Cloud, Fog, and Edge Computing: A Software Engineering Perspective

Conference Paper · August 2018

DOI: 10.1109/COMAPP.2018.8460443

Rawan Abulibdeh
University of Guelph

Saleh Moutaz
Qatar University

Cloud, Fog, and Edge Computing: A Software Engineering Perspective

Amal Al-Qamash, Iten Soliman, Rawan Abulibdeh, [$]Moutaz Saleh
Department of Computer Science and Engineering
Qatar University
Doha, Qatar
[$]moutaz.saleh@qu.edu.qa

***Abstract*—Cloud, Fog, and Edge Computing paradigms have been introduced for data-driven organizations in order to facilitate data computation and processing in an easier manner. Considering that there has been a huge increase in the amount of data produced over the past couple of years, and it is expected that the amount of data produced will exponentially grow, recent research has focused on utilizing these paradigms in order to satisfy the growing demand of fast computation and data storage. In order to recognize the most suitable use for these models, this paper will evaluate the three computing paradigms: cloud computing, fog computing, and edge computing in terms of their architectures. Furthermore, a comparative analysis of the non-functional requirements is conducted and used to propose the use of each paradigm in real-life applications.**

Keywords— software, cloud computing, fog computing, edge computing, applications

I. Introduction

The recent development in data-driven applications has given rise to the advancements in computational and storage resources. The Cloud, Fog, and Edge computing infrastructures are used today in various applications that rely on data. Such paradigms provide the organizations with the ability to use various computing and data storage services depending on the organizational requirements. These computing architectures appear similar but rather vary greatly in terms of their characteristics. This allows them to meet different requirements that are needed to satisfy certain real-world applications. In order to highlight these characteristics and understand how these paradigms can be used, the architectures of each paradigm is discussed in detail.

A. Cloud Computing

Cloud computing is an advancing paradigm in software development where it allows data and programs to be removed from PCs and organizations' server rooms and installed into the compute cloud. In another sense, it refers to the general geographical shift of computation. This influences all parts of the computational ecosystem starting from the user and ending at the software developer, IT manager, as well as the hardware manufacturer [1].

Cloud computing is defined by the applications that are delivered as services via the Internet as well as the data centers software and hardware that provide such services. In terms of the services themselves, there are generally three different models; SaaS (Software as a Service), IaaS (Infrastructure as a Service), and PaaS (Platform as a Service) and in terms of the cloud there are various types of architectures; public, private, hybrid, and community clouds. The general infrastructure of the numerous possible service models and cloud architectures are explained in this section below.

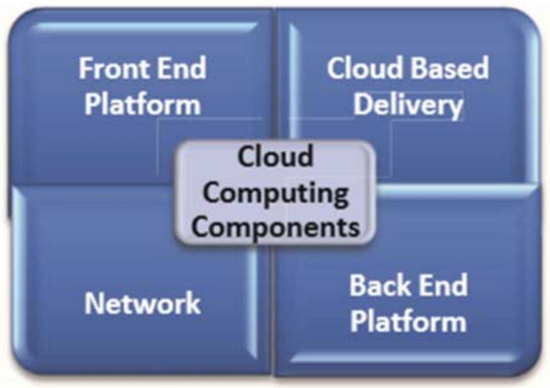

Fig. 1. Architectural components of cloud computing [3]

The architecture for cloud computing is made of various components that are necessary for this type of computation as shown in figure 1.

Front end platform: This is the platform that is visible to the cloud clients. The interface that the client uses to access the cloud can be any software or hardware, depending on the type of cloud computing used to get these services. Some examples include: browsers, operating systems, tablets, mobile phones, or other devices.

Backend platform: It is the end used by a service provider. This platform consists of servers and storage resources which are generally defined as the cloud. There are various deployment models of the cloud. For example, a cloud available as a pay-as-you-go manner for the public, is referred to as the public cloud where the services that are sold on this cloud are called utility computing. These clouds are managed, owned, and operated by either the government, academic or business organization, or any combination of them. On the other hand, any internal data centers of businesses and organizations are known as private clouds. They are administered for the use of a single organization exclusively in which it has multiple consumers like business units [2].

Other deployment models are Hybrid and Community clouds. For community clouds, the infrastructure is administered for use by only a selected community of consumers from organizations having similar concerns. It can be owned by one or multiple organizations in the community. On the other hand, Hybrid clouds are infrastructures composed of two or more diverse cloud infrastructures. These distinct infrastructures continue to be unique entities that are bound through a standardized technology enabling data and application transportability [4].

Cloud based delivery: SaaS, or Software as a Service, provides the consumer the capability to use the applications of the provider that are running on a cloud infrastructure. They are accessible through different client devices by a thin client interface, like a web browser, or a program interface. In this model, the consumer can possibly manage limited user specific application configuration settings. On the other hand, PaaS provides consumers with the capability to deploy onto the cloud infrastructure any applications that are created or acquired by the consumer using programming languages, libraries, services, and tools maintained by the provider. However, the consumer cannot control or manage the underlying cloud infrastructure that controls the installed applications and likely the configuration setting of the application-hosting environment. Lastly, IaaS provides consumers with the capability to deliver processing, storage, network, as well as other computing resources in which the consumer can install and run random software and applications. In this model, the consumer is able to control the OS, storage, and install applications with minimal control of some networking components. In all the service models mentioned the consumer cannot control or manage the underlying cloud infrastructure [4].

Cloud computing, is known as the combination of SaaS and utility computing but generally do not include the small or medium sized data centers, i.e. the private cloud. People have the ability to be providers or users of either SaaS or utility computing; however, some actors can have multiple roles in the infrastructure [2].

Although cloud computing has many advantages and efficient uses, its centralized nature has proven to be inefficient for latency-sensitive applications in terms of transferring and processing the data. Due to the growing amount of data being produced every second, slow transmission rates are expected due to heavy traffic considering that processing occurs in the cloud. Accordingly, since computation powers and the network bandwidth are finite with no major improvement in the latter [5], cloud computing cannot accommodate the transmission of vast amounts of data and real-time processing. This is especially crucial in the next decade as according to [6], by 2020, the number of connected devices will reach more than 50 billion, and by 2019, the data produced might exceed 500 zettabytes [7].

B. Fog Computing

To address these limitations, Fog Computing, introduced by Cisco in 2012, allows for local processing of data by extending "the traditional cloud computing architecture to the edge of the network" [8]. According to [9], reducing the degree of involvement of the cloud by bringing the processing units closer to the end-devices allows fog computing to improve the utilization of the computation power, task execution time, and processing time. With the recent fast-paced development of Internet of Things and all the related sub-fields that connect people and devices, it is apparent that IoT requirements rely heavily on the need for autonomous devices to process, sense, and track incoming data. This is needed in order to facilitate the required services with the majority of the applications requiring fast response time and flexible mobility- both of which are enabled by fog computing [5].

The fog computing paradigm resembles a layered model that extends the traditional cloud computing model by offering a distributed and decentralized platform. According to [5] and [8], the model can be divided into three major layers: cloud, fog, and terminal as shown in figure 2.

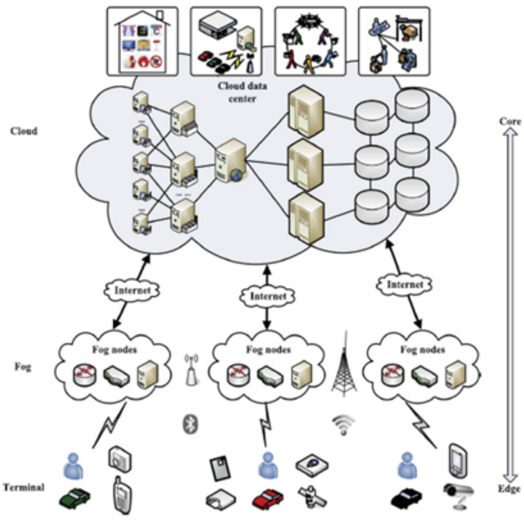

Fig. 2. Fog computing paradigm model [5]

The cloud layer, referred to as the cloud stratum, represents the data centers and servers with high storage and computation powers that manage, operate and process the data received from the fog layer. The fog layer is composed of multiple fog nodes, also referred to as fog cells [9], where each includes a set of network devices with sufficient processing and temporary storage capabilities. The fog nodes receive requests from the IoT or end-user devices and are responsible for recognizing their processing needs to decide whether it should be processed locally, or sent to the cloud [10]. Following that, the terminal layer, also referred to as the device layer, consists of two domains: the IoT devices and the end user devices, where either is sufficient to complete the Fog Paradigm. These devices are responsible for sensing and collecting data from the physical world and sending the data to the fog layer. Thus, the layered nature of this paradigm allows each layer to efficiently communicate

978-1-5386-4371-6/18/$31.00 ©2018 IEEE

with the neighboring layers in order to process the data in a timely manner.

Although fog computing is considered to be a huge improvement from cloud computing for real-time applications, its performance is still limited in terms of latency and bandwidth and its dependency on the cloud can still be considered a drawback. In an effort to maximize these resources and improve performance, Edge Computing was introduced.

C. Edge Computing

Edge computing is a novel computing model that places computing resources and storage at the edge of the network closer to the end user. It provides intelligent services by collaborating with cloud computing [11]. Hence, edge computing can be described as a more localized version of fog and cloud computing. According to [12], edge computing is built on the concept of cloudlets that was first proposed in 2009. The idea behind cloudlets is based on computing "hotspots" which is similar to the Wi-Fi hotspot, as it provides cloud services but without Wi-Fi internet connectivity. The only restriction imposed on the location of the cloudlet is for it to be at the edge of the network or in the proximity of the end user [13].

According to [13], edge paradigm has two advantages; it reduces latency of the communication between end devices and the cloud, and it efficiently utilizes the resources of the cloud and the local network devices, since in edge computing, devices act as both data consumers and data producers. This means that requests between end devices and the cloud are bidirectional as shown in figure 3. Additionally, besides collecting data from the existing database in the cloud and sending them to the user, nodes at the network edge perform many computing tasks including data caching, IoT management, computer offloading and privacy protection.

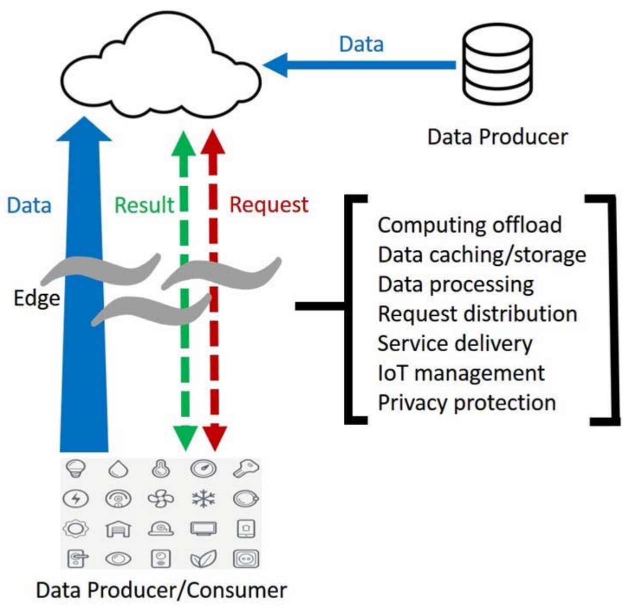

Fig. 3. Edge Computing Paradigm [14]

Moreover, a distinct benefit of this paradigm is its distributed architecture since edge computing is driving the research of the Internet of Everything (IoE). While the Internet of Things (IoT) focuses on the connection between machines and IoT devices, IoE's main focus is the intelligent communication between people, process, data and things. One main reason edge computing is more capable to meet the requirements of IoE as it has a lower chance of data leaks since communication is closer to the users.

II. LITERATURE REVIEW

There has been extensive research on the three computing paradigms. In this section, proposed models of the Cloud, Fog and Edge paradigms are discussed in terms of their architectures and uses. For cloud computing, the SaaS model architectures proposed by [15], [16], [17] are discussed and their benefits are outlined in terms of the various service models. Furthermore, some research papers that revolve around PaaS and the IaaS service models are briefly mentioned. For fog computing, two architecture types are examined with a focus on application provisioning models proposed in [20], [21], [22] in terms of how they satisfy the general non-functional requirements expected. Finally, for edge computing, recently proposed architectural and deployment models in [12] and [23] are discussed. In addition to that, [23], [24], [13] are analyzed in terms of their latency and bandwidth performance.

The cloud model consists of three service models as shown in figure 4, in which each contains different features.

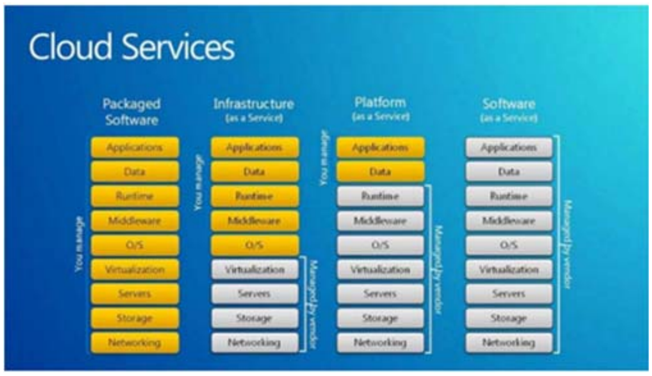

Fig. 4. Cloud service models [15]

Satyanarayana [15] offers an extensive overview on Software as a Service, or the SaaS model in which a comprehensive overview of the service is shown in figure 4. This paper discusses in detail the SaaS layers, architectural maturity in levels, and the offerings and tools of SaaS. Furthermore, some successful SaaS architectures were mentioned as well, discussing the difficulties SaaS removes in terms of providing software products. For example, cost reduction for the customer and provider and ease of upgrading/testing the software.

On the other hand, Cusumano [16] discusses the development SaaS and cloud computing provides in terms of software application delivery. Furthermore, it states the importance for vendors to share their SaaS or cloud infrastructure technology to different product companies which results in the creation of an industry platform, providing Salesfource.com as a good example. Finally, this source mentions the effects SaaS and cloud have on the network and the effects it has on software vendors.

In terms of the rest of the service models, Bhardwaj, Jain, & Jain [17] consider the Infrastructure as a Service, or IaaS and Platform as a Service, or PaaS in which they are shown in figure 4 providing a detailed overview of the variation of such models to the previously discussed SaaS model. This paper briefly mentions SaaS and PaaS software models, providing a general description of both. Later it describes IaaS in detail providing a comprehensive overview of this model. Finally, the paper mentions IaaS providers and consumer and the roles they play in the infrastructure, mentioning the views each have on IaaS.

In terms of the benefits and challenges each of these various cloud service models provide, Gibson et al [18] delivers a comprehensive overview on this topic. This paper also gives a best model solution in terms of balancing control requirements with hardware reduction, configuration & maintenance cost.

Considering that fog and edge computing are relatively new technologies and are still under development, research is still at an early stage. Many papers [8], [19] explain the non-functional requirements of fog computing for the purpose of evaluating the proposed architectures for specific applications. On the other hand, [9] offers a complete list of the functional specifications for the fog computing framework which are classified into the cloud, and the fog management. The cloud management functions include identifying the closest node to the fog node that is requesting data, provisioning the resource demands, executing the incoming request and keeping a record of the services and incoming data. Likewise, the fog management constitutes topology identification, resource provisioning and distributed storage used to be able to share the incoming data with other devices and the cloud layer.

In [8], proposed fog computing architectures in recent years are classified into two types: application agnostic and application specific architectures. It also provides a detailed comparison of the proposed architectures in terms of the development, deployment and management lifecycle stages by evaluating how the following criteria are met: heterogeneity, QoS management, scalability, mobility, federation and interoperability. The results of the comparison show that the majority of the proposed models facilitate certain attributes that is usually required from fog-based applications but are not comprehensive in the met requirements and so, research is still ongoing on proposing optimal models. The application specific architectures are customized architectures that mostly target healthcare and smart applications. Alternatively, the application agnostic architectures mostly focus on the end-user applications and the different attributes associated with the internal operations of the applications that implement fog computing including resource management, application provisioning, communication between the nodes, and the deployment of the cloud and fog service, however only the application provision aspect is focused on in the next part.

Since the development of the fog computing paradigm, a lot of focus has been given to fog-based healthcare, smart cities, connected devices and smart living applications. All these applications fall under the domain of IoT considering that they are latency-sensitive applications that require real-time communication between the nodes to offer the services requested. It is notable that [8] shows a list of the proposed application-specific architectures and demonstrate that only a few manage to achieve certain criteria measurements such as mobility, heterogeneity, and interoperability whereas the rest are unsuccessful in meeting any of the criteria. No optimal model was proposed that can achieve all the criteria, however this might be due to how specific attributes are not critical to some applications, and so, are not preliminary required. These will be further discussed in detail in the next section.

Some of the evaluated application agnostic architectures that satisfy the main functionalities of the fog paradigm outlined in [8] are [20], [21] and [22]. The model proposed by [20] is a Mobile-Fog programming architecture for IoT applications provisioning which gives the developers an opportunity to program any of the nodes found in the fog paradigm in order to facilitate simpler development methods of a large number of distributed, heterogeneous devices. The architecture performance was evaluated by simulating two applications: a vehicle-to-vehicle video streaming and a Mobile Complex Event Processing (MCEP) where in both cases the transfer or request of data is random and occurs between nearby nodes. The results show that with small query ranges, the vehicle-to-vehicle video streaming fog architecture outperforms the cloud-based architecture. However, with the MCEP, it is the opposite as the query ranges are relatively large due to frequent aggregation of sensor-data found in the fog nodes. Quality of Service (QoS) and service federation were not taken into account when developing the model in [20].

Alternatively, [21] proposes a distributed dataflow (DDF) programming model that also targets application provisioning. This is achieved by utilizing the cloud and fog infrastructures which was evaluated by emulating a smart environment. This architecture provides a simple way to develop and design dynamic IoT applications which was proven to be successful; however, some limitations were identified including the lack of federation and communication between remote devices similar to [20]. Another work that focuses on application provisioning is [22] which proposes a PaaS layered architecture in a hybrid cloud and fog environment to automate the provisioning process. The model is then employed in an IoT fire-detection and dispatch application to evaluate its performance. It was concluded that the delay was considerably minimized the closer the application components are to the IoT devices. However, the challenge was the positioning of the components themselves and the number of components that should be placed in the fog layer for optimal performance leading to weak heterogeneity capabilities. Each of these architectures facilitate network scalability, mobility of the nodes, and interoperability of the devices but have varying degrees of efficiency considering that they satisfy different criteria expected from different fog-based applications.

Similarly, a very noticeable point for edge computing is that while it is a promising field, there are several challenges to overcome and the research still needs major development especially for topics such as data abstraction and security measures [14]. In the recent studies done, it was proved that Edge has less latency rates than cloud, where one very recent study [23] proposed actor-based framework for edge computing which is compatible with existing technologies and then did several experiments for latency, jitter, and bandwidth. This framework outperformed cloud computing measures of these metrics proving that edge computing is likely to have better QoS and performance. However, on studying the exact rate, [24] published in January 2017 made an assumption that the latency associated with radio link is negligible or non-existent for a certain Mobile Edge Computing (MEC) architecture. In October of the same year, [13] performed a series of measurements on the same architecture and found that a simple design approach could easily be deployed with noticeable low latency rate compared to cloud.

In fact, MEC has the lion's share of the research. [12] is a survey on the ability of MEC architectures and computation offloading to cope with the requirements of real time application. Its findings show that MEC is a promising technology; however, there are several critical challenges that emphasize the need for further research.

In an effort to combine the benefits of fog and MEC, [25] proposes a hybrid 5G Enabled-Edge (5GEE) model that targets Mobile Crowd Sensing (MCS) by utilizing the functionalities offered by edge in multimedia sharing, and the mobility and low latency offered by fog computing. The paper provides a detailed description of the capabilities of both paradigms and shows how its architecture was able to utilize them by providing a use case example of two MCS scenarios that accommodates the 5GEE. The first example is about collecting and sensing data, whereas the second requires video streaming which is facilitated by edge caching complemented by fog computing operations. Nonetheless, several challenges were identified including the lack of interoperability and reliability of the architecture as well as the unsuitability of the existing orchestrator modules that are needed to handle the compatibility and convergence of the network.

Based on the previous research done, identifying the requirements expected from each paradigm can link the proposed architectures to some applications that could possibly be improved by utilizing these paradigms. Thus, the objective of this paper is to provide a comparative study on cloud, fog, and edge computing from the software engineering perspective by exploring the non-functional requirements associated with each paradigm and the most suited application for each model. The paper will be divided as follows: section II provides an overview of the non-functional requirements that are critical for each paradigm, section III provides a detailed analysis of the functional requirements of some applications to identify the most suited paradigm for it, and section IV summarizes the findings.

III. COMPARISON AND DISCUSSION OF REQUIREMENTS

Each of the three computing paradigms have specific requirements that are suitable for certain applications. Considering that cloud, fog and edge computing are built on one another and each is developed in order to overcome some of the limitations faced in the preceding paradigm, there are some similarities in their properties. However, the differences and requirements are what define each model and facilitate the intended services and applications. Each varies in the architecture as seen in Section I but they all share the same purpose in terms of providing computation and storage facilities. The characteristics of these paradigms are achieved by certain non-functional requirements outlined in this section.

TABLE I. ON-FUNCTIONAL REQUIREMENTS COMPARISON

Criteria	**Cloud Computing**	**Fog Computing**	**Edge Computing**
Scalability	Supported	Supported	Supported
Interoperability	Supported - Needed functionality for when using services from multiple cloud providers to be able to move workloads between them as well as having the ability to mitigate between providers.	Supported – system components can be distributed over different service providers and locations and can be manufactured by various providers.	Supported - The increases interest in IoT development by various vendors and providers and the heterogeneity of edge computing requires high interoperability and flexibility
Mobility	Not supported	Supported - End devices and fog nodes should be able to move dynamically in order to communicate	Highly Supported
Heterogeneity	Not supported	Supported - The nodes in the fog and cloud layers differ in their storage and computational performance capabilities	Supported IoT devices belong to different vendors and providers with different computing power, applications and storage resources
Geographical distribution	Cloud is naturally a distributed storage but it does not support the geographical distribution of devices	Supported - Decentralized and distributed deployment to be able to accommodate mobile and fixed IoT/end-user devices	Supported - key characteristic of the deployment of IoT applications based on sensor networks that benefits from edge computing
Location Awareness	Not supported	Supported	Supported
Performance	Congestion or server failures when processing can affect cloud service which can increase the delay	Supports fast response, low latency, and low bandwidth requirement	Supports the shortest response time, most efficient processing and smallest network pressure which by all means enhance performance
QoS Management	Supported for non-real time processing	Supported for real-time processing and communication	Supported for real-time and provides better QoS (Quality of Service) and lower latency to the end users

Table 1 shows a summarized description of each paradigm and the relevant non-functional requirements. In terms of scalability, for SaaS applications in cloud computing, it is meant to be an on-demand business model. Therefore, it is difficult to predict the load that on the system. The system must be designed to dynamically scale up or down depending on the real-time load on the system [26]. Similarly, Fog networks are expected to have many nodes and components and so, they need to be able to adapt to the application scalability requirement whether it is for a small or a large network. Edge services on the other hand seem to be really promising to scale out as more and more devices are being connected to the edge of the network due to the increase interest of developing IoT devices. Moreover, the fog, edge and cloud layers in all three paradigms are expected to be operated on by different service providers. So, interoperability in cloud computing is an essential functionality for when using services from multiple cloud providers to be able to move workloads between them as well as having the ability to mitigate between providers [27]. This is the same for fog and edge computing where the components of the system can be manufactured by various providers or distributed over different service providers and locations.

With regards to the security of the models, in cloud computing, this can be applied to the cloud provider or the developed system. For the former, the application developer is under the providers' security measures like the network isolation mechanisms or the physical infrastructure access policies since the functionality and ownership of the system, from a SaaS provider's perspective, is with the provider. This means it's the responsibility of the provider to create an implementation that can track the usage of the system as well as the occurrence of events within it. On the other hand, privacy in fog computing is handled through reducing the propagation of data across the network by locally processing the data which reduces the security hazards and increases the data privacy. For edge network, data is also processed locally but closer to the user than fog computing which enforce the privacy even more.

In terms of performance, both edge and fog paradigms have proven their lead, particularly when measuring the bandwidth and response time. For Edge computing nodes, they must be in the proximity of the end devices in order to receive smaller portions of data which make it easier to achieve low end-to end latency, high bandwidth and low jitter services [28]. Moreover, less data is transmitted to the data centers in fog computing by two ways: locally preprocessing the data, and directly communicating with the requests sent from the connected devices which reduces the bandwidth used.

Whereas with cloud computing, the bandwidth usage is much higher as data is constantly sent to the cloud and the cloud dependency is very high. However, consumers are expecting a fast operating system with SaaS applications for cloud computing. Therefore, during design, the potential performance bottleneck must be taken into consideration and implement designs that use notions such as asynchronous processing, micro services architecture, multi-data availability, and more [26]. Moreover, availability is the most crucial NFR for SaaS models especially if the SaaS application handles business critical solutions. Unexpected downtime can cause loss of SaaS customers so it is critical that the design has no point of failure. The Recovery Tune Objective (RTO) and Recovery Point Objective (RPO) factors must be considered [26].

Some other shared features of fog and edge computing is their support for heterogeneity since the nodes can have various forms with no fixed standard. This is especially important as with the recent development of IoT, various set of devices belonging to different vendors and providers are expected to be connected. Furthermore, the end devices and the fog/edge nodes differ in their storage and computational performance capabilities. The edge and fog nodes are much more limited in terms of computation and storage in comparison to the cloud infrastructure [25]. Furthermore, end devices and fog nodes are expected to move dynamically and communicate to facilitate the computing flexibility for IoT devices. Location awareness is another important functionality in fog-based models as end-devices require sensing and locating abilities in order to make the decision of where the data needs to be processed. Also, this would facilitate preprocessing, filtering and caching at different locations [30] which also assists in reducing latency. This property is limited in cloud computing due to the nature of its architecture [9].

The non-functional requirements are necessary to identify the capabilities of each paradigm and how it can better suit applications in the real world as outlined in the next section.

IV. Applications

A. Real-Time Applications

1. Video Streaming

A massive amount of the data generated by IoT devices are bandwidth-intensive [29], including videos from surveillance cameras, police patrol cars and user devices for applications where users continuously transmit video from their smart-phone to the cloud for content analysis. If cloud data centers were used alone for transmitting and storing these videos, there will be huge bandwidth requirements and high latency on the cloud. An example stated by [28] "in modest-city 12,000 users uploading 1080p video would require a link of 100 gigabits per second; a million users would require a link of 8.5 terabits per second".

So, by placing computational resources one-hop away from high-bandwidth data sources as in Edge paradigm, less data need to be sent to the distant cloud data centers [30]. For instance, a cloudlet would be a better choice as data travels the shortest distance to the nearby cloudlet reducing the bandwidth by three to six orders of magnitude. Videos and sensor data from critical locations can be processed locally in the cloudlet to provide real-time information as needed in public safety applications [28]. Using computer vision analytics in real-time, Cloudlets can send the results as metadata to the cloud.

2. Video Analytics

Nowadays, different kinds of cameras are widely deployed in the urban area and in each vehicle. They carry a lot of useful data that could be used for many purposes like searching for a missing child or as a police investigation evidence. However, data from these cameras will usually not be uploaded to the cloud because of privacy issues or traffic cost. Even if the data is accessible on the cloud, uploading and searching a huge quantity of data could take a long time.

With the edge computing paradigm being the first point of contact in the infrastructure for IoT sensor data, it can enforce the privacy policies of its owner prior to release of the data to the cloud which resolves the privacy issue. Hence, any "thing" or edge node like smart phone can perform the request and search its local camera data and only report the result back to the cloud much faster compared to the cloud [14].

In fact, fast transmission time is the one benefit that led edge to be the technology used in ''Live Video Analytics'' project from Microsoft. According to [31], "this system will work across a geo-distributed hierarchy of intelligent edges and large". One functionality of this project is predicting traffic flow using data gathered from all the available cameras in the local area which requires fast transmission to provide time-sensitive data which is guaranteed by the hierarchical architecture of edge computing.

3. Mobile Gaming

Mobile gaming is generally referred to as a network-based electronic game or an application that is game-like in which the gamer has the ability to be mobile while playing. This generally means that the mobile has a network-based communication capability that can extend game functions beyond the device when needed. The advantage of this is that the user has computer-processing and data-storage resources access beyond the ones available on the mobile that is already mobile in nature. The resources are made available through a central computing device generally referred to as a gaming server. These servers are not necessarily centralized but can rather be widely distributed.

Additionally, mobile gaming provides players with collaboration and competing capabilities with one or more different players. The connection can be direct or indirect in which indirect communication means that numerous players have the capability to affect the gaming environment, i.e. the virtual space that users interact in. However, mobile gaming still has the capability to be stand-alone when necessary. Furthermore, mobile gaming devices can be used for multiple purposes such as training and testing of an emergency-response team by providing them with such devices in which an artificial emergency is created for them to learn with through trial and error. Enhancement of mobile games can be conducted through making use of a location-based virtual space influences. This refers to the effect of the virtual space depending on the actual geographical location of the user.

Due to the mobility of the devices and the localization feature of mobile gaming, edge computing is deemed the best fit for this application. Furthermore, edge computing supports geographically distributed networks in which this application requires. Edge computing would provide the necessary computational speed for mobile gaming that is required for fast response to the players using the application unlike cloud in which it would be too difficult to do all the computation there providing too high of a load on the cloud.

4. Healthcare

Healthcare is one of the IoT rise domains, it applications and services requires privacy, real-time and QoS requirements for which are best achieved by edge resources. For instance, the data generated by the body worn sensors need to be processed immediately in case of an emergency. Deploying closer servers to such real-time devices reduces latency through high LAN bandwidth and less number of hops [30].

Moreover, healthcare applications that demands geographically distributed data processing like electronic medical record (EMR) are a good use of edge paradigm as virtual shared data views can be created for such application and exposed to end user via a predefined service interface provided by multiple stakeholders that are geographically distributed and connected through the network edge. To explain further, the hospital summarizes and shares the information and the symptoms of the patient, then provide patient's perception the pharmacy that contact the insurance company about the payment of the medicine and so on. Several parties collaborate and share data and the computation only occurs in the participant's data facility such that the data privacy and integrity can be ensured [14].

B. Near Real-Time Applications

1. Smart Cities

Generally, smart city applications are composed of "complex and large distributed systems" that communicate and share information. They require near real-time monitoring, distribution of nodes, and the interoperability of different systems. This can be facilitated by the fog or edge paradigms, depending on the critical need for real-time and latency requirements. Moreover, collecting, storing, and processing large amounts of information, gathered from the heterogeneous devices distributed over large geographic areas, is facilitated by the cloud layer present in both the fog and edge paradigms [32].

2. Smart Grid

Smart grid is an automated electricity distribution network that was developed as an alternative to the traditional grids. It is used to more efficiently manage the increased energy demand, energy consumption, and reduce the detrimental effect on the environment [33]. This is achieved by using green energy resources and allowing the users and the service providers to regulate and keep track of their use of energy in a near-real time manner. The smart grids utilize sensors or meters that gather information about electricity and energy consumption; however, in order to provide any monitoring services to the clients, it must also collect private information about each client such as their daily usage of the electronic devices. According to [33], all the information retrieved from the meters are transmitted to a cloud-based data center. Based on this, such a system must ensure security when transmitting the data and when storing the data to prevent unknown users or even the service providers from using this data. Thus, smart grid models must be able to provide data privacy and security to protect the clients' data from malicious users, scalability to accommodate large number of clients and any increase in the incoming data, and finally, reliability and flexibility in terms of the transmitted data to provide quality of service to the clients.

Although cloud computing can provide low cost computing resources and scalability through its on-demand access to the data and services provided, latency due to any congestion, inconsistency, and the lack of privacy that cloud is known for cannot be tolerated in this application. On the other hand, these requirements can best be provided by fog computing in two ways. The first way is by improving the client experience by its ability to offer locality of service and computation, a higher degree of data confidentiality and better quality of service by a decrease in delays since the fog nodes would be closer to the end users. The second way relates to the smart grid operations where fog computing allows the sensors to be geographically distributed which is normally the case in this application. It also allows for tracking of the monitoring devices and data aggregation; these features assist in reducing the data traffic and complement the need for the distribution of the sensors.

3. Connected Vehicles

Vehicular systems can be classified into autonomous and infrastructure-based systems. The autonomous systems utilize vehicles on the fly to support ad-hoc events where the fog nodes can communicate with each other. On the other hand, infrastructure systems depends on sensors or nodes that are placed on roadsides to constantly monitor and provide feedback about the vehicles. The vehicles are considered as the IoT devices and act as the sensors, so the vehicles can communicate directly or through an infrastructure. Considering that vehicles are mobile objects with the need to be aware of their locations, and since they require fast response to act accordingly to the feedback received, the response delays must be low.

Accordingly, fog computing and edge computing are best suited as they both support mobility and node distribution. The edge paradigm latency and bandwidth advantages are especially relevant in this context, in particular cloudlet technology is being used to establish approaches like real-time control and accident avoidance. The cloudlet can perform real-time analytics of high-data-rate sensor streams from the engine and other sources to inform the driver to imminent damage or the need for maintenance. Moreover, compatible with the cloud an integration for these information into the vehicle manufacturer's database and analyzing them might reveal model-specific defects that could be corrected in a timely manner [28]. However, recent research has only focused on the Vehicular Fog Computing models that target this application. In both cases, the nodes rely more on the local processing than on the cloud resources which matches the requirements of this application.

C. Non Real-Time Applications

1. Mobile Commerce

Mobile commerce is an application of mobile cloud computing. Mobile commerce refers to the business model of commerce using the mobile devices. These applications achieve tasks that need mobility such as mobile transactions and payments, mobile messaging, and mobile ticketing. They can be put into three categories; finance, advertising, and shopping. These applications are integrated into cloud computing as to solve some of the issues that m-commerce faces like low network bandwidth, great complexity of mobile device configurations, and security. This integration into the cloud computing environment allows for greater customer satisfaction, customer intimacy, and cost competitiveness.

2. Mobile Learning

This application is based off electronic learning and mobility. Traditional mobile learning applications have limits such as the high cost of devices and network, low network transmission rate, and limited education resources. To solve these issues, cloud-based m-learning applications were introduced. One example of addressing the issues includes the use of the cloud that has large storage capacities and powerful processing capabilities to provide the learners with deeper services of information size, lengthier battery life, and speedier processing speed. An example of the advantages of using cloud computing in m-learning is to improve the quality of the communication of the students with their teachers. This can be done through a smart phone software built on the open source JavaMe UI frame-work and Jaber for clients. The students can communicate with their teachers anytime through a website created on Google Apps Engine. Furthermore, the teachers are able to get information about the knowledge of the students in a timely manner.

V. Conclusion

In this paper, we present the three different models: Cloud, Fog, and Edge and provide a detailed analysis of their architectures. Additionally, a comparison of the non-functional requirements of each model is presented. This is done in order to classify real-world applications in accordance to the paradigm that best satisfies the fundamentals of each application and its goals. The applications discussed in this paper were classified into three sub-categories: real-time, near-real time and non-real-time applications. It was apparent that cloud computing as a stand-alone model is less frequently used today as fog and edge computing utilize the characteristics of the cloud and extends its implementation. This is achieved by including additional architectural layers to allow the network devices to be closer to the edge of the network in such a way that it makes use of the storage and powerful computation capabilities of the cloud, and concurrently allow for local processing of data.

Based on the comparisons, Cloud computing is well suited for non-real-time applications such as mobile commerce and learning since they do not require mobility, localization or real-time response, thus, the cloud resources can be fully utilized in these applications with less cost. On the other hand, Fog and Edge computing were found to be best suited for near-real time such as smart vehicles and smart grids, and real-time applications such as video-related uses, gaming and healthcare, respectively. To conclude, it is apparent that all three paradigms are still relevant today and are being used to fulfil different types of application requirements. It is also essential to recognize that the three paradigms complement each other and their architectures combined can be used to satisfy a great deal of real-world applications.

VI. ACKNOWLEDGMENT

We would like to thank Qatar University for all the support to realize this research study and its publication.

REFERENCES

[1] B. Hayes, "Cloud computing", Communications of the ACM, vol. 51, no. 7, p. 9, 2008.

[2] M. Armburst, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stocia and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing", 2009.

[3] A. Sharma, M. Kumar and S. Agarwal, "A Complete Survey on Software Architectural Styles and Patterns", *Procedia Computer Science*, vol. 70, pp. 16-28, 2015.

[4] P. Mell and T. Grance, "The NIST definition of cloud computing", 2011.

[5] P. Hu, S. Dhelim, H. Ning and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues", *Journal of Network and Computer Applications*, vol. 98, pp. 27-42, 2017.

[6] Cisco Global Cloud Index: Forecast and Methodology, 2014–2019 White Paper.

[7] D. Evans, "The internet of things: How the next evolution of the internet is changing everything." *CISCO white paper*, pp. 1-11.

[8] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," in *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416-464, Firstquarter 2018. doi: 10.1109/COMST.2017.2771153

[9] K. Bachmann, "*Design and Implementation of a Fog Computing Framework*", dissertation, Vienna University of Technology.

[10] A. V. Dastjerdi and R. Buyya, "Fog Computing: Helping the Internet of Things Realize Its Potential," in *Computer*, vol. 49, no. 8, pp. 112-116, Aug. 2016. doi: 10.1109/MC.2016.245

[11] J. Zhang, B. Chen, Y. Zhao, X. Cheng and F. Hu, "Data Security and Privacy-Preserving in Edge Computing Paradigm: Survey and Open Issues," in *IEEE Access*, vol. 6, pp. 18209-18237, 2018

[12] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading", *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628-1656, 2017.

[13] I. Hadžić, Y. Abe and H. Woithe, "Edge computing in the ePC", *Proceedings of the Second ACM/IEEE Symposium on Edge Computing - SEC '17*, 2017.

[14] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge Computing: Vision and Challenges," in *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, Oct. 2016.

[15] S. Satyanarayana, "Cloud Computing: SaaS", GESJ: Computer Science and Telecommunications, vol. 4, no. 36, 2012.

[16] M. Cusumano, "Cloud computing and SaaS as new computing platforms", *Communications of the ACM*, vol. 53, no. 4, p. 27, 2010.

[17] S. Bhardwaj, L. Jain and S. Jain, "CLOUD COMPUTING: A STUDY OF INFRASTRUCTURE AS A SERVICE (IAAS)", International Journal of Engineering and Information Technology, vol. 2, no. 1, 2010.

[18] J. Gibson, R. Rondeau, D. Eveleigh and Q. Tan, "Benefits and challenges of three cloud computing service models," *2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN)*, Sao Carlos, 2012, pp. 198-205. doi: 10.1109/CASoN.2012.6412402

[19] F. A. Kraemer, A. E. Braten, N. Tamkittikhun and D. Palma, "Fog Computing in Healthcare–A Review and Discussion," in *IEEE Access*, vol. 5, pp. 9206-9222, 2017. doi: 10.1109/ACCESS.2017.2704100

[20] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, & B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the internet of things". *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing* (pp. 15-20). ACM.

[21] N. K. Giang, M. Blackstock, R. Lea and V. C. M. Leung, "Developing IoT applications in the Fog: A Distributed Dataflow approach," *2015 5th International Conference on the Internet of Things (IOT)*, Seoul, 2015, pp. 155-162. doi: 10.1109/IOT.2015.7356560

[22] S. Yangui *et al.*, "A platform as-a-service for hybrid cloud/fog environments," *2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, Rome, 2016, pp. 1-7. doi: 10.1109/LANMAN.2016.7548853

[23] A. Aske and X. Zhao, "An Actor-Based Framework for Edge Computing", *Proceedings of the10th International Conference on Utility and Cloud Computing - UCC '17*, 2017.

[24] Y. Yu, "Mobile edge computing towards 5G: Vision, recent progress, and open challenges", *China Communications*, vol. 13, no. 2, pp. 89-99, 2016.

[25] P. Bellavista, L. Foschini and D. Scotece, "Converging Mobile Edge Computing, Fog Computing, and IoT Quality Requirements," *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*, Prague, 2017, pp. 313-320.

[26] J. Jayachandran, "Top 10 Critical NFR for SaaS Applications.", https://blog.techcello.com/, 2016. [Online]. Available: https://blog.techcello.com/2016/04/top-10-critical-nfr-for-saas-applications-part-1/. [Accessed: 19- May- 2018].

[27] P. Cripps, "Non-Functional Requirements and the Cloud", Software Architecture Zen, 2015. [Online]. Available: https://softwarearchitecturezen.wordpress.com/2015/04/24/non-functional-requirements-and-the-cloud/. [Accessed: 19- May- 2018].

[28] M. Satyanarayanan, "The Emergence of Edge Computing," in *Computer*, vol. 50, no. 1, pp. 30-39, Jan. 2017.

[29] G. Premsankar, M. Di Francesco and T. Taleb, "Edge Computing for the Internet of Things: A Case Study," in *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1275-1284, April 2018.

[30] A. C. Baktir, A. Ozgovde and C. Ersoy, "How Can Edge Computing Benefit From Software-Defined Networking: A Survey, Use Cases, and Future Directions," in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2359-2391, Fourthquarter 2017.

[31] W. Yu *et al.*, "A Survey on the Edge Computing for the Internet of Things," in *IEEE Access*, vol. 6, pp. 6900-6919, 2018.

[32] N. Mohamed, J. Al-Jaroodi, I. Jawhar, S. Lazarova-Molnar and S. Mahmoud, "SmartCityWare: A Service-Oriented Middleware for Cloud and Fog Enabled Smart City Services," in *IEEE Access*, vol. 5, pp. 17576-17588, 2017. doi: 10.1109/ACCESS.2017.2731382

[33] F. Y. Okay and S. Ozdemir, "A fog computing based smart grid model," *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, Yasmine Hammamet, 2016, pp. 1-6.