Edge Computing as an Architectural Solution: An Umbrella Review

Ajay Bandi[1] and Julio Ariel Hurtado[2]

[1] Northwest Missouri State University, Maryville MO 64468, USA
[2] Universida del Cauca, Popayán - Colombia
ajay@nwmissouri.edu
ahurtado@unicauca.edu.co

Abstract. Cloud computing architecture and cloud service applications follow a centralized architecture with bottlenecks in the cloud infrastructure. This infrastructure is significantly affected when services respond to many heterogeneous end devices because of the limitations of bandwidth and the servers' workload; consequently, it introduces a high latency. The advantages of using content delivery networks are to speed up web performance by caching web content on edge nodes near the user. However, there are challenges with streaming data. Researchers create an intermediary infrastructure to store, secure, and compute end devices' services became a new concept called edge computing. Edge computing can leverage applications that are sensitive to latency. However, other issues appear, such as security and deployability. This paper reviewed the literature to analyze edge computing as an architectural solution and identify the underlying architectural quality attributes, tactics, and strategies. The performance quality attribute drives the edge architecture, mainly to reduce the latency and jitter concerns. The quality requirements are addressed by caching, migration, and virtualization strategies. However, the solution introduces other quality attribute concerns such as security, deployment, and scalability. This paper is a first approach for unveiling the rationale behind edge computation from an architectural viewpoint.

Keywords: edge computing · fog computing · multi-access edge computing · software architecture · data streaming

1 Introduction

In recent days, due to massive increase in the usage of mobile devices around the world. Several businesses rely on mobile applications to serve their customers and develop free apps. Mobile applications are prevalent in dating, e-commerce, education, medical, health care, recreation, transportation, social media, research, entertainment, mission-critical systems among others. The emergence for connecting different objects through the internet, termed as the Internet of Things (IoT), allows the machine to machine communication with embedded sensors. These devices collect vast amounts of data for appropriate decision making and reduce the workload on the automation. Therefore, the exponential growth of data needed to be stored and retrieved efficiently [3].

These applications are heterogeneous and require diverse resources. Cloud computing is a desirable solution with advanced computing and communication network technologies [6]. Cloud computing is a model for gathering physically distributed resources such as processors, memory, bandwidth, and storage capacity to deliver on-demand services to users [6], [15]. A cloud can provide infrastructure, platform, and software as a service to users. Over the past decade, cloud computing deals with large-scale storing and computing data in data centers. These data centers usually connect with other data centers to form a data center network and provide end-user services as a single resource. However, due to an increase in smart devices, massive growth using IoT applications and augmented virtual reality requires real-time and quick responses based on context-awareness and location data [6]. Applications that use streaming data [2] has the high round-trip transmission time to and from the cloud. Examples of streaming data are sensors that collect the continuous data of industrial equipment, oversee the pieces of equipment's performance, identify the defects in advance, and automatically order a spare part. A dating app tracks the users' geographic location and provides partners' recommendations based on their profile. Similarly, online gaming apps track player-game interactions and provide relevant promotional offers to the player.

Latency is the delay between the users' request and the applications' response to that request, usually measured in milliseconds. Bandwidth is the amount of data transmitted through the network at a given time. Throughput is the amount of the data transferred over a certain period. Both bandwidth and throughput are measured in bits/second. For efficient network communication, the latency must be low, and bandwidth should be high. Even though the communication within the data center network is efficient with low latency, communication between the cloud and end-users is challenging. Also, it lacks the context-awareness and location-awareness of the users. The Business Insider website [17] estimated that the usage of IoT devices would be more than 41 billion by 2027, up from about 8 billion in 2019. Thus increase the need for a 5G network [10] to companies to transmit the big data generated by these devices to reduce the network traffic. Due to these challenges, the communication between the cloud and end-users degrades the quality of the service and experience.

The communications industry introduces [10], [24] edge computing to overcome the challenges of centralized cloud computing. Satayanarayanan et al. [20], [21] defined edge computing as computing and storing resources at the internet's edge near IoT devices. These resources are referred to as cloudlets and fog nodes (microdata centers). The internet's edge would decentralize the storing and processing of data from the cloud and add a middle tier between end devices and the original cloud datacenter. Researchers published literature reviews [1, 11], [13], [14], [5], [18], [23], [25], in machine learning, deep learning, IoT apps, security, and communication integrating with edge computing. In this paper, we presented an umbrella review of edge computing from an architectural perspective.

System architects operate with both tactics and patterns for making decisions addressing certain quality concerns. The architecture pattern is a well-organized strategy representing the system's high-level structure and behavior for its requirements. Similarly, a tactic is a design decision or specific ideas to accomplish the quality attributes. A group of tactics could be organized as an architectural strategy or pattern. In this paper, we draw the quality attributes of edge computing to architectural tactics and strategies. Edge computing is a decentralized architecture with robust storage and computing resources at the internet edge nearer to the end devices. Edge computing is introduced mainly to achieve quality attributes such as performance (latency, bandwidth, handling concurrent requests, accuracy), along with scalability, deployability, security, and portability. Because edge computing as architecture solutions lacks of a unified knowledge about the underlying rationale necessary for analyzing and developing edge applications, servers and services [19], we integrate the architecture patterns and tactics [8] related with edge computing in order to codify knowledge for supporting making decisions during architectural design.

The organization of the remainder of this paper is as follows. Section 2 explains edge computing implementations and architecture tactics. Section 3 illustrates the umbrella review process. Section 4 presents the detailed analysis of the results and conclusions in the final section.

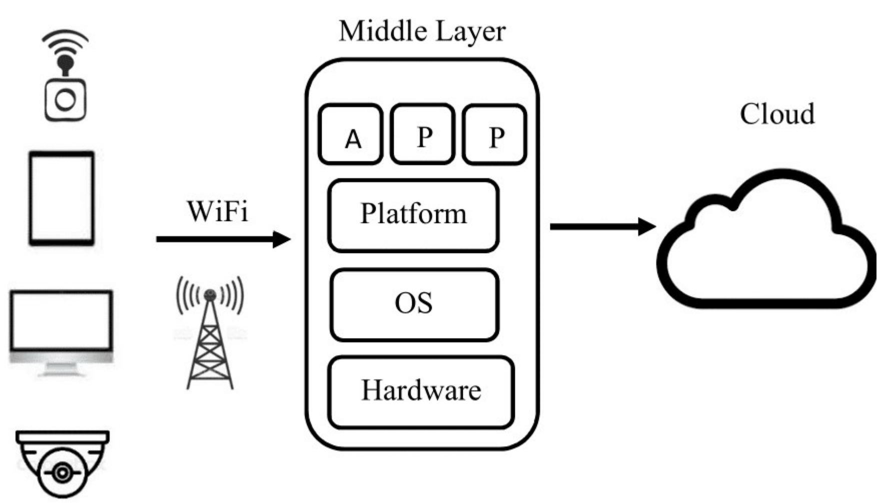

Fig. 1. Cloudlet

2 Related Work

This section explains the edge computing implementations of cloudlets, fog computing, and multi-access edge computing. We are discussed the background of integrating edge computing with architecture patterns and tactics.

2.1 Cloudlet

A cloudlet is a decentralized, datacenter in a box closer to the end devices connected via the internet. Cloudlets add a middle tier to the cloud and mobile devices and form a three-tier architecture to provide the computing and storage resources to the end devices within one wireless hop [20], [21]. Cloudlets are rich in resources and focus on latency-sensitive data and services that require high bandwidth sharing between a few users. These cloudlets are self-managed, and professional attention is not required. A cloudlet has a soft state, which means it has cache data or the data stored in another place. An end mobile device acts as a client to the nearby cloudlet for fast end-to-end response time. If there is no cloudlet within the closer proximity, the mobile device may degrade and use cloud services and gain the performance after identifying another cloudlet. However, cloudlet placement is challenging [20]. Zhao et al. [32] proposed a ranking based near-optimal algorithm placing a cloudlet to minimize the access delay of IoT applications in a software-defined network (SDN). This ranking based algorithm outperforms the K-median clustering algorithm in the access delay.

The usage of cloudlets provides low latency, high bandwidth, and low jitter responsive end to end services for the nearest mobile devices [20]. The applications using AR/VR, speech recognition, machine learning, computer vision, natural language processing offloads the intensive computation and storage resources to the cloudlet. The cloudlets help perform essential services during the failure of clouds due to cyber-attacks, network jamming in cyber wars, and physical destruction of the network infrastructure due to natural disasters [20].

2.2 Fog Computing

Fog computing is a decentralized, geographically distributed computing with fog nodes between the end devices and the cloud. A fog node is powerful in storage resources and processing capability with any equipment including but not limited to routers to base stations, switches, and access points to IoT devices. Fog nodes are multiple data centers distributed geographically with the closer proximity of end devices to offload services at the edge. For a given business, the number of fog nodes varies from ten to hundreds. Yi et al. [29] proposed a three-layered fog node with infrastructure, platforms, and applications with appropriate Application Program Interfaces (API).

Fog computing helps analyze the latency-sensitive data at the edge rather than sending it to the cloud to achieve comprehensive data privacy and security with low operational costs. Cisco introduced the first fog node called IoX [24] to host multiple applications and analyze the data generated by various end devices.

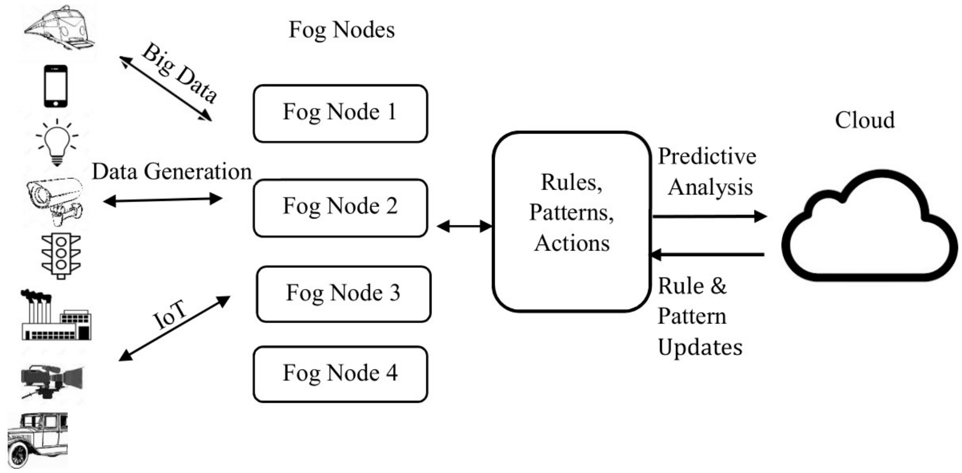

Fig. 2. Fog edge computing

Cisco IoX fog node is heterogeneous with the business networking operating system and Linux [24]. Figure 2 shows the CISCO's fog architecture.

2.3 Multi-Access Edge Computing

Edge computing in mobile devices and networks was referred to as Mobile Edge Computing (MEC) by the European Telecommunications Standards Institute (ETSI) [10]. MEC induces traffic and services' computing from a centralized cloud to the edge of the network and closer to the customer. By moving the data-intensive storage and computation power nearer to the edge, network operators can overcome the network traffic challenges. This decentralized architecture aggregates communication technologies for IoT devices and 5G systems to IT services to the end devices. In 2016, ETSI dropped the word mobile from MEC and renamed it as Multi-access edge computing, [16] also abbreviated as MEC.

MEC has many use case scenarios such as vehicle to any device (V2X) communication, healthcare, retail business, AR/VR, accelerated video, caching services, and IoT applications. MEC provides services to various stakeholders, including Over-The-Top (OTT) players, software vendors, mobile network operators, etc [25]. With the massive and exponential growth in IoT applications and the data generated by mobile devices, Verizon uses MEC and 5G as critical technologies for edge computing. With MEC's help, users of mobile end devices can effectively use the computing and storage resources. Massive bandwidth, reduced latency, reduced computing power on the device, and minimization of the network traffic are the advantages of the MEC. Figure 3 is the MEC architecture of Verizon's edge computing.

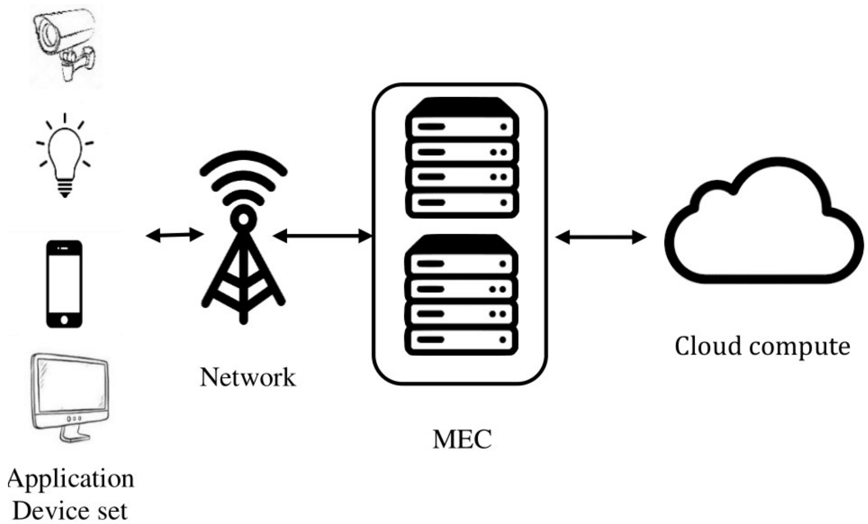

Fig. 3. Multi-access edge computing

2.4 Architecture patterns and tactics in edge computing

Khan et al. [11] discuss the significance of edge computing in real-world scenarios. They presented a new taxonomy of edge computing for cloudlets, fog nodes, and MEC. The work compares 23 primary studies to reduce latency, maximize resource utilization, ensure optimized privacy, strengthen security, and collect real-time data insights. Due to the increase in the need for 5G networks and MEC, Mao et al. [14] in their literature review focused on joint and radio computational resource management and Mach et al. [13] on computation offloading in MEC. Ni et al., [18], in their literature review, focused on securing fog nodes for IoT applications. Taleb et al. [25] focused on the 5G networks, orchestration, and deployment of edge computing. Chen and Ran [5] presented of a detailed and exhaustive deep learning techniques review to improve edge computing performance. Sittón-Candanedo and Casado-Vara [23] discussed the computing consortium reference architectures and presented a proposal for tier architecture. In addition, several researchers [1, 5, 11, 13, 14, 18, 23, 25, 31] discussed open research problems, opportunities, and challenges in edge computing. However, these studies lack the knowledge from the architecture perspective of the edge computing systems. Architecture knowledge [8] constitutes quality attributes, tactics, and strategies. We focus on integrating edge computing with architecture knowledge proposed by Harrison and Avgeriou [8].

- *Quality attributes* are features that the system has, such as reliability, security (authenticity, confidentiality, data integrity), scalability, deployability, usability, maintainability, and performance. Quality attributes satisfaction is always viewed within the specific scenario where there is a required output to reach within particular boundaries of a system facing an input [4].

- *A tactic* is a design decision or the sequence of actions to accomplish a design concern. Tactics are steps taken to enhance quality characteristics [4]; for instance, a tactic is defined as common and abstract services to improve the maintainability.
- *A pattern* accumulates several tactics that impact quality attributes either positively or negatively. Architecture patterns commonly describe the decomposition of the system's modules at a high and abstract level [8]. Layers, tiers, publish/subscribe are examples of architectural patterns.

3 Methodology

Our study follows a theoretical review [26] focusing on a model based on the conceptual model of Harrison and Avgeriou [8], which includes: quality attributes, tactics, and architectural patterns (strategies). Given this knowledge is scattered in the literature, this paper aims to organize the architectural level to edge computing as an emergent technology.

3.1 Research goal and research questions

The main goal is to unveil architectural concerns from secondary studies (surveys, reviews, or systematic reviews) on edge computing solutions. The main research questions are oriented to identify architectural concerns:

- What quality attributes and related design concerns are analyzed in the edge computing solutions?
- What tactics are unveiled from the edge computing solutions?
- What strategies are used as part of the edge computing solutions?

3.2 Research execution

First, the primary studies selected from relevant sources were used IEEE, Springer, Elsevier, and Scopus. The research string used and adapted to each repository was: "edge computing" AND architectur* AND (tactic OR "quality attribute" OR pattern) AND (survey or review). Once the papers were recovered (17), the abstract of each one was read, and duplicated works were removed. The papers were filtered, getting nine papers after using the inclusion and exclusion criteria. These nine studies are annotated as "[Primary Study]" in the references.

- Inclusion criteria: secondary studies, edge computing, works analyzing some architectural concerns such as quality attributes and architectural tactics, patterns, or strategies.
- Exclusion criteria: primary studies, proposal, architectural ideas without any real assessments.

Finally, we analyzed the nine papers to answer each research question, data collected were analyzed and the results were organized and analyzed. Each architectural aspect was evaluated considering the knowledge body conformed by Bass et al. [4], Harrison and Avgeriou [8], and Osses et al. [19].

4 Results and analysis

This section presents our studies' results and analysis, such as quality attributes, architectural tactics, and strategies.

4.1 Quality Attributes in edge computing

The first question related to quality attributes in edge computing shows that the main driver is the performance. The main concern is to resolve the delay in transmission and latency problems, particularly for latency-sensitive applications. Edge computing takes advantage of cloud computing but physically closer it to the end-user to achieve a quicker than cloud concerning the response time [11], [20]. However, adding an intermediary element between the devices and the cloud has consequences to other quality attributes. Thus, edge computing has other limitations for availability, security, and resource management regarding mobility, context awareness, and location awareness. Khan et al. [11] discussed

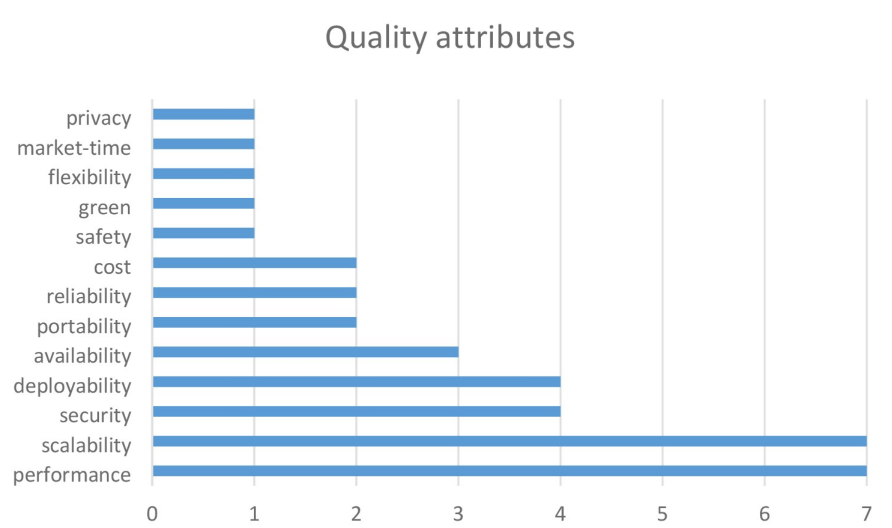

Fig. 4. Quality Attributes Identified

the key specifications required to use edge computing to address these limitations. Techniques such as dynamic billing mechanism of multi-vendor systems with network parameters such as latency, bandwidth, real-time application support, resource management, the common business design for deployment and management, scalability to accommodate various IoT applications execute consistently on the heterogeneous resources, anticipate robust and resilient edge computing systems over network outrages, and security with advanced cryptography schemes. [11]. The increased frequency of quality attributes taken into account in the works is shown in Figure 4.

Other frequent quality attributes in edge computing are scalability and deployability. It is essential to consider designing a scalable architecture to accommodate various IoT applications to execute consistently using heterogeneous resources. The resource virtualization, trust enabled technologies, and edge orchestration can achieve scalability [11]. The usage of containers over virtual machines in MEC allows portable run time services for mobile users [25]. Also, containers provide techniques for quick and secure packaging and deployment to various apps across the platform [25].

4.2 Architectural Tactics in edge computing

The more frequently used tactics in the literature were caching, resource management, resource allocation, and temporary storage, as shown in Figure 5. For instance, in a MEC, one fundamental problem is how to balance the trade-off between the massive database and finite storage capacity when data caching is realized by a single edge server, so temporary or transient storage appears as a general design decision.The allocation of computing resources is significantly associated with the offloaded application that allows parallelization or partitioning distributed to various computing nodes. Edge computing can offload part of the workload of the cloud. Traditionally, only the data is cached in intermediate servers, but the computations applied to the data are cached at the edge servers in the edge computing paradigm.

Fig. 5. Tactics Identified

4.3 Architectural Strategies in edge computing

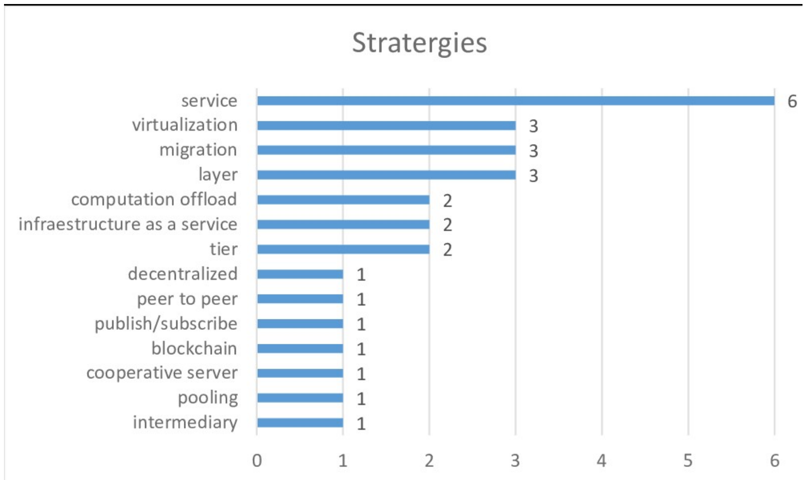

Fig. 6. Strategies Identified

The most common strategies found were services (software as a service, infrastructure as a service, security as a service, etc.), virtualization, process migration, intermediary, tiers, and layers. Computations as services is a strategy for migrating and executing computations offered by a well-defined service layer, an essential strategy for achieving the advantages of offloading tactic. The desired strategy was using an intermediary since edge self works as an intermediary between local devices and the cloud. This intermediary is designed typically using a tiered- architecture: data center tier(cloud), large/medium edge tier, small edge tier (for instance cloud-let), near layer(for instance, fog), and the device layer. Dolui et al. [7] presents a three-tiered architecture, including an upper-tier, intermediary tier, and low tier.

4.4 Tactics achieving quality attributes in edge computing

Tactics are abstract architectural decisions for resolving a specific quality attribute. Table 1 shows as the identified tactics allow to achieve some quality attributes. The plus sign represents the tactic that positively affects the quality attribute. The minus sign represents the corresponding tactic that negatively affects the quality attribute. The number of plus or minus varies the intensity of that particular tactic to achieve the quality attribute. The symbol 'o' represents that a particular tactic is neither positively nor negatively affect the quality attribute. Consider the *Offloading computations* tactic scenario applying computation offload from a mobile device to a proxy [12]. When the end-user runs

Table 1. Quality Attribute versus Tactics

Attributes vs Tactics	Caching	Resource management	Resource allocation	Transient storage	Computation Offloading
Performance	+++	++	++	++	+/-
Scalability	+	++	+++	++	-
Deployability	o	-	-	−	+++
Security	−	o	o	-	−
Availability	+++	++	++	+	++

the search-enabled applications on their mobile devices, the mobile application offloads the nearest proxy's computation. Figure 7 shows the essential components of the offloading tactic with a sequence of operations. Offloading benefits are less disruption to the end-user and reduce delays and improve user experience quality. Offloading has limitations as data and computations are cloned across different processors and heterogeneous storage devices; it is a security and privacy risk [28].

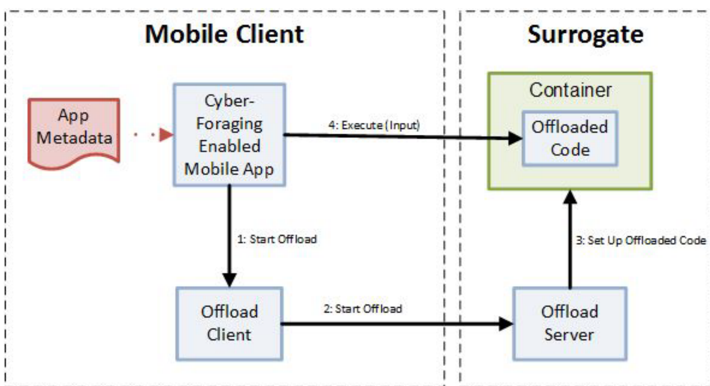

Fig. 7. Offloading Tactic taken from [12]

4.5 Strategies grouping tactics in edge computing

An architectural strategy can embody several tactics. Table 2 shows the identified strategies and some identified tactics; it is not a complete view about tactics and strategies related to edge computing, but the intention is that those can be taken as a start point for organizing the architecture knowledge.

For instance, the migration strategy packages the computation offloading and virtualization tactics. Yousafzai et al. [30] established a case study for processing computational offloading for IoT mobile edge computing. They developed a

Table 2. Tactics verses strategies

Tactics/Strategies	Services	Migration	Virtualization	Intermediary	Layered	Tiered
Caching	x			x	x	x
Resource management	x		x			x
Resource allocation	x		x	x		x
Transient storage		x		x		
Computation offloading		x	x			

framework for seamlessly migrating a mobile device with limited resources towards a computing infrastructure with more available resources. The framework includes components at the user side (app migration coordinator and migration preference manager) and components at the edge side (migration manager and admission control), as depicted in Figure 8. Yousafzai et al. [30] assess the performance of this framework using standard benchmarks and specific workloads considering eight distinct intensity levels in order to find some association between time, workload, and power savings. This showed a significant and positive impact on the framework performance.

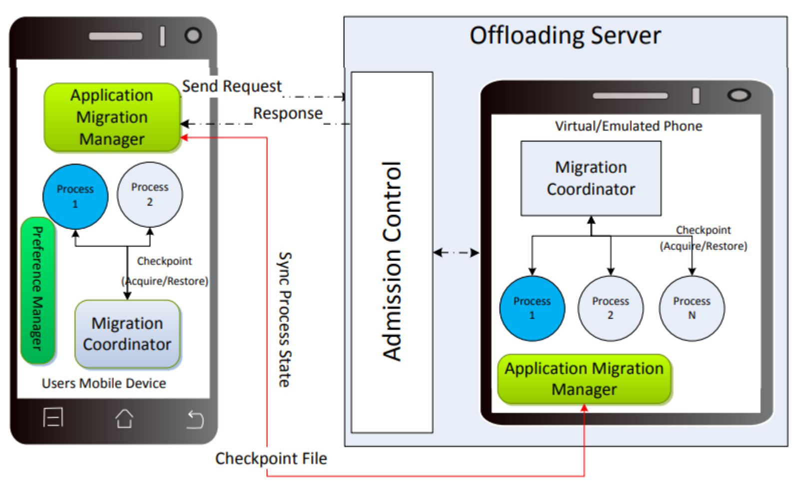

Fig. 8. Migration Strategy Proposed by Yousafzai et al. [30]

Computational offloading is an architectural strategy described by Sheng et al. [22], for answering questions such as: when a specific computation requires to be offloaded?, especially considering several closed servers satisfying the offloading conditions; How to select the server for executing the computation, including the virtual machine to execute the computation once the server had been executed. To address these questions, the authors propose a computation offloading strategy breaking into three steps:

1. Offloading decision-making is about comparing the local computation with the offloading computation. The offloading choice is in accordance with the computation concerns like performance and energy consumption.
2. Selecting an adequate server by balancing concerns such as performance, energy consumption, and server CPU resources.
3. Schedule and execute the computations on the suitable virtual machine so that the MEC servers can perform more computation at the same time interval.

4.6 Synthesis and Discussion

We used Harrison's conceptual framework [8] to synthesize and discuss edge computing as an architecture solution following a pattern style for its description.

1. *Name:* Edge Computing as Architectural Solution
2. *Problem:* Cloud computing architecture and cloud service applications follow a centralized architecture with bottlenecks in the cloud infrastructure. This infrastructure is significantly affected when services respond to many heterogeneous end devices because of the limitations of bandwidth and the servers' workload; consequently, it introduces a high latency.
3. *General Solution:* Edge computing brings computation and storage resources to the nearer location to intensify response times and save bandwidth. Edge computing architecture is a service-based and n-tiered solution, typically identified as an intermediary tier composed of edges nodes (edge tier) and defining specifics capabilities for each tier, including the upper tier (cloud layer), the edge tier, and lower-tier (edge devices tier).
4. *Participants:*
   (a) Cloud tier: A centralized data center and cloud computing infrastructure considering requirements for working with the edge tier (caching, computation migration, virtualization).
   (b) Edge tier: This tier contains devices that could be used as edge servers as general-purpose servers, platforms specifically deployed the satisfy edge needs (caching, computation migration, virtualization) or specific domains (traffic, for example)
   (c) Device tier: This tier contains every end device that generates massive data while executing on the end-user application. The edge devices include IoT apps on vehicles, computers, and mobile phones.
5. *Tactics and strategies used*
   (a) Resource management and provisioning: service and provisioning (on-demand resources and migration) and placement (VM and services)
   (b) Computations offloading and task scheduling

   (c) Security and infrastructure as a service (prevents attacks and on-demand redundancy resources)
6. *Consequences:*
   (a) Ultra-low latency closing services to end-users and the smart devices, and migrating the computations to more capable virtual machines.
   (b) Edge computing requires to minimize the service downtime and guarantee high availability of edge services.
   (c) New security issues must be addressed, such as vulnerabilities related to authorization and authentication, distributed denial of service attacks, malware injection attacks, and side-channel attacks.
   (d) Scalability issues because increasing the capacity to an existing edge require increasing the inter-edge working bandwidth to avoid congestion and reduce the system capacity.
7. *Known uses:* Edge Computing Reference Architecture 2.0 from Edge Computing Consortium (ECC) and Alliance of Industrial Internet (AII). Nokia launched its own MEC platform based on the Cellular Vehicle-to-Everything (C-V2X) technology protocol to achieve a latency below 20 ms, along with a flexible deployment model [9]. IMS and Dell deployed an IoT-connected edge computing platform to obtain and maintain data from their refrigeration systems and build management systems to automatically adjust the temperature of the food, maintaining the quality standards, lessen waste and optimize refrigeration costs [27].

5 Conclusions, Limitations, and Future Work

Edge Computing is a novel technology moving the services from the cloud to the most closer device. Edge computing defines a boundary between the cloud and the device tiers to resolve services' latency problems. This paper presented, preliminarily, that edge computing is an architectural solution. We used a small set of secondary studies to extract and analyze the architectural information related to quality attributes, architectural tactics, and edge computing strategies. The main contribution is codifying edge computing knowledge as an architectural solution for software architects' decisions. The problems can be viewed as a set of tactics and strategies and the positive and negative consequences on different quality attributes. The main key quality attribute addressed by edge architectural solution is the performance, mainly the latency concern addressed by caching, migration, and virtualization strategies. However, it introduces requirements on other quality attribute concerns such as security, deployment, and scalability. This edge tier implements the strategies but drag requirements on the other traditional tiers; for instance, offloading computations require that services communicate the network among edge servers, edge devices. Although this paper unveils the rationale behind edge computing from the architectural viewpoint, in the future, more extended studies are needed to achieve a more in-depth knowledge of edge computing, architectural quality attributes' tactics, and strategies.

References

1. Ahmed, E., Rehmani, M.H.: Mobile edge computing: opportunities, solutions, and challenges (2017), [Primary Study]
2. Bandi, A.: Data streaming architecture for visualizing cryptocurrency temporal data. In: Proceedings of 3rd International Conference on Computer Networks, Big Data and IoT (ICCBI 2020). Springer (2020)
3. Bandi, A., Fellah, A.: Design issues for converting websites to mobile sites and apps: A case study. In: 2017 International Conference on Computing Methodologies and Communication (ICCMC). pp. 652–656 (2017). https://doi.org/10.1109/ICCMC.2017.8282547
4. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. Addison-Wesley Professional, 3rd edn. (2012)
5. Chen, J., Ran, X.: Deep learning with edge computing: A review. Proceedings of the IEEE **107**(8), 1655–1674 (2019), [Primary Study]
6. Dolui, K., Datta, S.K.: Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. In: 2017 Global Internet of Things Summit (GIoTS). pp. 1–6 (2017)
7. Dolui, K., Datta, S.K.: Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. In: 2017 Global Internet of Things Summit (GIoTS). pp. 1–6 (2017)
8. Harrison, N.B., Avgeriou, P.: How do architecture patterns and tactics interact? a model and annotation. Journal of Systems and Software **83**(10), 1735 – 1758 (2010). https://doi.org/https://doi.org/10.1016/j.jss.2010.04.067
9. Hilt, V., Sparks, K.: Future edge clouds. Bell Labs Technical Journal **24**, 1–17 (2019)
10. Hu, Y.C., Patel, M., Sabella, D., Sprecher, N., Young, V.: Mobile edge computing—a key technology towards 5g. ETSI white paper **11**(11), 1–16 (2015)
11. Khan, W.Z., Ahmed, E., Hakak, S., Yaqoob, I., Ahmed, A.: Edge computing: A survey. Future Generation Computer Systems **97**, 219 – 235 (2019). https://doi.org/https://doi.org/10.1016/j.future.2019.02.050, [Primary Study]
12. Lewis, G., Lago, P.: A catalog of architectural tactics for cyber-foraging. In: 2015 11th International ACM SIGSOFT Conference on Quality of Software Architectures (QoSA). pp. 53–62 (2015)
13. Mach, P., Becvar, Z.: Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys & Tutorials **19**(3), 1628–1656 (2017), [Primary Study]
14. Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K.B.: A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials **19**(4), 2322–2358 (2017), [Primary Study]
15. Marapareddy, R., Bandi, A., Tirumala, S.S.: Cloud computing architectures: A retrospective study. Journal of Innovation in Computer Science and Engineering **2**(1), 1–5 (2012)
16. Morris, I.: Etsi drops "mobile" from mec. Light Reading (2016)
17. Newman, P.: The internet of things 2020: Here's what over 400 iot decision-makers say about the future of enterprise connectivity and how iot companies can use it to grow revenue, https://www.businessinsider.com/internet-of-things-report?r=USIR=T
18. Ni, J., Zhang, K., Lin, X., Shen, X.S.: Securing fog computing for internet of things applications: Challenges and solutions. IEEE Communications Surveys & Tutorials **20**(1), 601–628 (2017), [Primary Study]

19. Osses, F., Márquez, G., Astudillo, H.: Exploration of academic and industrial evidence about architectural tactics and patterns in microservices. In: Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings. p. 256–257. ICSE '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3183440.3194958
20. Satyanarayanan, M.: The emergence of edge computing. Computer **50**(1), 30–39 (2017), [Primary Study]
21. Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The case for vm-based cloudlets in mobile computing. IEEE pervasive Computing **8**(4), 14–23 (2009)
22. Sheng, J., Hu, J., Teng, X., Wang, B., Pan, X.: Computation offloading strategy in mobile edge computing. Inf. **10**(6), 191 (2019). https://doi.org/10.3390/info10060191
23. Sittón-Candanedo, I., Alonso, R.S., Corchado, J.M., Rodríguez-González, S., Casado-Vara, R.: A review of edge computing reference architectures and a new global edge proposal. Future Generation Computer Systems **99**, 278–294 (2019), [Primary Study]
24. Solutions, C.F.C.: Unleash the power of the internet of things. Cisco Systems Inc (2015)
25. Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., Sabella, D.: On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. IEEE Communications Surveys & Tutorials **19**(3), 1657–1681 (2017), [Primary Study]
26. Turner, J.R., Baker, R., Kellner, F.: Theoretical literature review: Tracing the life cycle of a theory and its verified and falsified statements. Human Resource Development Review **17**(1), 34–61 (2018). https://doi.org/10.1177/1534484317749680
27. Ud Din, I., Guizani, M., Hassan, S., Kim, B., Khurram Khan, M., Atiquzzaman, M., Ahmed, S.H.: The internet of things: A review of enabled technologies and future challenges. IEEE Access **7**, 7606–7640 (2019)
28. Wang, F., Diao, B., Sun, T., Xu, Y.: Data security and privacy challenges of computing offloading in fins. IEEE Network **34**(2), 14–20 (2020)
29. Yi, S., Hao, Z., Qin, Z., Li, Q.: Fog computing: Platform and applications. In: 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb). pp. 73–78. IEEE (2015)
30. Yousafzai, A., Yaqoob, I., Imran, M., Gani, A., Noor, R.M.: Process migration-based computational offloading framework for iot-supported mobile edge/cloud computing. IEEE Internet Things J. **7**(5), 4171–4182 (2020). https://doi.org/10.1109/JIOT.2019.2943176
31. Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J., Jue, J.P.: All one needs to know about fog computing and related edge computing paradigms: A complete survey. Journal of Systems Architecture **98**, 289–330 (2019)
32. Zhao, L., Sun, W., Shi, Y., Liu, J.: Optimal placement of cloudlets for access delay minimization in sdn-based internet of things networks. IEEE Internet of Things Journal **5**(2), 1334–1344 (2018)