Cloud Computing and Its Applications: A Comprehensive Survey

Jalal H. Kiswani*, Sergiu M. Dascalu*, and Frederick C. Harris, Jr.*
University of Nevada, Reno, USA. *

Abstract

Cloud computing is one of the most significant trends in the information technology evolution, as it has created new opportunities that were never possible before. It is utilized and adopted by individuals and businesses on all scales, from a cloud-storage service such as Google Drive for normal users, to large scale integrated servers for online social media platforms such as Facebook. In cloud computing, services are offered mainly on three levels: infrastructure, platform, and software. In this article, an extensive and detailed literature review about cloud computing and its applications is presented, including history and evolution. Moreover, to measure the adoption of cloud applications in industry and academia, we conducted a user-study survey that included professionals and academics from various levels. The user-study methodology, details, and results are also presented and discussed.

Key Words: Cloud Computing; Cloud Applications; Software as a Service, Cloud Adoption, Survey.

1 Introduction

Cloud computing has made tremendous changes and improvements in the information technology industry, where using 1,000 servers on the cloud for one hour is cheaper than using one server for 1,000 hours [5]. In fact, without cloud computing, many startup companies would not even exist [80] or would not achieve their current economies of scale [68]. As shown in Table 1, worldwide spending on public cloud services was almost $210 billion in 2016, and it is expected to reach $383 billion by 2020 according to Gartner news [72] .

Table 1: Total worldwide spending on cloud services forecast

Year	Spending (Millions of dollars)
2016	$209,244
2017	$246,841
2018	$287,820
2019	$332,723
2020	$383,355

Since the beginning of the computing industry in the last century, the offering of computation hardware or software was mainly based on the perpetual on-premise approach. Organizations would buy computers, install them locally on their premises, and use them for their computational needs. Even with the rise of personal computers in the 1980's, the same behavior continued, so that organizations and individuals followed the same approach. During those eras, some people and organizations thought about a different approach: why not offer computation hardware and/or software as utility services, same as electricity or telephone landlines? Thus, users could use these services on a subscription-based approach (e.g., such as monthly or yearly), and pay-as-they-use (i.e., usage-based costing) [88].

John McCarthy was the first to present the idea of offering computing as utility in 1961 [25]. Five years later, "The Challenge of the Computer Utility" book written by Parkhill discussed many aspects of computing as a utility [70] . Although it has been more than 50 years, Parkhill's definitions and discussion are still relevant. In particular, he defined utility computing as permitting a number of remotely located users to utilize a group of facilities of large central computers with the same ease and flexibility of using their on-premise computers. Furthermore, he emphasized the need of developing low-cost wide bandwidth transmission. In fact, he argued that having such facilities may eliminate the need of local storage and, more interestingly, it may allow direct memory-to-memory communication between remote computers, which may enable faster growth of distributed computing.

Time-sharing was one of the approaches that were used at that period, to enable the expensive main-frames to be shared between users [22], as shown in Figure 1. However, since these machines were used for internal private use by the owning organizations, they were only useful for users within the same organization, or for external parties with constrained use for the obvious reasons of security and scalability.

With the evolution of the Internet in the 1990's, things started to change, where centralized public servers have become available, and unified client applications to communicate with these servers over a standard protocol were developed. In particular, web browsers that communicate with backend web-servers over the Hypertext Transfer Protocol (HTTP) was created. This evolution, along with the massive growth in the wide communication bandwidth opened the opportunity for a new model of software delivery: Software as a Service.

Meanwhile, grid computing started to gain some traction to utilize the ability of distributing tasks between commodity computers that are available in different geographical locations, and offer computing power on demand [12, 25]. In fact, grid

*Department of Computer Science and Engineering. Email: jalal@nevada.unr.edu, dascalus@cse.unr.edu, fred.harris@cse.unr.edu

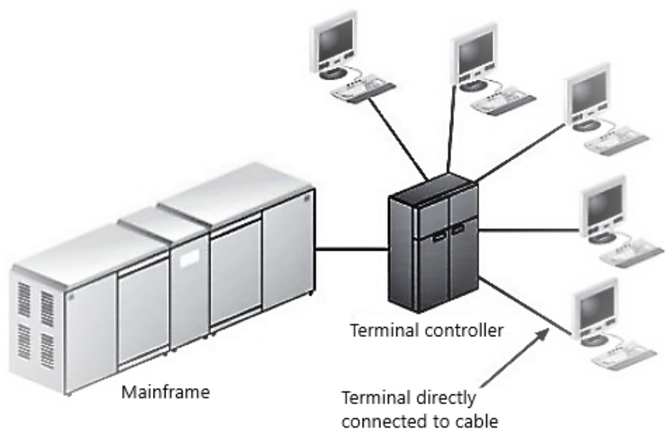


Figure 1: Mainframes time-sharing [69]

computing was later developed into cloud computing with the support of virtualization [10].

A generic definition of Software as a Service (SaaS) is: "applications delivered as a service over the Internet" [5]. Another more specific definition is: delivering software functionalities over the Internet for a large group of customers, based on a multi-tenant single instance software system [38]. However, software is a generic term and can include many categories and definitions [18], therefore, in this article, SaaS is referred to as cloud applications. From the 1990's era, Yahoo [90] and Hotmail (now Live) [60] are examples of cloud applications, which were targeting both individual consumers (B2C) and small-medium businesses (B2B) [1]. B2C model typically generates revenue by advertisements and subscription for premium services while B2B generates revenue by subscriptions.

On the other hand, new technology companies were founded to provide pure business applications based on the subscription model. Examples of such applications are the Customer Relation Management (CRM) software offered by SalesForce company [78], and Enterprise Resource Planning (ERP) by Oracle NetSuite [67]. With the massive growth and expansion of the Internet, companies started to offer new categories of new online services such as e-commerce, search engines, and social media. As a result, new technology giants started to gain significant traction and market shares, such as Amazon and Google. These companies have built large-scale data-centers to provide their services and products [5].

The availability of such large-scale data-centers, the drop in communication costs, and the decreased cost of electricity and hardware were the most important reasons for the rise of cloud computing and the offering of computation as a utility. Organizations started to offer Infrastructure as Service (IaaS) where computation, network, and storage were provided for customers [59]. In this case, however, customers still had to install the required software needed for running their systems. In addition, organizations started to offer another type of services where they provide platforms that enable domain-specific programming, such as Google App Engine [35] and Oracle Cloud Platform [66]. This type of service is called Platform as a Service (PaaS) [59].

In the beginning, providing IaaS and PaaS over the Internet was not efficient because infrastructure and platform resources were physically allocated for every customer, which caused many limitations, mainly related to scalability and management. These issues were the motivation for enhancing the so-called virtualization technology [10]. In fact, virtualization has been one of the primary enablers for many technology transformations that occurred in the last decade. In particular, physical resources are treated as a pool of virtual resources that can be shared between users, with the illusion of having dedicated resources for every user. Furthermore, virtualization provides elasticity, where the provisioning of services is automated and more efficient; compared to manual and physical provision, it may reach a near real-time provision.

Data centers that are used to host the three services (IaaS, PaaS, SaaS) based on the utilization of the virtualization technology embody what is called now cloud computing [5]. Figure 2 shows these services. Based on the National Institute of Standards and Technology (NIST), cloud computing is defined 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" [59].

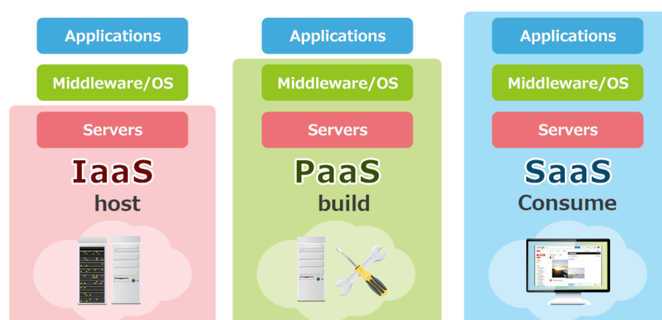


Figure 2: Services provided by cloud computing [15]

Organizations that provide cloud services are called service providers, while customers who use these services are called cloud users. In fact, service providers may also be service users at the same time. For example, an online software service provider (SaaS provider) for a student information system may host their applications on other organization's cloud infrastructure such as Amazon EC2 (IaaS provider) and be a cloud user (IaaS user) at the same time.

Cloud can be deployed in four categories: (i) private cloud, which is exclusively used and accessed by a specific organization, (ii) community cloud, which can be exclusively accessed and used by a community, such as the large datasets repository by Amazon that mainly targets data scientists [3], (iii) public cloud, which provides services open for use by the general public, and (iv) hybrid cloud, which consists of two or more deployment models [59]. In 2017, hybrid cloud was the most adopted model by enterprises [74].

As shown in Figure 3, this article presents a detailed survey of cloud computing in general, and cloud applications in particular, aimed at providing researchers and practitioners with comprehensive information on cloud computing history and its evolution.

In addition, we conducted a user study survey that gave us a more clear idea about the actual adoption of cloud computing and its applications by professionals from both industry and academia.

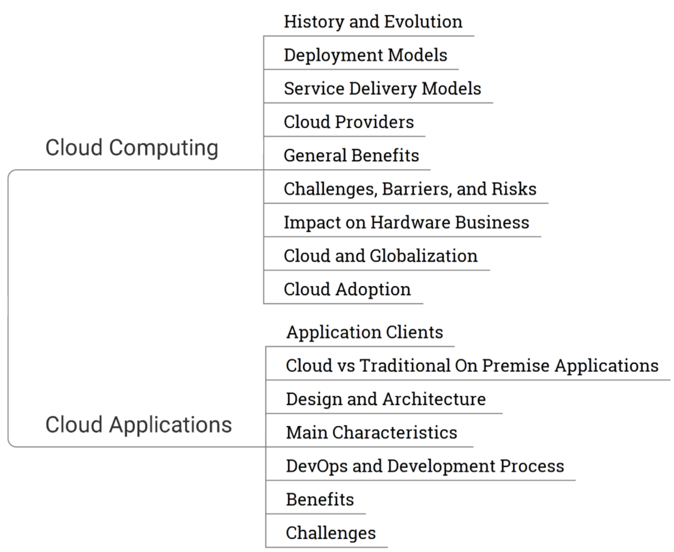


Figure 3: Review Structure of Cloud Computing and Cloud Applications

The rest of this paper is structured as follows: Section 2 discusses cloud computing, and Section 3 presents cloud applications. These sections follow the structure presented in Figure 3. Section 4 describes the user study survey which we conducted, including its methodology, results, and discussion. Finally, Section 5 concludes this article and identifies several directions of future research.

2 Cloud Computing

In the information technology industry, there have been many innovations that revolutionized how services are designed and delivered to customers. One of these important innovations is Cloud computing, which has affected almost every industry and discipline. With cloud computing, startup companies do not need to worry about investing in large data centers and hardware anymore. Software developers can start building software applications on top of platforms that can enable rapid-web application development, that can be deployed immediately. Enterprise organizations do not need to buy expensive software that may become stale with time and need large operational and maintenance costs. In addition, the new categories of Internet hardware clients (i.e. mobile and IoT devices) will be 10 times more than the traditional Internet clients, consisting of over 10 billion devices [62, 8]. Moreover, it will enable saving costs and delegate liabilities [36].

Cloud computing has opened new opportunities, trends, and needs such as mobile interactive applications, parallel application processes, the large growth in data size, the rise of analytics, the need of bringing the data near the applications, real-time decisions, and the Internet of Things.

Even though many definitions introduced in this article are about cloud computing, we find the following definition of cloud computing to be the most comprehensive:

> *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* [25].

Cloud computing has five main characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service [59]. Figure 4 shows the simplified cloud infrastructure.

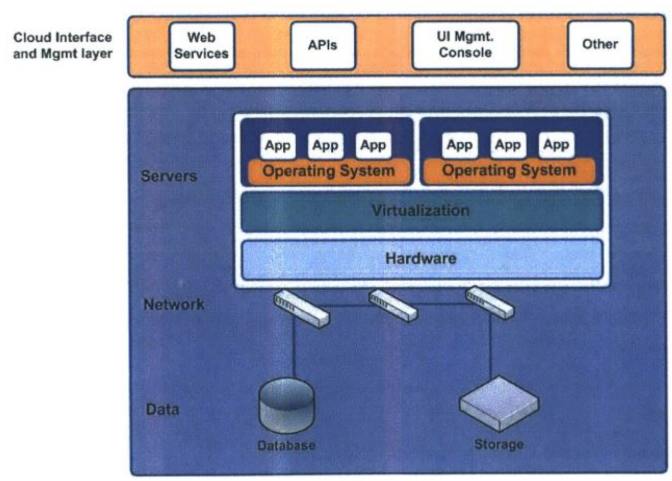


Figure 4: Simplified cloud infrastructure [36]

According to NIST, the main characteristics of cloud computing are: "on-demand self-service, broad network access, resources pooling, rapid elasticity, and measured service" [59].

Even though cloud computing started to get high traction after 2005 [36], the idea was discussed in the middle of the 20th century as Utility Computing. History and evolution are presented in Section 2.1. Furthermore, the section includes the enablers that caused the massive growth of this technology.

Cloud can be deployed on different models, private cloud, community cloud, public cloud, and hybrid cloud. The main difference between these deployment models is the target audience and public accessibility. More details about cloud deployment models are discussed in Section 2.2. However, for the rest of this article, the focus is on public and community clouds (assuming community is publicly accessible).

With cloud computing, several data centers will be turned into a single pool of computing utilities, which will enable the illusion of infinite resources. Cloud computing can deliver services using three different models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The sum of these services, data centers they operate on, and the software that runs and manages these data centers, is called "The Cloud" [5]. These three service delivery models are discussed in detail in Section 2.3.

As shown in Figure 5, the entities on which providers of cloud services operate are called cloud providers, and the entities in which users consume those services are called cloud users. Dedicated discussion about cloud providers, the advantages of being one, major providers and their classifications are presented in Section 2.4.

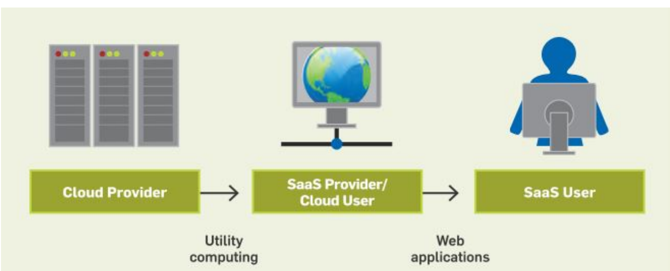


Figure 5: Providers and users of cloud services [4]

Furthermore, and since every technology introduces advantages and benefits on one side, there are disadvantages and challenges on the other side. Benefits of cloud computing are presented in Section 2.5, while challenges, barriers, and risks are discussed in Section 2.6 for both, providers and users.

The impact on the hardware industry is discussed in Section 2.7 and cloud globalization is briefly covered in Section 2.8. Finally, the adoption of cloud computing is presented in Section 2.9.

2.1 History and Evolution

Users of utility services such as electricity and telephones employ services based on a model called subscription. In the subscription model, the billing amount is based on service usage. The usage is measured by specific metrics based on the type of service. Usage-based billing is called the pay-as-you-go model. The rationale behind this approach of service delivery is based on resources and services shared among different customers. In particular, this service delivery will reduce the overall cost of resources and services and enable providing services at competitive prices. For example, the sharing of electrical lines connecting a whole neighborhood can reduce the cost by having only very short distance lines to be dedicated for individuals, which will enable a lower cost of service. On the other hand, having an electricity line for a person who lives in his own large farm may be very expensive, since it may require dedicated resources.

Offering information technology services on the utility-based subscription model is not a new idea. In fact, the first discussion about offering computing as a utility started in 1961 when John McCarthy said that computation might be offered as a utility someday [25]. Later on, in 1966, Parkhill discussed in on how a revolution in distribution and utilization of computer power may enable social changes and opportunities of human development [70]. In particular, he discussed benefits, challenges, and future directions and potential of computing utility. Parkhill said that data transmission, expensive hardware, and limited hardware capabilities were the main barriers. Furthermore, he discussed working to overcome these barriers to open the door for new opportunities of eliminating the need of local storage and allow direct memory-to-memory communication between two remotely located computers, and enable faster growth in distributed computing. In addition, Parkhill considered, this may enable better teaching and information sharing.

Meanwhile, companies like IBM were working on the time-sharing technique [22], as shown in Figure 1. In time-sharing, an expensive computer such as a mainframe was enabled to share computing resources with many users. However, due to the limitations described earlier (e.g., expensive hardware), this was not executed on a large scale and was based on a single organization or community.

The evolution of personal computers in the 1980's has made computers available for common people. This was one of the motivations to create the next big thing, the Internet. In the 1990's, a global network of networks was built based on standards protocol such as TCP/IP, SSL, HTTP, FTP, SMPT, and POP. This global network was named the Internet and has affected all aspects of our lives. Meanwhile, the communication technology and hardware were improving as well, which enabled cheaper and faster computing and data transmission.

Since the late 1990s, businesses, academia, governments, and the general public started to offer and use services over the Internet. In the beginning, these services were mainly offering software applications over the Internet. In 2000-2002, Intel failed to offer computing services for organizations and companies, because it required negotiations and long-term contracting, which did not enable scaling their approach [5].

In 2003, Jim Gray, the manager of Microsoft Research Lab in San Francisco, expressed his opinion about the future of distributed computing, where software can be available in different physical locations and communication with other software over standard protocols. He discussed how the free services provided by internet service providers were actually not free and were paid for by advertisements. In addition, he discussed the high cost of ownership, which reached \$1 trillion per year. An exciting point in his report is that in 2002, at Google, only 25 operation staff were managing a twopetabyte database and distributed it over 10,000 servers using automation tools. In fact, this was the main reason Google was generating profits since they had lower operational costs. This applies also to other giant vendors such as Yahoo and Hotmail. Gray discussed how the future of the Internet would depend on computer-to-computer interaction. However, the advertisement model, which was the main revenue stream at that time, was not sufficient, and companies needed to invent a new business model that leverages the new trend [37].

Later, smart-phones started to be everywhere, which increased the number of Internet users by orders of magnitude. Sensors and smart-device industries were growing, and the Internet of Things (IoT) became a major discipline. For example, a cloud computing approach was proposed to solve the problem of the increasingly growing size of satellite weather data [77]. On the other hand, the economic crisis in 2008 motivated all types of entities to look into cheaper and more efficient ways of doing their businesses. Moreover, space has become a serious issue for corporations, where a quarter of their data centers are out of space [80].

From technical perspectives, the drop-in hardware prices along with the increase in computation power and storage capacity enabled building data centers from commodity computers [25]. In addition, this was helped by the decrease in the cost of utilities, and communication of small to medium data centers [80]. Moreover, the operational cost was reduced due to the availability of automation software tools. Furthermore, wireless device adoption and smartphones removed the dependency of poor infrastructure and enabled even developing countries to be part of the revolution. Moreover, resource pooling using virtualization technology enabled the lower cost and more scalability [5, 36, 41]. Also, adoption of free open-source software, global workforces, and agile software processes played important roles in this evolution [36].

From business and economical perspectives, customer behaviors have changed from buying expensive long-term services and assets to subscription low-price services with low commitments. In particular, they were moving from capital and asset expenses (CapEx) to operational expenses (OpEx). In CapEx, organizations buy computing assets, while in OpEx organizations pay for cloud services on pay-as-you-go option, which can be beneficial from economic perspectives. In addition, the economic crisis in 2008 played a major role where organizations started to look at alternatives to reduce the cost of doing business [4, 41].

2.2 Deployment Models

Cloud computing can be deployed in different deployment models, based on the target audience and users. These models are (i) private cloud, (ii) community cloud, (iii) public cloud, (iv) and hybrid cloud.

Private cloud is a data center or set of data centers that are provisioned for and used by only one organization or corporation. The users of this model are the organization's internal and/or external users. In fact, this model is very similar to a traditional on-premise data-center; however, the utilization of cloud computing technologies (e.g., virtualization), and the possibility to be run and managed by third party organizations are key differences [59]. An example is a virtual data center built by a bank to deploy a core banking system.

A community cloud has a broader audience beyond the same organization. It serves a group of users who share common interests and concerns. For example, satellite image datasets and tools may be deployed on specific data centers managed by NASA or the National Science Foundation in the USA, to enable domain and data scientists to perform big-data analytics on these data sets [59].

In the public cloud, services are open to the general public on the pay-as-you-go model for open use. An example of this is the public hosting providers, which can enable businesses to deploy their Internet applications and support rapid scalability and monitoring [59]. This model is a popular type of cloud deployment model among individuals and small to medium businesses. On the other hand, hybrid cloud is preferred for enterprises [74].

2.3 Service Delivery Models

Cloud computing provides services in different service delivery models. Even though there is a unified classification for these services, for now we will go through NIST service models, which are Infrastructure as a Service, Platform as a Service, and Software as a Service [59]. These delivery models are shown in Figure 6.

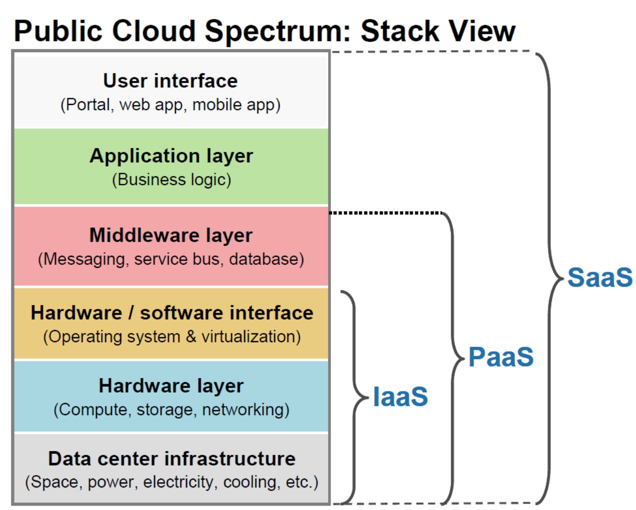


Figure 6: Cloud computing services delivery models [41]

Infrastructure as a Service (IaaS) provides the lowest level of services from cloud providers to cloud user (consumers). IaaS includes services such as computation, storage, networks, and any other fundamental infrastructure. In this model, cloud users will be able to install and deploy arbitrary software including operating systems and applications. However, users do not have the ability to manage the underlying cloud infrastructure. In particular, it is a collection of hardware and software that enables the special characteristics of cloud computing. IaaS includes virtual machines, servers, storage, load balancer, a network [36], and may include an operating system [74].

On the top of IaaS, Platform as a Service (PaaS) is provided. PaaS is a service that enables its users to build and/or deploy their applications using language, libraries, services and tools provided by PaaS providers on a cloud platform. As we might expect, PaaS users do not have any control over the platform or cloud infrastructure. However, they have control over their application's management [59]. PaaS may include execution runtime, database, web servers, and development tools [36, 74].

Software as a Service (SaaS) is defined as the applications provided over the Internet [5]. Another definition is the capability given by application providers (SaaS providers) to consumers (SaaS users) to use applications running over a cloud infrastructure. In this model, SaaS users cannot control the underlying infrastructure, the platform or the applications; however, a possible configuration for the user's specific instances may be provided [59]. A research group that works for IBM argues that SaaS should be a single instance of a software system, serving a large group of customers over the Internet, built on top of a multi-tenancy platform [38]. Some examples of SaaS are CRM, email, virtual desktops, and computer games [36].

However, many other flavors of services are provided, especially in industry, such as Container as a Service [74], Function as a Service, Testing as a Service, Database as a Service, Security as a Service, and even Metadata as a Service [17]. All these types of services are sometimes grouped as XaaS, where X is anything as a service [5, 76].

2.4 Cloud Providers

Low-level service delivery models such as infrastructure as a service require a large investment in properties, hardware, software, and operations. However, many technology giants such as Google, Amazon, and Microsoft already had these infrastructures in their data centers to be able to provide their services to their customers. The availability of these assets gave those companies an edge over other companies, by leveraging their current investments and thus getting a higher market share. In addition, this was one of the main enablers of cloud computing, as further discussed in Section 2.5. The companies that provide low-level services mainly related to hardware are called cloud providers. Figure 7 shows the list of top cloud providers in 2018, by RightScale [75].

Being a cloud provider can bring many benefits. In particular, depending on the economies of scale concept, long-term financial stability and high revenue are possible. In fact, based on the dynamics of innovations theory by Utterback [86], being part of the innovation will give them the chance to dominate the market. On the other hand, new innovations may, directly and indirectly, affect other businesses. For example, after offering the cloud infrastructure for a fraction of the price, hardware sales started to drop dramatically. In fact, this has forced hardware manufacturers such as Dell and HP to collaborate with research labs on cloud computing and start investing in this field to catch the wave.

Area	AWS	Azure	Google	IBM
% Adoption	68%	58%	19%	15%
YoY Growth in Adoption	15%	35%	26%	50%
% Adoption in Beginners	47%	49%	18%	14%
% with Footprint >50 VMs	58%	44%	17%	14%
YoY Growth in Footprint> 50 VMs	14%	38%	42%	56%

AWS leads
Other vendors lead AWS
Source: RightScale 2018 State of the Cloud Report

Figure 7: Top cloud services providers [74]

Choosing a data center's location to provide low-level cloud services depends on many factors. For example, choosing locations with low-price property prices, low labor cost, and lower taxes could ensure offering the service at more competitive prices, while leveraging higher profit margins. On the other hand, to deliver the reliability of service, location selection criteria should include the quality of infrastructure and utilities, such as Internet speed and electricity reliability.

However, based on the physical theories, shipping photons over fibers optics is cheaper than shipping electricity [5], and cooling is still a challenge and consumes most of the electricity cost. Many research efforts are focused on this issue, for example, Google has applied for a patent for a water-based data center. Their patent application shows data centers on large ships utilizing the sea motion and water in both electricity generation and cooling [80]. In addition, setting up the data centers in cold areas might be an option; however, infrastructure availability may be a limitation. Moreover, current data center's design for modularity, power, and cooling requires new innovations [39].

2.5 General Benefits of Cloud Computing

Cloud computing provides benefits for both service providers and service customers.

Adopting different cloud services can be beneficial for both service providers and service users. For example, SaaS service providers will not need to maintain multiple versions of their software, since most likely they will have only one version running at any point in time, while utilizing the multi-tenant design to achieve higher scalability. On the other hand, SaaS users will be able to get higher-quality service and software that can be accessed from anywhere, with a fraction of the price of traditional software, and with risk delegation to service providers.

A general benefit for service providers is to utilize the economies of scale concept, which means having a large customer base with recurring revenue. From the IaaS perspective, as discussed in Section 2.4, the providers can gain many benefits such as leverage current investments and defend their franchise. From PaaS perspective, PaaS providers will have higher opportunities of customers lock in, which will reduce the risk of customer turn-over.

A general benefit for service users is the reduction in the overall cost, and delegation of liabilities to service providers [36]. Moving from CapEx to OpEx will get better tax benefits as well as reducing the operational and administration costs. Liability delegation is achieved by transferring risks such as security, availability (e.g. DDos attack), and legal liabilities to service providers. In fact, the cost of protecting the cloud is less than the cost of attacking it, since the attackers will require huge resources and bandwidth to be able to attack a cloud data center, which results in this not being feasible in most of the cases for the attackers. In addition, it is possible to reduce the risk of procuring a service that does not achieve the business goals by leveraging the test-before-you-buy flexibility, since most of the cloud services are coming with no long-term commitment.

Furthermore, cloud computing can protect against the risk of load mis-estimation. In fact, the elasticity provided by cloud computing enables efficient scalability, which can make the service more reliable since the demand of service may be affected by many uncontrolled events such as news and holidays. Additionally, with cloud computing, there is no need for a large capital investment in hardware or operations, especially for startup companies. Figure 8 shows the difference between automated elasticity in scalability conditions compared to traditional non-automated solutions, and how automated elasticity can almost reach the actual demand without consuming higher resources.

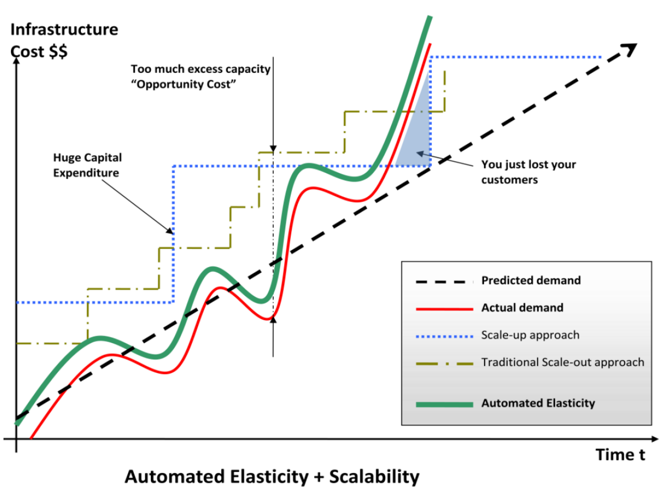


Figure 8: Automated vs traditional applications scalability comparisons [88]

From an operational perspective, cloud computing can benefit service users by enabling them to access the service anytime anywhere. In addition, collaboration and data sharing are easier. However, the risk of data security exists, as discussed in the challenges covered in Section 2.6. In addition, having more control over servers and installed applications, where 30% of on-premise servers for an enterprise have applications that nobody knows about, and they follow the rule of "Let's pull the plug and see who calls" [80].

From a business perspective, since the largest cost for enterprises is people, with about 1 employee for 100 servers [39], automated elasticity can also contribute to reducing the operational labor cost.

2.6 Challenges, Barriers and Risks

However, the advantages of cloud computing come with challenges, costs, and risks. Major concerns include security [14], data-privacy and data lock-in, bandwidth limitation, and availability. Also, the Internet neutrality [84], political conflicts and governmental regulations may provide a variety of constraints for both service providers and service users.

One of the disadvantages is that it is a single point of failure, where most of the cloud providers use the same infrastructure and the same software, in which any introduced bugs or issues can affect all service users. Obviously, bugs in large-scale distributed systems used by cloud providers are hard to debug and fix. Another disadvantage is performance unpredictability, since shared virtualization resources and dynamic elasticity may affect the performance of other users. Moreover, the provision of new services is still not happening in real-time. Finally, technical up-to-date expertise in cloud technology in general, and in cloud services management and administration in particular, is relatively not easy to find [41, 74].

One of the main challenges of cloud computing is data transfer, which is still a bottleneck. For example, moving large datasets is still cheaper and faster with a traditional mail service such as FedEx. Moreover, another challenge is internal organizational and business policies, which enforce the usage of internal data centers.

In addition, many customers have concerns in regards to security [14], data lock in, data confidentiality [32], data auditability, and control loss [36]. In fact, data can be exposed in multiple scenarios, it can be exposed during upload, while in the cloud, as well as during backup and restore [48].

From a business perspective, pricing uncertainty and cost model complexity, may not fit with some organizations' policies such as government. Fate sharing between service users and service providers is becoming more common, where a failure in the cloud provider's service may affect the reputation of service users as well [5]. In addition, since cloud services are based on heavy marketing campaigns, this may not be appropriate for some type of businesses such as some scientific disciplines. Furthermore, lowering consultancy revenue is another concern for some service providers.

Also, some broad license agreements are a critical risk, which may enable service providers to terminate the service any time for any reason, without any customer communication or feedback [32].

Moreover, politics and regulations can affect the evolution of cloud computing. For example, Internet neutrality [84], political and governmental conflicts may cause a suspension of services. In addition, some compliance regulations may put some special constraints, such as the European Union Data Protection act [29], where all European customers data should be saved in data centers located in the European Union [36].

Another barrier may be the rejection from freeware and privacy advocates such as Richard Stallman, founder of the Free Software Foundation [27] and the creator of the computer operating system GNU [28]. In his interview with the Guardian in 2008 [46] he argued that cloud computing is a trap for enforcing people to buy proprietary licenses, and the main reason for its adoption is the marketing campaigns.

In addition, availability is still a challenge, as service issues such as service suspension by Amazon and Google [24, 25, 80] add certain concerns. Furthermore, disasters may destroy infrastructure and interrupt service for days or perhaps weeks. On the other hand, service suspension may be caused by the unavailability of Internet access for some special places like underground halls or airplanes.

Figure 9 shows the barriers ranking for cloud users based on a study done in 2011 by Morgan Stanley financial services firm [41]. Based on this study, data security is the main concern, followed by cost uncertainty, loss of control, regulatory or compliance requirements, reliability, data portability/ownership, software compatibility, performance, and finally lock-in. Even after 7 years from that report, as presented in Figure 10, a report by RightScale published in 2018 shows many similarities in the challenges with the dramatic rise of *lack of the resources and expertise* challenge [75].

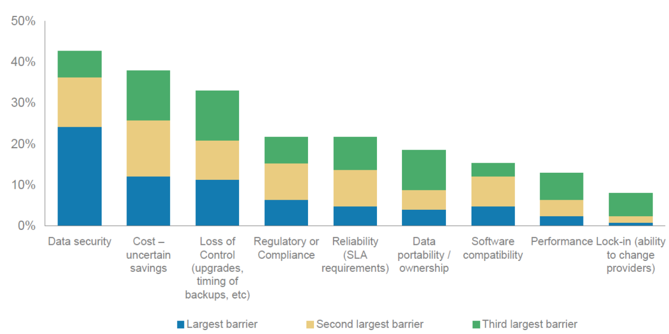


Figure 9: Cloud computing barriers ranking to users [41]

2.7 Impact on Hardware Business

Cloud computing has affected many businesses directly and indirectly. The computer hardware industry has directly been affected in many aspects. In particular, the behavior changes of customers toward OpEx instead of CapEx has affected hardware sales [41]. Customers now prefer virtual infrastructure instead of physical hardware for many reasons, including the illusion of

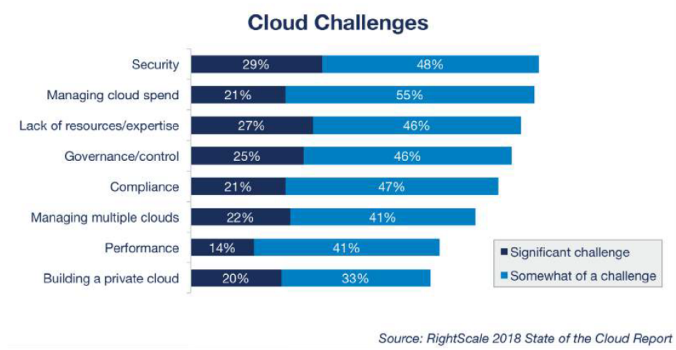


Figure 10: Cloud computing challenges in 2018 [75]

infinite resources available on demand and eliminating upfront investments [5]. On the other hand, cloud providers' orders of hardware have increased and are sold by a scale of containers. For example, about 2,500 servers were delivered by a 13-meter shipping container, then they were installed in a new Microsoft data center in Chicago and the center was up and running in only four days, including electricity and water supply for cooling and network setup [80].

In the hardware labs, power saving features are now one of the leading topics, to reduce the operational cost of data centers by reducing the cost of cooling and utilities. In addition, requirements for higher communication speed in routers and media WANs, and compatibility with virtualization technologies are significant. These factors and others, such as the new behavior of customers to go for short-term payments on pay-as-you-go option, led hardware manufacturers such as Dell, HP and IBM to jump-in and start providing cloud services [53, 5, 41].

2.8 Cloud and Globalization

Cloud computing may be the ultimate form of globalization, which may enable new worldwide business opportunities [80] and achieve higher economies of scale. In fact, even in the developing countries, cloud computing has been widely adopted from the first wave of cloud computing between 2006-2010. Countries such as China, India, and Turkey used cloud computing for E-Education, E-Health, and other applications [53].

2.9 Cloud Adoption

As discussed in the previous sections, cloud computing has many benefits that can reduce risks and increase profits. Cloud computing utilization is highly recommended over private data centers in many scenarios, in particular when the demand of service varies with time. This allows a benefit from elasticity and ensures efficient utilization of resources. In addition, cloud computing is recommended when the demand of service is not known in advance, such as the growing demand of new services or needs of startup companies. Furthermore, cloud computing is preferred with batch analysis jobs, which most likely will get results faster. Moreover, running out of space for new data centers may be a strong motivation to adopt the cloud. Finally, resource limitations such as the inability to provide extra utility power for cooling [80] is another motivation to move to cloud computing.

Based on the cloud maturity model of Rightscale [74], cloud users are classified in four categories: (i) cloud watchers, (ii) cloud beginners, (iii) cloud explorers, and (iv) cloud focused. Cloud watchers have not adopted any cloud technologies yet are still in the evaluation phase; however, they are planning a cloud strategy. On the other hand, cloud beginners started to do some experiments for cloud services such as proof of concepts or running these services on a small scale. The third category is cloud explorers, consisting of the users who have adopted cloud computing in serious work for multiple projects, and they have the required expertise to use and manage their cloud services. However, they are still exploring new opportunities to expand their business on the cloud. Finally, there are focused users, where the business is heavily based on cloud computing, and they work on cost and optimization for their cloud infrastructure. Figure 11 shows the cloud adoption percentages as of 2017.

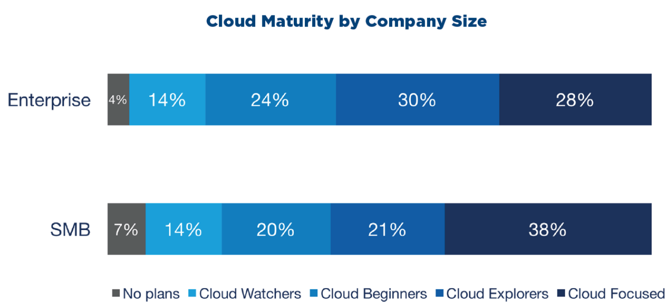


Figure 11: Cloud adoption as of 2017 [74]

3 Cloud Applications

Software as a Service (SaaS) is one of the service delivery models of cloud computing. In fact, SaaS is projected to have 54% spending share from cloud computing services by 2020 [30], as shown in Figure 12.

From technical and software engineering perspectives, cloud applications offered by SaaS services are different from traditional software applications. As Bill Gates, the founder of Microsoft, said: "We now live in a world where a subroutine can exist on another computer across the Internet" [80]. Several factors such as scalability, multi-tenant support [38], billing, monitoring, data-locality, and integrability enforced the creation of new terms in technology such as Native Cloud Applications (NCA) and Microservices Architecture (MsA), which enable utilizing the full benefits of cloud computing [7]. In addition, software development processes have improved, for example software operation activities may be assigned to software developers in what is called DevOps [9]. From small startup companies to large-scale enterprise software development houses, all are adopting these concepts. For example, Peter Zencke, SAP ERP new version development lead, indicated how exciting it is that any components of the software can be a service provided by other vendors [80].

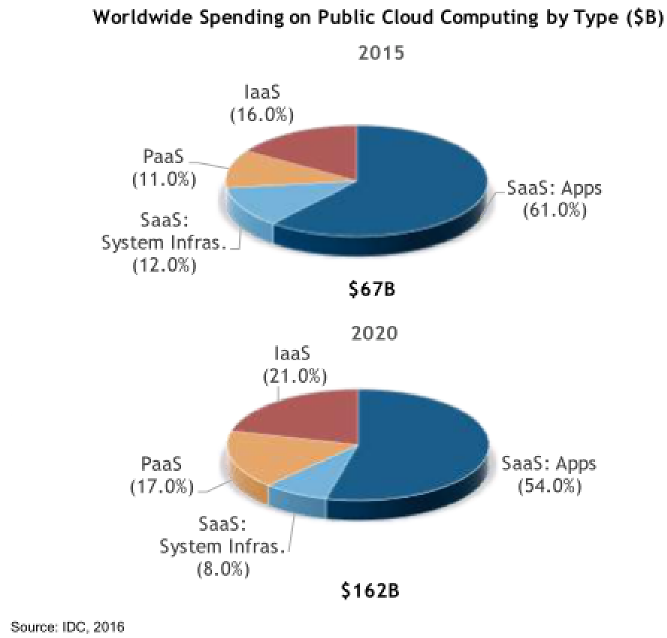


Figure 12: Cloud computing services worldwide spending [30]

As shown in Figure 13 software is a very generic concept, which can include operating systems, programming languages, tools, and applications [18]. Hence, these different categories of software can be part of any cloud service delivery model (i.e. IaaS, PaaS, and SaaS). In this paper the term Software as a Service is used interchangeably with Cloud Applications. A discussion about a proposed new taxonomy for service delivery is further presented in Section 5.

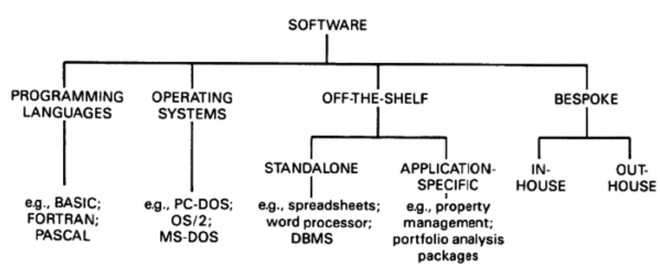


Figure 13: Software taxonomy [18]

Generally, cloud applications are defined as applications delivered over the Internet [5]. Cloud applications are most likely to run over PaaS (discussed in Section 2.3) [83]. A more detailed definition is the functionality provided over the Internet for a large group of clients, based on a multi-tenant platform, with a single instance of software applications; which also could be provided at the application level [38]. Section 3.1 presents the different types of clients that can access cloud services. Cloud applications are different from traditional on-premise applications in many aspects, and these differences are discussed in Section 3.2. The design and architecture of cloud applications are presented in Section 3.3. New patterns and techniques for developing cloud applications are required and offer advantages over traditional monolithic applications, as discussed in Sections 3.3.3 and 3.3.4. Benefits of cloud applications are discussed in Section 3.6, and related challenges are presented in Section 3.7.

3.1 Application Clients

Most cloud applications consist of server-side components (i.e. application backend) and client-side components (e.g. front-end). Since the beginning of cloud evolution, HTML -Internet standard front-end technology- was the main used technology. Internet browsers have been the most widely used HTML clients to communicate with application backends. Desktop widgets became another form of clients [80]. Meanwhile, and with the rise of mobile devices and Internet of Things, new browsers, desktop applications, and sensors applications were added to the client's stack. Consequently, the complexity of building Internet-based applications has increased, and the need for more sophisticated front-end, a new term was introduced: Rich-Internet-Applications (RIA).

RIA is commonly based on utilizing client-side features that depend on JavaScript, HTML5, and CSS3. In particular, the Asynchronous JavaScript And XMLHttpRequest technology (AJAX) is the main enablers for modern interactive web-based applications. Even though it has been the common approach, the user-experience of browser-based applications was not sufficient for scaling cloud applications and could not be used by non-technical people, which forced the way to native applications. In particular, native applications are developed using a programming language to build applications that can utilize native platform components. An example of native applications are applications built using the Java programming language to create Android platform apps. Another example is using Objective-C or Swift to build Apple IOS apps.

Another form of application clients is desktop widgets that communicate with back-end cloud services, such as weather or stock widgets [80].

The new trends of having different front-end clients require new architecture, design patterns, and tactics [40]. Design and Architecture of cloud applications are discussed in Section 3.3.

3.2 Cloud vs Traditional On-premise Applications

On-premise traditional applications are the software application instances that are designed to be installed on a client's environment (e.g. local data centers or computers) on the client's premises. The local installation of these applications includes all the application dependent artifacts and software systems, such as web servers, application servers, and databases. In on-premise applications, dedicated support for clients is provided, with different versions installed in every client's environment. In addition, there is no resource or access sharing with other clients. However, some form of integration with other systems, and external access to the client's services may be provided from that on-premise deployment.

The common licensing model for traditional applications is perpetual-licensing, where clients can use the software without any time limitations, and cost can be a one-time implementation [13]. In this model, the cost can be accurately estimated from the beginning. In addition, in this approach, clients have almost full control over the applications and its data. On the other hand, cloud applications can be licensed on pay as you go or rental licensing models [56, 65, 89].

There are many disadvantages and challenges for on-premise applications. Firstly, clients pay a relatively large amount of money for licensing compared to cloud applications. Secondly, on-premise applications require special upfront consulting and implementation costs [41]. In addition, long implementation time is one of the main risks. In particular, hardware procurement and installation, software environment configurations and setup, application deployment, and on-site implementation are all causing implementation delays. Moreover, most of the time support and upgrades are not included in the initial cost of the system.

Although cloud applications sound tempting, they are not fit for all types of applications [56]. For example, the traditional approach is more appropriate for real-time stock trading which requires microsecond precision [5], since performance is not guaranteed like on-premise deployment because of the sharing nature of cloud applications [41].

Adding to that, the on-premise approach can make perfect sense in organizations working in sensitive domains, such as governmental or financial organizations. In fact, some domain compliance regulations and procedures require only the internal existence of their applications. For example, central banks in some countries enforce the core-banking to be locally installed and managed to ensure availability and data privacy.

On-premise software is normally built in a monolithic fashion. However, cloud applications are designed and built based on self-independent services [80], in what is called Microservice architecture [7]. A comparison between Monolithic and Microservices architectures is presented in Section 3.3.3.

3.3 Design and Architecture of Cloud Applications

In all categories of software engineering processes (i.e. waterfall, agile or component-reuse), the design is a significant phase [81]. Architecture is the core part of the design. Software architecture is an abstract, technology-neutral, representation of software systems elements, their relations and how they interact with each other. Moreover, architecture is important to deliver the quality attributes and the non-functional requirements of software systems [55]. Furthermore, the architecture can be used as an input for development, documenting, and evaluating software [33].

Over the years, various software-architectural tools and techniques have been developed and evolved to enable a more systematic approach of designing software applications, such as architectural styles, patterns, and tactics [40]. These styles include but not limited to the monolithic, Service Oriented Architecture (SAO), and the Microservices-based approach [50].

This section discusses the evolution of cloud applications, from Monolithic applications to SOA, Microservices Architecture, and Cloud Native applications.

3.3.1 Monolithic Applications Architecture. Since the beginning of computer software development disciplines, building applications was mainly done using the monolithic approach. In particular, in the monolithic approach all the components of a software application are built as a single unit that should be compiled and deployed as a single instance on the edit-compile-link concept [85], and most likely with the same programming language or technology. Even though this type of software architecture is easier for software developers to understand, develop, deploy and operate, it has many disadvantages, including: full system compilation is required for any change, having all the teamwork on same technology or programming language, and harder horizontal scalability because of the application's heaviness. Another major issue with the monolithic architecture is that the system is a single point of failure, where a single error in the application can take the whole system down.

3.3.2 Service-Oriented Architecture. Service-Oriented Architecture (SOA) is an approach used to overcome some of the monolithic application's limitations [85]. In particular, SOA is about decomposing applications into smaller-unit (services) that integrate and are composed at runtime with each other using a standard protocol. Various XML based web-service protocols are used as standard protocols, including SOAP, WSDL, and UDDI [54]. Even though it was an elegant approach based on standard technologies, it didn't get high-traction from small-medium organizations because of its complexity, protocol overhead, and heaviness of the final full system. In addition, having remotely located services was not practical since the units of most SOA applications were communicating with the same data stores (e.g. databases), which caused many performance and reliability issues.

With all the limitations and issues discussed, new factors led to new requirements being needed. These factors include: (i) smart-phones and IoT require new lightweight yet interactive front-end technologies, (ii) entrepreneurship-wave and startup companies require faster time-to-market and lower development cost, (iii) cloud computing requires economies of scale models and scalability with minimal hardware and infrastructure cost. All these factors caused the innovation of many new technologies that have disrupted the software industry. Those new technologies have also created another issue of lack of human-resources.

On the other hand, there is a conflicted misunderstanding about the relationship between SOA and SaaS. In fact, SOA is a software construction model, while SaaS is a software delivery model [54, 83].

3.3.3 Microservices Architecture. To overcome all the constraints, limitations, and disadvantages of the monolithic approach and SOA, and to achieve the requirements enforced by the discussed new trends, Microservices architecture was innovated. As shown in Figure 14 the microservices architecture is a modern way of building cloud-based software applications, in which software applications are decomposed into small light components (services) that communicate with each other over light protocols and light messages exchange. The most commonly used protocol is Representational State Transfer (REST), which is a light-weight text-based solution over the HTTP protocol [23]. JavaScript Object Notation (JSON) is the common message format used by inter-services and service to front communications [16]. In Microservices, every service may be developed using different technology, must access its own data-store, and may not access any other service's data-stores directly, only over the exchange protocol. In fact, directly accessing other service's data-store directly will increase coupling and reduce the service's portability. Moreover, these services are also independently deployable [26]. Also, horizontal scalability of Microservices is light and more efficient than other architectures. In fact, scalability is performed on the service level, where the service which has more load will be replicated on another instance, and there is no need to replicate the whole system. Application containers such as docker are the main enablers for this feature.

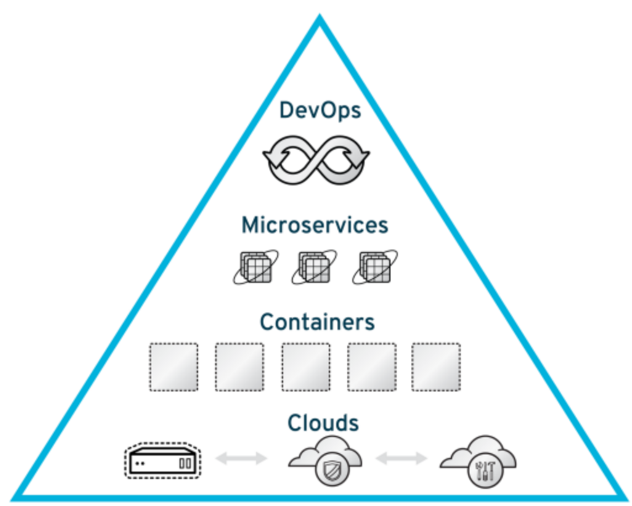


Figure 14: A pyramid of modern cloud native applications [20]

Furthermore, the microservices-based architecture can also come with other "dividends," such as enabling innovations for developers, since they have full control over the design of their microservice, and hence they can easily replace components and enable freedom of testing [49].

Even though the microservices style is similar to SOA in many aspects, such as decomposing software into smaller-deployable parts and communicating over a standard protocol [88], the lightness of communication based on REST, messages based on JSON, and separate data stores, might be the main differences.

Even though the microservices architecture solves many issues and problems, and creates potential for new opportunities, it introduces new complexity issues. In particular, special expertise is required to design and architect software solutions based on the microservices architecture. In addition, there is a complexity of integrating the services, testing, and deploying them. Moreover, monitoring and supporting services at runtime by the operations and support team is harder than supporting single processes applications such as the monolithic-based applications.

To reduce some of the risks of microservices architecture, intensive automation is required. In particular, automation can be achieved by applying automation software infrastructures such as Continuous Integration (CI), Continuous Delivery (CD), Test-Driven-Development (TDD), standard projects structure, and other [51].

3.3.4 Cloud-Native Applications. The complexity introduced by the microservices architecture, discussed in Section 3.3.3, has led to a new term: Cloud Native Applications (CNA). CNA are portable applications that exploit the full benefits of cloud computing without being dependent on a specific cloud provider or infrastructure [73]. Features such as services scalability, registry, binding, orchestration, and monitoring are supported out of the box. However, a platform is required to act both as middleware and application server for those services.

In addition, as shown in Figure 15, cloud-native applications integrate well with CD, Microservices, DevOps, and Containers.

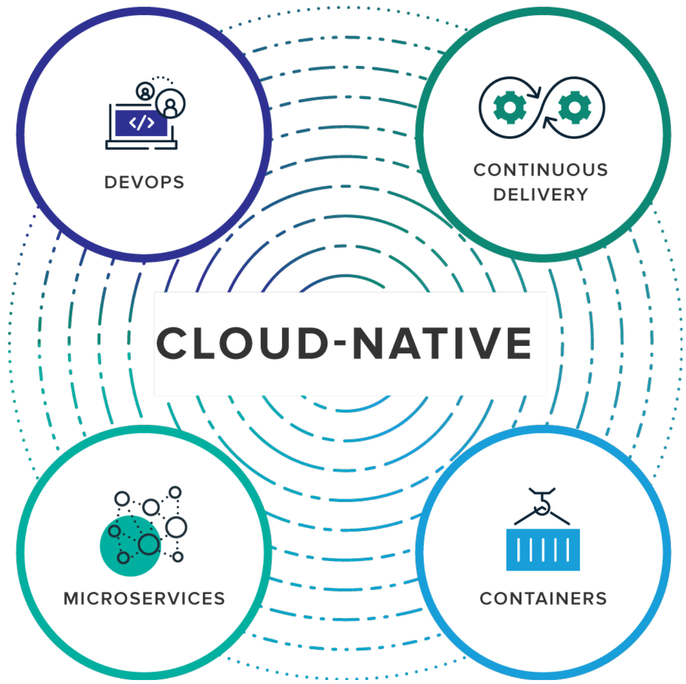


Figure 15: Cloud native applications external environment [73]

3.4 Main Characteristics of Cloud Applications

Cloud applications have special characteristics that make them different from traditional applications in many aspects [83]. These characteristics are:

1. **Scalability up & down (Elasticity):** In traditional applications, the scalability requirement includes scaling-up, so that a system should be able to handle a larger number of users if required, without modifying the software's code. This was normally achieved by vertical scalability [79]. In particular, vertical scalability is achieved by increasing systems resources, such as memory, storage or computing power [87]. On the other hand, horizontal scalability is widely used by enterprises, by adding extra nodes to the application cluster [25], but it is not common in small-medium organizations and businesses since it is relatively expensive, and not easy to configure and manage.

   Even though scaling-up is important in cloud applications to utilize large data volumes and a vast list of services [33], scaling down is also significant, because it will minimize resource utilization, which reduces the cost for service users [5, 88]. In addition, horizontal scalability is lighter and more cost-effective than vertical-scalability. In fact, application containers such as Docker [19] are the main enablers for lighter and more cost efficient horizontal scalability. These new trends (horizontal scalability and application containerization) led to the innovation of a new architecture, microservices architecture, which was discussed in Section 3.3.3.

   Designing an application for exact scalability needs is still a challenge, where having under-utilization, even with a small percentage, increases the cost more than actually needed, and over-utilization makes the services slower, and cause service users to look for alternatives [21]. Furthermore, in-advance testing and benchmarking of the cloud-application scalability can reduce the risk of downtime or load mis-estimation [31, 82].
2. **Support for different front-end technologies:** Currently, trends such as IoT and mobile devices have created the need for supporting widely different types of application clients (e.g. smart-phones, smart-cars, smart-televisions). A special design and architecture should be taken into consideration to support different types of front-ends without the need of modifying the back-end.
3. **Usage of Metrics:** Since most cloud applications are based on subscriptions and pay-as-you-go models, usage metrics should be taken into consideration from

the beginning since they will be the basis for financial billing [5, 80].

4. **Monitoring:** Application monitoring is required to directly ensure that expected quality attributes are being met at runtime, especially in non-normal scalability conditions, such as holidays for e-commerce platforms, or breaking stock market news for real-time trading applications. It may include frequent health checks, heart-beats, and resources visualizations.
5. **Offline support:** Even though Internet services have become more accessible and reliable over the years, customers still have access difficulties to the Internet in many locations and places (e.g. airplanes, underground floors, trains). In addition, with the rise of IoT, scientific devices and sensors may be deployed in some remote locations (e.g. deserts, mountains, oceans), which also may not have an available or reliable Internet connection. Consequently, an offline support feature is important. Having this feature gives service users the ability to use the service while disconnected from the Internet, which can be synchronized once re-connected later with the server. The offline support feature is critical for many applications, such as word editing tools, project management applications, and IoT devices.
6. **Configurability:** In multi-tenant cloud applications, clients should have the illusion of separate application instances, while service providers may maintain single instance to be able to maintain only one version and to achieve economies of scale. Designing the applications to be configurable and parameterized at runtime is important[2]. In fact, having the quality attribute of configurability can reduce support cost, and give more customization and preference features for clients, which can increase customer traction and reduce their turn-over [54]. Furthermore, variability modeling from Software Product Lines (SPL) [47] can also be implemented to achieve the configurability quality attribute [61][63].
7. **Data locality:** The decision on whether to pull or keep data on the cloud requires special attention and balance between performance, data transfer cost, and usability. In particular, data locality is important to improve performance. For example, keeping data on a server may be efficient for server-side processing (e.g. search, filters), however, it might be more efficient to pull data to client-side for visualization applications. Nevertheless, in general, data-locality can achieve better usability and processing performance [37].
8. **Quality of Service:** Finding a way for separating the quality of service for multi-tenant services is important to ensure a reliable service and the containment of the shared-fate issue discussed in Section 2.6.

The dynamic nature of cloud computing and the difference between physical environments and virtualized cloud environments plays an important role in distinguishing between the architectures of traditional and cloud applications [88].

3.5 DevOps and Cloud Applications Development Process

Developing and managing cloud applications has caused a serious issue of mis-coordination between development and operation. In fact, the microservices architecture is the main reason for the increase in this issue, as discussed in Section 3.3.3. To overcome this issue and potential conflicts, a new term was created: DevOps. The main concept behind DevOps is the idea that "you built it, you run it", where application/service developers are also responsible for supporting and maintaining their applications/services while in production [45] and reducing the friction that appears while in deployment and operation phases [6]. Another approach of DevOps is that the development and operation teams work closely with each other to reduce the gap and taking ownership of the project overall success [42]. In addition, this decreases the time between changing a system and reflecting that change into the live environment [11]. As of 2018, 84% of enterprises are adopting this approach. In fact, 30% of these companies implement this approach on a company-wide policy [74].

Adopting the DevOps approach and culture requires a lot of tooling and software infrastructure to be implemented, such as version control, continuous integration, continuous delivery, and artifacts repository [52].

In reference to the high diversity of roles involved in cloud application development (e.g. security, networks, business), DevOp has four main perspectives: (i) culture of collaboration where all team members from the different project life cycle stages have the required knowledge about the project, (ii) automation, continuous delivery, and deployment pipelines, (iii) high-level and accessible measurement and metrics, (iv) and sharing of knowledge, development, tools, techniques, and other aspects that can enable the required understanding for the system [43, 44]. Moreover, the knowledge, skills, and ability used in developing modern web-based applications were discussed in what is called "grounded theory" [9]. Figure 16 shows how DevOps changed the traditional structure of software development teams.

Moreover, the cloud has changed the role of the System Administrator to a Virtual System Administrator, where there is no need for any cabling/wiring, or server installation required, or any other manual activity, it is all now done through a web console that enables the network and the system to be administrated and managed virtually. This led to software developers and systems administrators being more collaborative and having more interaction [88].

Since DevOps is considered as a culmination of what the agile method started [64], with both encouraging running software over writing documentation [81], the main disadvantages of applying this approach is the risk produced when the developer leaves and not enough documentation is available.

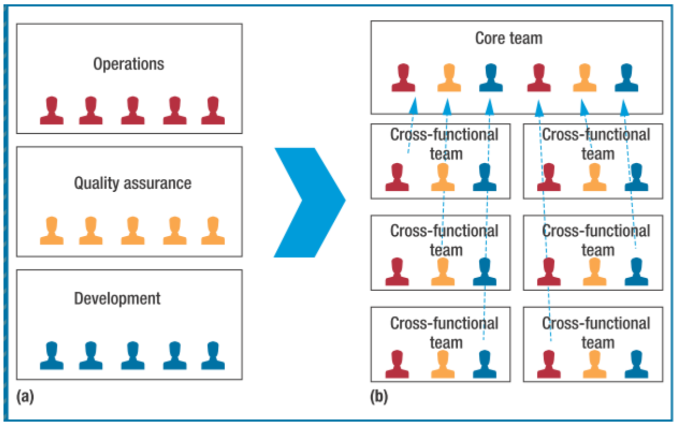


Figure 16: An example of DevOps team structure [11]

3.6 Benefits of Cloud Applications

The general benefits of cloud computing were discussed in Section 2.5, which also apply to cloud applications. In addition, in this section benefits of cloud applications are presented from the perspectives of both service providers and service users.

Common advantages are the almost zero upfront investment, just-in-time infrastructure, and reduced time to market [88].

For service providers, running single versions will simplify maintenance, and lower customer support consequently reduces the cost of operations; also, it can reduce the research cost. Furthermore, this will give the ability for small non-risky updates. From an operational perspective, service installation and deployment are easier, especially when utilizing the appropriate software infrastructure [51]. On the other hand, from an economic perspective, since organizations are not willing to pay a large amount of money for software anymore [80], providing applications on the cloud will enable providers to take advantage of this change in customer behavior to make more traction and profit. Moreover, software piracy is impossible, which is another major advantage for service providers [65].

The utilization of cloud applications can bring many advantages to service users. Firstly, low cost may be the most important factor. Secondly, data security may be better than on-premise applications, especially in small businesses, where most likely there is no dedicated support team to operate and support these applications. Furthermore, in general, cloud applications have better quality than on-premise applications and you always have access to the latest stable version of the system [13]. In fact, SaaS providers should invest in building higher quality software to ensure the increase of customers retention [34, 58].

3.7 Challenges of Cloud Applications

Cloud applications have many advantages; however, it also introduces many challenges and issues. Debugging a cloud application is not as easy as traditional application debugging. In addition, the support of multi-tenant, and adopting cloud native applications‘ properties discussed in Section 3.3.4 introduced an extra complexity for the application development, deployment and management [38].

Furthermore, even though building cloud applications based on economies of scale model sounds tempting, marketing cost is the main challenge for customer acquisition. In 2012, even with 90K customers, and a revenue of $2.3 billion per quarter, the profit margin for SalesForce was negative because of the high cost of sales and marketing to attract and keep customers [36].

Furthermore, the competition between service providers makes the customers more selective and the decision to switch to another service provider is easier than ever. Increasing the service cancellation cost of customers may be a solution, however, customers will not continue if the service is poor, or may sacrifice that extra cost if they found better quality elsewhere, so working on the application quality is significant to reduce customer turnover [58]. In fact, service providers need 12 months subscriptions on average to cover the expenses of a single customer [34].

Moreover, data integration and interoperability are challenging and include many concerns. These encompass difficulty in large data transmission, from both security and bandwidth perspectives; data integrity and support of transaction across the cloud; expensive data change detection; controlling data quality; and determining the original source of data [57].

4 Cloud Applications Adoption User Study

This section includes the details of the user study conducted to analyze the adoption of cloud computing and cloud applications in organizations from different levels.

4.1 Methodology

The user study is based on a survey conducted between the period of 24 to 31 December, 2018. It included 36 software technology practitioners and academics with various seniority levels, positions, and education. In addition, the participants were from different regions all over the world.

The user study survey was organized into three sections: Introductory Information, Participant Information, and Cloud Applications.

- Introductory Information: In the survey, this section included an introduction about the user study, so that participants could understand the goals. Then, it followed by a consent required by the Institutional Review Boards (IRB) process of the University of Nevada, Reno (UNR). After that, it included the email address of the participant, then (optionally) his/her name. Finally, we asked the participants whether we could contact them for future feedback or questions if needed.
- Participant Information: The second section, aimed to get general information about the participants themselves to ensure that we have a representative group. The

information questions included the participant's overall experience in the software development field, highest academic degree, current job title, years of experience, and their day-to-day activities working on software development projects.

- Cloud Applications: This section included questions about cloud applications significance, cost, and limitations. Specifically, the first question was whether choosing cloud applications is more beneficial than the traditional approaches for green-field projects. The second question was to verify if there is a shortage in experienced software engineers who can develop high-quality cloud-based applications. The third, fourth, and fifth questions were about the cost of software development, maintenance, and operation of cloud versus traditional applications respectively. The last question in this section was about the risk of uplifting traditional applications (i.e., migrating applications to the cloud).

4.2 Participants

As shown in Figure 17, the survey included a wide range of participants with different professional levels in cloud computing. Around 33% were professionals, 36% intermediate, and 27% experts in the field of study.

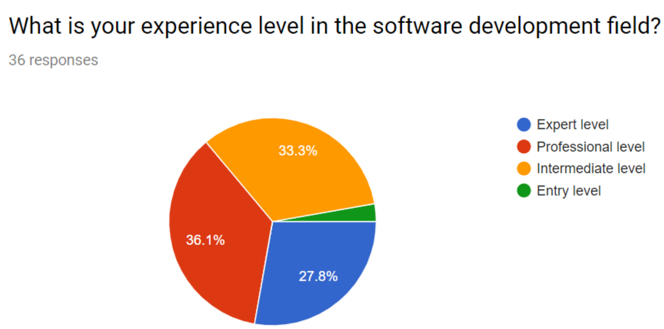


Figure 17: Experience levels of the user study participants

In addition, the participants came from different academic backgrounds, as shown in Figure 18. Around 55% have a bachelor degree, 30% a Masters degree, and 14% a PhD degree.

Moreover, Figure 19 shows the participants' day-to-day involvement in software development project activities, such that we could get a clearer understanding of the different points of view. The day involvement categories included: research, project management, business analysis, software architecture, software development, software implementations, testing, teams management, technical support, and teaching or training. As shown in the figure, the categories included participants with 72% of software development, 44% teams management and software architecture, and 50% software implementation.

Also, we asked a question about the job titles of the participants. The participants titles include: Solution Managers, System Architects, Associate and Assistant Professors, IT Directors, System Analysts, PhD Students, Implementation Managers, Services Manager, Integration Specialist, Technical Team Leader, Head of QA, Software Engineers, Senior Software Engineers, and Software Development Consultants.

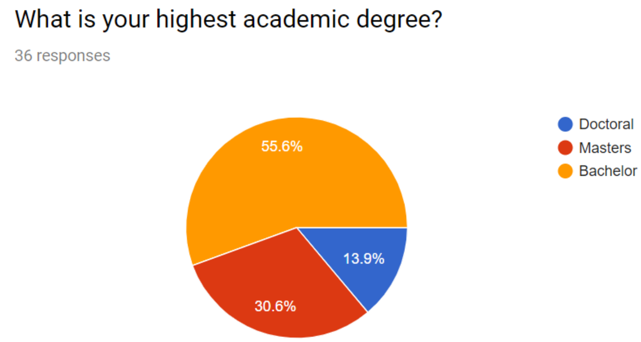


Figure 18: Academic levels of the participants in the user study

Finally, we asked the participants about their total years of job experience in the IT field. This total ranged from 3 to 20 years of experience.

4.3 Results and Discussion

The results of the user study are shown in Table 2, which includes the questions and their aggregate responses.

Nowadays, a cloud-based approach of software application development is one of the important current trends in the software engineering industry. However, we believe that there are many challenges surrounding the adoption of this approach in many organizations.

Before trying to identify these problems, we wanted to be sure that building applications following the cloud approach is the preferred way especially in green-field projects over the traditional approach, in organizations in different domains. As shown in Figure 20 more than 82% of the participants strongly agreed or agreed with the statement "For new software applications, choosing a cloud-based approach can be more beneficial than choosing a traditional approach." The average of the responses for this statement was 4.18/5.0, where 5 is Strongly Agree and 1 is Strongly Disagree.

In addition, and as shown in Table 2, overr 50% of the participants think that the development, maintainability, and operational cost of cloud applications is lower than traditional approaches. As shown in the table, the average of three related responses were 2.71, 2.72, and 2.80 respectively, where lower values means that the cloud is cheaper. The goal behind designing the survey for these questions in an opposite direction was to ensure that the participants completed the survey with reasoned inputs and did not rush their answers.

The above results shows the significance of cloud computing and cloud applications in reducing the cost of development, maintenance, and operations.

However, since averages of development, maintainability, and operational cost are all close to the median, we think that these

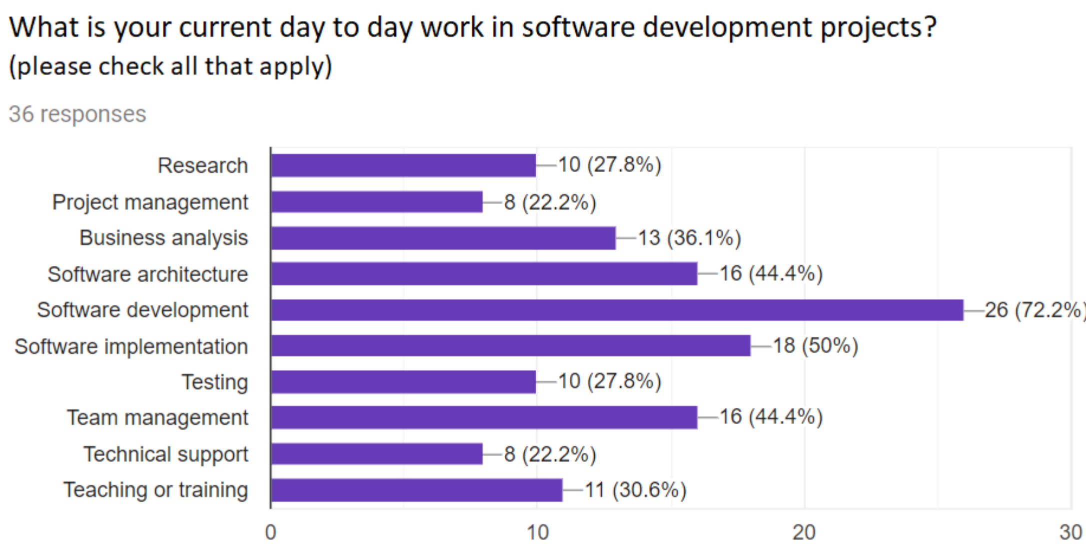


Figure 19: Involvement of the participants in software development activities

Table 2: The results of the user study

Question *Ranking Value*	**Strongly Disagree** (1)	**Disagree** (2)	**Neutral** (3)	**Agree** (4)	**Strongly Agree** (5)	**Average**
For new software applications, choosing a cloud-based approach can be more beneficial than choosing traditional approaches	-	1 (2.8%)	3 (8.3%)	19 (52.8%)	11 (30.6%)	4.18
There is a shortage of experienced software engineers who can develop high-quality cloud applications	-	-	10 (27.8%)	16 (44.4%)	10 (27.8%)	4
The cost of software development of cloud based applications is higher than cost of developing traditional software	1 (2.8%)	18 (50%)	6 (16.7%)	10 (27.8%)	-	2.71
The maintainability cost of cloud based applications is higher than that of traditional software applications	4 (11.1%)	19 (52.8%)	4 (11.1%)	6 (16.7%)	2 (5.6%)	2.72
The operational cost of cloud based applications is higher than that of traditional software applications	3 (8.3%)	15 (41.7%)	5 (13.9%)	10 (27.8%)	2 (5.6%)	2.80
Migrating applications developed using traditional approaches to be cloud-based is an expensive and risky process	1 (2.8%)	5 (13.9%)	8 (22.2%)	17 (47.2%)	2 (5.6%)	3.42

questions require more investigation.

On the other hand, all the previous questions were related to new projects in a green field situation. So, we also wanted to get the participants opinion about brown field projects to check if migrating current applications to be cloud-based is considered a risky process for the organizations, and the results showed that more than 40% of the participants think it is risky to uplift on-premises applications (e.g. migrating applications to the cloud), with an average of 3.42/5. However, we also think that this requires more investigation, since based on the feedback of some the participants privacy, compliance, and security concerns are still dominant.

So the question is, if most of the participants think that cloud applications are more beneficial than the monolithic based, then why are there still many on-premises applications in many organizations? The answer to this question is shown in Figure 21, where more than 72% of the participants believed that there is a shortage in experienced software engineers

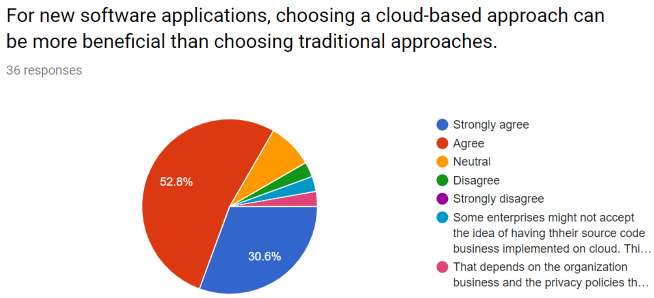


Figure 20: Cloud-based approach versus the traditional approach

who can develop high-quality cloud software applications, with an average of 4.0/5.0 of the participants thinking that there is a shortage in this area. This confirms the results of RightScale [75], discussed in Section 2.6, that the lack of resources and expertise has become a major barrier for adopting cloud computing. While the results in the RightScale report focused on cloud computing in general, our study shows that they could also be applied to cloud applications in particular.

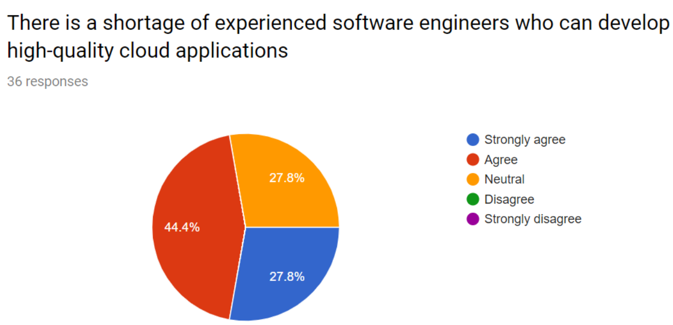


Figure 21: Shortage in experienced cloud application developers

We believe that the above results show there is a significant need for an approach that enables developing cloud-based applications in an efficient and effective way, without requiring particular expertise.

5 Conclusion and Future Work

This article has overviewed and discussed cloud computing and cloud applications. History and evolution of cloud computing were presented, and how the expensive hardware and infrastructure, along with the absence of economies of scale, might be the main reasons for delaying its adoption. These were followed by how the Internet, low-cost commodity-computers based data centers, smart-phones, and economic crisis played important roles in moving forward in cloud computing and offering computer services as utilities. The main advantages of cloud computing were presented as well, such as reducing total cost of ownership, time to market, and liabilities delegation. On the other hand, disadvantages and challenges were also examined, such as security, loss of control, regulations, and political conflicts. Moreover, the effects of cloud computing on startups, economic disciplines, and hardware businesses were also discussed.

In addition, the standard services delivery models (IaaS, PaaS, and SaaS) were presented. However, we think SaaS term is being misused, and service delivery models require a standardized new taxonomy. In particular, software is a generic term that includes operating systems, platforms, applications, and even virtualization technologies such as hypervisors. Consequently, all service delivery models are SaaS in some way. The main issue with this is that future regulations of taxation, billing, and licenses may be based on the categories of the software provided.

Even though we believe the work presented in this article is sufficiently comprehensive to serve as an introductory survey for cloud computing and cloud applications, having more details about the architectural styles and patterns for building cloud applications can also be beneficial for software architects and developers. Additionally, discussion about PaaS platforms will enable them to choose whether to build on one of the available options, develop a platform on top of another one, or even create from scratch a new domain-specific platform.

A primary issue with the current available cloud computing services and technologies is the lack of standardization. This increases the risk of service provider lock. Even though this can be mitigated by creating an abstract layer between the service user and provider, this will raise the development cost and may introduce buggy features, and will not allow full utilization of the services provided. In fact, we think Amazon is leading the de-facto standardization of cloud computing following the dominant design concept [86]. However, this situation is risky because long-term stability is not guaranteed, and increasing the number of proprietary services, technologies, and protocols are more likely to occur.

We also believe that standardization is significant because it could solve many constraints, risks, and challenges, and enable more user traction. Moreover, it may eliminate privacy and data constraint issues, and support interoperability.

On the other hand, since the design and architecture of cloud applications are challenging and require specialized expertise, the utilization of native cloud platforms such a Cloud Foundry, and the work presented in some published works [6][71] may reduce the cost, and enable proper utilization of cloud resources. In fact, worldwide spending on PaaS is expected to increase from 11% in 2015 to 17% in 2020, which may be a sign of emerging need of supporting native cloud applications out of the box [30]. However, the relative novelty of this field and its lack of standardization open the door for future research in designs and methods for building native cloud application platforms.

Finally, a user-study survey to understand the actual adoption of cloud-based approach of developing new software systems in industry and academia and in different domains was presented. The user study included 36 professionals from academia and industry from different regions of the world, with different expertise and job roles. The results of the study show that cloud-applications were the preferred approach for most of the participants, who also considered that the cost of adopting cloud applications is lower than the cost of traditional approaches. However, the participants indicated that the lack of expertise is in their view the main challenge of adopting cloud-based applications. Because the results on this item were close to the median, we believe that there is a need to further investigate the actual cost of development, maintenance, and operations of cloud-based applications versus the traditional on-premise category. In addition, finding an approach that enables effective and efficient development of cloud-based applications without the need for special expertise will be both useful and significant and will increase the adoption of cloud computing.

Acknowledgment

This material is based upon work supported by the National Science Foundation under grant number IIA-1301726. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The user study was approved by IRB at the University of Nevada, Reno (IRB #1362116-1).

References

[1] Allan Afuah and Christopher L Tucci. *Internet Business Models and Strategies*. McGraw-Hill New York, USA, 2001.

[2] Saiqa Aleem, Faheem Ahmed, Rabia Batool, and Asad Khattak. "Empirical Investigation of Key Factors for SaaS Architecture Dimension". *IEEE Transactions on Cloud Computing*, 2019. `https://ieeexplore.ieee.org/document/8669948` (Early Access).

[3] Amazon Web Services, Inc. "Registry of Open Data on AWS". `https://aws.amazon.com/public-datasets/` (Date last accessed January 20, 2021).

[4] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. "A View of Cloud Computing". *Commun. ACM*, 53(4):50–58, April 2010. `https://doi.org/10.1145/1721654.1721672`.

[5] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy H Katz, Andrew Konwinski, Gunho Lee, David A Patterson, Ariel Rabkin, Ion Stoica, and Mate Zaharia. "Above the Clouds: A Berkeley View of Cloud Computing". Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, February 2009. `http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html` (Date last accessed January 20, 2021).

[6] Leonardo G Azevedo, Leonardo P Tizzei, Maximilien de Bayser, and Renato Cerqueira. "Installation Service: Supporting Deployment of Scientific Software as a Service". In *the 7th IEEE Latin-American Conference on Communications (LATINCOM), pp. 1–6*, 2015. `https://ieeexplore.ieee.org/document/7430148`.

[7] Armin Balalaie, Abbas Heydarnoori, and Pooyan Jamshidi. "Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture". *IEEE Software*, 33(3):42–52, 2016. `https://ieeexplore.ieee.org/document/7436659`.

[8] Prith Banerjee, Richard Friedrich, Cullen Bash, Patrick Goldsack, Bernardo Huberman, John Manley, Chandrakant Patel, Parthasarathy Ranganathan, and Alistair Veitch. "Everything as a Service: Powering the Sew Information Economy". *Computer*, 44(3):36–43, 2011. `https://ieeexplore.ieee.org/document/5719575`.

[9] Soon K. Bang, Sam Chung, Young Choh, and Marc Dupuis. "A Grounded Theory Analysis of Modern Web Applications: Knowledge, Skills, and Abilities for DevOps". In *Proceedings of the 2nd Annual Conference on Research in Information Technology*, RIIT '13, New York, NY, USA, pp. 61–62, 2013. Association for Computing Machinery. `https://doi.org/10.1145/2512209.2512229`.

[10] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. "Xen and the Art of Virtualization". *SIGOPS Oper. Syst. Rev.*, 37(5):164–177, October 2003. `https://doi.org/10.1145/1165389.945462`.

[11] Len Bass, Ingo Weber, and Liming Zhu. *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional, USA, 2015.

[12] Rajkumar Buyya and Kris Bubendorfer, editors. *"Market-Oriented Grid and Utility Computing"*, volume 75 of *Wiley Series on Parallel and Distributed Computing* (Albert Y. Zomaya, Series Editor). John Wiley & Sons, USA, November 2009.

[13] Vidyanand Choudhary. "Software as a Service: Implications for Investment in Software Development". In *the 40th Annual Hawaii International Conference on System Sciences (HICSS 2007)*. IEEE, pp. 209a–209a, 2007. `https://ieeexplore.ieee.org/document/4076800`.

[14] Pushpinder Kaur Chouhan, Feng Yao, and Sakir Sezer. "Software as a Service: Understanding Security Issues". In *Science and Information Conference (SAI)*. IEEE, pp. 162–170, 2015. `https://ieeexplore.ieee.org/abstract/document/7237140`.

[15] CISCO. "Cloud Computing Services - PBXL - CISCO". `http://pbxl.co.jp/en/saas-paas-iaas/` (Date last accessed April 15, 2019).

[16] Douglas Crockford. "JSON (JavaScript Object Notation)". `https://www.json.org/` (Date last accessed April 15,

2019).

[17] Akon Dey, Gajanan Chinchwadkar, Alan Fekete, and Krishna Ramachandran. “Metadata-as-a-service”. In *the 31st IEEE International Conference on Data Engineering Workshops (ICDEW)*. IEEE, pp. 6-9, 2015. https://ieeexplore.ieee.org/document/7129536.

[18] Tim Dixon and Stephen Hargitay. *Software Selection for Surveyors: A Guide and Directory for surveyors in General Practice*. Springer, USA, 1989. https://link.springer.com/book/10.1007/978-1-349-21696-3.

[19] Docker Inc. “Empowering App Development for Developers — Docker”, 2019. https://www.docker.com (Date last accessed January 21. 2021).

[20] Markus Eisele. *Modern Java EE Design Patterns: Building Scalable Architecture for Sustainable Enterprise Development*. O'Reilly Media, USA, 2016. https://www.oreilly.com/library/view/modern-java-ee/9781492042266/.

[21] Javier Espadas, Arturo Molina, Guillermo Jiménez, Martín Molina, Raúl Ramírez, and David Concha. “A Tenant-Based Resource Allocation Model for Scaling Software-as-a-Service Applications Over Cloud Computing Infrastructures”. *Future Generation Computer Systems*, 29(1):273–286, 2013. Including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures https://doi.org/10.1016/j.future.2011.10.013.

[22] Robert M Fano and Fernando J Corbató. Time-Sharing on Computers. *Scientific American*, 215(3):128–140, 1966. https://doi.org/10.1038/SCIENTIFICAMERICAN0966-128.

[23] Xinyang Feng, Jianjing Shen, and Ying Fan. “REST: An alternative to RPC for Web Services Architecture”. In *the First International Conference on Future Information Networks (ICFIN 2009)*. IEEE, pp. 7–10, 2009. https://ieeexplore.ieee.org/document/5339611.

[24] Fortune.com. “Here's Why Amazon's Cloud Suffered a Meltdown This Week”, 2017. http://fortune.com/2017/03/02/amazon-cloud-outage/ (Date last accessed April 15, 2019).

[25] Ian Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu. “Cloud Computing and Grid Computing 360-Degree Compared”. In *2008 Grid Computing Environments Workshop*. IEEE, pp. 1–10, 2008. https://ieeexplore.ieee.org/document/4738445.

[26] Martin Fowler and James Lewis. “Microservices: a Definition of this New Architectural Term”, March 2014. https://martinfowler.com/articles/microservices.html (Date last accessed April 15, 2019).

[27] Free Software Foundation, Inc. “Front Page - Free Software Foundation - Working Together for Free Software”. https://www.fsf.org/ (Date last accessed April 15, 2019).

[28] Free Software Foundation, Inc. “The GNU Operating System and the Free Software Movement”. https://www.gnu.org (Date last accessed April 15, 2019).

[29] Julia M Fromholz. “The European Union Data Privacy Directive”. *Berkeley Technology Law Journal*, 15:461, 2000. https://lawcat.berkeley.edu/record/1117206?ln=en.

[30] John F. Gantz and Pam Miller. “The Salesforce Economy: Enabling 1.9 Million New Jobs and $389 Billion in New Revenue Over the Next Five Years”. Technical Report #US41691316, IDC, September 2016. https://www.salesforce.com/content/dam/web/en_us/www/academic-alliance/datasheets/IDC-salesforce-economy-study-2016.pdf.

[31] Jerry Gao, Pushkala Pattabhiraman, Xiaoying Bai, and Wei-Tek Tsai. “SaaS Performance and Scalability Evaluation in Clouds”. In *IEEE 6th International Symposium on Service Oriented System Engineering (SOSE)*. IEEE, pp. 61–71, 2011. doi: 10.1109/SOSE.2011.6139093 https://ieeexplore.ieee.org/abstract/document/6139093.

[32] Simson L. Garfinkel. “An Evaluation of Amazon's Grid Computing Services: EC2, S3, and SQS”. Technical Report TR-08-07, Harvard Computer Science Group, 2007. http://nrs.harvard.edu/urn-3:HUL.InstRepos:24829568.

[33] David Garlan. “Software Architecture: A Travelogue”. In *Proceedings of the International Conference on Future of Software Engineering, pp. 29-–39*, FOSE 2014, New York, NY, USA, 2014. Association for Computing Machinery. https://doi.org/10.1145/2593882.2593886.

[34] Yizhe Ge, Shan He, Jingyue Xiong, and Donald E Brown. “Customer Churn Analysis for a Software-as-a-service Company”. In *Systems and Information Engineering Design Symposium (SIEDS)*. IEEE, pp. 106-111, 2017. https://ieeexplore.ieee.org/document/7937698.

[35] Google Inc. “App Engine Application Platform — Google Cloud”. https://cloud.google.com/appengine/ (Date last accessed April 15, 2019).

[36] Eugene Gorelik. *“Cloud Computing Models”*. Master's Thesis, Massachusetts Institute of Technology, 2013. https://dspace.mit.edu/handle/1721.1/79811 (Date last accessed (01/21/2021).

[37] Jim Gray. “Distributed Computing Economics”. *Queue*, 6(3):pp. 63–68, May 2008. https://dl.acm.org/doi/10.1145/1394127.1394131.

[38] Chang Jie Guo, Wei Sun, Ying Huang, Zhi Hu Wang, and Bo Gao. “A Framework for Native Multi-Tenancy Application Development and Management”. In *The 9th IEEE International Conference on e-commerce Technology and the 4th IEEE International Conference on Enterprise Computing (CEC/EEE)*. IEEE, pp. 551–558, 2007. https://ieeexplore.ieee.org/document/4285271.

[39] James Hamilton. “Internet-Scale Service Efficiency”,

September 2008. Keynote Presentation at Large-Scale Distributed Systems and Middleware (LADIS) Workshop (Program: http://www.cs.cornell.edu/projects/ladis2008/program.html) (Slides: https://perspectives.mvdirona.com/2008/09/internet-scale-service-efficiency/).

[40] Neil B. Harrison and Paris Avgeriou. "How Do Architecture Patterns and Tactics Interact? A Model and Annotation". *J. Syst. Softw.*, 83(10):1735–1758, October 2010. https://doi.org/10.1016/j.jss.2010.04.067.

[41] Adam Holt, Simon Flannery, Sanjay Devgan, Atif Malik, Nathan Rozof, CFA1 Adam Wood, Patrick Standaert, Francois Meunier, Jasmine Lu, Grace Chen, et al. "Cloud Computing Takes Off". *Morgan Stanley Blue Paper*, 2011. http://www.dabcc.com/documentlibrary/file/cloud_computing.pdf (Date last accessed: 01/21/2021).

[42] Rick Kazman Humberto Cervantes. *Designing Software Architectures: A Practical Approach*. SEI Series in Software Engineering. Addison-Wesley Professional, USA, 1st edition, 2016.

[43] Jez Humble and David Farley. *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*. Pearson Education, USA, 2010.

[44] Jez Humble and Joanne Molesky. "Why Enterprises Must Adopt Devops to Enable Continuous Delivery". *Cutter IT Journal*, 24(8):6–12, August 2011. https://www.cutter.com/sites/default/files/itjournal/fulltext/2011/08/itj1108.pdf.

[45] Michael Hüttermann. *"DevOps for Developers"*. Apress, USA, 2012. https://www.apress.com/gp/book/9781430245698.

[46] Bobbie Johnson. "Cloud Computing is a Trap, Warns GNU Founder Richard Stallman". *The Guradian*, September 2008. https://www.theguardian.com/technology/2008/sep/29/cloud.computing.richard.stallman (Date last accessed January 21, 2021).

[47] Timo Käköla and Juan Carlos Duenas, editors. *"Software Product Lines: Research Issues in Engineering and Management"*. Springer, USA, 2006. https://www.springer.com/gp/book/9783540332527.

[48] Lori M Kaufman. *"Data Security in the World of Cloud Computing"*. *IEEE Security & Privacy*, 7(4):61 – 64, 2009. https://ieeexplore.ieee.org/document/5189563.

[49] Tom Killalea. "The Hidden Dividends of Microservices". *Commun. ACM*, 59(8):42—45, July 2016. doi: 10.1145/2948985.

[50] Jalal Kiswani, Sergiu M Dascalu, and Frederick C Harris Jr. "Cloud-RA: A Reference Architecture for Cloud Based Information Systems". In *ICSOFT*, pp. 883–888, 2018.

[51] Jalal Kiswani, Muhanna Muhanna, Sergiu Dascalu, and Frederick Harris. "Software Infrastructure to Reduce the Cost and Time of Building Enterprise Software Applications: Practices and Case Studies". In *the proceedings of ISCA 26th International Conference on Software Engineering and Data Engineering (SEDE 2017)*. pp. 93-98, ISCA, 2017. https://www.researchgate.net/publication/322267534_Software_Infrastructure_to_Reduce_the_Cost_and_Time_of_Building_Enterprise_Software_Applications_Practices_and_Case_Studies.

[52] Jalal Kiswani, Muhanna Muhanna, Sergiu Dascalu, and Frederick Harris. "Software Infrastructure to Reduce the Cost and Time of Building Enterprise Software Applications: Practices and Case Studies". In *Proceedings of ISCA 26th International Conference on Software Engineering and Data Engineering (SEDE 2017). ISCA*, 2017.

[53] Nir Kshetri. "Cloud Computing in Developing Economies". *Computer*, 43(10):47–55, 2010. https://ieeexplore.ieee.org/document/5530325.

[54] Phillip A Laplante, Jia Zhang, and Jeffrey Voas. "What's in a Name? Distinguishing between SaaS and SOA". *IT Professional*, 10(3), 2008. https://ieeexplore.ieee.org/document/4525542.

[55] Rick Kazman LenBass, Paul Clements. *Software Architecture in Practice*. SEI Series in Software Engineering. Addison-Wesley Professional, USA, 3rd Edition, 2012.

[56] Björn Link and Andrea Back. "Classifying Systemic Differences Between Software as a Service and On Premise Enterprise Resource Planning". *Journal of Enterprise Information Management*, 28(6):808–837, 2015. https://doi.org/10.1108/JEIM-07-2014-0069.

[57] Feng Liu, Weiping Guo, Zhi Qiang Zhao, and Wu Chou. "SaaS Integration for Software Cloud". In *3rd International Conference on Cloud Computing (CLOUD)*. pp. 402–409, IEEE, 2010. https://ieeexplore.ieee.org/document/5557968.

[58] Dan Ma and Robert J Kauffman. "Competition Between Software-as-a-service Vendors". *IEEE Transactions on Engineering Management*, 61(4):717–729, 2014. https://ieeexplore.ieee.org/document/6857369.

[59] Peter Mell and Tim Grance. "The NIST Definition of Cloud Computing". Technical Report SP 800-145, National Institute of Standards and Technology, Computer Security Division, Information Technology Laboratory, Gaithersburg, MD, September 2011. https://csrc.nist.gov/publications/detail/sp/800-145/final.

[60] Microsoft Corp. "Outlook – Free Personal Email and Calendar from Microsoft", 1996. https://outlook.live.com/owa/ (Date last accessed January 21, 2021).

[61] Ralph Mietzner, Andreas Metzger, Frank Leymann, and Klaus Pohl. "Variability Modeling to Support

Customization and Deployment of Multi-tenant-aware Software as a Service Applications". In *Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems*. IEEE, pp. 18–25, 2009. https://ieeexplore.ieee.org/document/5068815.

[62] Morgan Stanley Research. "The Mobile Internet Report: Ramping Faster than Desktop Internet, the Mobile Internet Will Be Bigger than Most Think". Technical report, Morgan Stanley & Co. Incorporated, December 2009.

[63] Taewoo Nam and Keunhyuk Yeom. "Ontology Model to Support Multi-tenancy in Software as a Service Environment". In *the International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, pp. 146–151, 2014. https://ieeexplore.ieee.org/document/6984188.

[64] Linda Northrop. "Trends and New Directions in Software Architecture", October 2014. Keynote Talk: Grace Hopper Celebration of Women in Computing: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=438673 (Video, Trascript, Slides).

[65] Arto Ojala. "Software-as-a-Service Revenue Models". *IT Professional*, 15(03):pp. 54–59, may 2013. https://doi.ieeecomputersociety.org/10.1109/MITP.2012.73.

[66] Oracle Inc. "Cloud Infrastructure — Oracle". https://www.oracle.com/cloud/platform.html (Date last accessed January 21, 2021).

[67] Oracle NetSuite. "Business Software, Business Management Software – NetSuite", 1998. https://www.netsuite.com/portal/home.shtml?noredirect=T (Date last accessed January 21, 2021).

[68] Arthur O'Sullivan and Steven M. Sheffrin. *Economics: Principles in Action*. Pearson Prentice Hall, USA, 2003.

[69] Michael Palmer and Michael Walters. *Guide to Operating Systems*. Cengage Learning, USA, 4th edition, 2012. https://www.amazon.com/Guide-Operating-Systems-Michael-Palmer/dp/1111306362.

[70] Douglas Parkhill. *The Challenge of the Computer Utility*. Addison-Wesley Educational Publishers Inc. US, USA, 1966.

[71] "Petcu, Dana and Macariu, Georgiana and Panica, Silviu and Crciun, Ciprian". Portable cloud applications-from theory to practice. *Future Gener. Comput. Syst.*, 29(6):417—-1430, August 2013. https://doi.org/10.1016/j.future.2012.01.009.

[72] C Pettey. "Gartner Says Worldwide Public Cloud Services Market to Grow 18 Percent in 2017". *Gartner, Press Release*, 2017. https://www.gartner.com/en/newsroom/press-releases/2017-02-22-gartner-says-worldwide-public-cloud-services-market-to-grow-18-percent-in-2017#:~:text=The%20worldwide%20public%20cloud%20services,%2C%20according%20to%20Gartner%2C%20Inc. (Date last accessed January 20, 2021).

[73] Inc Pivotal Software. "Spring Cloud-Native". https://pivotal.io/cloud-native (Date last accessed April 15, 2019).

[74] RightScale. "State of the Cloud Report". Technical report, RightScale, 2017. Slides:(https://www.slideshare.net/rightscale/rightscale-2017-state-of-the-cloud).

[75] RightScale. "State of the Cloud Report, Date to Navigate your Multi Cloud Strategy". Technical report, RightScale, 2018.

[76] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. "A Taxonomy and Survey of Cloud Computing Systems". In *Fifth International Joint Conference on INC, IMS and IDC (NCM'09)*. IEEE, pp. 44–51, 2009. https://ieeexplore.ieee.org/document/5331755.

[77] Remi Sahl, Paco Dupont, Christophe Messager, Marc Honnorat, and Tran Vu La. "High-Resolution Ocean Winds: Hybrid-Cloud Infrastructure for Satellite Imagery Processing". In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*. IEEE, 2018. https://ieeexplore.ieee.org/document/8457895.

[78] SalesForce.com. "CRM Software from Salesforce.com - Customer Relationship Management - Salesforce.com", 2021. https://www.salesforce.com/crm/ (Date last accessed January, 21, 2021).

[79] Theo Schlossnagle. *"Scalable Internet Architectures"*. Pearson Education, USA, 2006.

[80] Ludwig Siegele. "Let it Rise: a Special Report on Corporate IT". Special report, The Economist, 2008. https://www.economist.com/special-report/2008/10/23/let-it-rise (Date last accessed January 20, 2021).

[81] Sommerville. *"Software Engineering"*. Addison Wesley, USA, 10th Edition, 2016.

[82] Wei-Tek Tsai, Yu Huang, and Qihong Shao. "Testing the Scalability of SaaS Applications". In *IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*. IEEE, pp. 1–4, 2011. https://ieeexplore.ieee.org/document/6166245.

[83] WeiTek Tsai, XiaoYing Bai, and Yu Huang. "Software-as-a-service (SaaS): Perspectives and Challenges". *Science China Information Sciences*, 57(5):1–15, 2014. https://doi.org/10.1007/s11432-013-5050-z.

[84] Matteo Turilli, Antonino Vaccaro, and Mariarosaria Taddeo. "Internet Neutrality: Ethical Issues in the Internet Environment". *Philosophy & Technology*, 25(2):133–151, 2012. https://doi.org/10.1007/s13347-011-0039-2.

[85] Turner, Mark and Budgen, David and Brereton, Pearl. "Turning Software into a Service". *Computer*, 36(10):38–44, 2003. https://ieeexplore.ieee.org/document/1236470.

[86] James Utterback. *The Dynamics of Innovation*.

EDUCAUSE, Boulder, CO, USA, 1994.

[87] Vaquero, Luis M. and Rodero-Merino, Luis and Buyya, Rajkumar. "Dynamically Scaling Applications in the Cloud". *SIGCOMM Comput. Commun. Rev.*, 41(1):45–52, January 2011. https://doi.org/10.1145/1925861.1925869.

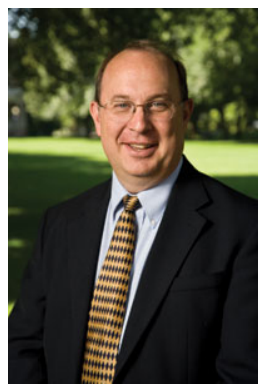
[88] Jinesh Varia. "Architecting for the Cloud: Best Practices". *Amazon Web Services*, 1:1–21, 2011. (Slides https://www.slideshare.net/AmazonWebServices/best-practices-in-architecting-for-the-cloud-webinar-jinesh-varia).

[89] Shiliang Wu, Hans Wortmann, and Chee-Wee Tan. "A Pricing Framework for Software-as-a-service". In *the Fourth International Conference on Innovative Computing Technology (INTECH)*. IEEE, pp. 152-157, 2014. https://ieeexplore.ieee.org/document/6927738.

[90] Yahoo Inc. "yahoo", 1996. https://mail.yahoo.com (Date last accessed April 15, 2019).

Jalal H. Kiswani is an academic, digital transformation advisor, and entrepreneur with more than 20 years of experience in industry and academia. In 2019 he received a PhD degree in Computer Science and Engineering from the University of Nevada, Reno, USA. He also holds a Master's degree in Enterprise Systems Engineering received in 2016 from the Princess Sumaya University and the German-Jordanian University, Amman, Jordan, as well as a Bachelor's degree in Computer Science earned in 2002 from Mu'tah University, Karak, Jordan. He is a certified expert by Oracle and Sun Microsystems in the Java technology; in particular, he is a certified Java Programmer, Java Developer, Web Components Developer, Business Components Developer, and Java Server Faces developer. He is the founder of Cloud-Wizard low-code no-code platform, Solid-Soft for information technology solutions, and Final Solutions for training and consulting. Currently, he is an Assistant Professor in the Computing and Informatics School at Al-Hussein Technical University (HTU), a technical digital transformation advisor at Arab Bank, and a cloud platform architect at cloud-wizard.com.

Sergiu M. Dascalu is a Professor in the Department of Computer Science and Engineering at the University of Nevada, Reno (UNR), which he joined in July 2002. He received his PhD degree in Computer Science (2001) from Dalhousie University, Canada and a Master's degree in Automatic Control and Computers (1982) from the Polytechnic of Bucharest, Romania. At UNR he is also the Director of the Software Engineering Laboratory (SOELA) and the Co-Director of the Cyberinfrastructure Lab (CIL). Since joining UNR, he has worked on research projects funded by federal agencies (NSF, NASA, DoD-ONR) as well as the industry. He has advised 11 PhD and over 50 Master students. He received several awards, including the 2009 Nevada Center for Entrepreneurship Faculty Advisor Award, the 2011 UNR Outstanding Undergraduate Research Faculty Mentor Award, the 2011 UNR Donald Tibbitts Distinguished Teacher of the Year Award, the 2014 CoEN Faculty Excellence Award, and the 2019 UNR Vada Trimble Outstanding Graduate Mentor Award. He is a Senior Member of the ACM.



Frederick C. Harris, Jr. received his BS and MS degrees in Mathematics and Educational Administration from Bob Jones University, Greenville, SC, USA in 1986 and 1988 respectively. He then went on and received his MS and PhD degrees in Computer Science from Clemson University, Clemson, SC, USA in 1991 and 1994 respectively. He is currently a Professor in the Department of Computer Science and Engineering and the Director of the High Performance Computation and Visualization Lab at the University of Nevada, Reno, USA. He is also the Nevada State EPSCoR Director and the Project Director for Nevada NSF EPSCoR. He has published more than 250 peer-reviewed journal and conference papers along with several book chapters, and has been co-editor of 13 books. He has had 14 PhD students and 79 MS Thesis students finish under his supervision. His research interests are in parallel computation, simulation, computer graphics, and virtual reality. He is also a Senior Member of the ACM, and a Senior Member of the International Society for Computers and their Applications (ISCA).