponentially growing IT sector, the current world demands for faster, smarter and an efficient software to handle all operations. With concepts like advanced networking and real-time apps, the need increases for a new software that can keep up with the high-speed requirements. The main aim of developing newer technologies is to make human life easier. This means more automation, less manual work and simple user interaction with software. One of the best ways to achieve this is through real-time applications and Internet of Things (IoT). The growing need for real-time applications such as self-driving cars and IoT devices led to the evolution of edge computing. In contrast to the traditional models of cloud-based architecture, infusing edge computation into our OS will reduce latency and improve the bandwidth utilization. Edge computation works close to data sources rather than working in a separate environment (cloud) hence reducing the time taken to transfer and compute data. Being an interface between hardware and software, an operating system is important for managing the complexities of edge computing environments. As an emerging technology, OS plays a major role in the management and optimisation of edge computing. The edge computing technology demands for various types of responsibilities and requirements which an OS must take car13e of along with doing its regular task. The next

sections focus on the evolution of OS and how it is an integral part for edge computing by laying out its roles and responsibilities, the challenges it faces and how it currently is being used in the market.

2. Overview of Edge Computing

Edge computing involves processing data at the edge of the network instead of relying on centralized cloud servers. “Edge Computing” refers to transferring computing power and intelligence from the central cloud to the network’s edge [1]. In this context, the term “edge” refers to the location where data is collected and processed [2]. Edge computing moves part of the storage and compute resources out of the central data centre and closer to the source of data itself. Rather than transmitting raw data to a central data centre for processing and analysis, that work is instead performed where the data is generated [3]. It consists of a distributed architecture at the edge of the network which, collects data and computes it. The workload is shared by all the nodes present in the architecture. This computed data is then sent to the next resource like a cloud or a database. The objective of edge computing is to process data as close as possible to the network or devices, rather than relying on storage like a cloud as in traditional cloud computing.

Cloud computing refers to the provision of on-demand computing resources, such as networks, servers, storage, applications, and services, accessible over the Internet on a pay-as-you-go basis [4]. These resources are used to process the data and perform operations on it while edge computing refers to the concept “conducting computational tasks at the network edge” [5], rather than using a cloud. Cloud computing enables the connection of resources to form a shared virtual pool, allowing users to procure services that can be rapidly provisioned and released with minimal management effort or service provider interaction [6,7] while on the contrary, Edge computing provides a distributed network system near the ‘network edge’ for data to be processed and transferred further. In simple words, cloud computing collects all data from the network, stores it, performs the required computation or processing and sends it back to the network if required while in edge computing, the processing takes place close to the network before it gets transferred to a storage location and gets sent back if required.

Fig. 1 shows the architecture of edge computing, which consists of three layers - Device layer, Edge layer and Cloud layer. The Device layer consists of all devices or end nodes that collect data and interact with the end user or environment. The Cloud layer is used to host the application in the cloud where all data is stored and retrieved. And the Edge layer is the intermediary layer between the network and the next resources (cloud) in which the computing happens. Edge computing makes use of this edge layer to perform computations. This newly added layer that has been added to the model which is also referred to as the “edge” of the network.

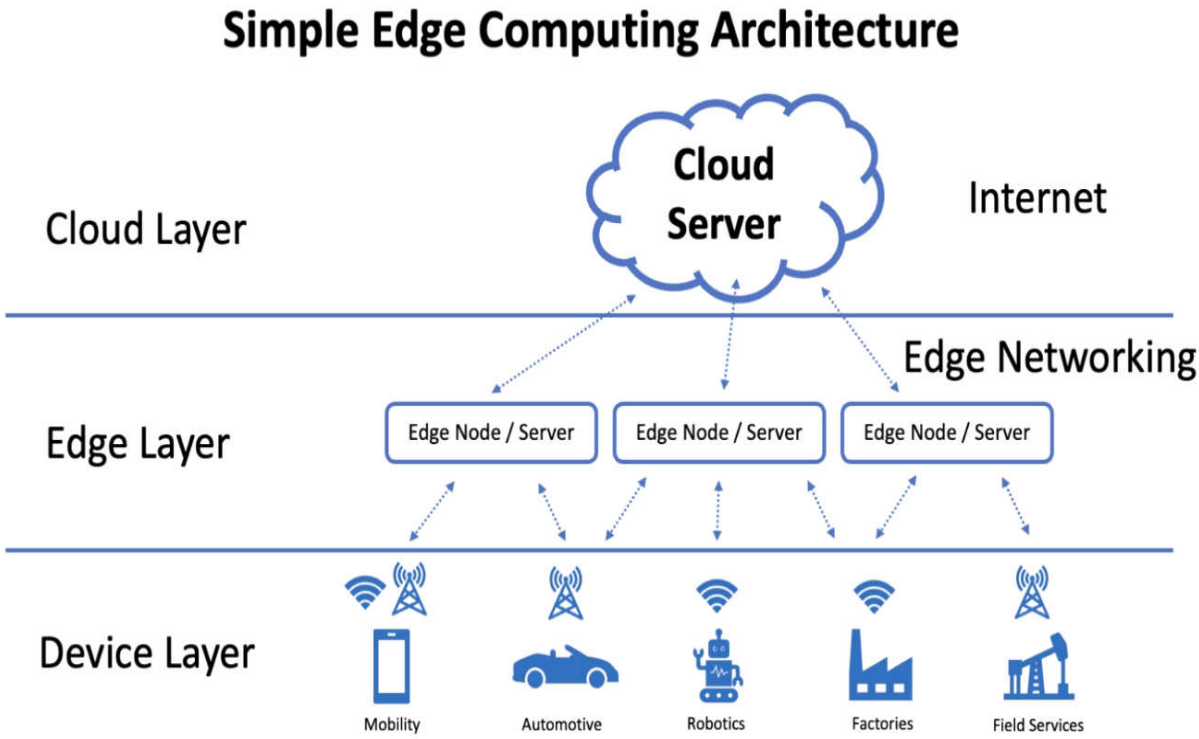

Fig. 1. Edge computing architecture [8]

The addition of the edge layer serves as a very useful component for computational purposes. The advantages of Edge Computing include low latency, as data is being processed closer and thus requires less time for communication, bandwidth reduction, due to the data being processed before being sent to cloud, data will unnecessarily not be transmitted and increased privacy, as data is spread among multiple nodes such as routers, gateways and end devices, sensitive data need not be sent to a remote server for computation; rather, gets computed locally [9-11]. With these advantages serving useful purposes for computation, its applications have a wide range. The applications of edge computing include online gaming for reduced latency, smart cities for reliable and optimal solutions for IoT devices, autonomous vehicles due to the accuracy and speed, AI/ML related applications for processing large data efficiently and many more. To ensure a smooth functioning of a system consisting of this architecture, an efficient Operating System is required. This OS plays a vital role in time management, resource allocation and manage the critical section.

3. Operating Systems in Edge Computing

Operating systems bear the responsibility for the smooth functioning of all the components of the edge layer. Since there are multiple nodes in the edge, it is essential to ensure that all these nodes function as per our requirement and perform the required task efficiently, hence the OS plays a major role in working of edge computing. A traditional OS is built for a single-device and hence cannot be used in edge computing where there is a decentralized

network of nodes. Hence there is a need to figure out a different OS made exclusively for edge computing which can handle all nodes and manage the process. The role of OS in edge computing is very essential. It includes most of the internal working processes of a system like scheduling, allocation and time management.

Resource allocation is the process of allocating and deallocating resources like memory, CPU, I/O ports etc. for processes to use them and run. It must be done in a manner that all processes get the required resource and avoid deadlocks. Traditional resource management techniques for the conventional cloud computing systems will not be able to meet the Quality of Service (QoS) required for the IoT connectivity applications, big data analytics applications, and cognitive computing applications [12]. Hence the next generation cloud computing systems aim to tackle the issue by supporting multiple providers in a decentralized computing paradigm closer to the end users, instead of using a remote, centralized data centers from a single provider. Here, a distributed set of resources are used across various geo-locations close to the network's edge rather than a pool of resources like traditional OS. So rather than all nodes using one set of resources, a resource is allocated to each node based on various factors like priority, requirement, deadline etc. The number of resources and what resource is allocated to which node differs from node to node based on these parameters. There will be many small sized data centers [13-15].

Not only resource allocation, but resource scheduling is also just as important for edge computing to function smoothly. Resource scheduling refers to the set of actions and methodology that participants use to effectively assign resources to the tasks that need to be completed and achieve the objectives of participants based on resource availability. Resource scheduling requires appropriate resource scheduling strategies from three perspectives – user, service provider and edge network [16]. The **user** level consists of many end devices. These end devices can either be static or dynamic and may require to be processed at high speeds with low energy requirements. Hence, good resource scheduling algorithms need to be used for all devices to work as expected. The **service** providers have many resources, hence the main objective of using good scheduling algorithms is to minimise cost and maximise profit. The edge network consists of multiple nodes which are completely distributed over the edge layer. These nodes need to be used optimally, and the workload should not be concentrated on one or few nodes. The distribution of work and optimal utilization of all nodes can be achieved through good scheduling algorithms. For creating an optimal scheduling algorithm, the data from end user side must be collected and the type of data incoming must be known.

There are two types of devices in the end user side – static and dynamic [16]. The static devices include devices like sensors which do not have any changes while processing and have a fixed sets of functionalities and codes. Dynamic devices are those devices which keep moving and do not have a fixed set of functionalities but rather have to keep a real time processing and change their functionality based on factors like environment, input, output etc. These include Unmanned Aerial Vehicle (UAV), autonomous vehicles, robots etc. The static devices can be easily managed by traditional OS, but the dynamic devices present a significant challenge.

Hence, the OS must make sure that it can process data in real time and keep giving feedback to the devices with low latency. With real-time processing data a major issue that comes into picture is the fault tolerance of a system. A distributed system involves multiple devices being distributed over a vast area hence managing all faults is going to be a complex procedure. The OS may encounter faults from multiple sections like hardware, network, security etc. The OS must detect these issues and be able to solve them as quickly as possible as mentioned in the above sections. The fault recovery mechanism should be fast and reliable due to the number of dependent devices, amount of data, speed of computation etc.

The edge layer consists of a huge network of distributed nodes, which function collectively to compute and process data. This layer is very complicated and cannot be managed by a simple OS. An OS is required that can work with multiple devices simultaneously, enable communication between each other and with the devices from another layer. Along with communication, the security aspect of data also should not be compromised, hence the OS must make sure that the data is transferred accurately and safely. A few problems that may arise is the traffic management and hence the OS should also be able to handle variations in network traffic without compromising the working of the devices. Similar to a load balancer in cloud computing, OS must be able to flexibly adjust to the traffic levels and provide a medium of communication without wasting the resources in times of less traffic and overuse of devices during high traffic. In case of failure of a node, the OS should detect it and provide an alternative solution by either removing the node until it is fixed or providing an alternative path for communication. Another major issue that could come up is the delays. The OS should make sure that all devices are synchronized and the latency is reduced. So the OS must make sure that the network functions properly and any anomalies must be detected and accounted.

Edge computing works directly with the user's data and there are high chances this data can be prone to multiple security and privacy concerns. The role of OS here will be to first detect what problems could be encountered, how to prevent it, how to detect and rectify it in the earliest possible time. There are multiple threats like Distributed Denial of Service (DDoS), malicious hardware, unauthorized control access etc. They can be prevented by keeping a set of rules/protocols and keeping a constant monitoring over the components and if any component breaks any rule, action should be taken on it [17]. Setting up authentication and authorization protocols. Introducing a firewall or an anti-intrusion system. Having complex ML and data analysis models to detect any anomalies or give future predictions based on current and past data. Having data backups and recovery methods in case of data loss.

With these roles being played by the OS, it will encounter multiple challenges that will be important to investigate before it is put to function. A major challenge is the distributed resource management. Resource management across heterogeneous nodes is more complex than that in centralized environments because of efficient scheduling and load balancing algorithms [18]. The resources need to be distributed and assigned to each device in the network individually. Another severe issue is the real-time processing requirements. Due to the presence

of real time devices, the OS must get real time data and feed it to the devices. The deadlines are crucial and hence real time operating system is required to meet them. Final and the most important one is security and privacy. Due to the deployment of edge nodes in less secure settings, the operating systems must render the strong security mechanisms to include encryption, authentication, and access control [19]. Despite these challenges, there are multiple operating systems being implemented for edge computing.

Currently, OS for edge computing has its implementations in multiple fields and use cases. Linux-based systems are one of the most common systems deployed at the edge. EdgeX Foundry is an open-source software that provides a common framework for IoT and edge computing which is managed by Linux [20]. It acts as an interface between the user, edge layer and the cloud which enables the data transfer between them. It works with the physical devices and enables data transfer between the cloud and the sensors and helps in performing actions. In simple terms, it is the middleware serving between physical sensing and actuating [20]. Linux being a high-powered OS requires too much power hence lightweight operating systems such as TinyOS and Contiki are designed for low-power, resource-constrained edge devices [21]. These OS use less power but provide good results. The major drawback is the availability of one resource scheduler and it uses FCFS [22] hence causing a less optimal scheduling which could cause delays and increase latency. This OS reduces complexity, makes the overall management easier as compared to regular OS. It also is cost efficient due to the low consumption of power. While these OS serve in the non-domestic fields, there are OS designed for homes, such as EdgeOSH. EdgeOSH stands for Edge Operating System Home which is an operating system designed for edge computing for a smart home. In this OS, the house has multiple sensors attached at every corner which gathers data and sends to the cloud. The data is computed by edge computing and gives real time data. All the devices are connected via the WiFi hence many services can be accessed and performed via your smartphone from a remote area. The lack of a home operating system makes it very difficult to manage devices, data, and services and the solution to this problem is EdgeOSH which is a home operating system for Internet of Things [22]. All devices are connected to this EdgeOSH instead of regular OS which makes the work easy for computing, communication, connection etc. It also enables humans to easily interact directly with the devices making it very user-friendly.

4. OS Architectures for Edge Computing

An appropriate OS architecture is required to ensure optimal resource allocation, time scheduling, real-time data processing, fault recovery, data security, low latency and overall system management. Before implementing an OS, the type of OS existing must be studied and understood to ensure an accurate selection of the architecture. There are three main architectures seen in the design of edge computing OS; these are monolithic, microkernel and RTOS.

4.1 Monolithic Architectures

In monolithic OS designs, all services, file management, networking, and device drivers run in a single large kernel space. This architecture is much easier to implement and brings performance benefits for edge devices, which have ample resources [23]. All processes use the one single large chunk of kernel to perform their actions which makes a centralised manager but it makes it difficult to provide for a distributed resource management. It may cause a single point of failure or a bottleneck.

4.2 Microkernel Architectures

Microkernel architectures separate critical kernel functions, like inter-process communication (IPC), from additional higher-level services running in user space. Only the important functions like communication and resource management take place in the kernel while the smaller services like files, I/O etc takes place in user space. This separation helps in improving modularity and security and makes microkernel OS architectures more suitable for heterogeneous and distributed edge environments [24]. It focuses on minimalistic function that focuses on major tasks. It is implemented as dynamically loaded plugins [25].

The key advantages of microkernel architecture are Dynamic Plugin Loading – Key to the architecture is the ability to load and unload plugins on demand, allowing for the selective activation of features as needed – and In Process Execution – Plugins operate within the same process space, promoting faster execution and efficient communication compared to inter-process communication in traditional microservices architectures. The architectures of the monolithic and microkernel OS designs are presented side by side in Fig. 2.

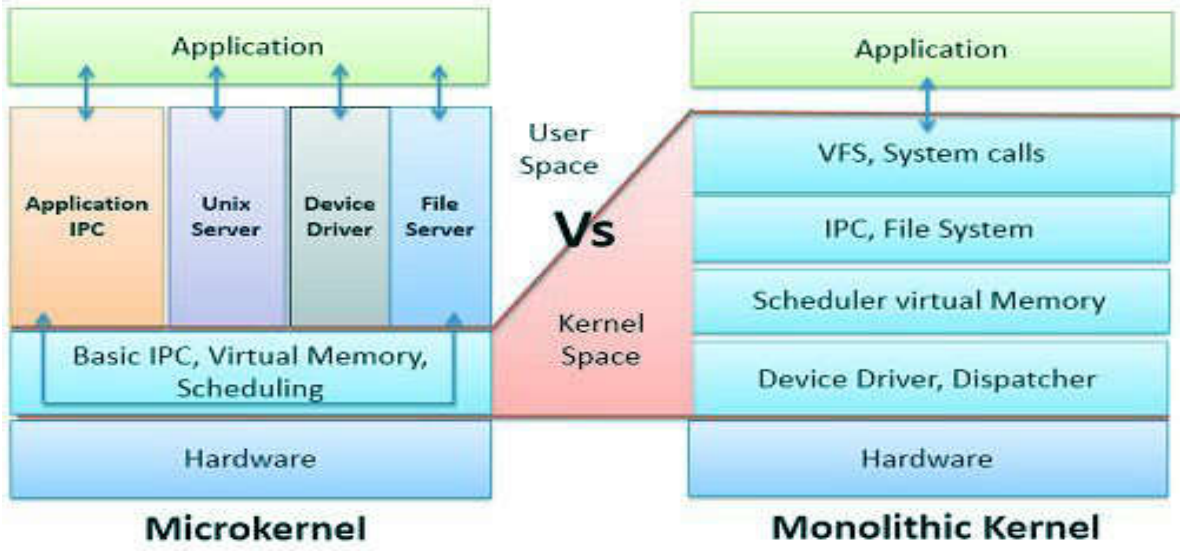

Fig 2. Microkernel vs Monolithic kernel [27]

4.3 RTOS architecture

RTOS stands for Real-Time Operating System, which focuses mainly on finishing deadlines. It manages all tasks keeping the time and deadline as the priority. It ensures all tasks are completed and critical functions are prioritised at the top. It makes optimal and efficient use of all the resources. It also allows real-time analysis of data and ensures that the system is responsive, reliable, and scalable, making it ideal for applications in the Internet of Things (IoT) and other Edge Computing scenarios [26]. It also comes with sets of challenges like resource synchronization and security concerns.

Conclusion

Edge computing is changing how data is processed and requires new approaches in design. This concept has completely modified the way a system is built and has optimised it to reduce latency and bandwidth consumption. Applications of edge computing are extensive, showing the importance of this new trend. OS plays a major role in managing the fundamental activities during edge computing. Choosing the right type of OS is crucial, given that there are so many types, each with a unique application. There are different types of architecture an OS can follow to execute edge computing, each one having its own set of pros and cons.

Future Directions

The OS design in edge computing is still in its immature stage. More research can be done in different types of OS being built to accommodate all the requirements of Edge computing. More optimal solutions can be found for the problems at hand. More efficient systems can be created to manage and handle all the tasks. Along with efficient OS, AI and Edge Computing OS can be merged for a highly yielding system. AI is expected to integrate into edge computing practices. If all complex tasks are handled automatically, the process will be a lot faster and simpler. AI-based algorithms may be able to adjust the dynamic resource management and task scheduling in edge nodes to the potential optimization of OS efficiency. AI can also increase the security due to the automatic encryption and decryption algorithms. While AI is one of the most booming technologies, integration of OS and edge computing with other technologies too can be researched. The boundary of edge computing is limitless hence other technologies like blockchain, communication systems etc. could use this idea to make their work easy. Optimising in other industries will be really easy as edge computing is infused. The boundary of edge computing can reach as high as upto the cloud too. The cloud currently offers IaaS, PaaS, SaaS to its users. It won’t be long before the cloud integrates edge into its services – Edge as a Service - and provides it to the users, making it very accessible and more used everywhere. This will revolutionise the business industry and customers will be benefited. The cost reduction and superfast computing will help gain profits and drive company growth.

References

[1] S. Douch, M. R. Abid, K. Zine-Dine, D. Bouzidi and D. Benhaddou, "Edge Computing Technology Enablers: A Systematic Lecture Study," in IEEE Access, vol. 10, pp. 69264-69302, 2022, doi: 10.1109/ACCESS.2022.3183634.

[2] W. Shi, G. Pallis and Z. Xu, "Edge Computing [Scanning the Issue]", Proceedings of the IEEE, vol. 107, no. 8, pp. 1474-1481, 2019.

[3] Stephen J. Bigelow, 'What is edge computing? Everything you need to know,' TechTarget. [online]. Available: https://www.techtarget.com/searchdatacenter/definition/edge-computing . [Accessed: Oct. 23, 2024].

[4] C. Gurkok, "Securing Cloud Computing Systems" in Computer and Information Security Handbook, Elsevier, pp. 897-922, 2017.

[5] H. Nouhas, A. Belangour and M. Nassar, "Cloud and Edge Computing Architectures: A Survey," 2023 IEEE 11th Conference on Systems, Process & Control (ICSPC), Malacca, Malaysia, 2023, pp. 210-215, doi: 10.1109/ICSPC59664.2023.10420123.

[6] M. Arvindhan, "Effective motivational factors and comprehensive study of information security and policy challenges" in System Assurances, Elsevier, pp. 531-545, 2022.

[7] A. Oludele, E. C. Ogu, K. Shade and U. Chinecherem, "On the Evolution of Virtualization and Cloud Computing: A Review", Journal of Computer Sciences and Applications, vol. 2, no. 3, pp. 40-43, Nov. 2014.

[8] Nicholas Holian, "Edge computing architecture," Edge computing understand the user experience. [online]. Available: https://www.wipro.com/infrastructure/edge-computing-understanding-the-user-experience/ . [Accessed: Oct. 23, 2024].

[9] X. Xu, C. Zhang, X. Wang, and L. Qi, "Edge computing architecture and practice: Resource, network, and application," IEEE Access, vol. 8, pp. 182102–182112, 2020.

[10] S. S. Gill, S. Tuli, M. Xu, M. R. Shabaz, P. Tiwari, P. Garraghan, and R. Buyya, "Transformative effects of IoT, Blockchain and artificial intelligence on cloud computing: Evolution, vision, trends and open challenges," Internet of Things, vol. 8, pp. 1–20, 2019.

[11] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," Internet of Things, vol. 1, pp. 100–130, 2018.

[12] R. A. Shehab, M. Taher and H. K. Mohamed, "Resource Management Challenges in the Next Generation Cloud based Systems: A Survey and Research Directions," 2018 13th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 2018, pp. 139-144, doi: 10.1109/ICCES.2018.8639220.

[13] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge computing: Vision and challenges", IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637-646, Oct 2016, ISSN 2327-4662.

[14] J. Pan and J. McElhannon, "Future edge cloud and edge computing for internet of things applications", IEEE Internet of Things Journal, vol. 5, no. 1, pp. 439-449, Feb 2018.

[15] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities", IEEE Internet of Things Journal, vol. 3, no. 6, pp. 854-864, Dec 2016, ISSN 2327-4662.

[16] Quyuan Luo et al., “Resource Scheduling in Edge Computing: A Survey”, IEEE Communications Surveys & Tutorials, vol. 23, no. 4, pp. 2131 – 2165, Aug 2021, ISSN 1553-877X.

[17] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider and M. Hamdi, "A Survey on Security and Privacy Issues in Edge-Computing-Assisted Internet of Things," in IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4004-4022, March 2021, doi: 10.1109/JIOT.2020.3015432.

[19] S. K. Sood and S. Singh, "A secure and efficient authentication framework for edge devices in the Internet of Things," IEEE Transactions on Information Forensics and Security, vol. 14, no. 3, pp. 606–616, 2019.

[20] A. Ramachandran and S. Rajesh, "EdgeX Foundry: An open-source framework for IoT edge computing," IEEE Internet of Things Journal, vol. 5, no. 6, pp. 4625–4632, 2018.

[21] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," Computer Networks, vol. 52, no. 12, pp. 2292-2330, 2008.

[22] A. Patil, R. M. Jagadish and S. Janthakal, "The TinyOS operating system for wireless sensor networks with a hybrid scheduler," 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON), Bengaluru, India, 2024, pp. 1-6, doi: 10.1109/NMITCON62075.2024.10699112

[23] A. S. Tanenbaum and H. Bos, Modern Operating Systems. Pearson, 2014.

[24] L. Liu, Z. Zhou, and S. Chen, "Microkernel-based edge computing architecture for trusted IoT," Proceedings of the IEEE International Conference on Edge Computing (EDGE), 2019, pp. 222-229.

[25] K. Nandy, S. SM, A. Bhadauria and S. Upadhyay, "Resource Optimization in Edge Through Microkernel Architecture," 2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C), Hyderabad, India, 2024, pp. 362-367, doi: 10.1109/ICSA-C63560.2024.00065.

[26] Lance Harvie, ‘Optimizing Real-Time Operating Systems for Efficient Edge Devices’, Embedded Related. [online]. Available: https://www.embeddedrelated.com/showarticle/1621.php#:~:text=An%20RTOS%20is%20a%20crucial,and%20other%20Edge%20Computing%20scenarios. [Accessed: Oct. 25, 2024].

[27] “Difference Between Microkernel and Monolithic Kernel,” Tech Differences. (Online). Available: https://techdifferences.com/difference-between-microkernel-and-monolithic-kernel.html. [Accessed: Jul. 14, 2025].