

## Unit-4

### Software as a Service | SaaS

Software as a Service (SaaS) is a software deployment and licensing model in which software is running on the cloud and is usually accessible for users through a web interface. National Institute of Standards and Technology of USA defines SaaS as cloud service model in which service consumer uses providers applications that run on a cloud infrastructure . It is one of the three main principles of cloud computing, alongside Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). One of the first and best-known companies that provide enterprise SaaS solutions is Salesforce [2]. The model of the company is Business-to-Business (B2B), it was founded in 1999 and specialized in Customer Relations Management (CRM) in the cloud. The solution was horizontal and targeted general business processes.

SaaS is also known as "**On-Demand Software**". It is a software distribution model in which services are hosted by a cloud service provider. These services are available to end-users over the internet so, the end-users do not need to install any software on their devices to access these services.

There are the following services provided by SaaS providers -

**Business Services** - SaaS Provider provides various business services to start-up the business. The SaaS business services include **ERP** (Enterprise Resource Planning), **CRM** (Customer Relationship Management), **billing**, and **sales**.

**Document Management** - SaaS document management is a software application offered by a third party (SaaS providers) to create, manage, and track electronic documents.

**Example:** Slack, Samepage, Box, and Zoho Forms.

**Social Networks** - As we all know, social networking sites are used by the general public, so social networking service providers use SaaS for their convenience and handle the general public's information.

**Mail Services** - To handle the unpredictable number of users and load on e-mail services, many e-mail providers offering their services using SaaS.



### Characteristics

Here are the characteristics of SaaS service model:

- SaaS makes the software available over the Internet.

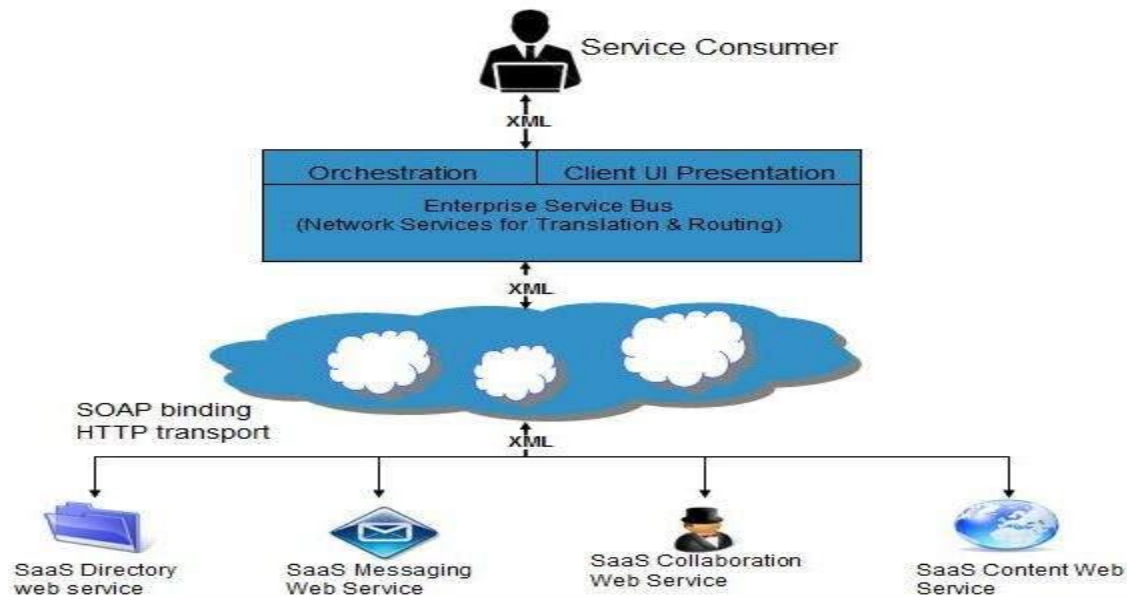
- The software applications are maintained by the vendor.
- The license to the software may be subscription based or usage based. And it is billed on recurring basis.
- SaaS applications are cost-effective since they do not require any maintenance at end user side.
- They are available on demand.
- They can be scaled up or down on demand.
- They are automatically upgraded and updated.
- SaaS offers shared data model. Therefore, multiple users can share single instance of infrastructure. It is not required to hard code the functionality for individual users.
- All users run the same version of the software.

## Open SaaS and SOA

**Open SaaS** uses those SaaS applications, which are developed using open source programming language. These SaaS applications can run on any open source operating system and database. Open SaaS has several benefits listed below:

- No License Required
- Low Deployment Cost
- Less Vendor Lock-in
- More portable applications
- More Robust Solution

The following diagram shows the SaaS implementation based on SOA:



## Advantages of SaaS cloud computing layer

### 1) SaaS is easy to buy

SaaS pricing is based on a monthly fee or annual fee subscription, so it allows organizations to access business functionality at a low cost, which is less than licensed applications.

Unlike traditional software, which is sold as a licensed based with an up-front cost (and often an optional ongoing support fee), SaaS providers are generally pricing the applications using a subscription fee, most commonly a monthly or annually fee.

## **2. One to Many**

SaaS services are offered as a one-to-many model means a single instance of the application is shared by multiple users.

## **3. Less hardware required for SaaS**

The software is hosted remotely, so organizations do not need to invest in additional hardware.

## **4. Low maintenance required for SaaS**

Software as a service removes the need for installation, set-up, and daily maintenance for the organizations. The initial set-up cost for SaaS is typically less than the enterprise software. SaaS vendors are pricing their applications based on some usage parameters, such as a number of users using the application. So SaaS does easy to monitor and automatic updates.

## **5. No special software or hardware versions required**

All users will have the same version of the software and typically access it through the web browser. SaaS reduces IT support costs by outsourcing hardware and software maintenance and support to the IaaS provider.

## **6. Multidevice support**

SaaS services can be accessed from any device such as desktops, laptops, tablets, phones, and thin clients.

## **7. API Integration**

SaaS services easily integrate with other software or services through standard APIs.

## **8. No client-side installation**

SaaS services are accessed directly from the service provider using the internet connection, so do not need to require any software installation.

# Disadvantages of SaaS cloud computing layer

## **1) Security**

Actually, data is stored in the cloud, so security may be an issue for some users. However, cloud computing is not more secure than in-house deployment.

## **2) Latency issue**

Since data and applications are stored in the cloud at a variable distance from the end-user, there is a possibility that there may be greater latency when interacting with the application compared to local deployment. Therefore, the SaaS model is not suitable for applications whose demand response time is in milliseconds.

## **3) Total Dependency on Internet**

Without an internet connection, most SaaS applications are not usable.

## **4) Switching between SaaS vendors is difficult**

Switching SaaS vendors involves the difficult and slow task of transferring the very large data files over the internet and then converting and importing them into another SaaS also.

## Popular SaaS Providers



The below table shows some popular SaaS providers and services that are provided by them –

Provider	Services
Salseforce.com	On-demand CRM solutions
Microsoft Office 365	Online office suite
Google Apps	Gmail, Google Calendar, Docs, and sites
NetSuite	ERP, accounting, order management, CRM, Professionals Services Automation (PSA), and e-commerce applications.
GoToMeeting	Online meeting and video-conferencing software
Constant Contact	E-mail marketing, online survey, and event marketing
Oracle CRM	CRM applications
Workday, Inc	Human capital management, payroll, and financial management.

## SaaS as a Software delivery model

SaaS is a software delivery and licensing model, according to which software provider stores and maintains software on the cloud, and software consumer (end user) accesses it through the web interface. Software delivery (or software deployment) is a process of transporting software to a user. The result of software deployment is ready-touse software on the user side. Software deployment can also be defined as processes between the acquisition and the execution of software . Even though SaaS is considered a delivery model, there is no fixed list of delivery models. The motivation for it in literature is that deployment process is a complex process that is unique for each product. The following stages of deployment lifecycle are identified: Release, Installation, Activation, Updating, Undeployment

(deinstallation) . Table 1 shows how those stages are handled in SaaS and on-premise models. Table 1. Deployment stages description in SaaS .

Deployment Stage	On-Premise Description	SaaS Description
<b>Release</b>	The software is packaged with information and artifacts which are sufficient for it to be installed	The software is uploaded to the server and is accessible through a web interface
<b>Installation</b>	All dependencies are pre-installed if packaged with software, files are copied to user's machine, registry records are written, environment variables are set.	The software is accessed the same way as a regular website
<b>Activation</b>	The activation usually requires an internet connection and consists of either signing into your account on the platform the software is connected to or entering a serial key	Activation consists of choosing a subscription plan. Usually, there is a free plan with certain limitations enabled by default
<b>Updating</b>	Depending on software and	The software is updated on the

	Operating System (OS), software is either automatically updated through the internet or reinstalled	server, the user always receives the latest version when he accesses the web client
<b>Undeployment</b>	Everything that was done during installation is being reverted	No action required except canceling the subscription

### Scalability

Scalability is an ability of a program to scale. A highly scalable program performs similarly on both small and big data sets. The term scalability is ambiguous in the context of SaaS and cloud computing. It may refer to scalability as a SaaS success factor , hardware scalability, data (database) scalability , architectural property (as it was in the literature review), and scalability in multi-tenancy . A SaaS application is an alternative to multiple instances of on-premise applications. Multiple instances of one application are independent of each other, which means the number of users (number of instances) does not affect them in any way. To compete with on-premise applications on the market, SaaS application also has to perform the same regardless of the number of users, so it has to scale effectively, which means scalability is an essential property of a SaaS application. Two main hardware scaling approaches are horizontal and vertical scaling, which can also be referred to as scaling out and scaling up respectively . Vertical scaling, in this case, means moving an application to a server with more computational power, horizontal scaling means adding more servers. The vertical approach usually does not require any changes in the software. The horizontal approach may not be suitable for some architectures. For example, suppose we have an application which works with high readwrite intensity data. If the data is stored in RAM and is dumped to the database on termination to achieve high

performance, such application would not be horizontally scalable, because after user writes data, only the server that was serving that client would have it and other servers would not. At the same time, vertical scalability is applicable for that case. Data or database scalability describes how database behavior changes with the increase of data amount. In SaaS application, one database can serve thousands of customers . In that case query execution typically slows down. There are various methods to improve database performance, for example, indexing and partitioning.

### **Multi-Tenancy**

Multi-Tenancy is a software architectural property which indicates that some resources in the application are shared among multiple tenants . Tenants, in that case, are users or groups of users who have control over the portion of resources provided by the application. Multi-tenancy can be achieved by either several instances or one instance of the application. Following these patterns, each tenant can have his own database or a share of a common database. Multi-tenancy is tightly interconnected with the other two architectural properties - scalability and customization. The key challenges in multi-tenancy architecture are to provide data isolation, cost-efficiency, scalability and customization . Isolation indicates that the data owned by one tenant cannot be accessed by another tenant and one tenant's customizations do not affect other tenants . Customization in the context of multi-tenancy mostly refers to customization of the data schemas which can be stored for each user, more specifically database schemas . Maturity models which were discussed earlier suggest that mature SaaS applications have some kind of a shared data schema, which leads to providing customization being a challenge. Scalability in multi-tenancy refers both to individual tenants being able to scale their applications if needed and to scale the whole system with the increasing number of tenants. For example, SaaS services usually have free and paid plans. Free plans offer fewer resources compared to paid plans. If a user moves to a paid plan from a free one, the application must grant him access to the higher amount of resources

### **Customization**

Customization as an architectural property in a SaaS application is a functionality that provides the user with an ability to make changes in certain parts of software through a specially designed interface . Customization is aimed at addressing specific needs of each customer. The ad-hoc software can be created separately for each customer based on the customer's individual needs (this is relevant for corporate customers). To compete with this model SaaS applications also need a method to address those needs. Customization can be achieved in the following steps: 1. Identifying the application areas which may change according to customer's needs 2. Build an architecture that allows different options for these areas 3. Provide an interface for the user to manage the options for those areas For example, PaaS Heroku has add-ons. This service allows users to choose an add-on that wraps some external service from a list of available add-ons and integrate it into the application. The service provider is responsible for integrating, managing add-ons, the user only chooses them through GUI. Add-ons include DaaS, monitoring and logging utilities, rendering utilities and more. An alternative to that model would be an on-premise server or IaaS, where the user can choose any external services, but he has to install, integrate, manage and update them manually (if they are not also provided via SaaS model). In [36] 6

possible customization levels are identified: Fixed variation points and fixed options - the example can be predefined layouts in online Integrated Development Environments (IDEs) Fixed variation points, which allow tenant-supplied options - a non-SaaS example is desktop wallpaper in an OS. A SaaS example is a themed mechanism in Telegram messenger . Users in both cases can supply any item as long as it fits pre-defined requirements. Allow tenants to create their own variation points and options Intelligent customization - applications are created by a user from a list of modules Customizable SaaS infrastructure - in addition to the application, the user can customize SaaS SaaS and PaaS configuration - in addition to application and SaaS, the user can customize PaaS

### **Virtualization**

A virtualization is an act and process of creating a virtual version of an object . Virtualization allows achieving encapsulation and modulation in the system. In the context of SaaS, it is mainly used to divide resources between users (tenants). Also, the majority of IaaS providers, including DigitalOcean, employ virtual machines. In virtualization is considered the highest level of SaaS maturity. In virtualization is considered an approach to support multi-tenancy.

### **Redundancy**

Redundancy is a replication of some elements of the system in order to increase its reliability . Reliability is critical in SaaS. On SaaS level, redundancy can be achieved by having multiple load balancers and more servers than the minimum amount required to serve users. Databases can have replicas with duplicated data.