

# Software as a Service

Nitu Gupta

Department of Computer Engineering and IT  
Veermata Jijabai Technological Institute  
Matunga, Mumbai, India

Varshapriya JN

Department of Computer Engineering and IT  
Veermata Jijabai Technological Institute  
Matunga, Mumbai, India

---

**Abstract**— Many small and medium scale industries could not afford the expensive enterprise software either on premise or through Application Service Providers as it incurred heavy installations cost, maintenance cost. Mean while the development of social, economic, increasing Internet technology lead to the evolvement of Cloud computing Technology which allowed the users to share the resources across the Internet. This development brought about a radical change in the computing technology posing new challenges as the requirements of the users kept on changing frequently and such an installed Application Service Providers on the premise of the user made it difficult to meet these changed user requirement. Thus the vendors developed a new architectural style based upon the constructional forms and changing the way the functionalities are delivered to the user. This shift in the way of delivery of software created a new business market. In this the user is freed from the need of installation of the software thus shifting the maintenance cost to the vendor. This type of business model is termed as Software as a service which is one-to-many model based on “on use pay” (subscription) basis. SaaS not much different from traditional ASP has some SaaS specific functions added to it like registry, billing etc. This paper reviews th is business model comparing it with the traditional ASP application and the Service Oriented Architecture. It describe the life cycle of SaaS by covering its maturity model and its architecture. It also highlights the advantages of SaaS along with some of the characteristics required / desired in any SaaS application.

**Keywords**— SaaS, ASP, Cloud Computing, Multitenancy, Configurability, Tenant, SOA

---

## I. INTRODUCTION

In traditional hosted service model software vendors started to use the Internet to deliver software applications to end users. These Application Service Providers (ASPs) basically allowed their customers to download software applications and the installation process was similar to the traditional CD-based software. Application Service Provider Abbreviated as ASP, a third-party entity that manages and distributes software-based services and solutions to customers across a wide area network from a central data center. Even though the ASP service model provided a relatively efficient means of getting software applications to the customer, it suffered a number of problems. The main problem that the users suffered was the size of the actual download file, which could run into hundreds of MB; and the subsequent maintenance updates which also had to be downloaded on a regular basis. Without these updates the software vendors could not always ensure that their customers were running the latest version of their software. The rapidly increasing Internet Technology and the economical and social development in the computing environment triggered to bring about big changes in the software business. The software providing vendors faced new challenges and opportunities with the emergence of Web and its infrastructure. Intuitively the development in the IT Technology and business environment lead to the evolvement of Cloud computing that allowed the Shared IT resources to be accessed over the internet as services rather than products to the end users. In 1999 the Pennie Group a consortium of Software Engineering Researchers from the University of Durham Keele University and UMIST had put forward the view that the future of the software lies in in not just developing a new architectural style based upon the constructional forms but changing the way the functionalities are delivered to the users. This paradigm shift in software delivery lead to rise of new software business that is termed as the Software as a Service (SaaS).

The term Software as a Service in camel back acronym is represented as SaaS, is a software business model. It has driven a new implementation technology since it was not offered as a product but as a service and internet being its backbone. SaaS was evolved with a target of capturing the market of small customers who could not afford the expensive enterprise software either on premise or through ASP. It has brought a radical change in the existing method of selling an installing the software thus posing a major challenge to the conventional licensed software model. SaaS has already received a widespread approval. It gained the momentum after 2005-2006 when the internet became high speed and affordable and also the customers became comfortable with doing the business on web. Today almost every global company is on the verge of providing SaaS services to the users thereby creating a new business market.

This paper describes this service in detail. The paper is organized as follows: Section II gives a brief explanation of SaaS and how SaaS differs from Application Service Provider and Service Oriented Architecture. It gives a brief description of Cloud Computing and Application Service Provider Section III describes the maturity models of SaaS. Its Architecture is briefly reviewed is in Section IV. The Section V focuses about the Characteristics of SaaS and the advantages of SaaS. The Section VI concludes the paper.

## II. SAAS, ASP, SOA

### A. Cloud Computing

Cloud computing a recent trend in IT involves large number of computers connected through a network such as internet and deals with computation, software data access and storage services. It has moved the data and the computing resources from desktop and PCs to large data centres with a main goal of making better use of the distributed resources. NIST defined Cloud computing as “Cloud computing is a model for enabling 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.” It deals with the virtualization, scalability, interoperability, delivery models of cloud namely private, public, and hybrid. The key advantages of using the cloud are scalability, reliability, availability and cost reduction with better utilization [1][2]. The different layers of cloud computing are depicted in the Fig 1. below. Each of the layer below represents the type of service that are provided by the cloud provider. Out of all the layers shown Application layer i.e. software as service is discussed in the paper.

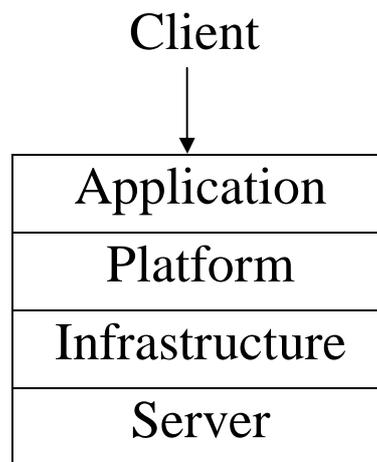


Fig. 1 Different layers of Cloud Computing

### B. Software as a Service(SaaS)

SaaS is a software delivery paradigm where software is hosted off premise and delivered via web to a large number of tenants via the subscription model. It is a newly emerged one to many business delivery model in which both the application and host environment are centrally managed and shared with several companies who lease and use application services. Gartner defined SaaS as software that is owned, delivered and managed remotely by one or more providers. In simple words it is also called as the reformed ASP where the users access the application using thin client via a web browser. SaaS are equivalent to the outsourcing vendors who handle all the needed maintenance, customization, and updating and charging for the services their client uses on monthly or yearly basis. The provider deliver the software based on one set of common code and data definitions along with the different configurations for the different types of tenant stored in the configuration descriptor. Multiple tenants (Users who subscribe for the SaaS) access the application with a feeling that they are the sole users of the application, however configuration setting for different users like the changes in the data, work flow, UI and business flow take place in the background. SaaS has enabled the shift from traditional waterfall like development model consisting of single phrase sequential steps to a more agile methodology consisting of multiple iterations within each phrase [1][3][5].

### C. Application Service Provider (ASP)

Application Service Provider abbreviated as ASP, a third-party entity that manages and distributes software-based services and solutions to customers across a wide area network from a central data center. In essence, ASPs are a way for companies to outsource some or almost all aspects of their information technology needs. They may be commercial ventures that cater to customers, or not-for-profit or government organizations, providing service and support to end users. The application software resides on the vendor's system and is accessed by users through a web browser using HTML or by special purpose client software provided by the vendor. Custom client software can also interface to these systems through XML APIs. These APIs can also be used where integration with in-house systems is required.

### D. Difference between SaaS and ASP

SaaS and standard web based software are not very different from each other with the SaaS application being provided with additional of SaaS specific functions like registering, billing, support etc. The difference between ASP and SaaS in on different delivering modes or application deployment methods, Hardware compliance, application design, development focus [4][7][8]. The difference between them is illustrated in the table1. below.

TABLE I  
 DIFFERENCE BETWEEN SAAS AND ASP.

Feature	Software as a Service	Application Service Provider
Delivery Method	Applications present at the SaaS Provider and central server and user delivered the services via Internet	User provided with applications installed on to their premises.
Hardware compliances	Not much issues of hardware as the entire maintenance is taken care by the SaaS Vendors.	Difficult since it contain a software that was not written to be hosted on web
Application Design	They are designed for Multiple customers having basic requirement more and less similar with some unique distinct feature that are taken care of by the configuration and metadata.	They are customers specific i.e. for each tenant a separate ASP has to be provided since they are installed on the premises.
Development Focus	This along with the application development also takes care of all the task that make up SaaS a business model.	Entire focus is on the application development itself
Pricing Mode	The users follow a subscription method where they are billed depending upon their usage and also provided with the options to use the services that they need	The user are needed to pay at the start when the software is installed or purchased at their premises. They also needed to pay for the maintenance of the software. Thus the total cost incurred is more.

*E. SaaS and Service Oriented Architecture*

SOA is a flexible set of design principles used during the phases of system development and integration. Both SaaS and SOA are beneficial trends in IT as both deliver the software in a specific way thereby giving a agile alignment between business needs and IT solutions and reducing the cost of software delivery. The difference in SaaS and SOA lies in designing and the manner in which the benefits are delivered. SOA is a means of designing and building a software thereby it's a manufacturing model. However SaaS is the means of receiving the model and hence it's a sales and distribution model. Likewise SOA provides a means to deploy and quickly reconfigure as the business changes. On the other hand SaaS provides a readily accessible means to outsource the application and database of a business [4][7][8].

**III. MATURITY MODEL AND ARCHITECTURE**

A well designed SaaS application should be scalable, multitenant efficient and configurable. The different maturity level of SaaS are as follows: [3][7][9]

*A. Level 1 Maturity I (Ad Hoc/Custom)*

The SaaS provided with this maturity level is much similar to traditional Application Service Provider. At this maturity level the software vendor concentrates on dedicated database and schema in the data layer thereby creating a separate instance of hosted application for each customer. This model of SaaS does not support content sharing and multitenancy and is quite easy for the transition of traditional ASP to this maturity level SaaS mode. System layer of this model consist of Ad hoc mutli instances which uses different applications by users.

*B. Level 2 Maturity II (Configurable) :*

At this level the software vendor hosts a separate instance of the hosted application for each customer but all the tenants share the same code implementations and each instance is configured with different GUIs and different characteristics. The instances however remain isolated from each other. In this phase the customers uses the shared

public database but with dedicated data schema. The system layer support configurable single instance and single tenant so that users built their service model with in predefined instance that is given b y service provider.

**C. Level 3 Maturity III (Configurable, Multi Tenant-Efficient) :**

This level focuses on multitenancy along with all the features of the above two levels. In other words database schema is shared as well as the database itself in the data layer to accommodate mutli-tenant simultaneously. In simple words at this level all the tenants share the single instance of the application and the metadata is configured to fulfil the requirements of different tenants. The authorization and security policies of each tenants are kept separately. However this level does not support scalability.

**D. Level 4 Maturity IV (Scalable, Configurable, Multi tenant Efficient):**

This is the most mature level of the SaaS software where the one feature missing in the third maturity level is also included. This level provides a high degree support for scalability. This scalability is achieved through the virtualization concept of cloud computing by which this model can increase and decrease the number of servers and instances to support arbitrary number of tenants. The level IV is the most complicated SaaS model and may have higher cost of development.

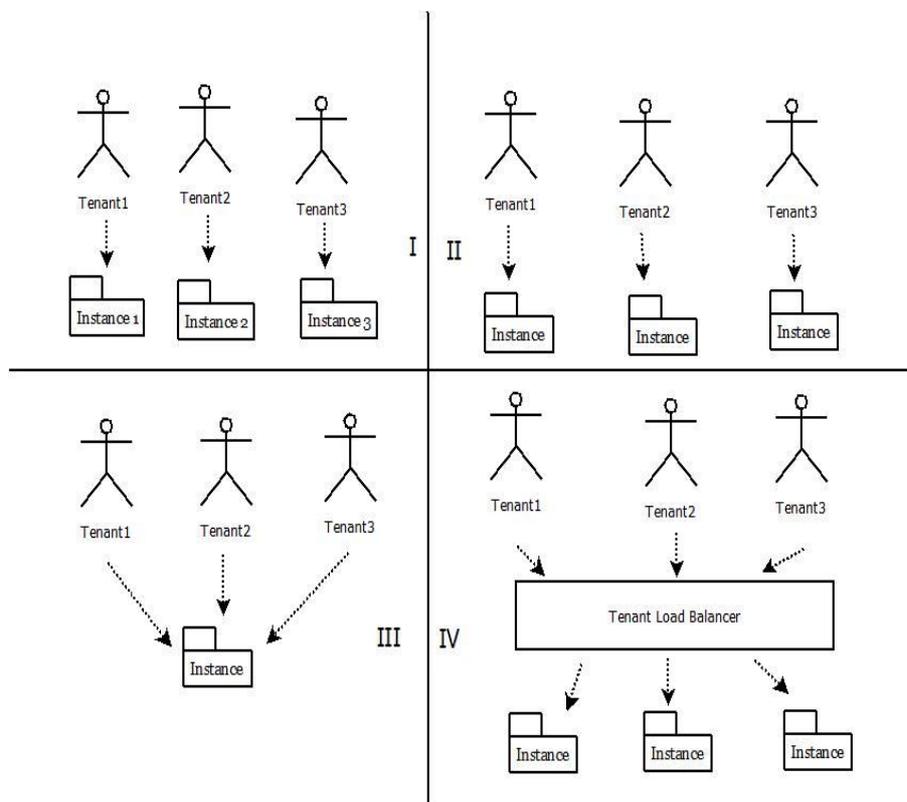


Fig. 2. SaaS Maturity Model

**IV. SAAS ARCHITECTURE**

The SaaS Application are built according the agreement of the SaaS provider and delivered to the user via internet without the need of requiring installing the software. The architecture of the SaaS is described by the Fig 3[10].

**A. Service Transport Layer:**

This layer interacts directly with the User. SOAP, Web Service and XML and relational network security protocols are adopted in order to ensure the accuracy of information analysis and delivery. These protocols ensure the exchange of structured data information in the web application.

**B. Service Wrapped and Schedule Layer :**

This layer is basically responsible for load balancing required for scalability. It also is responsible for customizing the application using the configuration data to meet users' personalized demand according to the user's business processes .And all users UI and application data are isolated through the authorization and security policy, etc .

**C. Service Technology Layer:**

SaaS applications provided by software vendors or ISVs are closely linked through service description, service discovery, service negotiation, service delivery, service composition, and so on.

*D. Applications and Services Layer:*

Not alike deployment of ASP software and services, SaaS software can be installed on anywhere, including service providers, ISVs, or the server located on the Internet. These software and services are linked through the appointment APIs or special Web Services.

*E. Data and Services Management Layer:*

Multitenancy is provided by Shared database and isolation data structure to allow all users to access their private datum in same database or database cluster with a different data set, so that each user can design different data models. This layer takes care of all the data base models to give the user sole application user feel. There can be three types of data models that can be used to support multitenancy- Isolated for each tenant, Shared schema for tenants and shared database and schema for multiple tenants. The configuration data can be stored in the XML file or in a relational database. The configuration data is important to support multitenancy since it enable to provide a personalized feel of the application to each of the user such that it feels that it is the only current user of the application although some other users might be using the application with their personalization. If the configuration data is stored in XML format then it provides extensibility but however such XML format supporting hierarchical data model does not provide rollbacks or any built in mechanism to control the access to the data. However the configuration data related to User Interface of the application can be stored in the XML format but the application data is always stored in relational data model since it provides integrity during concurrent access [3][8].

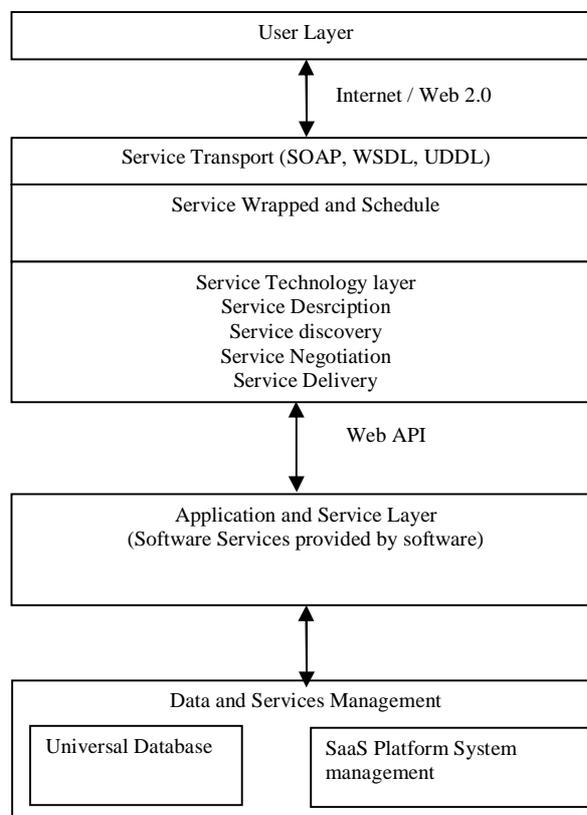


Fig. 3. SaaS Application Architecture.

**V. CHARACTERISTICS AND ADVANTAGES OF SAAS**

**A. Characteristics of SaaS Applications**

The SaaS application that supports the characteristics like custom built, scalability and multitenancy can be called to be as a well designed application. Not all SaaS application share all the characteristics but those commonly found in any SaaS application are as follows[1][3][4][5]

- *Configurability:*

Applications with this characteristic have same code base and providing the tenants with multitude of options such that each tenant can have unique software configuration and can have the feel of a sole user of the application. This is key stone to the success of SaaS application and comes in existence in all those applications which are of level II or beyond. For example every tenant might a get a homepage with their own logo and the background they specified. This can also be called as the customization of the application.

- *Multitenancy:*

With these characteristics the single application instances serve all the customers. The different data models that are available to support the multitenancy in SaaS are isolated databases, shared database isolated data construction, shared databases and data construction. Utilizing multitenant architecture SaaS vendors can innovate more quickly and save valuable time spent on maintaining numerous versions of outdated code. This characteristic is included in all the applications of level III or level IV.

- *Scalability:*

This characteristic is most difficult to add into the SaaS application and the most expensive one. The scalability is supported by virtualization but however taking into consideration the cost and complexity issues many a times application developer don't focus on this characteristic much.

*B. Advantages of SaaS:*

The main advantages of SaaS Applications are listed below [4][6][11] :

- *Cost Saving:* SaaS applications can be viewed by the users using thin clients via web browser. This saves the operational cost of the users along with the need of minimum hardware requirement thereby reducing the cost of hardware also. Likewise the maintenance cost, cost of purchasing the software are also minimized.
- *Better Resource Utilization:* Since the resources needed for the SaaS application are minimum the unused resources of the user can be used by him for the other purpose leading to a better utilization of them
- *Global Outsourcing Possibility:* The advancement of Web technology allows SaaS vendors to be located overseas and also offer the high quality services. The offshore outsourcing model allows SaaS users to save more IT expenses.
- *Scalability and Integration:* Usually, SaaS solutions reside in cloud environments that are scalable and have integration with other SaaS offerings. Comparing with the traditional model, users do not have to buy another server or software. They only need to enable a new SaaS offering and, in terms of server capacity planning, the SaaS provider will own that.

## VI. CONCLUSIONS

SaaS is a new way of delivering the application over internet. In the SaaS model, customers need not spend a lot of investment for hardware, software customization and maintenance engineers and do not have to buy software, but switch to lease those Web-based software functions and pay only certain fees for these leasing services. SaaS is defined not only by its delivery through internet but by the subscription and periodic payment. Based on the analysis of the ASP application framework, the benefits of SaaS almost every industry today are moving towards SaaS Application. SaaS has become one of the biggest technology trends to affect the business application in the recent years.

## REFERENCES

- [1] Yashpalsinh, Jadeja,kirit Modi, "Cloud Computing - Concepts, Architecture and Challenges", 2012 International Conference on Computing, Electronics and Electrical Technologies.
- [2] Sivadon Chaisiri, Bu-Sung Lee,Member, , and Dusit Niyato, Member, Optimization of Resource Provisioning Cost in Cloud Computing, IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 5, NO. 2, APRIL-JUNE 2012
- [3] Nitu, "Configurability in SaaS Applications", International Conference ACM 2009.
- [4] Qian Li, Shijun Liu, Ying Pan, "A Cooperative Construction Approach for SaaS Applications, Proceedings of 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design.
- [5] Jihyun,Sungju Kang, Ji Hur,"Web based development Framework for Customizing Java-based Business Logic of SaaS Application", ICACT 2012.
- [6] White paper Software as a Service Business models
- [7] Guoling Liu, He Jiang, Runian Geng, "Software Design on SaaS Platform",2010 IEEE International Conference on Computing.
- [8] Hong He, " Application Deployment on SaaS Platform", 2010 IEEE International Conference on Computing.
- [9] Weiping Li,Zhichao Zhang, Si Wu, Zhonghai wu, "An implementation of SaaS Level-3 Maturity Model for an Educational Credit Bank Information System", 2010 IEEE International Conference on Service Science.
- [10] Hancheng LIAO, "Design of SaaS -basedSoftware Architecture" , 2009 International Conference on New Trends in Information and Service Science.
- [11] White paper on Software as a service as an outsourcing Model.