IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 27, 2020, accepted March 11, 2020, date of publication March 25, 2020, date of current version April 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2983253

Fog Computing: A Comprehensive Architectural Survey

POOYAN HABIBI[1], (Student Member, IEEE), MOHAMMAD FARHOUDI[2], SEPEHR KAZEMIAN[2], SIAVASH KHORSANDI[2], AND ALBERTO LEON-GARCIA[1], (Life Fellow, IEEE)

[1]Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada
[2]Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran 159163-4311, Iran

Corresponding author: Siavash Khorsandi (khorsandi@aut.ac.ir)

ABSTRACT Fog computing is an emerging technology to address computing and networking bottlenecks in large scale deployment of IoT applications. It is a promising complementary computing paradigm to cloud computing where computational, networking, storage and acceleration elements are deployed at the edge and network layers in a multi-tier, distributed and possibly cooperative manner. These elements may be virtualized computing functions placed at edge devices or network elements on demand, realizing the ''computing everywhere'' concept. To put the current research in perspective, this paper provides an inclusive taxonomy for architectural, algorithmic and technologic aspects of fog computing. The computing paradigms and their architectural distinctions, including cloud, edge, mobile edge and fog computing are subsequently reviewed. Practical deployment of fog computing includes a number of different aspects such as system design, application design, software implementation, security, computing resource management and networking. A comprehensive survey of all these aspects from the architectural point of view is covered. Current reference architectures and major application-specific architectures describing their salient features and distinctions in the context of fog computing are explored. Base architectures for application, software, security, computing resource management and networking are presented and are evaluated using a proposed maturity model.

INDEX TERMS Cloud Computing, edge computing, fog computing, Internet of Things (IoT), advanced internet architecture.

I. INTRODUCTION

As virtualization technologies mature and are pervasively adopted, multi-tenancy is becoming possible not only in high-end computing servers but also in network elements and even end-user equipment. Thus, there is a trend towards creating network and user functions as virtual functions that are outsourced for execution in utility-based computing stores. This trend is driven by the emergence of universal composability that transforms monolithic applications into composable micro-services. The tasks and the associated micro-services vary widely in their requirements, including computing resources, elasticity, interactivity, and latency. These developments have given new life to the concept of ubiquitous computing and the notion of ''computing everywhere''. In this new environment, each and every computing resource may be selected as the best match for some virtual functions or tasks because of location, resources and requirements. Fog computing provides a framework for task segmentation, placement, offloading and execution in a distributed and collaborative environment and hence will play a major role in this new age of computing.

Cloud computing plays the leading role to provide on-demand location-independent computing services in cloud data centers that may be quite distant from the user. However, with the advent and widespread adoption of cloud computing, many new dimensions have been introduced to adapt it to the needs of various computing paradigms. Multi-tier cloud computing, edge computing, mobile edge computing and more recently fog computing are among the complementary trends emerged to help optimize resource utilization and to meet application requirements.

In the context of the internet of things (IoT), smart applications need fast response time as well as large-scale data

The associate editor coordinating the review of this manuscript and approving it for publication was Liang-Bi Chen.

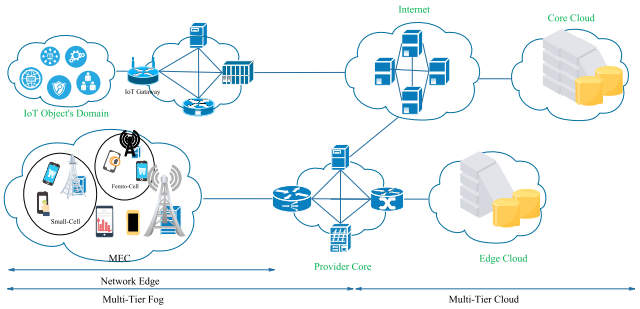


FIGURE 1. A view of multi-tier computing.

processing. The simultaneous satisfaction of both of these requirements requires multi-level data processing as shown in Figure 1. Computing resources at the edge may provide first-order time-sensitive processes as well as data aggregation and filtering. Higher-order processes could then be placed the more powerful resource at the core.

Fog computing presents a new computing paradigm where computation capability, storage capacity, and networking services are placed at the edge and/or in the network, rather than in the cloud over the Internet [1], [2]. Furthermore, management of computing, networking, and storage facilities and programming of networking resources is made possible in a harmonized coordinated manner in fog computing [3], [4]. It provides a ubiquitous and decentralized environment where devices communicate and potentially cooperate to perform storage and processing tasks that can be done with or without coordination with centralized cloud applications. As such, fog computing is a well generalized computing paradigm to be applied in IoT applications [5].

Practical deployment of fog computing needs attention to a number of different aspects such as system design, application design, software implementation, security, computing resource management and networking. The current literature does not cover a comprehensive review of all these aspects. The inter-related nature of these various system dimensions needs an integrated system view identifying the functional components and their interfaces in various layers of the system. Such an integrated view is the main thrust of this paper towards which current state-of-the-art literature has been comparatively reviewed and current trends and future research directions are identified . This paper, however, does not cover the underlying algorithms and enabling technologies in details. Due to their wide variety, they need to be covered in separate surveys in further detail.

The rest of this survey is organized as follows. In Section II, a review of the related works is presented. We have reviewed the existing surveys on fog computing and have drawn the distinction between this paper and previous surveys. In Section III, the proposed comprehensive taxonomy for computing research is presented. Section IV includes the review of multi-tier computing paradigms including fog computing and its competitive technologies. We have identified the differences and similarities of the proposed architectures starting with a comparative review of related technologies including cloud computing, edge computing and mobile computing to explore the road that has led to fog computing. In Section V, the existing reference architectures such as Clouds, Bonomi and AUT are presented. Section VI covers application-specific architectures such as IEC IOT and 5GPPP architectures as well as healthcare and connected vehicle fog architectures. In Section VII, five other architectural aspects including application, software, networking, computing resource and security architectures are introduced and major underlying areas of research in each case are explored. In Section VIII, a maturity model is developed to evaluate the current proposed architectures through which the future research directions are identified and finally, we draw the main conclusions in Section IX.

II. COMPARING TO THE EXISTING SURVEYS

A considerable number of surveys have already been published in the general field including [6]–[8]. In particular, recently, there has been a number of survey papers addressing various aspects of fog computing paradigm. A summary of the existing surveys on fog computing is presented in Table 1.

In a pioneering work in [9], Bonomi *et al.* identified two candidate applications, namely Smart Traffic Light System (STLS) and Wind farm and discussed their requirements. In particular, they identified the need for a middleware orchestrator to manage critical software components. Abstraction/orchestration/data APIs are marked as essential system components to provide resource/service/data abstractions. Foglet agents at fog node to interact with the orchestration engine and policy/capability databases are among other components. The work in [10] appropriately focused on identifying emerging security and privacy concerns due to the heterogeneous nature of fog systems. In their next survey [1], Yi *et al.* identified seven areas of research as networking among virtualized fog nodes, heterogeneous application quality of service, interfacing and programming model, accounting, billing and monitoring, computation offloading, resource management and security. Some of the related technologies in each case was reviewed.

In [11], the authors presented a general reference architecture for fog computing, but did not address other existing architectures. A comparison of fog computing with cloud and edge computing was provided and six key technologies in fog computing including computing, communication and storage technologies, naming, resource management, security and privacy protection were explored.

In [12] a reference architecture for fog computing is presented which contains five distinct layers and some related development and applications are also discussed. The bottommost layer contains the end devices (sensors) as well as edge devices and gateways. Elements from this layer use the network layer for communicating between themselves, and between them and the cloud. The cloud services and resources form the next layer that supports resource management and processing of IoT tasks that reach the cloud. On top of the

TABLE 1. A review of the existing surveys in comparison to this paper: x=not covered, ✓=covered.

Authors and publication date	Description of the coverage	Algorithms	Technologies	Reference Arch.	Specific Arch.	Application Arch.	Software Arch.	Security Arch.	Resource Manag. Arch.
F. Bonomi, et al. (2014) [9]	Key requirements and architectural components of fog computing in the context of two specific applications namely Smart Traffic Light System (STLS) and Wind farm are discusses and then the distributed nature of fog applications and its software architecture are covered.	X	X	X	X	✓	✓	X	X
Yi, et al. (2015) [10]	They surveyed new security and privacy challenges facing fog computing besides those inherited from cloud computing and corresponding solutions in a brief manner.	X	✓	X	X	X	X	✓	X
Yi, et al. (2015) [1]	They identified two application scenarios: augmented reality and content delivery and discussed 7 major problem areas and future directions in brief.	X	✓	X	X	✓	X	X	X
Dastjerdi, et al. (2015) [12]	It proposes a 5-layer reference architecture including edge, network, cloud, application and a software defined resource management layers. Also a number of applications and a case study and how they are played out in a fog environment are elaborated.	X	✓	✓	X	✓	X	X	X
Hu, et al. (2017) [11]	Six technologies to support deployment of fog computing were discussed which include computing, communication and storage technologies, naming, resource management, security and privacy protection.	X	✓	✓	X	✓	X	X	X
Byers (2017) [14]	They discuss a thorough list of 17 architectural requirements, including geo-distribution, data-rich mobility, and multi-tenancy, in the context of 13 critical Internet of Things use cases, and how fog computing techniques can help fulfill them.	X	✓	✓	X	✓	X	X	X
Mahmud, et al. (2018) [15]	An informal taxonomy of challenges and issues in 6 major categories and an appreciable list of needed future efforts is presented.	X	✓	✓	X	✓	X	X	X
Mouradian, et al. (2018) [16]	A very useful survey of applications, algorithms, and challenges benchmarked based on five main evaluation aspects.	✓	X	✓	X	✓	X	X	X
Naha, et al. (2018) [17]	A feature review of some proposed fog application as well as description of the basic 3-layer system model.	X	✓	X	X	✓	X	X	X
Yousefpour, et al. (2019) [18]	Computing paradigms are comprehensively compared and current research is categorized in seven subject areas.	✓	✓	✓	X	X	X	X	✓
This paper	Focuses on architectural aspects to create a comprehensive framework for classification of issues and research.	X	X	✓	✓	✓	✓	✓	✓

cloud layer, the resource management software exists which manages the whole infrastructure and enable quality of service to fog computing applications. The software defined layer is further broken down into eight functional blocks that mainly deal with platform functionalities such as data management and application life cycle management. In the topmost layer, there are intelligent applications that leverage the fog computing to deliver service to end-users.

In [14], they introduced thirteen critical Internet of Things applications and identified a number of use cases that can benefit from fog computing. A thorough list of 17 architectural requirements, including geo-distribution, data-rich mobility, agility, multi-layer programmability and multi-tenancy in the context of fog computing use cases are discussed.

A useful taxonomy for fog computing research was provided in [15]. They briefly reviewed issues involved in fog computing and presented a comparative review of existing work in the area of computation offloading models.

The work in [16] is an extensive survey concentrating on fog computing use cases and application scenarios specifically in the field of healthcare applications. In [16], only application architecture is covered and other architectural aspects are overlooked.

Naha *et al.* work [17] is the only paper that focuses on the architectural aspects but it merely addresses the application architectures. In line with the anticipation of large-scale deployment of fog computing, differences among architectural aspects with an emphasis on various system level architectures have been covered. Fog application architectures, as well as software, computing resource management, and security aspects, are also covered from an architectural point of view.

Yousefpour *et al.* [18] published the most recent survey. It is a formidable effort to shed light on fog computing and related computing paradigms. They provide a comprehensive comparative review of computing paradigms. They categorize the current research based on their objective and subject matter. From objectives point of view, the twelve categories defined are: foundation, QoS, cost, energy, bandwidth, security, mobility, scalability, heterogeneity, management, programmability, and reliability/availability/survivability. From a subject matter point of view, an elaborate taxonomy is proposed that defines 17 subject areas. They do not explicitly address architectural aspects but informally discuss some elements of system and resource management architectures.

In this survey, various aspects of fog computing from the architectural perspective have been investigated in order to fill the existing gap in the literature. First of all, the current taxonomies do not sufficiently cover all aspects of fog computing research. Also, they fail to address architectural alternatives in its various aspects. Each of the existing surveys provide a partial and mostly mixed review of algorithmic, enabling technologies and architectural proposals. In this paper, a novel taxonomy that encompasses the current literature in an structured and well-defined manner has been proposed based on which the existing architectures for system, application, software, resource management, network

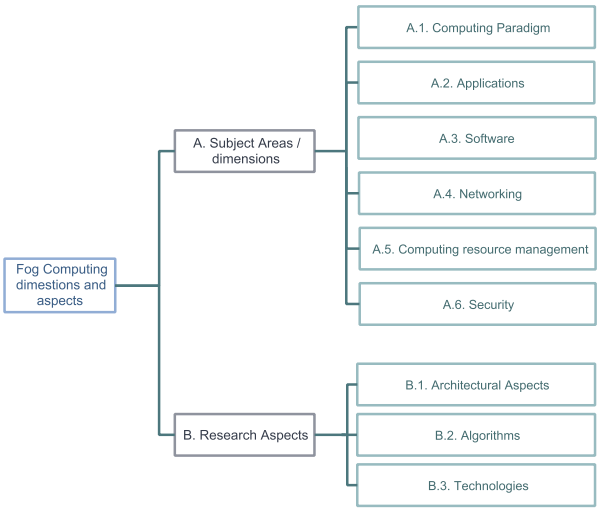


FIGURE 2. The proposed taxonomy of the fog computing dimensions and aspects.

and security are reviewed. In addition, a maturity model to measure the state of current technologies from various architectural aspects has been developed.

III. THE PROPOSED TAXONOMY FOR FOG COMPUTING RESEARCH

A large body of work is devoted to various aspects of cloud computing, edge computing and fog computing in recent years. Due to the vast and diverse nature of research and development efforts in the area of fog computing, providing a classification framework is essential to monitor the challenges and trends and shape future directions. In [1], the issues and challenges of fog computing were categorized in seven major problem areas: networking, quality of service, interface and programming model, computation offloading, accounting/billing/monitoring, resource management and security. In an attempt to establish a taxonomy for fog computing research, Buyya, *et al.* in [15], identified 6 categories of related technologies: fog node design, nodal collaboration, resource/service provisioning, management, networking and security. They addressed these issues mostly from a technology point of view and did not explicitly discuss the architectural aspects. Mouradian, *et al.* in [16] divided the current research and challenges into two categories: architectures and algorithms. In the architecture side, they focused on application architectures and algorithmic side, the resource management and energy management were elaborated.

Considering the previous classifications and the diversity of publications in this field, a new taxonomy which is shown in Figure 2 has been proposed. Architectural design, algorithms and technologies are the three top-level research aspects. Architectural dimensions provide a high-level view of the system identifying the main functional components and system organization. Algorithms and technologies follow the architectural dimensions and provide further details in terms of underlying processes, tasks and implementation on various

TABLE 2. A bird's-eye view of research issues based on the proposed taxonomy.

Subject Areas vs. Research aspects	Architectural aspects	Algorithms	Technologies
Computing paradigm	Reference architecture [53] Fog computing vs. Edge computing [5], [162] Fog-cloud interplay [62], [63], [184] Fog for 5G [66], [76], [79]	-	Orchestration, monitoring, accounting, management technologies [1], [67] Interfaces [70]
Applications	Hierarchical application architecture [9] Healthcare application architecture [85] Fog in Vehicular technologies [82]	Nodal collaboration [15] Application migration [46] Application partitioning [168], [170]	IoT applications [48], [64] Healthcare/Connected vehicles tech. [37], [139] Qos/SLA management [164]
Software	Platform architecture [19] Virtualization paradigms [20] Software architecture [99], [100]	Task/data flow optimization [100] Service elasticity algorithms [21]	Application platforms [55], [88] Programming models [99] Fog node design [9]
Networking	Network slicing [22] Network Function Virtualization [68] Software defined networking [5], [69], [186] D2D/M2M networking [188]	VNF placement algorithms [164] Controller placement algorithms [189] Green networking [190]	Low power comm. Tech. [166] Network slicing in 5G [23] SDN/NFV [79]
Computing resource management	Service Function chaining (SFC) [24] Resource allocation framework [162], [170]	Task offloading algorithms [182] Fog node placement algorithms [159] Automation [9] Energy optimization [180]	Content storage technologies [163] Caching [163] Code offloading [174]
Security	Security threats [102], [103] Threat mitigation [120], [121]	Attack detection [114] Malicious code analysis [134], [135] Privacy protection algorithms [34], [107]	Sandboxing technique [133] DDoS prevention [116], [117] Network security [131] OS security [136] Physical layer security [157]

aspects of the architectures. Each of these three aspects can be applied in a number of subject areas that are divided into six major categories: computing paradigm, application, software, networking, computing resource management and security. Therefore, a matrix of research areas is defined that encompasses previous classifications and provides a comprehensive framework to which all previous classifications can be easily mapped. Table 2 shows a bird's-eye view of the existing research issues based on the proposed framework. The main focus of this paper, however, is on the architectural aspects that is covered in the rest of this paper.

IV. MULTI-TIER COMPUTING PARADIGMS

Cloud computing, edge computing and mobile computing are three technologies that are directly related to fog computing, yet there are clear distinctions that are discussed in this paper in order to clarify the application scenarios of the fog computing.

A. CLOUD COMPUTING

Cloud computing has emerged as a compelling paradigm for managing and delivering on-demand services over the internet [25]. The fast-paced development of cloud computing is swiftly changing the landscape of information technology, and ultimately turning the long-held promise of utility computing into a reality [26]. This paradigm attracts service providers to dramatically reduce their provisioning plan overhead. In addition, it enables them to scale their resources based on demand [27]. Moreover, cloud computing offers reliable services through next-generation data centers that are made by virtualized compute, storage and network technologies. Furthermore, users are able to access applications and data from a cloud anywhere and anytime [28].

In cloud computing, users are granted resources to use for their infrastructure, platforms, and software from a shared pool of resources by cloud providers (such as Google and Amazon) for a fee [29]. Generally, public cloud vendors have built large data centers with enough computing resources to serve many users worldwide. Moreover, users can use resources on-demand and elastically [30]. Cloud computing provides several features that are shown in Figure 3. However, most of the Cloud datacentres are geographically centralized and located at remote sites, far from the proximity

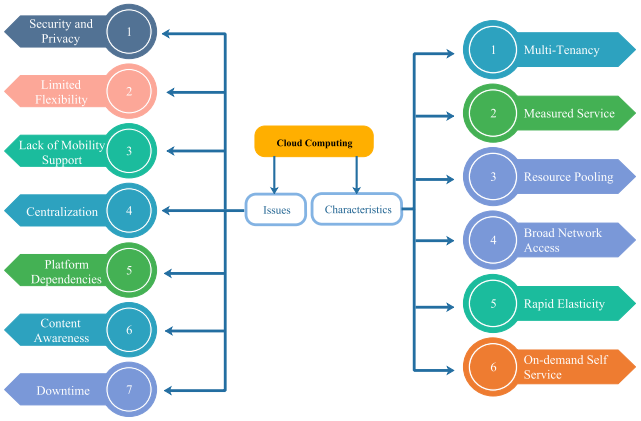


FIGURE 3. Cloud computing characteristics and issues.

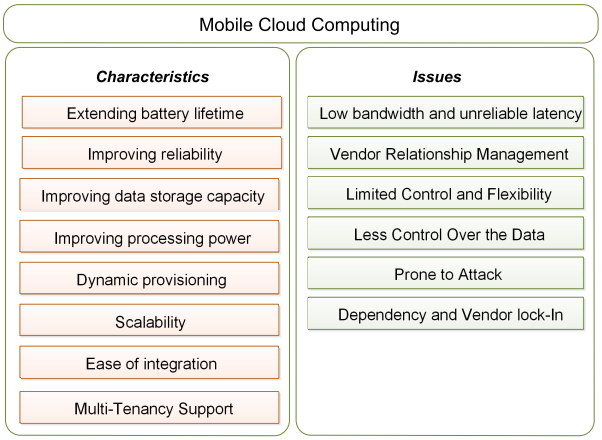


FIGURE 4. Mobile cloud computing characteristic and issues.

of the end-users. As a consequence, real-time and latency-sensitive computation service requests often endure large round-trip delay, network congestion, and service quality degradation [31]. Figure 3 shows several issues of cloud computing in a nutshell [32], [33] and [34].

To reduce network latency and distribute computing load over geographically diverse cloud resources, various forms of multi-tier cloud computing are proposed providing storage and computation along a succession of datacenters of increasing sizes [35].

B. MOBILE CLOUD COMPUTING (MCC)

Mobile devices are employed as an important tool for learning, entertaining, social networking, video and audio communications, and getting news. However, due to resource constraints of mobile devices (such as computation capacity, battery lifetime, and storage capacity), mobile users are not able to attain the same quality of experience in comparison with desktop users. Mobile cloud computing (MCC) has been introduced to solve the constrained resource problem of mobile devices using cloud computing. MCC is defined by the MCC forum as follow: "Mobile cloud computing at its simplest, refers to an infrastructure where both the data storage and data processing happen outside of the mobile device. Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing applications and mobile computing to not just smartphone users but a much broader range of mobile subscribers". In other words, MCC is defined as a combination of mobile applications and cloud computing where all complex computing modules are processed in the clouds [6], [36]. Mobile devices communicate with the cloud with the help of base stations, access points or satellites. Information that is transmitted from the mobile devices is processed on the network provider side. Hence, mobile network operators are in a strong position to provide valuable services such as m-healthcare [37], m-learning [38], m-gaming and m-governance [39] as these services are directly accessible from the mobile devices [40].

The main advantages and some related challenges of MCC are classified in Figure 4. Long WAN latencies are one of the main problems, particularly due to the dramatic increase in today's traffic loads [41]. Bandwidth limitation in wireless access networks and QoS support for mission-critical applications are among other issues facing MCC. On the security and privacy side, awareness by the service provider of the user's location and activities can possibly cause privacy issues. Security challenges in cloud computing in general such as data leakage and data outsourcing concerns also exist.

C. EDGE COMPUTING

To overcome the limitations of cloud computing and MCC, researchers and network engineers have developed innovative solutions such as cyber foraging [42]–[44], cloudlets [45], and computational offloading [46]. These techniques are proposed to offload parts of the computation from the cloud to a surrogate machine in the vicinity of the user. These solutions, generally referred as edge computing, take advantage of location awareness and can provide a far more timely response. Edge computing is about pushing the frontier of computing applications, data, and services to the logical extremes of a network and away from centralized cloud nodes [47].

Mobile edge computing (MEC) is a particular form of edge computing where a cloud server is running at the edge of the cellular networks and its role is performing tasks, such as augmentation of application performance and reduction of network congestion which could not be done with traditional network architecture (Figure 5) [48], [49]. MEC has been defined by ETSI as "a model for enabling business-oriented, cloud computing platform within the radio access network at the close proximity of mobile subscribers to serve delay sensitive, context-aware applications" [50].

A MEC platform allows computation and services to be hosted at the network edge, which reduces the total latency and bandwidth consumption of the subscribers [51]. Network operators can allow the radio network edge to be handled by third-parties, which in turn allows the rapid deployment

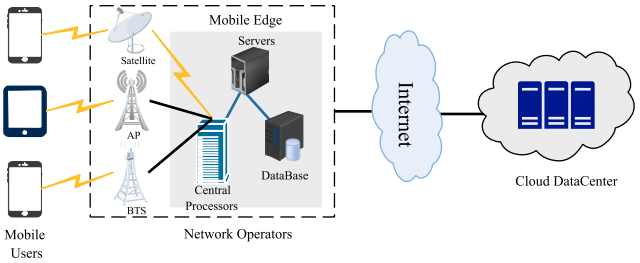


FIGURE 5. MCC/MEC computing architectures.

Mobile Edge Computing Characteristics	
Proximity	Edge server is nearby to devices; it can extract device information and analyze users behavior to improve services
Geographical distribution density	Dense geographical dispersed infrastructure contributes in various ways. Services can be provided based on user mobility without traversing to the core.
Low latency	Applications are hosted at the edge network; thus, the available bandwidth within the edge network is high in compare to the core network, average network latency is reduced
location awareness	Application developers use user location to provide context aware services to the user by the information which is gathered by base stations
Network context information	Real time RAN information are used to provide context related services to the mobile subscribers

FIGURE 6. Mobile edge computing characteristic.

of new applications and edge services to the mobile subscribers [49]. MEC provides a rich set of features that enable service providers to offer new services (Figure 6). As an example, the work in [52] offers a solution where real-time RAN information, such as network load, user's location, is provided to the application developers and content developers. This real-time network information is employed to provide context-aware services to mobile subscribers, thereby enriching the user's satisfaction and improving Quality-of-Experience (QoE).

D. FOG COMPUTING

Fog computing is a network architecture that uses near-end-user edge devices to accumulate a large amount of storage, communication, and computing resources that is used to carry out processing, control, configuration, measurement and management tasks. Since its inception, it is approached by researchers from various perspectives. As a result, there might not be a universal consensus about its meaning and definition. Some consider fog computing as being synonymous with or being an extension of edge computing [48]; from another perspective, it is considered an extension of cloud computing [13]. Regardless, there is a shared view about the following common properties of fog computing:

- It is based on a scenario where a huge number of heterogeneous ubiquitous and decentralized devices collaboratively perform network functions or applications.
- The term 'the fog' is not constrained to a particular technology area. It supports device and interface heterogeneity.
- It usually consists of a virtualized platform supporting basic network functions or new services and applications that usually run in a sandboxed environment.
- Fog computing supports features such as managed mobility, programmable communication and location awareness, to name a few [3].

Fog nodes form the physical infrastructure that provides resources for services at the edge and in the network. These devices are categorized in two discrete groups [15], [53]. One is resource-poor devices such as access points, routers, switches, and end devices. The other group, which is more elaborate, is resource-rich machines such as Cloudlets [54] and IOx devices [13]. A cloudlet is a trusted, resource-rich computer or cluster of computers that is well connected to the Internet and available for use by nearby mobile devices. In other words, it is a mobility-enhanced and small-scale cloud data center that is situated at the edge of the Internet [55]. It supports resource-intensive and interactive mobile applications with low latency and can overcome the challenge of high bandwidth demand of multiple users who are generating and receiving media interactively. IOx is a novel Cisco fog device that works by hosting applications in a Guest Operating System (GOS) running on top of a hypervisor on the Connected Grid Router (CGR). Cisco called IOx as "an application enablement framework for the IoT". It was the first software-only version of the IOS wrapped in with a Linux distribution. The rudimentary goals of Cisco were helping to solve "data tsunami" and "data gravity" issues for the IoT [13]. IOx is an industry scale realization of fog in the network. As one of the fog's imperative goals, Cisco's aim is to host applications and services on network middle-boxes by adding computing and storage ability to them. Doing so, it reduces a large amount of traffic load on the cloud by distributing processing load over entire networking elements.

E. EVALUATION METRICS AND CHALLENGES IN COMPUTING PARADIGMS

From the IoT application point of view, traditional cloud-based systems are challenged by the huge number of IoT devices, their heterogeneity, and high latency witnessed in some cloud ecosystems. Fog computing decentralizes applications, management, and data analytics into the network using a distributed and federated compute model. It promotes reduction of network traffic that makes it a suitable candidate for performing IoT tasks and queries [9]. Cloud computing is mostly about providing resources that are distributed in the core of the network, while fog computing is about providing elastic resources and services at the edge of the network. Fog computing, in some scenarios, may be used interchangeably with edge computing [8]. However, the idea in edge computing is to push data processing to the data source as much as possible while fog computing provides a more general

TABLE 3. Comparison of computing paradigms.

Paradigm/Criteria	Cloud Computing	MCC	MEC	Fog Computing
Latency	High	High	Low	Low
Delayed-Jitter	High	High	Low	Very Low
Location Awareness	No	Yes	Yes	Yes
Support for Mobility	Limited	Supported	Supported	Supported
Geo-Distribution	Centralized	Centralized	Distributed	Distributed
Real-time Interaction	Limited	Limited	Supported	Supported
Location of Service	Within the Internet	Within the Internet	At the Edge	Within the Edge
Distance	Multiple Hops	Multiple Hops	Mostly one Hop	Mostly one Hop
Storage Capacity	High	High	Few	Few
Permanency of Storing Data	Permanent	Permanent	Transient	Transient
Geographic Coverage	Global	Global	Local	Very Local
Response time	Seconds to Minutes	Seconds to Minutes	Milliseconds	Milliseconds

framework where a multitude of edge devices collaboratively provide the necessary computing platform. In multi-tier cloud computing models as in [7], the fog layer can be mapped to ''Smart Edge'' that consists of heterogeneous computing resources close to the user. In [56], the term ''Cloudlet'' is used to refer to mobility-enhanced small-scale computing resources at the edge of the Internet that can be viewed as being equivalent to the fog as it is defined here.

Latency, location awareness, geographical distribution, mobility, and heterogeneity are the main criteria used in the literature to evaluate and compare the computing paradigms [57]–[59]. Table 3 demonstrates a comparative study of the current computing paradigms from these perspectives. In the following, we discuss the main challenges and characteristics of fog computing in comparison to other computing paradigms based on the evaluation metrics in Table 3:

1) LATENCY

The implementation of fog computing at the edge of the network enables data to be processed near end devices instead of sending raw data to be processed in the cloud, speeding up the data processing and resulting in delay reduction [2]. Lower fog latency provides better support for time-sensitive applications responsible for analyzing the raw data gathered from the sensors in real time, sending control commands to the sensors and actuators, and finally providing summary reports for data visualization purposes to the central cloud [58]. Cloud robotics is an example of time-sensitive applications requiring real time analytic. A robot's motion control depends on data that is collected by sensors. This data is transferred to the robot's control system located in a fog node that facilitates making quick decisions for the robot's motions. Fly-by-wire aircraft and anti-lock applications are other examples.

From the proximity point of view, a fog node may not be the immediate point of access for end devices, for instance, the first router connected to the end device may not be resourceful enough to run a Fog computational Node (FCN) framework; therefore, closest fog node may occur multiple hops away. However, for Cloudlet and Mobile Edge Computing, the devices connect directly to the edge node over Wi-Fi or cellular network

2) GEOGRAPHICAL DISTRIBUTION

Fog computing is an intelligent distributed platform that geographically covers a wide area at the edge layer of the network. Compute, storage, and network resources are located near data collectors, which is distributed geographically and are managed coherently by platform owners. Examples include IoT use cases, such as smart cities and smart grids that are naturally distributed. For instance, for measurement of pollution, the sensors need to be distributed throughout a city.

From an implementation point of view, fog nodes may exist anywhere between end devices and the cloud to offer better flexibility in selecting computing devices. However, the computation and storage capacities of fog nodes are usually lower than Cloudlets and MEC devices. In edge computing also the frontier of computing is pushed to the edge of the network. In contrast, fog computing uses both edge and network devices in collaboration with cloud resources. In scenarios where edge computing is prevalent, fog and edge computing are used interchangeably.

3) LOCATION AWARENESS

Since fog nodes are geographically distributed and are close to the objects, they have the ability to precisely estimate the physical location of the objects especially when they are mobile. This allows fog nodes to track the objects to make smart decisions based on their location [9]. As an example, Smart Traffic Light Systems (STLS) make decisions based on the estimated location and speed of the objects. These decisions can be more easily implemented in fog using its location awareness capability.

In the context of request-handling mechanisms, there is a supervising entity overlooking the underlying nodes to collect information on resource status and availability. The more diverse and heterogeneous nature of fog environments in comparison with other computing paradigms call for an abstraction layer. MEC, on the other hand, has an advantage of having fine grained information of end-user's location and that can be employed for context aware application scenarios.

4) MOBILITY

Mobility support can be viewed from two different aspects. From the customer's point of view, they need to have a penchant to gain access to services from everywhere, at any time, without any limitation. By running on-demand services on the cloud, the opportunity for users to have access to their services from anywhere and at any time is provided. From the fog node's point of view, mobility of the objects or sensors, which have the responsibility of collecting data, is another issue since the mobile objects or sensors may move out of the fog node's coverage area. The support of mobile objects requires a flexible and coherent platform with the ability to reconfigure swiftly switching contexts. It manages the allocation of resources and the migration of objects from one location to a new one. As an object moves through the network, when it exits a fog node's coverage, another adjacent fog node will be allotted to this object. This service migration prevents loss of service and provides service continuity [2]. Higher layer protocols support application level communication with mobile devices. As an example, the Locator ID protocol (LISP) is an enabler technology that provides mobility for mobile devices through the network. This protocol defines a mechanism for LISP routers to encapsulate IP packet addressed with endpoint identifier (EID) for transmission across a network infrastructure that uses Routing Locators (RLOCS) for routing and forwarding.

5) FOG-CLOUD INTERPLAY

It is essential to specify which decisions should be made by the fog nodes at the edge of the network and which should be made by the cloud at the core of the network. Hence, there is a need for an appropriate interplay between fog and cloud. Here, the nature of decisions is vital. Fog nodes are the best choice for time-sensitive decisions, while cloud data centers are chosen for resource-intensive big data analytics on historical data. The decisions made by the fog nodes are measured in milliseconds to sub seconds, while big data analysis may take days or months and be stored in long-term storage. The responsibility of fog nodes is to process the local data that causes a reduction of traffic that is sent through the network. Processed data is sent to the cloud for long-term processing. For instance, in connected vehicles [60] and Smart City applications [61], real-time decisions are made by fog nodes, while high-level decisions that need more compute and storage resources, such as future planning, are made by the cloud. Mobile cloud computing has most of the features of cloud computing, however, MCC provides location awareness and mobility support as it is tailored for mobile devices.

6) SCALABILITY

The monitoring of a large-scale environment requires numerous sensors and actuators to collect the raw data. Sending this huge amount of data to the cloud is not efficient. Analysis of the data close to data producer is more reasonable and more sensors and actuators can be supported. Proper geo-distribution of fog nodes is essential in system scalability. When sensors need to cover a wide area, a fog node in each domain is required for local analysis of the data.

7) REAL-TIME INTERACTION

Some applications need real-time interactions between fog nodes and actuators. Fog nodes analyze data locally and issue control commands based upon the raw data that is collected by the sensors interactively [62]. Existence of fog nodes near the sensors and actuators is critical for supporting these time sensitive decisions. In the robot scenario, a fog node transmits motion control commands based on the collected data and updates state information of the robot. The new motion control command is then sent to the robot based on the updated state information. This kind of interaction coined ''real-time interaction'' that is made between fog nodes and actuators [63], [64].

8) HETEROGENEITY

Fog computing is inherently heterogeneous both in fog nodes themselves and in its network infrastructure. Fog nodes include high-end servers, edge routers, access points, set-top boxes and end-devices such as vehicles, sensors, and mobile phones. These nodes have different hardware platforms with various kinds of operating systems and applications based on their hardware and software capabilities. The nodes need to cooperate for data processing, hence the need for fog computing which provides a coherent platform among these nodes [65]. The fog network infrastructure includes high-speed links that connect enterprise data centers to the core, and multiple wireless access technologies used in the access layer of the network, such as 4G, 5G, and Wi-Fi [66]. Using a stack communication protocol, fog nodes collect and analyze data from various sources through various communication technology and protocols. Data analytics transforms raw data into a uniform information structure that is sent to the rest of the network.

V. REFERENCE ARCHITECTURES

The system or reference architecture is the main basis for the implementation of the system as it identifies the main functional components and their interconnections. A reference architecture often includes interfaces (or APIs) in order for functional blocks to interact both within and outside of the system to fulfill the vision and purpose of the architecture. There are a different number of references architectures developed for fog computing from various perspectives. In this section, reference architectures that are specifically designed for fog computing, are investigated and surveyed. Existing reference architectures are defined with different levels of abstraction. A highly abstract architecture shows different types of components, each providing a set of functions and tasks. A lower-level architecture may demonstrate the interactions or procedures within a function defined to perform a specific task. The architectures discussed below are varies in their abstraction levels.

A. OpenFog CONSORTIUM

OpenFog [67] is a hierarchical architecture for fog computing proposed by Open Fog Consortium. Figure 7 demonstrates the position of fog nodes in several different deployment views. The fog layer covers all computing facilities between the end points and the cloud servers in several hierarchical orders each of which can be deployed based on the type, size and latency requirement of the data processing jobs. Fog nodes can communicate with each other through wired or wireless channels. Each tier of fog nodes differs from other tiers in several parameters such as processing capabilities, networking abilities, and reliability of nodes. Each tier sifts and extracts meaningful data to create more intelligence. In the bottom tier of the architectural layer, data acquisition/collection, and data normalization are performed. In the upper layer, data filtering, compression, and transformation are accomplished. The upper fog layer, which is closed to the cloud, transforms the gathered data to a knowledge base for permanent storage. The overall system intelligence and capacity is increased as the layers are closer to the cloud. Architecturally, fog nodes at the edge may require less processing, communications, and storage than nodes at high levels. However, Input and Output (I/O) accelerators required to facilitate sensor data intake at the edge are much larger in aggregate than I/O accelerators designed for higher-level nodes. The conventional centralized cloud computing continues to remain an important part of computing systems as fog computing emerges.

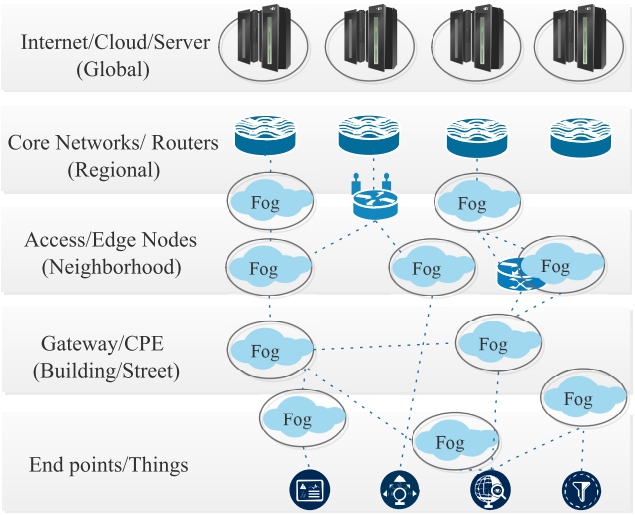


FIGURE 7. OpenFog deployment scenarios.

Figure 8 shows a more detailed view of OpenFog reference architecture where four perspectives and three views are defined. The perspectives cover the non-functional aspects of the system covering security, manageability, analytic and control, and fog business. Three views are software, node and system views. The system view is composed of one or more node views coupled with other components to create a platform. The node view covers pieces and components from

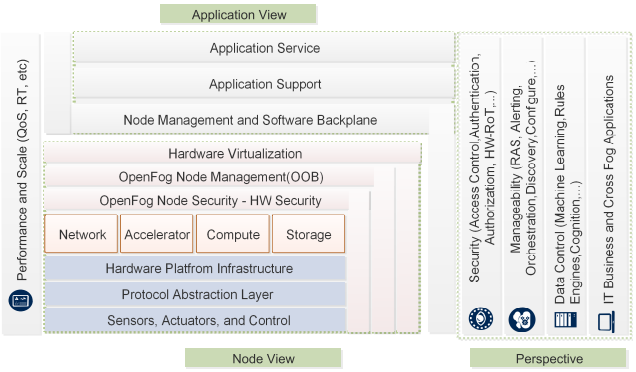


FIGURE 8. OpenFog layered architecture.

system developer's point of view, while the system view is concerned with components and nodes interconnection and performance from system designer's point of view. From the node view perspective, fog nodes need to support several functions including networking, computing, accelerating, storing, and controlled sensors and actuators. The node also needs to implement sufficient security mechanisms as well as management agents. As with diverse computation and networking heterogeneity, as abstraction layer need to be implemented to provide standard API for connecting to other system components. The fog nodes engaged in enhanced analytic need to configure accelerator modules such as graphics processing units, field programmable gate arrays, and digital signal processors to provide supplementary computational throughput. Many types of storage are employed in fog nodes to fulfill the required reliability, durability and data integrity of the system. In addition, fog nodes can be connected in a mesh topology to provide load balancing, resilience, fault tolerance, data sharing, and minimization of cloud communication. The system software is divided into three layers: application service, application support, node management and software back-plane. Application support includes a broad spectrum of software used by and often shared by multiple applications (micro services). A few of these services include run-time engine, security services, message and event bus, application storage and analytic tools and frameworks. Software back-plane includes OS, drivers and firmware, file system, virtualization and containerization.

B. CLOUDS LAB ARCHITECTURE

Another fog computing reference architecture is proposed in [12] which is depicted in Figure 9. This reference architecture comprises five layers namely, access (sensors, edge devices, and gateways), network, cloud services and resources, software-defined resource management, and IoT applications. At the lowest layer, end devices with their applications and edge devices are located. It also includes the gateways that are connected through the network layer and provides the connectivity services to the edge systems. The cloud layer provides the computing platform for resource management and IoT applications. The software defined

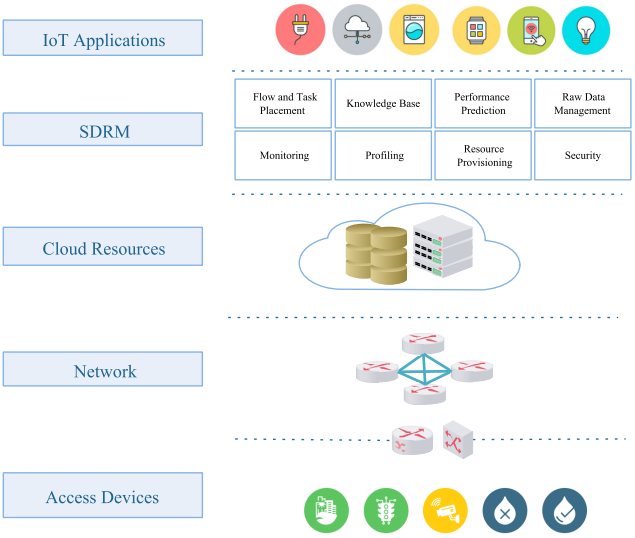


FIGURE 9. CLOUDS reference architecture.

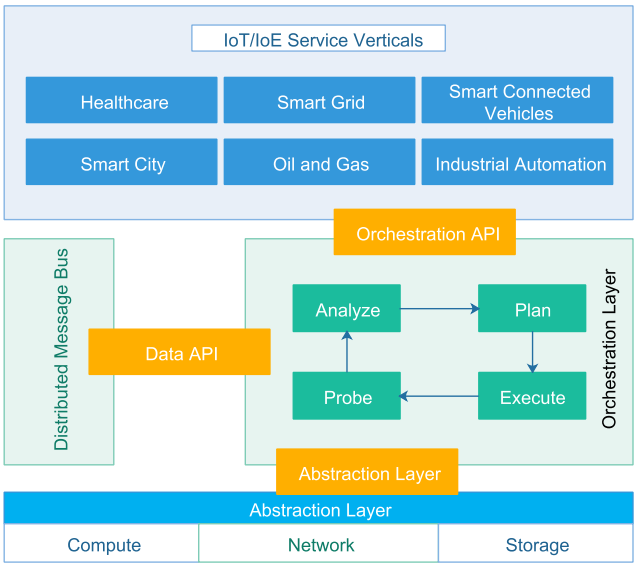


FIGURE 10. Cisco-Bonomi reference architecture [9].

resource management layer manages and orchestrates entire resources across the architecture based on an abstract view of the resources that significantly reduces the complexity of the decision making. In this layer, many middleware services are introduced to optimize the use of the cloud and fog resources for IoT applications. The aim of this layer is to increase the use of fog in order to improve application performance and reduce delay enabling quality of service for applications. Finally, the top layer contains the IoT applications using fog computing to provide innovative and smart services to the end users.

The Software-Defined Resource Management layer is responsible for managing fog/cloud resources and plays a pivotal role in enabling the fog computing. It consists of eight functional blocks described as follows. Flow and task placement keeps track of available resources in the fog and cloud in order to allocate incoming tasks and flows to the appropriate element. It communicates with the Resource Provisioning service to allocate/free-up resources to the flows and tasks and decides whether to accept upcoming tasks. The Knowledge Base component stores historical information about the history of resource allocation and demand types in order to improve the efficiency of the algorithms. Performance Prediction uses results of Knowledge Base system to make an estimation of resource performance which will be utilized by resource provisioning system. The Monitoring block keeps track of the performance and status of applications, and the Profiling block builds and maintains the resource and applications profiles based on information received from Knowledge Base component. Resource Provisioning is responsible for allocating cloud-fog and network resources to host the applications in a dynamic fashion. The last component is Security that performs authentication, authorization, and encryption.

C. CISCO-BONOMI REFERENCE ARCHITECTURE

Flavio Bonomi *et al.* [9] explores the need for fog computing and proposes a reference architecture as depicted in Figure 10. Fog computing is envisioned to support low-latency, geo-distribution, location aware, and distributed control applications that cannot be efficiently supported by cloud-only solutions. Fog nodes are heterogeneous in nature and deployed in various places such as core, edge, access networks, and endpoints. The fog resource management is required to provide seamless resource management across the diverse set of platforms. In addition, the fog architecture should be flexible enough to host diverse set of application use cases such as vehicular networks and IoT applications.

The reference architecture in [9] consists of five main components: the heterogeneous physical resources layer, the fog abstraction layer, the fog service orchestration layer, IoT services and distributed message bus.

The heterogeneous physical resources layer consists of fog physical resources including servers, edge routers, access points, vehicles, sensors and mobile phones. The fog abstraction layer has the responsibility to hide the heterogeneous nature of the fog platform and to provide a uniform and programmable interface in order to allow seamless resource management and control through the higher layers. In this layer, a set of generic APIs are defined for monitoring, provisioning and controlling physical and virtual resources. In addition, this layer determines security, privacy and isolation policies for different components of the architecture. The service orchestration layer is designed to manage a fully distributed fog computing environment and offers dynamic, policy-based life-cycle management for fog services. This layer contains four main components, namely policy manager, life-cycle manager, capability engine, and a distributed database containing business policies, fog nodes' state information and fog nodes' hardware and software capabilities. It implements a distributed policy engine with a single global view and local enforcement. Foglet is defined as a software agent that

expands orchestration functionalities on the edge devices. It uses abstraction layer APIs to monitor the health and state associated with the physical machine and its services that can be analyzed locally or globally. The distributed message bus is a scalable bus that is deployed to carry control messages for service orchestration and resource management.

Bonomi *et al.* [9] indicate that cloud and fog ecosystems support multi-tenancy. Although these ecosystems both could support the multiplicity of client-organizations without mutual interference, there are subtle differences in the nature of their client-organizations. Most of the time, heterogeneous physical resources in a cloud are deployed in a centralized manner. Fog - which its distributed infrastructure contains homogeneous resources - complements and extends the cloud to the edge and endpoints. Fog infrastructure includes data centers, the core of the network, the edge of the network, and endpoints. Fog–as well as the cloud–supports co-existence of different tenants' applications. Each tenant stipulates its topology and a virtual topology is allocated to it. The fog has three key resource classes, namely, computing, storage, and networking. The fog needs scalable virtualization in the aforementioned areas which can be obtained by a Virtual File System, a Virtual Block, and appropriate Network Virtualization Infrastructure. Cloud enjoys provisioning mechanism based on a policy. Similarly, fog uses a policy-based orchestration and provisioning mechanism on top of the resource virtualization layer to automatically manage resources.

Fog should alleviate the problem of seamless resource management across a variety of platforms. Fog platform hosts a diverse set of applications belonging to various verticals–smart connected vehicles to smart cities, industrial automation, to name but a few. The architecture provides data and control APIs that can be used by many applications. Data APIs used to access fog data store while control APIs allowed to specify how an application should be deployed.

Bonomi's platform is illustrated in Figure 10. Fog network infrastructure is heterogeneous ranging from high-speed links connecting enterprise data centers and the core to multiple wireless access technologies. In order to increase resource utilization, fog abstraction layer specifies the ability to run multiple service containers on a physical machine and provides security, privacy and isolation policies. Service orchestration layer provides the opportunity to manage services on a large volume of Fog nodes with a wide range of capabilities by using several Foglet agents. Physical machines' health and state are monitored by abstraction APIs provide by these Foglets. Bonomi's platform enjoys messaging bus to carry control messages for service orchestration, resource management and distributed database which is ideal for increasing fog's scalability and fault-tolerance.

D. AUT REFERENCE ARCHITECTURE

Habibi *et al.* [53] proposed a reference architecture for fog computing based on the extension and integration of ETSI NFV architectural framework [68] and ONF's SDN architecture [69]. To our knowledge, this is the first standard

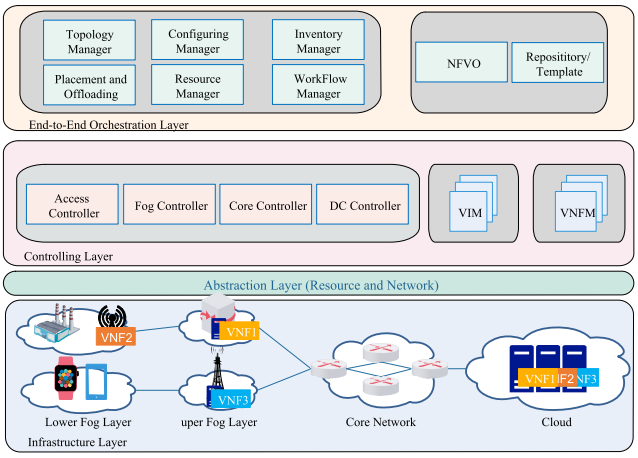


FIGURE 11. AUT reference architecture.

supported architecture proposed for fog computing. They also elaborated on the interfaces between system components and defined open interfaces that are mostly derived form OpenStack open APIs [70]. The five main components in this architecture, shown in Figure 11 are: infrastructure layer, resource abstraction layer, and control&management, application&services, orchestration layer.

The infrastructure layer has four main parts: ground (end devices), fog (hosting fog VNFs), cloud (data centers) and network (communication infrastructure). The fog nodes can be both tiny fog nodes with a small amount of network, storage and computational capabilities (such as servers, network devices, vehicles and smart devices) and rich fog nodes that have high network, storage and computational abilities (such as micro-clouds, cloudlets and base stations).

The resource abstraction layer contains an abstraction of both network and computing resources. This layer provides a high-level generic API to hide the heterogeneity and complexity of the underlying physical infrastructure.

The control&management layer comprises network SDN controllers as well as virtual resource management. SDN controllers are responsible to configure and manage network elements and provide generic APIs for monitoring, provisioning and managing resources. Federated SDN controllers offer scalability and reliability, allow incremental deployment, decrease complexity, and offer fine-grained privacy. The resource management module is responsible for physical inventory, virtualization and VNF life cycle management as well as monitoring and performance measurement.

The application&services layer deals with the user applications and services as well as virtual functions (VNFs) that are deployed and executed on the physical/virtualized resources by control&management layer. The last layer is End-to-End Orchestration (E2EO) that logically provisions the control&management layer according to the demands received from the application&services layer. This includes defining the network slices, network connectivity, allocating, instantiating, activating and administering the network

functions and resources that are required for an end-to-end service delivery. In this paper, inventory manager, resource manager, NFVO, topology manager, and placement manager are defined as the essential functional blocks of E2EO.

E. SUPPLEMENTARY REFERENCE ARCHITECTURES

Velasquez *et al.* [71] proposed a hybrid approach for service orchestration which is called SORTS. The SORTS infrastructure is divided into three layers: (1) IoT, (2) Fog, and (3) Cloud. The IoT layer includes Virtual Clusters (VC) that represent a group of communication terminals. The architecture presented in Figure 12 shows the orchestrator components which are used to manage and orchestrate the resources and functions. The overlapped instances of the architecture are to be replicated at different fog Instances and VCs that allow the use of the distributed choreography mechanisms; and also, at the cloud layer, a single instance is deployed for global orchestration. The Orchestrator is composed of different modules. The Communication Manager handles communication among the different orchestrator instances. The Resource Manager monitors the resource usage of the infrastructure. The Service Discovery enables the lookup of services available in the nearest location. The Security Manager provides different authentication and privacy mechanisms. The Status Monitor keeps track of activities in the system. The Planner Mechanisms schedule the systems' processes and the location where they will be placed. Finally, the optimization mechanisms which are meant to be applied at the upper layer, are used to improve the performance of the system.

SOAFI proposed by Brito *et al.* [72] leverages TOSCA and NFV MANO to build a reference architecture for fog computing. The architecture mainly designed as a client-based architecture, which is shown in Figure 13. It includes two main elements: Fog Orchestrator and Fog Orchestration Agent. The Fog Orchestrator (FO) is a centralized entity that forms the fog nodes into the logical groups called Logical Infrastructure. Using this formation, it is possible to handle multiple domains and perform federation. The responsibilities of the FO are divided into Infrastructure Management, Orchestration, Security, and Monitoring. Infrastructure Manager

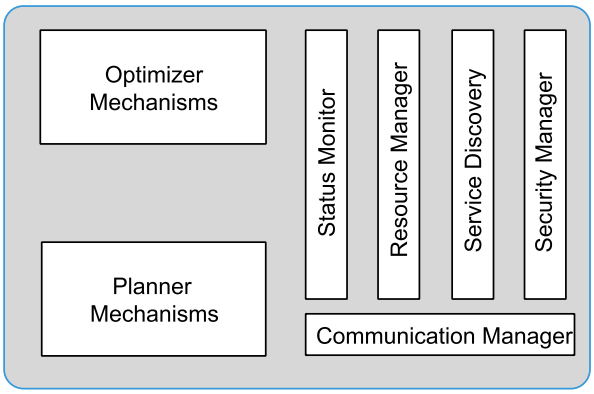


FIGURE 12. Hybrid orchestration architecture proposed in [71].

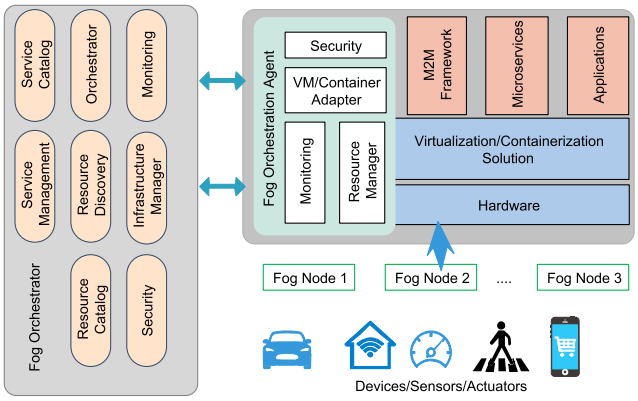


FIGURE 13. Fog orchestration components in [72].

makes an inventory of available resources in the fog domains, perform resource discovery and allocation. The information generated by the infrastructure management components is fed to the Orchestrator to perform various activities. The Monitoring module provides the required information for the Orchestrator such as topology of fog nodes formation within the logical domains. Each cloudlet includes a Fog Orchestration Agent (FOA) which provides a management interface to the Fog Orchestrator.

F. COMPARATIVE EVALUATION OF THE PROPOSED ARCHITECTURES

The proposed reference architectures address the fog computing's architectural challenges in different ways. The main challenges can be listed as support for federation, scalability, heterogeneity, orchestration, scheduling, discovery and allocation, latency, interoperability, resilience, prediction and optimization, path computation, security and privacy and AAA services. Table 4 summarizes the main characteristics of the reviewed architectures. The way these challenges are addressed heavily impacts the system's quality of service (QoS) and end user's quality of experience (QoE). This represents a currently active and challenging field of research to support the requirements of applications and services in the fog and improve service performance and user satisfaction. It should be noted that AUT and SORTS are the two most comprehensive architectures while other architectures provide innovative solutions in a select number of areas. Scalability and interoperability are the two most challenging issues that require open APIs, ontologies and standard description languages.

VI. APPLICATION-SPECIFIC ARCHITECTURES

Fog computing is considered to be applied in various fields and industries. IoT is considered to be the main application area for fog computing. A comprehensive review of fog computing applications in the context of IoT vertical markets can be found in [14]. 5G technology is also positioned to be the main enabling infrastructure for IoT applications. In this section, firstly, 5GPPP multi-tier computing

TABLE 4. Comparative review of the reference architectures.

Challenges	AUT [53]	SORTS [71]	CloudLab [12]	SOAFI [72]	Cisco-Bonomi [9]
Scalability	It is supported only at management and control plane using hierarchical design	N/A	N/A	N/A	Using The distributed message bus to provide scalable management channel
Orchestration	Yes	Yes	N/A	Yes	Yes
Heterogeneity	It is handled using virtualization and abstraction layers	N/A	Supported at the IoT device layer	Supported by fog agents within the level of fog nodes	Supported by fog abstraction layer within the level of fog nodes and physical resources
Scheduling	It is performed using VIM, VNFM, and VNFO	Using Planner mechanisms to schedule processes and their locations	It is performed using SDRM	Service management and catalog done by the Orchestrator	N/A
Path Computation	It is computed by SDN controllers	N/A	N/A	N/A	N/A
Discovery and Allocation	Service discovery module at the Orchestrator	Service discovery mechanisms at the Orchestrator to enable lookup	N/A	Handled by infrastructure manager	N/A
Interoperability	Interfaces are standardized using Openstack and REST APIs	N/A	N/A	M2M interoperability through standard communication	Using generic APIs (Orchestration and Data APIs) but detail is not clear
Latency	Service placement module at the Orchestrator	Service placement mechanisms at the Orchestrator	N/A	N/A	N/A
Resilience	It is supported using topology manager module	Survivability mechanisms at the Resource Manager	N/A	N/A	N/A
Prediction and Optimization	N/A	Global mechanisms to improve performance of the system	Supported using Knowledge Base system	Set of policies for virtual environments	N/A
Security and Privacy	N/A	Security manager provides different privacy mechanisms	N/A	Data security mechanisms as dependencies of the Orchestrator	N/A
Authentication, Access, control, and Accounting	It is handled using keystone and LDAP	Authentication mechanisms supported by its security manager	N/A	N/A	N/A

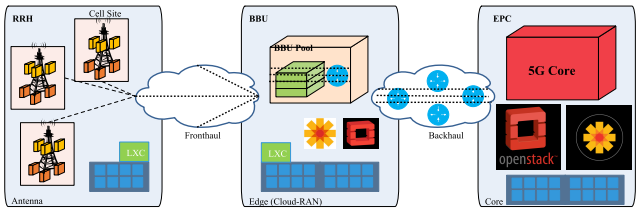


FIGURE 14. A multi-tier computing model in 5G.

model for 5G networks is presented. Then, IEC's general computing architecture for IoT and its main application-specific architectures–such as smart cities, healthcare, connected vehicles and fog-of-energy–are surveyed.

A. 5G-PPP COMPUTING ARCHITECTURE

Compared to existing 4G networks, 5G networks are aimed to achieve many-fold growth in network capacity and the number of supported connected devices, as well as better energy efficiency [73]. A multi-tier computing model adopted from 5G-PPP's proposed computing architecture is shown in Figure 14 To improve performance of 5G networks in terms of spectral and energy efficiency, enable direct device-to-device wireless communications, and support the growing trend of network function virtualization and separation of network control intelligence from radio network hardware, 5G will use the benefits of the centralized core cloud, cloud RANs at the network edge and cell-level distributed mobile cloud. This will create opportunities for companies to deploy many new real-time services that cannot be delivered over the existing mobile and wireless networks [74].

C-RAN incorporates cloud computing into RANs [75], [76]. However, the application storing and all radio signal processing functions are centralized at the cloud computing server in 5G core. However, billions of smart user devices need to transmit and exchange their data fast enough with 5G core, which requires high bandwidth and low latency. To avoid massive, backhaul capacity, one solution is to host 5G core functionality at the edge nodes close to cell sites. Content servers (or caching servers) can then be placed next to the distributed 5G core that can help significantly reduce backhaul traffic by having mobile devices download content immediately from the content server without having to pass the backhaul to reach 5G core.

Distributing 5G core closest to mobile devices is also the best way to achieve minimal end-to-end latency which is needed in mission-critical ultra-reliable and low latency applications such as remote controlled machines and autonomous driving. With fog computing, the C-RAN and 5G Core functionalities can also be invoked to take full advantage of local radio signal processing, cooperative radio resource management, and distributed storing capabilities in edge devices, which can decrease the heavy burden on front haul and avoid large-scale radio signal processing in the centralized baseband unit pool [77], [78]. The capabilities of cloud and fog computing can be spread even to the smart user devices, such as smartphones, IoT devices, sensors, etc. The devices form a local distributed peer-to-peer mobile cloud, where each device shares the resources with other devices in the same local cloud [79].

Cloud and fog computing complement each other to form an inter-dependent service continuum. Due to the varying nature of user services and applications such as business, operation, management and control tasks, some processes are naturally more suitable to be carried out in a centralized cloud, while some other processes are better suited to be performed at the fog level which consists of the edge and network devices between the end systems and the cloud.

B. IEC IoT COMPUTING ARCHITECTURE

The International Electro Technical Commission (IEC) with ISO has developed a reference architecture for IoT systems that defines various components in an IoT deployment. In IoT applications, various fog nodes may retrieve data from IoT devices and store them locally. When IoT applications need to access the data, their request should be forwarded to the relevant fog node with the aim of decreasing response time and delay. Hence, an efficient coordination and query processing is needed. Also, application developers need a set of standard

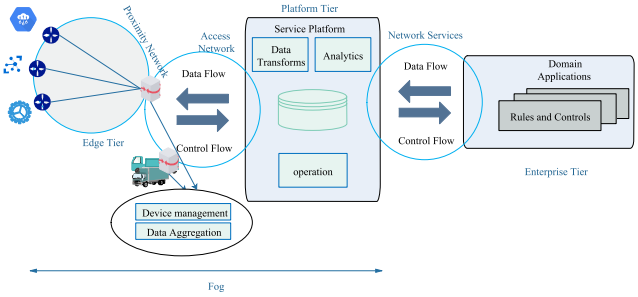


FIGURE 15. IEC multi-tier fog conceptual model for IoT.

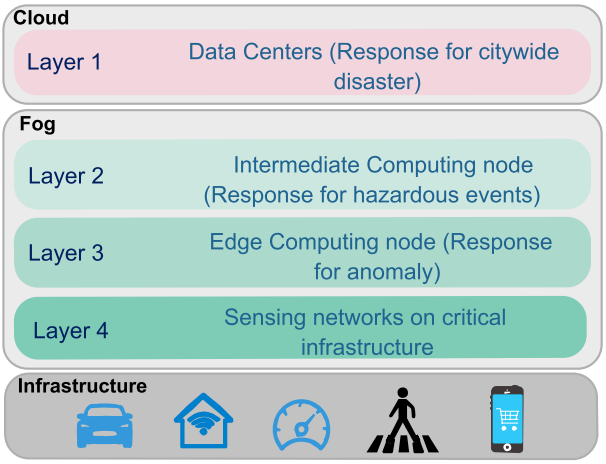


FIGURE 16. The 4-layer Fog computing architecture in smart cities, in which scale and latency sensitive applications run near the edge.

APIs that enable software designers to work in heterogeneous environment.

A 3-tier computing architecture shown in Figure 15. It includes edge, platform, and enterprise tiers that are connected by proximity, access, and service networks. The edge tier uses the proximity network to collect data from edge nodes that are the device or "things" deployed in the environment. This data is forwarded over the access network to the platform tier that uses the service network to communicate with enterprise tier. The platform employs various applications and components to provide fully interoperable IoT services and management of those services that includes controlling the physical devices as well as processing and relaying control commands from the enterprise tier back down to the edge tier. The enterprise tier provides end-user interfaces, control commands and domain-specific applications. It integrates the IoT functionalities with back-end applications such as CRM, ERP, billing and payments. It is generally accepted that real-time control and operation functions should be built near the edge devices. These low-latency functions are implemented at the edge and platform tiers in IEC's computing architecture.

In the fog computing conceptual model in Figure 15, the edge and platform tiers form the fog layer. Here, gateways are shown as part of an abstract fog layer. The remote enterprise cloud provides storage and processing capabilities when they are not sufficiently available in the fog. Generally, fog instances will also be available on other network hardware, such as the routers at the internet service providers (ISPs), providing the means to analyze and process data within the network closer to the end-user than in a centralized remote cloud. More powerful fog nodes may also offer the provisioning of virtual machines [80].

Local processing on gateway devices is made possible by advanced IoT hardware, which includes powerful smartphones and embedded single-board computers. This multi-tier architectural formation allows us to develop novel innovative services and applications. Some of the benefits gained by moving computation closer to the edge are as follows:

Minimizing Latency: Some IoT use cases involve a basic control loop; that is, on a certain condition, a certain action is triggered. These control loops, however, are often extremely time-sensitive. Analyzing raw data and making decisions locally or close to the data source can greatly reduce the latency of those control loops, since the distant cloud services will not be in line.

Improving Reliability: IoT includes using sensor data for public safety or to control critical infrastructure. The uplink to the distant cloud could turn out to be easily breakable. It would be valuable to have local processing as a fallback option, or exclusively use local processing altogether. Examples include industrial control loops or emergency response systems.

Addressing Privacy Concerns: Some IoT data will be sensitive or required by law to not be stored outside specific geographical boundaries. While cloud service providers are usually considered to be trusted, the user has no control over where the data are actually stored and who can access it. Local gateway and edge nodes are however under the control of the local operator and may provide better trust.

Conserving Bandwidth: The uplink bandwidth of IoT gateways is often severely limited, such as DSL or a 3G connection. It is not always feasible to transport vast amounts of data from edge devices to the cloud. Performing data processing locally and sending aggregated and filtered data to the cloud can significantly reduce the uplink bandwidth needed.

C. FOG COMPUTING ARCHITECTURE FOR SMART CITY APPLICATIONS

Smart city applications such as smart building, smart traffic and pipeline monitoring are one of the main applications areas of fog computing.

Tang *et al.* [61]. proposed an architecture for smart city applications as shown in Figure 16 The goal of this architecture is to support a huge number of infrastructure domains and services in future smart cities. Computing resources are organized into four layers. Sensors and IoT devices are deployed in the infrastructure layer. Layer 3 is composed of parallelized

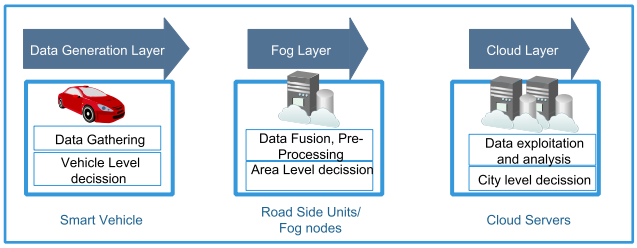


FIGURE 17. Fog computing platform for connected vehicle applications proposed by Huang *et al.* [82].

small computing nodes, or edge devices which performs two main computing tasks. The first task is to identify potential threat patterns on the incoming data streams from sensors using machine learning algorithms, which is called anomaly detection, and the second one is to perform feature extraction for further analysis in upper layers. Layer 2 consists of several intermediate computing nodes. Each node from this layer is connected to a set of edge nodes in layer 3. This layer includes components that make quick responses to control the infrastructure when hazardous events are detected. The result of data analysis, which is derived from the layer 3 and layer 2, is sent to the cloud layer. The cloud layer includes modules that are designed for high-latency computing tasks such as detection and prediction of long-term natural disasters. A prototype implementation was built but QoS and scalability were not discussed. The mobility of the sensors and/or the fog nodes is also not taken into consideration. Also, they do not provide any data model or standard interface for interoperability between modules and layers.

Brzoza-Woch *et al.* [81] also proposed an architecture for advanced telemetry systems that are able to support an automated flood risk assessment system. The proposed architecture includes three layers. The bottom layer includes sensors and their related networks. The authors proposed an edge computing layer in the middle, consisting of many distributed telemetry stations. It also includes components that collect data from the measuring layer, processing it, and sending the data to the central cloud. The cloud layer includes the communication layer which provides communication between the edge computing layer and the central cloud. In this work, which does not provide implementation detail, the heterogeneity of the fog nodes and cloud are not discussed. MQTT protocol is employed to transmit data from a telemetry station to the fog layer which provides a scalable communication channel and also a standard interface between these layers.

D. FOG COMPUTING ARCHITECTURE FOR CONNECTED VEHICLES

Vehicular related applications are one of the potential scenarios for fog such as the integration of fog computing with conventional vehicle ad-hoc networks (VANET) to form the Internet of Vehicles (IoV) or vehicular fog computing. In [82], they proposed an architecture which considered vehicles as intelligent devices that are mobile and equipped with multiple sensors and have the computational and communication capability to gather useful traffic information. The information can be collected from both intra-vehicle sensors and the environment. In this architecture, fog nodes can be deployed at the edge of vehicular networks in order to make the data collection more efficient together with processing, organizing, and storing traffic data in real time. They defined three main layers namely, the smart vehicles which collect data, the roadside units/fog nodes as the fog layer, and the cloud servers as the cloud layer (Figure 17).

Smart vehicles have an important role as data sources in a vehicular fog computing system, due to their real-time computing, sensing (e.g., cameras, radars and GPS), communication, and storage capabilities. The amount of data collected by the various sensors in a smart vehicle has been estimated to be around 25 GB/h in a single day. Some of these data can be processed by the smart vehicle, in order to perform real time decision making, while the rest of the data will be shared and uploaded to the fog nodes for further analysis such as traffic control planning. The roadside unit can be deployed in different spots in the city as a fog node which enables the platform to process collected data and send it to the cloud servers. This functionality can be extended as a middleware system that makes a connection between the cloud servers and the smart vehicles in a vehicular fog computing system. Unlike existing vehicular networks, these units/nodes will have more functions and provide more diverse services for mart vehicles, such as navigation, video streaming, and smart traffic lights. Cloud servers enable city-level monitoring, permanent data storing, and play a role as a centralized control system. These servers will obtain the data from all fog nodes to make globally optimal decisions. For example, they will monitor, manage, and control the city's road traffic infrastructures to achieve optimal city-level traffic control. In this paper, two use cases of fog computing based vehicular application, Local Traffic Control Subsystem and Global Traffic Management Subsystem are discussed.

Hou *et al.* [83] proposed an architecture called Vehicular Fog Computing (VFC) for vehicular applications. The authors conducted experimental analysis to study the impact of mobility on the vehicular network especially the connectivity and computational capacity using VFC. Their results show a great enhancement over traditional architecture, which was called Vehicular Cloud Computing (VCC). In this work, vehicles are the IoT devices and, at the same time, these vehicles act as the fog nodes; hence called smart vehicles. Smart vehicles support two kinds of communications. It can either perform vehicle-to-vehicle communication (V2V) or to the infrastructure, which is called vehicle-to-infrastructure (V2I). The fog layer connects to the cloud layer through RSUs. In this paper, there is no discussion on the heterogeneity of the fog nodes. The authors have only considered a unique type of the fog node. They indicate that VFC maintains communication continuity even when fog nodes enter new fog domains. However, they do not provide any details about the architectural module(s) responsible for mobility

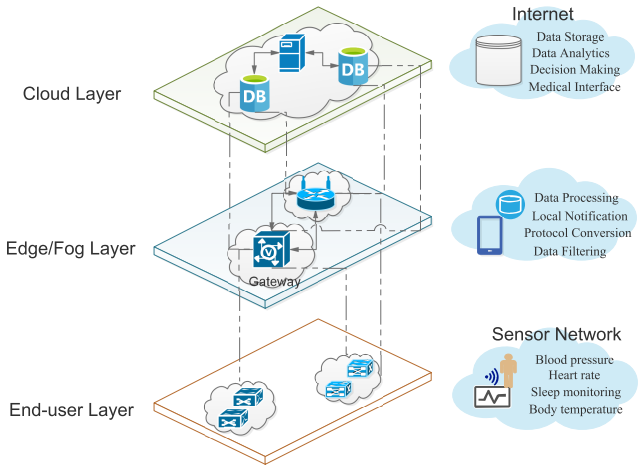


FIGURE 18. General architecture of fog based IoT health monitoring system [85].

management. Moreover, the authors acknowledge that there is a need for mobility models to build efficient VFC. Finally, the specification and the data interfaces that architecture's components used to interact with each other are not discussed.

E. FOG COMPUTING ARCHITECTURE FOR HEALTHCARE APPLICATIONS

IoT enables Electronic Health (eHealth), and Mobile Health (mHealth) that allow remote monitoring and tracking of patients living alone at home or treated in hospitals [84]. It is no longer sufficient to design standalone wearable devices, instead, it becomes vital to create a complete ecosystem in which sensors in a body area network seamlessly synchronize data to cloud services through the IoT infrastructure [85].

Given the large number of connected devices, the latency of the connection with the cloud could be significant. Moreover, these devices are power and bandwidth constrained, that make them unfit to directly connect to the cloud architecture.

As reliability in e-health application is of utmost importance and even short system unavailability often cannot be tolerated, the limited resources of medical sensor nodes render the use of general purpose gateways inefficient in most circumstances with respect to delay, energy, and reliability.

Fog Computing is an essential paradigm shift towards a hierarchical system architecture and a more responsive design. As shown in Figure 18, Fog is an intermediate computing layer between the cloud and end devices that complements the advantages of cloud computing by providing additional services for the emerging requirements in the field of IoT.

The intermediate layer handles the Heterogeneity and Interoperability by aggregating the heterogeneous data models and providing standard formats for the upper layer as proposed in [86]. This work introduced an IoT-based health monitoring architecture to extract ECG features at the edge to recognize cardiac diseases. The proposed architecture uses fog to save bandwidth, guarantee QoS, and send emergency

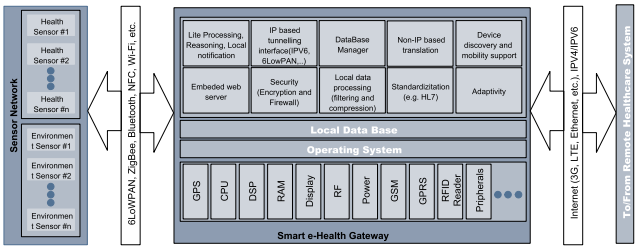


FIGURE 19. An architecture of fog gateway in health monitoring system [85].

notifications. They apply a Location Awareness module to keep track of patients to help them in case of emergency. The cloud layer permanently stores the patient's historical information and provides tools to perform data analysis.

Figure 19 illustrates a conceptual architecture of a smart e-health gateway utilizing the fog layer for data filtering, data compression, data fusion, and data analysis [85]. The sensitivity of the system is improved by applying local data analysis at the edge. It can assist the system to detect and predict emergencies. It also includes some other functional components such as hardware processing elements, networking protocols, and security modules.

F. FOG APPLICATION ARCHITECTURE FOR INTERNET OF ENERGY

Environmental concerns, alongside growing fuel prices, are channeling research efforts to energy consumption in the computing systems and in particular in the fog ecosystems. Oueis *et al.* [87], Mohamed *et al.* [88], Sarkar *et al.* [63], Sarkar and Misra [89], Jalali *et al.* [90], and Cao *et al.* [91] have analyzed and assessed energy consumption in the fog ecosystem architecture. Some other works - such as Oueis *et al.* [92], Deng *et al.* [93], Ye *et al.* [94], Xiang *et al.* [95], and Chen *et al.* [96] - have considered the design of strategies aiming at reducing energy consumption in fog systems.

Sarkar *et al.* [63], Sarkar and Misra [89], and Jalali *et al.* [90] focused on the analysis of energy consumption in the fog architecture. They compare several metrics including power consumption, service latency, and CO2 emission in a fog ecosystem compared to the cloud-based solutions. By simulating real-time IoT services in 100 cities served with 8 data centers, the authors conclude that fog computing is more efficient than cloud computing. From a latency perspective, they observe that with 25% of applications requesting real-time services, the service latency decreases by 30%. In turn, the power consumption decreases by 42.2% and CO2 emissions decrease by more than 50%, translating into significant reductions in cost. Mobility is not considered here that can potentially have a great impact on QoS parameters. Jalali *et al.* [90] considered nano data centers that form their fog nodes. The results show that nano data centers are more energy-efficient than the cloud data centers by pushing content close to end-users and decreasing

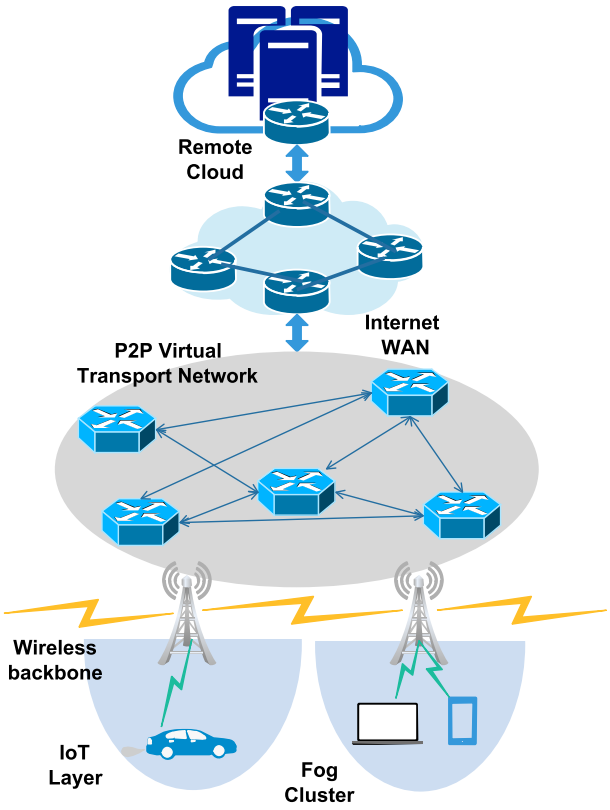


FIGURE 20. Fog Of Everything (FOE) architecture.

the energy consumption in the network. However, scalability was not considered since the evaluation results are obtained through the small-scale scenarios without support for mobility.

In the same context, the concept of energy-efficient Fog of Everything (FoE) platform was proposed by Baccarelli *et al.* [97]. In this application architecture (Figure 20), fog devices (IoT sensors, smart car, smartphone or any other station) are connected to the wireless base station via Fog to Things (F2T) and Things to Fog (T2F) two-way connectivity through TCP/IP connections functioning onto IEEE802.11/15 single-hop links. All fog devices connected with the same base station are considered to be in the same cluster. All base stations are considered as Fog nodes and will be connected via Fog to Fog (F2F) links by the inter-Fog physical wireless backbone. Container-based virtualization is used to make a virtual clone associated with physical things. The virtualization layer supports the efficient use of limited resources and generates the virtual clone of physical things. The Fog node physical server serves cloned things and an overlay inter-clone virtual network was established which allows P2P communication among clones depending on TCP/IP end to end transport connections. In this work, the authors considered the heterogeneity in the IoT devices and fog nodes; however, they did not discuss if their architecture is scalable in terms of the number of sensors, the fog nodes, or the fog domain.

VII. SUPPLEMENTARY ARCHITECTURAL ASPECTS

This section covers complementary architectural aspects of fog computing in the areas of application, software, networking, computing resource management and security.

A. APPLICATION ARCHITECTURE

Fog computing is introduced to support time-sensitive and real-time applications in a more efficient way considering bandwidth and latency limitations. In general, in this system, we are dealing with hybrid applications using both fog nodes and cloud data centers in a QoS-aware and context-aware fashion. Designing these applications is a challenging task due to the vast heterogeneity, scale and dynamicity of fog computing infrastructures. In a multi-tier fog architecture, mission-critical applications could be processed at fog layers close to the cloud layer for higher reliability and security. Real-time interactive and streaming applications must be processed as close to the end-user as possible. The upper fog layer is able to perform more intense processing such as deep data analysis applications. In CPU-intensive applications, which need a huge amount of processing resources, all layers of the fog may be involved. Best-effort applications such as e-mails can be processed in the cloud since there are no delay constraints.

In Figure 21, a view of a fog-assisted multi-tier computing environment is shown which is adapted from [9]. A real example is studied in [63] where a quantitative analysis of energy consumption in an IoT application is performed. Their results indicate that when 25% of the IoT applications demand real-time and low-latency services, the mean energy expenditure using fog computing is 40.48% less than that in a conventional cloud computing. Evaluation results show that fog computing is an improved, eco-friendly computing platform that can support IoT better compared to the existing cloud computing paradigm [64]. In [65], a different

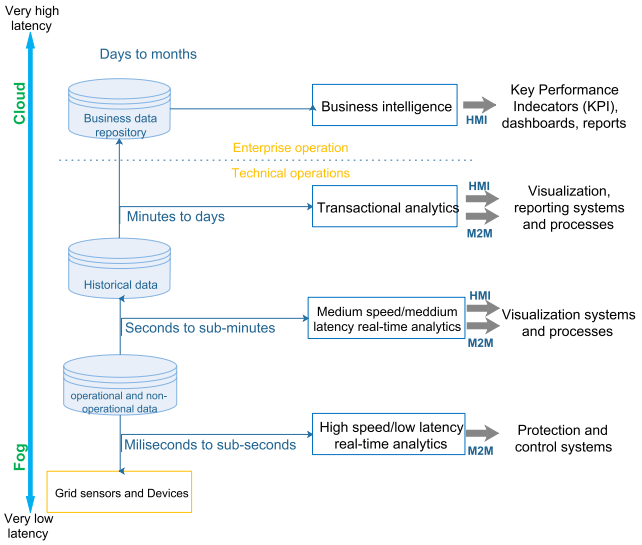


FIGURE 21. Multi-tier architecture of applications and services.

study compared the energy consumption of applications using centralized DCs in cloud computing with applications using Nano Data Centers (nDCs) based on fog computing. Flow-based and time-based energy consumption models for shared and un-shared network equipment were used. The results indicate that the best energy savings using nDCs can be attained for some applications that generate and distribute a large amount of data in end-user premises with low access data rate, such as video surveillance in end-users' homes. The tradeoff between power consumption and transmission delay in the fog-cloud computing system is also investigated in [66]. The segmentation of tasks into fog or cloud tasks is application specific. Simulation results in a medical emergency service use case demonstrate the benefits of a coordinated control and management of a combined fog and cloud system. Thus, the design of a coordinated orchestration and management to manage all cloud/edge resources in a joint framework system is critical [98].

B. SOFTWARE ARCHITECTURE

Fog computing is a novel computing paradigm, which demands a new programming model. It is required to design intuitive and effective tools and frameworks for developers, helping them orchestrate dynamic, hierarchical, and heterogeneous resources to build compatible applications on diverse platforms.

Some existing works focus on platform architectures and study their building blocks in the fog computing environment. A platform is a software environment allowing design and integration of the fog computing components. It provides solutions for some challenges such as heterogeneity, interoperability, security, and dependability [55], [88], [99].

In [100], the challenges involved in developing software platforms in fog computing is discussed and flexible architecture is proposed. The main challenges in developing fog computing software platforms are listed as follows:

1) Each node may provide a number of available services. Implementing automatic service discovery protocols in fog computing can be quite challenging.
2) Security and privacy considerations are complex in fog computing, and tasks from sensitive applications should be scheduled on more trustworthy nodes.
3) Data consistency can become complicated in fog computing ecosystem. When writing data objects in a fog environment, it's necessary to not only coordinate the back-end cloud servers, but also to invalidate the cached data on the fog nodes as well as on the client devices if strong data consistency is needed.
4) In task scheduling and migration as an example, some of the issues are how it is possible to provide a simple abstraction for developers to mark tasks that can be migrated, what choices and preferences should be delegated to users, how could be allowed developers to specify migration rules on heterogeneous devices.
5) Forcing developers to implement functionalities that will likely be common, such as distributed caching, workload balancing, system monitoring should be avoided.
6) Data management for fog computing applications also introduces new challenges. Perhaps the ideal abstraction for both users and developers is global storage, which can always be accessed, has infinite size, and yet performs with the speed of information stored locally. However, how to implement such a storage system is still an open question. What efficient algorithms can be used to shuffle data among devices? How can prefetching be best implemented to achieve the lowest latency? What namespace scheme should be used? How can sensitive and encrypted data be cached privately and effectively?
7) Furthermore, energy consumption and network usage must be conserved on mobile devices, as they typically have energy limits enforced by limited battery technology and data limits enforced by mobile carriers.

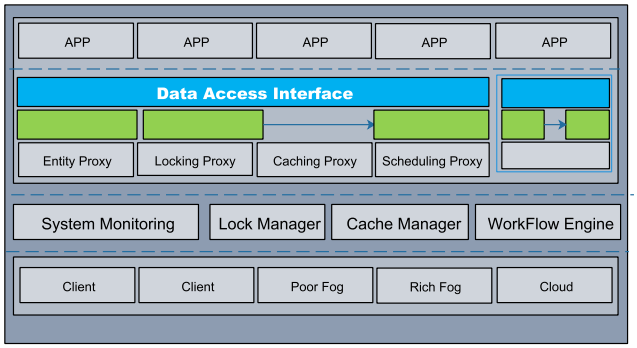


FIGURE 22. Proposed WM-FOG software architecture in [100].

Figure 22 depicts the architecture of WM-FOG. There are four layers in the figure. The design of WM-FOG embraces a flexible software architecture, which can incorporate different design choices and user specified policies. More specifically, WM-FOG provides a flexible way to define workflows that can be easily deployed and executed on fog-based systems. By properly scheduling the workflows on the system entities (that is, client devices, fog nodes, and back-end cloud servers), WM-FOG can take advantage of the fog computing paradigm and achieve considerable performance enhancement.

The top layer is the application layer, where user applications reside. User applications initiate workflow instances by writing input data to them, and receive results by reading output data from them. The next layer is the workflow layer, where workflow instances reside. Each workflow instance exposes a data access interface to user applications, through which its data items can be accessed. Moreover, each workflow instance has four proxies, that is, the entity proxy, locking proxy, caching proxy, and scheduling proxy. These proxies can be used to implement user specified policies on workflows. Under the workflow layer is the system layer,

where the system components – that is, the system monitor, lock manager, cache manager, and workflow engine – reside. These system components implement the fundamental mechanisms of WM-FOG, and workflow instances can communicate with them through the proxies to apply user-specified policies. The bottom layer is the entity layer, as the system entities (client devices, fog nodes, and the cloud) reside in this layer.

C. SECURITY AND PRIVACY ARCHITECTURE

Networked applications cover a broad spectrum of privacy-sensitive and mission-critical use cases including transfer of private information (such as photos, medical reports), daily routine tasks (such as shopping, transportation), and enterprise resource management (such as supply chains). Security and trust are essential for successful deployment of these applications. Security in a multi-tier computing environment involves new challenges that need to be identified and tackled. Several enabling technologies such as wireless networks, distributed and peer-to-peer systems, and virtualization platforms are employed. Such technology diversity calls for both protecting all these elements and also orchestrating the diverse security mechanisms in order to maintain the integrity of the ecosystem.

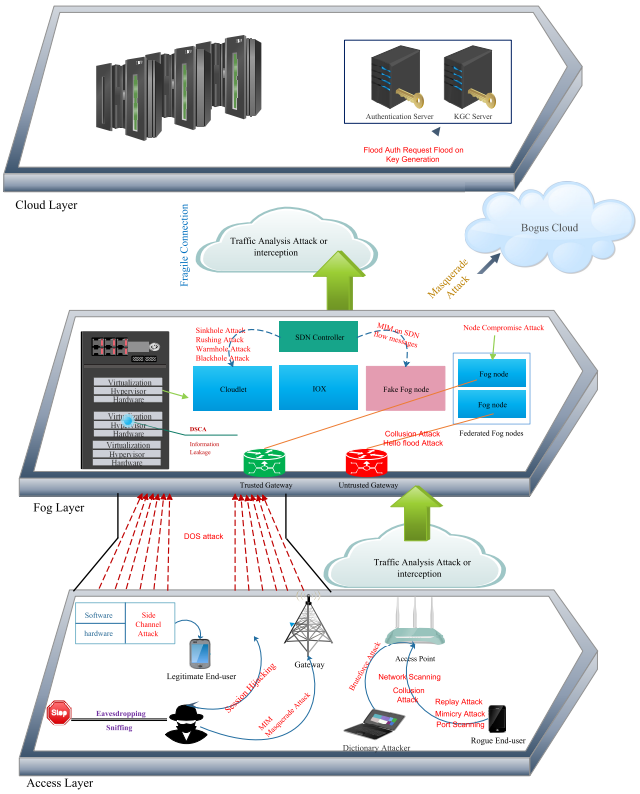


FIGURE 23. A layered security architecture for fog computing.

Figure 23 demonstrates our three-layer model for fog security. The flow of requests and data in the system is demonstrated and potential attacks in each layer are visualized. Fog-access is the point of connection to the end-user devices in the fog ecosystem which is responsible for end-device management and access control. Fog-computing includes all the computing nodes in the fog layer including computing at the access nodes. Vulnerabilities related to system integrity and availability and data confidentiality are addressed in this layer. Fog-cloud interconnection layer addresses potential threats in interconnecting the two subsystems.

1) ACCESS LAYER SECURITY

The access layer deals with authentication, authorization, access control and data protection of the edge devices connecting to the network in a widely distributed set up. Since traditional security mechanisms are covered in other studies, in this layer, specific issues in a fog ecosystem such as decentralized and distributed nature of edge paradigms, interoperability and mobility support, and location awareness are considered.

2) FOG LAYER SECURITY

The computing layer comprises all the nodes involved in sub-cloud computing. A computing node may also have access layer functionality. For example, fog nodes may be located at the edge of networks and can collaborate with each other in a distributed manner.

3) FOG-CLOUD INTERCONNECTION SECURITY

The cloud layer contains large number of servers that host the service applications and often can perform heavy computational processing on the data received from fog nodes. Interoperability of computing tiers and their secure interconnection is of utmost importance for the overall end-to-end security of the system. The security of the cloud layer itself as one of the main components in a multi-tier computing environment is also important but is separately discussed in other papers [101]–[105].

Table 5 summarizes the major attacks on a fog computing ecosystem based on ISO 27001 [106]. ISO 27001 is the most common standard and is a widely recognized structured methodology for information security. The impact of the major attack categories on the three dimensions of security KPI, namely, confidentiality, integrity and availability. DDoS mitigation, data integrity, key management, and single-sign-on are the primary issues that should be addressed.

D. COMPUTING RESOURCE MANAGEMENT ARCHITECTURE

The major complexity in multi-tier computing is the orchestration and management of computing resources in a coordinated and efficient way to utilize all the resources to support large-scale heterogeneous applications.

Figure 24 illustrates the proposed set of control and management blocks in a fog computing environment in [162]. The aim is to provide a coordinated and distributed management solution aiming at handling resources continuity from the edge to the cloud. The figure shows three main blocks.

TABLE 5. Summarization of major attacks on fog computing ecosystem.

Layer	Possible Attack	Violation of Security KPI		
		Confidentiality	Integrity	Availability
Access	Interruption [107]–[117]	✓		✓
	Sniffing attack [118]	✓		
	MiME [119]–[122]	✓	✓	
	Untrusted end-user [123], [124]	✓	✓	✓
Fog	Node compromisation [125]–[144]	✓	✓	✓
	Privacy preveserving [64], [65], [145]–[157]	✓		✓
	Virtualization [158], [159]	✓		✓
Cloud	Untrusted gateway [160]	✓		✓
	Interconnection security [161]	✓		✓

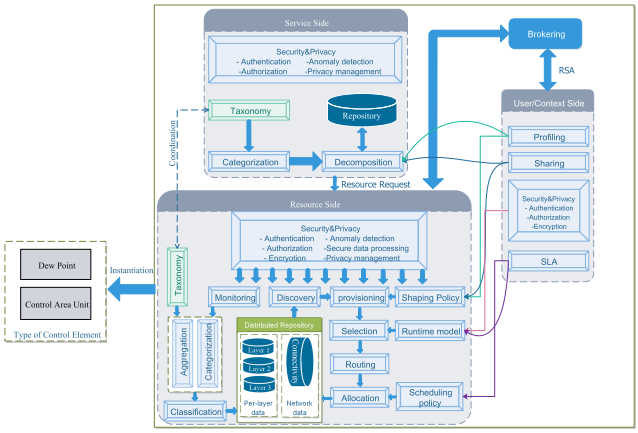


FIGURE 24. A distributed resource management architecture for layered fog to cloud services [162].

The first and also the largest block allocates the different functionalities that are expected to co-exist in reference to the user, service and resource-specific functions. The second block allocates the set of functions used for instantiations of the different modules, referring to the different control components and edge devices, i.e., the Dew Points (DP) and the Control Area Units (CAU). The last one, Brokering, is designed to be applied in a typical scenario of multi-ownership infrastructure, where different fog/cloud layers may belong to different parties, and a brokering concept is necessary for their joint deployment.

Resources are assigned to the service to be executed, depending on the service demands, runtime policies required (e.g., parallel or sequential execution, what unquestionably impacts the way computer, network and storage resources are selected) and resource availability (all managed through the functions Provisioning, Selection, Routing, Allocation, Runtime model as well as Scheduling and Sharing policies).

Resource management involves a number of components including VM and service placement, workload and task assignment, resource allocation, computation offloading and caching as shown in Figure 25 [163]. A brief review of current research in each of these categories is provided in the following.

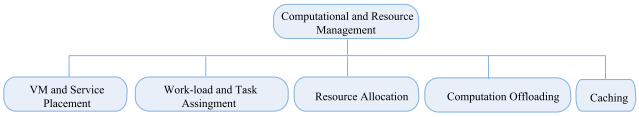


FIGURE 25. Computation and resource management taxonomy.

1) VM AND SERVICE PLACEMENT

Virtual Machine (VM)/Container and service placement is a critical operation for finding the best fog node to host the VMs or applications. It directly affects performance, resource utilization and power consumption.

Brogi and Forti [164] propose FogTorch, a general, flexible, and extensible platform to support QoS-aware IoT applications in fog infrastructures. Various algorithms are proposed to find suitable service deployment in the fog based on the QoS parameters. In addition, resource allocation and link capacity is taken into account.

Verma *et al.* [165] propose a multi-tenant heuristic algorithm to guarantee QoS for users through a load balancing algorithm. A three-layer architecture is considered that consists of End-user, Fog, and Management layers. Any fog node is managed by the fog management layer in a centralized manner. The algorithm includes two main components namely: Tenant Maximum Acceptable Delay (TMAD) and Tenant Priority (TP). For every application, the most appropriate fog node considering the tenant's maximum acceptable delay will be selected.

Most of the algorithms do not consider heterogeneity in terms of node computing and storage capabilities in the fog ecosystem. In a real world scenario, a significant degree of heterogeneity can produce a major impact on the performance and efficiency of the algorithms.

2) WORK-LOAD AND TASK ASSIGNMENT

Compared to cloud computing, edge and fog nodes do not have large computing and storage capacity. Therefore, efficient resource allocation is considered as an important research area in multi-tier computing. Deng *et al.* in [166] proposed a tradeoff between power consumption and transmission cost. They formulated a workload allocation problem considering load distribution between fog and cloud toward

the minimal power consumption with a constrained service delay. The computational complexity is tackled using an approximate approach by decomposing the primary problem into three sub-problems where each sub-problem is solved independently. Their simulation results vindicate the fact that the fog in conjunction with the cloud considerably reduces the overall communication latency. Due to the centralized nature of this work, the scalability of the framework, data replication overhead, and communication overhead must be reconsidered in a distributed set up.

Meng *et al.* [167] proposed an algorithm called hybrid computation offloading to minimize the total energy consumption in both cloud and fog environment. In this work, diverse computing and communication capabilities of cloud and fog are taken into account. To solve the problem, Computation Energy Efficiency (CEE) is defined as the amount of the computation tasks that are offloaded by consuming a unit of energy. The problem is broken down into four sub-problems. Based on the CEE and delay constraints, tasks are offloaded to the cloud, fog, or both.

None of the works in this area focused on mobility issues even though it is an important factor for fog resource management due to dynamic nature of computing nodes, which are constantly leaving and joining to the network. Thus, dynamic load balancing and task assignment become a very important challenge. To predict the pattern of end-users' mobility behaviors in order to assign tasks and manage resources in an effective way is an issue to be addressed. All of the works claim to have decreased the latency while delay, as an important criterion in this area, is not explicitly modeled.

3) RESOURCE ALLOCATION

Multi-tier computing substantially involves components of an application running both in the cloud and in the edge devices between sensors and the cloud, such as smart gateways, routers, or dedicated fog devices. These resources are pervasive and often vary dynamically. Therefore, a judicious management of resources is essential for maximizing the efficiency of the computing environment.

The work in [1] surveyed the resource allocation options for fog providers. Architectural frameworks for resource allocation in fog computing were presented in [46], [168]. Gupta *et al.* [169] formulated a toolkit, called iFogSim, to simulate IoT and fog environments and measure the impact of resource management techniques in terms of laten, network congestion, energy consumption and cost. Souza *et al.* [170] formulated the QoS-aware service allocation problem for combined fog-cloud architectures as an integer optimization problem. Their solution minimizes the latency experienced by the services and guarantees the capacity requirements.

None of the above solutions jointly considers the task completion time, user cost, and application performance to maximizing the service providers and users experience. Ni *et al.* [171] proposed a dynamic resource allocation strategy that comprehensively considers the capital and time costs as well as the credibility evaluation of both users and fog resources. Through this work, they improved the efficiency of resource utilization while satisfying the users' QoS requirements.

4) COMPUTATION OFFLOADING

Computation offloading is a related topic to the task assignment problem discussed earlier. Both have to do with decision on where to run each part of the applications but while task assignment is of an offline, proactive and centralized nature, the computation offloading takes an online and distributed approach. However, in the literature, they are used interchangeably in some cases. Computation offloading has been a hot topic due to the heterogeneous distributed nature of applications and resources. For example, offloading raw data processing to the edge devices in IoT diminishes the network load by reducing the amount of data sent to the cloud. Each application can be offloaded in coarse-grained application level manner [172], [173] fine-grained task level manner [174]–[178] or in parallel load-balanced manner [179], [180]. The offloading decision may be made in a centralized or decentralized manner [181], [182], for single end-point or multiple end-points.

Resource allocation and offloading were jointly optimized in [181] so as to conserve energy while satisfying end-point delay constraints. However, the energy consumption of each user equipment was set as a constant for simplicity, ignoring its time varying aspect. Computation offloading is not always efficient in terms of delay, bandwidth, and energy consumption. Offloading decisions should be made based the relative importance of these parameters in any application scenario.

5) CACHING

There are two main approaches in caching that can be used in fog computing: proactive and non-proactive caching. Proactive computing [183] has been studied in wireless content caching [184], where the content of the upcoming tasks is predicted and prefetched during the current task computation. As an alternative for proactive caching, in [95] the computing results are pre-fetched during an off-peak interval, thus reducing the backhaul burden. Elbamby *et al.* [185] suggested a proactive caching of popular and cacheable computing tasks considering both computing and storage resources to minimize service latency. Due to large network size, the end-users are clustered into disjoint groups based on distance-based Gaussian similarity and task popularity-based similarity. The cloudlet tries to minimize the latency by proactively caching the computation results. The cloudlet dynamically replaces computing results for the less popular tasks with results for more popular tasks. In this paper, a game theoretic approach is suggested between end-users and cloudlets to minimize the overall latency in cache-enabled networks.

Current research has mainly focused on the case of fog RAN and the heterogeneity and QoS are considered as the main focus of the research; however, the scalability in terms of elasticity and the mobility support are not fully considered

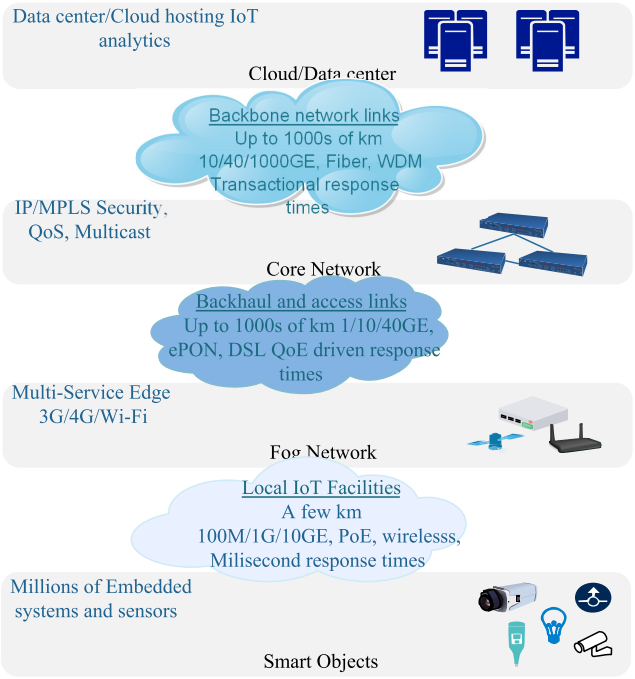


FIGURE 26. The heterogeneous network architecture in fog computing [186].

E. NETWORKING ARCHITECTURE

Ubiquitous and decentralized objects with various communication technologies should be connected to the computing ecosystem. Hence, the system is considered as a heterogeneous network environment that is responsible for connecting its end-users to the appropriate fog nodes and also the fog nodes to each other and to the cloud. Such a heterogeneous view of networking in a fog computing environment is illustrated in Figure 26 [186]. In the bottom layer, technologies such as wireless sensor networks (WSN), Zigbee and Lorawan is used to provide direct connection among sensors and/or their gateways. In a layer above, edge fog nodes are connected through mobile networks or wireless local area network technologies such as many variations of WiFi. Self-organizing networks, such as Mobile ad hoc Networks (MANET), are one of the main candidates for the future fog networking since they will enable the formation of densely populated networks without requiring available fixed and costly infrastructures beforehand [187]. In the upper layer, fixed networking technologies such as IP/MPLS infrastructures connect network devices and data centers at high speeds.

Software Defined Networking (SDN) and Network Function Virtualization (NFV) are enabling technologies for policy-based fog networking. SDN decouples the data plane and the control plane which is a centralized network-wide operating system and provides open APIs for network applications and services. SDN can benefit fog computing by enhancing resource sharing and isolating network traffic. SDN can also help integrate fog and cloud infrastructures by providing a global network view. A SDN controller itself can

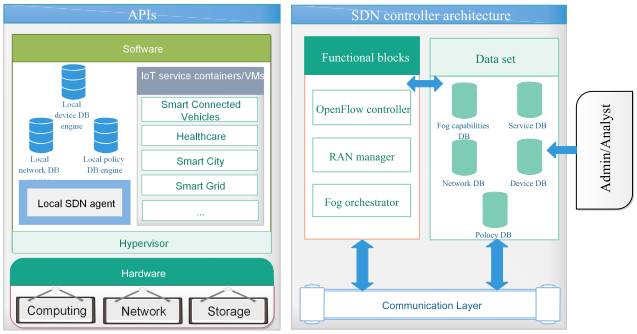


FIGURE 27. The internal architecture of fog nodes and SDN controllers in an SDN fog computing environment [186].

be used as a fog device [188]. NFV is a network architectural concept that takes advantage of virtualization technology to transform network functions from hardware-based entities to software agents. It helps virtualize functions such as routing, load balancing, and intrusion detection to be placed at the fog nodes. The challenges of NFV in fog computing include the performance of virtualized network appliances, specifying location of services, and defining services migration policies. The general internal architecture of fog nodes and SDN controllers is shown in Figure 27. The fog node contains an SDN agent to receive per-flow routing policies and the controller has a fog orchestrator module in addition to the controller module. The fog orchestrator manages device and service configuration such as IP address assignment and NFV placement and onboarding on fog nodes.

In [189] also a new VANET architecture, called FSDN, is proposed to integrates two emergent computing and networking paradigms, fog computing and SDN, as a prospective solution. They claimed that their architecture could resolve the main challenges of VANETs by augmenting Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Base Station communications. SDN takes advantage of its global view to optimize network resources. As a consequence of this optimization process and integration of SDN with fog computing, latency is also reduced.

In [190], an integrated three-tier programmable framework for heterogeneous networks has been introduced. It contains caching, computing, and data and control communication. This framework maximizes resource utilization, enhances users' experience, and decreases energy consumption. They argue that challenges related to integrating SDN into a fog network are to deal with node mobility, unreliable wireless links, wireless network virtualization, resource reservation based on end-user's privilege, maintaining the connectivity graph of the network in controller, and placing controllers in the fog network to meet the end-users' demands.

VIII. EVALUATIONS AND FUTURE DIRECTIONS

In order to evaluate the current state of the technology, and identify the potential future direction, in the following, a mature model that quantitatively measures the maturity

TABLE 6. Maturity level in six main subject areas and future directions.

Subject Areas	Well-versed Ontology	General Architecture	Reference Architecture	Universally accepted Components	Standard APIs and interfaces	Maturity Level	Future directions
Computing Paradigm	✓	✓	✓	✓	X	4	Existing architectures lack standard system level interfaces. Interoperable and reliable solutions for system management, monitoring and accounting is needed. Federation frameworks should be developed.
Applications	✓	✓	X	X	X	2	Well-defined logical functional components in major application areas to be identified. To move from proprietary interface solutions to standard APIs. Heterogeneity, scalability and interoperability challenges needs to be addressed through open system models. Standard solution for object identification, discovery and service composition is needed.
Software	X	✓	✓	X	X	2	Industry wide convergence of terminologies and software models is overdue. Well-developed platform solutions are needed.
Networking	✓	✓	✓	✓	✓	5	Proven solutions for security and privacy concerns as well as mobility management are needed.
Computing resource management	X	✓	✓	✓	X	3	Existing heterogeneity in various aspects of resource management is a hurdle on the way of scalable system deployment. Well-versed ontologies and reference architectures for convergence of efforts on resource management rather than disparate point solutions. Automatic service orchestration and resource management are needed.
Security	✓	✓	X	✓	X	3	Comprehensive reference architectures and open security workflows are needed. Open and standard APIs among various components to be developed.

level of the technology in the scale of 0 to 5 is proposed. It is based on 5 main evaluation criteria as follows:

1) Existence of well-versed ontology: To help define the terms and subjects in a clear and unambiguous way, to identify the challenges and issues and classify the solution spaces. Existence of comprehensive surveys and ecosystem designs are key required steps.
2) General architecture: To define the ecosystem layers and physical components as well as their interfaces and interactions.
3) Reference architecture: It defines the logical components, the interfaces and workflow/data flow in the system.
4) Universally accepted components: Existence of universally accepted components is a step toward consistent and interoperable development and deployment of the functional modules.
5) Standard APIs and interfaces: The components need to work together in a scalable manner for which standard APIs and interfaces could help.

The proposed maturity model is applied to the six main dimensions of fog computing technology as outlined in Table 6. The aim is to evaluate and compare their state of the art and to identify major future directions. We have used a subjective evaluation based on the existing literature to identify major research and development gaps. A more detailed evaluation could be the subject of future works. Developing standardized application architectures and software platforms stand out as the two most urgent areas that need attention. Heterogeneity, scalability, and interoperability are the three main issues identified in Table 6 to be handled in developing such platforms and architectures.

In terms of heterogeneity, most of the proposed architectures provide potential solutions for the heterogeneity challenge in fog computing; however, none has proposed semantic-based approach. In fact, most of them put the burden of handling heterogeneity on application side. A semantic-based approach is considered to be an explicit formal and standard specification of the architecture used in a given domain. Works [67] and [53] can be used for formal representations and naming of the properties, types, and relationships of the entities that set up a particular fog environment. In an specific fog ecosystem, defining appropriate ontologies that could also help describe the strong variety and specificities of the involved nodes from the cloud, fog, and IoT. It would contribute to the homogenization, standardization, and simplification of the applications that are provisioned over these distributed nodes. Indeed, several providers, as part of a fog system, might rely on heterogeneous description models, schemes, naming, and vocabularies. For instance, devices in the IoT world can be described by models such as the one proposed by [191] while the nodes in the cloud can be described by models such as the OASIS TOSCA. Obviously, the heterogeneity of the models is unsuitable for collaborative environments such as the fog system where providers from all environment and domains need a common understanding of resources in the time of provisioning applications. Consequently, research is needed in the design of appropriate and exhaustive ontologies to support fog heterogeneity.

Scalability is another important issue with connotations on all aspects of the system design. Although several architectures in the literature have explicitly addressed the scalability; however, the proposed solutions mostly tackle only one specific part of the whole system. For instance, the proposed architecture in [53] concentrates on the scalability of the networking and computing resource infrastructure, while the work presented in [9] only supports scalability of fog nodes from the nodal point of view. An area that needs particular attention in terms of system scalability is the design of novel mechanisms that enable discovering, utilizing and monitoring nodes and services across all providers. This requires to allow the system to be aware of both the current status of the available resources from all providers and to elastically scale function executions through the appropriate procedures.

Interoperability is the third main issue that closely impacts the support for heterogeneity and also the system scalability.

It calls for the design of signaling, control, and data interfaces between several domains that are part of the whole system. Two vital requirements should be fulfilled: a) implementing standard inter-domain and operational interfaces and b) effective mobility management and federation frameworks needs to be developed. These latter issues are largely ignored in current solutions and need to be considered in future research.

IX. CONCLUSION

In this article, a comparative study of competing technologies including cloud computing, mobile computing, edge computing and fog computing was carried out. Also, a taxonomy of fog computing research was developed that addresses various subject areas (system, application, software, security, resource management and networking) as well as research aspects, namely architecture, algorithms and technologies. A comprehensive survey of various architectural perspectives in fog computing is provided. These architectural perspectives complementarily elaborate on physical as well as logical components and modules in a fog computing environment and their respective roles and functionalities. Some architectures have a network wide view and some focus on node-level architecture. Related algorithms and technologies need to be developed in a consistent and robust manner to realize the full advantage of fog computing, some of which has already taken place. The proposed architectures were evaluated and compared based on a number of criteria. Finally, a survey of research issues and directions in each subject area was carried out and major challenges were identified and discussions on future research were provided. The results of this survey can help the research community to tune their efforts towards the existing gaps to help push this technology to maturity.

REFERENCES

[1] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Workshop Mobile Big Data*, New York, NY, USA, 2015, pp. 37–42.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput.*, New York, NY, USA, 2012, pp. 13–16.

[3] S. Yangui, P. Ravindran, O. Bibani, R. H. Glitho, N. Ben Hadj-Alouane, M. J. Morrow, and P. A. Polakos, "A platform as-a-service for hybrid cloud/fog environments," in *Proc. IEEE Int. Symp. Local Metrop. Area Netw. (LANMAN)*, Jun. 2016, pp. 1–7.

[4] M. Aazam and E.-N. Huh, "Dynamic resource provisioning through fog micro datacenter," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2015, pp. 105–110.

[5] O. Salman, I. Elhajj, A. Kayssi, and A. Chehab, "Edge computing enabling the Internet of Things," in *Proc. IEEE 2nd World Forum Internet Things (WF-IoT)*, Dec. 2015, pp. 603–608.

[6] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013.

[7] J.-M. Kang, H. Bannazadeh, H. Rahimi, T. Lin, M. Faraji, and A. Leon-Garcia, "Software-defined infrastructure and the future central office," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2013, pp. 225–229.

[8] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[9] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for Internet of Things and analytics," in *Big Data Internet Things: A Roadmap for Smart Environments*. Cham, Switzerland: Springer, 2014, pp. 169–186.

[10] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *Proc. Int. Conf. Wireless Algorithms, Syst. Appl.*, 2015, pp. 685–695.

[11] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: Architecture, key technologies, applications and open issues," *J. Netw. Comput. Appl.*, vol. 98, pp. 27–42, Nov. 2017.

[12] A. Vahid Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog computing: Principles, architectures, and applications," 2016, *arXiv:1601.02752*. [Online]. Available: http://arxiv.org/abs/1601.02752

[13] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug. 2016.

[14] C. C. Byers, "Architectural imperatives for fog computing: Use cases, requirements, and architectural techniques for fog-enabled IoT networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 14–20, Aug. 2017.

[15] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of Everything*. Singapore: Springer, 2018, pp. 103–130.

[16] 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," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, 1st Quart., 2018.

[17] R. Kumar Naha, S. Garg, and A. Chan, "Fog computing architecture: Survey and challenges," 2018, *arXiv:1811.09047*. [Online]. Available: http://arxiv.org/abs/1811.09047

[18] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *J. Syst. Archit.*, vol. 98, pp. 289–330, Sep. 2019.

[19] H. Hejazi, H. Rajab, T. Cinkler, and L. Lengyel, "Survey of platforms for massive IoT," in *Proc. IEEE Int. Conf. Future IoT Technol. (Future IoT)*, Jan. 2018, pp. 1–8.

[20] M. Eder, "Hypervisor-vs. container-based virtualization," in *Proc. Future Internet (FI) Innov. Internet Technol. Mobile Commun. (IITM)*, vol. 1, 2016, pp. 11–17.

[21] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system," *IEEE Trans. Comput.*, vol. 65, no. 12, pp. 3702–3712, Dec. 2016.

[22] H. Xiang, W. Zhou, M. Daneshmand, and M. Peng, "Network slicing in fog radio access networks: Issues and challenges," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 110–116, Dec. 2017.

[23] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Fog computing: Enabling the management and orchestration of smart city applications in 5G networks," *Entropy*, vol. 20, no. 1, pp. 4–15, 2017.

[24] D. Zhao, D. Liao, G. Sun, and S. Xu, "Towards resource-efficient service function chain deployment in cloud-fog computing," *IEEE Access*, vol. 6, pp. 66754–66766, 2018.

[25] A. Shawish and M. Salama, "Cloud computing: Paradigms and technologies," in *Inter-Cooperative Collective Intelligence: Techniques and Applications*. Berlin, Germany: Springer, 2014, pp. 39–67.

[26] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[27] P. Mell and T. Grance, "The NIST definition of cloud computing," Natl. Inst. Stand. Technol., Gaithersburg, MD, USA, Inf. Technol. Lab., Tech. Rep. 800-145, 2009.

[28] R. Jain and S. Paul, "Network virtualization and software defined networking for cloud computing: A survey," *IEEE Commun. Mag.*, vol. 51, no. 11, pp. 24–31, Nov. 2013.

[29] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 1–11, Jan. 2011.

[30] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Serv. Appl.*, vol. 1, pp. 7–18, May 2010.

[31] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," in *Advances in Computers*, vol. 82, M. V. Zelkowitz, Ed. Amsterdam, The Netherlands: Elsevier, 2011, ch. 3, pp. 47–111.

[32] C. Modi, D. Patel, B. Borisaniya, A. Patel, and M. Rajarajan, "A survey on security issues and solutions at different layers of cloud computing," *J. Supercomput.*, vol. 63, no. 2, pp. 561–592, Feb. 2013.

[33] I. Sahu and U. S. Pandey, "Mobile cloud computing: Issues and challenges," in *Proc. Int. Conf. Adv. Comput., Commun. Control Netw. (ICACCCN)*, Oct. 2018, pp. 247–250.

[34] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and privacy in cloud computing: A survey," in *Proc. 6th Int. Conf. Semantics, Knowl. Grids*, 2010, pp. 105–112.

[35] S. H. Mortazavi, M. Salehe, C. S. Gomes, C. Phillips, and E. de Lara, "Cloudpath: A multi-tier cloud computing framework," in *Proc. 2nd ACM/IEEE Symp. Edge Comput. (SEC)*, vol. 20, 2017, pp. 1–20.

[36] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Gener. Comput. Syst.*, vol. 29, pp. 84–106, Jan. 2013.

[37] D. B. Hoang and L. Chen, "Mobile cloud for assistive healthcare (MoCAsH)," in *Proc. IEEE Asia–Pacific Services Comput. Conf.*, Dec. 2010, pp. 325–332.

[38] G. Sun and J. Shen, "Facilitating social collaboration in mobile cloud-based learning: A teamwork as a service (TaaS) approach," *IEEE Trans. Learn. Technol.*, vol. 7, no. 3, pp. 207–220, Jul. 2014.

[39] K. Sabarish and R. S. Shaji, "A scalable cloud enabled mobile governance framework," in *Proc. IEEE Global Humanitarian Technol. Conf.-South Asia Satell. (GHTC-SAS)*, Sep. 2014, pp. 25–34.

[40] E. Ahmed, A. Gani, M. K. Khan, R. Buyya, and S. U. Khan, "Seamless application execution in mobile cloud computing: Motivation, taxonomy, and open challenges," *J. Netw. Comput. Appl.*, vol. 52, pp. 154–172, Jun. 2015.

[41] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 337–368, 1st Quart., 2014.

[42] M. Sharifi, S. Kafaie, and O. Kashefi, "A survey and taxonomy of cyber foraging of mobile devices," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1232–1243, 4th Quart., 2012.

[43] M. Satyanarayanan, "A brief history of cloud offload: A personal journey from odyssey through cyber foraging to cloudlets," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 18, no. 4, pp. 19–23, Jan. 2015.

[44] K. Ha, P. Pillai, W. Richter, Y. Abe, and M. Satyanarayanan, "Just-in-time provisioning for cyber foraging," in *Proc. 11th Annu. Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, New York, NY, USA, 2013, pp. 153–166.

[45] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervas. Comput.*, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[46] E. Ahmed, A. Akhunzada, M. Whaiduzzaman, A. Gani, S. H. A. Hamid, and R. Buyya, "Network-centric performance analysis of runtime application migration in mobile cloud computing," *Simul. Model. Pract. Theory*, vol. 50, pp. 42–56, Jan. 2015.

[47] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, May 2016.

[48] J. Pan and J. McElhannon, "Future edge cloud and edge computing for Internet of Things applications," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 439–449, Feb. 2018.

[49] M. Patel, J. Joubert, J. R. Ramos, N. Sprecher, S. Abeta, and A. Neal, "Mobile-edge computing introductory technical white paper," ETSI, Mobile-Edge Comput. Ind. Initiative, Sophia Antipolis, France, Tech. Rep., 2014, no. 1.

[50] *Mobile-Edge Computing (MEC); Proof of Concept Framework*, ETSI, Sophia Antipolis, France, 2015.

[51] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, "Mobile-edge computing architecture: The role of MEC in the Internet of Things," *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 84–91, Oct. 2016.

[52] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," ETSI, Sophia Antipolis, France, White Paper. 11, Sep. 2015.

[53] P. Habibi, S. Baharlooei, M. Farhoudi, S. Kazemian, and S. Khorsandi, "Virtualized SDN-based end-to-end reference architecture for fog networking," in *Proc. 32nd Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, May 2018, pp. 61–66.

[54] Z. Pang, L. Sun, Z. Wang, E. Tian, and S. Yang, "A survey of cloudlet based mobile computing," in *Proc. Int. Conf. Cloud Comput. Big Data (CCBD)*, Nov. 2015, pp. 268–275.

[55] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol. (HotWeb)*, Nov. 2015, pp. 73–78.

[56] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Cloudlets: Bringing the cloud to the mobile user," in *Proc. 3rd ACM Workshop Mobile Cloud Comput. Services (MCS)*, New York, NY, USA, 2012, pp. 29–36.

[57] H. Madsen, B. Burtschy, G. Albeanu, and F. Popentiu-Vladicescu, "Reliability in the utility computing era: Towards reliable fog computing," in *Proc. 20th Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Jul. 2013, pp. 43–46.

[58] M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, and M. Nemirovsky, "Key ingredients in an IoT recipe: Fog computing, cloud computing, and more fog computing," in *Proc. IEEE 19th Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, Dec. 2014, pp. 325–329.

[59] Y. N. Krishnan, C. N. Bhagwat, and A. P. Utpat, "Fog computing; network based cloud computing," in *Proc. 2nd Int. Conf. Electron. Commun. Syst. (ICECS)*, Feb. 2015, pp. 250–251.

[60] K. Kai, W. Cong, and L. Tao, "Fog computing for vehicular ad-hoc networks: Paradigms, scenarios, and issues," *J. China Universities Posts Telecommun.*, vol. 23, no. 2, pp. 56–96, Apr. 2016.

[61] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in *Proc. ASE BigData SocialInformatics*, New York, NY, USA, 2015, pp. 1–28.

[62] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Proc. Int. Conf. Future Internet Things Cloud*, Aug. 2014, pp. 464–470.

[63] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of Internet of Things," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 46–59, Jan. 2018.

[64] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[65] A. Botta, W. de Donato, V. Persico, and A. Pescape, "On the integration of cloud computing and Internet of Things," in *Proc. Int. Conf. Future Internet Things Cloud*, Aug. 2014, pp. 23–30.

[66] R. Vilalta, V. Lopez, A. Giorgetti, S. Peng, V. Orsini, L. Velasco, R. Serral-Gracia, D. Morris, S. De Fina, F. Cugini, P. Castoldi, A. Mayoral, R. Casellas, R. Martinez, C. Verikoukis, and R. Munoz, "TelcoFog: A unified flexible fog and cloud computing architecture for 5G networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 36–43, Aug. 2017.

[67] C. Byers and R. Swanson, "OpenFog consortium, OpenFog reference architecture for fog computing," OpenFog Consortium Archit. Working Group, Fremont, CA, USA, Tech. Rep. OPFRA001.020817, Feb. 2017.

[68] *Network Functions Virtualization (NFV) Architectural Framework*, ETSI, Sophia Antipolis, France, 2013.

[69] E. E. Haleplidis, E. K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, *Software-Defined Networking (SDN): Layers and Architecture Terminology*, document RFC 7426, 2015.

[70] T. D. Fleming, A. Gentle, L. Hochstein, J. Proulx, E. Toews, *OpenStack Operations Guide*, 1st ed. Newton, MA, USA: O'Reilly Media, 2014.

[71] K. Velasquez, D. P. Abreu, D. Goncalves, L. Bittencourt, M. Curado, E. Monteiro, and E. Madeira, "Service orchestration in fog environments," in *Proc. IEEE 5th Int. Conf. Future Internet Things Cloud (FiCloud)*, Prague, Czech Republic, Aug. 2017, pp. 36–329.

[72] M. S. de Brito, S. Hoque, T. Magedanz, R. Steinke, A. Willner, D. Nehls, O. Keils, and F. Schreiner, "A service orchestration architecture for fog-enabled infrastructures," in *Proc. 2nd Int. Conf. Fog Mobile Edge Comput. (FMEC)*, May 2017, pp. 127–132.

[73] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 6, pp. 1201–1221, Jun. 2017.

[74] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[75] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for mobile Networks—A technology overview," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 405–426, 1st Quart., 2015.

[76] M. Peng, Y. Li, Z. Zhao, and C. Wang, "System architecture and key technologies for 5G heterogeneous cloud radio access networks," *IEEE Netw.*, vol. 29, no. 2, pp. 6–14, Mar. 2015.
[77] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 46–53, Jul. 2016.
[78] Y.-J. Ku, D.-Y. Lin, and H.-Y. Wei, "Fog RAN over general purpose processor platform," in *Proc. IEEE 84th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2016, pp. 1–2.
[79] B. Blanco, J. O. Fajardo, I. Giannoulakis, E. Kafetzakis, S. Peng, J. Pérez-Romero, I. Trajkovska, P. S. Khodashenas, L. Goratti, M. Paolino, E. Sfakianakis, F. Liberal, and G. Xilouris, "Technology pillars in the architecture of future 5G mobile networks: NFV, MEC and SDN," *Comput. Standards Interfaces*, vol. 54, pp. 216–228, Nov. 2017.
[80] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.
[81] R. Brzoza-Woch, M. Konieczny, P. Nawrocki, T. Szydlo, and K. Zielinski, "Embedded systems in the application of fog computing—Levee monitoring use case," in *Proc. 11th IEEE Symp. Ind. Embedded Syst. (SIES)*, Kraków, Poland, May 2016, pp. 1–6.
[82] C. Huang, R. Lu, and K.-K.-R. Choo, "Vehicular fog computing: Architecture, use case, and security and forensic challenges," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 105–111, Nov. 2017.
[83] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.
[84] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare," *Future Gener. Comput. Syst.*, vol. 78, pp. 659–676, Jan. 2018.
[85] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, Jan. 2018.
[86] T. N. Gia, M. Jiang, A.-M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog computing in healthcare Internet of Things: A case study on ECG feature extraction," in *Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput. Commun., Dependable, Autonomic Secure Comput., Pervas. Intell. Comput.*, Oct. 2015, pp. 356–363.
[87] J. Oueis, E. C. Strinati, S. Sardellitti, and S. Barbarossa, "Small cell clustering for efficient distributed fog computing: A multi-user case," in *Proc. IEEE 82nd Veh. Technol. Conf. (VTC-Fall)*, Boston, MA, USA, Sep. 2015, pp. 1–5.
[88] N. Mohamed, J. Al-Jaroodi, S. Lazarova-Molnar, I. Jawhar, and S. Mahmoud, "A service-oriented middleware for cloud of things and fog computing supporting smart city applications," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug. 2017, pp. 1–7.
[89] S. Sarkar and S. Misra, "Theoretical modelling of fog computing: A green computing paradigm to support IoT applications," *IET Netw.*, vol. 5, no. 2, pp. 23–29, Mar. 2016.
[90] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog computing may help to save energy in cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1728–1739, May 2016.
[91] Y. Cao, S. Chen, P. Hou, and D. Brown, "FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation," in *Proc. IEEE Int. Conf. Netw., Archit. Storage (NAS)*, Boston, MA, USA, Aug. 2015, pp. 2–11.
[92] J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, Glasgow, U.K., May 2015, pp. 1–6.
[93] R. Deng, R. Lu, C. Lai, and T. H. Luan, "Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 3909–3914.
[94] D. Ye, M. Wu, S. Tang, and R. Yu, "Scalable fog computing with service offloading in bus networks," in *Proc. IEEE 3rd Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)*, Beijing, China, Jun. 2016, pp. 247–251.
[95] X. Hongyu, M. Peng, Y. Cheng, and H.-H. Chen, "Joint mode selection and resource allocation for downlink fog radio access networks supported D2D," in *Proc. 11th EAI Int. Conf. Heterogeneous Netw. Qual., Rel., Secur. Robustness (QSHINE)*, 2015, pp. 177–182.
[96] D. Chen, S. Schedler, and V. Kuehn, "Backhaul traffic balancing and dynamic content-centric clustering for the downlink of fog radio access network," in *Proc. IEEE 17th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Edinburgh, U.K., Jul. 2016, pp. 1–5.
[97] E. Baccarelli, P. G. V. Naranjo, M. Scarpiniti, M. Shojafar, and J. H. Abawajy, "Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study," *IEEE Access*, vol. 5, pp. 9882–9910, 2017.
[98] J. C. Guevara, L. F. Bittencourt, and N. L. S. da Fonseca, "Class of service in fog computing," in *Proc. IEEE 9th Latin-American Conf. Commun. (LATINCOM)*, Nov. 2017, pp. 1–6.
[99] V. Issarny, M. Caporuscio, and N. Georgantas, "A perspective on the future of middleware-based software engineering," in *Proc. Future Softw. Eng. (FOSE)*, May 2007, pp. 244–258.
[100] Z. Hao, E. Novak, S. Yi, and Q. Li, "Challenges and software architecture for fog computing," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 44–53, Mar. 2017.
[101] A. Singh and D. M. Shrivastava, "Overview of attacks on cloud computing," *Int. J. Eng. Innov. Technol.*, vol. 1, pp. 321–323, Apr. 2012.
[102] F. Sabahi, "Cloud computing security threats and responses," in *Proc. IEEE 3rd Int. Conf. Commun. Softw. Netw.*, May 2011, pp. 245–249.
[103] N. Gruschka and M. Jensen, "Attack surfaces: A taxonomy for attacks on cloud services," in *Proc. IEEE 3rd Int. Conf. Cloud Comput.*, Jul. 2010, pp. 276–279.
[104] R. L. Krutz and R. D. Vines, *Cloud Security: A Comprehensive Guide to Secure Cloud Computing*. Hoboken, NJ, USA: Wiley, 2010.
[105] M. Jensen, J. Schwenk, N. Gruschka, and L. L. Iacono, "On technical security issues in cloud computing," in *Proc. IEEE Int. Conf. Cloud Comput.*, Sep. 2009, pp. 109–116.
[106] M. A. Talib, A. Khelifi, and T. Ugurlu, "Using ISO 27001 in teaching information security," in *Proc. 38th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2012, pp. 3149–3153.
[107] M. Dabbagh and A. Rayes, "Internet of Things security and privacy," in *Internet of Things From Hype to Reality: The Road to Digitization*. Cham, Switzerland: Springer, 2019, pp. 211–238. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-99516-8_8
[108] J. Niu, Z. Ming, M. Qiu, H. Su, Z. Gu, and X. Qin, "Defending jamming attack in wide-area monitoring system for smart grid," *Telecommun. Syst.*, vol. 60, no. 1, pp. 159–167, Sep. 2015.
[109] M. Darwish, A. Ouda, and L. F. Capretz, "Cloud-based DDoS attacks and defenses," in *Proc. Int. Conf. Inf. Soc. (I-Society)*, 2013, pp. 67–71.
[110] N. Thakur, "Introduction to jamming attacks and prevention techniques using honeypots in wireless networks," *Int. J. Comput. Sci. Inf. Technol. Secur.*, vol. 3, no. 2, pp. 202–207, 2017.
[111] S. Bhunia, S. Sengupta, and F. Vazquez-Abad, "CR-honeynet: A learning & decoy based sustenance mechanism against jamming attack in CRN," in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2014, pp. 1173–1180.
[112] K. Sornalakshmi, "Detection of DoS attack and zero day threat with SIEM," in *Proc. Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Jun. 2017, pp. 1–7.
[113] Deepali and K. Bhushan, "DDoS attack defense framework for cloud using fog computing," in *Proc. 2nd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2017, pp. 534–538.
[114] M. Elsabagh, D. Barbará, D. Fleck, and A. Stavrou, "Radmin: Early detection of applicationlevel resource exhaustion and starvation attacks," in *Proc. 18th Int. Conf. Res. Attacks, Intrusions Defenses*, vol. 9404. Cham, Switzerland: Springer, Nov. 2015, pp. 515–537.
[115] T. K. Buennemeyer, M. Gora, R. C. Marchany, and J. G. Tront, "Battery exhaustion attack detection with small handheld mobile computers," in *Proc. IEEE Int. Conf. Portable Inf. Devices*, May 2007, pp. 1–5.
[116] C. J. Fung and B. McCormick, "VGuard: A distributed denial of service attack mitigation method using network function virtualization," in *Proc. 11th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2015, pp. 64–70.
[117] M. Ozcelik, N. Chalabianloo, and G. Gur, "Software-defined edge defense against IoT-based DDoS," in *Proc. IEEE Int. Conf. Comput. Inf. Technol. (CIT)*, Aug. 2017, pp. 308–313.
[118] A. A. A. El-Latif, B. Abd-El-Atty, M. S. Hossain, S. Elmougy, and A. Ghoneim, "Secure quantum steganography protocol for fog cloud Internet of Things," *IEEE Access*, vol. 6, pp. 10332–10340, 2018.
[119] O. Salman, S. Abdallah, I. H. Elhajj, A. Chehab, and A. Kayssi, "Identity-based authentication scheme for the Internet of Things," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2016, pp. 1109–1111.

[120] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, Sep. 2014, pp. 1–8.

[121] A. Vishwanath, R. Peruri, and J. He, "Security in fog computing through encryption," *Int. J. Inf. Technol. Comput. Sci.*, vol. 8, no. 5, pp. 28–36, May 2016.

[122] C. Li, Z. Qin, E. Novak, and Q. Li, "Securing SDN infrastructure of IoT–Fog networks from MitM attacks," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1156–1164, Oct. 2017.

[123] M. Tighe, G. Keller, M. Bauer, and H. Lutfiyya, "DCSim: A data centre simulation tool for evaluating dynamic virtualized resource management," in *Proc. 8th Int. Conf. Netw. Service Manage. (CNSM)*, 2012, pp. 385–392.

[124] Y. Jararweh, M. Jarrah, M. Kharbutli, Z. Alshara, M. N. Alsaleh, and M. Al-Ayyoub, "CloudExp: A comprehensive cloud computing experimental framework," *Simul. Model. Pract. Theory*, vol. 49, pp. 180–192, Dec. 2014.

[125] H. Han, B. Sheng, C. C. Tan, Q. Li, and S. Lu, "A measurement based rogue AP detection scheme," in *Proc. IEEE 28th Conf. Comput. Commun. (INFOCOM)*, Apr. 2009, pp. 1593–1601.

[126] H. Han, B. Sheng, C. C. Tan, Q. Li, and S. Lu, "A timing-based scheme for rogue AP detection," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 11, pp. 1912–1925, Nov. 2011.

[127] G. W. Kibirige and C. Sanga, "A survey on detection of sinkhole attack in wireless sensor network," *Int. J. Comput. Sci. Inf. Secur.*, vol. 13, pp. 48–52, May 2015.

[128] S. Jain and A. Kajal, "Effective analysis of risks and vulnerabilities in Internet of Things," *Int. J. Comput. Corporate Res.*, vol. 5, no. 2, Mar. 2015. [Online]. Available: http://www.ijccr.com/March2015/4.pdf

[129] A. TaheriMonfared and M. G. Jaatun, "Handling compromised components in an IaaS cloud installation," *J. Cloud Comput., Adv., Syst. Appl.*, vol. 1, no. 1, 2012, Art. no. 16.

[130] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad Hoc Netw.*, vol. 1, nos. 2–3, pp. 293–315, Sep. 2003.

[131] M. A. Hamid, M. O. Rashid, and C. S. Hong, "Routing security in sensor network: Hello flood attack and defense," in *Proc. IEEE ICNEWS*, Jan. 2006, pp. 2–4.

[132] O. Onolaja, R. Bahsoon, and G. Theodoropoulos, "Conceptual framework for dynamic trust monitoring and prediction," *Procedia Comput. Sci.*, vol. 1, no. 1, pp. 1241–1250, May 2010.

[133] T. Bläsing, L. Batyuk, A.-D. Schmidt, S. A. Camtepe, and S. Albayrak, "An Android application sandbox system for suspicious software detection," in *Proc. 5th Int. Conf. Malicious Unwanted Softw.*, Oct. 2010, pp. 55–62.

[134] T. Abou-Assaleh, N. Cercone, V. Keselj, and R. Sweidan, "N-gram-based detection of new malicious code," in *Proc. 28th Annu. Int. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 2, 2004, pp. 41–42.

[135] L. Martignoni, R. Paleari, and D. Bruschi, "A framework for behavior-based malware analysis in the cloud," in *Proc. 5th Int. Conf. Inf. Syst. Secur. (ICISS)*. Berlin, Germany: Springer, Dec. 2009, pp. 178–192.

[136] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: Behavior-based malware detection system for Android," in *Proc. 1st ACM Workshop Secur. Privacy Smartphones Mobile Devices (SPSM)*, 2011, pp. 15–26.

[137] X. Wang, Y. Yang, Y. Zeng, C. Tang, J. Shi, and K. Xu, "A novel hybrid mobile malware detection system integrating anomaly detection with misuse detection," in *Proc. 6th Int. Workshop Mobile Cloud Comput. Services (MCS)*, 2015, pp. 15–22.

[138] P. Ratha, D. Swain, B. Paikaray, and S. Sahoo, "An optimized encryption technique using an arbitrary matrix with probabilistic encryption," *Procedia Comput. Sci.*, vol. 57, pp. 1235–1241, Jan. 2015.

[139] N. Sathishkumar and K. Rajakumar, "A study on vehicle to vehicle collision prevention using fog, cloud, big data and elliptic curve security based on threshold energy efficient protocol in wireless sensor network," in *Proc. 2nd Int. Conf. Recent Trends Challenges Comput. Models (ICRTCCM)*, Feb. 2017, pp. 275–280.

[140] K. Ruan, J. Carthy, T. Kechadi, and M. Crosbie, "Cloud forensics," in *Proc. IFIP Int. Conf. Digit. Forensics*, vol. 361. Berlin, Germany: Springer, 2011, pp. 35–46.

[141] Y. Wang, T. Uehara, and R. Sasaki, "Fog computing: Issues and challenges in security and forensics," in *Proc. IEEE 39th Annu. Comput. Softw. Appl. Conf.*, vol. 3, Jul. 2015, pp. 53–59.

[142] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 772–782, Jun. 2017.

[143] Q. Yaseen, Y. Jararweh, M. Al-Ayyoub, and M. AlDwairi, "Collusion attacks in Internet of Things: Detection and mitigation using a fog based model," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Mar. 2017, pp. 1–5.

[144] S. A. Soleymani, A. H. Abdullah, M. Zareei, M. H. Anisi, C. Vargas-Rosales, M. K. Khan, and S. Goudarzi, "A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing," *IEEE Access*, vol. 5, pp. 15619–15629, 2017.

[145] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[146] M. Tebaa and S. E. Hajji, "Secure cloud computing through homomorphic encryption," *Int. J. Advancements Comput. Technol.*, vol. 5, pp. 29–38, Dec. 2013.

[147] S. Sharma, J. Powers, and K. Chen, "Privacy-preserving spectral analysis of large graphs in public clouds," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur. (ASIA CCS)*, 2016, pp. 71–82.

[148] S. Narayan, M. Gagné, and R. Safavi-Naini, "Privacy preserving EHR system using attribute-based infrastructure," in *Proc. ACM Workshop Cloud Comput. Secur. Workshop (CCSW)*, 2010, pp. 47–52.

[149] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 9, pp. 1621–1631, Sep. 2012.

[150] X. Liu, R. H. Deng, Y. Yang, H. N. Tran, and S. Zhong, "Hybrid privacy-preserving clinical decision support system in fog–cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 825–837, Jan. 2018.

[151] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: Protecting confidentiality with encrypted query processing," in *Proc. 23rd ACM Symp. Operating Syst. Princ. (SOSP)*, 2011, pp. 85–100.

[152] H. Hacigümüş, B. Iyer, C. Li, and S. Mehrotra, "Executing SQL over encrypted data in the database-service-provider model," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2002, pp. 216–227.

[153] T. Wang, J. Zhou, X. Chen, G. Wang, A. Liu, and Y. Liu, "A three-layer privacy preserving cloud storage scheme based on computational intelligence in fog computing," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 3–12, Feb. 2018.

[154] T. Wang, J. Zeng, M. Z. A. Bhuiyan, H. Tian, Y. Cai, Y. Chen, and B. Zhong, "Trajectory privacy preservation based on a fog structure for cloud location services," *IEEE Access*, vol. 5, pp. 7692–7701, 2017.

[155] K. M. Reena, S. K. Yadav, N. K. Bajaj, and V. Singh, "Security implementation in cloud computing using user behaviour profiling and decoy technology," in *Proc. Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, Mar. 2017, pp. 471–474.

[156] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: Privacy preserving fog-enabled aggregation in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3733–3744, Aug. 2018.

[157] Y. Huo, Y. Tian, L. Ma, X. Cheng, and T. Jing, "Jamming strategies for physical layer security," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 148–153, Feb. 2018.

[158] M.-M. Bazm, M. Lacoste, M. Sudholt, and J.-M. Menaud, "Side-channels beyond the cloud edge: New isolation threats and solutions," in *Proc. 1st Cyber Secur. Netw. Conf. (CSNet)*, Oct. 2017, pp. 1–8.

[159] V. Natu and T. N. B. Duong, "Secure virtual machine placement in infrastructure cloud services," in *Proc. IEEE 10th Conf. Service-Oriented Comput. Appl. (SOCA)*, Nov. 2017, pp. 26–33.

[160] F. Zhou, M. Goel, P. Desnoyers, and R. Sundaram, "Scheduler vulnerabilities and coordinated attacks in cloud computing," *J. Comput. Secur.*, vol. 21, no. 4, pp. 533–559, Sep. 2013.

[161] H. Wang, Z. Wang, and J. Domingo-Ferrer, "Anonymous and secure aggregation scheme in fog-based public cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 712–719, Jan. 2018.

[162] X. Masip-Bruin, E. Marin-Tordera, A. Jukan, and G.-J. Ren, "Managing resources continuity from the edge to the cloud: Architecture and performance," *Future Gener. Comput. Syst.*, vol. 79, pp. 777–785, Feb. 2018.

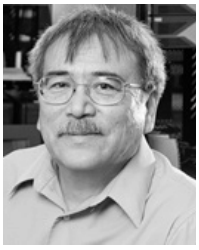
[163] S. Sengupta, J. Garcia, and X. Masip-Bruin, "Taxonomy and resource modeling in combined fog-to-cloud systems," in *Proc. Future Technol. Conf.*, 2018, pp. 687–704.
[164] A. Brogi and S. Forti, "QoS-aware deployment of IoT applications through the fog," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1185–1192, Oct. 2017.
[165] S. Verma, A. K. Yadav, D. Motwani, R. S. Raw, and H. K. Singh, "An efficient data replication and load balancing technique for fog computing environment," in *Proc. 3rd Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, 2016, pp. 2888–2895.
[166] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.
[167] X. Meng, W. Wang, and Z. Zhang, "Delay-constrained hybrid computation offloading with cloud and fog computing," *IEEE Access*, vol. 5, pp. 21355–21367, 2017.
[168] J. Liu, E. Ahmed, M. Shiraz, A. Gani, R. Buyya, and A. Qureshi, "Application partitioning algorithms in mobile cloud computing: Taxonomy, review and future directions," *J. Netw. Comput. Appl.*, vol. 48, pp. 99–117, Feb. 2015.
[169] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "IFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments," *Softw., Pract. Exper.*, vol. 47, no. 9, pp. 1275–1296, Sep. 2017.
[170] V. B. Souza, X. Masip-Bruin, E. Marín-Tordera, S. Sànchez-López, J. Garcia, G. J. Ren, A. Jukan, and A. J. Ferrer, "Towards a proper service placement in combined Fog-to-Cloud (F2C) architectures," *Future Gener. Comput. Syst.*, vol. 87, pp. 1–15, Oct. 2018.
[171] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, "Resource allocation strategy in fog computing based on priced timed Petri nets," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1216–1228, Oct. 2017.
[172] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.
[173] K. Liu, J. Peng, X. Zhang, and Z. Huang, "A combinatorial optimization for energy-efficient mobile cloud offloading over cellular networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.
[174] E. Cuervo, A. Balasubramanian, D.-K. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: Making smartphones last longer with code offload," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2010, pp. 49–62.
[175] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.
[176] Y.-H. Kao, B. Krishnamachari, M.-R. Ra, and F. Bai, "Hermes: Latency optimal task assignment for resource-constrained mobile computing," *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 3056–3069, Nov. 2017.
[177] H. Wu, W. Knottenbelt, K. Wolter, and Y. Sun, "An optimal offloading partitioning algorithm in mobile cloud computing," in *Proc. Int. Conf. Quant. Eval. Syst.* Cham, Switzerland: Springer, 2016, pp. 311–328.
[178] T. Q. Thinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.
[179] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.
[180] O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.
[181] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.
[182] V. Cardellini, V. De Nitto Personé, V. Di Valerio, F. Facchinei, V. Grassi, F. Lo Presti, and V. Piccialli, "A game-theoretic approach to computation offloading in mobile cloud computing," *Math. Program.*, vol. 157, no. 2, pp. 421–449, Jun. 2016.
[183] S.-W. Ko, K. Huang, S.-L. Kim, and H. Chae, "Live prefetching for mobile computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3057–3071, May 2017.
[184] S.-C. Hung, H. Hsu, S.-Y. Lien, and K.-C. Chen, "Architecture harmonization between cloud radio access networks and fog networks," *IEEE Access*, vol. 3, pp. 3019–3034, 2015.
[185] M. S. Elbamby, M. Bennis, and W. Saad, "Proactive edge computing in latency-constrained fog networks," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2017, pp. 1–6.
[186] S. Tomovic, K. Yoshigoe, I. Maljevic, and I. Radusinovic, "Software-defined fog network architecture for IoT," *Wireless Pers. Commun.*, vol. 92, no. 1, pp. 181–196, Jan. 2017.
[187] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, Oct. 2014.
[188] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," in *Proc. Australas. Telecommun. Netw. Appl. Conf. (ATNAC)*, Nov. 2014, pp. 117–122.
[189] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular adhoc network with fog computing," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 1202–1207.
[190] R. Huo, F. R. Yu, T. Huang, R. Xie, J. Liu, V. C. M. Leung, and Y. Liu, "Software defined networking, caching, and computing for green wireless networks," *IEEE Commun. Mag.*, vol. 54, no. 11, pp. 185–193, Nov. 2016.
[191] J. W. Walewski, M. Bauer, N.Bui, P. Giacomin, N. Gruschka, S. Haller, E. Ho, R. Kernchen, M. Lischka, J. De Loof, C. Magerkurth, S. Meissner, S. Meyer, A. Nettsträter, F. O. Lacalle, A. S. Segura, A. Serbanati, M. Strohbach, and V. Toubiana, "Internet-of-Things architecture D1.2-initial architectural reference model for IoT," Eur. Commission, Brussels, Belgium, Tech. Rep. D1.2, Jun. 2011.

POOYAN HABIBI (Student Member, IEEE) received the M.S. degree in computer engineering from the Amirkabir University of Technology, Tehran, Iran, in 2017. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Toronto, Toronto, ON, Canada. He was a Senior Research Assistant with the Iran Telecommunication Research Center, Tehran. His research interests include cloud computing, edge computing, and network virtualization. He was a recipient of the Edward S. Rogers Senior Departmental Graduate Fellowship. He holds a variety of IT certifications, including CCNA, CCNP, and CCDP, MCITP, and MCSE.

MOHAMMAD FARHOUDI received the bachelor's and master's degrees in computer engineering from the Amirkabir University of Technology, Tehran, Iran. He worked as the Network Administrator of computer engineering with the IT Department, in 2011. He also worked on Saba System Sadra and Fava Pars Company, for six years. His work experience includes computational networking and computer security. He is currently a Research Assistant with the Amirkabir University of Technology. His research interests are about SDN, NFV, fog, and cloud computing.

SEPEHR KAZEMIAN received the B.Sc. degree in computer engineering from the Amirkabir University of Technology, Tehran, Iran, in 2017. He is currently pursuing the M.Sc. degree in computer science with the University of Alberta, Edmonton, AB, Canada. He was a Research Assistant with the Amirkabir University of Technology. His research interests include wireless networking, edge computing, and machine learning.

SIAVASH KHORSANDI received the B.Sc. and M.Sc. degrees in electrical engineering from Tehran Polytechnic, in 1987 and 1989, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, in 1996. He joined Amirkabir University, in 2001, after a four-year Tenure, from 1996 to 2000, at Nortel Networks working as a Senior Engineer on advanced network architectures. He is currently an Associate Professor with the Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran. He has published more than 100 articles in international journals and conferences and directed several large-scale projects on next generation networking technologies. His subject areas of interest include computational networking, computer security, and modeling and evaluation of distributed and self-organized systems. He has also served as the Board Member of the Iran computer Society, the Head of the ICTRC Research Center, and the President of the Fava Pars Company, and Commit IT Solutions. He has paid several research visits to the University of Ottawa, the University of Carleton, and the University of Toronto.



ALBERTO LEON-GARCIA (Life Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1973, 1974, and 1976, respectively. He is currently a Distinguished Professor in electrical and computer engineering with the University of Toronto. He is author of the textbooks: *Probability and Random Processes for Electrical Engineering, and Communication Networks: Fundamental Concepts and Key Architecture*. His research is on application platforms for smart applications, including smart cities. He is a Fellow of the Institute of Electronics an Electrical Engineering "For contributions to multiplexing and switching of integrated services traffic".