JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# FLEXUOUS BASED EBULLIENT AND EXUBERANT EDGE COMPUTING FOR FUTURISTIC SCENARIO IN THE VIEW OF VISIONS AND CHALLENGES

**[1]Nazeer Shaik, [2]Dr.B. Harichandana, [3]Dr.P. Chitralingappa**

[1,2,3] Srinivasa Ramanujan Institute of Technology, Rotarypuram, B.K. Samudram, Anantapur

**ABSTRACT**

Edge computing, a new computing paradigm that involves processing data at the edge of the network, is emerging as a result of the proliferation of the Internet of Things (IoT) and the success of rich cloud services. Using edge computing, we can address a number of concerns, including response time requirements, battery life constraints, bandwidth cost savings, and data privacy and safety concerns. Our paper introduces edge computing and provides several case studies, such as cloud offloading, smart homes, smart cities, and collaborative edges, to illustrate its concept. In the end, we discuss several challenges and opportunities related to edge computing, and we hope that this paper will be of interest to the community and inspire more research to take place.

## I. INTRODUCTION

We have witnessed a tremendous change in the way we live, work, and study since the advent of cloud computing in 2005 [1]. For example, software as a service instance (SaaS), such as Google Apps, Twitter, Facebook, and Flickr, have become widely used. A scalable infrastructure is also essential. Besides Google File System [2], MapReduce [3], Apache Hadoop [4], and Apache Spark [5], there are other processing engines developed for cloud services that influence the way businesses run. Supply chain management was the first application of the Internet of Things (IoT) in 1999, followed by the concept of "making computers sense information without a network" in 2003. Human intervention was widely applied in many other fields, including healthcare, the home, the environment, and transportation [7, 8]. With IoT, we will enter the post-cloud era, in which a lot of data is generated by things that are immersed in our daily lives, and a lot of Data will be consumed at the edge by applications. According to Cisco's Global Cloud Index, the global data center IP traffic will reach only 12.4 zettabytes by 2023, despite people, machines, and things producing 500 zettabytes of data by then [9]. In 2023, 55% of IoT data will be stored, processed, analyzed, and acted upon near the edge of the network [10]. According to Cisco Internet Business Solutions Group [11], 60 billion things will be connected to the Internet by 2025. Some IoT applications might require quick responses, while others might not.

Using cloud services and IoT, we envision the edge of a network changing from a data consumer to a data producer. We attempt to contribute to the concept of edge computing in this paper. We begin by analyzing the reason we need edge computing, and then give our definition and vision of edge computing. In order to further explain edge computing in a detailed manner, several case studies are presented, including offloading data from the cloud, smart homes, and cities, and collaborative edges. There are several challenges and opportunities for future research and study concerning programmability, naming, data abstraction, service management, privacy and security, and optimization metrics. This paper is divided into the following parts. The second part discusses edge computing and gives a definition of edge computing. Section III shows some edge computing case studies. Section IV discusses the challenges and opportunities that may arise. In Section V, we conclude this paper.

## II. WHAT IS EDGE COMPUTING

Since data is increasingly produced at the edge of the network, it would be more efficient to process the data there as well. Cloud computing has not always been efficient for processing data produced at the edge of the network because micro data centers [12], [13], cloudlets [14], and fog computing [15] have been introduced to the community. For some computing services, edge computing is more efficient than cloud computing. We then define and describe edge computing in this section.

**A. Why Do We Need Edge ComputingPush from Cloud Services:**

Since the computing power on the cloud outperforms the capability of the things at the edge, it has proved a very efficient way to process data. The network bandwidth, however, has stagnated in comparison with the fast-developing speed of data processing. As the amount of data generated at the edge increases, the speed at which it is transported is becoming a bottleneck for cloud-based computing. A Boeing 787, for instance, will generate about five gigabytes of data every second, but there is not enough bandwidth for data transmission between it and the satellite or ground base station [16]. Another example is an autonomous vehicle. Gigabytes of data will be generated by the car every second and the decision-making process must be done in real-time in order for the vehicle to be successful [17]. It would take too long for the data to be processed if all of the data had to be sent to the cloud. The current network's bandwidth and reliability would be challenged because of how many vehicles could be supported at the same time. For a shorter response time, more efficient processing, and a smaller network load, it is necessary to process the data at the edge.

**Pull from IoT:**

All kinds of electrical devices will become part of the Internet of Things, and they will both produce data as well as consume it, including air quality sensors, LED bars, streetlights, and microwave ovens that can be connected to the Internet. In a few years, there will be more than billions of things at the edge of the network, producing enormous amounts of raw data. Consequently, most of the data produced by IoT will never be transmitted to the cloud, but rather consumed at the edge.

The conventional cloud computing model is shown in **Fig. 1.** Data producers generate raw data and transfer it to the cloud, and data consumers consume the data from the cloud. As indicated by the blue solid line. The red dotted line represents the request for consuming data sent by consumers to the cloud, and the green dotted line represents the result from the cloud. IoT, however, does not require this structure. As a result of large amounts of data at the edge, unnecessary bandwidth and computing resources will be consumed. Furthermore, the privacy protection requirement will hamper cloud computing in IoT. Finally, most IoT end nodes are usually energy-constrained, and wireless communication modules are generally very energy-hungry, so offloading some computing tasks to the edge could save energy.


**Fig 1: An Instance of Cloud ComputingChange From Data Consumer to Producer:**

As a result of cloud computing, end devices at the edge are usually data consumers, such as watching a YouTube video on a smartphone. Nowadays, people are also producing data using their mobile devices. The change from a data changing from being a consumer to being a data producer/consumer requires more edge function placement. It is common for people today to take pictures or make video recordings and then share them on social media sites like YouTube, Facebook, Twitter, and Instagram by uploading them to a cloud service. In addition, YouTube users upload 72 hours of video content every single minute; Facebook users share nearly 2.5 million pieces of content; Twitter users tweet nearly 300 000 times; Instagram users post nearly 220 000 new photos every single minute [18]. It could, however, take a lot of bandwidth to upload a large image or video clip. In this case, the video clip should be demised and adjusted to a suitable resolution at the edge before uploading to the cloud. Wearable health devices would be another example. Since the physical data collected by the things at the edge of the network is usually private, processing the data at the edge could protect user privacybetter than uploading raw data to the cloud.

**B. What is EDGE Computing**

In edge computing, computation can be performed at the edge of a network, on downstream data from the cloud or on upstream data from the internet of things. In this context, we define “edge” as any computing and network resource along the path between data and the cloud. Smartphones are the edge between body things and the cloud, gateways in smart homes are the edge between home things and the cloud, and micro-data centers are the edge between data sources and cloud data centers. Edge computing is the concept that computing should occur near data sources. The cloudlet [14] is the boundary between a mobile device and cloud. Compared to fog computing [19], edge computing focuses more on the infrastructure side of the thing, while fog computing focuses more on the other side of the thing. Our vision is that edge computing could have the same impact as cloud computing on our society.

In edge computing, things are not just data consumers but also data producers. **In Fig 2,** we see the two-way computing streams. Things can not only request content and services from the cloud at the edge, but they can also perform computing tasks from the cloud there. In addition to computing offloading, data storage, caching, and processing, edge can distribute requests and deliver services from the cloud to the user. To meet the requirements efficiently in service such as reliability, security, and scalability, the edge itself must be well-designed. The protection of privacy is also important.

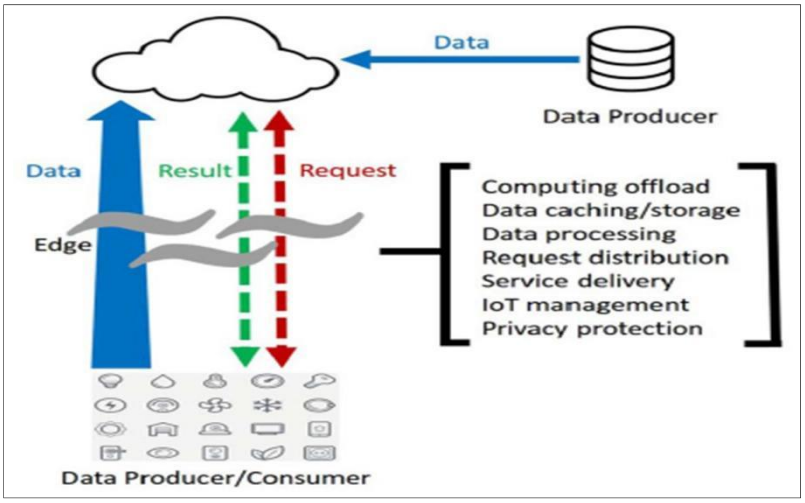


**Fig.2: An Instance of Edge computing**

### C. Edge Computing Benefits

In edge computing, we want to put the computing in proximity to data sources. This has several benefits compared to the traditional cloud-based computing paradigm. We illustrate the potential benefits here using early results from the community. Researchers created a proof-of-concept. A cloud-based platform for running face recognition applications is used in [20], and the computation is moved to the edge, resulting in a reduction of response time from 900 to 169 milliseconds. By offloading computing tasks for wearable cognitive assistance with cloudlets, Ha et al. [21] showed an 80-200ms improvement in response time. Additionally, cloudlet offloading can reduce energy consumption by 30% to 40%. The clone cloud in [22] combines partitioning, migration, and On-demand partitioning between the cloud and mobile and their prototype can reduce 20× running time and energy for tested applications.

## III. CASE STUDY

Our vision for edge computing is further illustrated in this section with several case studies.

### *A.* Cloud Offloading

Most computations happen in the cloud with cloud computing, which means data and requests are processed in a centralized cloud. As a result of such a computing paradigm, users may suffer from longer latency (e.g., long tail latency), which weakens their experience. Cloud offloading has been studied in a mobile-cloud environment in terms of energy-performance trade-offs [22]–[26]. Using edge computing, you can offload part of the workload from the cloud because the edge has certain computationresources.

At the edge servers of the traditional content delivery network, only data is cached. This is a result of the fact that the content provider provides the data over the Internet, which has been the case for decades. In the Internet of Things, the data is produced and consumed at the edge. The edge computing approach should cache both data and operations applied to the data.

An example of an application that could benefit from edge computing is online shopping services. Customers manipulate their shopping carts frequently. By default, all changes to their shopping carts are done in the cloud, and then the new shopping cart view is updated on their desktops. The process may take a long time depending on the network speed and the load on the servers. Due to the relatively low bandwidth of the mobile network, it can be even longer for mobile devices. With the increasing popularity of mobile shopping, it is crucial to make improvements. The user experience, especially latency-related issues. In such a scenario, offloading shopping cart updates from cloud servers to edge nodes will greatly reduce latency. Users' shopping cart data and related operations (e.g., adding an item, updating an item, deleting an item) are both cached at the edge node, as we mentioned. Once the user's request reaches the edge node, a new shopping cart view can be generated immediately. The cloud should synchronize the data at the edge node, but this can be done in the background as well.

- There is the possibility for navigation applications to move the navigation or search services to the edges of local areas, in which case only a few map blocks are used.
- Data volume could be reduced by filtering/aggregating content at the edge nodes.
- Edge nodes are useful for real-time applications such as vision-aided entertainment games,augmented reality, and connected health.

With edge computing, time-sensitive applications can benefit from reduced latency and improved user experience.

### *B.* Video Analytics

As mobile phones and network cameras become more widespread, video analytics becomes an increasingly important technology. Because of the long data transmission latency and privacy concerns, cloud computing is no longer suitable for video analytics applications. Here is an example of finding a lost child with cloud computing. When a child goes missing, it is very likely that a

camera will be able to capture him/her. Different kinds of cameras are widely deployed in urban areas and in each vehicle nowadays. In most cases, the camera data will not be uploaded to the cloud due to privacy concerns or traffic issues. Even if the data was accessible on the cloud, uploading and searching such a vast quantity of data would be very time-consuming. Searching for a missing child can take a long time, which is unacceptable. In the edge computing paradigm, a request to locate a child can be generated in the cloud and sent to all edge devices at the same time. There are things in a target area that can perform the request and search their local camera data, and then report the results back to the cloud. This paradigm allows for the exploitation of data and computing power across everything and results in much faster results than with a solitary cloud computing model.

### *C.* Smart Phone

Some IoT products are already available on the market, such as smart lights, smart TVs, androbot vacuums. IoT would greatly benefit the home environment. For a smart home, adding a Wi-Fi module to the current electrical device and connecting it to the cloud isn't enough. In a smart home environment, besides the connected device, a heap of wireless sensors and controllers should be deployedto a room, pipe, and even floor and wall. For reasons related to data transportation pressure and privacyprotection, these devices should report an impressive amount of data. This feature makes the cloud computing paradigm unsuitable for building a smart home. Edge computing, however, is considered perfect for building and in a smart home, things can be connected and managed easily in the home withan edge gateway running an edge operating system (EdgeOS), and data can be processed locally to reduce the demand for Internet bandwidth, as well as the service can be deployed using the EdgeOS. The fourth section-IV discusses more opportunities and potential challenges for better management and delivery.

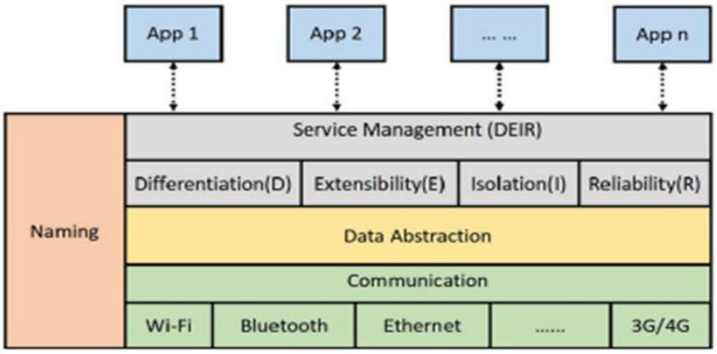


**Fig 3: The Schematic Representation of Structure of EdgeOS in the Smart Home Environment.**

This **Fig.3** shows a smart home version of EdgeOS. EdgeOS collects data from mobile devices and a variety of other devices through a variety of communication methods. The data abstraction layer fuses and massages data from different sources using methods such as Wi-Fi, Bluetooth, ZigBee, or a cellular network. An in-depth description of the data abstraction layer will be provided in Section IV-C. Service management is on top of the data abstraction layer. This layer supports differentiation, extensibility, isolation, and reliability requirements. We will address this issue further in Section IV-D. The naming mechanism is required for all layers with different requirements, so we leave it in a cross- layer fashion. Section IV-B discusses naming challenges.

### *D.* Smart Cities

It is possible to expand edge computing from a single home to a community, or even to a city scale. Edge computing advocates computing being done as close to the data source as possible. This design allows for the generation of a request at the top of the paradigm and considering the following characteristics, edge computing may be an ideal platform for the smart city.

*1)* **Large Data Quantity**: 'Approximately 180 PB of data will be produced per day by a city of one million people by the year 2019 [9], contributing to the overall workload and as a result of the heavy traffic workload, it is unrealistic to build centralized cloud data centers to manage all of the data, such as public safety, health, utilities, and transportation. By processing the data at the network edge, edge computing can be an effective solution in this case. As a result, health emergencies and public safety applications require predictable and low latency.

*2)* **Low Latency**: Applications requiring predictable and low latency, such as health emergencies and public safety, should use this protocol. As shown **in Fig. 4,** connected health is an example of a collaborative edge. Edge computing could also reduce the time it takes to transfer data and simplify network design. From the edge of the network, diagnosis and decisions can be made, which is more efficient. As opposed to centralized cloud-based data collection and decision-making.


**Fig 4: The Schematic Representation of an instance of Collaborative Edge: Connected Health.**

***3)*** **Location Awareness:** When it comes to transportation and utility management, edge computing is a better choice than cloud computing. The advantage of edge computing is its location awareness. It is possible to collect and process data based on geographic location without transferring it to the cloud with computing.

### *E.* Collaborative Edge:

In academia and industry, collaborative edge clouds are becoming the de facto platform for big data processing. A key promise of cloud computing is that the data should be accessible anywhere, at any time. In some cases, the data owned by stakeholders may already be stored in the cloud or will be transmitted there and eventually processed there. As a result, it is rare for stakeholders to share their data due to privacy concerns and the tremendous cost of transporting data. As a physical small data center that connects the cloud and end-user with data processing capabilities, edge can also be a part of the logical concept. Edge of collaboration, an approach that connects the edges of geographically distributed stakeholders is proposed [15]. Edges provide stakeholders with an opportunity to collaborate and share data.

As shown **in Fig 4,** connected health is one of the promising applications of the near future. The demand for geographically distributed data processing applications, such as healthcare, requires data sharing and collaboration between companies in multiple fields. By creating virtual shared data views, the collaborative edge is able to fuse geographically distributed data. The virtual shared data is exposed to users through a predefined service interface. By leveraging this public interface, you can create complex services for end users. These services are provided by participants of the collaborative edge, and the computation occurs only in the collaborative edge. Data privacy and integrity can be assured in the participant's data facility. In order to demonstrate the potential benefits of collaborative edge, a flu outbreak is used as the basis for our case study. As patients flow to hospitals, their electronic medical records (EMRs) are updated. This outbreak of the flu is summarized and shared by the hospital, including the average cost, symptoms, population, etc. In theory, patients follow prescriptions to get their prescriptions from pharmacies. There is a possibility that they do not follow the therapy. In that case, the hospital is responsible for rehospitalization.

Now, through collaborative edge, the pharmacy can provide the hospital with a copy of the patient's purchasing record, facilitating healthcare accountability significantly. Additionally, pharmacies are able to obtain the population of the flu outbreak using hospitals' collaborative edge services. The pharmacy is able to utilize data provided by hospitals to purchase drugs. Behind the scenes, the pharmacy can leverage data provided by hospitals. All drug warehouse locations, prices, and inventories are retrieved by pharmaceutical companies, and a transport price query request is sent to logistics companies by the pharmaceutical companies. As a result of the information retrieved, the pharmacy can make an order plan based on the total cost optimization problem. At this point, pharmaceutical companies can reschedule their production plan and rebalance the warehouse inventories if they receive a bunch of flu drug orders from pharmacies. In the meantime, the centers for disease control and prevention, as our government representative in our case, are monitoring the flu epidemic in a wide range of areas, and canraise a flu alert as a result.

Moreover, further actions can be taken to prevent the spread of flu outbreaks. Following an outbreak of the flu, insurance companies are required to pay for the patients' bills based on their policies. In order to adjust the policy price for the next year, insurance companies are able to analyze the proportion of people with the flu during the outbreak and the cost of the flu treatment. A patient can also obtain a personalized healthcare policy based on their EMR if they wish. The collaborative edge can reduce operational costs and increase profitability for most of the participants in this case. However, some of them, such as the workers, won't be able to use it to their advantage. Since hospitals are the major information collectors within the healthcare community, they could be considered pure contributors.

## IV. CHALLENGES AND OPPORTUNITIES

To realize the vision of edge computing, the systems and network communities must work together. We have described five potential applications of edge computing. In this section, we will further summarize these challenges in detail and present some potential solutions and opportunities worth further research, including programmability, naming, data abstraction, service management, privacy and security, and optimization metrics.

### *A.* Programmability

In cloud computing, users program their code and deploy it on the cloud. It is the cloud provider's responsibility to decide where computing will take place. Users do not have any knowledge of the way applications work. The infrastructure of cloud computing is one of the advantages of cloud computing. Usually, the program is written in one programming language and is compiled for a specific platform, since it only runs in the cloud. In edge computing, however, computing is offloaded from the cloud, and edge nodes are typically heterogeneous platforms. Since these nodes have different runtimes, it is extremely difficult for a programmer to write an application that can be deployed in edge computing. Stream computing refers to a series of functions/computing applied to the data as it propagates along a data propagation path in order to address the programmability of edge computing. As long as the application specifies where computing should occur, functions/computing can occur anywhere on the path, whether they are entire or partial functionalities of the application.

By using the computing stream, data can be processed on data-generating devices, edge nodes, and the cloud in an efficient and distributed manner. Using edge computing, a lot of computing is done at the edge instead of the main CPU. Cloud computing is a centric cloud. In this case, the computing stream helps the user determine what functions should be performed and how the data should be propagated. The function/computing distribution metric could depend on latency, energy cost, TCO, and hardware/software limitations. Section IV-F provides a detailed cost model. Data will be computed as close to the source as possible by deploying a computing stream. It is possible to reduce the cost of data transmission by reallocating the function in the computing stream, and reallocating the data and state along with it as well. Furthermore, collaboration issues (e.g., synchronization, data/state migration, etc.) need to be addressed across all platforms. The edge computing paradigm has multiple layers.

### *B. Naming*

The number of things in edge computing is huge, and this assumption is important in edge computing. It is common for edge nodes to run a number of applications, and each application has its own structure for delivering the services. In the same way as in any computer system, the naming scheme in edge nodes is similar. The edge computing paradigm is very valuable for programming, addressing, identifying things, and communicating data. In order to communicate with heterogeneous edge devices, practitioners must learn a variety of communication protocols and network protocols. The naming scheme for edge computing must address mobility, highly dynamic network topology, privacy and security protection, as well as scalability.

Currently, most networks are satisfied with conventional naming mechanisms, such as DNS and uniform resource identifiers. Due to the fact that most of the things at the edge can be highly mobile and resource constrained, they are not flexible enough to serve the dynamic edge network. Additionally, IP- based naming schemes may not be feasible for resource-constrained resources at the edge of the networkdue to their complexity and overhead.

Edge computing can also benefit from new naming mechanisms such as named data networking (NDN) [27] and Mobility First [28]. A NDN is a hierarchical structure for a content/data-centric network, and it provides good scalability and is human-friendly for service management. A proxy would be needed to fit into other communication protocols such as BlueTooth or ZigBee, and so forth. A secondary issue with NDN is security, which is difficult to isolate with service providers since hardware information is very difficult to isolate. The name can be separated from the service provider by MobileFirst. For edge services where mobility is high, a global unique identifier would provide better mobility support. (GUID) needs to be used for naming in MobileFirst, but this isn't required in fixed information aggregation services at the edge of the network, such as the home environment. MobileFirst for edge has the disadvantage that it is difficult to manage services because GUID is not user-friendly.

It may be possible to assign network addresses to each thing within a relatively small and fixed edge environment such as a home environment. It would be possible for each thing within one system to have a unique human-friendly name that describes the following information: location (where), role (who), and description (what), like “kitchen. oven2.temperature3.” The EdgeOS will then assign this thing an identifier and network address, as shown **in Fig. 5**. Its human-friendly name is unique. Service management, component replacement, and diagnostics can all be performed with this naming mechanism, which is very convenient for users as well as service providers. EdgeOS will, for example, notify the user when bulb 3 of the ceiling light (who) in the living room (where) failed, and then the user can easily replace the bulb without searching for an error code or reconfiguring the network address. In addition, this naming mechanism provides better programmability to service providers, while at the same time blocking them from accessing hardware information, improving the protection of data privacy and security. In EdgeOS, the identifier will be used to manage things. It is possible to map the human-friendly name to unique identifiers and network addresses. Various communication protocols, such as Bluetooth, ZigBee, and WiFi, will be supported with network addresses, such as IP addresses or MAC addresses. In highly dynamic environments, like city-level systems, it is still an open problem that needs further investigation by the community.

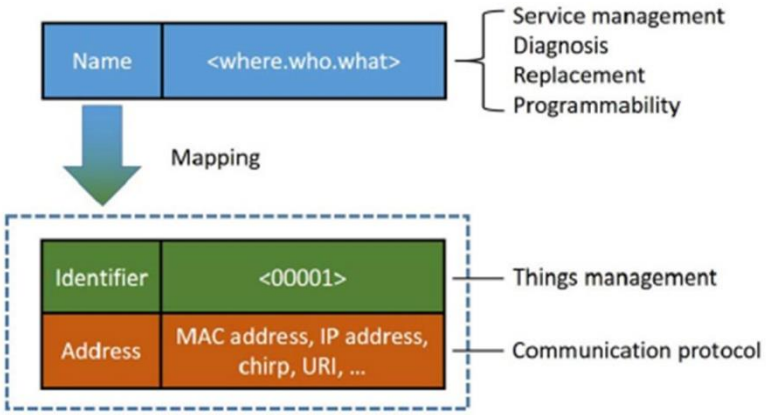


**Fig 5: The Schematic Representation of Naming Mechanism of in EdgeOS**

### *C.* Data Abstraction

Through the air position indicators provided by the service management layer, EdgeOS can run various applications that consume data or provide services. Wireless sensor networks and cloud computing paradigms have both discussed and researched data abstraction extensively. With IoT, there will be a large number of data generators in the network, which makes this issue more challenging. As an example, we take the smart home environment. Nearly every item in a smart home will report data to EdgeOS, not to mention the large number. A wide range of devices are deployed throughout the household. However, most of these devices only send their sensed data periodically to the gateway. For instance, a thermometer can send its temperature periodically to the gateway. Real users will likely only consume this data a few times a day, even though the temperature is monitored every minute. As an example, a surveillance camera can keep recording and sending video to a gateway, but its data will just be stored in the database for a certain time with and people don't actually consume it, and it's flushed out by new videos. We envision that the edge node should consume and process all the data in a proactive manner and minimize human involvement in edge computing. Data should be processed at the gateway level to remove noise/low quality, detect events, protect privacy, and so on. The processed data will go to the upper layer to be used for future services. This process will present several challenges.

There are various formats in which data is reported from different sources, as shown in **Figure 6**.A gateway application that runs on the gateway should be blinded from raw data for privacy and security reasons. Additionally, the knowledge they want to extract from the integrated data table can be defined easily with an ID, a time, a name, and a data field (for example, [0000, 12:34:56PM 01/01/2016, Kitchen.oven2. temperature3, 78] so that any edge thing's data can be included. However, the details ofsensed data have been hidden, which may affect their usability.

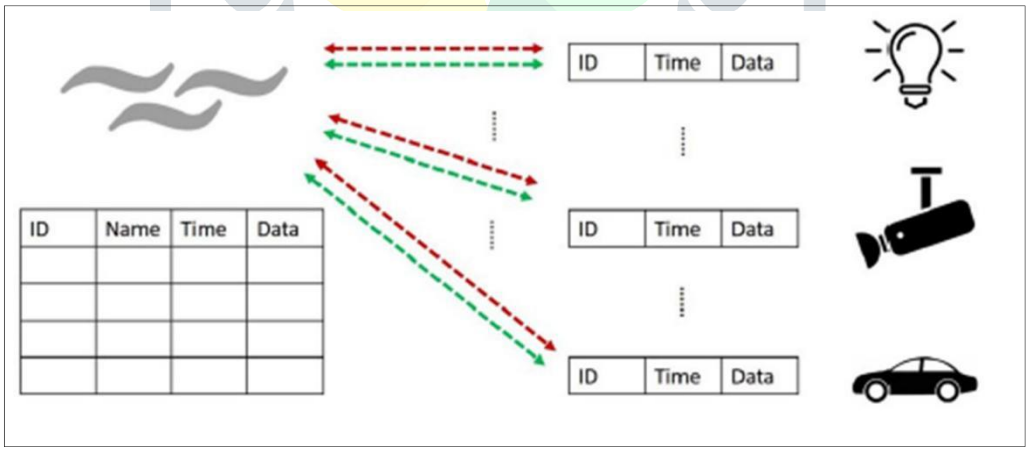


**Fig.6: In the Context of Edge Computing, there is An Issue of Data Abstraction**

Secondly, it can be difficult to decide how much data to abstract. If too much raw data is filtered out, some applications or services could not learn enough knowledge. However, if we want to keep a large number of raw data, data storage would be a challenge. Finally, data reported. Due to low precision sensors, hazardous environments, and unreliable wireless connections, things at the edge can be unreliable sometimes. IoT developers and applications still face the challenge of abstracting useful information from unreliable data sources in this case.

There is another issue with data abstraction - the operations that can be performed on the things. Data collection serves the application, and the application should be able to control the things (e.g., read and write to them) to complete certain services the user desires. Combining the data representation and in addition, since EdgeOS is heterogeneous, its data abstraction layer will serve as the public interface for all things connected to it. The allowed operations could diverge greatly, which also increases the barrier to universal data abstraction.If the conflict could be

### *D.* Service Management

To guarantee a reliable service management system at the edge of the network, four fundamentalfeatures should be supported, including differentiation, extensibility, isolation, and reliability.

**Differentiation:**

As IoT deployments rapidly grow, we expect to deploy multiple services at the edge of the network, such as Smart Home. There will be different priorities for these services. Critical services, for example, need to be handled earlier, such as things diagnosis and failure alarms. There should be a higherpriority given to health-related services, for example, fall detection or heart failure detection.

**Extensibility:**

As opposed to a mobile system, things in the IoT could be very dynamic, which could pose a huge challenge for extensibility at the edge of the network. Is it possible to easily add new things to the current service without any problems? Or when something wears out, it is possible to replace it easily. A flexible and extensible service management layer in the edgeOS should solve these problems.

**Isolation:**

In mobile OS, if an application fails or crashes, the whole system usually crashes and reboots. Isolation is another issue at the edge of the network. Or, in a distributed environment, the resource can be managed with different synchronization mechanisms, like locks or tokens. However, with a smart edgeOS, this issue may become more complex. There might be several applications that share the same data resource, such as light control. A user should still be able to control their lights, even if one application failed or was not responding. When a user removes the only application that controls the lights from the system, they should still be able to see the lights without losing contact with edgeOS. Introducing a deployment or un-deployment framework could potentially solve this challenge. When the OS detects an application before it is installed, a user can be warned and avoid potential access problems. The isolation challenge also includes separating private data from third-party applications. For example, your activity tracking application should not be able to access your electricity usage data. A well- designed control access mechanism should be added to address this challenge. A layer of the edgeOS that manages services.

**Reliability:**

The reliability challenge at the edge of the network is also a key challenge. We analyze reliability from the perspective of service, system, and data.

- From the service point of view, it can be very difficult to identify the cause of a service failure accurately in a field sometimes. As an example, a broken power cord, a malfunctioning compressor, or a dead battery could be the cause of an air conditioner not working. Depending on the circumstances, a sensor node could have lost connection very easily to the system if the battery died, the connection was bad, or the component wore out. At the edge of the network, it is not enough to just maintain a current service when some nodes lose connection, but it makes more sense to provide the action after node failure. As an example, edgeOS could inform the user if some parts of the system have a high probability of failure, or even alert the user ahead of time if some parts are not responding. The solution to this challenge could be adapted from a wireless sensor network or industrial network such as PROFINET [29].

- From a system point of view, it is very important for the edgeOS to maintain the network topology of the entire system, and each component in the system is capable of sending status/diagnosis information to the edgeOS. A number of services can be easily deployed at the system level using this feature, including failure detection, thing replacement, and data qualitydetection.

- Data sensing and communication pose the biggest reliability challenges from a data point of view. The edge of the network can fail for a variety of reasons, and it can also report low quality data under unreliable conditions, such as low battery levels [30]. We have also proposed several new communication protocols for the collection of IoT data. This protocol enables the support of a large number of sensor nodes and highly dynamic network conditions [31]. In spite of this, the connection reliability is not as good as Bluetooth or WiFi. When both sensing data and communication are unreliable, there is still a challenge in utilizing multiple reference data sources and historical data records to provide reliable service.

### *E.* Privacy and Security

Privacy and data security protection are the most important services at the edge of the network. If IoT is applied to a home, a lot of private information is transmitted. It is possible to make inferences from sensed usage data. For example, with the reading of electricity or water usage, one can deduce whether the house is vacant or not. As a result, how to support a service without harming privacy is a challenge. Some of the private information could be removed before processing, such as masking the faces in a video before processing. Keeping computing at the edge of data resources, which means in the home, is a good way to protect privacy and data security. The security and privacy of network usage remain challenging at the edge.

We can look at WiFi network security as an example. The community needs to be aware of privacy and security. Of the 439 million households that use wireless connections, 49% of them are unsecured, and 80% still use default passwords on their routers. A study found that 89% of public WiFi hotspots were unsecured [32]. The service provider, system and application developers, and end users need to understand that the users' data is at risk. A lack of protection would harm privacy at the edge of the network. For instance, IP cameras, health monitors, and even some WiFi-enabled toys could easily be connected by others.

There is also the issue of ownership of data collected by things at the edge. Like mobile applications, the service provider will store and analyze the end user data collected by things. For privacy protection, it will be better to leave the data at the edge where it is collected and to let the user own it. Users should be able to access data collected at the edge of the network, similar to health record data. To ensure user privacy, highly private data could also be removed during the authorization process by the things.

In addition, there are no efficient tools at the edge of the network to protect data privacy and security. Security protection methods for some things are resource-constrained, so they may not be able to be deployed because they consume a lot of resources. Furthermore, the network becomes vulnerable or unprotected as a result of the highly dynamic environment at the network edge. Some platforms, like Open mHealth, propose standardizing and storing health data [33], but more tools are still needed to handle diverse data attributes.

### *F.* Optimization Metrics

We have multiple layers in edge computing with different computation capabilities, which makes workload allocation a major issue. It is important to determine which layer should handle the workload or how many tasks each part should be assigned. For instance, there are many ways to allocate a workload. As much as possible should be performed on each layer. The extreme cases are fully operated on the endpoint or fully operated on the cloud. This section discusses several optimization metrics, including latency, bandwidth, energy, and cost, to determine an optimal allocation strategy.

**Latency:**

It is important to evaluate performance based on latency, particularly in interaction applications /services [34], [35]. During cloud computing, servers have high computational capacity. Complex workloads, such as image processing, voice recognition, etc., can be handled in a relatively short time,

but latency is not solely determined by computation time, but also by long WAN delays. This can have a dramatic impact on the behavior of real-time/interaction-intensive applications [36]. In order to reduce latency, workloads should be completed in the nearest layer that has the computational capacity to process things at the network edge. In the smart city case, we could use phones to process local photos before uploading all photos to the cloud, which would save a potential missing child's information.

The number of photos and their size will make it difficult to view It is much faster to pre-process at the edge. However, the nearest physical layer may not always be the best option. In order to avoid unnecessary waiting time, we must take into account resource usage information. Since the phone's computation resource is limited when playing games. If the gateway or micro-centre nearby is already occupied, upload a photo instead.

**Bandwidth:**

From a latency point of view, high bandwidth can reduce transmission time, especially for large data (e.g., video, etc.) [37, 38]. The edge can receive data over a high bandwidth wireless network, which will reduce latency compared to working on the cloud. On the other hand, if the workload can be handled at the edge, the latency will be greatly improved. In addition, the bandwidth between the edge and the cloud will be reduced. In the smart home scenario, for example, most of the data can be handled by the gateway via Wi-Fi or other high-speed transmission methods. A short transmission path also enhances reliability. However, the transmission distance cannot be reduced since the edge cannot satisfy the computation. If the data is pre-processed at the edge, the upload data size will be greatly reduced. It is best to pre-process photos before uploading in a smart city to reduce their size. It also saves the users' bandwidth, especially if they are using a carrier's data plan. As a result, bandwidth is saved in both situations and can be used by other edges to upload/download data. Therefore, we need to determine whether a high bandwidth connection is needed and at what speed. Additionally, the workload allocation in each department must be correctly determined. The computation capability and bandwidth usage information in layers should be considered to avoid competition and delays.

**Energy:**

Endpoints are the most valuable resources on the network, so offloading the workload to the edge can be viewed as an energy-free method [22], [39]. Does it make sense to offload the whole workload (or part of it) to the edge rather than to compute locally for a given workload? The key is to balance computation energy consumption with transmission energy consumption. It is important to consider the workload's power characteristics first. Is it computationally intensive? What resources will be required to run it locally? In addition to network signal strength [40], data size and bandwidth also influence the transmission energy overhead [28]. Edge computing is preferable only if the transmission overhead is lower than computing locally. However, if we care about the whole edge computing process rather than only focusing on endpoints, total energy consumption should be the accumulation of each layer's energy cost. As with endpoints, each layer's energy consumption can be estimated as local computation plus transmission cost. Consequently, the optimal workload allocation strategy may change. For instance, the local data center layer is busy, so the workload is continuously uploaded to the upper layer. As compared to computing on an endpoint, multihop transmission can result in a significant increase in overhead, which results in an increase in energy consumption.

**Cost*:***

Edge computing provides service providers, such as YouTube, Amazon, etc., with decreased latency and energy consumption, increased throughput, and an improved user experience. Consequently, they can earn more money for handling the same amount of workload. We can, for example, place a popular video on the edge of the building based on residents' interests. City layer edges are capable of handling more complex tasks, increasing the overall throughput. The service provider's investment is in building and maintaining the things in each layer. To fully utilize the local data in each layer, providers can charge users based on where the data is located. In order to guarantee the profit of the service provider as well as the acceptability of users, new cost models need to be developed.

Workload management isn't easy. There's a direct relationship between the metrics. For instance, due to energy constraints, some workloads must be completed in the city data center. This inevitably impacts the latency of the building server layer. Priority (or weight) should be given to different workloads so that a reasonable allocation strategy can be chosen. Additionally, the cost analysis needs to be performed at runtime as well as consideration of the interference and resource usage of concurrent workloads.

## CONCLUSION

Nowadays, more and more services are being moved from the cloud to the edge of the network because processing data at the edge ensures better reliability and shorter response times. Additionally, more data could be handled at the edge rather than uploaded to the cloud to save bandwidth. IoT and universal mobile devices have changed the role of the edge in computing paradigm from one of data consumers to one of data producers/consumers. Cloud services are increasingly being moved to the edge of the network because processing data at the edge ensures faster response times and better reliability. In addition, if more data could be handled at the edge rather than uploaded to the cloud, bandwidth could also be saved. With the advent of the Internet of Things and the universalization of mobile devices, the edge became more than a data consumer - it was a data producer and consumer. Data can be processed and massaged more efficiently at the edge of networks. In this paper, we describe the concept of edge computing, based on the rationale that computing should take place near data sources. In the following section, we discuss how edge computing can flourish in smart environments such as cities and homes by offloading cloud services. Additionally, we introduce collaborative edge, which connects end users and clouds physically and logically so that not only does the conventional cloud computing paradigm still hold true, but it can also connect long-distance networks for data sharing and collaboration due to the close proximity of data. Last but not least, we propose challenges and opportunities worth pursuing, such as programmability, naming, data abstraction, service management, privacy and security, and optimization metrics. Edge computing is here, and I hope we can take advantage of it. As a result of this paper, the community will be made aware of this issue.

## REFERENCES


1. M. Armbrust *et al.*, "A view of cloud computing," *Commun. ACM*,vol. 53, no. 4, pp. 50–58,2010.

2. S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system,"*ACM SIGOPS Oper. Syst. Rev.*, vol. 37, no. 5, pp. 29–43, 2003.

3. J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

4. K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Proc. IEEE 26th Symp. Mass Storage Syst. Technol. (MSST)*, Incline Village, NV, USA, 2010,pp. 1–10.

5. M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput.*, vol. 10. Boston, MA, USA, 2010, p. 10.

6. K. Ashton, "That Internet of Things thing," *RFiD J.*, vol. 22, no. 7,pp. 97–114, 2009.

7. H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, "Vision and challenges for realising the Internet of things," vol. 20, no. 10, 2010.

8. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions,"*FutureGener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

9. "Cisco global cloud index: Forecast and methodology, 2014–2019 white paper," 2014.

10. D. Evans, "The Internet of Things: How the next evolution of the Internet is changing everything," CISCO White Paper, vol. 1, pp. 1–11, 2011.

11. D. Evans, "The Internet of Things: How the next evolution of the Internet is changing everything," CISCO White Paper, vol. 1, pp. 1–11,2011.

12. A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, 2008.

13. E. Cuervo *et al.*, "MAUI: Making smartphones last longer with code offload," in *Proc. 8th Int. Conf. Mobile Syst. Appl. Services*,San Francisco, CA, USA, 2010, pp. 49–62.

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

15. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of things," in *Proc. 1st Edition MCC Workshop Mobile Cloud Comput.*, Helsinki, Finland, 2012, pp. 13–16.

16. *Boeing 787s to Create Half a Terabyte of Data Per Flight, Says Virgin Atlantic*. Accessed on Dec. 7, 2016. [Online]. Available: https://datafloq.com/read/self-driving-carscreate-2- petabytes-data-annually/172

17. *Self-Driving Cars Will Create 2 Petabytes of Data,What are the Big Data Opportunities for the Car Industry?* Accessed on Dec. 7, 2016. [Online].Available: http://www.computerworlduk.com/news/data/boeing-787screate-half-terabyte-of-data-per- flight-says-virgin-atlantic-3433595/

18. *Data Never Sleeps 2.0*. Accessed on Dec. 7, 2016. [Online]. Available: https://www.domo.com/blog/2014/04/data-never-sleeps-2-0/

19. (2016). *OpenFog Architecture Overview. OpenFog Consortium Architecture Working Group*. Accessed on Dec. 7, 2016.[Online]. Available: http://www.openfogconsortium.org/wp-content/

20. uploads/OpenFog-Architecture-Overview-WP-2-2016.pdf

21. 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)*, Washington, DC, USA, 2015, pp. 73–78.

22. K. Ha *et al.*, "Towards wearable cognitive assistance," in *Proc. 12thAnnu. Int. Conf. Mobile Syst. Appl. Services*, Bretton Woods, NH, USA,2014, pp. 68–81.

23. B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud:Elastic execution between mobile device and cloud," in *Proc. 6th Conf.Comput. Syst.*, Salzburg, Austria, 2011, pp. 301– 314.

24. A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, "Saving portable computer battery power through remote process execution,"*ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 2, no. 1,pp. 19–26, 1998.

25. G. C. Hunt and M. L. Scott, "The coign automatic distributed partitioning system," in *Proc. OSDI*, vol. 99. New Orleans, LA, USA, 1999,pp. 187–200.

26. K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

27. S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. IEEE INFOCOM*, Orlando, FL, USA,2012, pp. 945–953.

28. L. Zhang *et al.*, "Named data networking (NDN) project," Xerox Palo Alto Res. Center, Palo Alto, CA, USA, Tech. Rep. NDN-0001, 2010.

29. D. Raychaudhuri, K. Nagaraja, and A. Venkataraman, "Mobility First: A robust and trustworthy mobility-centric architecture for the future Internet," *ACM SIGMOBILE Mobile Compute. Commun. Rev.*, vol. 16, no. 3, pp. 2–13, 2012.

30. J. Feld, "PROFINET—Scalable factory communication for all applications, "in *Proc. IEEE Int. Workshop Factory Commun. Syst.*, Vienna,Austria, 2004, pp. 33–38.

31. J. Cao, L. Ren, W. Shi, and Z. Yu, "A framework for component selection in collaborative sensing application development," in *Proc. 10th IEEE Conf. Coll. Comput. Netw. Appl. Worksharing*, Miami, FL, USA, 2014, pp. 104–113.

32. F. DaCosta, *Rethinking the Internet of Things: A Scalable Approach to Connecting Everything*. New York, NY, USA: ApressOpen, 2013.

33. *WiFi Network Security Statistics/Graph*. Accessed on Dec. 7, 2016. [Online]. Available: http://graphs.net/wifi-stats.html/

34. *Open Mhealth Platform*. Accessed on Dec. 7, 2016. [Online]. Available: http://www.openmhealth.org/

35. K. R. Jackson *et al.*, "Performance analysis of high-performance computing applications on the Amazon Web services cloud," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Indianapolis,IN, USA, 2010, pp. 159–168.

36. A. Li, X. Yang, S. Kandula, and M. Zhang, "CloudCmp: Comparing public cloud providers," in *Proc. 10th ACM SIGCOMM Conf. Internet Meas.*, 2010, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/1879141.1879143

37. M. Satyanarayana, "Mobile computing: The next decade,"*SIGMOBILE Mobile Compute. Commun. Rev.*, vol. 15,no. 2, pp. 2–10, 2011. [Online]. Available: http://doi.acm.org/

38. 10.1145/2016598.2016600

39. A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *SIGCOMM Compute. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, Dec. 2008. [Online]. Available: http://doi.acm.org/10.1145/1496091.1496103

40. M. Armbrust *et al.*, "A view of cloud computing," *Commun.ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672

41. A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Compute.*, Boston, MA, USA, 2010, p. 4. [Online]. Available: http://dl.acm.org/citation.cfm?id=1863103.1863107

42. N. Ding *et al.*, "Characterizing and modeling the impact of wireless signal strength on smartphone battery drain," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 1, pp. 29–40, Jun. 2013. [Online]. Available: http://doi.acm.org/10.1145/2494232.2466586