

Name - Harshit Parsai
Section - K17TA
Roll no. – 09
Reg. No - 11702448
Subject - Cloud
CA 2 (Article)

MODELS OF CLOUD COMPUTING

Introduction - The cloud is a hot topic for small businesses all the way to global enterprises, but remains a broad concept that covers a lot of online territory. As you begin to consider switching your business to the cloud, whether it be for application or infrastructure deployment, it is more important than ever to understand the differences and advantages of the various cloud services.

There are usually three models of cloud service to compare: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Each of these has its own benefits, as well as variances, and it is necessary to understand the differences among SaaS, PaaS, and IaaS to know how to best choose one for your organization.

SaaS: Software as a Service

Software as a Service, also known as cloud application services, represents the most commonly utilized option for businesses in the cloud market. SaaS utilizes the internet to deliver applications, which are managed by a third-party vendor, to its users. A majority of SaaS applications run directly through your web browser, which means they do not require any downloads or installations on the client side. Due to its web delivery model, SaaS eliminates the need to have IT staff download and install applications on each individual computer. With SaaS, vendors manage all potential technical issues, such as data, middleware, servers, and storage, resulting in streamlined maintenance and support for the business.

SaaS Advantages

SaaS provides numerous advantages to employees and companies by greatly reducing the time and money spent on tedious tasks such as installing, managing, and upgrading software. This frees up plenty of time for technical staff to spend on more pressing matters and issues within the organization.

SaaS Characteristics

1. Managed from a central location
2. Hosted on a remote server
3. Accessible over the internet.
4. Users not responsible for hardware or software updates

SaaS Limitations and Concerns

1. **Interoperability.** Integration with existing apps and services can be a major concern if the SaaS app is not designed to follow open standards for integration. In this case, organizations may need to design their own integration systems or reduce dependencies with SaaS services, which may not always be possible.
2. **Vendor lock-in.** Vendors may make it easy to join a service and difficult to get out of it. For instance, the data may not be portable—technically or cost-effectively—across SaaS apps from other vendors without incurring significant cost or inhouse engineering rework. Not every vendor follows standard APIs, protocols, and tools, yet the features could be necessary for certain business tasks.
3. **Lack of integration support.** Many organizations require deep integrations with on-premise apps, data, and services. The SaaS vendor may offer limited support in this regard, forcing organizations to invest internal resources in designing and managing integrations. The complexity of integrations can further limit how the SaaS app or other dependent services can be used.
4. **Data security.** Large volumes of data may have to be exchanged to the backend data centers of SaaS apps in order to perform the necessary software functionality. Transferring sensitive business information to public-cloud based SaaS service may result in compromised security and compliance in addition to significant cost for migrating large data workloads.
5. **Customization.** SaaS apps offer minimal customization capabilities. Since a one-size-fits-all solution does not exist, users may be limited to specific functionality, performance, and integrations as offered by the vendor. In contrast, on-premise solutions that come with several software development kits (SDKs) offer a high degree of customization options.
6. **Lack of control.** SaaS solutions involves handing control over to the third-party service provider. These controls are not limited to the software—in terms of the version, updates, or appearance—but also the data and governance. Customers may therefore need to redefine their data security and governance models to fit the features and functionality of the SaaS service.
7. **Feature limitations.** Since SaaS apps often come in a standardized form, the choice of features may be a compromising tradeoff against security, cost, performance, or other organizational policies. Furthermore, vendor lock-in, cost, or security concerns may mean it's not viable to switch vendors or services to serve new feature requirements in the future.

8. **Performance and downtime.** Because the vendor controls and manages the SaaS service, your customers now depend on vendors to maintain the service's security and performance. Planned and unplanned maintenance, cyber-attacks, or network issues may impact the performance of the SaaS app despite adequate service level agreement (SLA) protections in place.

Examples of SaaS

These are several popular examples of SaaS, including: [Google GSuite](#) (Apps), [Dropbox](#), [Salesforce](#), [Cisco WebEx](#), [SAP Concur](#), and [GoToMeeting](#).

PaaS: Platform as a Service

Cloud platform services, also known as Platform as a Service (PaaS), provide cloud components to certain software while being used mainly for applications. PaaS delivers a framework for developers that they can build upon and use to create customized applications. All servers, storage, and networking can be managed by the enterprise or a third-party provider while the developers can maintain management of the applications. The delivery model of PaaS is similar to SaaS, except instead of delivering the software over the internet, PaaS provides a platform for software creation. This platform is delivered via the web, giving developers the freedom to concentrate on building the software without having to worry about operating systems, software updates, storage, or infrastructure. PaaS allows businesses to design and create applications that are built into the PaaS with special software components. These applications, sometimes called middleware, are scalable and highly available as they take on certain cloud characteristics.

PaaS Advantages

1. Simple, cost-effective development and deployment of apps
2. Scalable
3. Highly available
4. Developers can customize apps without the headache of maintaining the software
5. Significant reduction in the amount of coding needed
6. Automation of business policy
7. Easy migration to the hybrid model

PaaS Characteristics

1. Builds on virtualization technology, so resources can easily be scaled up or down as your business changes
2. Provides a variety of services to assist with the development, testing, and deployment of apps
3. Accessible to numerous users via the same development application
- Integrates web services and databases

PaaS Limitations and Concerns

1. **Data security.** Organizations can run their own apps and services using PaaS solutions, but the data residing in third-party, vendor-controlled cloud servers poses security risks and concerns. Your security options may be limited as customers may not be able to deploy services with specific hosting policies
2. **Integrations.** The complexity of connecting the data stored within an onsite data center or off-premise cloud is increased, which may affect which apps and services can be adopted with the PaaS offering. Particularly when not every component of a legacy IT system is built for the cloud, integration with existing services and infrastructure may be a challenge.
3. **Vendor lock-in.** Business and technical requirements that drive decisions for a specific PaaS solution may not apply in the future. If the vendor has not provisioned convenient migration policies, switching to alternative PaaS options may not be possible without affecting the business.
4. **Customization of legacy systems.** PaaS may not be a plug-and-play solution for existing legacy apps and services. Instead, several customizations and configuration changes may be necessary for legacy systems to work with the PaaS service. The resulting customization can result in a complex IT system that may limit the value of the PaaS investment altogether.
5. **Runtime issues.** In addition to limitations associated with specific apps and services, PaaS solutions may not be optimized for the language and frameworks of your choice. Specific framework versions may not be available or perform optimally with the PaaS service. Customers may not be able to develop custom dependencies with the platform.
6. **Operational limitation.** Customized cloud operations with management automation workflows may not apply to PaaS solutions, as the platform tends to limit operational capabilities for end users. Although this is intended to reduce the operational burden on end users, the loss of operational control may affect how PaaS solutions are managed, provisioned, and operated.

Examples of PaaS

Popular examples of PaaS include [AWS Elastic Beanstalk](#), [Windows Azure](#), [Heroku](#), [Force.com](#), [Google App Engine](#), and [OpenShift](#).

IaaS: Infrastructure as a Service

Cloud infrastructure services, known as Infrastructure as a Service (IaaS), are made of highly scalable and automated compute resources. IaaS is fully [self-service](#) for accessing and monitoring computers, networking, storage, and other services. IaaS allows businesses to purchase resources on-demand and as-needed instead of having to buy hardware outright. IaaS delivers cloud computing infrastructure, including servers, network, operating systems, and storage, through virtualization technology. These cloud servers are typically provided to the organization through a dashboard or an API, giving IaaS clients complete control over the entire infrastructure. IaaS provides the same technologies and capabilities as a traditional data center without having to physically maintain or manage all of it. IaaS clients can still access their servers and storage directly, but it is all outsourced through a “virtual data center” in the cloud. As opposed to SaaS or PaaS, IaaS clients are responsible for managing aspects such as applications, runtime, OSes, middleware, and data. However, providers of the IaaS manage the servers, hard drives, networking, virtualization, and storage. Some providers even offer more services beyond the virtualization layer, such as databases or message queuing.

IaaS Advantages

1. The most flexible cloud computing model
2. Easy to automate deployment of storage, networking, servers, and processing power
3. Hardware purchases can be based on consumption
4. Clients retain complete control of their infrastructure
5. Resources can be purchased as-needed
6. Highly scalable

IaaS Characteristics

1. Resources are available as a service 2. Cost varies depending on consumption 3. Services are highly scalable 4. Multiple users on a single piece of hardware 5. Organization retain complete control of the infrastructure 6. Dynamic and flexible

IaaS Limitations and Concerns

Many limitations associated with SaaS and PaaS models – such as data security, cost overruns, vendor lock-in and customization issues – also apply to the IaaS model. Particular limitations to IaaS include:

1. **Security.** While the customer is in control of the apps, data, middleware, and the OS platform, security threats can still be sourced from the host or other virtual machines (VMs). Insider threat or system vulnerabilities may expose data communication between the host infrastructure and VMs to unauthorized entities.
2. **Legacy systems operating in the cloud.** While customers can run legacy apps in the cloud, the infrastructure may not be designed to deliver specific controls to secure the legacy apps. Minor enhancement to legacy apps may be required before migrating them to the cloud, possibly leading to new security issues unless adequately tested for security and performance in the IaaS systems.
3. **Internal resources and training.** Additional resources and training may be required for the workforce to learn how to effectively manage the infrastructure. Customers will be responsible for data security, backup, and business continuity. Due to inadequate control into the infrastructure however, monitoring and management of the resources may be difficult without adequate training and resources available inhouse.
4. **Multi-tenant security.** Since the hardware resources are dynamically allocated across users as made available, the vendor is required to ensure that other customers cannot access data deposited to storage assets by previous customers. Similarly, customers must rely on the vendor to ensure that VMs are adequately isolated within the multitenant cloud architecture.

Examples of IaaS

Popular examples of IaaS include [DigitalOcean](#), [Linode](#), [Rackspace](#), [Amazon Web Services \(AWS\)](#), [Cisco Metacloud](#), [Microsoft Azure](#), and [Google Compute Engine \(GCE\)](#).

Conclusion - We hope that definitions and examples above made the differences between IaaS vs PaaS vs SaaS and where they fit into cloud computing stack much clearer. Ultimately, the philosophy behind cloud computing is to manage only the things you directly working on, without the need to handle all the additional stuff that goes with it, and distinct offer of PaaS vs SaaS vs IaaS allows customers to choose exactly what they need. At Apriorit, we provide SaaS development services for our clients and know firsthand how popular SaaS model really is. If you were still considering whether it is worth it to go cloud computing, we hope that the information, provided in our comparison helped you to make your mind and embrace this popular and convenient delivery model.