Fog Computing Applications: Taxonomy and Requirements

Arif Ahmed, HamidReza Arkian, Davaadorj Battulga, Ali J. Fahs, Mozhdeh Farhadi, Dimitrios Giouroukis, Adrien Gougeon, Felipe Oliveira Gutierrez, Guillaume Pierre, Paulo R. Souza Jr, Mulugeta Ayalew Tamiru, Li Wu,

Abstract—Fog computing was designed to support the specific needs of latency-critical applications such as augmented reality, and IoT applications which produce massive volumes of data that are impractical to send to faraway cloud data centers for analysis. However this also created new opportunities for a wider range of applications which in turn impose their own requirements on future fog computing platforms. This article presents a study of a representative set of 30 fog computing applications and the requirements that a general-purpose fog computing platform should support.

1 Introduction

Fog computing extends cloud computing platforms with additional compute, storage and networking resources that are placed in the immediate vicinity of end-user devices. Because of its proximity to end users and their IoT devices, fog computing promises to deliver extremely low network latencies between the end-user devices and the fog computing resources serving them, and to process transient data produced by the end-user devices locally. Research in the domain of fog computing is currently very active and many researchers propose new mechanisms to design the next-generation fog computing platforms [1].

Fog computing researchers however need to face a difficult challenge: currently, no large-scale general-purpose fog computing platform is publicly available. To design useful fog computing platforms they, however, need to understand in detail which kind of applications will make use of fog computing technologies and which requirements they will put on the underlying fog platforms. On the other hand, few (if any) developers will spend significant time building applications which exploit the capabilities of fog computing platforms unless these platforms already actually exist.

To break this vicious circle we propose to study a representative set of fog applications which either were already implemented, or simply proposed for future development. We carefully selected 30 actual or proposed applications that cover a wide range of usage types for future fog computing platforms. We then use this set of reference applications to

All co-authors contributed equally to this paper.

address a number of crucial questions about the functional and non-functional requirements that a general-purpose fog computing platform should have. We show that fog applications and their respective requirements are very diverse, and highlight the specific features that fog platform designers may want to integrate in their systems to support specific categories of applications.

This article is organized as follows. Section 2 presents a general background about fog computing platforms and applications. Section 3 discusses our methodology for selecting a representative set of reference applications. Then, Section 4 analyses the requirements that these applications put on fog computing platforms and Section 5 concludes this article.

2 Background

Fog computing was originally designed as an extension of cloud computing with additional compute, storage and communication resources located close to the end users [2], [3], [4], [5]. The main purpose of this technology was to support the specific needs of latency-critical applications such as augmented reality [6], and IoT applications which produce massive volumes of data that are impractical to send to faraway cloud data centers for analysis [7]. Several excellent surveys of fog computing technologies are available [1], [8], [9].

However, the expected availability of fog computing technologies has created opportunities for other types of applications than the originally anticipated ones. These applications bring their own sets of requirements which must also be taken into account in the design of future fog computing platforms. The published fog computing surveys focus mostly on fog computing technologies (i.e., the solutions). To our best knowledge, no article so far has attempted at deriving the requirements for fog computing platforms from a representative set of actual and anticipated fog applications. This is the purpose of the current article.

3 Methodology

We base this study on a review of literature describing specific fog computing applications. The objective is not to build an exhaustive list of all proposed applications for fog computing, but rather to identify a representative sample of typical usages of fog computing technologies. Because fog computing is still an emerging technology, we included descriptions of both

[•] A. Ahmed, H. Arkian, A. J. Fahs, A. Gougeon, G. Pierre and P. R. Souza Jr are with Univ Rennes, INRIA, CNRS, IRISA.

[•] D. Battulga and M. Farhadi are with U-Hopper srl.

[•] D. Giouroukis and F.O. Gutierrez are with TU Berlin.

[•] M.A. Tamiru and L. Wu are with Elastisys AB.

actual implemented fog application and proposed future ones. We selected papers based on the following criteria:

Detailed application description. We sought for papers containing a detailed technical description of the proposed application, and discarded papers which proposed an idea with no further technical details.

Publication venue. We sought for papers which were published in peer-reviewed international conferences and journals. In addition, we also included white papers published by reputable corporations such as Cisco or organizations such as E.U. projects and the OpenFog Consortium¹.

Economic sectors. We aimed at identifying applications which cover a broad range of economic sectors such as transportation, healthcare, entertainment, smart cities, supply chain management, smart factories, robotics, agriculture, and security.

No overlap. To keep the list of application short, we avoided including multiple applications which resembled each other too much. In such cases we kept the most detailed description in the list, and discarded the other similar applications.

Tables 1–3 show the resulting list of reference applications.

The rest of the paper is organized to answer specific questions about the types of requirements a fog computing system should fulfill. For each such question, we studied the full list of applications to build an understanding of the requirements that different types of applications would have. It is of course up to each future fog computing systems to decide whether they choose to address some or all of these requirements.

4 Analysis

4.1 Why use a fog?

Fog computing was mainly proposed to deliver IoT services (i.e., mobility support, context-awareness, geo-distribution and low-latency) from the edge of the network [2]. By extending cloud datacenters resources i.e. compute, storage, and network resources at the closest vicinity of end users, fog computing also promises to enhance performance of many applications that require low latency from IoT devices to their closest fog server, or applications that process data locally where it is produced [3].

The number of applications running in fog platform is growing. The new use cases of fog platform are driven by innovative application design that requires additional platform characteristics which can only be delivered if the application were deployed next to the end users [40]. Therefore, we study all the referred applications and characterize them based on their reasons for using a fog platform:

Reduce latency: latency-sensitive applications such as augmented reality games require end-to-end latency (including network and processing delay) under 10-20 ms [41], [42]. However, the latency between an end user and the closest available cloud data center comes in the range of 20–40 ms (over wired networks) and up to 150 ms (over 4G mobile networks) [43]. Therefore, such applications cannot realistically

1. The OpenFog Consortium recently merged with the Industrial Internet Consortium. Their use case descriptions are temporarily offline while being rebranded to IIC documents. In the mean time we make these documents available at http://www.fogguru.eu/tmp/OpenFog-Use-Cases.zip.

run in the cloud. An obvious solution to reduce the end-toend latency is to deploy the server part of these applications in fog platforms.

Bandwidth optimization: edge devices such as IoT sensors, video surveillance camera produce large amount of raw data everyday [44], [45]. Sending such enormous amount of collected data to the cloud for processing creates huge network traffic [46]. For such applications, fog computing plays an intermediate role to reduce the network traffic. Fog middlewares are deployed in between the end device and the cloud to pre-process the raw collected data at the source and only the residual outcome are sent to the cloud for further processing [7].

Computational offloading: edge devices such as smart phones, smart IoT devices have limited processing capacity. Running compute-intensive applications such as face recognition in those devices is painfully slow. Offloading some execution of the application to the moderate fog servers may improve the performance. Offloading could be another way around: for instance, when a cloud server is overloaded, server-side of the running applications cloud be offloaded to fog servers [47].

Privacy and security: privacy and security are main concerns for many applications. For example, E-health applications in health care management record enormous amount of patient data for further study. Usually, those recorded data are sent to the public cloud for the long term storage [48]. However, data theft of personal medical records is one concerning issue faced by many hospitals [49]. Private fog mitigates the data privacy and security issue by delivering storage capacity on-premise of the users or the hospitals [50].

Service management: a growing number of devices like IoT sensors and actuators potentially require powerful computing machines to operate and manage the devices such as service deployment, fault management, hardware installation and device turn on/off etc [51]. Fog computing works as a middleware to provide computing power which not only enables one the control of the devices but also allows to customize the services based on the environment [52].

Monitoring edge devices: monitoring technology has seen improvement in hospitals such as monitoring infusion pumps, heartbeats, etc. However, the integration of such devices with patients is still challenging which leads to the third-leading cause of death each year in the US [19]. Fog computing enables to remotely host applications that directly communicate with the monitoring devices which allow to takes response dynamically based on the real-time data [53].

Energy efficiency: energy consumption by large IoT devices remains one open issue in the IoT environment [54]. Fog computing enables these devices to take decisions intelligently such as switch on/off/hibernate that reduces overall energy consumption [55].

Cost saving: traditional cloud charges the rented resources based on the usage, also known as a pay-you-go model. However, for some applications, a one-time investment cost for acquiring the private fog resources is preferable to the total cost of cloud instances.

Content caching: content caching or content delivery network or content distribution in fog platform is one way to reduce network traffic and improve response time by caching

$\begin{array}{c} \text{TABLE 1} \\ \text{List of reference applications (1/3)} \end{array}$

ID	Name	Source	Economic sector	Description
App01	LiveMap [10]	Peer reviewed	Transpor- tation	LiveMap is a scalable mobile information system that synthesizes vehicular update streams in real-time. It provides fine-grain, deep-zoom details about road conditions and hazards such as "Dead deer in left lane at GPS location (x,y), here is an image;" or, "Fog detected at GPS location (x,y), visibility down to 30 feet, here is a short video clip."
App02	Wearable cognitive assistance [11]	Peer reviewed	Health	Today, over 20 million Americans are affected by some form of cognitive decline. Google glasses integrate first-person image capture, sensing, processing and communication capabilities. Through context-aware real-time scene interpretation (including recognition of objects, faces, activities, signage text, and sounds), we can create software that offers helpful guidance for everyday life much as a GPS navigation system helps a driver.
Арр03	Live video broadcasting [12]	OpenFog Consortium	Entertain- ment	Today's sporting events need to broadcast live video from all corners of the arena or race course with zero latency. Fans demand to view real-time action on their mobile devices over a race course that spreads miles over terrain. Hierarchical fog nodes shorten video latency and decrease backhaul bandwidth. The Fog delivers the agility to manage video services and video algorithms which distribute the video process services in different layers from camera to cloud, according to their performance requirements.
Арр04	Visual security and surveillance [13]	OpenFog Consortium	Smart cities	Fog computing provides the architecture to build cost-effective, real-time and latency-sensitive distributed surveillance systems that help to preserve privacy challenges in open environments. Also, Fog computing enables real-time tracking, anomaly detection and insights from data collected over long time intervals.
Арр05	Traffic congestion management [14]	OpenFog Consortium	Smart cities	Fog computing gives municipalities a new weapon in the fight against traffic congestion. Fog has the flexibility to leverage traffic-related big data, which enables municipalities to take measures to alleviate congestion by connecting and analyzing previously unconnected infrastructure devices, roadside sensors, and on-board vehicles devices, in order to redirect traffic based on real-time data.
Арр06	High-scale drone package delivery [15]	OpenFog Consortium	Supply chain	Commercial drones operate in many environments, from aerial to subterranean. Fog enables near realtime adjustments and collaboration in response to anomalies, operational changes or threats. Fog computing enables drones, as self-aware individual fog nodes, to interoperate and cooperate as a dynamic community.
App07	Process Manufac- turing [16]	OpenFog Consortium	Smart Factories	In order to meet market demand, food and beverage producers must be able to cope with small-quantity, large-variety products, along with product lifecycles with large fluctuations in demand periods and quantities. Fog computing helps process brewing by enabling digital twins of process in order to replicate key functions, enabling fog nodes to scale up or down to meet demand, and ensuring privacy of data.
Арр08	Smart Buildings [17]	OpenFog Consortium	Smart Spaces	Today's smart buildings are leveraging the IoT for improved business outcomes, such as better energy efficiency, improved occupant experience, and lower operational costs. They typically contain thousands of sensors measuring various building operating parameters such as temperature, humidity, occupancy, energy usage, keycard readers and air quality. This use case demonstrates how fog nodes at the room level, floor level, building level and cloud level can be hierarchically architected for efficient real-time processing, enabling dozens of new applications.
Арр09	Real-time Subsurface Imaging [18]	OpenFog Consortium	Energy-Civil- Environment Industry	Subsurface imaging and monitoring in real time is crucial for understanding subsurface structures and dynamics that may pose risks or opportunities for oil, gas and geothermal exploration and production. This use case integrates IoT sensor networks with fog computing and geophysical imaging technology. Fog's scalability enables real-time computation in remote field locations, including support for complex compute algorithms.
Арр10	Patient Monitor- ing [19]	OpenFog Consortium	Smart Healthcare	With its real-time communications and analytics requirements for data from thousands of low-level sensors, today's hospital patient monitoring requires the scalability and agility of fog. This use case is based on a virtual compute environment residing on a series of fog nodes that supports the flexible deployment of applications and streamlines the integration of healthcare systems.

 $\begin{array}{c} \text{TABLE 2} \\ \text{List of reference applications (2/3)} \end{array}$

ID	Name	Source	Economic sector	Description
App11	Autonomous	OpenFog	Smart	Autonomous Driving, which involves hundreds of simultaneous data
	Driving [20]	Consortium	Transpor-	processes and connections, can't be accomplished without fog. Fog
			tation	establishes trustworthiness of communications between low-level
				sensors while enabling high-bandwidth real-time processing. This
				use case validates how fog architectures for autonomous cars enable
				significantly greater scalability than any other architecture.
App12	Robots Simulta-	Peer reviewed	Smart	By leveraging key principles of fog computing that enable processing
	neous Localiza- tion And Map-		Robotics	to take place in close proximity to the robots, SLAM is enabled by
	ping [21]			high-performance real-time edge processing, optimized analytics,
	p8 [21]			and heterogeneous applications. The SLAM use case speeds up
				the time to process vast amounts of data required in life-or-death
App 19	Mobility-	Cisco white	Transpar	situations such as firefighting or rescue operations. Fog computing offers the potential to understand latent transport
App13	as-a-Service	paper	Transpor- tation	demand in real-time, and to rapidly assemble insights which can
	(MaaS) [22]	paper	tation	allow MaaS networks to quickly deploy services and get people
	(111445) [22]			moving. The objective here is to deliver a demand-responsive trans-
				port ecosystem, where the MaaS network enables multiple mobility
				operators to detect and understand customer demand in real-time.
App14	ARQuake [23]	Peer reviewed	Entertain-	ARQuake application is based in the old famous shooter called
711111	micadane [20]	1 cci icvicwed	ment	Quake. The augmented reality information (monsters, weapons,
			1110110	objects of interest) is displayed in spatial context with the physical
				world using 3D objects.
App15	FAST [24]	Peer reviewed	Health	Stroke (Brain attack) - distributed analytics system to monitor
111110	11101 [21]	1 cer reviewed	11001011	fall for stroke mitigation, fall detection algorithms and incorpo-
				rated them into fog-based distributed fall detection system, which
				distribute the analytics throughout the network by splitting the
				detection task between smart phones attached to the users and
				servers in the cloud.
App16	eWall [25]	Peer reviewed	Health	COPD and Mild Dementia are related to aging. eWall provides
	. ,			an intelligent home environment with personalized context-aware
				applications based on advanced sensing and fog computing on the
				front and cloud solutions on the back.
App17	Smart street	Peer reviewed	Smart cities	Safety and energy consumption of Street Lamp is a major concern
	lamp [26]			in Smart Cities. The smart street lamp application deploys various
				sensors in each lamp which collects surrounding real-time data such
				as the intensity of brightness, human presence, voltage level, current
				level, etc. They are also equipped with NB-IoT communication
				which is used to send the collected data to the managing server.
				The managing server analyzes the collected data and detects any
				fault in the lamp. The computing capacity in the individual lamp
				enables to adjust the brightness intensity of the light depending on
	_			the sensor data therefore, saves a huge amount of the energy.
App18	Power	Peer reviewed	Smart Grid	Modern home-based IoT devices such as electric sensors produce
	Consumption Manage-			huge amount of data, which are transferred to cloud for further
	ment [27]			processing using long-WAN. The proposed implementation offloads
	[2,]			some of the cloud tasks to Fog compute nodes and therefore reduce
				latency. The Fog compute nodes monitor the usage of electricity for
App10	Vehicular Video	Doon nor i arra 1	Trongram	each member to implement the home energy management system. Processing large volume of high-quality videos from vehicles is
Арр19	Processing [28]	Peer reviewed	Transpor- tation	challenging. The large volume of raw video data is usually sent to
	1 100003111g [20]		0401011	the cloud through long-WAN and then processed and analyzed and
				finally, end-users retrieve the result from the cloud. The authors
				study the feasibility of a large volume of video transmission at the
				local fog server which is formed combining the computing capacity
				of collocated vehicles.
App20	Vehicular Pollu-	Peer reviewed	Transpor-	This application aims to real-time process gas sensor data at the Fog
AFP20	tion Control [29]	1 cer reviewed	tation	server to reduce the latency. It deploys a fog node in each traffic post
	0.011 0.01101.01 [23]		56601011	to process the gas sensor data generated from surrounding vehicles.
				The k-means cluster algorithm groups the sensor data and identify
				the pollution level at real-time. If the level of pollution is critical
				then it notified to the Pollution control Board.
<u> </u>		<u> </u>		onon to nounce to one I one on colleton boate.

 $\begin{array}{c} \text{TABLE 3} \\ \text{List of reference applications (3/3)} \end{array}$

ID	Name	Source	Economic sector	Description
App21	FogLearn [30]	Peer reviewed	Healthcare	FogLearn is a three-layer cloud architecture framework for Ganga River Basin Management and for detecting diabetes patients suffering from diabetes mellitus. In order to reduce long WAN-latency, the Fog layer firstly pre-process the collected data from the the Edge Layer and then send to the Cloud for further analysis and long term storage. The Fog Layer also has the capacity to scale for analysis the data in case of heavy workload in the cloud.
Арр22	Telemedecine [31]	Peer reviewed	Healthcare	To diagnose and evaluate a patient, the healthcare professionals need to access the electronic medical record (EMR) of the patient, which might contain huge multimedia big data including X-rays, ultrasounds, CT scans, and MRI reports. The main focus has been given to secure healthcare private data in the cloud using a fog computing facility.
Арр23	SWAMP [32]	EU project	Agriculture	SWAMP develops a high-precision smart irrigation system concept for agriculture to enable optimizations of the irrigation system, water allocation, and consumption based on a sensor analysis collected from the fields. The collected data are transmitted to the central fog node deployed in agriculture office through the base station and hosted to access the field data to the farmers. However, the analysis and modeling of the field data is done in the powerful traditional cloud.
Арр24	Smart Waste Manage- ment [33]	Peer reviewed	Smart cities	Waste management is one of the toughest challenge that modern cities have to deal with. A city council may use sensor data to develop optimised garbage collection strategies, so they can save fuel cost related to garbage trucks.
Арр25	Drone traffic surveillance with tracking [34]	Peer-reviewed	Security	An urban speeding traffic monitoring system with tracking using drones, which are connect to Fog Nodes in order to process the images. Leveraging the divide-andconquer strategy, the subarea containing the vehicle of interests was identified and transmitted to the Fog node for processing.
Арр26	GPU-assisted Antivirus Protection in Android Devices [35]	Peer-reviewed	IT security	We first describe a GPU-based antivirus algorithm for Android devices. Then, due to the limited number of GPU-enabled Android devices, we present different architecture designs that exploit code offloading for running the antivirus on more powerful machines. This approach enables lower execution and memory overheads, better performance, and improved deployability and management.
Арр27	Cachier [36]	Peer-reviewed	Entertain- ment	Recognition and perception based mobile applications, such as image recognition, are on the rise. These applications are latency-sensitive. Cachier uses the caching model along with novel optimizations to minimize latency by adaptively balancing load between the edge and the cloud, by leveraging spatiotemporal locality of requests, using offline analysis of applications, and online estimates of network conditions.
Арр28	EdgeCourier [37]	Peer-reviewed	Broad	Using cloud storage to automatically back up content changes when editing documents is an everyday scenario. EdgeCourier proposes the concept of edge-hosed personal service, which has many benefits, such as helping deploy EdgeCourier easily in practice.
Арр29	MMOG [38]	Peer-reviewed	Entertain- ment	With the increasing popularity of Massively Multiplayer Online Game (MMOG) and fast growth of mobile gaming, cloud gaming exhibits great promises over the conventional MMOG gaming model as it frees players from the requirement of hardware and game installation on their local computers. CloudFog incorporates "fog" consisting of supernodes that are responsible for rendering game videos and streaming them to their nearby players.
Арр30	Edge Content Caching for Mobile Streaming [39]	Peer-review	Entertain- ment	Increasing popularity of mobile video streaming compels video service providers to move from traditional content caching to edge network content caching. The authors uses real-world dataset of mobile video streaming to study the request pattern and user behaviors. Then analyse the performance of edge content catching in the WiFi access network and cellular network. Based on the analysis authors proposed an efficient caching strategy in edge environment.

TABLE 4
Reasons for using fog computing.

Fog usages	Applications	Total
Reduce	APP02, APP03, APP04, APP06,	20
latency	APP08, APP09, APP11, APP12,	
-	APP13, APP14, APP15, APP16,	
	App17, App18, App20, App21,	
	App25, App27, App29, App30	
Bandwidth	App01, App02, App04, App05,	16
saving	App07, App08, App11, App14,	
	App15, App16, App18, App19,	
	App20, App21, App23, App24	
Computational	App02, App09, App18, App26,	5
offloading	App29	
Privacy & se-	App10, App18, App22	3
curity		
Service man-	App05, App08, App18, App24,	5
agement	App28	
Device moni-	App09, App10	2
toring		
Cost saving	App24	1
Energy	App17	1
efficiency		
Caching	App27, App30	2

popular content locally [56]. This practice has been well studied, matured, and was able to benefit many applications [57], [58]. Traditionally cloud has been used to deploy content delivery network [59]. However, with fog computing, the contents can be cached with fine granularity based on the user's locality and popularity of the content [60].

Table 4 presents a comparison of the surveyed applications based on the above fog usage. It is not surprising that most of the applications leverage fog platform to reduce end-to-end latency and optimizing bandwidth consumption. However one cannot ignore the other reasons why a fog platform may be used but different applications.

4.2 Fog deployment models

We can classify fog models based on the ownership of the fog infrastructure and underlying resources. There are four different types of fog models:

Private fog: A private fog is created, owned, managed and operated by some organization, a third party, or some combination of them. It may be deployed on or off-premises. The resources of private fog are offered for exclusive use by a single organization (e.g., business units).

Public fog: A public fog is created, owned, managed and operated by a company, academic institute or government organization, or some combination of them. It is deployed on the premises of the fog providers. The resources of public fog are offered for open use by the general public.

Community fog: The community fog is created, managed and operated by one or more organizations in the community, also a third party, or a combination of them. It may be deployed on or off-premises and the resources are offered for exclusive use, usually by consumers of a specific community of organizations that have shared concerns.

Hybrid fog: A hybrid fog is a form of fog computing that combines the use of public/private/community fog with public/private cloud (i.e., hybrid cloud). It can be useful due to physical resource limitations in the fog. Therefore, the platform is extended to the hybrid cloud to scale performance. A

TABLE 5
Fog infrastructure models

Fog	Applications	Total
model		
Private	App02, App04, App07, App08, App09,	13
Fog	APP10, APP12, APP15, APP17, APP19,	
	Арр20, Арр24, Арр30	
Hybrid	App01, App03, App05, App06, App11,	17
fog	APP13, APP14, APP16, APP18, APP21,	
	App22, App23, App25, App26, App27,	
	App28, App29	

hybrid cloud is scalable, elastic, and resources are available on-demand.

Table 5 classifies the applications based on the underlying fog models used to deploy the applications. We found that nearly half (13 out of 30) of the surveyed applications are deployed in the *private fog* and the majority (17 out of 30) in the *hybrid fog*. However, none of the applications are deployed in *public fog* and *community fog*. Remarkably the applications can be further categorized based on their requirements and functionality provided by the respective fog models. We, therefore, identified the following main reasons for choosing a *private fog*:

- Privacy and security: many applications that deals with personal data such as wearable devices produce a large volume of personal data which are too risky to deploy in the open cloud for privacy and security concern and, therefore, those applications are usually preferred to deploy in a secure fog cloud usually owned by the individual or trusted third party. Similarly, many industries prefer to deploy a secure cloud to run automatic robotic application due to security reasons.
- Latency sensitivity and modest resource requirements: applications that require moderate resources and low latency are deployed in fog platform such as web hosting.
- Cost-saving: traditional cloud charges rented resources based on the usage, also known as a pay-as-you-go model. For some applications, the cost is less expensive while deploying the application in on-premise, particularly those applications that do not need high scalability and maintenance. For such applications, one-time cost for acquiring the private fog resources is cheaper than the traditional cloud.

A hybrid fog mainly aims to scale the resources of fog platforms; therefore, applications that require an enormous amount of resources (i.e., computation, storage, etc.). We further categorized the applications deployed in hybrid fog based on the following criteria:

- Compute intensive applications: applications that require relatively high computation such as big data analytics in Swarm project APP23, face recognition APP04 are preferred to deploy in hybrid fog.
- Storage: applications such as APP25 that need to store a large amount of data for future reference use hybrid fog.

4.3 Types of access networks

Access networks connect the IoT devices to the fog platform, and are therefore a important basic building blocks for fog

platforms. Data generated from applications need to be processed and acted upon in terms of milliseconds, therefore, the network architecture should support ultra-low latency, and large data volumes. There are many standards and types of access networks that can be deployed on fog environment.

Considering the complexity of fog nodes topologies, and due to their distribution and mobility, wireless connectivity is essential in fog environment. Wireless connectivity provides flexibility, mobility, and reachability for levels of hierarchies of fog communication.

As mentioned in the OpenFog Reference Architecture [61], wireless support at the fog node will depend on variety of parameters: function and position in the hierarchy, mobility, coverage, range, throughput and data transfer rates, etc.

Various wireless technologies were classified in Table 6, depending on their frequency, coverage, data transfer rate, and power consumption:

- Low Power Wide Area Network (LPWAN) is a protocol for resource-constrained devices and networks over long ranges.
 It covers tens of kilometers, and provides low data rate and power usage. Agriculture is the perfect use case for using LPWAN technology. Some of the protocols based on LPWAN are LoRa and SigFox.
- Cellular networks are suitable for long distance communications in IoT applications. However, all mobile network protocols come at a high price due to their licensed Radio Frequency, intellectual property protection, and high power consumption. NB-IoT and LTE-M standards are aimed at providing low-power, low-cost IoT communication options using existing cellular networks, which can be categorized as LPWAN technology. The 5th Generation cellular network (5G) is starting to get commercialized, and will improve IoT communications. It also promises to lower costs, battery consumption, and latency.
- There exists a wide range of devices with Wi-Fi compatibility, with IEEE 802.11 being the most popular network protocol for Local Area Networking (LAN). Its high power usage, high data transfer rate, and medium-range make it a popular option for latency-aware fog applications. HaLow (IEEE 802.11ah) and Wi-Fi 6 (IEEE 802.11ax), were designed to address the constraints of IoT networks.
- The traditional frame format of MAC layer protocols was not suitable for IoT low power and multi-hub communications. IEEE 802.15.4 was created with a more efficient frame format that has become the most used IoT MAC layer standard. Applications such as home automation are suitable to use low data rate, medium range Zigbee, 6LoWPAN technologies.
- Lastly, other short range Personal Area Network (PAN) technologies such as Near Field Communication (NFC), Bluetooth Low Energy (BLE), and Radio Frequency Identification (RFID) can be used for personal IoT devices like wearable health and fitness trackers, asset tracking, check-in systems.

Depending on the area that must be covered by the fog platform, different types of antennas can be proposed. For example, directed Wi-Fi antennas can be used for Live Video Broadcasting application APP04.

Table 7 illustrates the possible access network usages for each application. Some applications are shown multiple times based on their usage of different access network types on levels of deployment. Also some of the applications have already been evaluated in real life testbed. For example, Power Consumption Management application (APP22) proposed to use Zigbee, Vehicle Video Processing (APP23) examined Dedicated Short Range Communication (DSRC) and LTE, and deployed promising Vehicular Fog Computing (VFC) architecture.

4.4 Hardware platforms

The servers which constitute a fog computing platform may be highly heterogeneous, not only physically but also in terms of resource capacity such as processing, storage, and network bandwidth [72], [73]. They are considered the building blocks of fog infrastructures [74]. Unlike in traditional cloud platforms, fog architectures are composed of large number of small computing fog nodes or servers which are placed in strategic locations across a vast geographical area to cover a large number of users [75]. Since the fog nodes can be deployed anywhere between the end users and the cloud, the latency between an user and the nearest fog nodes largely depends on the location where the fog nodes are deployed [76], [77]. Depending on the application requirements, application developers have to choose appropriate fog nodes to improve the QoS of the applications. We therefore explore different fog nodes used to deploy the surveyed applications.

Table 8 compares different characteristics of *potential* fog nodes. We broadly classify the fog nodes into static and mobile nodes.

Static nodes are placed in strategic locations. Some examples of static nodes are base stations, small-scale datacenters and personal laptops, switches, routers, etc. Such devices are practically not mobile, therefore need to be placed in a fixed location. Statics nodes can be further categorized based on the premise where they are deployed. For example, base stations, network resources (switches and routers), small-scale datacenters.

Mobile nodes are movable, physically small, less resourceful and flexible to install and configure. Some examples of mobile nodes are single-board machines [78] (RPIs, Pine A64+, etc.), drones, vehicles, etc. They may have small computation capacity, however the resources can scale horizontally by aggregating nodes [79] as discussed in Section 4.5.

Table 9 shows the types of fog nodes that are used to deploy our reference applications. Many of the surveyed applications may be deployed using one or multiple *potential* fog nodes depending on various requirements (i.e. computation capacity, proximity, etc.). We can however draw a number of conclusions for different types of applications:

Applications focusing on close proximity: IoT Applications that require compute resources at very close proximity of end-users, often take the benefits of single-board machines. Some examples of such applications are App08 and App17 collect sensor data from the surrounding environment of the fog nods and process locally. Due to the small physical size of single-board machines, they could easily be deployed and move from one place to another. Therefore applications such as App21 use such small machines to trail computation where the robot moves.

TABLE 6
Types of access networks.

Access Network	Technology	Frequency	Transfer rate	Range	Power consumption
LPWAN [62]	LoRaWAN	~900MHz	$0.3-50 \mathrm{kbit/s}$	2-5km (urban) 15km (rural) [63]	low
	SigFox	900MHz	10-1000bit/s	3-10km (urban) 30- 50km (rural) [63]	low
	NB-IoT [64]	various	250 kbit/s	<35km	low
Cellular	4G LTE [65]	700-2600MHz	100Mbit/s	1-10km	high
Cenular	5G	various [66]	700Mbit/s [67]	less than 4G	high
Wi-Fi	Wi-Fi 4 IEEE 802.11n [68]	2.4GHz 5GHz	$150 \mathrm{Mbit/s}$	70m (indoors) 250m (outdoors)	high
VV 1-F 1	Wi-Fi 5 IEEE 802.11ac [68]	$5\mathrm{GHz}$	860Mbit/s	35m (indoors)	high
	HaLow IEEE 802.11ah [69]	sub 1GHz	78Mbit/s	1000m	medium
	Wi-Fi 6 IEEE 802.11ax [70]	2.4GHz 5Ghz	600- 1800Mbit/s	76m (indoors)	medium
802.15.4 based	Zigbee	$2.4 \mathrm{GHz}$	250 kb/s	10-100m	low
802.15.4 based	6LoWPAN	\sim 900MHz 2.4GHz	250kb/s	10-100m	low
	Bluetooth Low Energy (BLE)	$2.4 \mathrm{GHz}$	1Mbit/s	15-30m	low
Other PANs [71]	RFID	125kHz 13.56MHz 902-928MHz	4Mbit/s	<200m	very low
	NFC	125Khz 13.56Mhz 860Mhz	106-424kbit/s	10cm	very low

TABLE 7
Classification of applications based on access network types.

Access	Applications	Total
Network		
LPWAN	App01, App08, App09, App20, App21	5
Cellular	App01, App03, App05, App06, App09,	15
	APP11, APP12, APP13, APP15, APP23,	
	Арр19, Арр25, Арр26, Арр29, Арр30	
Wi-Fi	App03, App04, App06, App07, App08,	16
	APP10, APP12, APP14, APP17, APP22,	
	App25, App26, App27, App28, App29,	
	App30	
802.15.4	App07, App08, App12, App16, App17,	8
based	App18, App22, App24	
Other	App02, App07, App10, App15, App22,	6
PANs	App24	

Vehicular-based applications: applications which collect data from roadside and process locally such as vehicle video processing, autonomous driving, traffic congestion management etc. often take advantage of in-built computation capacity of the vehicles.

Drone-based applications: drone-based applications such as APP25 and APP15 take advantage of drones to relocate computation resource from one place to another. The drone is equipped with single-board machines that allows to process locally and makes communication with other fog nodes.

Compute-intensive applications: applications such as APP04 and APP09 require relatively high computation power, therefore they are usually deployed in laptops or small-scale datacenters.

4.5 Distribution within the fog

Distribution is a key element of fog computing. Locating one node at the edge of the network is often not sufficient to deliver low latency, as nodes should be distributed to cover a certain geographical area. The diffusion of the nodes grants access to nearby resources for all the users located in a specified area. Fog computing platforms can provide distribution in two different forms:

- Hardware distribution: portrayed by the distributed nodes.
- Software distribution: portrayed by the distribution of the applications' instances and components.

For hardware distribution, two common solutions are available. The first is based on having multiple nodes at the same layer in the architecture, which is referred to as Horizontal Distribution of the nodes. Meanwhile, if the nodes vary in the size of their resources (e.g., through some hardware upgrade), then they implement Vertical Distribution. Typically the nodes that have limited resources are distributed on the edge, however, nodes with more resources will be placed in a higher vertical layer to serve a greater number of users. The end of these vertical layers is the cloud with unlimited resources.

Applications, in general, can be distributed over the cluster either using *Replication*, where more than one instance of the same application are placed in different nodes, or by splitting the application into *Multi-Component*, where each component is usually a microservice located in a separate node.

Most of the applications intending to run on top of fog try to invest and benefit from the distribution offered by fog. Some of the applications that are based on widely-distributed users require a congruent distribution of the fog nodes. This trend was notable in IoT-based applications, where for example, in [26] the IoT devices cover the lamps in the street, and the nodes should be placed according to the placement of these devices. This enforces a horizontal distribution with replication.

TABLE 8 Characteristics of fog nodes.

Characteristics Fog nodes	Processing	Storage	Network	Physical size	Distance from users	Mobility	Cost
Single-board computers	-	-	+/-	-	-	+	-
Vehicles (cars, buses)	-	-	+/-	-	-	+	-
Drones	-	-	+/-	-	-	+	+/-
Network resources	-	-	+	-	-	+/-	+/-
Laptop / PC	+/-	+/-	+/-	+/-	+/-	-	+/-
Small scale datacenters	+	+	+	+	+	-	+

Legends: "+" means high, "+/-" means neutral, "-" means low.

TABLE 9 Fog nodes used to deploy the applications.

Fog nodes	Applications	Total
Single-	App03, App05, App07, App08,	10
board	APP11, APP12, APP17, APP18,	
computers	App21, App23,	
Vehicles	Арр11, Арр19, Арр20,	3
(cars,		
buses)		
Drones	App06, App25,	2
Network re-	Арр03, Арр18, Арр30	3
sources		
Laptop /	App02, App03, App25, App26, App28	5
PC		
Small scale	App01, App04, App09, App10,	16
datacenters	App11, App13, App14,	
	APP15, APP16, APP17, APP18,	
	App22, App23, App24, App27,	
	App29,	

We have noticed that some applications rely on software distribution and more specifically replication of the same components over different nodes. This approach was evident in the case of processing-intensive applications like video stream processing. The reason behind this replication is lowering the latency by placing the replica in different nodes, and improving performance by providing more resources [28].

For applications that require vertical distribution, the fog cluster is layered in a way that the edge node will collect the data, that will later be sent to the fog nodes which will process the data and send only the results to the cloud. This architecture is very efficient in the case of data streams, such that one does not need to send all the collected data but rather the output results. This is an effective way to reduce latency and traffic volume over the Internet connection [28].

Most of the applications that require no distribution at all, are mainly applications that will run only on the edge or applications that will load balance between the edge and the cloud.

Table 10 summarizes the surveyed applications and the types of distribution they require. $\,$

4.6 Fog service models

Much like cloud platforms, there are multiple ways by which a fog computing platform may expose low-level or high-level virtualized resources to its users. We can classify them in three categories depending whether they offer infrastructure, platform, and software. We call them FogIaaS, FogPaaS and FogSaaS to differentiate them from them cloud-only counterparts.

 ${\sf TABLE~10}$ Hardware and software distribution according to the general category.

Category	App	Hardware	Software
Autonomous	App06	Horizontal	Replication
vehicle	App11	Horizontal	Replication
venicie	App12	Vertical	Multi-component
Data	App22	None	None
Storage	App28	Vertical	None
	App07	Horizontal	Replication
	App08	Both	Rep. & Multi.
	App09	Horizontal	Replication
	App10	Horizontal	Replication
	App13	Both	Replication
	App14	Horizontal	Replication
ІоТ	App15	Horizontal	Replication
101	App16	Horizontal	Replication
	App17	Horizontal	Replication
	App18	Both	Replication
	App20	Horizontal	Replication
	App21	Horizontal	Replication
	App23	Horizontal	Replication
	App24	Horizontal	Replication
	App01	Horizontal	Replication
Real	App02	None	None
Time	App05	Horizontal	Replication
Time	App25	Both	Replication
	App26	None	None
	App03	Both	Rep. & Multi.
Media	App04	Horizontal	Replication
Streaming	App19	Both	Replication
Streaming	App27	Vertical	None
	App29	Horizontal	Replication
	Арр30	Vertical	None

FogIaaS (Fog-Infrastructure-as-a-Service) allows users to take advantage of different hardware, such as CPU, network, disk, etc. without mentioning the hardware running behind it. The users have the independent choice to deploy any Operating System and other utilities on the provided resources.

FogPaaS (Fog-Platform-as-a-Service) provides endusers to access basic operating software and optional services for running applications and software development environment. FogPaaS builds on top of FogIaaS and makes development, testing, and deployment of a software quick, robust, and cost-effective.

FogSaaS (Fog-Software-as-a-Service) provides endusers to use software applications without installing them on their personal computer. The services are accessed from the web browser remotely through a network.

Table 11 classifies the reference applications based on the service models they rely on. In particular, only one application

TABLE 11 Fog service models.

Service models	Applications	Total
FogIaaS	APP01, APP02, APP03, APP04, APP05, APP06, APP07, APP08, APP09, APP10, APP11, APP12, APP14, , APP17, APP22, APP24, APP28, APP29 , APP30	19
FogPaaS	APP13, APP15,APP16, APP18, APP19, APP20, APP21, APP23, APP25, APP26	10
FogSaaS	App27	1

uses FogSaaS while the others uses either FogPaaS (10 out of 30) or FogIaaS (19 out of 30).

4.7 Required middlewares

As an increasing number of fog computing applications are being developed, fog platforms may need to provide greater numbers of middleware systems (in the form of FogPaaS services) to support easy application development. We survey the most used types of middlewares below.

Data stream processing systems were initially developed by the Big Data community [80]. However, multiple authors also recognized their interest in a fog computing environment, where they have the potential of reducing data transfers between IoT devices and the cloud [81]. Multiple systems have been developed with a number of variations in their provided features [82].

Function-as-a-Service supports the development of event-driven, serverless applications. It enables one to develop, run, and manage functions or pieces of code without provision or management of servers. Fog computing is aiming to leverage the possibility to use IoT devices in a serverless architecture, which is a type of Function as a Service. This is essentially some extension of the Cloud services to the edge where there are the IoT and mobile devices, web browsers, and other computing at the edge.

Message-oriented middleware (MOM) is a method of communication between software components in distributed systems. It can be defined as a software or hardware infrastructure which aims to support receiving or sending messages between distributed and heterogeneous components. It aims to reduce the complexity of developing applications across multiple operating systems and network protocols [83]. Message-oriented middleware are being used in fog computing environments to improve the scalability of Fog nodes and task scheduling [84].

Web application servers are software frameworks which provide an environment where applications can run regardless of what they do. They usually contain comprehensive service layers where each one addresses a separate concern. Common application servers running on the cloud can serve up web pages, provide a container model or services for applications, adhere to specification controlled by industry, distributed requests across multiple physical servers, and provide management and/or development tools. Fog computing enhances the capabilities of application servers in the direction of facilitating management and programming of computing, networking, and storage services between data centers and end devices.

TABLE 12 Classification of required types of middlewares.

Required middleware	Applications	Total
Stream	App18, App19, App21, App25, App30	5
Processing		
Engines		
Function as	App12, App16, App18, App19,	10
a Service	App20, App24, App25, App26,	
	App28, App29	
Message-	App01, App12, App13, App16,	7
oriented	App 17 , App 23 , App 24	
middleware		
Web	App02, App14, App17, App18,	11
application	App20, App22, App23, App26,	
servers	App27, App28, App29	
Unspecified	App03, App04, App05, App06,	20
	App07, App08, App09, App10,	
	App11, App15	

TABLE 13
Data processed by applications.

Data Type	Applications	Total
Textual informa-	App22, App28	2
tion only		
Sensor informa-	Арр07, Арр08, Арр10, Арр13,	12
tion (excluding	Арр15, Арр16, Арр17, Арр18,	
camera)	App20, App21, App23, App24	
Imaging data,	App09, App26, App27	3
CGI or GPU		
computing		
(excluding video		
processing)		
Video (and possi-	App01, App02, App03, App04,	13
bly other sensor	App05, App06, App11, App12,	
information)	Арр14, Арр19, Арр25, Арр29,	
,	App30	

Table 12 lists the types of middleware used by the reference applications. Some applications employ more than one type of middleware, so they are listed multiple times. Those applications that did not reference any specific type of middleware are described as unspecified.

4.8 Types of processed data

The type of data processed by an application provides an idea about the required computing capacity for the nodes constituting the Fog architecture. This information is closely related to the data volume produced by the applications and the timeliness of its processing, detailed in Section 4.10.

Fog architectures are often seen as a widely distributed network based on horizontal scalability [1]: if a pool of resource is insufficient, then a simple solution is to connect more nodes to the Fog. Nevertheless, the distribution of a single task over several nodes is rarely mentioned. Nodes are often supposed to be able to host a full task [47] and therefore should have adapted processing and storage capacity to respect the timeliness of applications.

Table 13 classifies the applications according to their processed data. A handful of applications process only textual information, and arguably do not require very large processing capacity. Applications processing sensor information may be delay-sensitive. Depending on the type and number of sensors they may require varying amounts of processing capacity. We

TABLE 14 Data volume.

Data Volume Scale	Applications	Total
Kilobytes	App12, App13, App16, App17, App20, App21, App30	7
Megabytes	APP02, APP05, APP06, APP10, APP14, APP15, APP19, APP24, APP26, APP27, APP28, APP29	14
Gigabytes	APP01, APP03, APP07, APP08, APP18, APP23, APP25, APP04,	7
Terabytes	App09, App11, App22	4

however observed that many of the most demanding applications actually process either static images or even video. This often requires the usage of specialized devices such as GPUs to process incoming (live) video streams in real time.

4.9 Data volume

The constant growth of data production is due to the emergence of the use and collection of sensor data and process automation. Data volume is the most important aspect in the point of view of the Big Data community, and it is regarding the amount, size, scale of data produced and stored [85].

We observe that few applications use the fog to store significant amounts of data. Data storage is usually delegated to cloud computing platforms whereas the fog platforms are mostly used to process streams of data which are supposed to be processed or filtered quickly, possibly before being returned to the users [73].

Table 14 shows a classification of the volume of data processed by each application in broad categories ranging from kilo-bytes to tera-bytes. We observe that a majority of applications handle relatively modest amounts of data, in the order of kilo-bytes to mega-bytes. This is because there is a high demand for real-time services which relies on receiving, processing, and forwarding the information without keeping any history or data persistence. For instance, data stream application or message-oriented applications may require greater storage capability in situations where it is necessary to maintain fault tolerance and high parallelism using multiple replicas. We observe similar strategies for applications which handle larger volumes of camera-generated data (e.g., App04 and App11).

Fog applications which belong to the tera-byte scale usually require persistent and distributed storage in fog nodes (e.g., APP22) either for keeping history data generated by cameras or personal files. However, the volume of data they handle largely depends on their number of users and IoT devices. Applications that use machine learning or deep learning (e.g., pattern recognition from videos cameras feeds) may need to frequently retrain their models so storing large amounts of historical data is important for them. All these applications usually make use of private for platforms where sufficient storage capacity may be provisioned for specific applications.

4.10 Data velocity and latency sensitivity

One of the fundamental advantages of fog computing is delivering low user-to-resource latency [2], [77]. The widely distributed nodes serve this purpose where some of the resources

TABLE 15
Data velocity requirements.

Bandwidth Latency	kBps or less	MBps or more	Unspecified
${<}10\mathrm{ms}$	Арр20 Арр06	App03 App04	App08 App13
		App11 App14	
$[10\mathrm{ms},\!100\mathrm{ms}]$	App 02 App 12	App 25 App 29	_
	App18	App30	
$[100\mathrm{ms,}1\mathrm{s}]$	App28	App27	
$[1{ m s}{,}10{ m s}]$		App09	
>10 s	App10	App01 App23	App05 App21
Unspecified	App 07 App 22	_	App15 App16
			App17 App19
			App24 App26

should be in the vicinity of the end-users. As a result, the promised low latency of fog is an incentive for applications seeking ultra-low latencies that may not exceed a couple of milliseconds regardless of the velocity of the input data.

A very similar notion promoted by the Big Data community is data velocity, which refers both to the speed at which new input data is being produced, and to their expected end-to-end processing latency. In the domain of IoT, the increasing availability of connected devices and the rapid development of interconnected applications are leading to continuously increasing rates of input data that must be processed in a timely fashion [86].

We classify the reference applications according to the two dimensions of data velocity:

Data production speed: depending on the applications, the input data may be produced in the order of kilobytes per second or less, megabytes per second, or more.

Expected response latency: depending on the application, the results may be expected within couple of milliseconds, hundreds of milliseconds, seconds, or possibly more.

Table 15 classifies the reference applications according to these two metrics. We can observe that most of the input data production of the applications are relatively low, in the order of MBps or even kBps. Such data production rates may initially seem easy to handle, however, most of the applications require a very quick response for the generated data. As a result, an increased data production rate could be challenging for any fog computing platform.

Throughout the reviewed applications, we have noticed certain trends. While some IoT-based applications such as APP17 and APP20 require no latency restrictions, a significant number of the applications mention low latency as a vital component for proper functioning. However, the exact meaning of "low latency" varies a lot depending on the applications. Some applications such as APP06 and APP29 require extremely low latency under 5 ms, whereas others such as APP08 and APP09 can operate with much softer delivery time constraints. The applications that require ultra-low latencies can be classified in two categories:

Real-Time Decision Making: For these applications such as APP06, APP11 and APP12, ultra-low latency is a matter of human safety. For example, a set of autonomous vehicles calculating their trajectory in a fog platform require very fast decisions to avoid crashes.

TABLE 16
Data providers.

Number of Data Providers	Applications	Total
Single data	App01, App02, App08, App09,	8
provider	Арр12, Арр14, Арр16, Арр20	
Multiple	App03, App04, App05, App06,	16
data	APP07, APP10, APP11, APP13,	
providers	APP15, APP17, APP18, APP19,	
_	App20, App22 App23, App24	

High-Quality User Experience: Applications falling in this category often belong to the entertainment sector for gaming or video streaming (APP02, APP29, APP30). In such applications, excessive latencies will not cost human lives but may seriously affect the users' experience.

Data velocity requirements are an important driver for the design of any large-scale fog computing platform which must be able to process input data very close to the location where they have been generated, and to forward only pre-processed data further to other fog or cloud-hosted resources.

4.11 Multiple data providers

A number of fog applications need to integrate data which originate from multiple independent data providers. For instance, APP06 provides item deliveries through the usage of drone vehicles. This application may need to know the current location of the recipient as well as other crucial parts of information, such as weather conditions or energy-efficient routes. These parameters help the decision-making process of the application and are usually not available locally altogether, hence the need to access them from other providers. These data have different owners and are provided by different companies with different benefits, possibly with no established trust relation between them. As external sources are under the control of different providers, often under different security protocols (e.g., different keys and algorithms), fog applications may need to deal with different security mechanisms defined by various data providers. The number of providers dictates the way the fog application may access and consume data since the providers employ various data security protocols and mechanisms, which leads to the application needing to address all intricacies posed by the different protocols.

Table 16 classifies our reference applications according to their data providers. Future fog computing platforms may need to provide specific mechanisms for the many applications which rely on multiple independent data providers.

4.12 Privacy sensitivity

In today's societies, data is acting as fuel for most of the services that we use [87]. On the one hand, users are requested to provide data about themselves in order to receive services. On the other hand, many citizens are concerned about the usage that may be made of these data, and prefer not to reveal too much about their personal life. For example a number of mobile health applications that receive every day physical activity of their users and provide advice for healthier life styles. Although the users are interested in such advices, they do not want their everyday physical activity to be accessible

TABLE 17
Data privacy requirements.

Privacy	Applications	Total
requirements		
Public	App01, App03, App05, App09,	9
	APP13, APP14, APP17, APP20,	
	App24	
Conditional	App07, App12, App18, App23	4
Private	App02, App04, App06, App08,	11
	APP10, APP11, APP15, APP16,	
	App19, App21, App22	

to their neighbors or colleagues. The European Commission published the General Data Protection Regulation (GDPR) to address concerns about the privacy of the citizens [88].

Privacy can be defined in many different ways: legal, technical, societal, etc. In this article we define data privacy as the rules and acts that can be taken to share personal data to only the legitimate, intended recipient. Privacy, as a legal term, refers to individuals' rights to keep their information private and not be accessed by non-authorized parties.

Because fog applications are located in the immediate vicinity of end users, they often have access to private user data. Fog computing therefore inherits all the privacy issues that already existed in IoT and Cloud computing. Additionally, in Fog computing, we also deal with data which normally belong to a user, which are in close proximity to a fog node. For example, a user's location may be inferred from the location of her closest fog node, which exacerbates the problem of user privacy.

We categorize data privacy into the following three levels:

Public: the data may be accessed by anyone. For example, street names in a city are public information.

Conditional: the data may be accessed by some depending on a number of conditions. For example, the list of streets in a city with higher crime rates may be private or public depending on the city's policy.

Private: the data may be accessed only by a restricted number of entities. For example, individual health-related data is usually private.

Table 17 classifies the reference applications with respect to the privacy level of the data they manipulate. It is clear that many applications manipulate private data, which creates important issues that future fog platforms will have to deal with.

4.13 Security sensitivity

An important concern in any large-scale computing infrastructure is the security of data. As Fog computing has access to IoT data whether in the form of data sensed from devices or commands (data) sent from Cloud to the device, it is important to provide guarantees with regards to data's integrity and confidentiality. Integrity means that any sent data should be received intact at its destination. Confidentiality means that data should only be accessible to the data source and the legitimate, intended destination. As an example for the need of confidentiality, health-related data sent from an IoT device to a fog application may have high value for unauthorized organizations, and is therefore at risk for potential attacks.

TABLE 18 Data security requirements.

Data security requirements	Applications	Total
Integrity	App07, App09, App05, App10,	5
	App22	
Confidentiality	App02, App07, App08, App10,	12
	APP11, APP16, APP19, APP21,	
	App22, App25, App27, App30	

One such attack happened in 2005 to Anthem where nearly 80 millions users' health records were stolen by an attacker [89].

Existing solutions utilize encryption as means to provide data confidentiality. However, cryptography is a compute-intensive process so not all IoT devices have sufficient processing capacity on board. In Fog computing we are dealing with heterogeneous IoT devices with different cryptographic abilities, thus a fog computing platform may also need to consider the importance of confidentiality as the other important vector of security in the Fog/IoT space.

In order to avoid data breach, security mechanisms may utilize context-aware security process where the system alternates between different security levels according to its own surroundings. As an example, a node may switch to more powerful encryption schemes because nodes in proximity to the current one have been marked as infected by malware.

Table 18 categorizes the requirements of our reference applications with respect to data integrity and confidentiality. Obviously, all applications would prefer operating in a safe and secure environment. We only list here the applications where integrity or confidentiality breaches would cause major issues for their users.

4.14 Workload characteristics

Fog computing platforms are necessarily widely distributed to be located close to their end users. As such, they experience many challenges in running highly distributed and large-scale fog applications as both the fog computing infrastructure itself. To guarantee the best possible performance and high quality of service, fog computing platforms need to adapt to dynamic workloads and also be able to identify and remedy misbehaving workloads.

We can characterize the workload produced by our reference applications in two general categories and few subcategories based on the characteristics of their workloads:

Stable: the workload is almost static.

Dynamic: the workload varies according to some criterion:

- Location: the workload varies according to the location of fog node.
- Time: the workload varies as a function of time.
- User: the workload varies according to the load generated by users.

Table 19 classifies the reference applications according to their type of workload. We see that the workloads for most of the fog applications are stable. In these scenarios, sensors collect data periodically and send them to fog application for processing. For example, APP04 uses surveillance cameras to take pictures at periodic intervals and sends them to the fog

TABLE 19
Workload characteristics of fog applications.

Category	Sub category	Applications	Total
Stable		APP01, APP02, APP03, APP04, APP06, APP07, APP08, APP10, APP12, APP14, APP15, APP16, APP17, APP18, APP19, APP20, APP21, APP22, APP23, APP24, APP25, APP26, APP28	23
	Location	Арр05, Арр09, Арр11	3
Dynamic	Time	App13	1
	User	App27, App29, App30	3

for analysis. These types of applications are present in different sectors such as transportation, health, entertainment, smart cities, smart factories, smart buildings, and smart grid.

Seven out of the surveyed applications have dynamic workloads among which three are dynamic with respect to location, one is dynamic with respect to time and the rest three are dynamic with respect to users. Applications that are dynamic with respect to time and user are the same as typically found in web services running in the cloud. However, applications that are dynamic with location are specific to fog environments. When the location changes, a number of surrounding sensors may change as well as in App05. In App09 the number of self-adaptive added stations may change. In App11 the number of surrounding fog nodes with which the fog node communicates may changes. Finally, App13 collects real-time transport demand in a mobile way which makes the workload change.

We believe that the awareness of workload characteristics of fog applications helps in the design of effective and efficient management and operation of fog computing platforms. Especially for fog applications with dynamic workloads, it is necessary for the fog infrastructures and applications to be designed and deployed in a scalable manner. Meanwhile, fog management platforms need to incorporate intelligent application placement, dynamic resource allocation mechanisms and automated operation systems to ensure acceptable QoS is guaranteed.

4.15 Implementation maturity

The reference applications used in this article widely differ in the level of maturity of their implementation. We therefore classify them in four main categories:

- Conceptualization
- Simulation
- Prototyping
- Production

Table 20 classifies applications according to their implementation maturity. Interestingly, no application has already progressed to a "production" level of maturity. This shows that the fog computing technologies and applications still have a long way to go before they become mainstream. In fact, only 13 out of 30 applications have been developed at all. Three are being simulated and analyzed with data set or simulation platform, whereas 10 have actually been prototyped and tested by setting up fog platform or emulating

2018,

TABLE 20 Maturity level of fog applications.

Maturity	App.	Details	Total
	App19	simulated with dataset:	
Simulation		Luxembourg SUMO Traffic	
		scenario (LuST)	
	App29	simulated on Peersim and	
		PlanetLab	
	App30	simulated with dataset	3
	App01	small scale data center:	
Prototyping		cloudlet	
	App02	cloudlet	
	App13	Cisco Kinetic, with realistic	
		workloads	
	App14	wearable computing plat-	
		form with tracking system	
	App17	self-developed fog platform	
	App18	smart socket as the termi-	
		nal nodes, gateways as the	
		fog node	
	App25	drone as the terminal node,	
		laptop as the fog node	
	App26	host machine as the edge	
		server	
	App27	laptop as the edge server	
	App28	laptop as the edge server	10

fog node with similar devices such as cloudlet, laptop, and gateways.

5 Conclusion

Fog computing application are extremely diverse, and they logically impose a very varied types of requirements on the fog computing platforms designed to support them. Although no current general-purpose fog platform can pretend addressing all these requirements, this study aims at helping future platform designers make informed choices about the features they may or may not support, and the types of applications that may benefit from them.

Acknowledgements

This work is part of the FogGuru project which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 765452. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

References

- 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," *Journal of Systems Architecture*, 2019.
- [2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. workshop on Mobile computing*, 2012.
- [3] L. Vaquero, "Finding your way in the fog: Towards a comprehensive definition of fog computing," ACM SIGCOMM Computer Communication Review, vol. 44, no. 5, 2014.
- [4] OpenFog Consortium, "OpenFog reference architecture for fog computing," 2017, https://www.openfogconsortium.org/ra/.

- IEEE "IEEE [5] Standards Association, 1934-2018 IEEE standard for adoption ofOpenFog ref-2018. erence architecture for fog computing, https://standards.ieee.org/standard/1934-2018.html.
- [6] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Dis*tributed Systems, vol. 26, no. 4, 2014.
- [7] H. Atlam, R. Walters, and G. Wills, "Fog computing and the Internet of Things: a review," Big Data and Cognitive Computing, vol. 2, no. 2, 2018.
- [8] R. Mahmud, R. Kotagiri, and R. Buyya, Fog Computing: A Taxonomy, Survey and Future Directions. Springer, 2018.
- [9] R. K. Naha, S. K. Garg, D. Georgekopolous, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," CoRR, vol. abs/1807.00976, 2018. [Online]. Available: http://arxiv.org/abs/1807.00976
- [10] W. Hu, Z. Feng, Z. Chen, J. Harkes, P. Pillai, and M. Satyanarayanan, "Live synthesis of vehicle-sourced data over 4G LTE," in *Proc. ACM MSWIM*, 2017.
- [11] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proc. MobiSys*, 2014.
- [12] OpenFog Consortium, "Out of the fog: Use case scenarios (live video broadcasting)," 2018, http://www.fogguru.eu/tmp/OpenFog-Use-Cases.zip.
- [13] ——, "Visual security and surveillance scenario (3.2)," 2017, https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_0
- [14] —, "Transportation scenario: Smart cars and traffic control (3.1)," 2017, https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_0
- [15] —, "Out of the fog: Use case scenarios (high-scale drone package delivery)," 2018, http://www.fogguru.eu/tmp/OpenFog-Use-Cases.zip.
- [16] —, "Process manufacturing âĂŞ beverage industry," 2018, http://www.fogguru.eu/tmp/OpenFog-Use-Cases.zip.
- [17] —, "Smart cities scenario (3.3)," 2017, https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_0
- https://www.iiconsortium.org/pdf/OpenFog_Reference_Archi [18] ——, "Real-time subsurface imaging," 2018,
- http://www.fogguru.eu/tmp/OpenFog-Use-Cases.zip.
 [19] ——, "Patient monitoring,"
- http://www.fogguru.eu/tmp/OpenFog-Use-Cases.zip.
- [20] —, "Autonomous driving," 2018, http://www.fogguru.eu/tmp/OpenFog-Use-Cases.zip.
- [21] S. Dey and A. Mukherjee, "Robotic SLAM: A review from fog computing and mobile edge computing perspective," in *Proc.* MOBIQUITOUS, 2016.
- [22] Cisco, "Enabling MaaS through a distributed IoT data fabric, fog computing and network protocols," White paper, 2018, https://alln-extcloud-storage.cisco.com/ciscoblogs/5c0a6ea91edbb.pdf.
- [23] B. Thomas, B. Close, J. Donoghue, J. Squires, P. D. Bondi, M. Morris, and W. Piekarski, "ARQuake: An outdoor/indoor augmented reality first person application," in *Proc. Interna*tional Symposium on Wearable Computers, 2000.
- [24] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," *Journal of Network and Computer Applications*, vol. 98, Nov. 2017.
- [25] S. Kyriazakos, M. Mihaylov, B. Anggorojati, A. Mihovska, R. Craciunescu, O. Fratu, and R. Prasad, "eWALL: an intelligent caring home environment offering personalized context-aware applications based on advanced sensing," Wireless Personal Communications, vol. 87, no. 3, 2016.
- [26] G. Jia et al., "SSL: Smart street lamp based on fog computing for smarter cities," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, 2018.
- [27] Y.-D. Chen, M. Z. Azhari, and J.-S. Leu, "Design and implementation of a power consumption management system for smart home over fog-cloud computing," in *Proc. International Conference on Intelligent Green Building and Smart Grid*, 2018.
- [28] C. Zhu et al., "Vehicular fog computing for video crowdsourcing: Applications, feasibility, and challenges," in *IEEE Communications Magazine*, vol. 56, no. 10, Oct. 2018.
- [29] R. Rajesh and V. Shijimol, "Vehicular pollution monitoring and controlling using fog computing and clustering algorithm," in International Journal of New Innovations in Engineering and Technology, Mar. 2016.

- [30] R. K. Barik, R. Priyadarshini, H. Dubey, V. Kumar, and K. Mankodiya, "FogLearn: Leveraging fog-based machine learning for smart system big data analytics," International Journal of Fog Computing, vol. 1, no. 1, 2018.
- [31] H. A. A. Hamid, S. M. M. Rahman, M. S. Hossain, hmad Almogren, and A. Alamri, "A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography," IEEE Access, vol. 5, 2017.
- [32] SWAMP Project, "Smart water management platform," 2018,
- [33] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by Internet of Things," Transactions on Emerging Telecommunications Technologies, 2013, https://doi.org/10.1002/ett.2704.
- [34] N. Chen, Y. Chen, Y. You, H. Ling, P. Liang, and R. Zimmermann, "Dynamic urban surveillance video stream processing using fog computing," in Proc. BigMM, 2016.
- [35] D. Deyannis, R. Tsirbas, G. Vasiliadis, R. Montella, S. Kosta, and S. Ioannidis, "Enabling GPU-assisted antivirus protection on android devices through edge offloading," in Proc. EdgeSys,
- [36] U. Drolia, K. Guo, J. Tan, R. Gandhi, and P. Narasimhan, "Cachier: Edge-caching for recognition applications," in Proc. ICDCS, 2017.
- [37] P. Hao, Y. Bai, X. Zhang, and Y. Zhang, "Edgecourier: An edgehosted personal service for low-bandwidth document synchronization in mobile cloud storage services," in Proc. ACM/IEEE SEC, 2017.
- [38] Y. Lin and H. Shen, "CloudFog: Leveraging fog to extend cloud gaming for thin-client MMOG with high quality of service,' IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 2, 2017.
- [39] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu, "Understanding performance of edge content caching for mobile video streaming," IEEE Journal on Selected Areas in Communications, vol. 35, no. 5, 2017.
- [40] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the internet of things," in Proc. workshop on Mobile cloud computing, 2013.
- [41] M. Abrash, "Latency âĂŞ the sine qua non of AR and VR," 2012, http://blogs.valvesoftware.com/abrash/latency-the-sine-qua-non-of-ar-and-vW. Abdelrahman, A. B. A. Mustafa, and A. A. Osman, "A
- [42] OculusRift, "Delivers some home truths on latency," $https://oculusrift-blog.com/john-carmacks-message-of-latency/68 \stackrel{[69]}{}$
- CLAudit Project, "Planetary-scale cloud latency auditing platform," 2016, http://bit.do/bS4js.
- [44] Cisco Inc., "Internet of things (iot) data continues to explode exponentially. who is using that data and how?" 2019,
- [45] IHS Inc, "Big, big, big data: The rise of HD video surveillance cameras spurs information explosion," 2013.
- [46] M. R. Shahid, G. Blanc, Z. Zhang, and H. Debar, "IoT devices recognition through network traffic analysis," in Proc. Biq Data, Dec. 2018.
- J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing," in *Proc. VTC*,
- [48] S. Basu, A. H. Karp, J. Li, J. Pruyne, J. Rolia, S. Singhal, J. Suermondt, and R. Swaminathan, "Fusion: managing healthcare records at cloud scale," Computer, vol. 45, no. 11, 2012.
- [49] Trend Micro. "Healthcare under attack: What happens to stolen medical records?" https://www.trendmicro.com/vinfo/pl/security/news/cyber-attack/36hel.hltRcaceson.dar-attacksstolen-KaathicaNredoKisown, G. Parulkar,
- [50] S. J. Stolfo, M. B. Salem, and A. D. Keromytis, "Fog computing: Mitigating insider data theft attacks in the cloud," in Proc. IEEE symposium on security and privacy workshops, 2012.
- [51] A. R. Biswas and R. Giaffreda, "IoT and cloud convergence: Opportunities and challenges," in Proc. IEEE World Forum on Internet of Things, 2014.
- [52] M. Taneja and A. Davy, "Resource aware placement of IoT application modules in fog-cloud computing paradigm," in Proc. Symposium on Integrated Network and Service Management, 2017.
- "Patient monitoring," [53] OpenFog Consortium. 2019. https://www.openfogconsortium.org/wp-content/uploads/Patient-Monttorsing-Ish Entel patie Raspberry Pi cloud cluster experiment," in

- [54] R. Arshad, S. Zahoor, M. A. Shah, A. Wahid, and H. Yu, "Green iot: An investigation on energy saving practices for 2020 and beyond," IEEE Access, vol. 5, 2017.
- [55] F. Jalali, A. Vishwanath, J. De Hoog, and F. Suits, "Interconnecting fog computing and microgrids for greening IoT," in Proc. ISGT-Asia, 2016.
- [56] R. Buyya, M. Pathan, and A. Vakali, Content delivery networks. Springer, 2008, vol. 9.
- [57] A.-M. K. Pathan and R. Buyya, "A taxonomy and survey of content delivery networks," Grid Computing and Distributed $swamp-project.org/communication/WRNP2018_lamina_SWAMP_ENSpections. Laboratory,\ University\ of\ Melbourne,\ Technical\ Report,$
 - [58] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, "A survey on content-oriented networking for efficient content delivery," IEEE Communications Magazine, vol. 49, no. 3, 2011.
 - [59] C. Papagianni, A. Leivadeas, and S. Papavassiliou, "A cloudoriented content delivery network paradigm: Modeling and assessment," IEEE Transactions on Dependable and Secure Computing, vol. 10, no. 5, 2013.
 - X. Wang, S. Leng, and K. Yang, "Social-aware edge caching in fog radio access networks," IEEE Access, vol. 5, 2017.
 - IEEE Standards Association, "IEEE standard for adoption of OpenFog reference architecture for fog computing," 2018, https://standards.ieee.org/standard/1934-2018.html.
 - [62] G. Margelis, R. Piechocki, D. Kaleshi, and P. Thomas, "Low throughput networks for the IoT: Lessons learned from industrial implementations," in Proc. WF-IoT, 2015.
 - [63] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Longrange communications in unlicensed bands: The rising stars in the IoT and smart city scenarios," IEEE Wireless Communications, vol. 23, no. 5, 2016.
 - [64] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," ICT Express, vol. 3, no. 1, 2017.
 - G. S. Nitesh and A. Kakkar, "Generations of mobile communication," International Journal of Advanced Research in Computer Science and Software Engineering, 2016.
 - 3GPP, "Early progress on rel-16 bands for 5G," 2019, https://www.3gpp.org/news-events/2025-early-progress-on-rel-16-bands-for
 - $\mathbf{E}.$ Kauppalehti, ${\rm ``Real"}$ world 4GLTEbandwidth)," 5Gtest benchmark: 14xhttps://react-etc.net/entry/real-world-4g-lte-vs-5g-test-benchmark-14x-bar
 - comparison between IEEE 802.11 n and ac standards," 2015. N. Ahmed, H. Rahman, and M. I. Hussain, "A comparison of 802.11 ah and 802.15. 4 for IoT," ICT Express, vol. 2, no. 3,
 - Cisco, "IEEE 802.11ax the sixth generation of Wi-Fi," 2018, https://www.cisco.com/c/dam/en/us/products/collateral/wireless/white-pa
- https://blogs.cisco.com/datacenter/internet-of-things-iot-data-continues-to-expressed expression expressi of things (iot) communication protocols," in *Proc. ICIT*, 2017. [72] E. Marín-Tordera, X. Masip-Bruin, J. García-Almiñana,
- https://news.ihsmarkit.com/press-release/design-supply-chain-media/bAg-blykng-Garb-Rev.htm/dicoZhurviAnance-ell-meally-know-mhat a fog node is? current trends towards an open definition," Computer Communications, vol. 109, 2017.
 - S. S. Adhatarao, M. Arumaithurai, and X. Fu, "FOGG: A fog computing based gateway to integrate sensor networks to internet," in Proc. ITC, vol. 2, 2017.
 - M. Iorga, L. Feldman, R. Barton, M. Martin, N. Goren, and C. Mahmoudi, "Fog computing conceptual model, recommendations of the national institute of standards and technology," NIST Special Publication, 2018.
 - A. Ahmed and G. Pierre, "Docker container deployment in fog computing infrastructures," in $Proc.\ IEEE\ EDGE,\ 2018.$
 - J. Rexford, M. Satyanarayanan, O. Sunay, and A. Vahdat, "Democratizing the network edge," SIGCOMM Comput. Commun. Rev., vol. 49, no. 2, May 2019.
 - [77] A. Fahs and G. Pierre, "Proximity-aware traffic routing in distributed fog computing platforms," in Proc. IEEE/ACM CC-Grid, 2019.
 - Wikipedia, "Single board computer," 2019.https://en.wikipedia.org/wiki/Single-board_computer.
 - P. Abrahamsson, S. Helmer, N. Phaphoom, L. Nicolodi, N. Preda, L. Miori, M. Angriman, J. Rikkilä, X. Wang, K. Hamily et al., "Affordable and energy-efficient cloud computing clus-

- Proc. Intl. Conference on Cloud Computing Technology and Science, 2013.
- [80] P. Carbone, G. E. Gévay, G. Hermann, A. Katsifodimos, J. Soto, V. Markl, and S. Haridi, Large-Scale Data Stream Processing Systems. Springer, 2017.
- [81] H. Lee, J. Oh, K. Kim, and H. Yeon, "A data streaming performance evaluation using resource constrained edge device," in *Proc. ICTC*, 2017.
- [82] Q.-C. To, J. Soto, and V. Markl, "A survey of state management in big data processing systems," The VLDB Journal, vol. 27, no. 6, Dec. 2018.
- [83] E. Curry, "Message-oriented middleware," Middleware for communications, 2004.
- [84] H. Gupta, S. B. Nath, S. Chakraborty, and S. K. Ghosh, "SDFog: a software defined computing architecture for QoS aware service orchestration over edge devices," CoRR, vol. abs/1609.01190, 2016. [Online]. Available: http://arxiv.org/abs/1609.01190
- [85] Ishwarappa and J. Anuradha, "A brief introduction on big data 5Vs characteristics and Hadoop technology," Procedia Computer Science, vol. 48, 2015.
- [86] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International Journal of Information Management*, vol. 35, no. 2, 2015.
- [87] Economist, "Data is giving rise to a new economy," 2017, https://www.economist.com/briefing/2017/05/06/data-is-giving-rise-to-a-new-economy.
- [88] Council of European Union, "Council regulation (EU) no 269/2014," 2014,
 - https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=OJ:L:2016:119:FULL&from=EL.
- [89] Cloud Security Alliance, "The treacherous 12 top threats to cloud computing + industry insights," 2017, https://downloads.cloudsecurityalliance.org/assets/research/top-threats/treacherous-12-top-threats.pdf.