

CLOUD COMPUTING PLATFORM AS A SERVICE PAAS

http://www.tutorialspoint.com/cloud_computing/cloud_computing_platform_as_a_service.htm

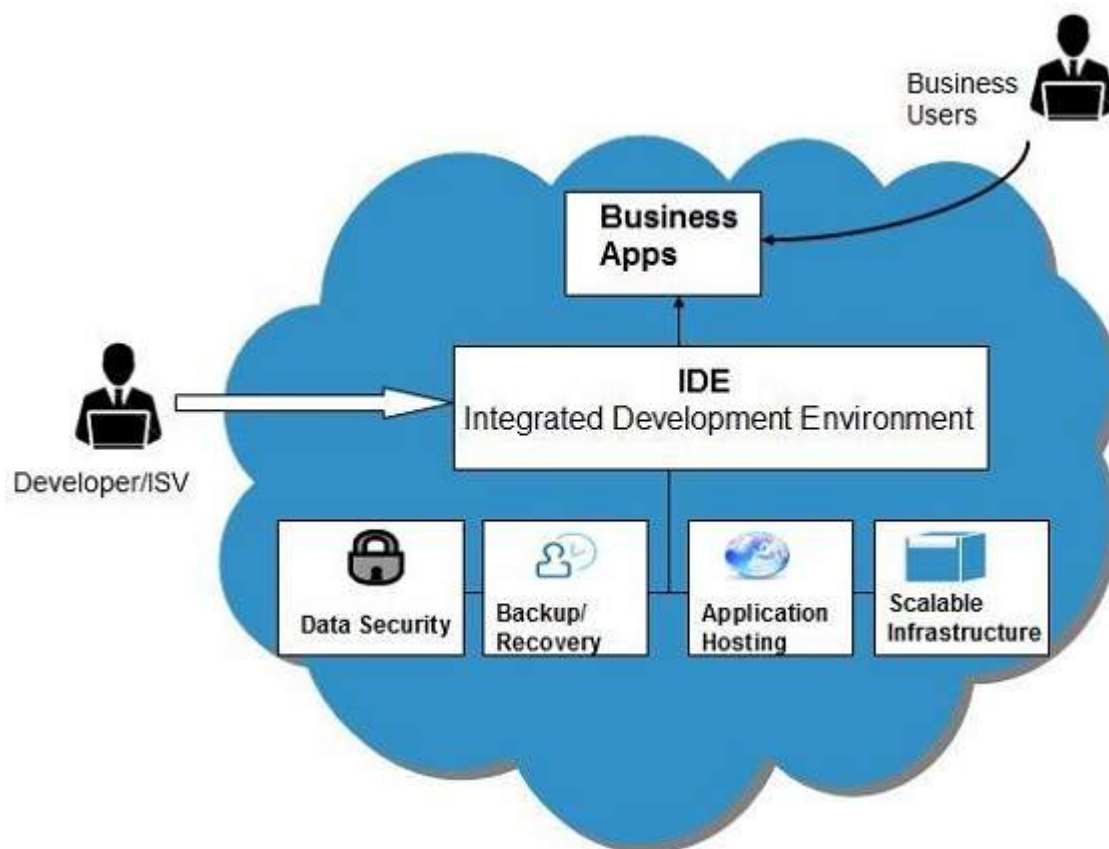
Copyright © tutorialspoint.com

Platform-as-a-Service offers the runtime environment for applications. It also offers development and deployment tools required to develop applications. PaaS has a feature of **point-and-click** tools that enables non-developers to create web applications.

App Engine of Google and **Force.com** are examples of PaaS offering vendors. Developer may log on to these websites and use the **built-in API** to create web-based applications.

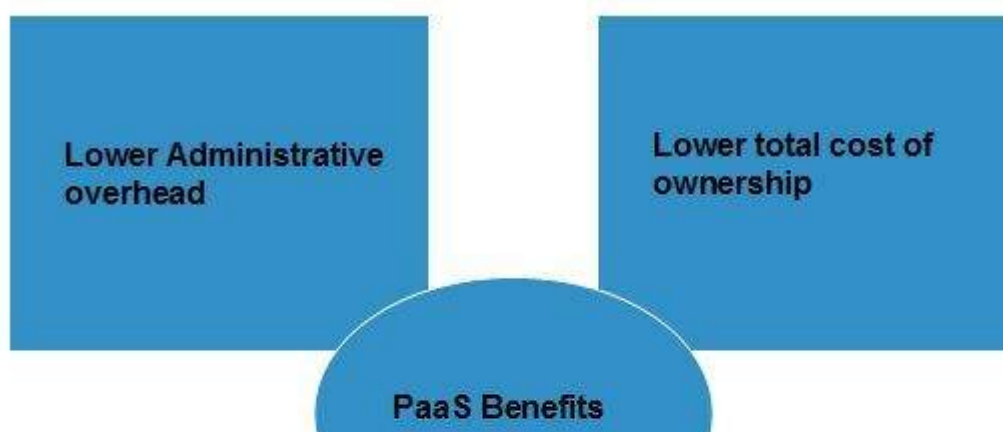
But the disadvantage of using PaaS is that, the developer **locks-in** with a particular vendor. For example, an application written in Python against API of Google, and using App Engine of Google is likely to work only in that environment.

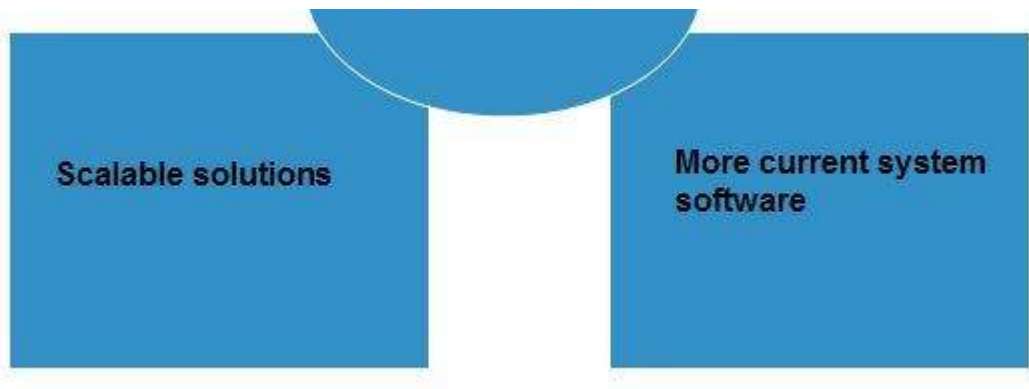
The following diagram shows how PaaS offers an API and development tools to the developers and how it helps the end user to access business applications.



Benefits

Following are the benefits of PaaS model:





Lower administrative overhead

Customer need not bother about the administration because it is the responsibility of cloud provider.

Lower total cost of ownership

Customer need not purchase expensive hardware, servers, power, and data storage.

Scalable solutions

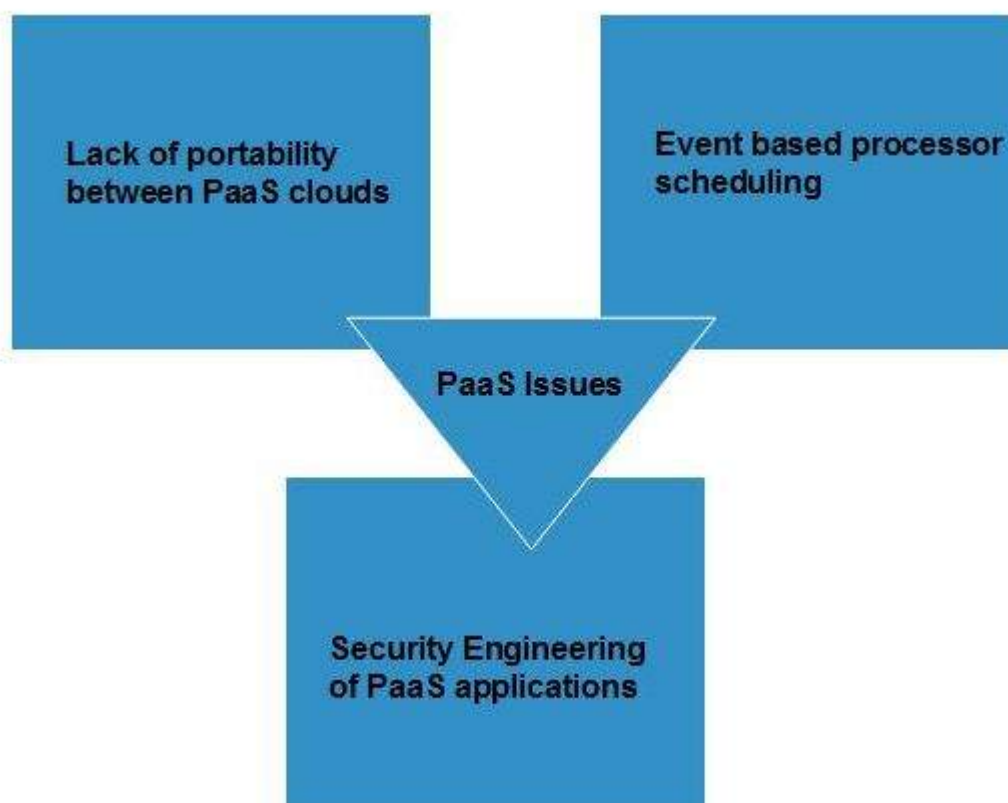
It is very easy to scale the resources up or down automatically, based on their demand.

More current system software

It is the responsibility of the cloud provider to maintain software versions and patch installations.

Issues

Like **SaaS**, **PaaS** also places significant burdens on customer's browsers to maintain reliable and secure connections to the provider's systems. Therefore, PaaS shares many of the issues of SaaS. However, there are some specific issues associated with PaaS as shown in the following diagram:



Lack of portability between PaaS clouds

Although standard languages are used, yet the implementations of platform services may vary. For example, file, queue, or hash table interfaces of one platform may differ from another, making it difficult to transfer the workloads from one platform to another.

Event based processor scheduling

The PaaS applications are event-oriented which poses resource constraints on applications, i.e., they have to answer a request in a given interval of time.

Security engineering of PaaS applications

Since PaaS applications are dependent on network, they must explicitly use cryptography and manage security exposures.

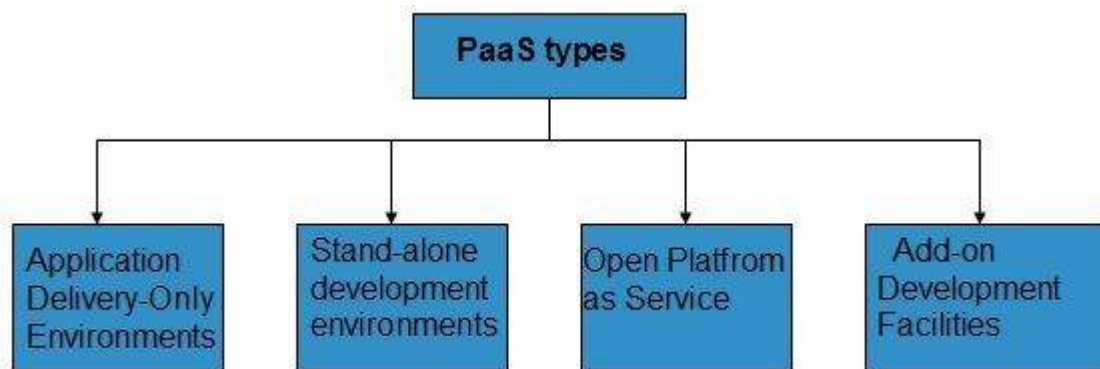
Characteristics

Here are the characteristics of PaaS service model:

- PaaS offers **browser based development environment**. It allows the developer to create database and edit the application code either via Application Programming Interface or point-and-click tools.
- PaaS provides **built-in security, scalability, and web service interfaces**.
- PaaS provides built-in tools for defining **workflow, approval processes**, and business rules.
- It is easy to integrate PaaS with other applications on the same platform.
- PaaS also provides web services interfaces that allow us to connect the applications outside the platform.

PaaS Types

Based on the functions, PaaS can be classified into four types as shown in the following diagram:



Stand-alone development environments

The **stand-alone PaaS** works as an independent entity for a specific function. It does not include licensing or technical dependencies on specific SaaS applications.

Application delivery-only environments

The **application delivery PaaS** includes **on-demand scaling** and **application security**.

Open platform as a service

Open PaaS offers an **open source software** that helps a PaaS provider to run applications.

Add-on development facilities

The ~~add-on PaaS~~ allows to customize the existing SaaS platform.

Loading [MathJax]/jax/output/HTML-CSS/jax.js