Beyond The Basics of The Cloud and Making Smart Platform Decisions

Introduction

When discussing the cloud landscape, most focus on the three main layers: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). This three-layer framework is a useful starting point for describing cloud products, but it's far from perfect.

Nowadays, saying you work in one of these three businesses is almost as broad as saying "I work for a software company." The lines between these categories are blurring and within these categories there are numerous subcategories that describe a whole range of different approaches.

It's time to have a closer look at the cloud landscape, to really understand what IaaS, PaaS, and SaaS are (and are not), and to distinguish the different categories within. This white paper serves three important functions. It will:

1. Propose a framework to categorize the different approaches seen on the market today.

2. Illustrate that not all cloud platforms are created equal and offer decision makers the ability to make more nuanced comparisons based on the app projects within their organization.

3. Showcase why increased demand for innovating and differentiating apps requires a platform that abstracts away from technical details across the app lifecycle, creating the greatest efficiency to custom app development.

The Complete Cloud Landscape

CLOUD-BASED SERVICES MODEL			
SAAS LAYER	Applications provided as cloud services		
MODEL-DRIVEN PAAS LAYER	Model-Driven aPaaS	Model-Driven iPaaS	baPaaS (Business Analytics Platform-as-a-Service)
PAAS LAYER	aPaaS (App Platform-as-a-Services) and Language Runtimes	iPaaS (Integration-Platform-as-a-Service) and ESB-as-a-Service	dbPaaS (Database Platform-as-a-Service)
FOUNDATIONAL PAAS LAYER	Application Containers	Routing, Messaging, and Scheduling System	Object Storage
IAAS LAYER	Virtual Machines	SDN (Software-Defined Networking)	Block Storage
	COMPUTE	COMMUNICATE	STORE

**Note that the higher layers in this model do not necessarily need to be built on the exact components below it.*

The Cloud Landscape Described

IaaS Layer: Virtual Machines

Server virtualization lays the foundation for all of the scalability and flexibility advantages that cloud computing provides. Most PaaS and SaaS products use some form of virtualized infrastructure in order to provide flexible services. If virtualized infrastructure is being provided to users as an on-demand commodity with additional capabilities, services, and resources, those services are known as IaaS offerings.

The additional services are what separates IaaS from basic server virtualization. IaaS offerings are made available through a provider's virtualized datacenter, which means that the provider abstracted the logical view of the servers and pooled them into universal compute resources that users buy on-demand. This is usually cheaper than traditional server hosting because the provider can manage compute resources more easily and use server space more efficiently.

Examples: Amazon EC2, Digital Ocean, Rackspace, Windows Azure, Google Compute

IaaS Layer: SDN

Network virtualization, often referred to as Software Defined Networking (SDN), decouples and isolates virtual networks from the underlying network hardware, in the same way that server virtualization abstracts the underlying physical hardware. The network hardware is only used for the physical link; virtual networks can be created programmatically.

These virtual networks offer the same features and guarantees of a physical network, but with the benefits of virtualization like higher utilization of the hardware, more flexibility, and easier management.

Examples: Nicira (acquired by VMWare) and OpenStack Neutron (formerly Quantum)

IaaS Layer: Block Storage

Storage virtualization is the third field of the IaaS layer. Decoupling storage volumes from storage hardware allows vendors to manage advanced features like high availability, caching, snapshotting and others at the software layer, independent of the hardware

brand. This opens the door to more flexible optimization and better storage performance.

Examples: Amazon Elastic Block Storage (EBS), OpenStack Cinder

Foundational PaaS Layer: Application Containers

In the foundational PaaS layer, the focus in the compute column moves to application components that are infrastructure-agnostic. Once again, this provides additional flexibility benefits similar to the IaaS layer, where compute resources become hardware-agnostic. To run an application on any computing infrastructure you need to have: compiled programming code (Java, C#, PHP, etc.), a runtime (a Java Virtual Machine, or the Common Language Runtime for example) and sometimes additional middleware (Tomcat, Spring, Ruby on Rails, etc.).

A package with these pieces can be deployed in an application container and used as a template for application development and deployment. The container also automates the handling of application dependencies. Dependencies are supplemental programs that are required in order for the application to run. All this allows the developer to concentrate on the application itself rather than the ancillary components required to run it.

Examples: Docker containers, Heroku Dynos, WSO2 Cartridges, Cloud Foundry Warden Containers.

Foundational PaaS Layer: Routing, Messaging and Scheduling System

The communicate column of this layer focuses on routing messages among app containers, storage systems and external systems. This includes routing, queuing, and scheduling system instructions for app lifecycle management, as well as the smart routing and load-balancing of incoming requests to application instances.

Examples: the messaging / NATS component of CloudFoundry, Heroku routing layer, CloudFoundry router

Foundational PaaS Layer: Object Storage

Unlike block storage, object storage contains data components that are

ready to be retrieved and manipulated in an application. An object storage service automatically handles mounting, partitioning, and formatting virtual disks. With cloud-based object storage, the costs are usually low because you only pay to store the objects you actually use. In most cases, you can retrieve objects via an HTTP application programming interface (API).

Examples: Amazon S3, Ceph Storage, Rackspace Cloud Files

PaaS Layer: aPaaS and Language Runtimes

While the foundational PaaS layer dealt with binary code, the PaaS layer above it deals with the application code in its human-readable form, before compilation. Developers can just upload their code to an aPaaS without having to worry about compilation or middleware.

aPaaS, which is essentially an application server-as-a-service, is the core component of the general PaaS category that is so often talked about in the IT industry. It consists of the programming language-specific runtime that allows customers to run their applications.

Examples: Heroku Buildpacks, Cloud Foundry Buildpacks

PaaS Layer: iPaaS

In the communicate column, we see what is called iPaaS (integration Platform-as-a-Service). An iPaaS provides the communication among applications, whether in the cloud or on-premise. This communication is defined using integration flows. An iPaaS could be seen as an ESB (Enterprise Service Bus) in the cloud and contains features like routing, transformations, queuing, and in most cases, a range of pre-defined adapters.

Examples: TIBCO Cloud Bus, WS02 StratosLive, Windows Azure BizTalk Services

PaaS Layer: dbPaaS

Database Platform-as-a-Service (dbPaaS) delivers the entire database management system (DBMS) as a cloud service to the customer. dbPaaS

solutions can be provided as component services of larger PaaS offerings or as individual services unaffiliated with any larger PaaS offering.

Examples: Amazon RDS and RedShift, Lunacloud Cloud Mongo, Heroku Postgres

Model-driven PaaS Layer

The Model-Driven PaaS layer builds on language runtimes and provides additional support for domain specific languages (DSLs) that make certain kinds of application development faster and less error prone. These DSLs are often easy enough for business-level users with less software development expertise to understand and use effectively.

These DSLs won't be able to do some of the deep customizations that standard programming languages can provide, but most model-driven platforms can also harness standard languages when they are needed. This layer can also be considered a subset of aPaaS.

Examples: Salesforce Force.com, Mendix App Platform

SaaS Layer

We call the applications on this layer Software-as-a-Service (SaaS). SaaS is as-a-Service from the user perspective. From the developer/provider perspective, SaaS can be built on top of the previously mentioned layers, but if an application for example isn't built using a PaaS and runs on dedicated hardware, it can still be SaaS.

This basically holds for each layer in the framework: it can be, but is not necessarily built on top of the layers below. It's a matter of choice what part of their stack they get from another provider as a service and what part of the stack they build themselves.

Making the Right Cloud Platform Decisions

Looking at the complete framework above, we see that the five layers provide a nice continuum between infrastructure and applications. Moving up, each layer abstracts further away from technical details until we end up with the applications themselves.

This framework provides a much more accurate classification of the current cloud landscape than the overused and oversimplified SaaS-PaaS-IaaS (SPI) framework.

And with a more nuanced framework, IT teams can align their cloud platform selections with their strategic plans and business objectives. Consider the following:

1. Where can your IT organization offer the most strategic value?

2. Where should you focus your IT investments?

3. What is the cloud solution that best aligns with your strategic plan?

By answering these questions in this order, you will quickly arrive at the service type (and ideal vendor functionality) that fits your strategic plans.

Understanding Your Strategic Value

Every IT team faces a challenge when it comes to new application and feature delivery. But you need to take a step back to prioritize your backlogs in the context of the bigger picture.

Digital transformation is driving demand for custom apps that innovate business practices and differentiate your business within the market. SaaS solutions don't exist in these areas because the needs are specific to each company's unique processes and business models. Moreover, the demand for this type of application is greater than ever before. Not only does the business need more of them, but they need them faster.

Currently, 71% of IT teams are behind, unable to meet business demand for app development. With greater pressure on all businesses to innovate, there is greater need for IT to become a more strategic partner – delivering on the systems of innovation and differentiation that will drive overall success.

Before delving into your backlog, recognize the unique features of these new systems of innovation and differentiation. Across the board, the business is looking for:

- Multi-channel, multi-device apps
- Faster turnaround of new apps and new app features
- Greater involvement in the app development process
- More frequent updates to iterate and improve each app

Unfortunately, you won't get the results you need by streamlining deployment alone, as most PaaS solutions do. You need to enhance the complete application lifecycle by combining streamlined deployment with a new way of building of apps.

But you also need a sustainable way forward, with existing maintenance completed alongside new business-oriented innovation projects. Put another way, you need a "fast lane" to support strategic business initiatives.

Focusing Your Investment Dollars on a Transformative Solution

By creating a bimodal strategy within your IT department, you can continue to leverage your development group for large-scale system maintenance efforts. At the same time, you can enable less-technical users to participate in rapid development cycles through the use of the right cloud services.

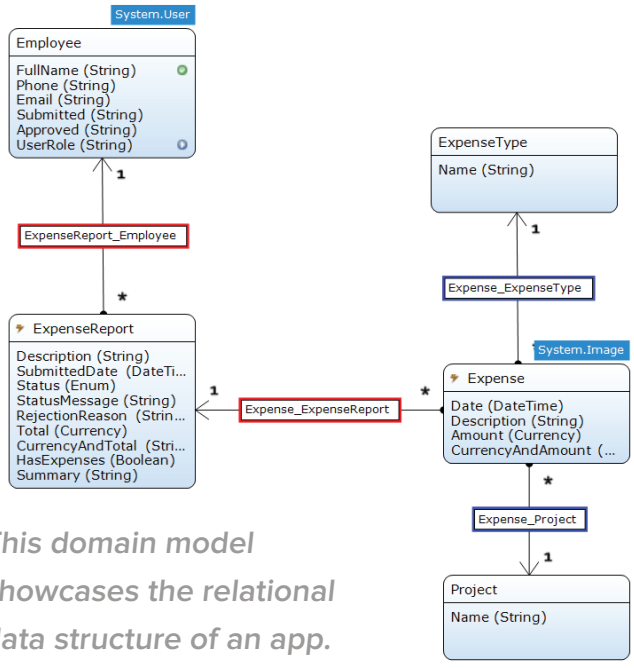
However, the nuances of the cloud landscape truly become apparent in this scenario. While the general PaaS category offers a number of enhancements to deployment – it is not enough. We need further abstraction in order to meet the speed and agility requirements associated with these app projects.

The layer best suited to meet these needs is model-driven aPaaS. You need a platform that supports visual modeling, and more agile and collaborative development.

A visual model

Previously, the complexity of IT meant that only highly-skilled specialists could work in complex programming languages. But faced with the need for greater speed and simplicity, companies are looking to domain experts to take an active role in the app development process.

Model-driven aPaaS makes it easy enough for business-level users to build apps themselves, as well as easily understand/review functionality developed by others. In the process, professional developers are freed up to focus on more technical details of applications, such as integration or performance.



This domain model showcases the relational data structure of an app.

Focusing Your Investment Dollars on a Transformative Solution

A more agile/collaborative process

The inherent value of model-driven aPaaS is that it brings IT and the business together, enabling more rapid, iterative and collaborative development. Because visual models are used to define application logic, process flow and layouts, these platforms empower both developers and business users to rapidly build applications, without the need for low-level coding.

In addition, model-driven aPaaS ensures short feedback cycles as changes to a model can quickly and directly be tested in the actual application. It also provides an excellent communication mechanism to align business and IT stakeholders, thereby ensuring greater quality and more successful outcomes.

Questionnaire: Making The Right Platform Decision

To innovate and differentiate within the market, you need a host of multi-channel, multi-device applications that integrate with, and extend, your systems and processes. The question isn't whether you need help fulfilling the demand for these apps – it's what you should use to design, develop, deploy, and manage these apps.

Model-driven aPaaS is a rising category within the cloud landscape, but not all vendors provide the same features and functionality. Use the questionnaire below to ensure that you select the right vendor for your app delivery needs.

1. How easy is it to build applications?

Incremental improvements to code-based methods simply won't suffice. You'll want to pay careful attention to how easy it is for non-developers to actively participate in the process.

- ☐ Does it leverage a proprietary programming language that requires highly-specialized developers? Or is it built around a visual paradigm that allows both developers and business users to build and review apps?
- ☐ Does the platform execute visual models in the runtime, which means the model is the application? Or does it generate code from the model in order to run the app, a technique that's inflexible and results in maintainability challenges?
- ☐ Is there an app store full of pre-built widgets and components that can be easily used in new apps, saving time and effort?

2. Can the platform be used to integrate with, or extend, any existing systems?

While certain platforms are geared towards extending specific systems (i.e. CRM), they may fall short in addressing your full range of app delivery and integration needs.

- ☐ Can you easily build new apps and extend existing enterprise systems (e.g. SAP, Oracle) with modern, multi-channel apps?
- ☐ Can you seamlessly integrate new apps with any existing process and system?
- ☐ Are there capabilities like workflow integration, open APIs and prepackaged connectors?

Questionnaire: Making The Right Platform Decision

3. Are there capabilities to facilitate IT/ business collaboration?

Many aPaaS solutions overlook crucial capabilities for keeping all stakeholders aligned and engaged throughout the project lifecycle.

- ☐ Are there native social collaboration features, including Facebook-like activity streams and built-in chat?
- ☐ Are there agile project management tools for creating user stories, managing sprints and releases, and tracking progress with scrum boards and burn-down charts?
- ☐ Can developers quickly share prototypes that are working applications with the team or groups of end users for feedback?

4. What scalability and performance features are available?

Certain aPaaS solutions may only be suited for lightweight departmental apps, not core enterprise systems. Depending on your specific needs, you may want to ask:

- ☐ Does the platform offer the required scalability, performance, security?
- ☐ What are some performance benchmarks for the platform?
- ☐ Are there clients who have successfully delivered core systems?

Questionnaire: Making The Right Platform Decision

5. Can you deliver truly multi-device applications?

While enterprise mobile apps are hot, it's important to remember that mobile is not- and show never be – an island.

- ☐ Can you build an application once and make it accessible to users via all devices without developing for multiple platforms and form factors?
- ☐ Are there capabilities for easily building responsive mobile and tablet-specific interfaces?
- ☐ Does the platform support native, hybrid, or web app development, and how will this impact the user experience?
- ☐ Can you easily manage all of the back-end requirements of enterprise mobile apps?

6. How fast and flexible is the deployment process?

Deployment issues should not slow a project down or even concern your development team. When evaluating aPaaS, it's essential that deployment is as fast and easy as plugging an appliance into a power outlet.

- ☐ Can you deploy an app to the cloud with one click?
- ☐ Can you move seamlessly from test to acceptance to production environments?
- ☐ Are there flexible deployment options: public cloud, private cloud, hybrid or on premise?
- ☐ Can you deploy to multiple public cloud providers?

Make The Right Choice for Your Business

Technology continues to develop and we've moved passed the time where a three layer cloud environment make sense. In our five-layer framework, we aim to help all users understand the nuances associated with application delivery.

Model-driven aPaaS is the hottest category within the new cloud landscape, and especially critical for organizations in need of fast and iterative app development. Don't forget our questionnaire when evaluating potential providers. Contact Mendix for more information on how you can make the right choice for your business.

Contact Mendix to Learn More.

Learn More Now ▸

mendix.com

Mendix is the app platform company for the enterprise. We enable companies to build, integrate and deploy web and mobile applications faster and with better results, effectively driving ROI in days not months. Learn more, join our user community and get started for free at **now.mendix.com.**