Platform as a Service (PaaS)

- **PaaS" stands for "Platform as a Service,"** which is a cloud computing model where a third-party provider delivers a complete platform for developing, deploying, and managing applications, allowing users to focus on building their applications without managing the **underlying infrastructure like servers, operating systems, and databases**; essentially, it provides a ready-to-use environment for application development in the cloud.

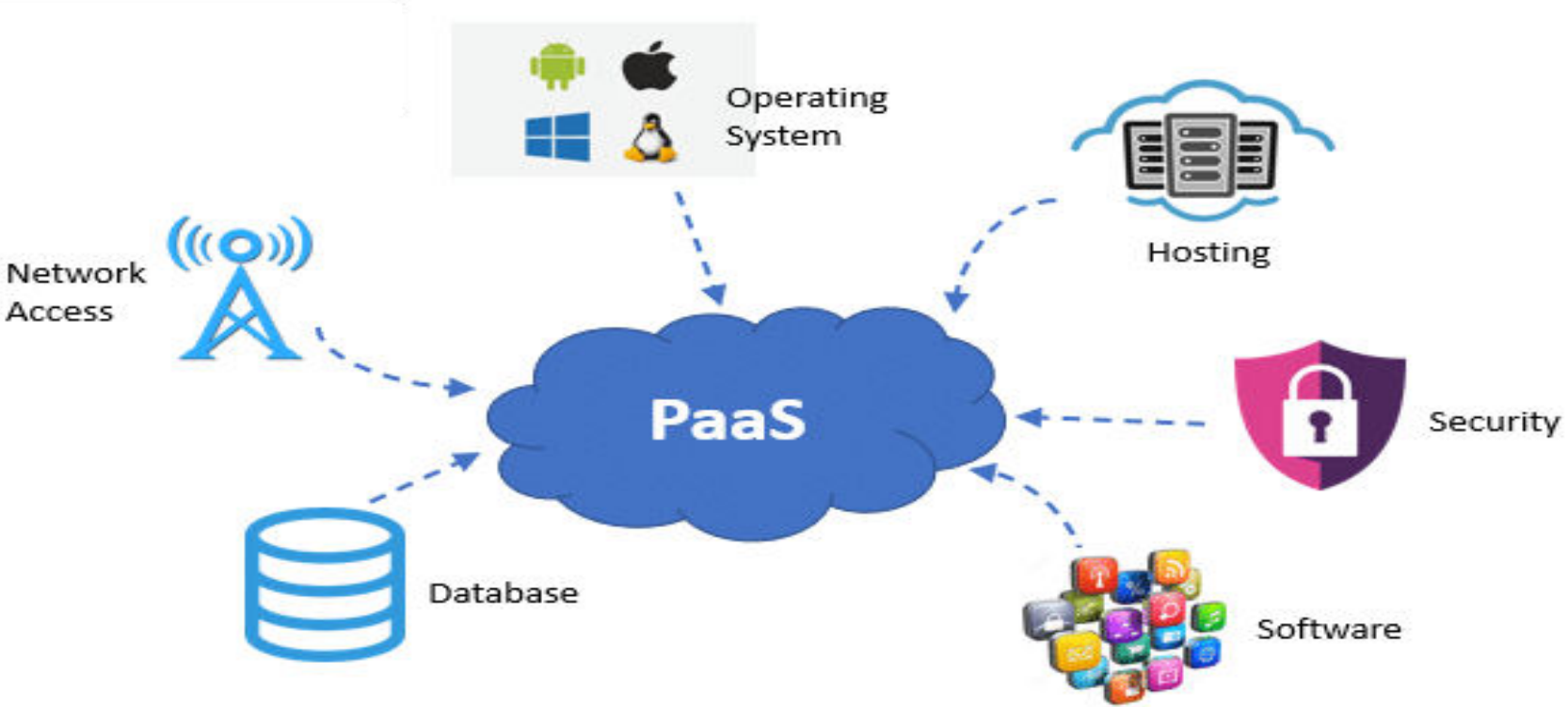
- Platform as a service (PaaS) is a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications.

- The Platform as a Service model describes a software environment in which a developer can create customized solutions within the context of the development tools that the platform provides.

- Platforms can be based on specific types of development languages, application frameworks, or other constructs. A PaaS offering provides the tools and development environment to deploy applications on another vendor's application.

- PaaS tool is a fully integrated development environment; that is, all the tools and services are part of the PaaS service. To be useful as a cloud computing offering, PaaS systems must offer a way to create user interfaces, and thus support standards such as HTLM, JavaScript, or other rich media technologies.

Platform as a Service (PaaS)



Platform as a Service (PaaS)

- In a PaaS model, customers may interact with the software to enter and retrieve data, perform actions, get results, and to the degree that the vendor allows it, customize the platform involved.
- The customer takes no responsibility for maintaining the hardware, the software, or the development of the applications and is responsible only for his interaction with the platform.
- The vendor is responsible for all the operational aspects of the service, for maintenance, and for managing the product(s) lifecycle.
- The one example that is most quoted as a PaaS offering is Google's App Engine platform. Developers program against the App Engine using Google's published APIs.
- The tools for working within the development framework, as well as the structure of the file system and data stores, are defined by Google.
- Another example of a PaaS offering is Force.com, Salesforce.com's developer platform for its SaaS offerings.

Platform as a Service (PaaS)

- Force.com is an example of an add-on development environment. A developer might write an application in a programming language like Python using the Google API.
- The vendor of the PaaS solution is in most cases the developer, who is offering a complete solution to the customer.
- Google itself also serves as a PaaS vendor within this system, because it offers many of its Web service applications to customers as part of this service model.
- You can think of **Google Maps, Google Earth, Gmail, and the myriad** of other PaaS offerings as conforming to the PaaS service model, although these applications themselves are offered to customers.
- The difficulty with PaaS is that it **locks the developer** (and the customer) into a solution that is dependent upon the platform vendor.
- An application written in **Python against Google's API using the Google App** Engine is likely to work only in that environment. There is considerable vendor lock-in associated with a PaaS solution.

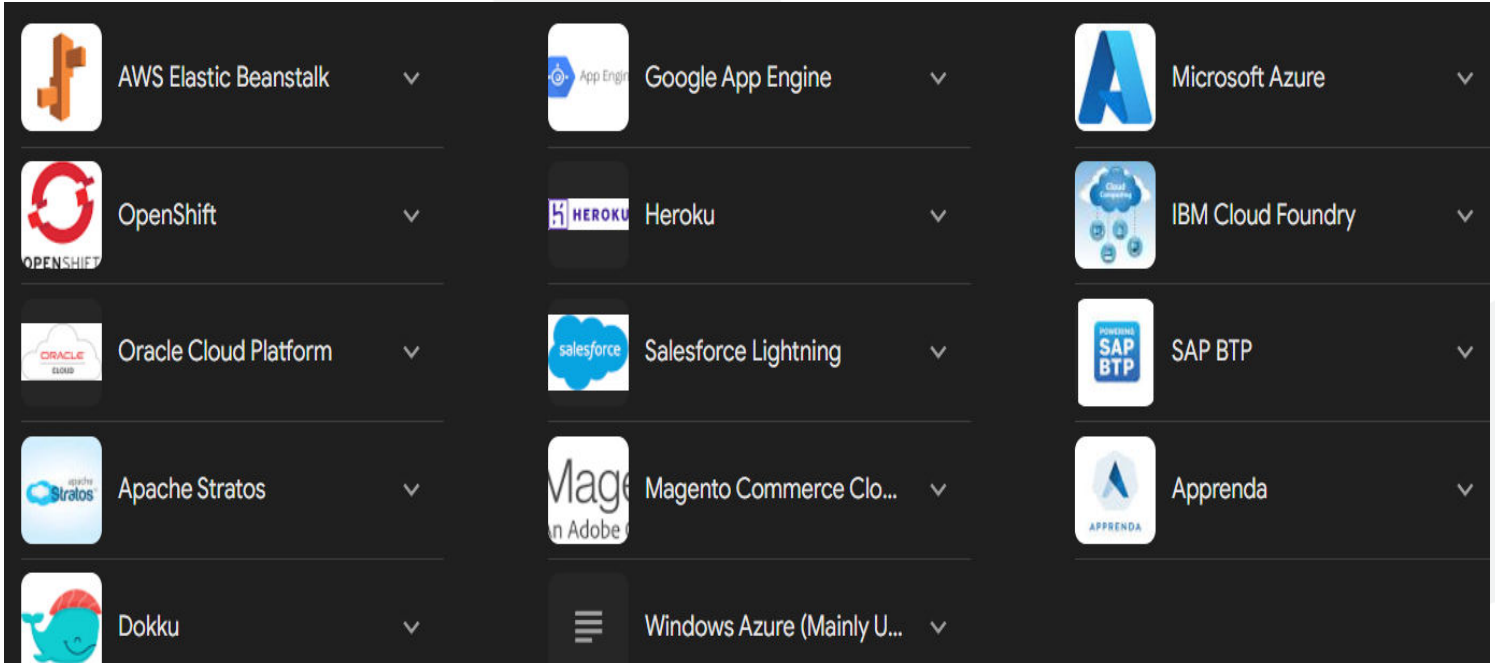
Platform as a Service (PaaS)-Benefits

- **Reduce coding time**: With PaaS development tools, teams can reduce the amount of time spent coding new applications with built-in pre-coded components like workflow, security, search, etc.
- **Increased development capabilities:** PaaS offerings provide development teams with sophisticated tools and added capabilities, without having to bring on new team members to get the job done.
- **Support dispersed or remote teams:** Cloud-native computing environments allow **remote teams to collaborate** and communicate in real-time from different locations.
- **Ability to develop for multiple platforms:** Certain PaaS service providers provide development options for different platforms like computers, mobile, and web browsers.
- **Web application lifecycle support:** With PaaS solutions, development teams have access to all the tools they need to effectively build, test within a virtual machine, deploy, and update apps within an integrated environment.
- **Cost-effective development:** Without having to start from scratch, application development teams that use PaaS can focus on building apps that provide an exceptional user experience. This results in cost savings when it comes to equipment and worker productivity.

Platform as a Service (PaaS)-Benefits

- PaaS is leading the way for a new era in business innovation. Cloud computing is the only way forward for businesses looking to grow in an era of digital transformation.
- Companies big and small should consider transitioning to cloud-based systems for increased flexibility, productivity, and business continuity.
- It offers a cost-effective solution to the development challenges posed by complex infrastructures, allowing businesses to focus their workforce while benefiting from secure data storage, sophisticated tools, and streamlined operations.

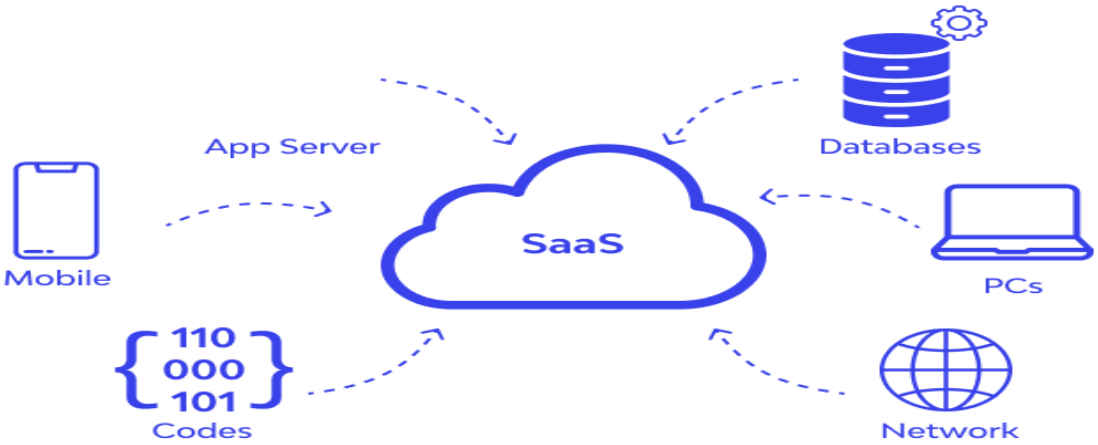
Platform as a Service (PaaS)-Examples



Software as a Service (SaaS)

- SaaS, or Software as a Service, is a cloud-based model for delivering software applications to users over the internet. SaaS is a popular and commonly used cloud computing model.

- Software as a service (SaaS) allows users to connect to and use cloud-based apps over the Internet**. Common examples are email, calendaring, and office tools (such as Microsoft Office 365).** SaaS provides a complete software solution that you purchase on a pay-as-you-go basis from a cloud service provider.



Software as a Service (SaaS)

- The most complete cloud computing service model is one in which the computing hardware and software, as well as the solution itself, are provided by a vendor as a complete service offering.
- It is referred to as the Software as a Service (SaaS) model. SaaS provides the complete infrastructure, software, and solution stack as the service offering
- Software as a Service (SaaS) may be succinctly described as software that is deployed on a hosted service and can be accessed globally over the Internet, most often in a browser.
- With the exception of the user interaction with the software, all other aspects of the service are abstracted away.
- Every computer user is familiar with SaaS systems, which are either replacements or substitutes for locally installed software. Examples of SaaS software for end-users are Google Gmail and Calendar, QuickBooks online, Zoho Office Suite, and others that are equally well known.
- SaaS applications come in all shapes and sizes, and include custom software such as billing and invoicing systems, Customer Relationship Management (CRM) applications, Help Desk applications, Human Resource (HR) solutions, as well as myriad online versions of familiar applications.

Software as a Service (SaaS)-Benefits

- Many people believe that SaaS software is not customizable, and in many SaaS applications this is indeed the case.
- For user-centric applications such as an office suite, that is mostly true; those suites allow you to set only options or preferences.
- However, many other SaaS solutions expose Application Programming Interfaces (API) to developers to allow them to create custom composite applications.
- These APIs may alter the security model used, the data schema, workflow characteristics, and other fundamental features of the service's expression as experienced by the user.
- Examples of an SaaS platform with an exposed API are Salesforce.com and Quicken.com. So SaaS does not necessarily mean that the software is static or monolithic.

Software as a Service (SaaS)

Software as a Service (SaaS)-Benefits

- The software update is not required.
- It does not create expenses like licensing and purchasing the software.
- Server cost is avoided.
- You pay as much as you use the software with no need of purchasing it.
- Since **SaaS** is web-based, you are able to access your data anywhere at any time.
- Needs such as technical maintenance and IT staff disappear.
- You do not need to allocate a budget for security.
- Physical servers are backed up at regular intervals as a precaution against any malfunction.
- The cloud server service is configured in the Cluster infrastructure where multiple computers are running as copies of each other, it continues to run on one of the other redundant servers without interruption when a problem is encountered.

Software as a Service (SaaS)-Characteristics

- The software is available over the Internet globally through a browser on demand.
- The typical license is subscription-based or usage-based and is billed on a recurring basis.
- The software and the service are monitored and maintained by the vendor, regardless of where all the different software components are running.
- There may be executable client-side code, but the user isn't responsible for maintainin that code or its interaction with the service.
- Reduced distribution and maintenance costs and minimal end-user system costs generally make SaaS applications cheaper to use than their shrink-wrapped versions.
- Such applications feature automated upgrades, updates, and patch management and much faster rollout of changes.
- SaaS applications often have a much lower barrier to entry than their locally installed competitors, a known recurring cost, and they scale on demand (a property of cloud computing in general).
- All users have the same version of the software so each user's software is compatible with another's.
- SaaS supports multiple users and provides a shared data model through a single-instance, multi-tenancy model.
- The alternative of software virtualization of individual instances also exists, but is less common.

Shrink-Wrapped versus SaaS Licensing

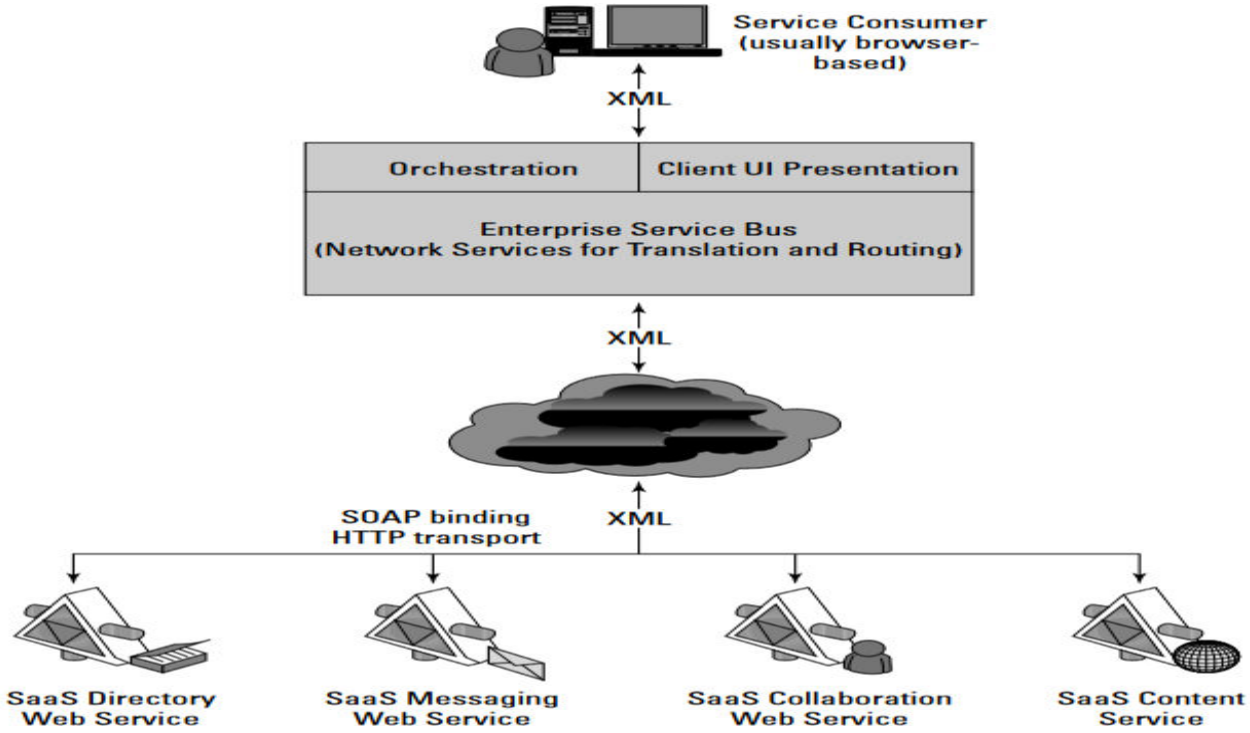
	Shrink-Wrapped Software	Hybrid Model	SaaS
Licensing	Owned	Subscription (flat fee)	Metered subscription
Location	Locally installed	Available through an application	Cloud based
Management	Local IT staff	Application Service Provider (ASP)	Cloud vendor through a Service Level Agreement (SLA)

Open SaaS and SOA

- A considerable amount of SaaS software is based on open source software. When open source software is used in a SaaS, you may hear it referred to as Open SaaS.
- The advantages of using open source software are that systems are much cheaper to deploy because you don't have to purchase the operating system or software, there is less vendor lock-in, and applications are more portable.
- The popularity of open source software, from Linux to APACHE, MySQL, and Perl (the LAMP platform) on the Internet, and the number of people who are trained in open source software make Open SaaS an attractive proposition.
- The impact of Open SaaS will likely translate into better profitability for the companies that deploy open source software in the cloud, resulting in lower development costs and more robust solutions.

Open SaaS and SOA

A modern implement of SaaS using an Enterprise Service Bus and architected with SOA components



Open SaaS and SOA

- The componentized nature of SaaS solutions enables many of these solutions to support a feature called mashups.
- A mashup is an application that can display a Web page that shows data and supports features from two or more sources.
- Annotating a map such as Google maps is an example of a mashup. Mashups are considered one of the premier examples of Web 2.0, and that is technology's ability to support social network systems.
- A mashup requires three separate components: An interactive user interface, which is usually created with HTML/XHTML, Ajax, JavaScript, or CSS.
- Web services that can be accessed using an API, and whose data can be bound and transported by Web service protocols such as SOAP, REST, XML/HTTP, XML/RPC, and JSON/RPC.
- Data transfer in the form of XML, KML (Keyhole Markup Language), JSON (JavaScript Object Notation), or the like.

Open SaaS and SOA

- Mashups are an incredibly useful hybrid Web application, one that SaaS is a great enabler for. The Open Mashup Alliance (OMA; see http://www.openmashup.org/) is a non-profit industry group dedicated to supporting technologies that implement enterprise mashups.

- This group supports the developing standard, the Enterprise Mashup Markup Language (EMML), which is a Domain Specific Language (DSL). This group predicts that the use of mashups will grow by a fac- tor of 10 within just a few years.

- Gartner Group predicts that approximately 25 percent of all software sold by 2011 will use the SAAS model, offered either by vendors or an intermediary party, sometimes referred to as an aggregator.

- An aggregator bundles SaaS applications from different vendors and presents them as part of a unified platform or solution.

Salesforce.com and CRM SaaS

- The best-known example of Software as a Service (SaaS) is the Customer Relationship Management software offered by Salesforce.com whose solution offers sales, service, support, marketing, content, analytical analysis, and even collaboration through a platform called Chatter.
- Salesforce.com was founded in 1999 by a group of Oracle executives and early adopters of many of the technologies that are becoming cloud computing staples.
- Salesforce.com extended its SaaS offering to allow developers to create add-on applications, essentially turning the SaaS service into a Platform as a Service (PaaS) offering called the Force.com
- Platform. Applications built on Force.com are in the form of the Java variant called Apex using an XML syntax for creating user interfaces in HTML, Ajax, and Flex.
- Nearly a thousand applications now exist for this platform from hundreds of vendors.

Salesforce.com and CRM SaaS

Salesforce.com is the largest SaaS provider of CRM software and a pioneer in this type of cloud computing software. This is the company's home page.