# A brief study on Software as a Service in Cloud Computing Paradigm

Waleed S Alnumay*
King Saud University, Riyadh, SA,wnumay@ksu.edu.sa

**Abstract**

Software as a Service (SaaS) is a paradigm that provides end users or clients easy and seamless access to various applications, using the Internet without requiring any infrastructure or related software. To virtualize the access to applications and functionalities, SaaS providers use the cloud-computing environment to rent resources, thereby reducing both the capital and operational expenditure. SaaS uses the cloud-computing infrastructure to distribute applications to many users, irrespective of their location or infrastructure capacity. This one-to-many model with centralized control has the potential to transform the behavior of traditional IT architecture, its pricing, partnering, and management. With the emergence of SaaS as a delivery system, the whole software environment is moving toward a producer-consumer system, where both are distributed globally. We point out the motivation for accepting the new technologies, such as to reduce the expense, energy consumption, maintenance, etc. We also point out the challenges and risks associated with the paradigm shift. The present study categorically presents state-of-the-art research on the security issues of collaborative SaaS cloud computing and integrating service-level-agreement and quality-of-service-related issues of SaaS in cloud-computing environments.

**Keywords:**

Software engineering; software-as-a-service; cloud computing; service-level agreement; quality of service, security

**Article history**: Accepted: March 26, 2020

## 1.Introduction

In the recent years, the proliferation of bring your own device (BYOD), handheld devices, social networks, and related applications has been posing challenges to the service providers, in terms of on-demand network access, ubiquitous connectivity, resource sharing, resource configuration, down time, speed of provisioning and re-provisioning of networks, and most importantly, the amount of human intervention and effort involved. In the era of automatic computation featuring convenient allocation and withdrawal, seamless connectivity with the support of mobility, and reduced capital expenditure (capex) and operational expenditure (opex), enterprises are keener to use services without hosting them on their physical systems.

This phenomenon introduces the concepts of cloud computing and cloud resource sharing [1]. Service providers are searching for robust and scalable on-demand solutions with minimal management responsibilities, interoperability,vendor-agnostic hardware, flexibility, programmability, and secure control over resources. With the incorporation of the cloud model, all these above-mentioned features can be handled with minimal service-provider interaction and centralized control. The cloud model can be viewed as a bouquet of five important features, namely, on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. These service models can be categorized as software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). The cloud can be deployed using any or a combination of four deployment models: private cloud, community cloud, public cloud, and hybrid cloud [2]. This paper explores the business model of SaaS along with traditional software engineering (SE) and service-oriented architecture techniques. The present paper summarizes the legacy techniques and their usability in designing and developing a SaaS model and obtaining the best out of the designed model. SaaS [3] is a software fabrication and distribution model where applications (plug-and-play) are hosted by a service provider. Customers can access these applications through networks without adding any physical load to their network or experiencing security threats (secure access to private confidential data). The customers need not worry about the physical server where the applications are loaded, the processes involved, or the generation of output. They only pay for the services used and achieve the desired results.

Therefore, the new era of SE can be viewed

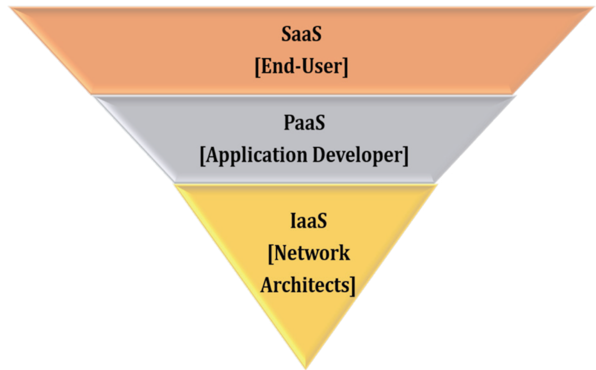


Fig. 1. High-level architecture of Cloud Computing

as service-oriented SE that comprises, integrates, and incorporates the best features of both the service and cloud-computing archetypes to make the development process more advantageous for software development and applications. In light of this, this study discusses an applicable working guideline on the new exemplar in software industries. It also aims to bridge the gap by suggesting substantial best practices, based on analyses of cases of importing SaaS as a running business model [4], [5].

*Major Contribution and Organization of the paper:*

This paper presents a detailed survey of SaaS, considering the traditional SE and service-oriented SE approaches. It analyses the effects of SE processes in developing quality software. Further, it explores several techniques of SE models and reviews to make SaaS a potential business model. The remainder of this paper is organized as follows: Section 2 discusses the basic architecture, benefits, maturity, risks and challenges, and other issues of SaaS. In Section 3, a detailed exploration of the issues associated with the migration from traditional SE to SaaS is presented. Finally, Section 4 concludes the article by discussing the role of SaaS in changing the whole paradigm of software development.

## 2. SaaS in Cloud Computing

Cloud computing is a robust and scalable dynamic platform for computing, where configurable resources are made available as services to users over a standard HTTP medium. The underlying advantage is that it is not necessary for the users to have knowledge, expertise, or control over the technology infrastructure in the "cloud" that supports them. A cloud-computing system can be seen as an integration of three subsystems: compute, storage, and management [6].

• Compute refers to the computing power needed to formulate the complex calculations, which can be achieved by installing high-end complex processors in the hardware to provide computing capability.

• Storage is the most important component of any cloud-computing system. It provides the service of storing large amounts of data in the hardware.

• Management takes care of various types of application programming interface (API) functions and other management capabilities that the system must depict and organize for processing.

The high-level architecture of cloud computing is depicted in figure 1. The provider supplies the solution (software, infrastructure, or platform) on the Internet, and one or more users can consume that service "on demand" by paying for the service. Presently, all major companies have their own cloud-computing frameworks, in which different types of services are available [4]. In the following parts, we briefly discuss the categories of services:

• IaaS: When the cloud offers a client with only infrastructure or the actual hardware (servers and disks), the system is essentially an IaaS.

• PaaS: Platform as a service provides all the hardware and computing infrastructure and maintains the platforms that are installed on the hardware. It can be thought of as an additional layer on top of the IaaS.

• SaaS: This is the topmost layer of the cloud framework. SaaS can be defined as "software deployed as a hosted service and accessed over the Internet." [4]. SaaS makes applications available over the Internet, as services. Instead of paying for the in-house installation and maintenance of software, clients can simply access the software via the Internet, without bothering about the internal complexity of software and/or hardware management [4], [7], [8]. These applications are also known as web-based software, on-demand software, or hosted software. They run on the provider's server and the customers need not install, maintain, or update hardware/ software. All these applications can be accessed using the Internet only. Therefore, a single application can serve multiple customers using the multi-tenant architecture. SaaS is developed with a "one-to-many" model, whereby an application is shared across multiple clients. It has the potential to change the way people view of software and therefore, has an impact on buying, trading, and using software [9]. A solution attracts market and investors only when it proves its fidelity to gain. Enterprises and organizations of all sizes can be categorized into two service-oriented dimensions:

• Line-of-business services are aimed at facilitating business processes such as finances, supply-chain management [4], and customer relations, and they handle large-scale configurable and customizable solutions. They are provided on a subscription basis.

• Consumer-oriented services are offered to the public, sometimes, at low costs and may be even at no cost, with the support of advertisement. Moving from on-premise services to on-cloud services is a huge step, which has broad impacts in three interrelated areas: the business model, application architecture, and operational structure [10].

### 2.1. Benefits of SaaS

By nature, SaaS resides in the Cloud. SaaS sales reached an estimated 10 billion in 2010 and it has been projected to reach 21.3 billion by 2020. If deployed and accessed successfully, SaaS can provide several benefits for both the providers and customers. In the following section, we briefly discuss the benefits of SaaS:

• Risk - Most providers provide a 30-day free trial of the SaaS software's full version. The clients can utilize it for a month before buying and verify whether it fulfils their requirements.

• Commitment - SaaS contracts can be cancelled any time. As a pilot approach, most of the solution providers allow the customers to cancel their contract and pay for only their period of use. Alternatively, it can be hired on a monthly basis.

• Cost - The "pay as you go" approach for SaaS is more attractive than the cost of software licenses, computer hardware, support contracts, and version updates. Thus, SAAS solutions are cost effective [11].

• Deployment - The clients can use the services on demand using SaaS. No prior deployment time, skilled staff, or office setup is required.

• Utilization - SaaS is continuously evolving based on user feedback. Moreover, it is easy to use and less complex, from the end-user point of view. Thus, it is more user-friendly, and is utilized more than the traditional software [12].

• Security - As SaaS is controlled centrally, it is easier to implement security updates, feature enhancement, bug fixing, and reliability solutions in SaaS, compared to the traditional on-premise software environments [13].

• Productivity - With the notion of work from anywhere, anytime, and on any device, SaaS solutions are more productive than the traditional ones.

• Seamless access to the latest and greatest - Ideally, with SaaS, the client does not need to worry about the latest releases and upgrades. The services are always offered with the latest version of software and are being improved continuously.

• Flexibility - The cloud infrastructure can be scaled rapidly to cope with the needs of the customer. In this way, SaaS is a customizable and flexible solution for all types of software requirements. It is a recommended solution for a small firm that wants to expand, or for a firm planning to combine traditional and service-oriented software and reduce capex and opex [14].

## 2.2. Maturity Model of SaaS

There are several ways service providers can host applications in the SaaS-cloud framework [14–16]. The maturity model SaaS framework is illustrated in figure 2.

• Psudo-SaaS: offers an ad-hoc and customized version of the hosted application, and runs its own instance of the application at the servers.

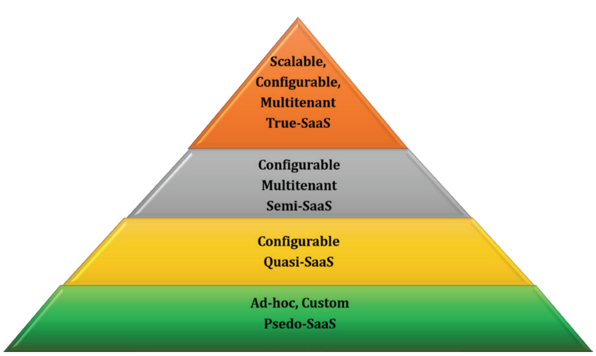


Fig. 2. Maturity model SaaS Framework

• Quasi-SaaS: The provider maintains a separate instance of the application for each customer (or tenant). Each instance of the application is the same at the coding level, but individually customized for each tenant. The customer has the option to specify the appearance and behavior of the application.

• Semi-SaaS: A single instance serves all customers, with the help of configurable metadata that provide a distinct user experience and feature set for each user. Isolation, authorization, and security policies are implemented to guarantee the privacy of data of each client. These sharing architectures are hidden from the end user. This approach allows efficient use of computing resources at low costs. However, the scalability of the application is a crucial issue of this approach.

• True-SaaS: This approach supports multi-tenancy with application scalability, as the provider hosts multiple clients on a load-balanced framework of identical instances, where data isolation is achieved. Furthermore, because of the configurable metadata, each client (tenant) receives a unique user experience and customized feature set.

## 2.3. Issues

Although SaaS provides many benefits to organizations, as it is in its nascent stage, proper planning, evaluation, and screening is essential to make SaaS acceptable to enterprises. The aim of SaaS is to provide small to medium-sized enterprises with enhanced mobility, office productivity, and improve team collaboration by utilizing the expediency of transferring the responsibility of managing software products to a SaaS provider. It is crucial to understand that SaaS needs robust system architecture capable of supporting peak usage demands and secure and seamless processing of large numbers of transactions in an environment [12]. However, some issues need to be considered carefully:

Regulatory and legal risks- In some countries, Government data-privacy laws and regulations make it difficult for a SaaS user and provider to share critical data when the provider is outside the user's country [17].

Downtime risks-If an organization moves many functions to the cloud with SaaS providers, a critical analysis should be done on the risks of increased downtime. To minimize the risk, before trusting any SaaS provider, the organization must verify the history of the provider from the availability and fault-tolerance points of view, for their real-time mission critical data. Alternatively, enterprises may restrict the processing of sensitive data, in house [18].

Data security- Data security is a big concern for public clouds such as iCloud and Dropbox. With any SaaS solution, there is a risk of sensitive-data breach. To minimize this risk, the security measures of the SaaS providers should be implemented with due diligence [19].

## 3. Software Engineering and SaaS

Software engineering [1990] (SE) can be defined as an application of a systematic, disciplined, and quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. Software development, on the other hand, is a planned and structured process of developing a software product. Nevertheless, these terms are often used interchangeably, in literature. To make a transition from traditional SE to SaaS cloud computing is a notion of paradigm shift. It is not a small or low-impact technology review. It calls for changing many aspects of software development and operation [20]. In this section, we explore and scrutinize these challenges toward successful SaaS cloud computing.

• Service and service level: The notable benefit of SaaS is the provision of robust, resilient, and flexible services. In order to realize this, a service catalog describing all business terms must be created. This catalog must be able to control and constantly manage the list of services in the cloud environment, from the providers. As it follows a pay-per-go scheme, the service level should be defined and differentiated, and it must be applicable to all processes such as strategy making, supplier management, service portfolio management, and service catalog management [21].

• Problem management: One of the prevalent differences between the traditional SE and SaaS is the change in organization and management structures. There is a need for holistic visualization and seamless problem management, with readiness to solve customer problems fast.

• Risk and impact: SaaS can be more made appealing with traditional feasibility analysis and risk and impact analysis. In the cloud era, it is necessary to have an automatic resource-release-and-occupy setup, capable of dealing with changes in the requirement.

• Capacity: Before provisioning a service, the SE process must predict the total capacity. Therefore, a good prediction model that can make predictions in terms of SaaS properties is necessary [22].

• Finance: Strategic decision making for pricing and operations support system/ business support system (OSS/BSS) is another big challenge for SaaS. Flexibility in terms of use can be provided at a higher price.

• Supplier handling: It is highly necessary to integrate different IT services according to the defined SLAs to SaaS [23], [24].

• Policy enforcement and supplier management can help select a supplier from potential vendors with healthy competition.

• Service portfolio management: To explore new business opportunities and manage more clients, SAAS uses the service portfolio management process. This process is supposed to be robust enough to react to every possible situation.

• Common operation support: With the promise of low downtime and faster provisioning, SaaS must feature quick software service development. The development process must be agile, clearly defined, and approved in advance. As SaaS claims to lower opex, tool selection is critical so that it can support high-level dynamic environments [25].

• Service validation and testing: With the traditional SE development cycle, a service validation and testing system is needed for prompt provisioning and flexible scale-up.

From the above discussion, it can be seen that traditional SE approaches must be refurbished in order to make SaaS a reality. Versateeg et al. proposed a provisioning algorithm to ensure several enterprise challenges such as licensing, distribution, and configuration, and similar to achieving the main objective of SAAS, maximizing customer satisfaction and minimizing cost. They claimed that their proposed algorithm could reduce the total cost by 48% and the number of SLA violations by 45%, compared to the best of the previously proposed algorithms [19].

In [26], Park et al. presented a study on the issues, risks, and potential of the SaaS-cloud computing environment, and showed the efficacy of such a system with some case studies. Tariq et al. proposed a new method in [27], for the requirement engineering of the SaaS-cloud model and considered a CMMI modification by adding a new requirement element to the existing process. Several requirement categories were identified to make SaaS more user-friendly and easier to develop, during the software-development cycle.

Wu et al. proposed a secure admission control and resource-scheduling algorithm [28] to improve the overall QoS of the SaaS providers. They demonstrated how their algorithm could help the service providers use their critical resources in real time. Using mathematical models and simulations, they demonstrated that the scheme outperformed the reference ones across all ranges of variation in QoS parameters.

Alsarhan et al. proposed an SLA framework for cloud computing SaaS [29], in which, with the help of reinforced learning and adaptive methods, they showed how the QoS for all SaaS clients could be handled efficiently without violating the SLA. With empirical and numerical analyses, they demonstrated the providers' profit under various cloud-environment conditions.

Peng et al. proposed a systematic framework for monitoring, analyzing, and improving the system performance of SaaS environments [30]. In addition, the authors represented the complex resource-allocation process using an elitist archive and the K-means-based mathematical model. Using some empirical case studies, they claimed that their framework efficiently met the requirements of end users (tenants) in a distributed environment.

SLAs are critical for cloud deployment, the adoption of SaaS, as well as for the replacement of the traditional software model. Considering this, in [31], Mubeen et al. investigated the existing research on the management of SLAs in the IoT applications and categorized them into seven main technical classifications: management, definition, modeling, negotiation, monitoring, violation and trustworthiness, and finally, evolution. They surveyed the existing proposals analytically and discussed their shortcomings.

Boukerche et al. developed a task-centric energy-aware cloudlet-based mobile cloud model [32] to address the offloading performance, scalability, security, and availability problems, aiming at increasing the cloudlet processing throughput, reducing the energy cost on the remote cloud, and improving the offloading execution efficiency and energy efficiency on mobile devices. The authors claimed that the proposed energy-aware offloading model could efficiently improve the offloading performance for mobile devices.

In [33], Nandi et al. proposed an intuitive model for the elastic cloud paradigm of dynamic SLA. They designed the model in such a way that it could capture the anticipated license requirement variations of SaaS users. The model used greedy heuristic-based optimization approaches for the service provider to solve the tenant on-boarding problem with dynamic SLA constraints. They demonstrated the efficiency of their solution by improving the performance in terms of the overall resource utilization and economy for both the service provider and tenants.

Wu et al. proposed a solution [34] to solve the traditional enterprise application distribution and configuration-related challenges. They proposed a resource-allocation algorithm to minimize the infrastructure cost to convert the traditional enterprise to SaaS so that the maximum service level could be guaranteed to the end user using dynamic resource sharing and virtual machines in the cloud environment.

Ghosh et al., proposed software defined networking- based paradigm [35],[36 where multiple controllers have been used to provide service level guarantee for software-as-a-service in smart grid networks.

## 4. Security and QoS Issues in SaaS and Cloud Computing

In this section, some state-of-the-art studies security concerns of collabirative SaaS and service-level-agreement (SLA)-based quality-of-service (QoS) maximization for SaaS in the cloud-computing environment have been discussed. The principal concerns for security related to collaborative SaaS Cloud have been depicted in Fig 3.

From the end user point of view, the main challenge is selecting an ideal SaaS cloud provider and secure the collaboration service offered by it to prevent unauthorized disclosure of sensitive information.

There is active research being carried out since the conceptualization of Cloud computing. In [37], Khalil et al., extensively discuss the basics of cloud security in the point of vulnerability, risk models, various


Fig 3: Security Concerns in Collaborative SaaS

attacks and similar security flaws. They categorically discussed the shortcomings of existing security modules. Incorporation of Cloud computing SaaS in any industry and real time solution, quality of service (QoS) plays a vital role.

In [38], Zheng et al., proposed a prediction framework for QoS ranking for cloud services. Using the real-world QoS data from past customers the algorithm directly predicts the QoS rankings of several cloud services. The efficacy of integrating of cloud computing in modern service model and empirically discussed several QoS selection problem by providing a comparison framework to evaluate the QoS and performance of cloud models have been discussed in [39]. To make SaaS a reality, cloud service description (CSD) is an important factor to be considered. The lack of standardization in CSD in several techniques spans from language, standards, ontologies, models, etc. creates a huge misconception in technical, operational, business, and semantic aspects of SAAS. Ghazouani et al., in [40] presented a comparative study of CSD issues from different perspectives. They showed the proposed approach USDL (Unified Service Description Language) has the potential to provide appropriate service description by covering three aspects like technical, operational, and business.

In [41], Lovas et al., proposed a novel software container-based cloud orchestration framework to manage both aspects of cyber physical systems server-side framework for sensor-based networks and configurable simulation tool for predicting the behavior of manufacturing systems as well. Software-Defined Networking (SDN) opened up new paradigm for networking by enabling programmability, scalability, dynamic control and fast reconfiguration.

Several works [42-44] studied extensively how these characteristics can be merged to cloud computing to get more benefits in point of traffic engineering, network virtualization, power optimization, reconfiguration, fast failure, performance and security in multi-cloud environment. SDN can take lead role to redesign the service provisioning model of IaaS and SaaS without changing the physical network at all to provide more flexible and efficient cloud computing along with quality and performance of SaaS services [45].

With the proliferation of cloud computing in several business model and solutions, diverse security and privacy issues are coming out which in-cumber the adoption of this new computing paradigm by common users. To guarantee a secure and trustworthy service level cloud environment, addressing security and privacy challenges become very important. In [46–48], critical security challenges and factors have been identified and analysed regarding embedded system, application, storage system, clustering where SaaS can provide efficient solutions. Both Private and Public cloud security recommendations have been listed [49].

Rath et al., investigated issues like system and data security, latency, QoS, communication security, availability of cloud resources [50]. To strengthen the security and resilience of Cloud SaaS application, a general guideline for developing such applications has been formalized. Several useful case studies have been given for security patterns and solutions in AWS and Azure [51], [52].

## 5. Future Research Directions

In order to make SaaS more popular to Enterprise and IT end users, SDN can play a vital role. SDN controlled wide area network (SD-WAN) can be a powerful choice to get high performance, availability and reliability from their SaaS deployments. Without depending on the data center traffic, using SD-WAN data can be migrated by identifying and offloading internet and cloud traffic in order to achieve consistent, ripple free SaaS reliability, even during link failure. Dynamic and runtime decision can be taken to reroute the information and get higher performance. Network Function Virtualization (NFV) is another important tool for virtualizing important functionalities without changing the physical substrate. NFV can handle the migration of in network computation from dedicated physical hardware substrate to SDN enabled do as needed basis model. Recent trend in wireless communication and Cloud-RAN architecture for radio access networks including future 5G wireless networks can be examined and SDN-NFV can be used for all cloud communication concepts. To provide seamless, fast reliable service to the SaaS end users and increasing deployability of SaaS open issues and future opportunities can be explored.

## 6. Conclusions

From the above discussion and review of recent literature, it can be easily seen that SaaS has the potential to change the whole paradigm of software development and distribution in the near future. This paper reviewed several aspects of the SaaS in the cloud computing paradigm. Some challenges of service-level-agreement (SLA)-based quality-of-service (QoS) maximization for SaaS in the cloud-computing environment have been pointed out and discussed. Security and virtualization are two main aspects to consider making customers more attracted to the services of SaaS. The present paper also conducts a overview study to move forward with these cutting edge research trends and technologies.

## References

[1] Kim, W., "Cloud Computing: Today and Tomorrow," Journal of Object Technology, Vol. 8, No. 1, pp. 65-72, 2009.

[2] Song, Y.; Pang, Y. "How to Manage Cloud Risks Based on the BMIS Model". J. Inf. Process. Syst. 2014, 10, 132-144.

[3] Thomas, D., "Enabling Application Agility - Software as A Service, Cloud Computing and Dynamic Languages", Journal of Object Technology, Vol.7, No. 4, May-June, 2008.

[4] Salesforce "Power Your Business with the Best of the Web" https://www.salesforce.com/SaaS/

[5] Ghosh, U et al., "MMIP: A New Dynamic IP Configuration Scheme with MAC Address Mapping for Mobile Ad hoc networks". National Conference on Communications, IIT Guwahati, January 2009.

[6] Gartner. Gartner Says Cloud Computing Will Be As Influential As E-business, www.gartner.com.

[7] Django: https://www.djangoproject.com/

[8] IBM Cloud: http://www.ibm.com/cloud-computing/in/en/SaaS.html

[9] Banerjee A., "A formal model for multi-tenant software-as-a-service in cloud computing". In Proceedings of the 5th ACM COMPUTE Conference: Intelligent & scalable system technologies (COMPUTE '12). ACM, New York, NY, USA.

[10] Nuno Santos et al., "Towards trusted cloud computing". In Proceedings of the 2009 conference on Hot topics in cloud computing (HotCloud'09). USENIX Association, Berkeley, CA, USA.

[11] T. V. Raman, "Cloud computing and equal access for all," In Proceedings of the 2008 international cross disciplinary conference on Web accessibility (W4A 2008), ACM, April, 2008.

[12] Neill, C.J., and Laplante, P.A. "Requirements Engineering: The State of the Practice", IEEE Software 20(6): 40-45, 2003.

[13] Chang, V et al., "A review of cloud business models and sustainability". In Proceedings of 3rd IEEE International Conference on Cloud Computing, Miami, FL, USA, 5-10 July 2010.

[14] Overview and Issues for Implementation of the Federal Cloud Computing Initiative: Implications for Federal Information Technology Reform Management. Available at http://fas.org/sgp/crs/misc/R42887.pdf.

[15] Jae Yoo Lee, et al., A Quality Model for Evaluating Software-as-a-Service in Cloud Computing, In Proceedings of Seventh ACIS International Conference on Software Engineering Research, Management and Applications, SERA, 2009.

[16] Pang Xiong and Wen, Li Dong. "Quality Model for Evaluating SaaS Service", In Proceedings of Fourth International Conference on Emerging Intelligent Data and Web Technologies, 2013.

[17] Alexander Benlian, Thomas Hess, "Opportunities and risks of software-as-a-service: Findings from a survey of IT executives," Decision Support Systems, 52 (1), December 2011, pp. 232-246.

[18] Yvonne Coady et al., "Distributed Cloud Computing: Applications, Status Quo, and Challenges." SIGCOMM Comput. Commun. Rev. 45, 2, April 2015, pp.38-43.

[19] S. Versteeg, S. Kumar Garg, L. Wu and R. Buyya, "SLA-Based Resource Provisioning for Hosted Software-as-a-Service Applications in Cloud Computing Environments", IEEE Transactions on Services Computing, vol.7, no. 3, pp. 465-485, 2014.

[20] Malkawi, M.I. The art of software systems development: Reliability, Availabili-

ty, Maintainability, Performance (RAMP). Human-Centric Comput. Inf. Sci. 2013.

[21] Chatterjee, P. et al., A trust enhanced secure clustering framework for wireless ad hoc networks. Wireless Networks, 20(7), pp.1669–1684.

[22] Chi-Yao Hong ey al., Achieving high utilization with software-driven WAN. In Proceedings of the ACM SIGCOMM 2013, ACM, New York, NY, USA, 5-26.

[23] The cloud for modern business: http://azure.microsoft.com/en-in/.

[24] Improving human productivity through SaaS: http://www.SaaS.com/ta/hp.jsp

[25] Whyudin, D. M. et al., In-time project status notification for all team members in global software development as part of their work environments. In: Proceeding of SOFPIT Workshop 2007, SOFPIT/ICGSE, Munich, pp. 20-25.

[26] Park S., Lee S., Park Y.B. (2015) Best Practices in Software Engineering for SaaS-Cloud Era. In: Park J., Stojmenovic I., Jeong H., Yi G. (eds) Computer Science and its Applications. Lecture Notes in Electrical Engineering, vol 330. Springer, Berlin, Heidelberg.

[27] A. Tariq, S. A. Khan and S. Iftikhar, "Requirements Engineering process for Software-as-a-Service (SaaS) cloud environment," 2014 International Conference on Emerging Technologies (ICET), Islamabad, 2014, pp. 13-18.

[28] Linlin Wu, Saurabh Kumar Garg, Rajkumar Buyya, SLA-based admission control for a Software-as-a-Service provider in Cloud computing environments, Journal of Computer and System Sciences, Volume 78, Issue 5,2012, pp. 1280-1299.

[29] Ayoub Alsarhan, Awni Itradat, Ahmed Y. Al-Dubai, Albert Y. Zomaya, Geyong Min, "Adaptive Resource Allocation and Provisioning in Multi-Service Cloud Environments", Parallel and Distributed Systems IEEE Transactions on, vol. 29, no. 1, pp. 31-42, 2018.

[30] Gongzhuang Peng, Hongwei Wang, Jietao Dong, Heming Zhang, "Knowledge-Based Resource Allocation for Collaborative Simulation Development in a Multi-Tenant Cloud Computing Environment", Services Computing IEEE Transactions on, vol. 11, no. 2, pp. 306-317, 2018.

[31] Saad Mubeen, Sara Abbaspour Asadollah, Alessandro Vittorio Papadopoulos, Mohammad Ashjaei, Hongyu Pei-Breivold, Moris Behnam, "Management of Service Level Agreements for Cloud Services in IoT: A Systematic Mapping Study", Access IEEE, vol. 6, pp. 30184-30207, 2018.

[32] Azzedine Boukerche, Shichao Guan, Robson Eduardo De Grande, "A Task-Cen-

tric Mobile Cloud-Based System to Enable Energy-Aware Efficient Offloading", Sustainable Computing IEEE Transactions on, vol. 3, no. 4, pp. 248-261, 2018.

[33] B. B. Nandi, A. Banerjee, S. C. Ghosh and N. Banerjee, "Dynamic SLA based elastic cloud service management: A SaaS perspective," 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), Ghent, 2013, pp. 60-67.

[34] L. Wu, S. K. Garg and R. Buyya, "SLA-Based Resource Allocation for Software as a Service Provider (SaaS) in Cloud Computing Environments," 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Newport Beach, CA, 2011, pp. 195-204.

[35] U. Ghosh, P. Chatterjee and S. Shetty, "A Security Framework for SDN-Enabled Smart Power Grids," 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), Atlanta, GA, 2017, pp. 113-118.

[36] U. Ghosh, P. Chatterjee, D. Tosh, S. Shetty, K. Xiong and C. Kamhoua, "An SDN Based Framework for Guaranteeing Security and Performance in Information-Centric Cloud Networks," 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), Honolulu, CA, 2017, pp. 749-752.

[37] Khalil, I.M.; Khreishah, A.; Azeem, M. "Cloud Computing Security: A Survey", Computers 2014, 3, 1-35.

[38] Z. Zheng, X. Wu, Y. Zhang, M. R. Lyu and J. Wang, "QoS Ranking Prediction for Cloud Services," in IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, pp. 1213-1222, June 2013.

[39] Nitin Upadhyay, "Managing Cloud Service Evaluation and Selection", Procedia Computer Science, Volume 122, 2017, Pages 1061-1068.

[40] Souad Ghazouani and Yahya Slimani. 2017. "A survey on cloud service description". J. Netw. Comput. Appl. 91, C (August 2017), 61-74.

[41] R´obert Lovas, Attila Farkas, Attila Csaba Marosi, et al., "Orchestrated Platform for Cyber-Physical Systems," Complexity, vol. 2018, Article ID 8281079, 16 pages, 2018.

[42] [Jungmin Son and Rajkumar Buyya. 2018. A Taxonomy of Software-Defined Networking (SDN)-Enabled Cloud Computing. ACM Comput. Surv. 51, 3, Article 59 (May 2018), pp.1-36.

[43] R. Jain and S. Paul, "Network virtualization and software defined networking for cloud computing: a survey," in IEEE Communications Magazine, vol.51, no. 11, pp. 24-31, November 2013.

[44] W. Xia, Y. Wen, C. H. Foh, D. Niyato and H. Xie, "A Survey on Software-De-

fined Networking," in IEEE Communications Surveys & Tutorials, vol.17, no. 1, pp. 27-51, 2015.

[45] Haolong Fan, Farookh Khadeer Hussain, Muhammad Younas, Omar Khadeer Hussain, "An integrated personalization framework for SaaS-based cloud services," Future Generation Computer Systems, v.53 n.C, p.157-173, December 2015.

[46] Yuhong Liu, Yan Lindsay Sun, Jungwoo Ryoo, Syed Rizvi, Athanasios V. Vasilakos, "A survey of security and privacy challenges in cloud computing: solutions and future directions", J. Comput. Sci. Eng., 9 (2015) 119-133.

[47] Arshad, J., Townend, P., Xu, J., & Jie, W. (2012). "Cloud Computing Security: Opportunities and Pitfalls." International Journal of Grid and High Performance Computing (IJGHPC), 4(1), 52-66.

[48] Saurabh Singh, Young-Sik Jeong, and Jong Hyuk Park. 2016. "A survey on cloud computing security". J. Netw. Comput. Appl. 75, C (November 2016), 200-222.

[49] Hari Krishna, S. Kiran, G. Murali, R. Pradeep Kumar Reddy, "Security Issues in Service Model of Cloud Computing Environment", Procedia Computer Science, Volume 87,2016, Pages 246-251.

[50] Rath, A.; Spasic, B.; Boucart, N.; Thiran, P. "Security Pattern for Cloud SaaS: From System and Data Security to Privacy Case Study in AWS and Azure", Computers 2019, 8, 34.

[51] Joanne Martin and Koos Lodewijkx. 2014. "Securing Cloud Environments for Enterprise Computing". In Proceedings of the 6th edition of the ACM Workshop on Cloud Computing Security (CCSW '14). Association for Computing Machinery, New York, NY, USA.

[52] Li, C., Li, L.Y. "Optimal resource provisioning for cloud computing environment". J Supercomput 62, 989–1022 (2012).