Modern Applied Science; Vol. 13, No. 8; 2019
ISSN 1913-1844 E-ISSN 1913-1852
Published by Canadian Center of Science and Education

Cloud IoT as a Crucial Enabler: a Survey and Taxonomy

Dr. Nidhal Kamel Taha El-Omari[1]

[1] Department of Software Engineering, Faculty of Information Technology, The World Islamic Sciences and Education University (WISE), Amman- Jordan

Correspondence: Dr. Nidhal Kamel Taha El-Omari, Department of Software Engineering, Faculty of Information Technology, The World Islamic Sciences and Education University (WISE), Amman – Jordan. E-mail: nidhal.omari@wise.edu.jo; omari_nidhal@yahoo.com

Received: March 1, 2019 Accepted: May 22, 2019 Online Published: July 27, 2019

doi:10.5539/mas.v13n8p86 URL: https://doi.org/10.5539/mas.v13n8p86

Abstract

Commonly, the technology of information scales by an order of magnitude and with high probability reinvents itself every five years or so. However, the long-standing dream has definitely become a reality today that saying you need merely a credit card to get on-demand instant accesses to a large pool of thousands, if not millions of computers founded in tens of data centers scattered across the globe.

As a matter of fact, Cloud Computing is indeed a new radical paradigm shift that evolved out of utmost needs for hosting and delivering all the things electronically as well-defined services over the Internet. Its aim is not only to provide improved computerized services but also innovative ones to every user from ordinary-home end-users to professional workers.

Another further long-held dream of computing that has recently emerged as a reality is the CloudIoT paradigm where the cloud-based application platforms are enhanced to generate smart decisions and usable intelligence based on Internet-connected semi-autonomous smart small sensors that can sense, interrupt, and interchange data between each other as well as with the same computing clouds.

With the intention of reaching the right cloud computing vision, it is, however, insufficient to just track a set of research and development actions that address the major concerns without practical support from both industrial and research communities. Rather, there is a necessity to enact a series of insight development strategies and policies that not only ensure that the right issues are well-timed addressed, but also that the most appropriate actions are taken and accomplished.

Additionally, one of the key points to consider in this paper is that this in-depth evaluation of using Cloud computing may find points lacking in the cloud environments that could open up new research opportunities to be further investigated or enable new speed-to-market scenarios.

Keywords: Big Data, Cloud Computing (CC), Digital Image Processing, Grid Computing (GC), Infrastructure-as-a-Service (IaaS), Internet-of-Things (IoT), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS), Utility Computing (UC), Virtualization

1. Introduction

Six extensive technology revolutions make paradigm shifts upon the globe and mostly they are the most invasive and world-changing discovery of this modern era. They are enough to rock the world and make significant qualitative changes in how individuals work and live. They have benefitted individuals as well as the entire societies. The first one started in the mid-1950s, is the jumping to transistors as an alternative for the earlier Cathode Ray Tubes (CRTs). Prior to this digital revolution, the computer filled the size of an entire large room and consumed vast amounts of electric energy. This revolution has had a dramatic impact on the way computers are designed and structured and, in turn, revolutionized the way one thinks and works.

The second revolution is the manufacturing of the microprocessor, which tightly integrates the CPU functions onto a single tiny microchip. This revolution practically injects intelligence into the most dummy machines and ignites a new breakthrough in the world of electronics. Followed 15 years later, the advent of the Internet makes the third revolution that forms the unbounded world with the ubiquitous data (Prajkta and Keole, 2012). Going forward, the invention of this communication network is considered an IT boom that accelerates the communication capacity across the globe and rapidly revolutionizes the landscape of information technology (Shawish and Salama, 2014). It is the lifeblood that changes the face of computing and reshapes the entire world into an increasingly connected world by enabling any communicating entity to communicate with any other communicating entity located elsewhere in the world without much care for the physical distance between them (Malik et al., 2014). Thus, as long as there is Internet connectivity, you can be highly available wherever/whenever you are not! (Ramzan and Alawairdhi, 2014)

Then, the community of academic and industrial researchers is trying to rely on the power of the software to overcome and conquer infrastructure dependency. Thereby, the next major revolution, hardware disconnection, becomes to free the developers from thinking about the actual physical computer architectures of their applications by reducing hardware and operating system (OS) dependence. This is by virtualization technology, abstraction, and bytecode engines. Virtualization of a computer system as a whole is a conceptually way to mask a system as another system and simulate its functionality in a means to be pretending as more than one system. It is a technique to provide isolation and parallel synchronized resource sharing of the underlying hardware components and to facilitate one piece of physical hardware to run a bunch of multiple virtual environments synchronously. It is a way to multiplex hardware resources into several virtual ones and therefore enabling multiple instances of operating systems to run concurrently on the same underlying physical platform. It might either be a programming environment, an operating system or more generally an arrangement of both together. On the other hand, bytecode, also termed portable code, is platform-independent software to allow the execution of the same code on different hardware platforms. It is a way of relieving up the applications from the platform constraints. The “Microsoft .NET Runtime Engine (.NET)”, and the “Java Runtime Engine (JRE)” are the most widely used cases of the bytecode engines. While they may both use similar virtual-machine architecture, both are nonetheless often akin to “write once, run anywhere (WORA)”. (Aspen and Kaitlyn, 2017)(Anne-Lucie *et al.*, 2017)(Bardsiri and Amid, 2012)(Sommerville, 2015)(Deitel Pau and Deitel, 2017)(Sameera and Iraqi, 2017)(Prajkta and Keole, 2012)

Each one of the virtualization and the bytecode creates a layer of abstraction that coat the underlying physical infrastructure and work as a shield that cleverly hides the underlying complexity (Ashdown and Kyte, 2015). This layer empowers the physical computer architectures to serve certain needs in a transparent manner without keeping consumers just worrying about how that is accomplished (Ashdown and Kyte, 2015)(Marques *et al.*, 2018)(Yuzhong and Lei, 2014).

By far, rapid improvements in the Internet services and the virtualization techniques have given birth to the next industrial revolution, the fifth one, which becomes the fundamental need of the hour (Sareen, 2013)(Naveen and Harpreet, 2013)(Han *et al.*, 2016)(Laverty, Wood and Turchek, 2014)(Ali *et al.*, 2015). This revolution, cloud computing (CC), is the hottest great change in the software industry - most likely the biggest one of the five. It is the extensive technology revolution that rapidly changes the IT-based industry since it evolved up and makes a radical change toward a sharable informative society (Edlund, 2012)(Chowdhary and Rawat, 2013)(Laverty, Wood and Turchek, 2014). It is the revolution that has taken customers from running applications on their non-networked standalone computers to complete services, including infrastructure and applications, running in the cloud-hosted environment (Michael and Rajiv, 2012)(Edlund, 2012)(Filippi and McCarthy, 2012)(Tsz Lai, Trancong and Goh, 2012)(Taneja, Taneja and Chadha, 2012)(Foster *et al.*, 2008)(Sareen, 2013)(Naveen and Harpreet, 2013)(Subbiah, Muthukumaran and Ramkumar, 2013)(Chowdhary and Rawat, 2013)(Deitel Pau and Deitel, 2017). It is the mechanism of migrating Web applications to the cloud services and integrating these information-rich services efficiently into the existing computing infrastructures (Michael and Rajiv, 2012)(Edlund, 2012)(Filippi and McCarthy, 2012)(Tsz Lai, Trancong and Goh, 2012)(Prajkta and Keole, 2012)(Taneja, Taneja and Chadha, 2012)(Foster *et al.*, 2008) (Sareen, 2013)(Naveen and Harpreet, 2013) (Chowdhary and Rawat, 2013). These electronic measured services, that the user consumes while connected to the cloud, are a combination of hardware and software and may be any and all resources, network and physical hardware components, computational power, application software, data storage, back-up facilities, and various development tools (Michael and Rajiv, 2012)(Edlund, 2012)(Filippi and McCarthy, 2012)(Tsz Lai, Trancong and Goh, 2012)(Taneja, Taneja and Chadha, 2012)(Sareen, 2013)(Naveen and Harpreet, 2013) (Chowdhary and Rawat, 2013)(Chandra and Neelanarayanan, 2017)(Ali *et al.*, 2015).

Since then, it is rapidly growing as an alternative to conventional computing where many research papers devoted to the cloud migration have been published to recognize cloud computing as a well-known research area and to turn the Internet into a service-oriented delivery infrastructure (Laverty, Wood and Turchek, 2014)(Gholami, Daneshgar and Beydoun, 2017). Even with its relatively short life span, the structural impacts of Cloud technology upon the globe are bigger and more complex than the radical structural changes seen during the past industrial revolution (Filippi and McCarthy, 2012). Just as the industrial revolution progressively alienates labors from the revenue of production and creates an imbalance in authority structures, CC is regarded as a player with a significant role in enabling greater reengineering of the software industry (Filippi and McCarthy, 2012). As it is the next stage in the Internet's evolution, gathered under the name "cloud computing", the software industry continues shifting towards service-based business models that increase productivity, shrink timelines, reduce the long-term operational and maintenance costs, and allow an enterprise to focus on its core activities rather than trying to solve computer automation problems (Temkar, 2015)(Aspen and Kaitlyn, 2017)(Arora *et al.*, 2017).

By comprising the use of information and communication technologies (ICT), researchers and practitioners on CC have been strongly characterized by a multidisciplinary nature. Whereby, by this big change "mobility" and widely-used social technologies are made possible by the cloud which comprises the use of ICT in order to deliver special on-demand cost-effective services to the clients, and entails the using of new technological trends in new structural processes. (Michael and Rajiv, 2012)(Edlund, 2012)(Filippi and McCarthy, 2012)(Laverty, Wood and Turchek, 2014)

Although that is the ideal scenario, the globe is operating on a different scale. Day by day, there are increasingly multiple billions of dollars being spent by the likes of Microsoft, Google, IBM, and HP to create real commercial large-scale data centers holding hundreds of thousands of connected computing elements that are widely spread over the world. (Foster *et al.*, 2008)

The latest frontier revolution, "Internet-of-Things (IoT)", forms the next wave in the era of computing and presents clearly the smart vision of the world. Despite that, this frontier paradigm is still under progress but stands to become revolutionary in many different computing arenas. However, the notion of this term, IoT, is not a new concept to IT professionals; it firstly appeared in 1998 within a presentation by Kevin Ashton (Perera *et al.*, 2014):

"*The Internet of Things has the potential to change the world, just as the Internet did. Maybe even more so*".

Ever since, researchers and practitioners continue to be successful in this trend. In view of that, it is anticipated that more than 50 billion smart things will be connected to the Internet within the next few coming years. As such, it is expected in the near future that the number of connected objects over the Internet will be more than seven times of world human population and it is anticipated to keep growing at a fast pace. Although the research into the IoT is in its infancy, IoT isn't considered as "science fiction" or something that may or may not happen in the future; it is definitely a reality and it is fast becoming an affordable technology!! Even though that CC provides the virtual infrastructure that forms the computational part of IoT discipline, this revolution will be outside the realm of the other revolutions. (Alessio *et al.*, 2014)(Roy and Sardda, 2016)(Al-Fuqaha *et al.*, 2015)

While there are greater interactions between CC computing and its successive technology, IoT, the former provides the virtual infrastructure that forms the computational part of the latter discipline. IoT, on the other hand, is based on several other prominent research communities such as wireless-linked Sensor Networks (WSN), Mobile Computing (MC), Cyber-Physical Systems (CPS), and others. Consequently, the never-ending advances in each field of these communities will actually speed the progress in IoT. Going further, devices, including smart sensors and smartphones, will also accelerate the progress towards further development upon IoT. (Maria G. Koziri and Loukopoulos, 2017)(Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)

To explore further the arguments discussed so far, this paper consists of eighteen sections. Section 1 elaborates what Cloud technology actually is. Section 2 gives a closer look at how people are already using CC without realizing that it is a CC existing. The author in Section 3 describes what distinguishes CC from other relevant research areas. Section 4 defines the CC and then states the essential characteristics of this research area which make CC being clouds. While Section 5 states how both virtualization and cloud computing works in a complementary manner, Section 6 states the different virtualization categories and the strong relationships among them.

Although there are powerful similarities between CC and other close-related technologies, there are also many considerable differences between them. So, the distinction between Cloud Computing (CC) and Grid Computing (GC) is discussed in Section 7. After that, the architecture of CC is described in Section 8. The different services as well as the different deployment models of clouds are discussed in Section 9 and Section 10, respectively. Section 11 investigates the perceived benefits that can be reaped by migrating to the cloud technology, whereas Section 12 discusses several key success factors and research issues that must be considered carefully and understood clearly prior to adopting and shifting to the cloud. Section 13 briefly discusses the key characteristics of the Internet of Things (IoT) and the Big Data and how they are working with CC in a complementary framework. Section 14 shows how the high-productivity image processing functions are coherently integrated within the CC paradigm. Section 15 sheds light on the implementation proposed frame for getting into the Cloud and Internet of Things. While Section 16 suggests how to select the right cloud-like services that most closely fit the requirements, Section 17 points to the required arrangements that should be before CC transition. The final section, Section 18, concludes this paper and highlights new avenues for further research in this area.

2. CC is Already Exists

Without people realizing, they are already using some elements of cloud-like services, SaaS in specific, in their day-to-day life such as Facebook, Twitter, Google+, Instagram, WhatsApp, LinkedIn, YouTube, and other widely-used social media websites (Abdu *et al.*, 2017)(Karim *et al.*, 2017)(Gaur and Anurag, 2017). Still more, public e-mails (such as Yahoo mail, Google mail, Outlook.com, Windows Live Hotmail, and so on), cloud storage services (such as Microsoft OneDrive, Dropbox, and so on), online photo storage services, e-bank services, e-trade services, and online office document management services (like Google Docs and iWork Pages) are also good examples of the communication channels that are based on cloud technology, especially SaaS (Sommerville, 2015)(Abdu *et al.*, 2017)(Gaur and Anurag, 2017). So, most of us, individually and organizationally, use these services in reality without essentially being aware of giving much thought that they are actually broad examples of CC.

Not surprisingly, E-mail service that runs obviously in a Web browser can be taken as a simple example on the already mentioned point; this service gives the end-user the ability to compose a message from his\her client through any e-mail browser, then attach any necessary files and, finally, send this email remotely to one or more recipients. So the scenario is that the e-mail is sent from an e-mail client to an e-mail server operated by the service provided. Besides that all the recipients can read the same email by their different Web browsers, the three sides (the sender, the e-mail server, and the recipients) are working remotely on the same server and using the service of the Web browser (Sommerville, 2015). In addition, the software itself and the exact data storage for their e-mail accounts do not actually exist on their own computers, rather they surely exist on the server-side of the e-mail service provided (Sareen, 2013)(Gaur and Anurag, 2017). Therefore, both of the remote destinations (the sender and the recipients) are logging remotely into a web application that runs on the server-side of the e-mail service provider (Gaur and Anurag, 2017).

3. CC and Other Styles of Computing

Broadly speaking, this new phenomenon, Cloud computing (CC), has borrowed some fundamental notions from other relevant preceding computing research areas such as Virtualization, Abstraction, High-Performance Computing (HPC), Resource Pooling, Content Outsourcing, Internet Delivery, and Real-Time Monitoring (Foster et al., 2008) (George Pallis, 2010)(Bratterud, Happe and Duncan, 2017)(Beacham and Duncan, 2017)(Arora et al., 2017). Furthermore and as indicated in Figure 1, CC is not completely a new paradigm, but rather a new operational model that it is indeed evolved out of other relevant interrelated prominent computing models that have been subject to extensive years of research and development, such as Grid Computing (GC), Utility Computing (UC), Multi-processor systems, Multicore architectures, Parallel Computing, Internet, Cluster Computing, and Distributed Systems (Foster et al., 2008)(Jiang and Yang, 2010)(Al-Ta'ee, El-Omari and Kasasbeh, 2013)(Kaur, 2015)(C.Vijaya and P.Srinivasa, 2016). So, CC nearly includes everything that can be obtainable by other already existing technologies (George Pallis, 2010).

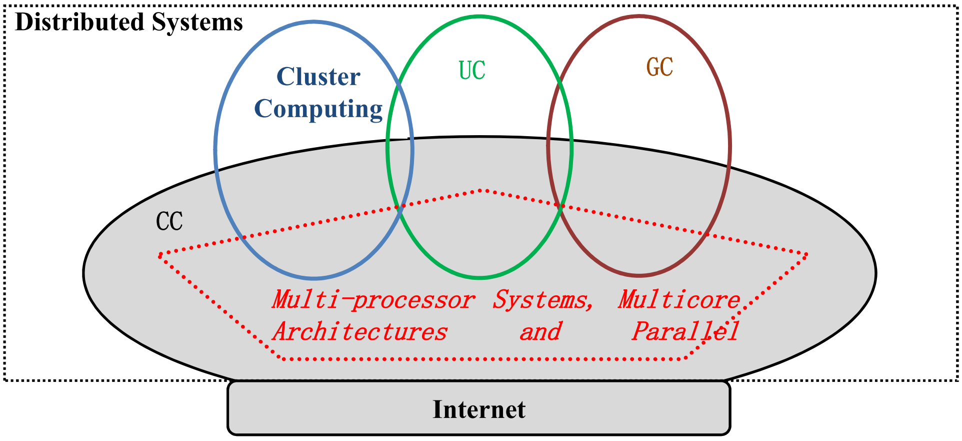

Distributed Systems
Cluster Computing
UC
GC
CC
Multi-processor Systems, Multicore
Architectures and Parallel
Internet

Figure 1. The Organization of Distributed Systems

While there are greater interactions among these communities and the precise boundaries between these technologies are difficult to recognize, it is difficult to draw borderlines between them. However, all of them are promising to deliver IT ranging from infrastructure to applications as innovative services and they fuel the explosive growth of CC. This section strives to give a review and taxonomy to compare and contrast CC from the above stated relevant technologies.

3.1 Multicore & Multiprocessor Architectures

Beyond any doubt, Parallel Processing, with the advent of multicore architectures and multi-processor systems, is widely used as a driver for satisfying the computationally intensive requirements of customers (Kaur and Chana, 2015)(Venkatachalapathy *et al.*, 2016)(C.Vijaya and P.Srinivasa, 2016). In comparison with single-core processors, multicore architectures significantly improve computing efficiency by facilitating concurrent and parallel processing of multiple operations and, in turn, reducing infrastructure costs, cutting down its operating energy, and increase its capabilities (C.Vijaya and P.Srinivasa, 2016). Since multiple "cores" on the same CPU chip can work in parallel on separate operations, each added core is engineered to significantly improve the overall-potential system performance (C.Vijaya and P.Srinivasa, 2016). To be exact, this is what it's so-called CMP, or Chip-level Multiprocessing. Multiple processing cores, from the perspective of energy, are nearly consuming the same amount of electric power and cooling load.

On the other hand, scientists and experts from both research and industrial communities have eventually succeeded to chain two or more CPUs, allowing them to work in parallel as a multiprocessor system to parallelize tasks across them. This is what it's called SMP, or Simultaneous Multiprocessing. Figure 2 demonstrates schematically how by partitioning the overall program and the input data, the execution of a single program can be virtualized and scheduled across a set of multiple computers in a distributed system, different CPUs in a multiprocessor system, different cores within the same CPU in a multicore architecture, and different threads within the same core in a multithread architecture.(Mirarab, Fard and Shamsi, 2014)(Kaur and Chana, 2015)

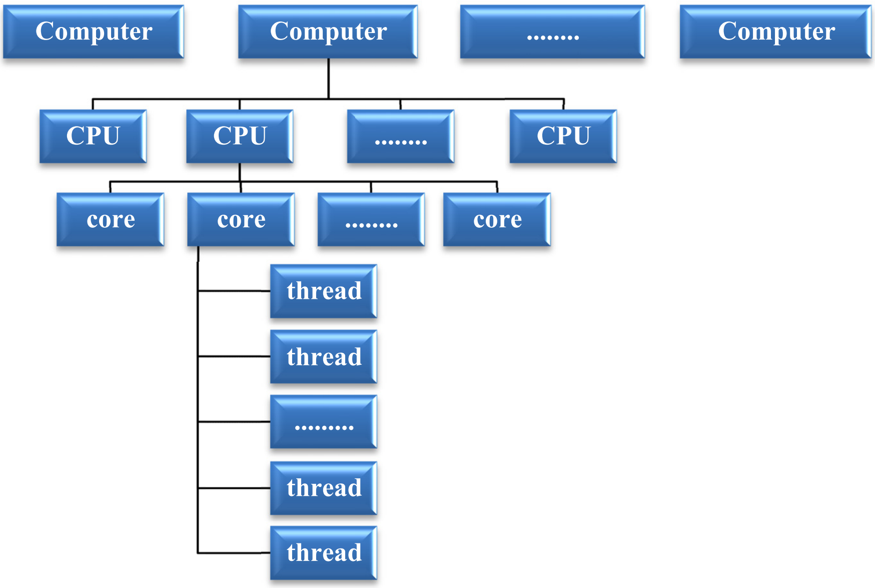


Figure 2. Different Workers of Timesharing

3.2 Conventional Distributed Scenarios

Researchers and scientists throughout the years continue inventing things that make incredible changes in computing. Earlier, large computers, mainframes, were used for large professional software systems and kept behind glass walls and closed doors in an engineered environment that is fully air-conditioned, where only highly skilled operators can operate them and only large enterprises may be in a position to afford them (Foster *et al.*, 2008)(Jiang and Yang, 2010).

Then things are different than they were ago. There is a greatly increased demand for computation-intensive power especially for analyzing massive-scale data or solving a single unmanageable big problem (Akshatha and Manjunath, 2016). For example, some firms might need high-performance computing elements to solve complex problems that need CPU-intensive calculations and/or large-scale storage elements. However, these computational resource elements are only needed temporarily and may not available locally. Whereby, buying these elements is cost-effective but it may have technical and financial risks. (Foster *et al.*, 2008)(Jiang and Yang, 2010)(Akshatha and Manjunath, 2016)

Short ahead, in their ways of seeking high-level computation in terms of response time and budgeting issues, scientists and experts have increasingly duplicated their efforts to coordinate many networked resources to work together to accomplish a big unmanageable problem by dividing it into small manageable sub-problems and then solving them through a coordinated manner by using a number of connected processing elements (Akshatha and Manjunath, 2016). These different resource elements, such as mass storage devices, data sources, and computational power are interconnected by a computer network and they are deeply cooperated in accomplishing the assigned main problem and can be exploited by users/clients as a single-unified resource (Akshatha and Manjunath, 2016). However, these consumed resources may be geographically distributed in different places, sometimes across the globe, and the users are unaware of their data physical locations (Bardsiri and Amid, 2012)(Sommerville, 2015)(Deitel Pau and Deitel, 2017)(Mirarab, Fard and Shamsi, 2014).

If the distributed system architecture is designed correctly, the failure of a single resource should not take down other resources in the system. Whenever any component, namely processing element or node, fail to function in a satisfactorily mode then other elements will take care of it. (Foster *et al.*, 2008)(Jiang and Yang, 2010) (Venkatachalapathy *et al.*, 2016)

Likewise, as these nods are not specifically organized just only to work together to accomplish a single large job, the arrangement of computers all over the place are being randomly added and removed from the set working on the problem. Evidently, the great growths in networks and its communication protocols paved the way for seeing this reasonable technology that has two faces of the same coin: the decentralization of processing at the system level, and the integration of the information resources at the logical level (Foster *et al.*, 2008)(Al-Ta'ee, El-Omari and Kasasbeh, 2013). Consequently, since a single calculation may be running across a set of machines, researchers and practitioners, in turn, refer to this practical solution as Distributed Systems (Al-Ta'ee, El-Omari and Kasasbeh, 2013)(Venkatachalapathy *et al.*, 2016). Their rational targets are to connect users with their needed IT resources in open-transparent, cost-effective, scalable, and reliable environments.

However, even with the reaped benefits of moving in the direction of Distributed Systems, there are some concerning things related to this integrated technology (Kaur, 2015)(Akshatha and Manjunath, 2016). The most depressing thing is that when working with these earlier distributed systems there may be some long latencies and unexpected failures and therefore information cannot be delivered to the right users at the right time (Kaur, 2015)(Akshatha and Manjunath, 2016). This necessitates the development of the next successive technology, namely Cluster Computing (Ali *et al.*, 2015)(Kaur, 2015)(Akshatha and Manjunath, 2016).

3.3 Cluster Computing

As the need to perform complex data manipulations grew, different researchers and experts still look for overpowering the centralized systems while getting more robust hardware and better network interconnection (Ali *et al.*, 2015)(Kaur, 2015)(Akshatha and Manjunath, 2016). To this aim and inspired by the concepts of parallel programming and distributed system, Cluster technology was designed where a collection of interconnected stand-alone computers were combined together to work logically as a single integrated entity, called a cluster (Kaur, 2015). But never the less cluster computing is still distributed-parallel computing that is closely tied to advances in communications and network technologies (Kaur, 2015)(Ali *et al.*, 2015) (Akshatha and Manjunath, 2016).

Each cluster as a whole appears as a single system to users and applications. Each stand-alone computer within a cluster is called a node which, in turn, has a single or multiprocessor system with memory, input/output facilities, and operating system (OS). Generally, these nodes are of similar homogeneous components; they may be incorporated in a single cabinet, or physically separated within the same location and connected to each other over some high-speed local area networks (LANs). (Jiang and Yang, 2010)(Gumbi and Mnkandla, 2015)(Kaur and Chana, 2015)

With the advent of this technology, users simply get the real feeling that they are working with a single computer, but actually, they are utilizing a large number of components viewed as a single coherent image. It is proved that a big cluster has almost the same power as a scientific supercomputer at a reasonable cost. Because each overall cluster has similar components, formerly referred to as nodes, the fault in one component does not affect the availability of the whole cluster. When one component goes down for any technical reason, then the other components take its place and the system continues working with the remaining components giving the users one system image. Indeed, the cluster can be available with only one component, but its processing power is further decreased by the idle components. Furthermore, collaboration and synchronization between these nodes are done by using Massage Passing Interface (MPI). (Jiang and Yang, 2010)(Manju and Sadia, 2017)(Kaur, 2015)(Kiranjot and Anjandeep, 2014)

However, in spite of the benefits of moving in the direction of Cluster Computing, scientists and experts from both research and industrial communities increasingly realized that these high-end clusters are both quite expensive to activate and, perhaps more importantly, require large teams to operate. In addition, as the amount of data has exploded in the last few decades, there is an ever need for a larger computing power that cannot be reached with Cluster Computing that has a low capability at the computational end (Malik *et al.*, 2014). Due to these limitations, the next major technology, namely GC, comes in the picture to provide the ability to federate a large number of resources to perform computational extensive tasks for end-users. (Manju and Sadia, 2017) (Kaur, 2015)(Kiranjot and Anjandeep, 2014)

3.4 Grid Computing (GC)

Later, in the mid-1990s, GC (or the use of a "computational grid") was coined to allow the creation of a "virtual supercomputer" by assembling spare computational resources within other computing centers. This technology obtains an on-demand large-scale computation power harnessed from the idle processing power of various autonomous computing elements. Although each autonomous processing unit with its own applications can be managed autonomously, it can corporate in its idle time to solve a one or more piece of the grid problem. However, these federated computing units may be heterogeneous, loosely coupled, and may exist in computing centers geographically distributed across multiple sites and countries in different places, sometimes across the globe. In fact, this large number of computation units is incorporated by a grid to be as a single system image used for running large-scale computational applications that require high-performance computing (HPC), like image processing, nuclear researches, numerical weather forecasting, DNA sequencing, and computer games. Although that GC has originally emerged from academic and non-profit research institutes, it later entered the commercial and industrial world, like banks, hospitals, and different factories. (Foster *et al.*, 2008)(Jiang and Yang, 2010)(Ali *et al.*, 2015)(Yuzhong and Lei, 2014)

Despite the fact that the entire cluster's computational resources are dedicated and worked as a single unit and nothing else, a grid offers a way of utilizing and federating the idle processing power of different resources to compute one large job. Back-and-forth, GC is more popularly known as a distributed architecture of a collection of servers bonded together in a network to carry out a single complex problem. Obviously, only one computer can remotely control all these units. This remote computer is used to utilize the idle processing power to yield one large unmanageable job. The end-users of this remotely controlled computer get instant access to these computational resources according to their demands with very limited knowledge about the details of the distributed operations or at least where these resources are physically located. (Jiang and Yang, 2010)(Manju and Sadia, 2017)(Kaur, 2015)(Kiranjot and Anjandeep, 2014)

Being more specific, as this technology is a way of offering more IT resources, the job itself is grained down into a number of small dependent or independent portions, called processes or tasks, which can be executed synchronously on different computing units in a truly efficient environment (C.Vijaya and P.Srinivasa, 2016). Although these tasks are distinct conceptually, they needn't essentially to be mutually exclusive; they can be performed simultaneously (Bardsiri and Amid, 2012)(Foster *et al.*, 2008)(Jiang and Yang, 2010)(Kaur, 2015). As soon as these tasks are accomplished, the results are sent back to the initiating machine to integrate a cohesive output that equated to the original large job (Foster *et al.*, 2008)(Bardsiri and Amid, 2012)(Kaur, 2015)(Jiang and Yang, 2010). Thus, grid technology, in its simplest form, may be defined as one "super virtual computer" composed of many networked loosely coupled computers working together to execute humongous tasks grained from a certain job (Foster *et al.*, 2008)(Bardsiri and Amid, 2012)(Kaur, 2015)(Jiang and Yang, 2010).

Functionally, GC can be further classified into two types: Computational Grids which focuses primarily on using CPU's for computation-intensive operations rather than managing data, and Data Grids which primarily concentrates on managing and timesharing of large amounts of distributed datasets rather than using CPU's. (Manju and Sadia, 2017) (Kaur, 2015)(Kiranjot and Anjandeep, 2014)(Bardsiri and Amid, 2012)(Jiang and Yang, 2010)

Since it is possible to access all grid resources as though they were multiple owned by more than one enterprise, this gave rise to the notion of the research collaborations which is what it's idiomatically termed as the Virtual Organizations (VOs). These VOs bring together researchers, experts, and scientists, working in the same field, from different research institutes and universities around the world. Generally, these organizations are funded by some governments and/or leading universities and they are mainly available for non-profit work. While Grids are mostly used by these VOs as bases for their scientific researches, end-users as customers of grids might participate in one or more VOs to share some or all of its offered computational resources. (Bardsiri and Amid, 2012)(Sommerville, 2015)

Although cluster computing and GC are both from other vital parts of distributed systems and both have distinct elements that interact with each other, the former is subsumed by the latter and so the former can be considered as a special case of the latter. However, the latter, namely GC, is evolved out of the former one to offer a way of using the IT resources more optimally. Since there is often some confusion about the difference between them, the following bullet points clarify these distinctions: (Foster *et al.*, 2008)(Bardsiri and Amid, 2012)(Kiranjot and Anjandeep, 2014)(Jiang and Yang, 2010)(Malik *et al.*, 2014)(Manju and Sadia, 2017)

- The big notable difference is that a cluster is a traditional onsite aggregation of homogenous computational resources while grids involve aggregation of heterogeneous computational resources in a more dynamic and adaptive way. In other words, while all cluster computational resources have the same underlying physical hardware and tied to only one OS, the resources that are part of a grid may have different hardware facilitating several OSs.
- Another notable difference that distinguishes grid technology from cluster technology is that the cluster elements tend to be tightly coupled while the grid elements tend to be loosely coupled.
- While the cluster's computational resources are normally contained in a single location or complex in close proximity to clients, Grid may be geographically dispersed across multiple sites and countries across the world and connected via WANs.
- Accounting for the way resources are handled and viewed, there is another major difference. Every Grid processing element, namely node, behaves as an autonomous entity with its own resource manager. Since each node looks as if it were a single rational system vision, the job management and the scheduling system are actually distributed evenly among all the different nodes.

  In the case of Cluster Computing, there is, however, another situation. The whole system of each cluster with its all incorporated nodes looks as if it were a single coherent system vision and the different elements are managed well by only one centralized resource manager which has only one centralized job management and scheduling system.
- The cluster is normally owned by one firm, whereas the grid is owned by one or more firms.

Similar to previous technologies, despite the benefits of moving in the direction of Grid Technology, there are still some hard limitations in terms of this integrated technology. First and foremost, because each grid has its special requirements on the running environment, it will be restricted to a particular type of programming languages (i.e. its native programming language), software libraries, and applications. Second and more importantly, if one or more libraries that are needed by a program in regard to a specified job are not accessible, the whole job will be terminated even if there are enough available resources. The same thing may take place when one program needed by a job is no longer accessible. (Jiang and Yang, 2010)(Al-Ta'ee, El-Omari and Kasasbeh, 2013)

Another concerning thing is that, since different grid systems in the real-world deploy quite different ways for customers to express their special requirements and constraints, the job description file prepared for one grid may not suitable for another one. One thing often ignored is that there are no mature tools for debugging and measuring the behavior of grid applications. (Jiang and Yang, 2010)(Al-Ta'ee, El-Omari and Kasasbeh, 2013)

Generally speaking, since developers should know many details about the grid environment, developing grid applications is a complex task and places a heavy burden on application developers (Foster *et al.*, 2008)(Bardsiri and Amid, 2012)(Jiang and Yang, 2010). A more concerning thing related to GC is that grid has limited storage space and computing power while scientists try to reach up to hypothetically unlimited scalability that scales up and out dynamically in terms of capacity and functionalities (Malik *et al.*, 2014). Finally, to take away the said limitations and to support the mechanism of computing with some additional capabilities, the next major technology, namely UC, comes into action.

3.5 Utility Computing (UC)

From the time of coining GC, scientists and experts have increasingly extended all their efforts toward achieving the following radical changes:

- Significantly reduce the time needed to complete the original problem by using the unused processing power more effectively and maximizing the available resource elements (C.Vijaya and P.Srinivasa, 2016). This allows complex, data-intensive and high-computation operations to be picked out and manipulated (Manju and Sadia, 2017)(Kaur, 2015).
- Make remarkable achievements in speed and scale of the communication technologies, while they are relatively decreasing the data communication costs, cutting down the operating expenses, and increasing the reliability and productivity (Bardsiri and Amid, 2012)(Foster *et al.*, 2008)(Jiang and Yang, 2010)(Kaur, 2015).
- They try to make a paradigm shift in focus from an infrastructure that offers block storage and computing power to a new one that not only delivers abstract resources but also delivers innovative IT services in an economy-based norm. Going forward, they try to coin other associated technologies that allow computing power, data, and software as on-demand bases which switch grids to be used for business applications besides it is originally used for non-commercial scientific applications. (Manju and Sadia, 2017)(Kaur, 2015)(Bardsiri and Amid, 2012)(Foster *et al.*, 2008) (Jiang and Yang, 2010)
- Increase the resources' productivity and, among other advantages, to be more profitable (Manju and Sadia, 2017)(Kaur, 2015).

As discussed beforehand, coupled with the timely advances in communications and (Inter) network technologies, this has given birth to the "service-oriented" practice, as in UC which focuses mainly on providing key services, rather than technologies. It allows integrating applications out of discrete-loosely-coupled services where the failure of one service will not disrupt other services. It involves dynamical aggregation of heterogeneous computational resources, all of which are integrated together and present themselves to a client as a single intelligible pool of trusted computational resources. (Bardsiri and Amid, 2012)(Sommerville, 2015)(Smith, 2016) (C.Vijaya and P.Srinivasa, 2016)(Manju and Sadia, 2017)(Kaur, 2015)

Since the data communication costs are relatively reduced and computational resources are allowed to be dispersed across multiple sites over many countries or continents around the globe, some providers of UC tend to build their data centers in regions with the lowest overall costs for electrical energy, taxes, labor wages, real estate, etc. Thus, the concept of this trend is a flexible arrangement for the consolidation of both hardware and software resource elements where physical resources are shared remotely by a number of applications and users. Rather than an enterprise that can run its IT services in-house over its own resources, it can contract to an external UC service provider who can offer these resources in a subscribed manner. However, the enterprise itself pays only just for the using of these resource elements. This is like the Grid Technology where the customers can access these electronic services based on their requirements regardless of where these well-defined services are hosted or how they are delivered to them. The offered principal resources include, but are not limited to, data storage capacity, and the shared computing power of the virtual computing environments. (Manju and Sadia, 2017)(Kaur, 2015)(Bardsiri and Amid, 2012)(Foster *et al.*, 2008)

What is more, similar to any public utility company, UC also creates a new model of trade markets where an ever-growing number of resource owners might have plentiful opportunities to distribute or sell the excess computing capacities of their data centers, or even their home computers (Laverty, Wood and Turchek, 2014) (C.Vijaya and P.Srinivasa, 2016)(Manju and Sadia, 2017)(Kaur, 2015). The offered services are metered and charged to a wide range of customers just only for their active utilization of the IT infrastructure and the software resource elements rather than a flat rate (Manju and Sadia, 2017)(Kaur, 2015). For instance, while the mass storage capacity may be charged per GB/MB/TB of the active stored data, the virtual computing environments may be paid per hour/week/month of resource usage and with the exact data transfer (Foster *et al.*, 2008) (Bardsiri and Amid, 2012)(Sommerville, 2015).

Thus, this model is about coining new business models in which computational resource elements are packaged together and delivered as commodity services to a wide range of customers over the network or the Internet like delivering the other four-basic existing utilities (electricity, water, gas, and telephone) by the public service utilities (Assunçãoa *et al.*, 2015)(Jain, Sumit and Kumar, 2017)(Laverty, Wood and Turchek, 2014)(Essandoh, Osei and Kofi, 2014)(C.Vijaya and P.Srinivasa, 2016)(Ali *et al.*, 2015). To state a fact, this was exactly the Amazon's future vision in 2006 to become an IT utility company for serving the computing utility for the general community as the 5th basic utility, similar to any public utility company (Laverty, Wood and Turchek, 2014)(Jain, Sumit and Kumar, 2017). Less than five years, many large organizations, such as IBM, with its SmartCloud, and Oracle, with its OracleCloud, followed Amazon's smart innovative vision by introducing the next flaring technology, namely CC (Laverty, Wood and Turchek, 2014).

3.6 Cloud Computing

Along the journey toward “service-oriented architecture” or as formerly known “computing as a utility”, Cloud Computing was eventually evolved up where web services are hosted by some service providers and offered on a UC infrastructure (Bardsiri and Amid, 2012)(Sommerville, 2015)(Deitel Pau and Deitel, 2017)(C.Vijaya and P.Srinivasa, 2016). As soon as this new integrated technology is introduced for reducing costs and increasing efficiency of the firms, it directly finds its path right the way through the daily life by making network access to an infinite pool of on-demand shared computational resources (Dhabhai and Gupta, 2106)(Alessio et al., 2014)(Gholami, Daneshgar and Beydoun, 2017). These resources can be deployed briskly, configured easily, operated and managed in a seamless manner (Michael and Rajiv, 2012)(Edlund, 2012)(Filippi and McCarthy, 2012)(Tsz Lai, Trancong and Goh, 2012) (Sareen, 2013)(Naveen and Harpreet, 2013)(Subbiah, Muthukumaran and Ramkumar, 2013)(Gholami, Daneshgar and Beydoun, 2017). As a result, most software houses are striving to integrate their services as cloud-based in order to meet the competitive global market (Essandoh, Osei and Kofi, 2014)(Gholami, Daneshgar and Beydoun, 2017). These houses, over the last years, have expanded all their efforts on retaining all the perceived benefits of more complicated processes whilst presenting the customers with reliable secure interactive interfaces with which they can access their services easily by iteratively submit queries and see rapid responses (Foster *et al.*, 2008)(Bardsiri and Amid, 2012)(Sommerville, 2015). These interfaces are dubbed as application programming interfaces (APIs). However, the actual complex processes are taking place within a system, but the complex working details are simply abstracted from the user.

Since clouds make the development process much easier and more efficient, running tasks in clouds is much easier than grids. The software and hardware complexity of the platform is shielded from the developers, the resource reservation and configuration can be done by a few mouse clicks, and the constraints laid by grids on the running programs are fewer. Thus, the pains of application development are greatly eased so that the developers can focus their salient attention on their designing. Really, it is clear from the already stated statements that the actual working on these clouds is almost as simple as working on the local computers. (Foster *et al.*, 2008) (Bardsiri and Amid, 2012)(Kiranjot and Anjandeep, 2014)(Jiang and Yang, 2010)(Yuzhong and Lei, 2014)

As service providers of UC offer storage capacity and virtual computing servers based on demand, some people wrongly think of CC as virtual-shared servers available over the Internet; they narrowly define it as an updated version of UC (Subbiah, Muthukumaran and Ramkumar, 2013)(Sareen, 2013)(Han *et al.*, 2016). Others go very broad, arguing that Cloud Technology is a way for extending IT's existing capabilities by on the fly services without investing more in new infrastructure, upgrading to up-to-date versions, licensing new software, or training new personnel (Anne-Lucie *et al.*, 2017)(M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018). However, UC does not rely on CC and can be applied using any server environment (Bardsiri and Amid, 2012)(Smith, 2016)(Kaur, 2015)(Manju and Sadia, 2017). Unlike CC, UC can be used for smaller usage and smaller scale-needs (Bardsiri and Amid, 2012)(Smith, 2016)(Kaur, 2015)(Manju and Sadia, 2017). As such UC is a prominent choice for less demanding applications that has lower peak usage fluctuations (Bardsiri and Amid, 2012)(Smith, 2016)(Kaur, 2015)(Manju and Sadia, 2017).

New technologies, like clouds, introduce a new set of concepts, such as abstraction which is truly the most widely-used concept among them. Hidden complexity masked by sheer simplicity is really what it's formally called “Abstraction” which is one of the object-oriented (OO) programming concepts. It is the process of amplifying the important aspects of anything and ignoring or hiding the irrelevant aspects such as hiding the unwanted implementation details. It is the process of offering a wide range of key services to customers in terms that they understand. In effect, CC is actually a developing technology that evolved out of GC. Sure, both are similar in applying the concept of abstraction, they both use networks that abstract processing tasks. The visibility of the physical computing environment is only through interactive interfaces that facilitate the communication process for receiving inputs and providing outputs. Generally speaking, the ideal scenario is as follows: data exist on multiple servers, not needed details of network connections are hidden, the outputs are present, but how these outputs are computed is completely hidden from the end-users. (Tsz Lai, Trancong and Goh, 2012)(Foster *et al.*, 2008)(Pooyan, Ahmad and Pahl, 2013)(Han *et al.*, 2016)(Kiranjot and Anjandeep, 2014) (Yashodha Sambrani, 2016)(Gholami, Daneshgar and Beydoun, 2017)(Deitel Pau and Deitel, 2017)

Importantly, the level of abstraction is higher in the cloud than GC, it eliminates more details with lower network latency and higher bandwidth. To say the truth, CC is so named because a virtual cloud is often used to hide unwanted knowledge of inner workings; the end-users are none the wiser (Abdu *et al.*, 2017). Moreover, the real physical and virtual digital clouds have a correlated meaning with a high beautiful shelter that covers everything related to daily life (Bertolino, Nautiyal and Malik, 2017).

Although both the server-side of the grid and CC are nearly alike, their respective clients are different. Instead of a few clients running large grid jobs, thousands or millions of clients are serviced by the cloud (Bardsiri and Amid, 2012)(Deitel Pau and Deitel, 2017). Therefore, in order to achieve better system performance, CC has a large pool of highly available and scalable interconnected resource elements that have a higher automatic-scaling feature than GC (Foster *et al.*, 2008)(Bardsiri and Amid, 2012)(Deitel Pau and Deitel, 2017).

One thing to keep an eye on is that CC depends at most events on the use of the already existing computing technologies and, therefore, both grid and UC can be considered as implementations of CC. Ultimately, Cloud technology derives many best-known things from the Linux paradigm which is truly the most popular full-featured server operating system. One of them is, having one massive element for doing all things is not as efficient as having multiple elements, each excellent at doing one particular task. Another useful thing is that everything is stored as a file which it will be seen later in the section of CC virtualization where the actual hardware infrastructure is implemented as software-based files, typically called images or Virtual Machines (VMs) (Australian Government, 2013). Furthermore, with the concept of programmability, many physical hardware connections are eliminated and replaced with easy-configured software files (Rehman and Annapurna, 2017).

It is vitally important to mention that distributed systems, CC in specific, use the multilayered capabilities of Linux to overcome the heterogeneity (i.e. the degree of diversity) among the distinct components of the underlying layers (Kaur, 2015) where each underlying layer hides all the complexity and the lower-level details from the layer above in a more efficient manner (Kaur, 2015)(Alessio *et al.*, 2014).

4. CC Characteristic

Many researchers, practitioners, analysts, and designers have proposed definitions and explanations for CC. But, until now, CC has no formally accepted conceptualization which adequately addresses its reality; though it has not yet been sufficient standardized (Chowdhary and Rawat, 2013)(Alessio *et al.*, 2014)(Bertolino, Nautiyal and Malik, 2017). As a matter of fact, it is a rapidly evolving technology that overlaps with the other existing technologies (George Pallis, 2010). But in its simplest form, CC can be portrayed within this context of content outsourcing, data, and processes as virtual servers available over the Internet (Yashodha Sambrani, 2016)(Ali, 2016). The company that can provide and complement these activities as measured services is referred to as a cloud vendor (Bertolino, Nautiyal and Malik, 2017).

One of the acceptable broad definitions of CC that are widely accepted by the research community today was originally proposed by (Foster *et al.*, 2008) as:

"*A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet.*"

Another well-known academic definition is defined by (Hussein and Mousa, 2012) as:

"*Cloud computing is a large-scale parallel and distributed computing system. It consists of a collection of interconnected and virtualized computing resources that are managed to be one or more unified computing resources. Through Cloud Computing services are delivered on demand to the end-user over high speed Internet.*"

As a consequence, the National Institute of Standards and Technology (NIST) in (The National Institute of Standards and Technology (NIST), 2011) gives a basic and direct definition for CC as:

"*a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*".

A simple definition for the cloud of the term "*Cloud Computing*" itself is defined by (Sommerville, 2015) as:

"*a computing cloud is a huge number of linked computer systems that is shared by many users. Users do not buy software but pay according to how much the software is used or are given free access in return for watching adverts that are displayed on their screen*".

It is clear from these definitions that the philosophy behind CC is not so much about some new technologies, rather it is more about new economic strategies and operational modes for delivering electronically both hardware and software computational resources to customers as on-demand services that are always on (Laverty, Wood and Turchek, 2014)(Ali *et al.*, 2015). Based on this, the IT industry is changed considerably by adding flexibility to the ways of buying and selling IT (Assunçãoa *et al.*, 2015)(Ramzan and Alawairdhi, 2014).

In light of the above, CC is a multidisciplinary domain and can target various sets of lofty goals and objectives. In its simplest form, it can be considered massively as a cluster or clusters of hardware machines, referred to as a server, connected through the Internet to serve many customers effectively, efficiently, safely, and concurrently (i.e., in parallel) (Yashodha Sambrani, 2016)(Klement, 2017). It may be also considered as various computational resources in the form of a Cloud platform that offered as utilities on infrastructure hosted by major providers. These resources look to the user as close as a single computational resource but they are actually unified from multiple ones. However, this service model is different than the other online web services by not least of which are:

4.1 Abstracted Infrastructure

Broadly, CC architecture can be understood in an abstracted aspect as it comprises of two abstracted parts connected to each other via the Internet and maintains a higher degree of correlations: a frontend and a backend. It is the visible part that the clients see and interact with, it consists of the computing devices and the applications together with the interfaces used to access the cloud applications. The cloud itself is the backend that forms an in common framework for delivering these measured services. (Manju and Sadia, 2017)(Patel, Patel and Panchal, 2017)

To provide a high level of abstraction that eliminates or hides the unwanted details, the backend complexity is shielded from the frontend. This allows the users to focus their efforts on the application itself rather than on the underlying low-level operations. Therefore, they concentrate their selves on their high-level work rather than on the minutiae of how the low-level operations should be implemented for the specific version of the software. (Michael and Rajiv, 2012) (Edlund, 2012)(Filippi and McCarthy, 2012)(Tsz Lai, Trancong and Goh, 2012)(Taneja, Taneja and Chadha, 2012)(Foster *et al.*, 2008)(Sareen, 2013)(Naveen and Harpreet, 2013) (Chowdhary and Rawat, 2013)(Neves *et al.*, 2016)

From a conceptual standpoint, the CC stakeholders are built on three majorly groups of pillars: Service Providers (i.e. Vendors), Developers, and Customers (i.e. end-users). The first group is the actual providers of the CC software environments (Tsz Lai, Trancong and Goh, 2012)(Taneja, Taneja and Chadha, 2012)(Foster *et al.*, 2008)(Ali, 2016). While the first group breaks the hard-coded connections between the second group and the third group, the first two groups are considered as the supplier for the third group (Smith, 2016)(Ashdown and Kyte, 2015)(Ali, 2016). Even though different stakeholders have different perspectives for looking at the Cloud Computing, Cloud Computing encapsulates the relationships between these three groups of stakeholders on a consolidated basis by the use of electronic means (Tsz Lai, Trancong and Goh, 2012)(Ali, 2016).

With the objective of reaching a large wide-ranging base, members of the first group, namely service providers, acquire services from members of the second group, namely developers, and lease them out to members of the third group, namely end-users. Service providers should ensure the hosting of completely flexible and fully upgradeable environments that offer a wide scope of Efficiency, Portability, Scalability, Compatibility, Reliability, and all the other related properties (Chraibi et al., 2017)(Sharma, Singh and Kaur, 2016)(Bratterud, Happe and Duncan, 2017)(C.Vijaya and P.Srinivasa, 2016). Since a system using the services of another separated system is called a client (Solanki and Shaikh, 2014), the third group, end-users, together with their computing devices are formally referred to as Clients; they are categorized into three basic kinds (Abdu *et al.*, 2017)(Tsz Lai, Trancong and Goh, 2012)(Edlund, 2012)(Sareen, 2013)(Sommerville, 2015)(M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018):

- **Mobile Clients:** These hand-handled devices not only include personal digital assistants (PDAs), but also iPhones, Windows, and Android smartphones, or even any mobile device connected to the Internet as client hardware.
- **Thin Clients:** As they are typically designed to be especially small, they are just computers without internal memory or hard disk drives. Since they haven't their own OS, mass storage, or the ability to execute their own programs, they are serving as a frontend to other backend computational resources where the application processing is carried out on the server-side. Along with requiring continuous server communications, thin clients are also totally relying on the servers for handling the bulk of the data processing and they act as a simple terminal to the server. Therefore, all they do is just displaying the computational results.
- **Thick Clients:** also called fat clients, where some or all of the computational tasks are implemented on the client-side. Since they typically have their own OS, hard disk drives, and the ability to execute their own processing tasks, there is no need for constant communication with the servers and, therefore, the majority bulk of the computational workload occurs on the same client. Laptops, tablet devices, iPads, and notebooks are classified under this category.

Evidently, the primary interests of the cloud service providers (CSPs) revolve around providing more refined sophisticated services to support end-users (Ali *et al.*, 2015). But members of this third group, end-users, don't worry about the resource management overhead, or about the exact locations or the types of the resources their applications are running on (Tsz Lai, Trancong and Goh, 2012)(Jain, Sumit and Kumar, 2017)(Ali, 2016). Since their primary interest focus on easy-to-use resources, they don't care about the underlying technical details of the resources that are abstracted away from them by the other two groups (Aspen and Kaitlyn, 2017). Likewise, they may not know where their data are going to be kept in the cloud, if, when, or how it is backed up or restored again due to computer crashes or patches (Tsz Lai, Trancong and Goh, 2012). They only shift their data to the cloud without knowing or worrying about the other specific details (M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018). However, their perspectives for Clouds are merely as a norm for providing resources and services in a highly available fashion (Aspen and Kaitlyn, 2017) allowing them to perform a wide variety of tasks ranging from data generation and entering, going by data querying and retrieving, and ending up with all facets of data quality assurance and control. Sure, all that is simply provisioned by the CSP and visible to them remotely through a standard web browser with attractive well-designed Graphical User Interfaces (GUIs) that conform to the industry norms.(Tsz Lai, Trancong and Goh, 2012)(Sharma, Singh and Kaur, 2016)(Deitel Pau and Deitel, 2017)

Since the cloud subscribers (i.e. end-users) don't need any knowledge of how these computational resources work, communicate, or exist, they treat these allied computational resources like a black box whose has well-suited interfaces for receiving input data and providing related outputs. The needed behavior of this box can be achieved with very limited knowledge about using these interfaces. (Foster *et al.*, 2008)(Bardsiri and Amid, 2012)(Sommerville, 2015)(Jiang and Yang, 2010)(Deitel Pau and Deitel, 2017)

And, above all, cloud subscribers can use this service model like using electricity for running any electrical machine, such as television without realizing how it is being generated or from where that electricity is coming in (Essandoh, Osei and Kofi, 2014).

In that manner, the way of computing these outputs is completely hidden and the level of abstraction is raised as high as to be closer to the actual problem domain and to be at a distance from the lower-level details. For that reason, sometimes they are referred to as "non-technical" end-users. (Edlund, 2012)(Kiranjot and Anjandeep, 2014)(Ramzan and Alawairdhi, 2014)(Smith, 2016)

On the other hand, since service providers aim to relatively reduce the charges of using and maintaining their infrastructure, the resource utilization should be optimized and reduced to a minimum level while keeping their cloud-hosted services to the last group, end-users, in the desired quality (C.Vijaya and P.Srinivasa, 2016).

4.2 Broad Network Access

All of the CC servers are connected to a relatively high-speed broadband network which allows end-to-end data to flow not only over the Internet but also among different types of computing and storage elements (Ramzan and Alawairdhi, 2014)(Yunchuan *et al.*, 2014)(Yashodha Sambrani, 2016). This is done at the service level without the need for central coordination, hence the term "peer-to-peer" or "distributed architecture".

4.2.1 Resource Pooling and massive-scale Sharing

Not just the applications and system software are delivered by CC as measured services, but also the actual IT infrastructure that provides the functionality of these services (Naveen and Harpreet, 2013) (Bardsiri and Amid, 2012). To this aim, the different physical and virtual resources are dynamically pooled to serve a wide range of multiple preauthorized customers; these resources are dynamically assigned and reassigned from anywhere in the world, at any time there is Internet connectivity and communications authority to access the clouds using customized portals and built-in apps (Alessio *et al.*, 2014)(Ali, 2016).

This coherent trend ties together over the web a vast network of resources and integrates them with their underlying hardware component, OSs, and local resource management with more rock-solid information security as if they were from the same enterprise (Sareen, 2013)(Naveen and Harpreet, 2013)(Bardsiri and Amid, 2012) (Sommerville, 2015).

4.2.2 Heterogeneity & Geographical Scalability

By heterogeneity, a system is described that is made up of many distinct components that include different types of computer and mobile devices (Kaur, 2015)(Sommerville, 2015)(Boudi1 *et al.*, 2018).

The various resources (e.g., networks, servers, storage media, applications, sensors, scientific instruments, services, and interfaces) are in general heterogeneous and may be dispersed across networks on multiple geographic locations as distributed systems; they are interlinked and delivered as a set of well-defined services over the network apart from their physical locations or heterogeneous structure. Then customers are allowed to access them broadly as a shared pool of computational resources over global on-demand network access. Since data replication to a number of data centers distributed in different places across the world is possible, this characteristic is also referred to as geographical scalability or board-network-access. (Naveen and Harpreet, 2013)(Tsz Lai, Trancong and Goh, 2012)(Edlund, 2012)(Foster *et al.*, 2008)(Chowdhary and Rawat, 2013)(Sommerville, 2015) (Arora *et al.*, 2017)

4.2.3 Centralized Services

The different resources within the data centers of the cloud service providers (CSPs) are often centralized in few geographical locations and connected with relatively high-speed network connections (Edlund, 2012)(Tsz Lai, Trancong and Goh, 2012)(Foster *et al.*, 2008)(Naveen and Harpreet, 2013)(Subbiah, Muthukumaran and Ramkumar, 2013)(Chowdhary and Rawat, 2013) (Acharjya and Ahmed, 2016)(Akshatha and Manjunath, 2016)(Ali *et al.*, 2015).

4.2.4 Economy Based

Cloud computing has emerged out from boundless and never-dying diverse sources for ever-sooner and ever-low-priced computation (George Pallis, 2010). These sources are ranging from academic going to industrial and ending with commercial worlds (George Pallis, 2010). While GC delivers storage and/or compute resources, CC delivers more abstract resources and services via abstract interactive interfaces that can yield long-term cost benefits and improving productivity (Foster *et al.*, 2008)(Naveen and Harpreet, 2013)(Temkar, 2015)(Ashdown and Kyte, 2015)(Ashdown, Kyte and McCormack, 2018).

When the cloud service providers (CSPs) integrate their different cloud-related services for activation on the utility-pricing basis, it will be more cost-effective in all kinds of IT measures than most individuals or companies can provide on their own, when the entire cost of ownership is considered as a consistent measure (M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018)(Sareen, 2013)(Kuo *et al.*, 2014)(C.Vijaya and P.Srinivasa, 2016). For instance, there is no need like before for high-powered and high-priced computers to adopt cloud-hosted solutions (Pandey, Mishra and Tripathi, 2017); ordinary regular Internet-connected computers are enough. Beyond that, when CC is adopted and implemented on a large scale, it reduces or cuts the time and cost of hardware, software, maintaining infrastructure, electrical power consumption, air conditioning costs, floor space possession, human resources and management expenses, and other expenditures (M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018)(Kuo *et al.*, 2014)(C.Vijaya and P.Srinivasa, 2016)(Ghwanmeh, El-Omari and Khawaldeh, 2015)(Al-Ta'ee, El-Omari and Ghwanmeh, 2013).

Another important aspect is the fact that some cloud services are offered nearly free of charge when CSPs can generate revenue from these services using other avenues. As in the cases of Google, Yahoo, and many others, some cloud services may be used without charges or may be relatively charged with decreased costs when there are opportunities for generating revenues from some streams other than the users, such as from advertisement or user list sales. Since their services may be funded in exchange for being exposed to ads that are displayed on every user's monitor, they make them with free access without involving any direct payment from end-users (Al-Ta'ee, El-Omari and Kasasbeh, 2013).

4.2.5 On-demand Computing with Metered Billing

Since the web-based services are measured for business needs, CC entirely adopts the utility-based pricing scheme at the service level and, therefore, end-users as customers are only charged by their cloud service provider (CSP) precisely for their actual level of resource usage not for what they can access. For instance, they pay by the second or minute for the exact storage or bandwidth they actually use. Thus, due to this load balancing criteria, this characteristic is also referred to as "pay-as-you-go (PAYG)", "pay-as-you-use (PAYU)", "pay-as-needed (PAN)", or "pay-per-use (PPU)" mode where providers offer instant services as needed and customers are only billed for their actual level of consumption for the resources and workloads they use. Therefore, there is a need for a system of precise metric billing that drives profits up toward providers and prices down toward end-users. This pricing system of on-demand computing should necessarily include at least the three fundamental models: accounting, billing, and auditing. (Foster *et al.*, 2008)(Tsz Lai, Trancong and Goh, 2012)(Edlund, 2012)(Chraibi *et al.*, 2017)(Dhabhai and Gupta, 2106)(C.Vijaya and P.Srinivasa, 2016)

4.2.6 Robust Real-Time Monitoring

Since the data can be generated, reached, accessed, managed and shared remotely whenever required at the right time and in the right form, real-time or near-real-time monitoring is been presented to track and follow the status of all the related resources, applications and interactions (Foster *et al.*, 2008)(Bratterud, Happe and Duncan, 2017)(Beacham and Duncan, 2017)(Arora *et al.*, 2017)(C.Vijaya and P.Srinivasa, 2016). Besides that this fundamental requirement continuously monitors and measures what is happening with a system around the time, it also essential to log automatically the entire life-cycle of all the virtual instances as they are created, deployed, or deleted (Beacham and Duncan, 2017)(Anne-Lucie *et al.*, 2017)(Bratterud, Happe and Duncan, 2017).

4.2.7 Easy and Automatic Resource Deployment

This key characteristic is also known as on-demand self-service where the CC customers don't care about the complexities of the operating environment or the underlying technologies. As the complexity is hidden from them and the cloud-hosted services are defined according to their perspective, customers only choose their desired services according to their business needs and then these services are fixed up and configured automatically by the supplier of the service to suit these needs (Edlund, 2012)(Tsz Lai, Trancong and Goh, 2012)(Smith, 2016). Thus, within just a few seconds, the new deployed cloud-based application is provisioned new instances on an on-demand basis.

As a consequence, customers do not have to worry about how the data are actually stored and retrieved from the tables or storage space, nor about the required types and specifications of the resources, nor about hardware compatibility, nor about the complexities of direct hardware management or even about the physical states of their hardware (Tsz Lai, Trancong and Goh, 2012)(Edlund, 2012)(M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018). What is more, by moving data to centralized services via Clouds, end-users don't have to worry about, software running and upgrading, or unnecessary details (Naveen and Harpreet, 2013)(M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018).

4.2.8 Dynamic Workload Balancing

In reality, when there are many coupled working machines, some machines operate at heavy loads and others remain idle. This implies that some resources are over-utilized more than necessary and others are not utilized as optimally as they could. Besides that the number of consumed resources is further increased, this situation leisurely consumes power and causing a decline in system performance which leads to lower input/output throughput and a longer I/O response time. (Gumbi and Mnkandla, 2015)(Smith, 2016)(Kaur and Chana, 2015)(C.Vijaya and P.Srinivasa, 2016)

To overcome this problem, workload balancing between multiple resources and customers is required to be mapped to the associated cloud services. In effect, the virtualization technology that forms the heart of any cloud environment has the capabilities of maximizing resource utilization and minimizing the CPU idle time where the workload and resource utilization is distributed more evenly over the entire set of resources. (Gumbi and Mnkandla, 2015)(Smith, 2016)(Kaur and Chana, 2015)(C.Vijaya and P.Srinivasa, 2016)

4.2.9 Energy Savings

Underutilized resources may stay idle for a long time, leisurely withdrawing electrical power while doing nothing. On the contrary, over-utilized resources require more power for handling their workload that is beyond their capacities. As such, load balancing between underutilized and over-utilized resources is definitely essential. As a consequence, there is a need to balance peak loads across the different resources so that the workloads and the utilization of resources will be distributed more fairly over the entire set of resources. (Jain, Sumit and Kumar, 2017)(Kaur and Chana, 2015)(C.Vijaya and P.Srinivasa, 2016)(Ghwanmeh, El-Omari and Khawaldeh, 2015)(Al-Ta'ee, El-Omari and Ghwanmeh, 2013)

The energy consumption and the level of resource utilization are closely related to each other. Actually, workload balancing through virtualization leads to resource utilization, which, in turn, helps in reducing energy consumption. (Gumbi and Mnkandla, 2015)(Jain, Sumit and Kumar, 2017)(Kaur and Chana, 2015)(C.Vijaya and P.Srinivasa, 2016)(Ghwanmeh, El-Omari and Khawaldeh, 2015)(Al-Ta'ee, El-Omari and Ghwanmeh, 2013)

4.2.10 Self-Healing, Availability, and Reliability

In order to ensure continuous service, CC should support self-healing, referred to as self-recovering, which allow workloads to recover from many unavoidable hardware/software failures without disruption (Michael and Rajiv, 2012)(Edlund, 2012)(Filippi and McCarthy, 2012)(Tsz Lai, Trancong and Goh, 2012)(Taneja, Taneja and Chadha, 2012)(Foster *et al.*, 2008)(Sareen, 2013)(Naveen and Harpreet, 2013)(Chowdhary and Rawat, 2013)(Yashodha Sambrani, 2016)(Chandra and Neelanarayanan, 2017). Because virtual hardware does not fail as the physical hardware, CC can take advantage of virtual environments of being more reliable (Venkatachalapathy *et al.*, 2016)(Ali *et al.*, 2015)(Yunchuan *et al.*, 2014). In addition, CSPs have subject matter experts in different computing areas and available to support the computing cloud customers with regard to high availability (HA) and disaster recovery (DR) design and best practices (Naveen and Harpreet, 2013)(Ali *et al.*, 2015)(Yunchuan *et al.*, 2014).

Back-and-forth, fully redundant, physical and/or virtual, resources in the data center should be highly available to be used in case of the other resource failures. This includes electrical power supplies, cooling systems, network access equipment, and the other critical hardware infrastructure. Failure at one resource will not disrupt other resources where any disrupted resource is directly removed from the infrastructure and the system should be capable immediately to continue functioning in a swift manner with the other undisrupted resources. (Ali et al., 2015)

The data of all the cloud servers must be backed up consistently and, perhaps more importantly, should be stored remotely in different locations far away from the original main cloud (Yashodha Sambrani, 2016)(Ali *et al.*, 2015). This certainly helps in meeting business continuity requirements and what it's termed as "just-in-time" service availability (Yashodha Sambrani, 2016)(Naveen and Harpreet, 2013)(Ali *et al.*, 2015).

4.2.11 Abstract Accessibility across the Globe

The major shift in thinking about information operations is tied with the use of web-based hosting and the "service-oriented" practice. By this, Cloud applications based on web surfing has, therefore, become a wildly-popular practice among numerous millions of computer users. (Subbiah, Muthukumaran and Ramkumar, 2013)

4.2.12 Transparency

Physically dispersed resources should be accessed and seen as a single coherent system without regard to their physical locations or heterogeneous structure (Sommerville, 2015)(Arora *et al.*, 2017). By taking storage as an example on transparently, data transparently over multiple disks make them appear as a single-fast-large disk. So transparency, in this context, can be viewed as hiding the existence of the network and the simple data distribution over numerous places (Sommerville, 2015).

4.2.13 Scalability

By auto-scaling, the proposed cloud system should have significant capabilities to cope automatically with any new growing environment like facing the peak of computes and storages that are needed for a short time span rather than most of the time. Scalability takes into consideration not only the users' current needs but also their future growths. (Kiranjot and Anjandeep, 2014)(Assunçãoa *et al.*, 2015)(Sommerville, 2015)

For the sake of reflecting these changing conditions, scalability has two margins: vertical and horizontal. Each one of these two margins has two sub-margins. Figure 3 shows this auto-scaling. The first one, namely scale-up, is often based on the automatic addition of more computational power to the same single node and in this arrangement it could be the addition of more CPU's, disk space, or RAM to the existing computer. In contrast, scale-down is the automatic deletion of some computational power form an existing node. Within this context, scale-down versus scale-up schemes are referred to as vertical scaling. Thus, this norm of scaling is done through spreading the computational load balance over the resources of the same machine. (Kang and Lee, 2016)(Sareen, 2013)(Pandey, Mishra and Tripathi, 2017)(Sharma, Singh and Kaur, 2016)(Venkatachalapathy *et al.*, 2016) (VMware Inc., 2019)

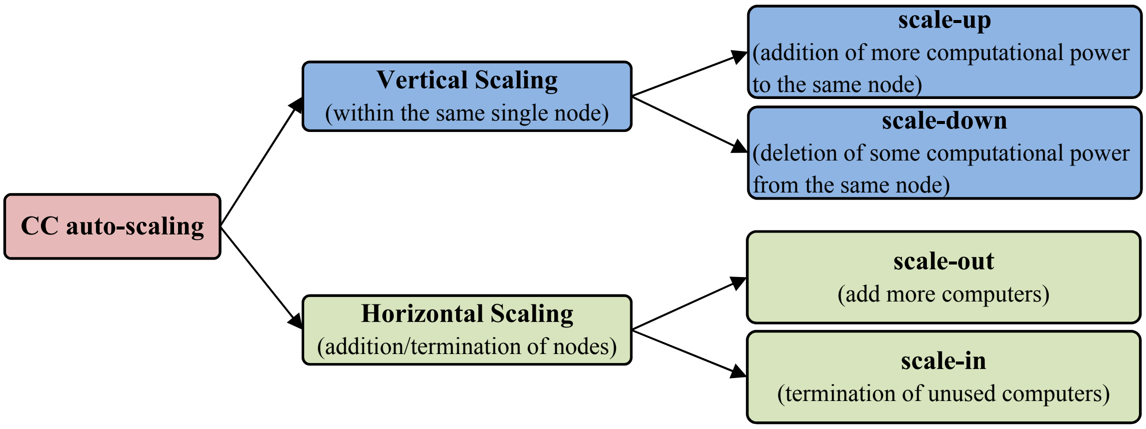

CC auto-scaling
Vertical Scaling
(within the same single node)
Horizontal Scaling
(addition/termination of nodes)
scale-up
(addition of more computational power to the same node)
scale-down
(deletion of some computational power from the same node)
scale-out
(add more computers)
scale-in
(termination of unused computers)

Figure 3. The CC auto-scaling

The other one, namely scale-out, is to add more nodes (i.e. computers) into the same pool of resources when the demand is increased. Each node contains only a part of the computational problem data. Unlike scale-out, scale-in is the automatic termination of unused existing nodes when the request is decreased. Within this context, scale-in versus scale-out is called horizontal scaling. While solving the computational problem of the horizontal scaling is related to more than one node, the solving in the vertical scaling resides on a single node. Since the scale-up is often limited by the upper-limit capacity of a single machine, the scale-out is often more dynamically to scale than the other margin. (Kang and Lee, 2016)(Sareen, 2013)(Pandey, Mishra and Tripathi, 2017)(Sharma, Singh and Kaur, 2016)(Venkatachalapathy *et al.*, 2016)(VMware Inc., 2019)

Since the cloud-like resources are of on-demand nature and they are dynamically adjustable according to the current demand, moving them to centralized services via clouds increases the systems' capabilities to cope with the customers' demands (Manju and Sadia, 2017)(C.Vijaya and P.Srinivasa, 2016). Accordingly, they are provided and scaled up or out as the computing needs increased and scaled down or in when no longer needed (Bratterud, Happe and Duncan, 2017)(Dhabhai and Gupta, 2106)(Manju and Sadia, 2017)(C.Vijaya and P.Srinivasa, 2016).

In practice, CC has a highly automatic-scaling feature that allows critical workloads to be deployed and scaled both dynamically and automatically based on the need at the point of time (Laverty, Wood and Turchek, 2014)(C.Vijaya and P.Srinivasa, 2016). For instance, the addition of more hardware units in this computing paradigm, like adding more hard disks, or increasing network bandwidth can be done easily (Tsz Lai, Trancong and Goh, 2012)(Edlund, 2012). Furthermore, to avoid the very serious problem of "Potential Performance Degradation", CC allows the same resource type to be accessed and shared remotely by an ever-growing number of end-users from different geographical locations with minimal effort (Naveen and Harpreet, 2013)(Edlund, 2012).

4.2.14 Agility

Cloud-hosted resources can be easily accessed by the different cloud-like services and then instantly released back to the pool in high-agile fashion. The users can choose their desired cloud-related services from a catalog or a menu encapsulating a number of underlying supported services. Then, they can use them easily and quickly over the Internet using their web browsers without having to know internally how these operations are being carried out. To come up with this focal point, end-users can easily initiate, utilize, and terminate their cloud-related server instances properly and only whenever needed, hence the term "agility". (Yashodha Sambrani, 2016)(Anne-Lucie *et al.*, 2017)(Hussein and Mousa, 2012)(Hussein and Mousa, 2012)

4.2.15 Dependable Access

One of the customers' core objectives is a need for dependable service. They often require assurances from the Cloud Service Providers (CSPs) that their web-enabled services are delivered under an established Quality of Service (QoS). This implies that these customers will receive their selected services with expectable, continual, and high levels of performance. (Sommerville, 2015)(Kiranjot and Anjandeep, 2014)(Chraibi *et al.*, 2017)(Patel, Patel and Panchal, 2017)

4.2.16 Fault Tolerance

If the CC architecture is designed correctly by having a well-suited-adaptive policy, it can be dynamically tolerant of some hardware and software failures (Sommerville, 2015)(Kiranjot and Anjandeep, 2014). This does not imply that resources must be absolutely available all over the time but it can tailor its behavior to extract the maximum performance from the remaining available resources after excluding the failed ones (Kiranjot and Anjandeep, 2014).

In summary, the above-discussed characteristics of CC are what make it a cost-effective, robust, easy to use and scale environment for enterprises to build their practical solutions and for end-users to consume such solutions (Ali, 2016). What's more, an environment is called as "cloudified" if it satisfies the above cloud-related principles (Edlund, 2012).

5. CC Virtualization

The traditional using of any system entails running a single OS on one physical server (Pandey, Mishra and Tripathi, 2017)(Chandra and Neelanarayanan, 2017)(Bratterud, Happe and Duncan, 2017). Figure 4 demonstrates the classical way of hosting applications and their required data storage. Within this context, the physical resources are obviously operated and controlled only by the OS itself.

But Virtualization, virtual reality rather than actual reality, supports abstracting of an enormous amount of different resources from a large-scale shared pool of resources by virtual reforming the close relationship between the underlying hardware infrastructure and the OS (Tsz Lai, Trancong and Goh, 2012)(Naveen and Harpreet, 2013)(Daylami, 2015)(Bardsiri and Amid, 2012)(Kaur and Chana, 2015) (Klement, 2017)(Arora *et al.*, 2017)(VMware Inc., 2019). It can be presented as a virtual framework to simulate and duplicate dedicated physical hardware, such as mass storage and network devices, by software-based instances (Yunchuan *et al.*, 2014). It can also be portrayed as a synonym for the capability of making a virtual machine (VM) behave in the same manner as though it was a real-dedicated physical machine (PM) (Naveen and Harpreet, 2013)(Daylami, 2015)(Bardsiri and Amid, 2012)(Klement, 2017)(Maruf and Albert Y., 2017) (VMware Inc., 2019). As a result, the same physical hardware resource can be logically subdivided into many VMs, hence the term "VM" (Jiang, 2018)(Chandra and Neelanarayanan, 2017). Dating back to the early 1970s, VMs are widely used as an alternative to the simulation discipline (Jiang, 2018).

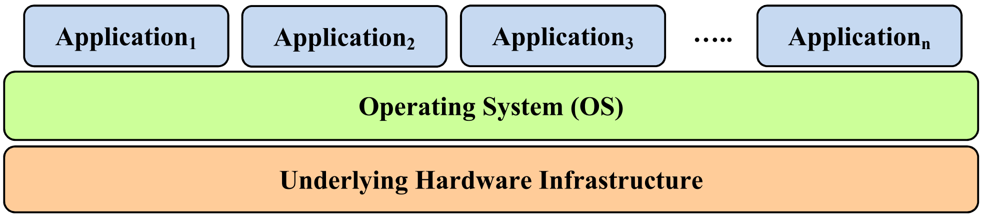


Figure 4. The classical architecture of computer

Besides that each PM on which the VMs are executed is referred to as a host, each VM with its own environment is called a virtualized guest (Klement, 2017)(Pandey, Mishra and Tripathi, 2017)(Sharma, Singh and Kaur, 2016)(VMware Inc., 2019)(Jiang, 2018). Multiple isolated VMs might be created dynamically on the same server, then they are been running simultaneously over a single physical PM (Naveen and Harpreet, 2013)(Daylami, 2015)(Bardsiri and Amid, 2012)(Klement, 2017)(Maruf and Albert Y., 2017)(VMware Inc., 2019)(Pandey, Mishra and Tripathi, 2017). These VMs are working so that they are completely isolated from each other's (Klement, 2017)(Pandey, Mishra and Tripathi, 2017)(Jiang, 2018). Since they are logically isolated, a failure of any VM does not affect others (Jiang, 2018).

Each beaked up VM is emulated by the software to appear as a tightly isolated container or view that encloses particular hardware, OS, network functions, and a stack of applications (Jiang, 2018). In practice, each beaked-up VM is completely independent and can be represented and stored as a single file on the actual PM; this file is normally referred to as an image. When this file is executed by the VM application appears to the user as an actual PM. Going forward, the entire state of each VM can itself be encapsulated as easily as a file that can be backed and moved from one machine to another and, therefore, each virtual machine can be identified easily. (Sareen, 2013)(Pandey, Mishra and Tripathi, 2017)(Sharma, Singh and Kaur, 2016) (Venkatachalapathy *et al.*, 2016)(Arora *et al.*, 2017)(VMware Inc., 2019)

Every VM image is viewed dynamically as having its own separate focus according to the element it is used on (Yashodha Sambrani, 2016)(Jiang, 2018). Accordingly, it can be started or stopped as needed and support many types of OSs, file systems, and applications (Laverty, Wood and Turchek, 2014)(Sareen, 2013)(Pandey, Mishra and Tripathi, 2017)(Sharma, Singh and Kaur, 2016)(Venkatachalapathy *et al.*, 2016)(Botta, Alessio, De Donato and Persico, Antonio, 2016). As it is illustrated in Figure 5, this is often done by a particular dynamic tool called a "hypervisor" which is also termed as a "Virtual Machine Monitor (VMM)" (Laverty, Wood and Turchek, 2014)(Pandey, Mishra and Tripathi, 2017)(Venkatachalapathy *et al.*, 2016) (C.Vijaya and P.Srinivasa, 2016). This tool work as a thin layer of software used to create and operate many concurrently VMs on the same underlying physical host (VMware Inc., 2019)(Jiang, 2018). It also bridges and aligns all the potential differences between the physical and virtual domains to appear as if it were a real domain (Boudi1 *et al.*, 2018). Since each multiplexed VM has its own configuration with an isolated environment, the hypervisor allocates the associated required computational resources for every VM hosted on the PM (Laverty, Wood and Turchek, 2014)(Pandey, Mishra and Tripathi, 2017)(Venkatachalapathy *et al.*, 2016) (C.Vijaya and P.Srinivasa, 2016)(VMware Inc., 2019).

Figure 5 also shows that every VM can run its particular OS, regardless of the original PM's operating system, and a full range of applications autonomously of the original PM (Han *et al.*, 2016)(Gumbi and Mnkandla, 2015)(Bardsiri and Amid, 2012)(Kaur and Chana, 2015). This figure also illustrates that each host computer (i.e. PM) can support multiple isolated virtualized guests (i.e. VMs) whereas each virtual guest (i.e. VM) has its own particular virtualized resources, dedicated user-space instance, and root file system (Pandey, Mishra and Tripathi, 2017). It is clear that each beaked up VM has an isolated guest OS that is executed inside (Jiang, 2018)(Chandra and Neelanarayanan, 2017)(Pandey, Mishra and Tripathi, 2017). So, the primary virtualized PM might concurrently host multiple applications and multiple OSs by encapsulating them as many VMs that are isolated from each other (Han *et al.*, 2016)(VMware Inc., 2019). Thus, instead of raw hardware infrastructure, the capabilities of the service can be highly extended and accessed over the broad network through different computing devices (Mobile, Thin, or Thick Clients) (Tsz Lai, Trancong and Goh, 2012)(Edlund, 2012)(Sareen, 2013)(Han *et al.*, 2016).

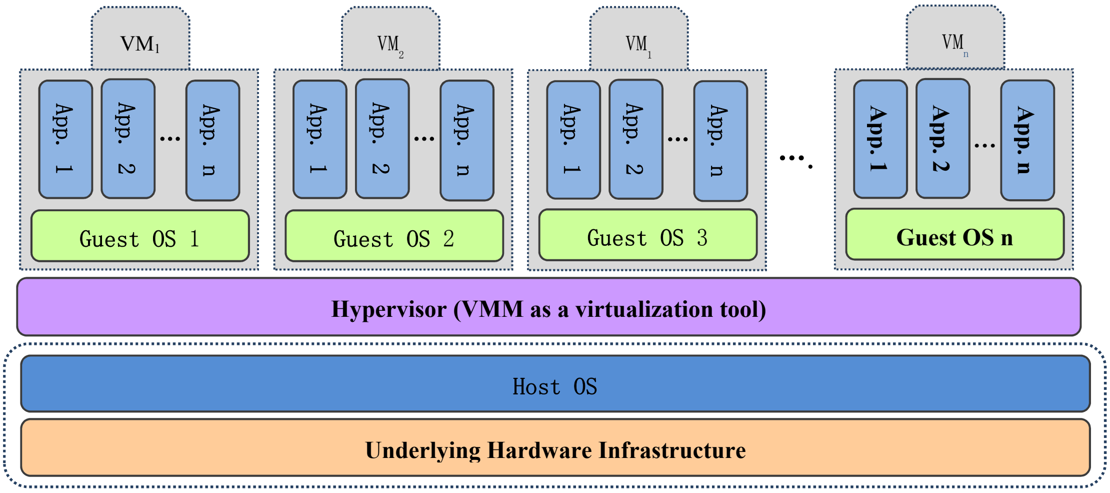


Figure 5. Virtualized Stack Architecture

The virtualization from the perspective of end-users allows them to get the actual feeling of using the resources as though they were working on their own computer systems regardless of their physical locations or heterogeneous structure (Vikram and Bhatia, 2016)(Arora *et al.*, 2017). They can create their own “virtual computer” with their desired application software and systems software without they are constrained with any OS such as Windows, Linux, or Mac (Klement, 2017). For instance, a user could set up a Windows-based virtual machine but run it inside a Mac machine or this user may run the Mac Operating system inside Windows (Klement, 2017). While this key characteristic significantly improves the compatibility of document formats, it is also used to run the preexisting legacy systems that need special old-fashion environments (Gholami, Daneshgar and Beydoun, 2017)(Jiang, 2018)(Prajkta and Keole, 2012).

In essence, every single PM is logically divided into a bundle of multiple logical VMs. Then, each activated VM acts as if it were a real physical computer with its own elements, such as CPU, memory, storage media, and bandwidth. Obviously, the sum of all the required resources within their elements should not be beyond the capacity of the hosted PM. (Han *et al.*, 2016)(C.Vijaya and P.Srinivasa, 2016)

Not just that, the virtualization term might be used much more generally in the long-run to nearly relate any aspect or portion of a system (Jiang, 2018). This major characteristic is broadly used in the cloud paradigm to consolidate multiple individual services and provide them as one virtualized service that runs in different geographically distributed locations and across different countries. This allows serving multiple customers by the same array of physical devices which can further result in increasing overall resource utilization with more electrical power savings and lesser heat dissipation (Gumbi and Mnkandla, 2015)(Bardsiri and Amid, 2012)(Kaur and Chana, 2015)(C.Vijaya and P.Srinivasa, 2016)(VMware Inc., 2019)(Klement, 2017).

Generally speaking, virtualization is facilitated with the help and support of the advent of multicore architectures where a single processor is composed of multiple cores. In other words, multicore architecture is extensively behind the task of running multiple OSs on a real PM. Moving forward with the dual implementation of the multicore architectures and virtualization techniques pave the way for further parallel processing development (Gumbi and Mnkandla, 2015)(Kaur and Chana, 2015)(C.Vijaya and P.Srinivasa, 2016).

Likewise, since these VMs can be easily migrated from one node onto the next, CC has a robust failure management process (Bardsiri and Amid, 2012). Furthermore, users can benefit from the ability to fully customize the OS to meet their specific needs. To sum up, Table 1 tries to highlight some of these key properties that are associated with these VMs.

To tell the truth, most virtualized applications show a performance reduction of the entire system than that originally running under the native systems. This is due to the times that are taken by the conversion processes from virtual to real that is performed by the hypervisor layer. (Marques *et al.*, 2018)(Boudi1 *et al.*, 2018)

Table 1. Key properties of VMs (VMware Inc., 2019)(Sareen, 2013)(Klement, 2017)(Pandey, Mishra and Tripathi, 2017)(Sharma, Singh and Kaur, 2016)(Venkatachalapathy *et al.*, 2016)(Arora *et al.*, 2017)(Jiang, 2018)

Property	*Why*
Multiplexing (one-to-many relationship)	Every single PM can enable multiple applications with different OSs to be run simultaneously. This highlights that the one-to-one relationship between hardware and the OS is no longer needed.
Encapsulation	Each beaked-up VM encapsulates internally a whole machine. VMs are actually manipulated as special files.
Partitioning	Since several VMs can be associated with just a single PM, the different system resources are logically divided among these VMs.
Mobility	Since VMs are eventually software files, any VM can be migrated without difficulty to another PM just by with copy-and-paste or drag-and-drop. Not only that, but live migration is also possible to relocate the currently running VMs from one PM to a different one.
Complete Isolation	The virtualization layer isolates and hides the heterogeneity of the hardware layer (i.e. the degree of diversity). At the same time, the OS of the PM is separated from each one of the VMs. Not just that, every guest OS of each VM is also logically isolated from the other OSs and, consecutively, the different input/output operations of the different VMs aren't affecting each other.

6. Classification of Virtualization Models

There is certainly a wide group of models proposed in the today's growing field of virtualization which makes it difficult to make a decision for the most suitable model selection, especially that most of them provide a convenient style to implement (Klement, 2017)(Pandey, Mishra and Tripathi, 2017)(Plauth, Feinbube and Polze, 2017). Depending on user-related and technical-related aspects, each model has its own specification (Klement, 2017)(Chandra and Neelanarayanan, 2017)(Plauth, Feinbube and Polze, 2017) and, therefore, these techniques can be classified into four overlapped categories that Figure 6 illustrates. To state a truth, the first two categories are based on the architectural design of OS (Pandey, Mishra and Tripathi, 2017) and both are generally called OS virtualization (Pandey, Mishra and Tripathi, 2017)(Chandra and Neelanarayanan, 2017). It is worth noticing that machine emulator, or simply emulation, is different than virtualization in the sense that emulation is an OS property that tries to transform the behavior of some peripheral devices into software-based programs. However, emulation is slower than the hypervisor virtualization technique.

The greatest noteworthy challenges in virtualization mechanisms are the decline in system performance that may lead to some losses in real-time features, which mostly result from controlling the so complicated shared resources (Jiang, 2018). In favor of this, academics, industrialists, and other practitioners are looking for a virtualization model which effectively empowers the following key capabilities (Jiang, 2018)(Marques *et al.*, 2018)(Boudi1 *et al.*, 2018)(Pandey, Mishra and Tripathi, 2017):

- Improved service scalability, reliability as well as easy accessibility.
- Improved real-time features related to predictability and timing accuracy.
- The reduced unwanted overhead caused by conventional software.
- Better system performance of the whole system. Operations that may lead to great performance-loads, extra system overhead, or longer response time are minimized or replaced by relatively shorter ones.
- Better scheduling and prioritization so that the right resources, no matter software or hardware, are assigned at the right time to the right tasks.
- Perfect isolation that enables isolated restricted device access and, in turn, more secure access.

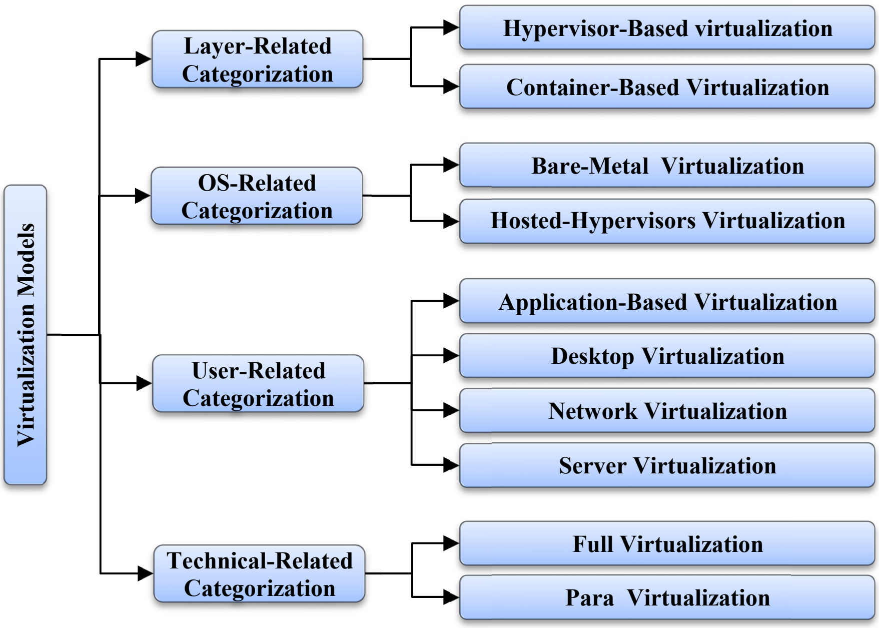

Virtualization Models
Layer-Related Categorization
OS-Related Categorization
User-Related Categorization
Technical-Related Categorization
Hypervisor-Based virtualization
Container-Based Virtualization
Bare-Metal Virtualization
Hosted-Hypervisors Virtualization
Application-Based Virtualization
Desktop Virtualization
Network Virtualization
Server Virtualization
Full Virtualization
Para Virtualization

Figure 6. Categories of Virtualization Technologies

6.1 First Virtualization Category

The first categorization of virtualization models is also referred to as OS-based virtualization (Pandey, Mishra and Tripathi, 2017). Depending on the existence of a dedicated layer for the hypervisor, this classification encloses the following two basic categories:

- **Container-Based Virtualization:** The classical layer of Virtualization decouples VMs from their host using the VMM. But, Linux operating systems do not support using hypervisor for the management mechanism of VMs and, therefore, this virtualization model is widely used as an alternative. (Marques *et al.*, 2018)(Boudi1 *et al.*, 2018) (Chandra and Neelanarayanan, 2017)(Pandey, Mishra and Tripathi, 2017)(Chandra and Neelanarayanan, 2017)

  It is important to realize that most complicated and critical tasks related to process and resource control, like scheduling and prioritization, are definitely done by the most important portion of the OS that is known as Linux kernel (Jiang, 2018)(Marques *et al.*, 2018)(Boudi1 *et al.*, 2018)(Pandey, Mishra and Tripathi, 2017). This includes, but is not limited to the different contentions on the input/output operations and the scheduling process between the applications of the VMs (Jiang, 2018)(Marques *et al.*, 2018)(Boudi1 *et al.*, 2018). To this aim, the virtualization process of this model is already contained within the same Linux OSs and implemented as a service at the host kernel-level (Marques *et al.*, 2018)(Boudi1 *et al.*, 2018)(Chandra and Neelanarayanan, 2017). Each Linux container, no matter the type or the architecture, is set as an isolated-operating framework which has its particular hosting space, processes, libraries, and all the essential components that are required to run and interoperable the desired software (Marques *et al.*, 2018)(Boudi1 *et al.*, 2018)(Chandra and Neelanarayanan, 2017)(Plauth, Feinbube and Polze, 2017) (Pandey, Mishra and Tripathi, 2017). While the OS kernel itself is shared by all the VMs, each beaked-up VM instance has its own level of abstraction with its own OS that has also its own root file system (Pandey, Mishra and Tripathi, 2017). Linux Containers (LXC) (Marques *et al.*, 2018) and Docker containers (Boudi1 *et al.*, 2018) are examples of this kind of virtualization.

- **Hypervisor-based Virtualization:** since it presents an additional dedicated virtual layer that separates the OS layer and the application layer, this virtualized model is also referred to as Layer-Based Virtualization. As it is already shown by Figure 5, a dedicated VMM layer is added to the abstracted-hierarchy and a set of virtual resources formulate another level of abstraction on the top of this layer. This hierarchical separation isolates and hides the physical resources from the upper layers and replaces them by a set of virtual resources. On the other side, this separation also isolates the set of applications of the different VMs from each other and allows these applications to function rightly. (Jiang, 2018)(Chandra and Neelanarayanan, 2017)(Plauth, Feinbube and Polze, 2017)(Pandey, Mishra and Tripathi, 2017)

Since the Linux kernel itself performs the managing process directly on the inside without the need for the presence of a particular virtualized layer, the virtualization of the first model, namely container-based virtualization, has relatively the highest performance among the other models in the sense that it finds quality virtualization in relatively a short running time that may be close to the native OSs (Marques *et al.*, 2018)(Boudi1 *et al.*, 2018)(Chandra and Neelanarayanan, 2017)(Plauth, Feinbube and Polze, 2017). Beyond that, this branch of virtualization generally offers a number of salient benefits in terms of security, latency and traffic reduction that make it as one of the most efficient virtualization models, such as:

- As there is no hypervisor layer as in the case of the classical virtualizations, the unwanted overhead is reduced and the overall-potential performance is somewhat improved in terms of input/output throughput and response time (Marques *et al.*, 2018)(Boudi1 *et al.*, 2018)(Chandra and Neelanarayanan, 2017)(Jiang, 2018).
- The activation or deactivation of containers can be more rapidly than the traditional VMs. Thus, virtualized instances are created, initialized, operated, functioned more rapidly compared with the classical hypervisor-based environments (Marques *et al.*, 2018) (Boudi1 *et al.*, 2018)(Pandey, Mishra and Tripathi, 2017).
- In the conventional virtualization system, when a hypervisor-based application running within a VM issues an input or/and output request, it is first handled and directed to a specified virtual driver of the same VM. After interpreting that request, the interrelated driver issues an interrelated request to the stated device as a response to the original request. As a result, the underlining VMM layer carries out that request. (Marques *et al.*, 2018)(Jiang, 2018)(Chandra and Neelanarayanan, 2017)(Pandey, Mishra and Tripathi, 2017)

  As an alternative of this tedious indirection and interposition which causes additional processor overhead, most container-based devices are plugged and played directly within the OS kernel without any need for low-layer drivers (Marques *et al.*, 2018)(Jiang, 2018) (Chandra and Neelanarayanan, 2017)(Pandey, Mishra and Tripathi, 2017).
- As this model is deeply counting on the underlying kernel in providing more secure and virtualized functions, this virtualization model is more secure compared to the other model. (Boudi1 *et al.*, 2018)(Pandey, Mishra and Tripathi, 2017). In essence, each virtualized instance is isolated from the other instances by the necessary secured functions that are presented by the kernel itself (Marques *et al.*, 2018)(Boudi1 *et al.*, 2018)(Chandra and Neelanarayanan, 2017)(Pandey, Mishra and Tripathi, 2017).

In comparison with the previous model, the second layer-based model comes, among others, with the following potential drawbacks (Marques *et al.*, 2018)(Jiang, 2018)(Chandra and Neelanarayanan, 2017)(Plauth, Feinbube and Polze, 2017)(Pandey, Mishra and Tripathi, 2017):

- Significant software overhead: Handling, scheduling, prioritization, running, and other complicated resource management related to shared resources cause some extra performance-loads in terms of both software and hardware reaction time. Unfortunately, this unwanted overhead effects the overall-potential performance and leads to lower input/output throughput and longer I/O response time.

  On the other hand, the physical resources related to the container-based virtualization are directly operated and controlled by the kernel itself rather than the VMM. It does not allow these VMs to share resources among themselves regardless if the abstracted-assigned resources are idle, unutilized, or even if the same VM is broken down.
- Despite that, the VMM layer of the Layer-Based virtualization presents a considerable role in usage and security, the container-based virtualization address a more mature role. This is evolving from the fact that this virtualization model relies upon the kernel itself in order to harden the security.

6.2 Second Virtualization Category

While the first categorization depends on whether the VMM exists as a separate layer on not, the second categorization depends on whether the VMM takes place either as a ring over of the host OS or directly over the hardware. The virtualized models are categorized into the following two basic categories regarding the hypervisor:

- **Hosted-Hypervisors virtualization:** in this virtualized model, both the underlying hardware and the OS of the same PM (i.e. the host) are required to be virtualized. As it is already illustrated in Figure 5, the hypervisor makes a thin layer of software that exists on the top of the existing OS of the underlying hardware. After installing and running the virtualized software, users of this model are being able to use multiple VMs in parallel on the same existing desktop OS. So, compared to other solutions, this virtualized model is more proper for personal computing than the other one. (Jiang, 2018)(Pandey, Mishra and Tripathi, 2017)(Plauth, Feinbube and Polze, 2017)

- **Bare-Metal virtualization:** in this virtualized model, the hypervisor makes a thin layer of software that exists directly above the underlying hardware of the same PM so that the same IT infrastructure is virtualized independently of the host OS. Therefore, no hosted OS is essential to make a virtualized environment. This model is simply demonstrated in Figure 7; it is also called "native hypervisor". Typical examples of this model are Xen and VMware vSphere®. (VMware Inc., 2019)(Jiang, 2018)

  Because the VMs of this virtualized model run directly on the top of the underlying hardware component rather than on the top of a host OS, these VMs may allocate computational resources better than the other style and may consequently yield a notable reduction in software overhead which significantly increase the entire system performance (Jiang, 2018)(Pandey, Mishra and Tripathi, 2017)(Plauth, Feinbube and Polze, 2017).

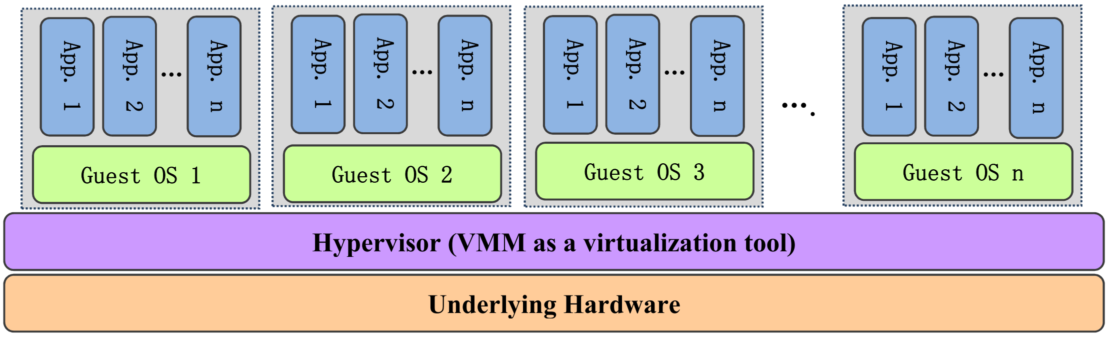


Figure 7. Virtualized Stack Architecture

6.3 Third Virtualization Category

The third categorization follows a more user-related approach and includes the following categories:

- **Application-Based Virtualization:** In this model, any application is encapsulated from the running OS and provided with a separate running environment through an application virtualization layer. Although this model is found to be more memory consuming than other models, it is justified for the benefits of the abstraction of host OS by the application virtualization layer (Pandey, Mishra and Tripathi, 2017).
- **Desktop Virtualization**: it is sometimes called a workstation virtualization (Klement, 2017). It is in the form of common complete hardware like a desktop, notebook, or a mobile touch device (Klement, 2017). Regardless of the underlying OS, this truly provides developers with the ability to run various OSs on only one PM (Sharma, Singh and Kaur, 2016)(Maruf and Albert Y., 2017). It may be utilized to make a special complete virtual environment for performing the testing mechanism that looks as if it were the same native environment (Klement, 2017)(Sharma, Singh and Kaur, 2016).
- **Network Virtualization:** Scientists, researchers, and experts paid a remarkable attention towards using the concept of programmability approaches in inventing the software-defined networking (SDN). In SDN the adoption of networks' virtualization is carried out by replacing many hardware connections with software models (Rehman and Annapurna, 2017). So, the physical network is emulated to create a virtual network that is actually viewed as if it were the original one. This includes, but is not limited to: routers, switches, firewalls, ports. Moreover, this model also involves networking services such as load balancers (VMware Inc., 2019).
- **Server Virtualization**: this virtualized model is also formally referred to as infrastructural virtualization or hardware virtualization (Klement, 2017). It involves simulating the hardware on a server to create many virtual machine instances (VMs) where each computer-generated VM has its own existence with its virtual CPU, memory, disk space, input/output, and network devices (Klement, 2017)(Sharma, Singh and Kaur, 2016)(VMware Inc., 2019). It is merely in the form of special hardware, server, disk arrays, and/or hypervisor tools (Klement, 2017)(VMware Inc., 2019).

  From another perspective, many servers are virtualized from a single physical server where each beaked up VM has its own independent OS and capable of running its own dedicated applications on top of that OS (Sharma, Singh and Kaur, 2016)(VMware Inc., 2019). As a result, the actual one-to-one relationship between hardware and the OS is no longer required (VMware Inc., 2019).

6.4 Fourth Virtualization Category

Another alternative to classifying virtualization is related to the full- versus para-virtualizations. Within this context, the fourth categorization follows a more technical aspect and, therefore, it is also formally referred to as hardware virtualization. The virtualized models of this category fall into two broad categories:

- **Full-Virtualization**: as the name suggests, the underlying hardware is completely simulated to allow the different applications that rely on the OS of the virtualized guest to be run without making any essential adaptations (Klement, 2017)(Sameera and Iraqi, 2017). The most leading examples of this model are "Kernel-based Virtual Machine (KVM)", Virtual-Box, "Microsoft Virtual PC", and "VMWare Workstation" (Jiang, 2018).
- **Para-Virtualization:** In this virtualized model, the underlying hardware component is not completely simulated, and some operations are executed directly on the guest hardware. Since only some hardware is simulated under this virtualized model, different applications need to be adapted before being launched on their virtualized guest OSs (Klement, 2017)(Sameera and Iraqi, 2017). Xen, "VMWare ESX" are broad examples of this model (Jiang, 2018).

6.5 How to Choose the Right Virtualization Model

Many models of virtualization are available; each one has its own specifications and requirements. The decision for a particular model selection is no longer an easy task; the followings are, among others, the critical factors which influence people's decisions in choosing one of them:

- The memory limits that are granted for VMs (Pandey, Mishra and Tripathi, 2017).
- File system isolation (Pandey, Mishra and Tripathi, 2017).
- Does the selected virtualization model support the using of USBs (Pandey, Mishra and Tripathi, 2017)?
- With regards to how the software of virtualization is used and distributed, the virtualization models are either open-source (a.k.a. freeware-licensed) or closed-source software (a.k.a. proprietary-based licensed) models. Furthermore, while many of the proprietary-based software are used as commercial software, the freeware-licensed software is generally free and originally grew out from academic and non-profit research institutes. However, the fact that software is free from any accessing fees does not necessarily state that it must be freeware-licensed software and not closed-source software. This is because freeware-licensed software is also licensed. (Pandey, Mishra and Tripathi, 2017)(Chraibi *et al.*, 2017)(Solanki and Shaikh, 2014)(K.Pranathi, 2016)(Prajkta and Keole, 2012)
- Nested virtualization support can be achieved when the VM has a further capability to run another hypervisor on the top of a preexisting one or inside another VM (Pandey, Mishra and Tripathi, 2017)(Chandra and Neelanarayanan, 2017). However, Not all virtualization models support nested virtualization (Pandey, Mishra and Tripathi, 2017).
- How the already presented functionalities of the input/output legacy devices are abstracted and mapped to a computer-generated of reality. This often addresses the following important concerns: Do the selected virtualization model map and abstract all the already-existing legacy input/output devices? Do all the old-style drivers have direct access interfaces? Is the selected virtualization model easy to use? (Jiang, 2018)(Prajkta and Keole, 2012).

To get around these stated factors, Table 2 is used to cover some of them (Pandey, Mishra and Tripathi, 2017)(Kang and Lee, 2016). It is an obvious rule that the total virtual CPU loads of all VMs should be within the total actual capacities of all PMs (C.Vijaya and P.Srinivasa, 2016). In light of the above, Figure 8 illustrates the abstraction mechanism of the VM.

Finally, it has been gleaned from the discussion of this subsection that the virtualization categories are overlapped with each other and some virtualization models fall into two or more categories. For instance, "VMWare MVP" are considered as hosted and para-virtualization solutions at the same time (Jiang, 2018)(VMware Inc., 2019).

Table 2. A comparison between Some Virtualization Models

Virtualization Models	*OS*	*Open-source*	*Nested virtualization*
QEMU	Linux, Windows, Mac OS X, UNIX	×	✓
Sandboxie	Windows	×	×
Spoon	Windows	×	×
VirtualBox	Windows, Linux	×	×
VMware ESXi	Windows	×	✓
VMware Fusion	MAC OS X	×	×
VMware ThinApp	Windows	×	×
OpenStack	Ubuntu server 64 bit	✓	✓
PVAMU	Windows, Linux, Mac OS, iOS, Android	✓	✓
Hyper-V	Windows	✓	✓
Microsoft Virtual Server	Windows XP, Windows Vista, Windows Server 2003	✓	×
VMware Server	Windows, Linux	✓	✓
VMware Workstation	Windows, Linux	✓	✓
VMware Workstation Player	Windows, Linux	✓	×
Windows Virtual PC	Windows -7	✓	×

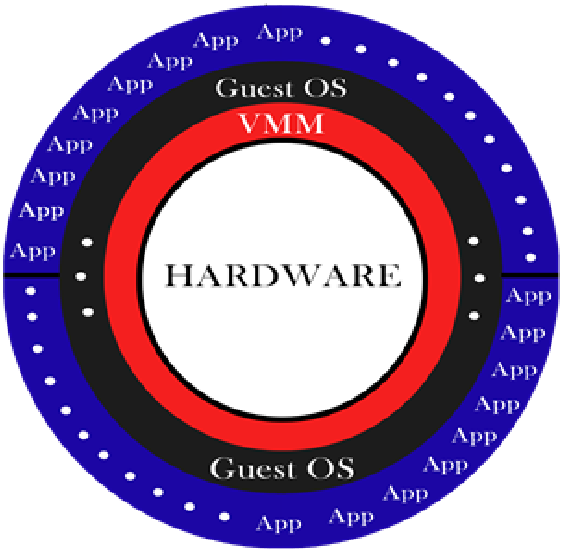


Figure 8. VM Abstraction

7. Grid vs. CC

It is always effective to understand a new thing by comparing it with the existing ones. Here, GC is selected because not only they, grid and CC, interacting with each other but also with the other existing technologies. Because the concepts of CC and GC are not always mutually exclusive where there are various distinct elements which overlap with each other, the real differences between them are often hidden on an inferior level of abstraction and hard to grasp. Consequently, both are repeatedly mistaken for each other; though that is not the case at all. For these reasons, Table 3 is used for examining clouds from the perspective of grids and to make a 32-point comparison between them. Without this across-sectional comparison, both concepts are often seen as the same computing physiologies under multiple names.

Table 3. Grid vs. Cloud Computing (Foster *et al.*, 2008)(Subbiah, Muthukumaran and Ramkumar, 2013)(Bardsiri and Amid, 2012)(Kiranjot and Anjandeep, 2014)(Jiang and Yang, 2010)(Ramzan and Alawairdhi, 2014) (Yunchuan *et al.*, 2014)(Jain, Sumit and Kumar, 2017)(Alessio *et al.*, 2014)(Patel, Patel and Panchal, 2017)

Parameter	*Grid Computing (GC)*	*Cloud Computing (CC)*
Conceptually Foundation (i.e. Origins)	It stems from the scientific and academic communities, or more precisely from the field of HPC. Later, it entered the commercial world.	It was originally drawn from both academia and industry.
Resources Type	Limited (because the hardware is limited).	Virtually unlimited (i.e. device-independent), it integrates together more diverse computational resources than grids.
Resources Place	Resources exist in computing centers that may be dispersed across different sites and countries in different places even in the entire globe.	CC relies on the principle of deploying servers and software in centralized locations and then offers them to end-users web-based services with multiple-access.
Data Centralization/Decentralization	Data Decentralization: the data may be organized in multiple locations and it can be accessed instantly from many remote places.	Data Centralization: the data are mostly stored in one location and it can be accessed instantly from remote places.
Security	Has lower security.	CC often has higher security.
Level of abstraction	Poorly (need for more details)	Highly (eliminate details)
Critical object	Computational resource	measured and well-defined service
Resource management	Distributed	Dynamic and adaptive distributed nature. Both centralized and decentralized.
Operating System (OS)	Unified OSs. To provide high-performance computing services, the existing grid resources are usually very tightly bound to a specific OS which may be any standard OS that has the ability to control the resources.	Different OSs provide users with higher response time, larger data storage, and higher I/O throughput. The primary virtualized PM might concurrently host multiple VMs, each beaked-up VM can be controlled by its own OS.
Allocation & Scheduling	Decentralized	They are both centralized and decentralized.
Failure management	Limited	Strong
Elasticity	Limited; the computational resources almost have a limited scaling. Scaling them up/down according to the current demand shows some difficulty.	High; the computational resources and services are scaled in both directions corresponding to the current demand. This may be done automatically.
Configuration	Difficult, IT specialists with enough technologically proficient to drive the configuration process.	Easier to configure.

Parameter	*Grid Computing (GC)*	*Cloud Computing (CC)*
open-source/closed-source software	Grids are open-source software. End-users and service providers alike can cooperate together in managing their grid.	Clouds are mostly closed-source software that uses automated systems management. However, only cloud service providers (CSPs) know how to manage their Clouds.
Virtualization	Not a commodity. Applications are strictly tied to the actual physical components of the system due to the virtualization absence.	Since the abstraction of virtualization is vital and essential, the developers are free from thinking about the actual infrastructure of their applications.
Procedure	GC involves fine graining of a large major task, called Job, into many minor sub-tasks that are executed in parallel on multiple separated computing units (i.e. servers or individual computers of the distributed system).	Each cloud is more than a collection of computational resources. It is an arrangement of computational resources that allows users to avail of various services that can be integrated to achieve the desired job result. It offers a controlling mechanism for viewing, sharing, and managing these resources. Controlling involves, but is not limited to, role provisioning, requesting and change requests, reimaging, workload balancing/rebalancing, and monitoring.
Pricing Model	Service-level pricing: when there are payments, Grids use fixed payment perused service.	Utility-pricing: while the computational resources are paid by cloud service providers (CSPs), end-users are usually billed only for their actual level of consumption, namely pay-per-use.
User management	Decentralized and also (VO)-based	Centralized or can be delegated (i.e. outsourced) to a third-party provider.
Users Type	On the whole, Grids are commonly available for not-for-profit work and still, there are no commonly accepted commercial-running grid services available on the IT market.	Generally, commercial businesses of all sizes or researchers with generic IT needs.
Usage	Grids can offer a large number of domain-specific high-level services that satisfy the customer-specific requirements. It usually used in narrow fields for running large-scale computational applications that are resource-intensive and require HPC. Most of these applications with their associated supporting tools are for academic computing needs and on a non-commercial basis, like image processing, nuclear researches, numerical weather forecasting, and bioinformatics applications (as DNA or protein sequence).	It can be widely used in all fields. It can support a myriad of purposes and well-defined services ranging from word processing, right up to the web hosting and real-time operations. However, clouds do not offer many of the domain-specific high-level services that are ordinarily provided by grids, such as nuclear researches, numerical weather forecasting, and others.

Parameter	*Grid Computing (GC)*	*Cloud Computing (CC)*
User-friendly	User-unfriendly; bearing in mind that the term “user-unfriendly” is unique to each person's skill and depends on his technical ability levels.	Even though there are still some difficulties in setting up sophisticated virtual machines to support complex applications and workflows, CC is relatively more user-friendly than GC. The ease-of-access and ease-of-use of CC let at least non-expert end-users get started easily. In this respect, the functional hosting of the cloud control panel is more than that of GC and offers a number of highly configurable options.
Complexity	Generally speaking, developing applications on a grid is a complex task and users of grids require some level of expertise.	Less complicated. There’re so many pronounced indicators that programmers with no experience of parallel and distributed systems can easily utilize the cloud-based resources for building cloud-hosted solutions.
Response Time	Not real-time; need to be scheduled	Real-time services or near-real-time
Transmission	Suffers from Internet delays	Faster than GC.
Service Type	Hardware and network services	Everything may be interfaced as a measured service through the Internet.
Requests type	Few but large allocation	Lots of small allocation
Scalability	Normal; Scalability is mainly reached by increasing the number of working nodes.	High; automatic resizing for scalability is done by dynamic reconfiguration.
Transparency	Almost low. GC is actually not transparency as CC.	High; CC has higher transparency.
Infrastructure	Using Low-level command	It comes with a set of high-level services which will be mentioned in later sections.
Network Interconnection	It has higher transmission latency and lower bandwidth.	It has lower transmission latency and higher bandwidth.
Ownership	Multiple owners.	A single owner in most cases. Each linked cloud has a high probability that it is possessed and operated by a single enterprise which is what it's known as a vendor or a CSP.
Quality of Service (QoS)	They are committed to a concrete QoS.	QoS is an Internet-based environment.

Even though there are some considerable differences in the fundamental concepts of these two technologies that do not necessarily mean that they are always mutually exclusive. Thus, adopting one of them needn’t excludes or precludes the other one. On the contrary, both of these orchestrated technologies may complement each other, forming a harmonious whole. Consequently, it is quite feasible to have a computational that encompasses one of them as a part of the other. For instance, it is quite feasible to have a computational grid as a part of a cloud, and vice versa. As such, both architectures might be used in the same computer network, even though they are represented in two different topographies. In addition, as CC relies on GC, the latter can be considered as the backbone of the first one and therefore it forms its core and infrastructure support (Maruf and Albert Y., 2017)(Bardsiri and Amid, 2012)(Kiranjot and Anjandeep, 2014)(Jiang and Yang, 2010).

But never the less both are still distributed computing that is strictly tied to the advances in communications and

(Inter) network technologies. When they are merged together, they form a huge business segment that can be foreseen as a business enabler for innumerable speed-to-market scenarios. This frontier paradigm can host and deliver many free and commercial platforms and frameworks. (Kiranjot and Anjandeep, 2014)(Jiang and Yang, 2010)(Anne-Lucie *et al.*, 2017)

Viewed in a broad sense, both trends seem to have closely related capabilities, such as the followings (Foster *et al.*, 2008) (Bardsiri and Amid, 2012)(Kiranjot and Anjandeep, 2014)(Jiang and Yang, 2010):

- Even though the focus view varied depending on each research community, one fundamental aspect remained fairly viewed in a broad sense: that is they are both indeed prime examples of distributed computing architectures regardless of where the network's computing elements are widely spread over a large-scale geographical area.
- CC and GC applications may benefit from each other since many are both compute and data-intensive.
- While both are used to economize computing by getting the most out of existing-computational resources, both models are networks that eliminate details and abstract processing tasks in an extensive manner.
- Both trends are around the core of modern computing: "service-oriented" and multitasked.
- They are both coming into action to offer enormous shared resources (processor, memory, mass storage device, input/output facilities, database, etc.) as an on-demand service model.
- Both architectures support heterogeneous hardware and software resources.
- They are both providing hardware virtualization. However, CC goes one further extra step to fully utilize virtualization technology to support elastic resource provisioning and sharing.

8. CC Architecture

Functionally, as depicted in Figure 9 and to deliver different levels of services to customers, the architecture of CC is typically envisioned as a stack of four abstract layers where each underlying layer hides the lower-level specific details from the layer above and unify all the differences amongst the different components of the underlying layers (Kaur, 2015)(Marques *et al.*, 2018)(Prajkta and Keole, 2012). Since each layer is loosely coupled with the layers underneath and above in a dynamic and adaptive manner, each layer is evolved independently on the other layers and should be appropriately configured, operated, managed and secured in a consistent manner (Foster *et al.*, 2008)(Naveen and Harpreet, 2013)(Arora *et al.*, 2017).

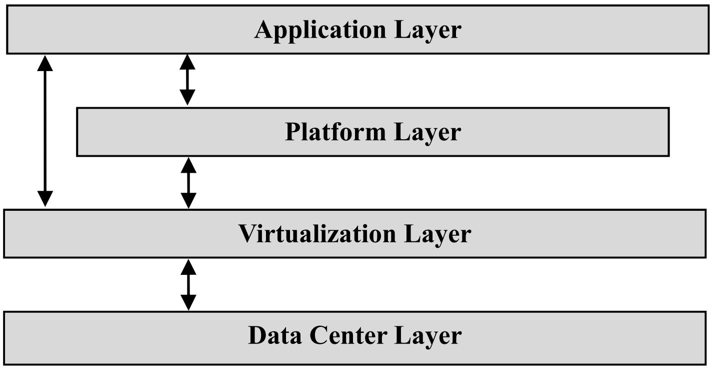


Figure 9. CC Architectural Design

In essence, this smart-multi-layer integration and data sharing is similar to the Open System Interconnection (OSI) framework of network protocols where each underlying layer can provide the necessary services to the layer directly above and can be seen as a client of the layer underneath (Patel, Patel and Panchal, 2017). A layer-to-layer comparison is shortly described below (Foster *et al.*, 2008)(Naveen and Harpreet, 2013)(Han *et al.*, 2016)(Bardsiri and Amid, 2012)(Sommerville, 2015)(Venkatachalapathy *et al.*, 2016)(Ashdown and Kyte, 2015):

- **Data Center Layer:** At the lowermost level of the hierarchy, this layer exists and used for managing the physical hardware resources of the CC. Both the hardware and the software of this layer are recognized as the actual components of the cloud. This Layer is usually made up of a large pool of well-configured interconnected machines and, thereby, it is referred to as Data Center Layer, Hardware Layer, Physical Layer, or Fabric Layer. It contains the raw hardware level resources including physical servers, mass storage units, interconnected network units (such as switches, routers, and so forth), code repository, and other supporting units (like electrical power and cooling systems). These computational resources may be physical and/or virtual resource elements. This layer may have thousands, if not millions of servers that are organized in racks and interconnected through network units.

- **Virtualization Layer:** Above the Data Center Layer, the Virtualized Layer exists and it is also formally **referred** to as the Unified Resource Layer or the Infrastructure Layer. This layer is built by using virtualization technologies that simulate the basic relationship between the OS and the underlying hardware component to allow multiple virtual machines to run on one physical machine. However, the different resource elements are highly abstracted and encapsulated to be offered as an unlimited pool of storage and integrated computational resources. When a customer requests a computational resource through this layer, a certain background process is identified behind the scenes to accomplish the request virtually and, in turn, allocate that resource to the customer.
- **Platform Layer:** This layer exists between the Virtualization Layer and the Application Layer and works as an interfacing gateway between them. Its main role is to reduce the burden of deploying the different applications of the Application Layer directly into the Virtualization Layer. It consists of OSs, and application frameworks which consist of a collection of special tools and services.
- **Application Layer:** on top of the said layers, the application layer exists and it is all about providing the actual simplified and improved agility applications. These meaningful applications are extended to be run on the "computing clouds" which are been accessed over the Internet by any authorized end-user who has the authority to access the cloud. However, these sophisticated applications should be different from the unclouded conventional applications, such as desktop applications, standalone applications, or window-based applications. For instance, their web-enabled computing environment is different than the desktop-based one by many aspects such as automatic-scaling, high availability, and lower operating cost.

From another perspective, when an individual uses the clouds, he or she is dealing with systems that must have three related aspects:

- Data: working with cloud data is different from conventional in-house systems. It requires special-efficient programming methods and tools to integrate, store, clean, filter, transform, and retrieve the data items. (Arora *et al.*, 2017)
- Applications: the CC applications and all the related services, including all the required communication protocols and all the related interactive interfaces, should be tuned enough to be within the range of the CC environment. Without this, the developing of any cloud-based application is no longer acceptable. Another a closely related matter, since they are not initially tailored for fully exploiting the cloud environment, conventional in-house applications are usually not adequate for the new cloud-hosted platform by many aspects such as automatic-scaling, high availability, solid reliability, and the hosting environment. (Naveen and Harpreet, 2013)(Tsz Lai, Trancong and Goh, 2012)(Edlund, 2012)(Foster *et al.*, 2008)(Subbiah, Muthukumaran and Ramkumar, 2013)(Chowdhary and Rawat, 2013)
- End-users: the users have obviously computers with standard web browsers connected to a suitable Internet connection (Daylami, 2015)(Smith, 2016)(Aspen and Kaitlyn, 2017)(Gaur and Anurag, 2017).

All the stated aspects listed above could be loosely coupled and widely dispersed along thousands of miles apart. So in conclusion, without solid and reliable application performance, customers could not have the seen productivity of this frontier paradigm. As a final point, this layered architecture is also called ring architecture (Marques *et al.*, 2018).

9. CC Service Models

Most of today's firms call for economical reliable solutions that drive real-time decision support, powerful content delivery, manage complex volumes and varieties of data items, and handle the heterogeneity of various computing environments (Al-Ta'ee, El-Omari and Kasasbeh, 2013)(Alessio *et al.*, 2014)(Patel, Patel and Panchal, 2017). Thereby, the philosophies behind clouds come to align the customers and providers perspectives and to satisfy the different customers' needs: storage capacity, processing power, operating system (e.g., Linux, MS Windows, Android, and Apple's Macintosh), software, etc.

From a service viewpoint and as it is illustrated in Figure 10, these various needs are basically categorized into three broad levels: Infrastructures, Platforms, and Software that will be delivered over the networks as Web services (Chowdhary and Rawat, 2013)(Khedr and Idrees, 2017). Each level covers one or more CC services (Prajkta and Keole, 2012). Close related to this and in order to provide the different types of computing services, three commonly service models have been proposed: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service

(PaaS), and Software-as-a-Service (SaaS) (Chowdhary and Rawat, 2013)(Yashodha Sambrani, 2016)(Sharma, Singh and Kaur, 2016)(Australian Government, 2013)(Roy and Sardda, 2016).

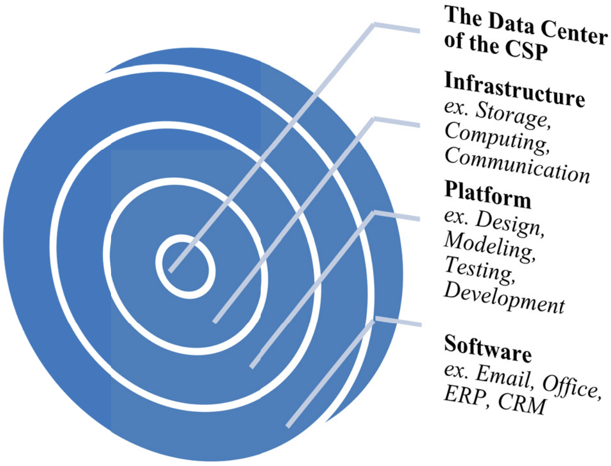
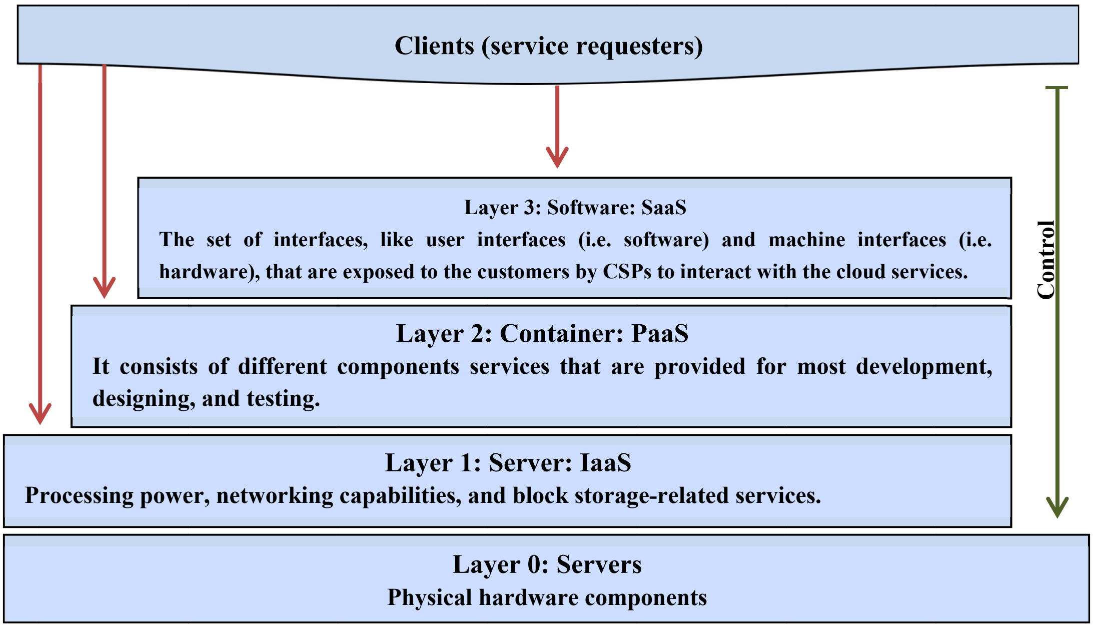

The Data Center of the CSP
Infrastructure
ex. Storage, Computing, Communication
Platform
ex. Design, Modeling, Testing, Development
Software
ex. Email, Office, ERP, CRM

Figure 10. Hierarchical View of CC

From a technical point of view and as it is clearly explained in Figure 11, these three typical service models are abstracted and viewed in a layered architecture (Khedr and Idrees, 2017)(Venkatachalapathy *et al.*, 2016)(Malik *et al.*, 2014)(Tsz Lai, Trancong and Goh, 2012)(C.Vijaya and P.Srinivasa, 2016).



Layer 3: Software: SaaS
The set of interfaces, like user interfaces (i.e. software) and machine interfaces (i.e. hardware), that are exposed to the customers by CSPs to interact with the cloud services.

Layer 2: Container: PaaS
It consists of different components services that are provided for most development, designing, and testing.

Layer 1: Server: IaaS
Processing power, networking capabilities, and block storage-related services.

Layer 0: Servers
Physical hardware components

Figure 11. The General layered Architecture of CC

Obviously, a layer is higher than another one when its delivered services are shaped out of the services produced by the underlying layer (Prajkta and Keole, 2012). Going forward with this layered architecture, this figure illustrates how the PaaS is built upon the IaaS and how the PaaS is utilizing the different computational resource elements of IaaS (Malik et al., 2014)(Tsz Lai, Trancong and Goh, 2012)(Ali, 2016). Likewise, SaaS is also being built upon both PaaS and IaaS (Malik *et al.*, 2014)(Tsz Lai, Trancong and Goh, 2012)(Ali, 2016) and, therefore, can support a myriad of purposes and services ranging from word processing, right up to the web hosting and real-time operations. Without this rationale of layers, customers couldn’t interact easily with this abstraction (Prajkta and Keole, 2012)(Malik *et al.*, 2014)(Tsz Lai, Trancong and Goh, 2012).

In essence, the lowest layer, Layer 0, is used to implement the cloud. It includes a collection of all the physical computational resources that are within the data-center of the cloud, such as servers, mass storage and network devices, electrical power and air conditioning systems, and among others. In order to hide its complexities efficiently from the outside world, this layer is hidden from customers and not exposed to them; it is only seen by the CSP (Solanki and Shaikh, 2014)(C.Vijaya and P.Srinivasa, 2016).

9.1 Infrastructure-as-a-Service (IaaS)

This service model is the lowest one of the three cloud-based services. However, it is the most expensive model among the other models. Regardless of their architecture, the same data-center can provide a mechanism to share different well-suited infrastructure resources among various end-users (Ramzan and Alawairdhi, 2014) (Bertolino, Nautiyal and Malik, 2017). These hosted resource elements are interconnected and appeared to the clients as a single coherent entity according to their needs (Ramzan and Alawairdhi, 2014). This allows CSPs to efficiently target a variety of customers' groups according to their needs and preferences (Bertolino, Nautiyal and Malik, 2017). All the offered resource elements of this model are in related to the base cloud infrastructure and may include, but are not limited to, OSs, servers, service computational power, storage-related services, memory, networking capabilities, bandwidth, input/output facilities, and other essential computational resource elements that meet different customers' needs (Gumbi and Mnkandla, 2015)(Daylami, 2015)(Bardsiri and Amid, 2012)(Ramzan and Alawairdhi, 2014)(Yashodha Sambrani, 2016)(Kaur and Chana, 2015)(C.Vijaya and P.Srinivasa, 2016)(Ali, 2016). As a consequence, customers can easily transfer their existing workload to the clouds and build their own system(s) on the top of these virtual infrastructures (Gumbi and Mnkandla, 2015)(Daylami, 2015)(Bardsiri and Amid, 2012)(Kaur and Chana, 2015). Since the cloud service providers (CSPs) are accountable for all of the administrative and technical assistance, all the end-user needs to do is just using these computational resources (Gumbi and Mnkandla, 2015)(Daylami, 2015)(Bardsiri and Amid, 2012)(Kaur and Chana, 2015)(Ali *et al.*, 2015).

Except for some particular applications that need some special configurations, end-users of this model are granted no authority over the already provided underlying supported infrastructure and they have just a narrow access control over certain functional components of the overall cloud networks such as the firewalls (Chowdhary and Rawat, 2013)(Daylami, 2015)(Bardsiri and Amid, 2012)(Laverty, Wood and Turchek, 2014)(Sharma, Singh and Kaur, 2016)(Ali, 2016). However, customers are given some authority to further control and customize OSs, mass storage devices, and their own deployed applications (Chowdhary and Rawat, 2013)(Daylami, 2015)(Bardsiri and Amid, 2012)(Laverty, Wood and Turchek, 2014)(Sharma, Singh and Kaur, 2016)(Ali, 2016). Not just that, they can also freely set and activate any form of software and programming environment on the top of the offered services as though they were utilizing their local-home computers (Tsz Lai, Trancong and Goh, 2012). For instance, they have the facilities to install their own OSs, set up their mass storage devices, and install their particular applications (Ali, 2016). An important thing that should be stated here that customers are not renting specific infrastructure, like server, hard-disk drive, or network router, but instead they are using the cloud-based infrastructure (Daylami, 2015)(Tsz Lai, Trancong and Goh, 2012)(Deitel Pau and Deitel, 2017).

In accordance with the security functions of this model, it is important to realize that the CSPs definitely manage and control the Cloud infrastructure and the related security issues. In essence, the physical security functions are controlled by the CSP and the other security-related aspects of the virtual system are left to the customers themselves. (Chowdhary and Rawat, 2013)(Daylami, 2015)(Sharma, Singh and Kaur, 2016)(Ali, 2016).

By taking storage as an example of this service model, when the consumers' data continue to grow exponentially, their systems should be expanded to meet their actual demands of anticipated data growth (Chaudhari and Patel, 2017)(Venkatachalapathy *et al.*, 2016). As an alternative to driving up the sizes of their computer storage, they can use IaaS as a storage service to provide them with massive-scale data storage capacity (Venkatachalapathy *et al.*, 2016). This is especially true when they found that their data are too big to be limited to their own mass storage devices. Being more specific, some references also referred to this storage service as Storage-as-a-service (STaaS) to point out that the necessary storage infrastructure can be provided to the users as on-demand service (Sommerville, 2015). Users just "pay-per-use" for their exact rate of resource usage without purchasing any extra hard disk or even knowing something about where their data are stored or manipulated. However, they can also customize these computational resources upon their needs.

IBM, with Autonomic Computing Manifesto (Ali, 2016), Amazon, with its Amazon Elastic Compute Cloud (EC2) (Daylami, 2015)(Tsz Lai, Trancong and Goh, 2012)(Aspen and Kaitlyn, 2017)(Deitel Pau and Deitel, 2017)(Bertolino, Nautiyal and Malik, 2017)(Prajkta and Keole, 2012) or Amazon S3 (Simple Storage Service) (Ali, 2016), are arguably ones of the leading pioneers related to this service model. The IaaS services provided by Rackspace company is another example (Tsz Lai, Trancong and Goh, 2012)(Australian Government, 2013)(Deitel Pau and Deitel, 2017) (Gholami, Daneshgar and Beydoun, 2017).

9.2 Platform-as-a-Service (PaaS)

The platform represents the abstraction layer between the two layers: IaaS and SaaS. It is like an intermediate bridge between the so complicated infrastructure and the end-users (Yuzhong and Lei, 2014). In other words, this service model is the in-between bridge between the virtualized computer architectures, offered by IaaS, and the software-based applications, offered by SaaS (Sareen, 2013). Here, the raw hardware infrastructure is supported besides the preconfigured systems running the required OSs, databases, programming languages, tools, web servers, and the whole execution environment (Aspen and Kaitlyn, 2017)(C.Vijaya and P.Srinivasa, 2016)(Ali, 2016). As such, it is considered as the outgrowth of IaaS (Laverty, Wood and Turchek, 2014)(C.Vijaya and P.Srinivasa, 2016).

Rather than installing the software and the different tools on their own computer, software developers use this model as a computing platform or a middleware for building their own applications on the service provider's cloud infrastructure without the cost or complexity of setting up the hardware and software in-house. Within this context, they use this model like a cloud-hosted production environment for developing, testing, debugging, deploying, running and managing their own applications remotely. They are facilitated with a set of pretested APIs to carry out their interactions and rapidly accelerate their development. In other words, working on this service is likely to be treated as a means for renting OSs and the underlying middleware but by using the usage-based pricing model: "pay-as-you-use". (Chowdhary and Rawat, 2013)(Bardsiri and Amid, 2012)(Daylami, 2015)(Laverty, Wood and Turchek, 2014)(Sharma, Singh and Kaur, 2016)(Anne-Lucie *et al.*, 2017)(Deitel Pau and Deitel, 2017)

Microsoft's Windows Azure^{TM}, as a CSP foundation and platform, is one of the best-known examples of this type of clouding (Daylami, 2015)(Australian Government, 2013)(Deitel Pau and Deitel, 2017)(Ali, 2016). It supports both PaaS and IaaS services and runs Windows and SQL Server (Daylami, 2015)(Alwada'n, 2016)(Sharma, Singh and Kaur, 2016). "Google's App Engine", "SalesForce-Force.com", "Amazon Relational Database Services", and "Amazon Web Services™ Elastic Beanstalk" are other examples of this service model (Tsz Lai, Trancong and Goh, 2012)(Daylami, 2015)(Sharma, Singh and Kaur, 2016)(Australian Government, 2013)(Prajkta and Keole, 2012)(Deitel Pau and Deitel, 2017)(Ali, 2016) where the developers find in these platforms an elastic high-productivity environment for developing and provisioning their full applications (Daylami, 2015)(Alwada'n, 2016)(Sharma, Singh and Kaur, 2016)(Australian Government, 2013)(Ali, 2016).

9.3 Software-as-a-Service (SaaS)

Besides that this service model is the simplest layer of this category (Khedr and Idrees, 2017), it is the highest one among the three cloud-hosted services and, therefore, sometimes it is also referred to as "Cloudware". While PaaS target programmers, SaaS target clients (i.e. end-users) (Prajkta and Keole, 2012). In its simplest form, this service is basically a middleware for end-users to use their full cloud-hosted applications that are provided by their vendors over the cloud infrastructure (Daylami, 2015)(C.Vijaya and P.Srinivasa, 2016). They, obviously, use these applications over their own connected devices by using well-designed interactive interfaces that are actually no more than web-based browsers accessed over the Internet (Ali, 2016)(M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018). Since no hardware to buy and activate, no software to install and set up, and no lengthy implementation to endure, this model is in contrast to the classical installation and distribution process which implies a complete time-consuming installation and launching on actual physical infrastructure (Anne-Lucie *et al.*, 2017)(Aspen and Kaitlyn, 2017)(M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018).

The attractive idea behind this service model is its target of replacing your special applications running on your own computer with that are based on SaaS and delivered over the cloud as an on-demand basis (Aspen and Kaitlyn, 2017)(C.Vijaya and P.Srinivasa, 2016). These cloud-like applications run on some servers elsewhere and only require a web browser and Internet connectivity, without any special local hardware or software (Daylami, 2015)(Deitel Pau and Deitel, 2017)(Aspen and Kaitlyn, 2017). Absolutely, you don't really have enough time to create everything from scratch or at least buying the software and upgrading it (Aspen and Kaitlyn, 2017)(M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018). So SaaS with its pay-per-use pattern is relatively more affordable than the other choice which, in turn, helps to minimize the total costs and to maximize the investments (Sommerville, 2015)(Sharma, Singh and Kaur, 2016)(Deitel Pau and Deitel, 2017). Working on this service is like purchasing the use of a working application like Supply Chain Management (SCM), Human Resource Management (HRM), Enterprise Resource Planning (ERP), Customer Relationship Management

(CRM), and all other known and unknown cloud applications (Malik *et al.*, 2014)(Laverty, Wood and Turchek, 2014). "Microsoft Office 365" and "Google Docs" are the most dominant examples of this service model where the documents can be edited and stored through the delivered cloud-based software (Tsz Lai, Trancong and Goh, 2012)(Aspen and Kaitlyn, 2017).

In essence, different SaaS application components are typically accessed by end-users using thin client interfaces for publishing and orchestrating these applications as measured services (Abdu *et al.*, 2017)(Tsz Lai, Trancong and Goh, 2012)(Edlund, 2012)(Sareen, 2013)(Sommerville, 2015). And moreover, many enterprises use this service model for storing their critical important and real-time information distributed through the Internet as a fully functional software service (Daylami, 2015). Their applications are completely run on the cloud provider's infrastructure and they have the ability to access them via the Internet through different computing devices without an essential need to install any software on their own devices or even updating them (Sareen, 2013)(Daylami, 2015). At the other end, the CSP maintains and upgrades only a single instance for each application that is used by all end-users (Khedr and Idrees, 2017) without giving them any capability to manage or control the underlying cloud infrastructure (Chowdhary and Rawat, 2013)(Daylami, 2015)(Bardsiri and Amid, 2012)(Australian Government, 2013).

9.4 Other Models of "as a Service" Family

Although the services discussed so far are the most frequently cited service levels, the "as-a-Service" tag has evolved into other cloud-like services:

9.4.1 Database as a Service (DBaaS)

As the name suggests, this sophisticated service model is used for delivering database functionality and it may be considered as a special subspecialty of PaaS. While the database itself is installed, configured, operated, orchestrated, maintained, and secured by the service providers themselves, all the end-users need to do is just only using the database. An end-user may know the existence of some operations encapsulated inside a catalog of supported services and, in turn, select and implement some of them properly without having to know internally how these operations are being carried out. Even with complicated database operations, like resizing a cluster, the abstraction level provided by the DBaaS is high and everything becomes a simple call to a well-defined API. To sum up, it is vitally important to realize that the abstraction level provided by the DBaaS is high and the motivation of developers and database administrators (DBAs) will be concentrated on the application itself rather than on the underlying boring minutiae of the DB operations. (Neves *et al.*, 2016)(Patel, Patel and Panchal, 2017)(Prajkta and Keole, 2012)

9.4.2 Big Data as a Service (BDaaS) & Analytics as a Service (AaaS)

These two generic and data-intensive services may be possibly classified as being two scalable subgroups under the umbrella of SaaS and they are used for applications that are relying on vast data volumes, the so-called "Big Data" (Devang Swami, 2016)(Neves *et al.*, 2016). This big data usually require real-time or at least near-real-time processing and, therefore, there is a need for fast and power-efficient techniques where CC may be the predominant choice for this (Khajenasiri *et al.*, 2016)(Neves *et al.*, 2016)(Dhabhai and Gupta, 2106)(Ali *et al.*, 2015). It offers an innovative high-productivity environment for information management and big-data analysis (Venkatachalapathy *et al.*, 2016)(Neves *et al.*, 2016)(Patel, Patel and Panchal, 2017).

Since BDaaS is based on the concept that the software product has an enormous amount of data and these data should be delivered on demand to the customers, most big-data analysts have increasingly focused their efforts toward implementing massive-scale data management and archiving via cloud platforms (Neves *et al.*, 2016). As a result, they come with AaaS as a practical way to correlate data, extract new patterns and predict novel trends by analyzing these big data stored in BDaaS (Sareen, 2013)(Neves *et al.*, 2016).

Both of these models, on the other hand, are usually used commercially by both business and scientific areas for selling some important data such as the data that are utilized for forecasting weather or analyzing financial stock markets (Devang Swami, 2016)(Sareen, 2013)(Neves *et al.*, 2016).

9.4.3 Security as a Service (SECaaS)

Accounting for the rapid growth of security vulnerabilities, this model was coined. It is a business model in which some CSPs integrate their security and privacy services (e.g., anti-virus, anti-malware/spyware, intrusion detection, authentication, and security event management) into this model (Sareen, 2013)(Laverty, Wood and Turchek, 2014)(Boudi1 *et al.*, 2018).

9.4.4 Test environment as a Service (TaaS)

It is another outsourcing business model that can be classified as a further subgroup of SaaS. Besides that the cloud itself is used as a testing environment, this service model has an effective way of automated testing that is better than the conventional ones ever were (Bertolino, Nautiyal and Malik, 2017).

Rather than using the formal manual testing, this business model is used for cloud testing in which the set of the testing activities are usually spanned to a third-party software company that has on-demand test labs and professional software applications with its associated simulated data. Provided that this outsourced third-party is specialized in simulating multi-platform testing over representative environments, it hosts centrally business and enterprise applications and its associated real data and then tests them in a way that provides management virtual view of multi-platform and real-world situations (Sommerville, 2015)(Boudi1 *et al.*, 2018)(Bertolino, Nautiyal and Malik, 2017). The CloudTestLite provided by SOASTA company is one of the known examples of this model (Bertolino, Nautiyal and Malik, 2017).

In favor of the automation, there are particularly three further coherent sub-models that are part of this service model:

- Functional Testing as a Service: the testers in this sub-model are concerned with the software functionalities to guarantee that it works properly and, thus, it is used for functional testing such as Unit testing, integration testing, regression testing, system testing, and user interface testing (Sommerville, 2015)(Bertolino, Nautiyal and Malik, 2017). It is extremely important to realize that the various test cases are practically derived from the system specification (Bertolino, Nautiyal and Malik, 2017).
- Performance Testing as a Service: Unfortunately, some cloud-based systems exhibit some severe risks of performance demanding, more specifically when they are becoming heavily loaded with large volumes of workload (Sommerville, 2015). Along the way of revealing the overall timing-accurate problems and in order to check the tested systems for tolerating a large number of users and transactions, there will be a necessity to mimic the real-world situation over performing stress and load testing. Thus, this sub-model of non-functional testing allows heavy volumes of workload to be deployed and scaled-up quickly by creating a large number of concurrent virtual users or transactional operations, then power them to concurrently access the tested web-applications and, finally, generate many more operations than that are likely to occur in practice. For this purpose, there are many software-checking tools that allow cloud-based services to be examined and tested automatically for withstanding performance. (Bertolino, Nautiyal and Malik, 2017)
- Security Testing as a Service: in this sub-model of non-functional testing, the testbeds are interrelated with the security of the software and, hence, it is used for scanning the applications and various websites with the intention of finding any potential vulnerability which might be abused by some malicious attackers (Boudi1 *et al.*, 2018)(Bertolino, Nautiyal and Malik, 2017). Again, there are many security-testing tools that can automatically run this set of security tests. It is worth remarking that the different testing events of this model may be possibly outsourced to a third-party provider (Bertolino, Nautiyal and Malik, 2017).

9.4.5 Choosing the Right Service Model: IaaS, PaaS, or SaaS

Recently, several cloud data centers are being designed and built at various scales for providing many innovative cloud services (George Pallis, 2010). However, some computing architectures among IaaS, PaaS, and SaaS are often mistaken for each other; though that is not the case at all (Daylami, 2015). This is due to the followings:

- The actual difference between the services models listed above, however, is not always clear-cut because there are several characteristics in common between them and, moreover, they are not always mutually exclusive (Prajkta and Keole, 2012).
- There are no clear or formally standard definitions for services, and it is likely that some vendors describe their cloud services differently than other vendors. For instance, most cloud vendors, as Amazon, do not define and use the same terms that are familiar within the cloud-based society such as SaaS, PaaS, or IaaS (Daylami, 2015)(Bardsiri and Amid, 2012). Back-and-forth, these boundaries between these services models can be scaled up and down by some providers according to customers' needs (Prajkta and Keole, 2012).
- The tangible difference between IaaS and PaaS are sometimes hard to grasp because some IaaS providers may deliver other services that fall under the family tree of PaaS; such as OSs, database, or application development platform (Daylami, 2015). Moreover, while some vendors call PaaS as "cloud software environments", many other vendors call IaaS as Hardware-as-a-Service (HaaS).

In both SaaS and PaaS, the customers have no facility to manage, control, or change the already provided infrastructure (Chowdhary and Rawat, 2013)(Daylami, 2015)(Bardsiri and Amid, 2012)(Laverty, Wood and Turchek, 2014)(Sharma, Singh and Kaur, 2016)(Ali, 2016). Although the customers are given no control over the base cloud infrastructure, they are given some facilities of configuration options for setting and customizing the hosted applications according to their specific needs (Chowdhary and Rawat, 2013)(Daylami, 2015)(Bardsiri and Amid, 2012)(Laverty, Wood and Turchek, 2014)(Sharma, Singh and Kaur, 2016)(Ali, 2016). This is clearly depicted in Table 4. To a better understanding of these cloud services, the last row of this table also gives some analogies to compare the three commonly service models with the classical one. As opposed to the CC model which stores the software remotely, the classical model is like buying a vehicle where the vehicle owner is fully responsible for the fuel, pump up tires, oil replacement, and garage maintenance. However, upgrading to a new model means buying a new one.

Table 4. A comparison between the three commonly service models

Parameter	*IaaS management*	*PaaS management*	*SaaS management*
Users' data	End-user	End-user	End-user
Hosted applications	End-user	End-user	CSP
Development tools	End-user	CSP	CSP
Management tools	End-user	CSP	CSP
OS	End-user	CSP	CSP
Virtualization	End-user	CSP	CSP
Servers and storage	CSP	CSP	CSP
Networking resources	CSP	CSP	CSP
Data center	CSP	CSP	CSP
Suitable for	IT administrators	Software developers	End-users
Examples	“Google’s Compute Engine” and “Amazon Elastic Compute Cloud (EC2)”	“Google’s App Engine”, “SalesForce CRM”, and “Amazon Relational Database Services”	“Google Docs”, Gmail, and “Microsoft Office 365”
Figurative examples using car as a service (CaaS)	Leasing a car: You can choose the car you want and drive it wherever you wish to your desired places. However, somebody else has possession of this car but you can use it for a specified period as if you were the owner. Upgrading means returning it back and renting another.	Taking a taxi: seat and relax, the driver will drive the taxi to your desired places. You can take your own luggage with you. Obviously, the taxi isn’t yours and you can simply take another one.	Going by public bus: Any public pus has predefined agreed routes. You can choose the bus that is more proper for your trip. Obviously, you share the route with other passengers and you aren't the owner of the bus.

It is worth noticing in this table that the end-users have more control and liberty with IaaS. Another notable difference is that the PaaS model may leave some responsibilities to the third-party providers like testing, debugging, and managing applications remotely. Thus, in this model, only the cloud-hosted applications and data portion are operated by the end-user and the rest of them are operated and managed by the same CSP.

In the SaaS model, end-users can only manage their own data part, all the other responsibilities (i.e. applications, infrastructure, and resources) are operated and directed by the CSP. It is worth mentioning, that the applications of this model are configured in such a way that end-users may just need some short limited technical support from the CSP.

What's more, besides the above-cloud-hosted services, there is still much room for improving existing services or at least coming up with new effective ones and therefore there is a scope for further researches upon these areas. Finally, to conclude the discussion of this section, without virtualization technology the concept of CC is not plausible; this is specifically true for IaaS (Prajkta and Keole, 2012).

10. CC Deployment Models

Aside from the cloud-like services, there is also an additional taxonomy which depends on the owner of the cloud infrastructure. Based on Figure 12, they fall into four different deployment models: Private, Public Cloud, Community, and Hybrid Clouds.

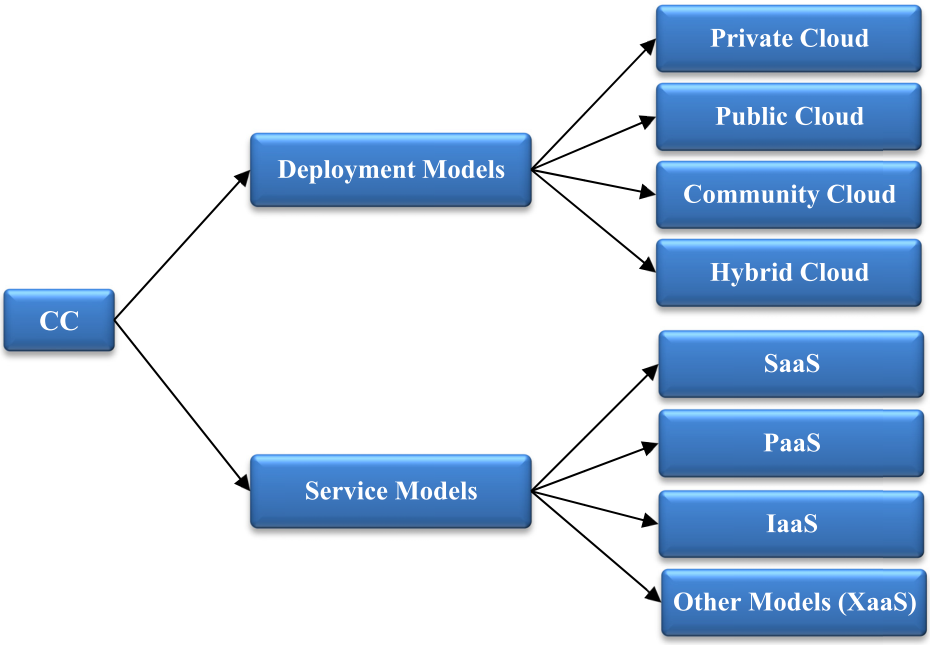


Figure 12. Service & Deployment Models of CC

10.1 Private Clouds

In this deployment model, the actual cloud infrastructure is exclusively dedicated to be used by a particular group of consumers belonging to a single enterprise that operates a particular business; it is not available for general public usage (Abdu *et al.*, 2017)(Jain, Sumit and Kumar, 2017). In that way, the different cloud-based services are delivered from a business' data-center to this specified group of customers and, therefore, any private cloud is only accessible and shared by authorized users within their private network (i.e. intranet) (Abdu *et al.*, 2017). However, the exchange of information outside of this intranet is controlled and restricted by secure channels. VMWARE, IBM, and Cisco are leading pioneers in this deployment model (Yashodha Sambrani, 2016). They are applicable when those internal users aim to further maintain greater control and privacy over their own critical data (Gumbi and Mnkandla, 2015)(Daylami, 2015) (Kaur and Chana, 2015).

Because private clouds need a fully-trained staff and continuous on-going maintenance is required by the IT office, the utilization of this model is more expensive than the other ones (Daylami, 2015)(Alwada'n, 2016) (Gaur and Anurag, 2017). Data Centers and IT department for private companies are the most prevailing examples of this type of clouding where the stored data are only accessible to any employee within the office who is granted an abstract virtualized access to the private cloud (Servan, 2014).

It may also be used to share the data and services more effectively and efficiently across the different branches of a large enterprise (Assunçãoa *et al.*, 2015). Equally important, since all resource elements and applications are available only for a particular business, this model is often selected for research and teaching facilities (Servan, 2014).

Beyond that, there are four variations of private clouds (Abdu *et al.*, 2017)(Jain, Sumit and Kumar, 2017)(Alwada'n, 2016):

- **On-premise clouds:** The private deployment is referred to as on-premise clouding when the cloud is only hosted (i.e. owned, installed, configured, operated and managed) to be used by the enterprise itself. This deployment technique has the highest level of security and also has the better ability to recover from component failure than the other modules ever are. It is also referred to as "Typical Private Cloud" or "internal clouds". The cases like military-related organizations that contain huge sizable confidential data imply an efficient example of this type of clouding. Because it is a single-owned by the enterprise itself, the maintenance and support are obviously on-site.

- **Managed Private Cloud**: although in this arrangement the infrastructure is also a single owned by the enterprise itself, a third-party provider might possibly take care of operating and managing the cloud infrastructure.
- **Hosted Private Cloud**: This private clouding is widely used when the cloud itself is hosted by an external provider specialized in cloud infrastructure, but that cloud is solely dedicated to a certain enterprise. It is also referred to as “Externally hosted Clouds” or “off-premise” clouds. This model is cheaper than the previous two sub-models. Small businesses utilizing services from Amazon and VMware are the most dominant examples of this type of clouding.
- **Virtual Private Cloud (VPC)**: in this arrangement an enterprise acquires computational resources from the public cloud and provides secure access to its customers through virtual private networks (VPN), which hides the fact that the cloud-like resources do not exist locally at the enterprise location. This extends the role of CSPs to the virtualization of communication networks in addition to the servers and applications. In this model, CC takes one step further by leveraging the VPN in addressing the limitations of both private and public clouds. This is why Private Cloud and VPC are used interchangeably in many cases. This model is considered as the cheapest type among the other private Clouds.

10.2 Public Clouds

Public clouds are suitable when a cloud service provider (CSP) wishes to make all the computational resources accessible to the customers or the general public for open usage as services over the cloud (Gumbi and Mnkandla, 2015)(Subbiah, Muthukumaran and Ramkumar, 2013)(Servan, 2014)(Abdu *et al.*, 2017) (Kaur and Chana, 2015). These computational resources, physical and/or virtual, often encompass networks, servers, stored data, applications, and among others. The public cloud supports multiple service models including IaaS, PaaS, and SaaS.

Since the cloud-related services might be sold to be accessible publicly by anyone on the Internet and are not restricted, this type is also referred to as open access or external model (Subbiah, Muthukumaran and Ramkumar, 2013)(Pooyan, Ahmad and Pahl, 2013)(Servan, 2014). Besides that this deployment method is owned and managed effectively by CSPs, it usually implies that a third-party provider runs the resources, delivers the cloud service over the Internet, and then bills the users using fine-grained utilities (Michael and Rajiv, 2012)(Edlund, 2012)(Filippi and McCarthy, 2012)(Tsz Lai, Trancong and Goh, 2012)(Taneja, Taneja and Chadha, 2012)(Foster *et al.*, 2008)(Sareen, 2013)(Naveen and Harpreet, 2013). While this deployment method is the most classical one notably within small businesses, it has easy resource management and more flexibility within the billing system; just it is fixed up and configured to be used by the public on demand, typically by the minute or hour.

Since the computing infrastructure is more accessible and shared remotely by the general public, public clouds have a lack of confidentiality and less secure as compared to private clouds but they help in bringing down operational IT costs and therefore reducing capital expenditure (Servan, 2014)(Gaur and Anurag, 2017). In addition, compliance issues like the lack of visibility and control over the computing infrastructure are also present (Pooyan, Ahmad and Pahl, 2013)(Servan, 2014). Nevertheless, this does not seem to be a huge concern for early-stage-startup businesses.

As this deployment model is generally owned and operated by large organizations, its main related CSPs are “Microsoft Azure”, “IBM Soft Layer”, “Google Compute Engine (GCE) ”, and “Amazon Web Services (AWS)” (Abdu *et al.*, 2017).

10.3 Community Clouds

This deployment model was born because of the criticality in the distribution of the sharing of rights and privileges over the different computational resources (Subbiah, Muthukumaran and Ramkumar, 2013). It is an integration to use the CC technology between a particular set of some firms which have similar objectives and requirements (Gaur and Anurag, 2017)(Ali, 2016). It is like a private cloud that is accessed and shared remotely by several businesses that have shared compliance considerations in common for using the cloud services, such as: missions, visions, policies, security requirements, compliance considerations, and other related shared interests (Subbiah, Muthukumaran and Ramkumar, 2013)(The National Institute of Standards and Technology (NIST), 2011)(Abdu *et al.*, 2017)(Australian Government, 2013)(Ali, 2016).

To this aim, mission-critical services and crucial-sensitive data that require more strict security requirements are typically governed on a private cloud or within the control of the same firm (Daylami, 2015) (Subbiah, Muthukumaran and Ramkumar, 2013)(Servan, 2014)(C.Vijaya and P.Srinivasa, 2016) (Ali, 2016). On the other side, less-sensitive data and information are normally hosted and outsourced on the public cloud (Daylami, 2015)(Subbiah, Muthukumaran and Ramkumar, 2013)(Servan, 2014)(C.Vijaya and P.Srinivasa, 2016)(Ali, 2016). This combination should be bonded together by standardized technology to enable the portability of data and application.

While the cloud infrastructure of the private cloud and community cloud is provisioned for exclusive use, the main difference between them is that private cloud involves only one organization, whereas community cloud involves multiple organizations that belong to a particular community such as banks, hospitals, trading firms, and so forth (Gaur and Anurag, 2017)(C.Vijaya and P.Srinivasa, 2016).

In this model, the involved enterprises establish a consortium agreement based on which one the shared data centers are allocated either on or off-premise (Abdu *et al.*, 2017). Since all the computational resources and applications are allocated, accessed, operated, managed, and shared remotely among businesses within the same community, the actual infrastructure ownership here is the community itself, as the name implies (Subbiah, Muthukumaran and Ramkumar, 2013). However, if there is more than one administrative center, all of them should have in common purpose, requirements, and regulations (Servan, 2014). Hence, it has more degree of economic scalability than the private clouding. A good example of this type of clouding is a group of hospitals might create a healthcare community cloud to hold Electronic Health Records (EHR) of patients and share the information among medical specialists, patients, medical insurance agents, technicians, and other close-related participants (Servan, 2014)(Daylami, 2015)(M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018)(Mirarab, Fard and Shamsi, 2014).

10.4 Hybrid clouds

Hybrid clouds are, as their name suggests, designed to hybridize an arrangement of at least two distinct deployment models (private, community, or public) along with their local infrastructures to provide a well-managed computing framework (Gumbi and Mnkandla, 2015)(Kaur and Chana, 2015)(Ali, 2016). It is clear that there is a need to think intensely that this model fuses the positive capabilities of two or more models that are fused together but remains functioning as individual entities (Subbiah, Muthukumaran and Ramkumar, 2013)(Servan, 2014)(C.Vijaya and P.Srinivasa, 2016). Functionality, this model cannot be categorized as one of the above-stated models: public, private, or community. This is because its scope follows at least two of the three models and also it crosses the providers of public and private clouds with their boundaries (Subbiah, Muthukumaran and Ramkumar, 2013)(Servan, 2014)(C.Vijaya and P.Srinivasa, 2016)(Ali, 2016). It is practically used whenever the criticality, scalability, and flexibility required for a specified service do not entirely fall into only one cloud, private, public, or community cloud (Subbiah, Muthukumaran and Ramkumar, 2013). It also mixes between the affordability and the high security from both private and public clouds with orchestration and automation between them (Subbiah, Muthukumaran and Ramkumar, 2013)(Servan, 2014)(C.Vijaya and P.Srinivasa, 2016)(Ali, 2016).

However, this model comprises, among others, the following important drawback: the networking complexity becomes more and more complicated and, in turn, some critical issues related to the networking might be generated. This is recognized by the existence of both public and private clouds. It is being observed that most enterprises find hybrid clouds as the best alternative for hosting their cloud-based services. For example, their e-commerce website is implemented within a private cloud, where they have more control and security than public cloud ever is. However, their catalog website is provided through a public cloud, where it is more affordable. (Subbiah, Muthukumaran and Ramkumar, 2013)(Servan, 2014)(Australian Government, 2013)(C.Vijaya and P.Srinivasa, 2016)

Another example, if a hospital wished to maintain a database for its own patients while also allowing certain resources of information to be highly available to the other medical centers, a hybrid cloud system could be more advantageous; in this case: private and community (Servan, 2014). Furthermore, this hospital may use a program running in a public cloud to manipulate the data stored in a private cloud (Australian Government, 2013).

10.5 Which Deployment Model to Choose

Each one of the above-stated models, including its associated supporting tools, has its potential benefits as well as drawbacks that define which deployment to go with and which to ignore. Since there's a wide variety of features, customers may promote one feature over the others. So, this subsection is here to help these customers in figuring out the most suitable model based on their various needs.

While more scalability and cost benefits can be retained with the public cloud, more control and maximum customizability can be retained with the private cloud (Daylami, 2015)(Alwada'n, 2016)(Gaur and Anurag, 2017). As a matter of fact, public clouds share the underlying infrastructure between numerous customers

(Daylami, 2015)(Alwada'n, 2016)(Gaur and Anurag, 2017). Since its cloud belongs only to a particular business exclusively, private cloud is more secure compared to public cloud and, therefore, it is broadly used when data are too strategically critical and sensitive to be deployed on the public clouds (Gaur and Anurag, 2017). So, it is improved that private cloud lets a particular business benefit from cloud technology without the risk of possible public disclosure of its data (Daylami, 2015)(Alwada'n, 2016).

However, the last one, hybrid clouding, has a bit of both. It gives the best practical way to make cloud work to cover the business terms, support business processes, and balance risk and opportunity. It is worth mentioning that the heterogeneity of computational resources in public cloud is much higher than it is in private clouds (Servan, 2014)(Daylami, 2015)(Alwada'n, 2016)(Abdu *et al.*, 2017)(Jain, Sumit and Kumar, 2017)(Gaur and Anurag, 2017).

Given that there are no such clouding systems with a 100-percent secure architecture, Figure 13 demonstrates a high-level picture of the CC spectrum related to the security, control, cost, and heterogeneity. As one moves to the bottom in the CC spectrum, the more focus shift to significant cost saving but the less control and secure(Prajkta and Keole, 2012)(Malik *et al.*, 2014)(Naveen and Harpreet, 2013)(K.Pranathi, 2016)(Yunchuan *et al.*, 2014)(Boudi1 *et al.*, 2018).

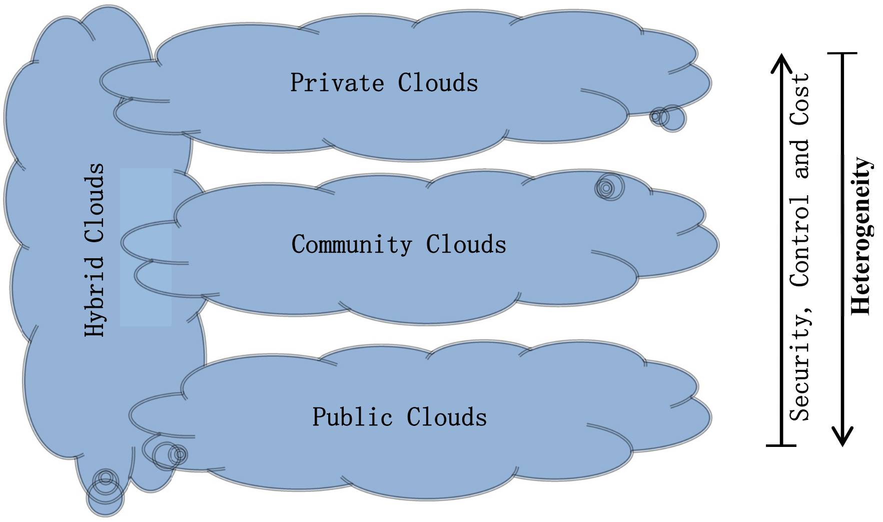


Figure 13. CC Spectrum

Since trusting a single cloud is risky at any point in time, there may be a possibility for facing some potential loss of data or a lack of resources and services availability. Enterprises, though, usually try on relying on two or more clouds for getting services in a truly efficiently and effectively manner (Chowdhary and Rawat, 2013).

Ultimately, there are two opportunities to adopt clouding: either migrate existing solutions to a cloud environment after making some adjustments or implement innovative business models and processes. However, in some cases, the previous existing IT systems should be thrown away because of the overall expenses of investing in them are relatively higher than the new systems costs. (Gholami, Daneshgar and Beydoun, 2017)

11. Potential Advantages of CC

Coupled with the beforehand advantages, there are more other proven benefits these provide the momentum for the CC migration:

- **Increase Efficiency:** One of the most vitally important things that consumers face by using computers is that the software can be developed up, upgraded, and evolved over time whereas the physical hardware can't (Edlund, 2012). Hardware might be possibly outdated and couldn't satisfy new needs before it can be manufactured and shipped (Shawish and Salama, 2014). By CC, though, there is no need like before for each firm to purchase and maintain additional computer hardware or even software (Gaur and Anurag, 2017)(Aspen and Kaitlyn, 2017). Customers of each firm are free from the burden of worrying about their low-level computers or about investing more for hardware and software (Aspen and Kaitlyn, 2017). Just they need normal computers that are connected to the Internet at minimum some sort of access (Gaur and Anurag, 2017)(Aspen and Kaitlyn, 2017). Then through their ordinary web browsers, they can run the required software as a service over the Internet without locally installing that software (Gaur and Anurag, 2017)(Aspen and Kaitlyn, 2017).

In addition, while CC can cut down the hardware maintenance and support, it relatively lowers staffing costs as well as their training costs where few IT employees are enough to run a large-scale enterprise (Ali, 2016). Day by day, all these vital factors may radically lead to a further increase in efficiency and a significant decrease in the IT expenses of any enterprise running clouding.

- **A lesser amount of errors:** Since the cloud interactions of the end-users are finally coming through dependable and pretested APIs, the cloud-based applications are generally less prone to errors (Prajkta and Keole, 2012)(Tsz Lai, Trancong and Goh, 2012)(Ali, 2016).
- **Latest version availability as soon as updates are released:** Because of the CSP does all the required updates on the server-side without local installation at the client-side, the clients always connected to the latest version of their selected services in a seamless swift manner as long as they are connected without any danger of having an outdated version of such services (Chowdhary and Rawat, 2013)(Daylami, 2015)(Ali *et al.*, 2015)(Mirarab, Fard and Shamsi, 2014). In a similar fashion, shared documents that are hosted on the cloud are always having the latest version regardless of the number or location of editors.

  As everything can be done remotely in the cloud, they just use the browsers of their local machines to access these innovative cloud services through the Internet (Subbiah, Muthukumaran and Ramkumar, 2013)(Anne-Lucie *et al.*, 2017)(Deitel Pau and Deitel, 2017)(George Pallis, 2010). They needn't worry about reinstalling or upgrading their software to the up-to-date version, solving software conflicts and synchronization, or purchasing new software licenses (Subbiah, Muthukumaran and Ramkumar, 2013)(Anne-Lucie *et al.*, 2017)(Deitel Pau and Deitel, 2017)(Ali *et al.*, 2015)(M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018). They have to pay only for that actual exact time for which resources have been being utilized by them and all the other associated sophisticated issues are done by the suppliers of these services (Subbiah, Muthukumaran and Ramkumar, 2013)(Gaur and Anurag, 2017).
- **Access from Innumerable Options:** Instant access to the cloud-based activities can be acquired not only from the usual PCs but through a wide variety of Internet-connected devices connected to the Internet as clients (Tsz Lai, Trancong and Goh, 2012)(Edlund, 2012)(Ramzan and Alawairdhi, 2014)(Sareen, 2013)(Han *et al.*, 2016).
- **Remain Competitive:** Computing technologies continue to evolve very rapidly mainly in terms of greater capacity, larger input/output throughput, higher bandwidth, lower network latency, higher performance, less variance, and better scalability (Jiang, 2018) (Essandoh, Osei and Kofi, 2014). Striving to accomplish the new requirements of their customers, vendors and developers should use these new technologies to remain competitive and improve their business capabilities (Essandoh, Osei and Kofi, 2014). But, as these new technologies are introduced, changing to them might not be an easy and direct task (Essandoh, Osei and Kofi, 2014). It would also be very costly and might have gone beyond budgeting. With the help of cloud technology, there is no need to purchase the physical infrastructure resources by cloud customers and then endure further money on upgrading or maintaining them (Michael and Rajiv, 2012)(Edlund, 2012)(Sareen, 2013)(Naveen and Harpreet, 2013)(Subbiah, Muthukumaran and Ramkumar, 2013)(Chowdhary and Rawat, 2013)(Essandoh, Osei and Kofi, 2014). Rather, they can outsource them from the cloud service provider (CSP) and remove the need for buying relatively expensive software and paying more money for their licensing costs (Michael and Rajiv, 2012)(Edlund, 2012) (Filippi and McCarthy, 2012)(Tsz Lai, Trancong and Goh, 2012)(Taneja, Taneja and Chadha, 2012)(Foster *et al.*, 2008)(Sareen, 2013)(Anne-Lucie *et al.*, 2017)(Essandoh, Osei and Kofi, 2014).
- **Risk Transferring:** Failures in the Cloud services have less chance to happen than of the other unclouded conventional solutions. Due to this, it is advisable to use the CC in outsourcing the critical infrastructure which in turn transfers business risks from the cloud customers themselves to the cloud service providers (CSPs) who often are better equipped for handling these risks. (Ali, 2016)(Tsz Lai, Trancong and Goh, 2012).
- **Increased Productivity:** As well CC has a significant impact in lowering costs and improving products' qualities, it delights the end-users with the best experiences of doing their work and interacting with customers (Ali, 2016). Instead of services that are localized for a particular market, it allows users to assemble, package, reformat and deliver their highly formatted content in any applicable format.

  On the other hand, the CC lets the IT managers be more focus on their core business needs and to concentrate on the strategic plans (Ali, 2016)(VMware Inc., 2019). It frees up the developers' times by letting them spend the most percentage of their development time on the business functions that are more relevant in advancing their businesses rather than spending that time on trying to solve computer automation problems or just thinking about the software and the underlying hardware (Aspen and Kaitlyn, 2017)(Ali, 2016).

- **Crucial Business Enabler:** as emerging economies are developed and new innovative technologies become available, business and society are now going in a global rapidly changing environment that has an increase in business competition and a rising in customers' expectations to face the increase in the business requirements (Zughoul, Al-Refai and El-Omari, 2016)(Ali, 2016). As such, businesses of all sizes have to significantly respond to face these challenges of the competitive markets by rapidly developing their existing software and in some cases changing them (Vikram and Bhatia, 2016)(Zughoul, Al-Refai and El-Omari, 2016)(Essandoh, Osei and Kofi, 2014).

  There's no doubt that CC has greatly revolutionized not only the way that business software is organized but also the way businesses and social interactions behave and, in turn, the mode the society conducts its business processes (Bardsiri and Amid, 2012)(Sommerville, 2015)(Deitel Pau and Deitel, 2017)(Essandoh, Osei and Kofi, 2014)(Ali, 2016). Since many businesses are now highly distributed and may have team members from different places across the world, CC has nearly become a mandatory part of many business ventures applications, only a few of which are currently deployed (Essandoh, Osei and Kofi, 2014)(Arora et al., 2017)(Ali, 2016). This is especially true for Small and Medium Enterprises (SMEs), who are perceived to be the actual soundless drivers of a nation's economy and the engine of growth in socio-economic development (Vikram and Bhatia, 2016)(Essandoh, Osei and Kofi, 2014)(Ali, 2016)(Pandey, Mishra and Tripathi, 2017). By this flexible business model, employees of cloud-based enterprises can work virtually on unlimited available resources and a large range of services as long as they can be online over the Internet no matter they have desktop or laptop computers or even any mobile device connected to the Internet as a client (Ali, 2016)(Pandey, Mishra and Tripathi, 2017)(Patel, Patel and Panchal, 2017). As opposed to the traditional method of working with an enterprise that stores the software locally, there is no need for employees to come to their offices to use their traditional software; they can exist anywhere and work more effectively whenever they can acquire Internet connectivity.

  As CC is truly becoming a competitive business model that transforms the way enterprises are doing their business, it involves many tools to automate processes in a different way and achieve faster turnaround time (Temkar, 2015)(Essandoh, Osei and Kofi, 2014). Going further, as this business model draws the milestones of the economic strategy and operational modes of any enterprise, any business that doesn't keep up may be left behind.

  On the whole, cutting down the charges of the product delivery can be reached by spending less money on using IT-based solutions and electronic services, like cloud-related ones (Pandey, Mishra and Tripathi, 2017). This, in turn, makes more opportunities for businesses to fulfill a competitive pricing edge that leads to relatively minimize their costs, maximize their selling margins and then maximize their investments.

- **Enterprise Resource Planning (ERP):** It is crucially important to realize that ERP frameworks are now becoming one of today's most widespread IT-based solutions. ERP offers a single solution that might contain an umbrella for all application needs. However, SMEs could not afford these systems due to their heavy costs (Vikram and Bhatia, 2016)(Zughoul, Al-Refai and El-Omari, 2016)(Pandey, Mishra and Tripathi, 2017). To overcome this problem, CC is used as a kernel part of operations for these SMEs (Ali, 2016)(Pandey, Mishra and Tripathi, 2017). Without Cloud mechanisms, these systems are only afforded by large-scale companies that have a large budget (Venkatachalapathy *et al.*, 2016)(Ali, 2016)(Pandey, Mishra and Tripathi, 2017).

- **Organizational Software Evolution**: since the cloud-based applications are deployed as Web services that deliver specific functionality from the service providers' data centers, there is no need to install, change, or upgrade these cloud-hosted applications on users' PCs like before. Thus, every application is deployed only once at the service provider's data-center which seems like a great cost-saving alternative and it is. (Sommerville, 2015)(Ali, 2016)

- **Dynamic Resource Provisioning:** This perceived benefit addresses the capability of automatic acquiring and releasing resources on-demand according to each individual scenario that may vary from time to time (Solanki and Shaikh, 2014)(C.Vijaya and P.Srinivasa, 2016)(George Pallis, 2010). It is around the capability of improving usage by adding more capabilities at high-peak claims and removing them whenever unneeded at low-peak demands (George Pallis, 2010). This automated service provisioning and sharing of resources can, therefore, be used to reduce the resources overprovisioning- buying more computational resources than is needed on average in order to cover the peak times of that abruptly stretches existing infrastructure to its upper limits and peak resource utilization (Solanki and Shaikh, 2014)(C.Vijaya and P.Srinivasa, 2016)(George Pallis, 2010). For instance, the seasonal extensive usage of the registration procedures at the beginning of academic semesters may significantly lead to a rapid increase in electronic service demands that may be beyond the maximum design load and communication limitations (Ghwanmeh, El-Omari and Khawaldeh, 2015). Another example, the periods of the financial crisis certainly lead to unanticipated increases in customer demands on using computer power, such as CPU quota, memory limits, disk space bounds, input/output limit rates, and among others (Pandey, Mishra and Tripathi, 2017). With clouds, there is no need as before for over-provisioning of resource elements to handle sudden future loads and to prepare for unexpected rushes in crisis service demand when there are many more messages than that are likely to occur in normal times (Solanki and Shaikh, 2014)(C.Vijaya and P.Srinivasa, 2016)(George Pallis, 2010). Thus, since the pay is only for the exact usage of the consumed resources, an enterprise can start with the cloud from the minimum required resource elements and dynamically allocated or deallocated them as needed (i.e. starts small, grows and evolves as required) (VMware Inc., 2019)(George Pallis, 2010).

12. Implementation Challenges of CC Adoption

Despite that, implementing cloud technology creates numerous unique opportunities for developing countries' enterprises and entrepreneurs; it imposes many distinctive challenges that have to be adequately addressed at a broader scale to promote the success of this technology. However, meeting these barriers is no longer easy; they are the critical factors that influence people's decisions to adopt CC initiatives (Gholami, Daneshgar and Beydoun, 2017)(Ali *et al.*, 2015)(Ali, 2016). Meanwhile, they can assist or limit the public sector's effort to diffuse CC initiatives and has a great effect in moving the delivery of services from push-model-based to pull-model-based (Ali, 2016). On the other hand, the risks of migrating to CC should be compared to the risks of staying without this trend by using the existing-outdated legacy environment that doesn't support cloud and cannot be executed directly (Gholami, Daneshgar and Beydoun, 2017)(Ali *et al.*, 2015)(Bertolino, Nautiyal and Malik, 2017)(Jiang, 2018)(Prajkta and Keole, 2012). According to analysts, nowadays it is very rare to find anyone living without Internet access and it is anticipated in the next few coming years that no any running business without these innovative cloud services (George Pallis, 2010). Advances and economic reasons will make it mandatory for all types of enterprises to adopt CC technology and, therefore, it is no longer seen as a luxury but a necessity (Gholami, Daneshgar and Beydoun, 2017)(Ali, 2016).

The successful adoption and usage of cloud framework in the third world countries rely on mitigating and solving the problems and constraining factors concerning these challenges. Followings are some of these key factors that have greatly limited the development of this field and have not been sufficiently addressed:

12.1 Culture and Resistance to Change

Since meeting new challenges require new ways of thinking, CC required changing in organizational culture towards service orientation and performance accountability (Ali, 2016)(Essandoh, Osei and Kofi, 2014). In order to help in increasing the project's chance of success, entrepreneurs, as well as employees, should share the same service culture and have some kind of comprehensive visions and strategies in common (Ali, 2016)(Essandoh, Osei and Kofi, 2014). For some time to come, there are a number of arrangements for this; let's just try to highlight three of them:

- All official and public firms should have a clean public image of Cloud technology. A sustainable and a viable-coherent vision of the developing countries future relating to CC needs to be developed and promulgated as an attempt towards closing these gaps.
- Active participation of employees is imperative: Employees should be included in the experimentation loop, flow involved in the service process and worked in a consultancy style (Sommerville, 2015)(Al-Ta'ee, El-Omari and Kasasbeh, 2013).
- Making training programs that are designed to provide technical knowledge and skills for all levels of participants to improve their awareness and satisfaction knowledge about the potential gaining of these e-services. This is very crucial since it is the lifeblood on which CC can be operated in an informed way. (Edlund, 2012)(Filippi and McCarthy, 2012)(Prajkta and Keole, 2012)

12.2 Customers' Satisfaction

For enormous faith from customers, Cloud service providers (CSPs) should work as closely as possible with their customers and end-users to ensure that the quality of cloud-hosted services is being provided as agreed and anticipated (Bertolino, Nautiyal and Malik, 2017). In some cases and in order to work effectively and successfully cater CC environments to the fullest, CSPs should work together with their customers to make some adaptations for certain elements of the current old-style legacy approaches, practices, policies, strategies, processes, and procedures to handle this innovative high-productivity environment (Laverty, Wood and Turchek, 2014)(Chaudhari and Patel, 2017)(Prajkta and Keole, 2012). Without this, there may be some gaps between the customers' expected view for the cloud-based systems and the actual system designs.

Another really important facilitator for CC adoption is represented by the possibility to provide personalized services that are able to meet the actual needs and variable demands of the end-users and help in increasing their satisfaction. This can be done by developing multiple service channels to capture the essential characteristics of the service process that meet their needs and accordingly building new functionalities or modifying old ones (Prajkta and Keole, 2012)(Edlund, 2012). To allow end-users to favor one programming feature over the other ones, another related key factor can be achieved by permitting customers to customize or create their own planned queries relevant to their needs and save them for upcoming usages.

One thing often ignored is that continuous adequate training should be as part of the CC adoption process. Another important thing is related to the continuous improvements in the quality of the provided services (Ali, 2016).

It is clear that all the already mentioned factors have great effects in delivering better innovative services and in managing any form of customers' resistance that can be encountered towards any new technology (Essandoh, Osei and Kofi, 2014)(Ali, 2016).

12.3 Top Management Support

Decision-makers and policymakers should play an active-fundamental role in shaping the development of technologies in the IT sector at large (Essandoh, Osei and Kofi, 2014)(Al-Ta'ee, El-Omari and Ghwanmeh, 2013). There's an accumulation of obvious evidence that their valuable support and involvement are definitely part of the story related to the seamless adoption of the cloud-like services (Essandoh, Osei and Kofi, 2014). However, senior managers were with high probability found to be prone to short-term planning, which prevents them from anticipating the long-term potential of CC.

12.4 Trust of Locations

Generally, end-users should be given full assurance that their data will be absolutely secure and out-of-the-way to other outsider users (Yashodha Sambrani, 2016)(Patel, Patel and Panchal, 2017). But cloud-based end-users have no control over or knowledge about the precise locations of the provided resources (Edlund, 2012)(Jain, Sumit and Kumar, 2017). These locations may be different than the places of the end-users and they are usually given at a high level of abstraction, as a country, state, city, or data-center (Edlund, 2012)(Jain, Sumit and Kumar, 2017). This certainly makes a mean of unconfidently and, therefore, rated as one of the most extreme challenges (Yashodha Sambrani, 2016).

12.5 Infrastructures

The immediate availability of adequate technological infrastructure is rated on the top of these challenges that need to be resolved. Unreliable IT infrastructure will certainly degrade the CC performance (Essandoh, Osei and Kofi, 2014)(Jiang, 2018). Likewise, the heterogeneous (i.e. diversity) nature of the different computing environments is another issue that needs to be tackled. For resolving this heterogeneity, there is a need to integrate them together as one system whereby this integration remains costly and time-consuming (Chraibi *et al.*, 2017)(Maria G. Koziri and Loukopoulos, 2017)(Ali *et al.*, 2015). Within this context, some unreliable outdated IT infrastructure should be kept away in order not to impact the CC performance.

12.6 Internet

Since the Internet goes hand-in-hand with CC, it has been recognized as a powerful vehicle to push down the cost of these cloud-like services. Since these electronic services always require continuous Internet linking and do not perform well with low-speed links, working with the cloud is facing up significant challenges and at the mercy of Internet availability. (Pooyan, Ahmad and Pahl, 2013)(Servan, 2014)(Daylami, 2015)(Vikram and Bhatia, 2016)(Smith, 2016)(Essandoh, Osei and Kofi, 2014) (Bertolino, Nautiyal and Malik, 2017)

However, the quality of the Internet connectivity cannot be guaranteed when a large number of users are accessing the cloud-based services at the same time and generate many more messages than that are likely to occur in the normal times which significantly increases the scale of the communication and data traffic. Due to this, the Internet may exhibit some severe degradation, go beyond the maximum design load, and, consequently, cause the unavailability of the Internet which certainly makes the cloud-related data to be unavailable. (Khajenasiri *et al.*, 2016)(Kuo *et al.*, 2014)(Acharjya and Ahmed, 2016)

In fact, nearly most Internet Service Providers (ISPs) claim in the Service Level Agreements (SLA) that they are definitely offering unlimited high-speed broadband Internet access, unlimited cloud storage services, and high-data-transfer rates without any consideration that the Internet may reach a point where demands go beyond capacity. However, their term “unlimited” is nothing more than a marketing tactic and advertising strategy. And moreover, they do not actually handle customer refunds when things go wrong as agreed and expected even if there is a guarantee that their web-based services will always be accessible. (El-Omari and Alzaghal, 2012)(El-Omari and Alzaghal, 2010b)(El-Omari and Alzaghal, 2010a) (El-Omari and Alzaghal, 2009)(Essandoh, Osei and Kofi, 2014)

As a matter of fact, CC requires a high speed of Internet access and, perhaps more importantly, a high-grade of broadband connection that is active all the time at reasonable prices. To come up with this focal point, efforts must also be doubled on the way of addressing poor Internet access on top of broadband coverage. Within this context, replacing or upgrading network infrastructure and rising Internet bandwidth are closely related. (El-Omari and Alzaghal, 2012)(El-Omari and Alzaghal, 2010b)(El-Omari and Alzaghal, 2010a) (El-Omari and Alzaghal, 2009)(Essandoh, Osei and Kofi, 2014)

12.7 Data Integration for Legacy Applications

Many firms deeply rely on their business-critical systems, which may have old-style applications that have been developed over relatively long periods and contain outdated computing techniques for managing data that do not meet the cloud competitive environment which essentially required good performance in handling computational complexities. Therefore, these in-house applications need to be revised and may be no longer valid when the organization's information is moved over the cloud infrastructure. (Michael and Rajiv, 2012) (Filippi and McCarthy, 2012)(Pooyan, Ahmad and Pahl, 2013)(Yashodha Sambrani, 2016)(Gholami, Daneshgar and Beydoun, 2017)(Ali *et al.*, 2015)(Prajkta and Keole, 2012)

Unify and integrate complex and diverse multi-structured data from multiple sources and systems is definitely not an easy process, especially when there are two environments: the traditional pre-cloud one and the cloud-based one (Gholami, Daneshgar and Beydoun, 2017)(Essandoh, Osei and Kofi, 2014)(Ali *et al.*, 2015). It requires in-depth studying for all the requirements regarding these two environments and reengineering the legacy systems to cope with cloud-based enabled (Gholami, Daneshgar and Beydoun, 2017)(Ali *et al.*, 2015). After that, it involves building new approaches which have great flexibility in handling, deploying, and distributing data within the enterprise-grade security. However, the existing obsolete approaches should be eliminated inside this new competitive environment. One more thing, besides that this new environment essentially involves advanced tools, techniques, frameworks, and methodologies (Chaudhari and Patel, 2017)(Gholami, Daneshgar and Beydoun, 2017), it may also involve automating many processes inside (Pooyan, Ahmad and Pahl, 2013)(Yashodha Sambrani, 2016)(Temkar, 2015). For the reason that there is a necessity for new solutions to work smoothly with already existing systems, compatibility is another significant enabler that greatly influences the adoption of CC (Essandoh, Osei and Kofi, 2014)(Gholami, Daneshgar and Beydoun, 2017) (Ali, 2016).

12.8 Legal Barriers

Since customers don’t aware of the locations of the actual places where their exact data are going to be hosted and stored on the cloud, it becomes particularly a serious problem to maintain data security and privacy protection. This is mainly true when it isn’t ignored that the working environment of the cloud is virtual and usually dispersed over different geographical locations around the globe and various countries might have diverse legal governance to govern any dispute of data privacy. (Naveen and Harpreet, 2013)(Subbiah, Muthukumaran and Ramkumar, 2013)(Yunchuan *et al.*, 2014)(Malik *et al.*, 2014)

It is important to recognize that the nonexistence of legislation that standardizes CC and its associated subjects is one of the most difficulties that encounter cloud adoption for some time to come (Karim *et al.*, 2017)(Essandoh, Osei and Kofi, 2014)(Sameera and Iraqi, 2017). Therefore, different government agencies should develop legal and ethical policies and procedures to govern the CC legislative requirements (Karim *et al.*, 2017)(Sameera and Iraqi, 2017).

12.9 Security, Privacy and ICT Policies

The moment anyone hears about the term “Cloud Computing”, he or she usually concerns the different parameters related to cloud security. These tricky parameters should be taken at uttermost consideration during CC adoption. Table 5 covers some of these related parameters. There are many valuable indicators that any violation of them is a key issue that acts as a stopping force against any advancement of CC and, in turn, any one of its consecutive technologies. On the contrary, satisfying them is the bedrock for the successful implementation of CC (Prajkta and Keole, 2012)(Tsz Lai, Trancong and Goh, 2012) (Ali, 2016). However, some of these features can be regarded as conflicting objectives and hard to be met (Prajkta and Keole, 2012)(Yunchuan *et al.*, 2014)

(K.Pranathi, 2016) (Bratterud, Happe and Duncan, 2017)(Ali, 2016).

Despite the fact that moving data to centralized services of operations via Cloud Technology offers a large number of computational resources and services, CC may present some security issues that should be considered (Bardsiri and Amid, 2012)(Essandoh, Osei and Kofi, 2014). Unfortunately, a disastrous state of the whole system may occur whenever data are been changed without the knowledge of their owner (Malik *et al.*, 2014). Not just that, a disaster might also occur when the sensitive data which are shifted to the cloud gets to be bankrupt or merely acquired by other firms. In light of the above, it is generally accepted that these risks are different than the conventional IT-based solutions due to many reasons, such as:

Table 5. Parameters of Security

Parameter of Security	***Purpose***
Encryption/ Decryption	For protecting and securing the traveling of data outside the local network, cloud-related data are nearly always encrypted. This is also when backing it up over external media storage. (Malik *et al.*, 2014)(Solanki and Shaikh, 2014)(K.Pranathi, 2016)(Tsz Lai, Trancong and Goh, 2012)
Transaction Management	It is required to ensure that any transaction between the client and the external cloud over the network cloud must be completed entirety (Naveen and Harpreet, 2013)(Subbiah, Muthukumaran and Ramkumar, 2013)(Malik *et al.*, 2014).
Segregation of Data	Since one customer's data may be stored alongside another customer's data at an in commonplace in the same cloud, separation of storage spaces is required (Solanki and Shaikh, 2014)(Chowdhary and Rawat, 2013)(Yunchuan *et al.*, 2014)(C.Vijaya and P.Srinivasa, 2016).
Traffic Control	An efficient analyzing of traffic characteristics is a useful guide in designing network infrastructure and making robust management and planning decisions (Venkatachalapathy *et al.*, 2016)(C.Vijaya and P.Srinivasa, 2016). This, in turn, increases the efficiency and productivity of the cloud (C.Vijaya and P.Srinivasa, 2016)(Tsz Lai, Trancong and Goh, 2012).
Intrusion Control/ Ownership	Since an organization's data could be stored next to its competitor's data, an intrusion control mechanism is required to protect the integrity and confidentiality of data (Chraibi *et al.*, 2017)(Neves *et al.*, 2016).
Risk Management	To properly deploy the cloud framework and to track the existing and the new risks, all security risks, like disaster recovery, must be properly managed as part of the data governance plan (Chraibi *et al.*, 2017)(Neves *et al.*, 2016)(Ali *et al.*, 2015)(Yunchuan *et al.*, 2014).
Confidentiality, Integrity, and Availability	Robust security mechanisms are of paramount importance to maintain the confidentiality, integrity, and availability, of the stored data (Malik *et al.*, 2014)(K.Pranathi, 2016) (Yunchuan *et al.*, 2014)(Bratterud, Happe and Duncan, 2017)(Beacham and Duncan, 2017). This integrated parameter also points towards the possibility of other external entities or applications which may hack into a cloud on behalf of the actual ones (Malik *et al.*, 2014)(Bratterud, Happe and Duncan, 2017) (Tsz Lai, Trancong and Goh, 2012). This is more specifically at the IaaS service model (Malik *et al.*, 2014).
Firewalls	The data should be kept away from the hackers by properly deploying strong firewalls, external or internal, and monitoring systems for all the related applications and interactions (Malik *et al.*, 2014)(Solanki and Shaikh, 2014)(K.Pranathi, 2016)(Bratterud, Happe and Duncan, 2017)(Beacham and Duncan, 2017).

- Shared Data: Evolving from the fact that various computational resources over the Internet, which is an open network, are usually shared among different users, the number of risks attached to the cloud environment is enormous. For instance, data leakages may occur whenever data are transferred to unauthorized parties. (Naveen and Harpreet, 2013) (Subbiah, Muthukumaran and Ramkumar, 2013)(Chraibi *et al.*, 2017)(Solanki and Shaikh, 2014)(Qin *et al.*, 2018)

To confirm that information is only accessed by specific individual users who have permission to access it, access control procedures are required to govern who gets to see what and how sensitive data move around the cloud (Malik *et al.*, 2014)(Solanki and Shaikh, 2014)(Tsz Lai, Trancong and Goh, 2012). Furthermore, it's important that each cloud service provider (CSP) maintains his own strict policies for hiring practices, usage and access rights, and rotation of individuals.

- SLA: The trusted data boundaries of information security are usually not clearly mentioned in the cloud provider's SLA that dictating the different responsibilities between the customers and the CSPs (Chraibi *et al.*, 2017)(Solanki and Shaikh, 2014)(K.Pranathi, 2016). In this respect, these SLAs do not usually mention any emergency response plan for instantly resolving an occurrence of an incident (Tsz Lai, Trancong and Goh, 2012).
- Data Locality: Clients only shift their data to the cloud environment which is actually a web-based virtual space simulated and abstracted over the Internet as a real space instance. Then, they reuse these data without having a pronounced idea where their data are. (Naveen and Harpreet, 2013)(Subbiah, Muthukumaran and Ramkumar, 2013)(Yunchuan *et al.*, 2014)(Malik *et al.*, 2014)
- Data Mobility (Data Location/Relocation): CSPs usually have contractual agreements with each other to use the resources of one another (K.Pranathi, 2016)(Tsz Lai, Trancong and Goh, 2012). This often addresses the ever-growing important concerns that have security relevance: Are users' data portable? Should the data stay in a particular location or reside on a given known server? Should the CSPs precisely inform their customers where their data reside or migrate from a cloud to another? (Tsz Lai, Trancong and Goh, 2012)(Ali, 2016)
- End-users' interactions with cloud-based data, mainly businesses contemplating public cloud adoption, may dramatically increase the potential of cyber-attacks and raise privacy and security issues (Sareen, 2013) (Naveen and Harpreet, 2013).
- Migrating data to the Cloud means that the cloud itself may be delegated (i.e. outsourced) to a third-party provider who will also have complete control over the cloud-hosted data. For instance, in a health management system (HMS) environment, the data management duties may go further than the hospital or the clinical center to other more supportive parties: the cloud service provider itself, and, in a more active way, the outsourced service provider. (Naveen and Harpreet, 2013)(Subbiah, Muthukumaran and Ramkumar, 2013)(Chraibi *et al.*, 2017)(Yunchuan *et al.*, 2014)
- The needed cloud services are provisioned or de-provisioned easily and rapidly as needed to satisfy customer demands (Anne-Lucie *et al.*, 2017)(Manju and Sadia, 2017).

Down to the above-stated limitations, it is important to realize that the cloud technology is still not able to reach its expected intended future of protection (Chowdhary and Rawat, 2013)(Bratterud, Happe and Duncan, 2017)(Yunchuan *et al.*, 2014). This causes some critical security concerns; to name a few, secure data storage, secure computation, secure transmission, data privacy, data protection, robust data availability, usage privacy, network security, location privacy, identity management, etc. (Yunchuan *et al.*, 2014)(Yashodha Sambrani, 2016). Therefore and to avoid unauthorized access and data breaches, studying all the security requirements regarding privacy safety and information confidentiality is essential and the CSPs with the national policymakers must be assured that all security mechanisms are provided to get the precise information to the exact people at the right time and place (Ali *et al.*, 2015). For instance, there is a need for new security protocols to govern how computers communicate with each other over the network and control the electronically stored information (Al-Ta'ee, El-Omari and Kasasbeh, 2013).

Being more specific, the already challenging security problem is too complicated even more with the need of taking into account the developing of all different cloud-related policies, practices, strategies, and standards (Sommerville, 2015)(Essandoh, Osei and Kofi, 2014)(Sameera and Iraqi, 2017)(Tsz Lai, Trancong and Goh, 2012). What's more, each enterprise may establish its private security and CC technology policies, under which their owned resources can be established, configured, accessed and managed remotely (Sommerville, 2015)(Laverty, Wood and Turchek, 2014)(Tsz Lai, Trancong and Goh, 2012).

12.10 Shortage of trained and highly-skilled expertise

Since the installation process of this new style of the distributed system is disconnected from the actual hardware infrastructure (Anne-Lucie *et al.*, 2017)(Prajkta and Keole, 2012) and the end-to-end process execution cater to horizontal and vertical scaling, the current and upcoming cloud-related applications should be different from the unclouded in-house applications (Ali, 2016). And in order to support clouds in providing innovative-meaningful services through the Internet, a greater number of highly trained technical staff is required to operate and maintain the clouding facilities (Abdu *et al.*, 2017)(Ali *et al.*, 2015) where their knowledge, IT skills requirements, and experience are different than the conventional ones (Karim *et al.*, 2017)(Essandoh, Osei and Kofi, 2014). This is definitely regarded as one of the key players for the successful execution of CC and its related applications (Karim *et al.*, 2017)(Essandoh, Osei and Kofi, 2014)(Ali *et al.*, 2015).

In reality, the lack and inadequacy of the required highly skilled cloud workers are actually due to, but is not limited to, the following hard limitations:

- CC is comparatively a new market phenomenon and most of its experts are generally qualified over conventional computing environments and nonqualified over the cloud environment that may have an enormous number of users, large data volumes, and great performance-loads (Kuo *et al.*, 2014)(Acharjya and Ahmed, 2016)(Karim *et al.*, 2017). For instance, many programming approaches and computational techniques that work well with small-size data do not scale up to fit voluminous data (Acharjya and Ahmed, 2016) and they should be refactored and modified to support the CC features (Gholami, Daneshgar and Beydoun, 2017).
- Furthermore, they have not enough knowledge and adequate understanding of the cloud-based business models that span selling a software product into selling full-supported services that may be dynamically distributed across multiple countries or continents across the world (Ali, 2016)(Al-Ta'ee, El-Omari and Kasasbeh, 2013)(Al-Ta'ee, El-Omari and Kasasbeh, 2013)(Al-Ta'ee, El-Omari and Kasasbeh, 2013)(Tsz Lai, Trancong and Goh, 2012).
- The lack of ICT training programs in the field of CC and its associated subjects, like Big Data and e-payments, requires good contemporary training and an efficient educational system to allow students to witness and practice the "real" CC world (Karim *et al.*, 2017)(Essandoh, Osei and Kofi, 2014)(Ghwanmeh, El-Omari and Khawaldeh, 2015).
- Highly skilled trained and skilled labors comparatively request higher salaries in the IT market and only large enterprises could afford this (Gumbi and Mnkandla, 2015)(Al-Ta'ee, El-Omari and Kasasbeh, 2013).

To end with, the future educational plans, though, should be certainly changed to reflect this shortage by offering a pool of skilled labor. This is almost certainly for heightened education (Essandoh, Osei and Kofi, 2014).

12.11 The Realism of CC Testbed Environment

Cloud-based software development should be oriented to fully exploit the essential characteristics of the CC environments such as distributed programming, multi-core parallel execution, and so forth (Kaur and Chana, 2015). However, programming in conventional computing scenarios, like single-core sequential execution, might not adequate for this new paradigm. On the other side, since they are not initially designed for handling the cloud environment and all its characteristics, the adaptation of the existing applications into the respective cloud framework is not an easy and direct task. This requires highly skilled IT specialists to drive the cloud adoption process (Foster *et al.*, 2008)(Gumbi and Mnkandla, 2015)(Sommerville, 2015) (Akshatha and Manjunath, 2016).

Related to the different CC use cases of the testing environment, there is some missing knowledge about appropriate testing parameters (Chraibi *et al.*, 2017). Another a closely related matter, since more and more real-world applications and services are being deployed on clouds, there is a need to test cloud-hosted solutions in real-life testbed environment outside of research labs with larger repeatability workloads that almost nearly more evidence to the realistic scenario and within the realm of CC adoptions (Maruf and Albert Y., 2017)(Boudi1 *et al.*, 2018). Therefore, the use of realistic captured data through data-intensive experiments is more suitable for reliable cloud-based solutions than that of the simulated data which are usually based only on the academic experiences that may be far away from the CC realm (William *et al.*, 2017)(Ali, 2016).

12.12 Programming Models

Before the World Wide Web (WWW), applications were conventional and running only on local single computers or computer clusters (Sommerville, 2015). And because the communications were local, these computers were only accessible from within the same enterprise by using special-purpose user interfaces (Sommerville, 2015). Furthermore, most programming approaches, algorithms, and data structure techniques were still oriented towards the unclouded in-house applications and not prepared for exploiting the cloud web-based characteristics which have a different hosting environment (Tsz Lai, Trancong and Goh, 2012)(Ali, 2016). But the ways of which these applications are structured, organized, and viewed have been varied considerably with the advent of the web to fully exploit the new web-based frameworks, especially the cloud-based one (Bertolino, Nautiyal and Malik, 2017).

12.13 The Complexity of the Licensing Model

Unlike old-fashioned software, tracking the CC software licensing model and its associated hardware becomes more challenging; this may have a number of reasons, such as (Anne-Lucie *et al.*, 2017)(Sommerville, 2015):

- Cloud services are hosted on the cloud and not installed on users' local machines. They are also shared among users and follow a subscription model as an alternative for buying specific copies of the software.
- The complexity of the software lifecycle leads to some understandable relations between software usages, associated computer hardware, and the associated licensing scheme.
- The multiplication of actors leads to a complexity in the licensing model.

Altogether lets some software vendors develop their applications without having a preference to deliver them with a utility-pricing basis, which is a fundamental element of the cloud delivery model. Rather, they insist on the old-fashioned pricing model, such as customer-procured basis or self-service that uses fixed payment perused service. (Kiranjot and Anjandeep, 2014)

12.14 The Lack of Financial Visibility

CC frameworks follow the model of operational expenses (OpEx) which means the firm pays only for the usage of the cloud services. While the traditional model uses capital expenses (CapEx) which means the organization buys hardware and software and pays full cost ahead of time. The nature of OpEx is harder to predict than CapEx since it depends on the usage which is not easy to plan ahead. As a result, a problem of lower financial visibility may occur. (Anne-Lucie *et al.*, 2017)

12.15 Electricity

To ensure a constant current, the adoption process of the CC requires high-quality electrical energy that all the time connected. Sure, this regular durability is not the standard norm of most developing countries that suffer from not only frequent power outages but also unstable electrical energy. (Essandoh, Osei and Kofi, 2014)(Ghwanmeh, El-Omari and Khawaldeh, 2015)(Al-Ta'ee, El-Omari and Ghwanmeh, 2013)

13. Internet of Things (IoT) & Big Data

Along the journey toward "automation-of-everything-around-us" or as formerly known "smart-world", and with the increasing availability of broadband networks, researchers and experts from both research and industrial communities eventually invents the concept of Internet-of-Things (IoT) in their ways of maximizing existing resources and improving service delivery (Al-Fuqaha *et al.*, 2015)(Acharjya and Ahmed, 2016)(El-Omari and Alzaghal, 2017). As a result and with the intention of meeting the competitive global market, most software houses are increasingly striving to make their different web-enabled services on IoT bases (Al-Fuqaha *et al.*, 2015). While IoT brings altogether people, technologies, data, and things, the core concept is cycling around forming and making use of the networked objects by connecting physical objects and devices that have the capabilities of identifying, sensing, gathering, computing and interchanging data (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Miorandi *et al.*, 2012)(El-Omari and Alzaghal, 2017). Then, these connected objects exchange data and send them to other systems for processing and visualization which will be later presented as information and knowledge (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Miorandi *et al.*, 2012)(El-Omari and Alzaghal, 2017).

Even though IoT is one of the most widely debated subjects, it is also one of the least agreed subjects (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Alessio *et al.*, 2014). Until now, IoT is still evolving and has no formally an in-depth and holistic accepted definition or clear conceptualization; though it has not yet been standardized (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Alessio *et al.*, 2014) where different stakeholders have different perspectives for looking at. It is an evolving orchestrated technology that interacts with the other existing technologies. But in its simplest form, IoT is within this context of joining digital and physical entities by compacted bridges (Miorandi *et al.*, 2012).

One of the most common definitions of this prominent research sector is defined by Daniele Miorandi et al. in (Miorandi *et al.*, 2012) as:

"*an umbrella keyword for covering various aspects related to the extension of the Internet and the Web into the physical realm, by means of the widespread deployment of spatially distributed devices with embedded identification, sensing and/or actuation capabilities.*"

Back-and-forth, IoT will not be tacit as separable systems but as an integrated infrastructure of groups of distributed wireless-linked sensors and actuators upon where many useful high-agile applications and services are linked together and run (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Miorandi *et al.*, 2012)(Khajenasiri *et al.*, 2016)(Maria G. Koziri and Loukopoulos, 2017). Furthermore, there will be systems-of-systems that synergistically interact with each other to form completely new and unpredictable services through the Internet (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Miorandi *et al.*, 2012).

Overall, the philosophy behind IoT is not new concepts to the IT world and there is a belief that it is a problem specific. What are new are the expanding options and capabilities around coining another new business model in which sensing and actuation in the form of an IoT are packaged together and delivered to customers (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Miorandi *et al.*, 2012). To tell the truth, IoT does not revolutionize the field of computing, but it adds a new face and a required field to the ICT industry.

As a final point, the research community of IoT frequently uses the terms: “objects”, “smart objects”, “nodes” and “things” interchangeably to give the same mean (Alessio *et al.*, 2014). Additionally, some researchers address this technology with the Internet of Everything (IoE) to give more emphasize on the Internet-enabled smart objects and their ubiquitous existence.

13.1 IoT Functions

As a matter of fact, IoT is indeed a new frontier paradigm shift on the way of using the Internet in delivering IT services (Miorandi *et al.*, 2012). Hence, besides using the Internet for connecting end-user devices, it is also used for interconnecting objects which may be either physical or digital entities (Patel, Patel and Panchal, 2017). While digital entities are considered to do some norm of computational jobs, physical ones can be in the form of sensors and actuators, (Patel, Patel and Panchal, 2017). On the other hand, these objects are either interconnected among themselves or/and with other entities in the network or/and with end-users (i.e. humans) (Miorandi *et al.*, 2012)(Patel, Patel and Panchal, 2017). Usually, besides that these objects are combined with the capabilities of capturing data, they are also supported with the capabilities of communication for the purpose of sending these data (Patel, Patel and Panchal, 2017).

As in the case of CC, IoT is also a multi-dimensional domain and can be directed for various sets of goals and objectives. Since the connected smart devices in the form of the IoT platform will become a utility, the ICT market paradigm is shifted toward the developed smart world. (William *et al.*, 2017)(Patel, Patel and Panchal, 2017)(Maria G. Koziri and Loukopoulos, 2017)(Marques *et al.*, 2018)

While any narrow definition of IoT is no longer appropriate, this paradigm has the following core tenets which make smart objects being smart to distinguish this discipline from other research areas:

- **Addressable:** On the basis of standard communication protocols, objects should be uniquely addressable. When an object is assigned an IP address, it can be used within the planet to extract knowledge and share it with other pooled resources. Since customer demands for utilizing smart objects are exponentially increased, there is a need for utilizing a larger addressing space to meet customer demands. This leads to using IPv6 to overcome the restrictions, unavailability of new addresses, and limitations of IPv4. (Al-Fuqaha *et al.*, 2015)(Patel, Patel and Panchal, 2017)
- I**ntegration of Information and communication:** Information and communication systems will be integrated and invisibly embedded in the IoT environment in one form or another. (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Miorandi *et al.*, 2012)(Marques *et al.*, 2018)
- **Sharing information and coordinating decisions:** since the smart-physical objects “talk” together, they IoT enables them to “see”, “hear”, “think”, “identify”, and then “perform jobs” (Al-Fuqaha *et al.*, 2015). In other words, IoT shed light on a vision for a smart connected world that has a higher reliance on shared usage (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Miorandi *et al.*, 2012). This interconnectivity allows objects to exchange data with the other neighboring objects (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Miorandi *et al.*, 2012).
- **Sensors’ interaction**: In order to observe the environment, different types of wireless-linked and semi-autonomous sensors are widely embedded inside both remote-sensing platforms and sensor networks (Marques *et al.*, 2018). An actuator, on the other hand, is a device or a mechanism used by any control system to switch the state of the system’s environment onto the next (Kiranjot and Anjandeep, 2014). For instance, an actuator may control television by changing its state from “open” to “closed” (Kiranjot and Anjandeep, 2014). So, actuators switch the environment according to the information provided by sensors (Kiranjot and Anjandeep, 2014)(Miorandi *et al.*, 2012). While sensors and actuators may interact among themselves through the sensing and actuation capabilities, they may interact with other entities in the network and/or with end-users (Maria G. Koziri and Loukopoulos, 2017). Thus, these smart objects can build networks of interconnected entities that interact with the local environment by exchanging worthy data about the

environment which will be later presented for further processing. In this respect, these smart objects can also create electronic services with or without direct human intervention (Miorandi *et al.*, 2012)(Maria G. Koziri and Loukopoulos, 2017).

13.2 IoT Elements (Al-Fuqaha et al., 2015)(Perera et al., 2014)

Again as in the case of CC and close related to the framework of the OSI model, especially transport and network layers, IoT can be encapsulated to have a five-layer architecture where each layer covers one or more facilities and the bottom two layers contain the actual physical components of hardware infrastructure. (Patel, Patel and Panchal, 2017). Besides that this rationale of layers is clearly described in Figure 14, a layer-to-layer comparison is shortly described below:

- Perception Layer: as it exists at the lowest hierarchy level of the IoT architecture, this layer usually comprises, among others, sensors, actuators, digital cameras, bar code readers and scanners, GPS, and RFID systems. By using these interconnected devices, this layer is used to recognize, collect, and extract refined data from the defined environment.
- Network Layer: above the perception layer, this layer exists and forwards to the Internet the already extracted data streams from the underlying perception layer. This layer usually encompasses the different network management units, like switches and routers, and other supporting components.
- Middleware Layer: this underlying layer works as a middleware for information processing. It exists between the Network Layer and the application layer. It generates outputs to be used as inputs to the application layer. The role of this layer includes, among others, the storage and processing of data. The output of this layer is passed forward to be handled by the application layer.
- Application Layer: upon the user needs, the generated data of the middleware layer are presented as useful information by an abroad set of innovative-meaningful services.
- Business Logic Layer: on the top of the said layers, the business layer logically exists to initiate myriad opportunities for producing money.

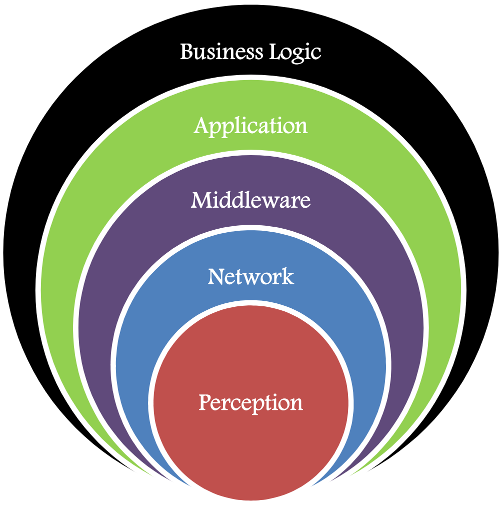


Figure 14. Hierarchy Levels of the IoT Architecture.

It is generally accepted that the following six elements are used for examining IoT from the perspective of its elements:

1. Object Identification: to distinguish objects from each other, each smart object should be uniquely identified by an object id, or at least to be identified as belonging to a given class of objects that are uniquely identified (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014).
2. Sensing: sensors are wide-ranging from the small-tiny-light ones to the smart appliances. While some sensors are dummy ones, others are enhanced to generate smart decisions. Besides they are getting cheaper than ever before, they are becoming more powerful and much smaller in size. Consequently, it is predicted that the numbers of deployed sensors will flare rapidly over the next few coming years for both personal and business needs. (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Alessio *et al.*, 2014)

3. Communication: the indispensable part of IoT is the smart interconnectivity of objects that allow them to collaborate and exchange data with the cloud-based application existing at the back-end clouds. Most modern systems typically contain sensors and actuators to realize and change their states. While, sometimes, sensors and actuators should be disjoint and protected from each other, it makes sense, at many other times, to share data and information between themselves and to work together as one enhanced system. Although sensors could be homogeneous or heterogeneous, different networks of different technologies and protocols could be connected together in an integrated and automated fashion. Even though most deployed sensors nowadays are wireless, sensors could be connected between themselves using either wireless or wired technologies. (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Kuo *et al.*, 2014)(Acharjya and Ahmed, 2016)
4. Computation: processing units, like microprocessors and microcontrollers, and different high-quality software applications represent the "brain" part of IoT.
5. Knowledge Extraction from Big Data: it refers to the ability to extract knowledge smartly by different machines (Neves *et al.*, 2016). Since there is a need for extracting smart knowledge, there is also a need for embedding usable intelligence into everything of the environment by using more smart sensors (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014) (Alessio *et al.*, 2014). As there may be innumerable sensors and each one may generate massive-scale data, the size and complexity of these underlying datasets are considered as "Big Data" (Acharjya and Ahmed, 2016)(Dhabhai and Gupta, 2106)(El-Omari and Alzaghal, 2017)(Patel, Patel and Panchal, 2017). This big data, however, has no value unless it has been specifically collected, analyzed and then understood to extract usable forms of knowledge by the helping of the CC (Acharjya and Ahmed, 2016)(Neves *et al.*, 2016)(Alessio *et al.*, 2014)(Patel, Patel and Panchal, 2017).

   To get around this problem, all these operations on this big data should be done in a reasonable way that ensures not to impact the overall-potential performance measure in any means (Alessio *et al.*, 2014). In the course of this work, the cloud framework may be the best alternative or just the only viable option for its fairly cost, on-demand unlimited storage capacity, and time-saving (Khajenasiri *et al.*, 2016)(Kuo *et al.*, 2014)(Acharjya and Ahmed, 2016)(Akshatha and Manjunath, 2016)(Neves *et al.*, 2016)(Dhabhai and Gupta, 2106)(Alessio *et al.*, 2014). And so, it is clear that the cloud is here to complement the Big Data, not to replace it (Patel, Patel and Panchal, 2017).
6. Services: different well-designed and high-agile services are required to be provided according to the extracted knowledge (Kuo *et al.*, 2014)(Acharjya and Ahmed, 2016).

13.3 Shared Features among CC, IoT, & Big Data

Not surprisingly, IoT is actually another advance phase that implies another further step in the evolution of the CC and the Internet that it has already been (Patel, Patel and Panchal, 2017). They are both closely tied to the never-ending advances in communications and network technologies and with the other relevant technologies (Patel, Patel and Panchal, 2017). While they have a number of distinct elements that interact with each other, each one of them can exist on its own. Figure 15 is a schematic diagram that depicts the relation between them (Khajenasiri *et al.*, 2016)(El-Omari and Alzaghal, 2017).

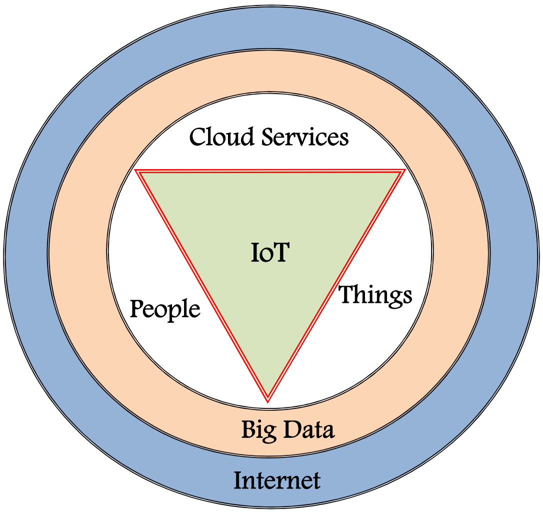


Figure 15. IoT, Big Data & CC Schematic Diagram

As soon as data have been captured in the clouds it stored, archived, analyzed, classified and organized, interpreted, and turned into actions. Actually, data aggregation, processing, and analytics are often performed in the clouds directly in real-time mode. Eventually, entrepreneurs and different customers are increasingly benefited from the innovative knowledge extracted from collected data no matter how the collected data are uploaded and managed remotely, how these web-enabled services are intelligently created, how they communicate with the cloud-hosted platform, nor where they are running these services. (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Miorandi *et al.*, 2012)(Arora *et al.*, 2017)(Patel, Patel and Panchal, 2017)

From an IoT perspective, since everything might possibly generate events, the cloud-based systems need to be enhanced to support intelligence. In the course of this problem, the different applications should go beyond conventional applications into sophisticated ones that embed intelligence into its paradigms.

13.4 IoT Adoption Roadblocks

But like everything that appears too good to be true, some challenges face the vision of the IoT. However, many of the management roadblocks associated with the IoT vision are nearly the same challenges associated with CC. This section identifies the roadblocks that should be assessed and minimized or removed, in order to confirm the success of IoT solutions. Without this, these roadblocks might be the dark side that leveraging the widespread adoption of this integrated technology. Here, the focus is on the most critical roadblocks:

1. **IoT Embedded Devices**: while the traditional embedded devices usually perform only one function on the same piece of hardware, the new IoT embedded ones perform multi-functions simultaneously. This growth of resource usage causes a huge amount of big data and, in turn, brings considerable unwanted overheads that impact the IoT performance. Furthermore, the traditional virtualization layer that maps the virtual devices to the actual physical ones may no longer sufficient to upkeep these needs and then there is a need for a new virtualization technique. (Marques *et al.*, 2018)(Boudi1 *et al.*, 2018)

   On the other hand, some types of embedded devices aren’t supported by the computer hardware virtualization, and, therefore, they need exceptional treatments (Marques *et al.*, 2018).

2. **Heterogeneity of Infrastructure Elements:** the scope of Internet-connected resources has grown beyond expected ranges in terms of sizes, scales, ranges qualities, brands, protocols, and compatibilities. This significantly increases the need for better management of these objects, as well as of the computer architectures that support them. (Alessio *et al.*, 2014)(Marques *et al.*, 2018)(Boudi1 *et al.*, 2018)

3. **IoT Environment:** While most of today’s enterprises have an increasing heterogeneity of the technologies related to sensing and actuation devices, most of these interconnected devices are originally designed for Intranet and not for the Internet of Things. However, the adaptation of these existing devices into the respective IoT environment and moving them to a real Internet of Things might not be an easy and direct task. Besides that, the degree of the density of sensing and actuation coverage is a further challenge to be tackled. (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Miorandi *et al.*, 2012).

4. **Big Data:** It has been postulated that sensors and actuators continuously generate solar amounts of real-time raw data streams that are collected annually to petabytes of data, which are ultimately non-structured or semi-structured big data. When the entire cost of ownership is considered as a pragmatic measure, dealing with big-data solutions is comparatively costly in particular for industries that cannot afford dedicated big-data systems. This is because big-data technologies require relatively costly hardware solutions as well as software and highly skilled workers. To overcome this problem with a feasible solution, borrowing such services from the cloud, mainly public cloud, is required to make cost balancing. Therefore, better management of resources as well as of the infrastructure that supports is attained by cloud. (Devang Swami, 2016)(Kuo *et al.*, 2014)(Acharjya and Ahmed, 2016)(Al-Fuqaha *et al.*, 2015)(Neves *et al.*, 2016)(Dhabhai and Gupta, 2106)(Alessio *et al.*, 2014)

   On the other hand, since the data are generated at a drastic pace, there will be a necessity to develop many programming approaches and computational techniques that imply collecting, storing, archiving, accessing, processing, visualizing, sharing, analyzing, and presenting these data as usable forms of knowledge to the right people at the right time (Neves *et al.*, 2016)(Alessio *et al.*, 2014). Afterward, specific actions may be taken based on this in common knowledge by a wide range of business (Al-Fuqaha *et al.*, 2015)(Neves *et al.*, 2016).

5. **Communication Technology:** as the IoT implicates by definition producing a huge amount of information sources (Alessio *et al.*, 2014), the capabilities of wireless-linked technologies, such as Radio-Frequency identification (RFID) and remote sensing, should be essentially raised enough to meet the IoT features (Acharjya and Ahmed, 2016).

6. **IoT Experiments:** Most of the CloudIoT techniques/algorithms are designed, constructed, and applied over simulation-based approaches without exposing them to extreme abnormal conditions where they may not react in a similar reliable mode or even function in the way expected (William *et al.*, 2017)(Boudi1 *et al.*, 2018). And, most IoT experiments were mostly conducted in research labs that nearly different than the real environment where there may be some mismatch between the reality and the theoretical results (William *et al.*, 2017)(Boudi1 *et al.*, 2018). In turn, these labs may not initially design for fully exploiting the IoT environment and may have some missing environments to conduct experiments on. (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014) (Miorandi *et al.*, 2012)(Maruf and Albert Y., 2017)

   Moreover, there is definitely a notable difference between the theoretical-academic world and the real-life practice world. That is, solving of lab-problems without exposing them to real use is subject to guesswork and experimentation (William *et al.*, 2017). On top of all that, most of these labs are with only an academic or theoretical vision and may have been subject to some research setting with little to no external support available.

   To overcome this problem and to ensure that the right Quality of Service (QoS) is delivered, there is a need to test IoT solutions on the outside of research labs on a broad range of possible scales that almost nearly the real-world environment (Boudi1 *et al.*, 2018)(Patel, Patel and Panchal, 2017).

7. **Systems Integration:** In the near few coming years, there will be truly trillions of things (i.e. objects) that are connected to the Internet via sensors and actuators. There will be systems-of-systems that synergistically interact with each other to form completely new and unpredictable services that try to highlight an in-depth vision for a smart world. (Devang Swami, 2016)(Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)

   While there may be multiple systems in IoT, each individual system could have its own strategy, assumptions, and configuration to further control the real physical world variables. However, each individual special setting may differ from the others. Therefore, unless careful considerations are been taken, it will be a serious challenge and perhaps the most critical challenge among the other challenges. (Devang Swami, 2016)(Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)

8. **Energy-Optimized Solutions:** since almost all IoT devices are ordinarily supplied by batteries, one of the key components that have greatly limited the development of this field is the need of ten thousands of small battery-powered nodes. Therefore, Power saving emerges as one of these concerns. (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014) (Miorandi *et al.*, 2012)(Marques *et al.*, 2018)(Patel, Patel and Panchal, 2017)

9. **Security Limitations:** However, due to some security limitations, IoT is still not able to reach its expected intended future of protection (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Miorandi *et al.*, 2012)(Boudi1 *et al.*, 2018). Additionally, the lack of efficient ICT policies, systematic strategies, practices, and standards that tackled IoT is rated as the biggest barrier among the above-stated challenges (Alessio *et al.*, 2014)(Ghwanmeh, El-Omari and Khawaldeh, 2015). Being more specific, the already security problems for providing authorized and secured access to the data are too complicated even more with the need of taking into account the developing of all different IoT-related policies, strategies, and standards (Al-Fuqaha *et al.*, 2015)(Perera *et al.*, 2014)(Miorandi *et al.*, 2012).

10. **Networks & Internet:** The key physiology behind real-time (i.e. on-the-fly) applications is to essentially return the computation results easily in timely regulatory submissions even if there are vast quantities of data and jobs (Acharjya and Ahmed, 2016)(Alessio *et al.*, 2014)(Arora *et al.*, 2017). Poor Internet access and broadband connectivity are definitely fatal mainly for real-time or even semi-real-time applications (Acharjya and Ahmed, 2016)(Khajenasiri *et al.*, 2016). Therefore, there is a need for high-speed Internet connectivity with high availability and reliable solid network infrastructure that guarantees high-quality data access to authentic users anywhere and anytime (Khajenasiri *et al.*, 2016)(Essandoh, Osei and Kofi, 2014).

    In effect, networks are usually subject to some restrictions and limitations, such as bandwidth, input/output throughput, latency, and response time. Furthermore, the number of smart objects deployed in different places around the world is increasing at a rapid pace which may exceed the availability of the usual Internet. (Boudi1 *et al.*, 2018)

14. Digital Image Processing, IoT, & CC

In reality, exaggerative numbers of millions of information-rich digital images and videos are generated in various digital IoT devices every single hour (Qin et al., 2018)(Yuzhong and Lei, 2014). Not only that, most of the datasets held within the image processing technology contain huge amounts of mass data with rich-mix multimedia combined at the same time with an increasing level of details that are recognized as big data (M

Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018). However, these datasets may be distributed to different participants across multiple places all over the world (Yuzhong and Lei, 2014)(Karim *et al.*, 2017). Another issue, the majority of these systems are relatively sophisticated and, in turn, require scalable computational power and high-level communicational capabilities (Qin *et al.*, 2018)(Yuzhong and Lei, 2014)(Mirarab, Fard and Shamsi, 2014). Besides the above, there is still one more issue that needs to be considered where many of these image processing techniques require real-time or near-real-time reading/writing access for processing to take place (Yuzhong and Lei, 2014). And, above all, the underlying hardware and software involved within these digital imaging-based systems are very expensive and involve time-consuming and so complicated procedures of installation, maintenance, and support (Mirarab, Fard and Shamsi, 2014). Due to these stated reasons and as most image processing applications could be applied remotely, there is an utmost need to integrate the field of image processing within the CC environment which is truly the most tolerable place to process the big data (M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018)(Qin *et al.*, 2018)(Mirarab, Fard and Shamsi, 2014). This three-field integration (CC, big data, and image processing) has recently attracted extra further attention from both academic and industrial communities and becomes as a fundament high-productivity platform for the image processing distribution (Mirarab, Fard and Shamsi, 2014)(Kang and Lee, 2016). As a consequence, there are some domain-specific clouds to fulfill the image processing storage and computing power demands like the cloud of PVAMU (i.e. Prairie View A&M University) and the OpenStack (Kang and Lee, 2016). Different than the other clouds, these domain-specific clouds should at least assure the followings:

- Build a robust bridge with user-friendly and interactive interfaces to strongly enable the end-users to break into the so complicated up-to-the-minute computer architectures (Yuzhong and Lei, 2014)(Kang and Lee, 2016).
- An open and shared environment that provides a novel PaaS and supports the most popular-used programming languages and models for both image processing developers and researchers as C, C++, Python, Matlab, and Java. While software developers can use these clouds as an image processing production environment, researchers can use them for their research studies. To simplify their programming efforts, all actors, including developers and researchers, might use their familiar programming languages without having to study any domain-specific language (DSL). (Yuzhong and Lei, 2014)(Mirarab, Fard and Shamsi, 2014)
- Built-in support for multilevel parallelism and distributed operations: Software developers can design and implement their own image processing algorithms with very limited knowledge about using parallelism and distributed operations without worrying about the other specific details. So, they can be closer to the actual problem domain at a distance from the lower-level details like the details of the parallel distributed database. (Edlund, 2012)(M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018)(Yuzhong and Lei, 2014)
- Large mass storage to store images as well as videos (Yuzhong and Lei, 2014).
- The capabilities for processing images with acceptable performance (Yuzhong and Lei, 2014).

However, it is a challenge to meet all the above-discussed image processing challenging requirements, especially that practitioners of this area usually use different programming languages, as an example, in designing and deploying their image processing algorithms (Yuzhong and Lei, 2014).

The special cloud of PVAMU is a high-performance image processing portal that has a data-center fabricated on the top of many High-Performance Computing (HPC) clusters. It provides a farm, based on “Apache CloudStack” which contains a large number of highly-available VMs presented to the end-users as two-widely-utilized-service models: IaaS and PaaS. While the first highly-scalable IaaS service model is based on the Apache CloudStack, the second one, namely PaaS, is based on combining the so-popular image processing library OpenCV within the Hadoop platform. Their large-scale image-processing cloud is a domain-specific platform that can be utilized for implementing a set of well-defined image processing algorithms in parallel and, moreover, conducting image processing scientific researches. For efficiency reasons, their platform (i.e. PaaS of PVAMU cloud) supports multiple programming languages with a flexible and upgradeable environment such as C/C++, Ruby, Python, and Java. As previously shown in Table 2, PVAMU is also an open-source cloud that supports the following operating systems: Windows, Linux, Mac OS, iOS and Android. Going further, there are other clouds that are based on the Hadoop platform like HipiImage (HIPI) and MapReduce. The HIPI cloud uses the HipiImageBundle (HIB) as an input file format for assembling and packaging the set of images altogether into a single big file along with the required metadata for describing the images’ layouts. Different than HIPI, the MapReduce process uses every original image directly as input without integrating multi images into a single large one. (Yuzhong and Lei, 2014)

The authors of (Kang and Lee, 2016) use a total of ten full-sized virtual servers to build an integrated geospatial image processing environment based on the OpenStack cloud. In order to conduct many empirical geo-based experiments, they used this influential cloud to build a satellite image application (Kang and Lee, 2016). Again, as already shown in Table 2, this cloud is also a freeware-licensed (i.e. open-source) cloud but it only supports the operating system of "Ubuntu server 64 bit" (Kang and Lee, 2016).

By taking medical imaging applications as a widely-spread example of Image processing, any sort of medical data needs to be handled and conveyed from one point of healthcare to another like processing and transmitting medical images to the desired medical specialists (M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018)(Mirarab, Fard and Shamsi, 2014). Above and beyond that, an increasing amount of more-detailed mass data is required to be digitized and stored inside the Electronic Health Record (EHR) for every single patient combined with the rapid growths of diagnostic details (Mirarab, Fard and Shamsi, 2014). Most of these medical data are digital images which are assembled by medical imaging devices such as the devices of X-ray, Computerized Tomography (CT) scan, Magnetic Resonance Imaging (MRI), Ultrasound, Positron Emission Tomography (PET), and other types of medical images (Mirarab, Fard and Shamsi, 2014). Therefore, the cloud migration of the web-based medical applications, as well as the associated datasets, is really an essential part of the healthcare-related data communication systems (M Gokilavani, GP Mannickathan and MA. Dorairangaswamy, 2018)(Mirarab, Fard and Shamsi, 2014).

Ground truth, more proven benefits can be gained by the migration of these image-processing medical systems to the cloud. As a matter of fact, the integrated medical model of image processing and CC is a shared open environment that allows the medical specialists and the other interrelated participants anywhere in the world to join, share their results, and contribute with their opinions related to some complicated medical cases. Besides that this deeper collaboration gives better chances for getting more optimal solutions for these special cases, it creates an open rich-knowledge environment that can be used for training and sharing knowledge broadly among different medical specialists, technicians, and the other close-related participants. Among the proven benefits of this integration and data sharing, novel global trends may be discovered when the medical data are aggregated and tracked over a long time from different geographical places around the globe. Another close interrelated matter, these integrated systems also support the researchers and the data analysis in carrying out various clinical trials and research studies, locally or remotely (Mirarab, Fard and Shamsi, 2014).

Another widely-used example, the unification of the Image processing technology with the fields of CC and IoT can be employed to control and clearance of the traffic jams mainly for emergency situations that are closely bound up with life-saving priorities. The authors of (Vardhana *et al.*, 2018) use this information-rich integration to avoid traffic jams in a very efficient way where the arrival of any ambulance is detected and handled according to its image and siren sound using both object detection and audio reorganization techniques, respectively. To clear the traffic and finding the best track for the ambulance, the route information is passed from one station to the next one along the way to the hospital. Furthermore, by automatic detection of the car license plate number, a violation could be issued automatically by the traffic department to any vehicle that doesn't abide by the announcement of the ambulance's siren (Vardhana *et al.*, 2018).

15. Getting Developing Countries into CloudIoT

As a result of the technological advancements in communication and networking, more and more large enterprises realized that CC is a must for their business ventures (Essandoh, Osei and Kofi, 2014)(Gholami, Daneshgar and Beydoun, 2017) and, therefore, need to expand their centralized systems to take care of the enormous number of users, solar amount of data, and great performance-loads.

Entrepreneurs of developed countries have put significant effort into developing and expanding cloud services through the Internet, and these countries have reached a more mature stage for adopting cloud technology (Essandoh, Osei and Kofi, 2014). Third world countries, by contrast, have a long journey towards reaching a mature CC capability (Essandoh, Osei and Kofi, 2014). As of now, all their concept of cloud technology is only centered on storing their files and backing them up. Hence, there is much work to be done before shifting to the cloud, many opportunities to be considered, many challenges to be encountered, and many expected pitfalls to be evaded in order to fulfill the potential capabilities (Essandoh, Osei and Kofi, 2014). The resulting challenges and opportunities facing third world countries are mostly in common to all other countries (Essandoh, Osei and Kofi, 2014). They are much the same as the rest of the world and notably the Middle East countries.

On the other hand, there is a lack of cloud studies in most (though not all) countries of the developing world that are involved in moving to the cloud paradigm and assessing the potential benefits around this switch. Beyond that, the majority of these technical studies have actually been carried out in the developed countries, including the industrialized ones, without addressing the cloud needs of the developing countries. Learning from the success of adopting cloud technology in the developed countries is vitally important for the developing countries, but taking the same cloud experience of the developed countries and try to apply it as is in the developing countries will not work due to the different environments prevalent. (Essandoh, Osei and Kofi, 2014)

Even with the fact that the Internet connectivity acts as the driver or accelerator towards the adoption and use of cloud services, it is a challenge to have high-speed Internet at affordable prices in most third world countries. The authors in (El-Omari and Alzaghal, 2012)(El-Omari and Alzaghal, 2010b)(El-Omari and Alzaghal, 2010a)(El-Omari and Alzaghal, 2009)(El-Omari and Alzaghal, 2017)(Al-Ta'ee, El-Omari and Ghwanmeh, 2013) have contributed their research efforts on removing the roadblocks and paving the way in the direction of "digital cities" so these information-rich services can highly benefit a wider range of people. They offered general strategic guidelines for the decision and policymakers of Jordan, in particular, and many developing countries' governments in general, to build public wireless network communication infrastructure to connect their cities and to provide free high-speed Internet access to residents as well as visitors. These guidelines when implemented can help in bridging the digital divide and, therefore, increasing the socio-economic development (El-Omari and Alzaghal, 2012)(El-Omari and Alzaghal, 2010b)(El-Omari and Alzaghal, 2010a)(El-Omari and Alzaghal, 2009)(El-Omari and Alzaghal, 2017)(Al-Ta'ee, El-Omari and Ghwanmeh, 2013).

16. How to Choose the Right web-based Services

Since there is no universal model or style that is appropriate for all circumstances and with so many available measured services and deployment models, it is a challenge for cloud users to make a suitable decision among many other critical decisions. So before they go any further, end-users with their computer consultants should consider many indispensable issues that directly influence their sensitive selection about the most appropriate cloud vendor and services that match their needs and requirements (Essandoh, Osei and Kofi, 2014). These various critical issues include, but are not limited to, the followings:

- **Robust Secure Communications:** secure network connectivity has a great effect on selecting the right web-enabled services. For instance, private clouds can offer more secure network connectivity than being retained with public clouds. However, because of some improper configurations, data leakage might also occur within the space instance of some private clouds, but the frequency of occurrence is greater in the public cloud. (Qin *et al.*, 2018)(Yunchuan *et al.*, 2014)(Naveen and Harpreet, 2013)(Malik *et al.*, 2014)(K.Pranathi, 2016)
- **Hosting Service Models:** the decision of selecting the proper model should be considered in the light of technical issues as well as organizational factors. It is clear that there is a need to think thoroughly about which kind of hosting service one should go with to suit his business needs (C.Vijaya and P.Srinivasa, 2016)(Acharjya and Ahmed, 2016).
- **Deployment models:** the selected deployment model should appropriate with the requirements and reflect the customers' needs. There's assured evidence that this selection should be carefully designated on the basis of the firm size, privacy, access, budget, and scope (locally, nationally, or globally) (Ali, 2016).
- **Application Areas:** the customers with their IT consultants should decide whether they need some window-based applications, mobile applications, a special version of the software, or even if their websites need special software. Without any doubt, it's always better to check before using them whether these applications go with all the necessary functions (Chraibi *et al.*, 2017).
- **Cloud Service Provider (CSP):** Now, the good thing is that the list of providers of cloud-based services is growing extremely in size which increases the selection criteria with more opportunities in choosing the most appropriated one. However, since different customers have different needs and requirements, it becomes a challenge to pick up the right web hosting provider who sufficiently suits their current and future needs and who can throw the lifesaving rope promptly whenever needed. For instance, it's a fundamental aspect to know the disaster scenarios that can be followed for recovering the data, and the length of the data recovery process. (Naveen and Harpreet, 2013)(Tsz Lai, Trancong and Goh, 2012)(Subbiah, Muthukumaran and Ramkumar, 2013)(Chowdhary and Rawat, 2013)(Yunchuan *et al.*, 2014) (Malik *et al.*, 2014)(Solanki and Shaikh, 2014)(Ali *et al.*, 2015)

  One critical factor that has a dramatic impact on the selection of the most proper CSP is whose security and privacy mechanisms fall upon the enterprises' security requirements and who has rules, strategies, standards, and procedures in place to mitigate them (Vikram and Bhatia, 2016)(Ali, 2016). Another fundamental factor that can be employed in evaluating the right CSP is who has the right comprehensive documentation for how to solve problems by the users themselves.
- **Upgrading Options**: they should study the various web-hosting upgrading options and how big, or small,

their web traffic volume can be in the future (Venkatachalapathy *et al.*, 2016).

- **Financial Commitment:** Budgeting decision is also a vital factor that directly related to the financial costs of computer hardware adoption and usage, software licensing and other accessing fees (Anne-Lucie *et al.*, 2017), web services and site accreditation, and proper communications infrastructure and capabilities. To get around this problem, the right or most appropriate cloud-like services should be delivered to the right people within the planned and available budget that definitely addresses reality. So, an innovative financing mechanism is required to address this problem.

Thereby, before shifting to cloud technology, any enterprise has to take into account the options in terms of what type of clouding will be the best to attain its business goals. It's very crucial to decide how to use the cloud and balance control and flexibility with cost and agility (Yashodha Sambrani, 2016) and more importantly to involve an independent consulting company to fairly guide this enterprise through the process.

17. General Future Developments

It’s no wonder that the following relevant points, which often are ignored, should be taken at uttermost considerations before undertaking any CC transition:

- **Cloud Globalization:** an increasing number of users of online services lead to an increasing amount of Internet usage, which in turn exponentially increases the loads upon the networks. Unfortunately, networks are constrained in some restrictions and limitations with regards to bandwidth, input/output throughput, latency, variance, scalability, and system performance.

  When there has been seen a notable increase in Internet usage that may exceed the availability of individual providers, a simple cloud provisioning model may be insufficient to support. In turn, there is a need to think thoroughly on highly distributed web services that span multiple service providers. However, this may entail for joining more CSPs from other countries to innovate and extend this limited infrastructure. (Chraibi *et al.*, 2017)(Ali, 2016)(Tsz Lai, Trancong and Goh, 2012)

- **Legal and political support:** Current federation and interoperability support are still too weak to realize and weigh the risks behind CC technology. To take away this critical roadblock, very strict guaranteed laws should definitely be followed to ensure that cloud service providers (CSPs) have standard strategies in place and they are accountable to their customers for any security and integrity breathes associated with the migrating of their data to the cloud. On top of all that, the overall strategies and rules should be continuously reviewed and updated accordingly to reflect this rapidly changing technology. This may require forming new additional ones to guarantee that the system proceeds with laws and regulations of the society and with the ethical necessities. (Yunchuan *et al.*, 2014)(Malik *et al.*, 2014)(Solanki and Shaikh, 2014) (K.Pranathi, 2016)(Essandoh, Osei and Kofi, 2014)(Sameera and Iraqi, 2017)

- **Global IT Standardization:** Since the future IT infrastructure is growing extremely in size and heterogeneity, global IT standardization must exist that doesn't restrict the diversity of resource elements and key services. The main concern regarding CC is that each CSP provides different programming methods and services for data storage and processing that are different than the others. The fact that the different CSPs do not follow an in common standard when providing their services means that much more significant efforts are needed in migrating data and solutions from one CSP to another. In addition interoperability between services from different CSPs is limited. Therefore, this lack of standardization demotivates new users from leveraging the power of CC to its fullest. (Shawish and Salama, 2014)(Alessio *et al.*, 2014)(Essandoh, Osei and Kofi, 2014) (Sameera and Iraqi, 2017)(Ali *et al.*, 2015)

- **Cloud Portability:** Because of technical reasons and/or economic reasons, a point might come at which cloud portability is required as business necessities (Chandra and Neelanarayanan, 2017). It is the ability of migration data, applications, and the whole cloud services between CSPs (i.e. cloud-to-cloud migration) or redeploying the cloud deployment model from one to another, such as between public, private and hybrid clouds (Chandra and Neelanarayanan, 2017).

  Since the users’ data are residing on a provider's cloud, the SLA should contain an exit strategy that enable customers to switch to a new CSP (Chraibi *et al.*, 2017)(Essandoh, Osei and Kofi, 2014). To prevent vendor lock-in, this exit strategy is commonly used when a decision is reached to reinitiate their working from any CSP's clouds to another or even moving out of the CC and come back to the previous traditional onsite deployments (i.e. de-clouding or reverse cloud migration) (Essandoh, Osei and Kofi, 2014). As anticipated from the literature, the lack of global standards and compliance requirements makes it very challenging to switch between vendors (Essandoh, Osei and Kofi, 2014)(Sameera and Iraqi, 2017)(Tsz Lai, Trancong and Goh, 2012).

One thing often ignored is that there are many CSPs today providing data storage services, so to enable data portability they need to ensure that data storage is as universal as possible and not reliant on specific CSP data management systems. Hence, this is making it possible for some forms of data and cloud portability, regardless of the original cloud. (Maruf and Albert Y., 2017)(Essandoh, Osei and Kofi, 2014)

Finally, to conclude the discussion of this section, all the SLA's terms that govern the relationship between CSPs and customers must be tackled appropriately by the customers and fully negotiated with their computer consultants for flexible and favorable terms and conditions (Chraibi *et al.*, 2017)(Laverty, Wood and Turchek, 2014)(Essandoh, Osei and Kofi, 2014)(Sameera and Iraqi, 2017)(C.Vijaya and P.Srinivasa, 2016). Especially that the reality might be a little bit different, these SLA's must definitely guarantee, among others, the following focal points (Essandoh, Osei and Kofi, 2014)(C.Vijaya and P.Srinivasa, 2016)(Ali, 2016)(Patel, Patel and Panchal, 2017):

- The delivered availability should be truly measured. The mere titles such as "five nines, 99.999 availability" or "continuous uptime, 24/7 all year long" are not enough.
- All the expected business requirements are delivered as promised in the SLAs.
- The security functions and privacy measures are met and reasonably rely on them. Again, mere titles and claims such as "a guarantee of a very secure cloud", "100% secure services", "we have the only secure cloud", or "our company is here just only to help you" aren't enough in today's CC services.
- The right Quality of Service (QoS) is delivered to the right people at the right time.
- Pronounced points to guarantee any possible incredible growth of Internet usage.
- Farsighted-reaction strategies that are placed on standby for emergencies and disasters.
- Assuage any possible confusion that might possibly exist between the two sides: CSPs and customers.

18. Conclusion and Future Works

As the in-house hosting is relatively costly, web-enabled computing environment has currently become an enormously widely-used practice among numerous millions of end-users. Electronic services on the fly, as CC, are based on using a web browser over the Internet. It has facilitated shaping today's attitudes to information and communication technologies (ICT) by giving instant access to a broader range of usages through normal Internet connectivity (Taneja, Taneja and Chadha, 2012)(Foster *et al.*, 2008)(Naveen and Harpreet, 2013)(Ghwanmeh, El-Omari and Khawaldeh, 2015).

Cloud concepts are not entirely new and are the natural continuation of distributed trends in previous technologies including cluster computing, GC, and UC. With the timely advances in distributed processing and distributed computing that occurred in the OSs arena, Cloud technology is quickly growing as the best alternative to the ordinary conventional computing and soon become the norm across the globe.

To solve their continually increasing in computing and storage capacities, developing countries' entrepreneurs and industry leaders should rush to adopt CC or at least play a more active role in shaping this new IT-based environment. Besides it gives a deep insight inside the CloudIoT, this research paper enables these enterprises to define reasonable milestones that must be set towards employing this integrated technology at a reasonable cost. It helps in defining what is, and what is not required in the CC roadmap and, perhaps more importantly, reviews the important issues that need to be sufficiently addressed in order to achieve a successful outcome.

It is recommended that developing countries define a comprehensive portal with highly formatted content that offers some online services and includes extensive information about adopting CC in these countries. This portal should be highly available in multi-languages and acts as a practical gateway to all government e-services such as company registration, renewing smart national ID cards, payment for different kinds of donations, and register companies in the ICT field (Ali *et al.*, 2015). Really, this portal may be widely used to provide better quality services for the developing countries; it provides valuable information by enabling collaboration among all levels of stakeholders, ranging from employees to the investors.

To sum up, this paper calls to all researchers and other practitioners all over the globe to increasingly extend their literature towards using CloudIoT technology in developing their countries and to extend their efforts in investigating the potential benefits and barriers of using this coherent trend.

References

Abdu, H. *et al.* (2017). The Integration of Cloud Computing Technology in Academic Library Operations towards Effective Library Services, *International Journal of Research In Science & Engineering (IJRISE), 3*(1), 82-88.

Acharjya, D. P. & Ahmed, K. P. (2016). A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools, *International Journal of Advanced Computer Science and Applications, 7*(2), 511-518. https://doi.org/10.14569/IJACSA.2016.070267

Akshatha, A. & Manjunath, R. (2016). The Open Stack Nova Case for Bcd: Big Data, Cloud Computing and Distributed Computing, *International Journal of Computer Engineering and Applications,* Special Is, 141-148.

Al-Fuqaha, A. *et al.* (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, *IEEE Communication Surveys & Tutorials, 17*(4), 2347-2376. https://doi.org/10.1109/COMST.2015.2444095

Al-Ta'ee, M., El-Omari, N. & Ghwanmeh, S. (2013). Innovative Study Utilizing EMIS in Supporting Participatory Urban Decision Making Process in Jordan, *International Journal of Applied Information Systems (IJAIS), published by Foundation of Computer Science, 5*(2), 1-13.

Al-Ta'ee, M., El-Omari, N. & Kasasbeh, W. Al (2013). *Information Systems Analysis and Design*. 1st edn. Edited by D. A.-M. for Printing-Publishing. Amman, Jordan: Dar Al-Massira for Printing-Publishing.

Alessio, B. *et al.* (2014). On the integration of cloud computing and internet of things, *International Conference on Future Internet of Things and Cloud (FiCloud2014), International Conference on IEEE*, 23-30.

Ali, O. *et al.* (2015). Anticipated Benefits of Cloud Computing Adoption in Australian Regional Municipal Governments: An Exploratory Study, *IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD), IEEE*, 1-17. https://doi.org/10.1109/CSCWD.2015.7231017

Ali, O. M. S. H. (2016). *The Perceived Benefits of Cloud Computing Technology for Regional Municipal Governments and Barriers to Adoption*. The University of Southern Queensland, Australia.

Alwada'n, T. (2016). Cloud Computing Topology: Towards Enhancing the Performance, *International Journal of Computer Science and Information Security, 14*(11), 654-658.

Anne-Lucie, V. *et al.* (2017). Software License Optimization and Cloud Computing, *The Cloud Computing 2017: The Eighth International Conference on Cloud Computing, GRIDs and Virtualization*, 115-121.

Arora, N. *et al.* (2017). Data Virtualization Technology, *International Conference on Latest Concepts in Science, Technology, Management and Humanities (ICLCSTMH-2017),* 430-437.

Ashdown, L. & Kyte, T. (2015). Oracle Database Concepts 11g, *Oracle Press, 2*(E40540-04), 1-472.

Ashdown, L., Kyte, T. & McCormack, J. (2018). Oracle Database 12c, Database Concepts, *Oracle Press, 18c*(E84295-04), 1-654.

Aspen, O. & Kaitlyn, F. (2017). Platform as a Service Effort Reduction, *The Cloud Computing 2017: The Eighth International Conference on Cloud Computing, GRIDs and Virtualization*, 60-65.

Assunçãoa, M. D. *et al.* (2015). Big Data computing and clouds: Trends and future directions, *Journal of Parallel and Distributed Computing, 79*(80), 3-15. https://doi.org/10.1016/j.jpdc.2014.08.003

Australian Government (2013). Cloud Computing Strategic Direction Paper: Opportunities and applicability for use by the Australian Government, *Department of Finance and Deregulation, Version 1.1*, 1-43.

Bardsiri, S. M. H. & Amid, K. (2012). Cloud Computing Vs. Grid Computing, *ARPN Journal of Systems and Software (AJSS), 2*(5), 188-194.

Beacham, N. & Duncan, B. (2017). Development of a Secure Cloud Based Learning Environment for Inclusive Practice in Mainstream Education, *The Cloud Computing 2017: The Eighth International Conference on Cloud Computing, GRIDs and Virtualization*, 135-138.

Bertolino, A., Nautiyal, L. & Malik, P. (2017). Annotated buzzwords and key references for software testing in the cloud, *International Conference on Computing, Communication, and Automation (ICCCA), IEEE,* 893-900. https://doi.org/10.1109/CCAA.2017.8229933

Botta, Alessio, W., De Donato, V. & Persico, Antonio, P. (2016). Integration of cloud computing and internet of things: a survey, *Future Generation Computer Systems, 56*(2016), 684-700. https://doi.org/10.1016/j.future.2015.09.021

Boudi1, A. *et al.* (2018). Assessing Lightweight Virtualization for Security-as-a-Service at the Network Edge, *IEICE Transactions on Communications*.

Bratterud, A., Happe, A. & Duncan, B. (2017). Enhancing Cloud Security and Privacy: The Unikernel Solution, *The Cloud Computing 2017: The Eighth International Conference on Cloud Computing, GRIDs and Virtualization*. Athens, Greece, 79-86.

C. Vijaya & P. Srinivasa (2016). A Survey on Resource Scheduling in Cloud Computing, *International Journal of Pharmacy & Technology (IJPT), 8*(4), 26142-26162.

Chandra, K. N. & Neelanarayanan, V. (2017). A Survey on Cloud Application Portability, *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), 5*(1), 1039-1046.

Chaudhari, P. & Patel, B. (2017). Future of Big Data, *International Research Journal of Engineering and Technology (IRJET), 4*(1), 595-597.

Chowdhary, R. & Rawat, S. (2013). One Time Password for Multi-Cloud Environment, *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), 3*(3), 594-597.

Chraibi, M. *et al.* (2017). Policy Based Context Aware Service Level Agreement (SLA) Management in the Cloud, *The Proc. of the Cloud Computing 2017: The Eighth International Conference on Cloud Computing, GRIDs and Virtualization*. Athens, Greece, 122-128.

Daylami, N. (2015). The Origin and Construct of Cloud Computing, *International Journal of the Academic Business World, 9*(2), 39-43.

Deitel Pau, L. & Deitel, H. (2017). *Java How To Program*. 11th edn. Pearson Education In. ISBN-13: 978-0-13-479140-1,New York, USA.

Devang Swami, B. S. (2016). Service Delivery Model for Big Data as a Service, *Data Mining and Knowledge Engineering, 8*(6), 171-176.

Dhabhai, A. & Gupta, Y. K. (2016). Study of Big Data in Cloud Environment with their Related Challenges, *Study of Big Data in Cloud Environment with their Related Challenges, 6*(8), 2511-2514.

Edlund, A. (2012). Cloud Computing: Challenges and Opportunities for Swedish Entrepreneurs, *Swedish Economic Forum Report, Entrepreneurship, Norms and The Business Cycle, Braunerhjelm (Ed.), Publication number 5*, 1-54.

El-Omari, N. K. & Alzaghal, M. H. (2009). Municipal WiFi as a Public Safety Network: Case Study; Amman-Jordan, *The International Conference on Information and Communication Systems (ICICS09)*, 97-101.

El-Omari, N. K. & Alzaghal, M. H. (2010a). Exploiting Wireless Technologies Infrastructure for Digital Inclusion in Jordan, *The International Arab Conference of e-Technology (IACe-T'2010)*.

El-Omari, N. K. & Alzaghal, M. H. (2010b). Utilizing WiMAX as a Public Safety Network in Jordan, *The IEEE 24th International Conference on Advanced Information Networking and Applications (IEEE AINA 2010)*, 350-355. https://doi.org/10.1109/AINA.2010.182

El-Omari, N. K. & Alzaghal, M. H. (2012). Shifting towards City-Wide Wireless Jordanian Cities, *ACM chapter: The Research Bulletin of Jordan ACM, 2*(2), 59-63.

El-Omari, N. K. & Alzaghal, M. H. (2017). The Role of Open Big Data within the Public Sector, Case Study: Jordan, *The 8th International Conference on Information Technology (ICIT 2017), Internet of Things, IEEE*, 182-186. https://doi.org/10.1109/ICITECH.2017.8079997

Essandoh, E., Osei, Y.-B. & Kofi, A. (2014). Factors Influencing the Adoption of Cloud Computing by Small and Medium Enterprises in Developing Economies, *International Journal of Emerging Science and Engineering (IJESE), 2*(4), 13-20.

Filippi, P. De & McCarthy, S. (2012). Cloud Computing: Centralization and Data Sovereignty, *European Journal of Law and Technology, 3*(2), 1-22.

Foster, I. *et al.* (2008). Cloud Computing and Grid Computing 360-Degree Compared, *The IEEE Grid Computing Environments Workshop (GCE'08)*, 1-10. https://doi.org/10.1109/GCE.2008.4738445

Gaur, A. & Anurag, J. (2017). Glimpse of Cloud Computing, *International Journal of Advance Research, Ideas and Innovations in Technology (IJARIIT), 3*(2), 901-906.

George Pallis (2010). Cloud Computing: The New Frontier of Internet Computing, *IEEE internet computing, 14*(5), 70-73. https://doi.org/10.1109/MIC.2010.113

Gholami, F., Daneshgar, F. & Beydoun, G. (2017). Key challenges during legacy software system migration to cloud computing platforms - an empirical study, *Elsevier Ltd.: Information Systems, 67*, 100-113.

https://doi.org/10.1016/j.is.2017.03.008

Ghwanmeh, S., El-Omari, N. K. & Khawaldeh, S. (2015). An Innovative Study to Discovering and Enhancing Jordanian Universities Students Responsiveness of Green ICT, *The 2nd International Conference on Mathematical Sciences and Computer Engineering (ICMSCE 2015), Langkawi, Malaysia*. Langkawi, Malaysia, 112-115.

Gumbi, L. N. & Mnkandla, E. (2015). Investigating South African Vendors' Cloud Computing Value Proposition to Small, Medium and Micro Enterprises: A Case of the City of Tshwane Metropolitan Municipality, *The African Journal of Information Systems, 7*(4), 1-16.

Han, G. *et al.* (2016). An Efficient Virtual Machine Consolidation Scheme for Multimedia Cloud Computing, *Sensors, Multidisciplinary Digital Publishing Institute (MDPI), 16*(2), 1-17. https://doi.org/10.3390/s16020246

Hussein, M.-K. & Mousa, M.-H. (2012). A Light-weight Data Replication for Cloud Data Centers Environment, *International Journal of Engineering and Innovative Technology, 1*(6), 169-175.

Jain, R., Sumit, C. & Kumar, R. (2017). Green Approach for Next Generation Computing: a Survey, *International Journal of Advanced Engineering Research and Science (IJAERS), 4*(1), 10-19. https://doi.org/10.22161/ijaers.4.1.12

Jiang, J. & Yang, G. (2010). Examining Cloud Computing From the Perspective of Grid and Computer-Supported Cooperative Work, *Cloud Computing: Principles, Systems, and Applications,* 63-76. https://doi.org/10.1007/978-1-84996-241-4_4

Jiang, Z. (2018). *Real-Time I/O System for Many-core Embedded Systems*. The University of York. Available at: http://etheses.whiterose.ac.uk/22608/1/Thesis_Final_V4.pdf.

K.Pranathi (2016). Data Security on Cloud Computing, *International Journal of Latest Trends in Engineering and Technology (IJLTET), 7*(2), 150-153. https://doi.org/10.21172/1.72.525

Kang, S. & Lee, K. (2016). Auto-Scaling of Geo-Based Image Processing in an OpenStack Cloud Computing Environment, *Remote Sensing, 8*(8), 662-671. https://doi.org/10.3390/rs8080662

Karim, S. *et al.* (2017). Is Big Data too Big for SMEs in Jordan?, *The 8th International Conference on Information Technology (ICIT 2017), Internet of Things*, 1-9. https://doi.org/10.1109/ICITECH.2017.8079968

Kaur, E. R. (2015). A Review of Computing Technologies: Distributed, Utility, Cluster, Grid and Cloud Computing, *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), 5*(2), 144-148.

Kaur, T. & Chana, I. (2015). Energy Efficiency Techniques in Cloud Computing: a Survey and Taxonomy, *ACM computing surveys (CSUR), 48*(2), 1-47. https://doi.org/10.1145/2742488

Khajenasiri, I. *et al.* (2016). A Review on Internet of Things Solutions for Intelligent Energy Control in Buildings for Smart City Applications, *Internet of Things Solutions for Intelligent Energy Control in Buildings for Smart City Applications, 8th International Conference on Sustainability in Energy and Buildings*. Turin, ITALY, 770-779. https://doi.org/10.1016/j.egypro.2017.03.239

Khedr, A. E. & Idrees, A. M. (2017). Enhanced e-Learning System for e-Courses Based on Cloud Computing, *Journal of Computers, 12*(1), 10-19. https://doi.org/10.17706/jcp.12.1.10-19

Kiranjot, K. & Anjandeep, K. R. (2014). A Comparative Analysis: Grid, Cluster and Cloud Computing, *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), 3*(3), 5730-5734.

Klement, M. (2017). Models of Integration of Virtualization in Education: Virtualization Technology and Possibilities of its Use in Education, *Elsevier Ltd.: Computers & Education, 105*, 31-43. https://doi.org/10.1016/j.compedu.2016.11.006

Kuo, M.-H. *et al.* (2014). Health big data analytics: current perspectives, challenges and potential solutions, *International Journal of Big Data Intelligence, 1*(1-2), 114-126. https://doi.org/10.1504/IJBDI.2014.063835

Laverty, J. P., Wood, D. F. & Turchek, J. (2014). Micro and Macro Economic Analysis of Cloud Computing, *Issues in Information Systems, 15*(2), 293-302.

M Gokilavani, GP Mannickathan & MA. Dorairangaswamy (2018). A Survey of Cloud Environment in Medical Images Processing, *Monthly Journal of Computer Science and Information Technology, 7*(11), 68-73.

Malik, R. *et al.* (2014). Data Security and Privacy in Cloud Computing: Threat Level Indications, *Science International (Sci-Int), 26*(5), 1991-1996.

Manju, S. & Sadia, H. (2017). Analyzing the Difference of Cluster, Grid, Utility & Cloud Computing, *IOSR Journal of Computer Engineering (IOSR-JCE), 19*(1), 55-60. https://doi.org/10.9790/0661-1901045560

Maria G. Koziri & Loukopoulos, T. (2017). Sensor Selection for Resource-Efficient Query Execution in IoT Environments, *The Cloud Computing 2017: The Eighth International Conference on Cloud Computing,* 39-43.

Marques, W. dos S. *et al.* (2018). Evaluating container-based virtualization overhead on the general-purpose IoT platform, *IEEE Symposium on Computers and Communications (ISCC),* 8-13. https://doi.org/10.1109/ISCC.2018.8538602

Maruf, A. & Albert Y., Z. (2017). An Automated Lightweight Framework for Scheduling and Profiling Parallel Workflows Simultaneously on Multiple Hypervisors, *The Cloud Computing 2017: The Eighth International Conference on Cloud Computing, GRIDs and Virtualization*. Athens, 16-25.

Michael, M. & Rajiv, R. (2012). Cloud Genius: Decision Support for Web Server Cloud Migration, *The 21st international conference on World Wide Web,* New York, 979-988.

Miorandi, D. *et al.* (2012). Internet of things: Vision, applications and research challenges, *Ad Hoc Networks, 10*(7), 1497-1516. https://doi.org/10.1016/j.adhoc.2012.02.016

Mirarab, A., Fard, N. G. & Shamsi, M. (2014). A Cloud Solution for Medical Image Processing, *International Journal of Engineering Research and Applications (ijera), 4*(7), 74-82.

Naveen, D. & Harpreet, K. (2013). Cloud Computing Security: Issues and Concerns, *International Journal of Emerging Technology and Advanced Engineering, 3*(3), 331-335.

Neves, P. C. *et al.* (2016). Big Data in Cloud Computing: features and issues, *The 2016 International Conference on Internet of Things and Big Data,* 307-314. https://doi.org/10.5220/0005846303070314

Pandey, D. R., Mishra, B. & Tripathi, S. K. (2017). Operating System Based Virtualization Models in Cloud Computing, *International Journal of Computer, Electrical, Automation, Control and Information Engineering, 11*(3), 284-287.

Patel, V. A., Patel, V. K. & Panchal, G. (2017). Integration of IoT and Cloud Computing and its issues: A Survey, *International Journal of Engineering Technology Science and Research (IJETSR), 4*(10), 903-907.

Perera, C. *et al.* (2014). Context Aware Computing for the Internet of Things: a Survey, *IEEE Communication Surveys & Tutorials, 16*(1), 414-454. https://doi.org/10.1109/SURV.2013.042313.00197

Plauth, M., Feinbube, L. & Polze, A. (2017). A Performance Evaluation of Lightweight Approaches to Virtualization, *The Cloud Computing 2017: The Eighth International Conference on Cloud Computing, GRIDs and Virtualization*. Athens, Greece, 4-9.

Pooyan, J., Ahmad, A. & Pahl, C. (2013). Cloud Migration Research: A Systematic Review, *IEEE Transactions on Cloud Computing, 1*(2), 142-157. https://doi.org/10.1109/TCC.2013.10

Prajkta, P. C. & Keole, R. R. (2012). Ontology of Cloud Computing, *International journal of Computer Technology & Applications (IJCTA), 3*(3), 1189-1193.

Qin, Z. *et al.* (2018). Privacy-preserving Image Processing in the Cloud, *IEEE Cloud Computing, 5*(2), 48-57. https://doi.org/10.1109/MCC.2018.022171667

Ramzan, M. & Alawairdhi, M. (2014). Cloud Expedite: A Novel Generic Business Model for Emerging Paradigm Shift in Cloud Computing, *Life Science Journal, 11*(5), 299-306.

Rehman, A. & Annapurna, D. (2017). SDN: A New Approach to Networking Technology, *International Research Journal of Engineering and Technology (IRJET), 4*(2), 1605-1609.

Roy, S. & Sardda, D. (2016). The Role of Cloud of Things in Smart Cities, *International Journal of Computer Science and Information Security (IJCSIS), 14*(11), 683-698.

Sameera, A. & Iraqi, Y. (2017). Feasibility of Digital Forensic Examination and Analysis of a Cloud Based Storage Snapshot, *Journal of Digital Information Management (JDIM), 15*(1), 19-29.

Sareen, P. (2013). Cloud Computing: Types, Architecture, Applications, Concerns, Virtualization and Role of IT Governance in Cloud, *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), 3*(3), 533-538.

Servan, R. M. (2014). Cloud Computing and Patient Engagement: Leveraging Available Technology, *The Journal of Medical Practice Management, 30*(2), 89-93.

Sharma, S., Singh, S. & Kaur, R. (2016). Virtualization in Cloud Computing, *International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), 2*(4), 181-186.

Shawish, A. & Salama, M. (2014). Cloud computing: paradigms and technologies, *Inter-cooperative collective intelligence: Techniques and applications, Springer Berlin Heidelberg*, 39-67. https://doi.org/10.1007/978-3-642-35016-0_2

Smith, D. B. (2016). Cloud Computing Deployments Should Begin With Service Definition, *Gartner Inc., Report ID: G00294396*, 1-3.

Solanki, S. S. & Shaikh, N. (2014). Cloud Computing: Data Separation Issues, *International Journal & Magazine of Engineering, Technology, Management and Research, 1*(11), 155-160.

Sommerville, I. (2015). *Software Engineering*. 10th edn. New York, USA: Pearson Education Inc.

Subbiah, S., Muthukumaran, S. S. & Ramkumar, T. (2013). Enhanced Survey and Proposal to secure the data in Cloud Computing Environment, *International Journal of Engineering Science and Technology (IJEST), 5*(1), 49-53.

Taneja, K., Taneja, H. & Chadha, D. (2012). Cloud Computing: A Catalyst for Commercial Success of Computing Trends, *International Journal of New Innovations in Engineering and Technology (IJNIET), 1*(1), 22-30.

Temkar, P. (2015). Clinical operations generation next. The age of technology and outsourcing, *Perspectives in Clinical Research, 6*(4), 175-178. https://doi.org/10.4103/2229-3485.167098

The National Institute of Standards and Technology (NIST) (2011). Definition of Cloud Computing, *NIST Special Publication, 800-145*, 1-7. Available at: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf (Accessed: 9 March 2019).

Tsz Lai, W., Trancong, H. & Goh, S. (2012). “Cloud Computing” in “A Fresh Graduate’s Guide to Software Development Tools and Technologies”, *National University of Singapore (NUS), Chapter 1*, 1-27.

Vardhana, M. *et al.* (2018). IoT based real time trafic control using cloud computing, *Cluster Computing,* 1-10. https://doi.org/10.1007/s10586-018-2152-9

Venkatachalapathy, D. *et al.* (2016). Big Data with Cloud Virtualization for Effective Resource Handling, *International Journal of Control Theory and Applications (IJCTA), 9*(1), 435-444.

Vikram, G. & Bhatia, S. S. (2016). Developing Assurance Framework of Cloud Computing in the implementation of ERP: A Literature Survey, *International Journal of Computer Science and Information Technology & Security (IJCSITS), 6*(4), 84-92.

VMware Inc. (2019). Virtualization Essentials, *VMware*, 1-10. Available at: https://www.vmware.com/solutions/virtualization.html (Accessed: 10 March 2019).

William, H. *et al.* (2017). Advancing the Micro-CI Testbed for IoT Cyber-Security Research and Education, *The Eighth International Conference on Cloud Computing, GRIDs and Virtualization*, 129-134.

Yashodha Sambrani, R. (2016). Efficient Data Backup Mechanism for Cloud Computing, *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), 5*(7), 92-95.

Yunchuan, S. *et al.* (2014). Data Security and Privacy in Cloud Computing, *International Journal of Distributed Sensor Networks, 10*(7), 1-9. https://doi.org/10.1155/2014/190903

Yuzhong, Y. & Lei, H. (2014). Large-Scale Image Processing Research Cloud, *The Fifth International Conference on Cloud Computing, GRIDs, and Virtualization*, 88-93.

Zughoul, B., Al-Refai, M. & El-Omari, N. (2016). Evolution Characteristics of ERP Systems that Distinct from Traditional SDLCs, *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), 5*(7), 87-91. https://doi.org/10.17148/IJARCCE.2016.5718

Copyrights