

---

# To PaaS or not to PaaS – that is the question

## A look at the evolving role of Platform as a Service

By Dale Vile, April 2013

Originally published on



<http://www.theregister.co.uk/>

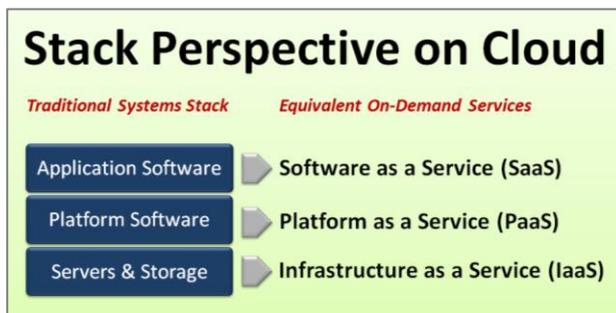
Most forms of cloud computing are starting to find their place in mainstream IT delivery. Infrastructure as a Service (IaaS) is providing greater flexibility and better economy for those interested in grabbing server or storage capacity from the cloud. It might not be totally replacing traditional forms of infrastructure hosting, but the offerings in this space are now mature enough for prime time, and provide significant advantages in many scenarios.

The delivery of business applications over the wire (or air) on a subscription basis via the Software as a Service (SaaS) model is also now well-established. Whether it is horizontal capability like email, office or information management, or more complex solutions like CRM or ERP, we are again some distance from world-domination, but mainstream use is clearly growing significantly. We are certainly beyond the early adoption stage among larger enterprises.

Meanwhile, the picture with so-called 'Platform as a Service' (PaaS) is a lot less clear-cut. One of the problems here is that it's hard to make generalised statements because of the ambiguity about what exactly PaaS is, and where it fits. Before we can have a sensible conversation about the role of PaaS in the context of mainstream IT delivery, we therefore need to quickly review some of the basic principles and variants.

### The layer in the middle

A quick way of orienting yourself when considering the different categories of cloud services is to use a simplified, abstract version of the traditional systems stack as a reference:



Few would argue with this high level model and the notion that PaaS generally sits between IaaS and SaaS in the stack view of the world. When it comes to the specifics, though, there is still a lot of disagreement.

## **The nature of the beast**

Much of the marketing around PaaS is focused on hosted services, and the notion of providing an application platform in the public cloud. Most of the big service providers tend to promote PaaS in this way, including Microsoft, Salesforce.com, Google and Amazon, for example.

The functionality included varies quite a bit, but the basic proposition is to go beyond simply providing raw server and compute resources (IaaS style) to deliver middleware and management capability as a service too, potentially everything up to but not including the business application itself.

Even then, there is some dispute about where IaaS stops and where PaaS begins. If the service provider makes a fully managed virtual server available to you, for example, including inherent resilience, elastic scalability, and a console for deploying and migrating workloads, you could argue that this represents a pretty sophisticated platform, and some understandably regard this as PaaS. Others, however, say that such a service still doesn't provide everything required to support an application, and assert that true PaaS includes essentials such as app server, web server, and database management capability at a minimum.

Building on this, more comprehensive services also provide things like authentication, access and security frameworks, workflow/BPM engines, application performance management, and even a full-blown set of development tools, APIs and associated libraries. Indeed there are examples of services out there that tick all of the required boxes necessary to provide a self-contained environment in which applications can exist from cradle to grave.

Meanwhile, there is a camp that says the focus on hosted service is a red herring. The discussion that really matters is about architecture, the aim being to define a platform that can be installed in your own datacentre, as well as hosted by a third party. The advantage of this philosophy is that the platform becomes delivery-agnostic, allowing you to mix and match on-premise and hosted deployments depending on business requirements.

Regardless of where you put the emphasis, though, the potential for lock-in exists.

## **A question of commitment**

Application portability is obviously not a new consideration in the IT industry. Those of us who were around in the 1980's, for example, are well aware of both the concept of application platforms, and the work involved in porting an application from one platform to another. While exploiting an integrated Oracle environment based on the SQL\*Forms, the Oracle RDBMS, and associated reporting tools saved a lot of time, the resulting application was pretty much tied into that platform. Porting it to the Informix or Ingres equivalent essentially meant a significant rewrite.

In the current world of PaaS, you are required to commit to a combination of platform components in a similar way. The difference is that there are arguably a lot more moving parts. Back in the 80's, your platform often ran in a single box, e.g. a VAX or some Unix-based mini-computer. When it comes to platforms in today's world of distributed computing, there's usually a whole landscape of servers and services involved.

Of course the whole point of PaaS is to shield developers and application administrators from a lot of what goes on under the covers, but behind the scenes each PaaS implementation is based on a different set of mechanisms for accessing system resources, controlling application execution, reading and writing data, and so on. While industry and de-facto standards can help to isolate applications from some aspects of this, the reality is that each PaaS environment presents a different set of core capabilities, tools, APIs, and associated technical constraints, which in turn impacts the way you design, code, deploy and manage.

The upshot is that an application built for Salesforce's Force.com platform is not going to run on Microsoft Azure, the Google App Engine, or your custom in-house platform without a lot of work.

The trick to navigating through this potential minefield is to focus less on the technology aspects of PaaS, and more on what you are trying to achieve.

## Horses for courses

The most highly publicised PaaS offerings provide a comprehensive environment for developing and operating Web applications in particular. These are great, for example, if you are putting together a customer-facing website for which growth and/or the level of fluctuation in demand is unpredictable. Your application can scale up and down easily and (subject to terms) you pay only for the resources consumed. You also benefit from the provider's communications infrastructure, i.e. you don't have to worry about the mechanics of how users connect with your application.

Not surprisingly, tech start-ups often take advantage of such services, but we are coming across more examples of mainstream enterprises and public sector organisations building on this kind of PaaS too. A good example here is the Microsoft Azure based application that underpins the ['Love Lewisham'](#) programme, in which residents report graffiti, fly-tipping and other public eye-sores by uploading pictures of the offence from their smart phone.

Hosted PaaS also commonly comes in the form of facilities that allow you to extend or customise SaaS applications. The market leading hosted CRM solutions, for example, each provide an environment in which entire new applications can be built and deployed, complete with graphical development tools for defining data structures, and policy-driven forms and workflows, all wrapped up in a tight security framework and subject to version control.

In most cases, such facilities are only available to subscribers of the core SaaS services, but some providers decouple the PaaS environment so enterprises and ISVs can build standalone solutions using the platform. Largely speaking, this kind of PaaS is aimed primarily (though not exclusively) at applications deployed to your workforce rather than the outside world, and provides an interesting alternative to wean people off all of that distributed DIY development (think Excel and Access).

This last point brings us to services that provide PaaS-like facilities, even though we would not traditionally put them into the PaaS category. Hosted collaboration services, e.g. based on SharePoint or similar, can be used to develop and deploy lightweight applications very rapidly by leveraging the embedded data management, forms and workflow capability.

To complete the picture, at the other extreme, some service providers are putting together hosted platforms that take some of the pain and cost out of developing hard core HPC and big data applications. Sometimes these are rolled into broader PaaS environments, e.g. Azure embracing Hadoop, and sometimes they are presented as discrete services within a portfolio.

## Looking forward

Pulling all this together, the hosted PaaS opportunity in a mainstream business environment at the moment largely revolves around discrete tactical application development and deployment. Application portability and interoperability constraints probably mean you should think carefully before betting the farm on any single service or platform architecture at a more strategic level until standards and commercial models mature a bit more. Even then, the horses for courses principle will still apply.

Closing the loop on the argument about whether PaaS is about building a service oriented architecture or delivering platform capability as hosted cloud services, the key is to 'think hybrid'. Perfect portability can only come about if the exact same platform is being run both in-house and in the hosted environment. However some suppliers are getting pretty close by making sure the same (or very similar) APIs and tools can be used in both worlds. Either way, unless you are an ISV, interoperability and consistency of management is likely to be more important to you than portability per se.

Regardless of where you stand on the definitions, principles and practicalities of PaaS, our overriding advice is to think inclusively, and in a service-centric manner. What you deliver in terms of functionality, usability and QoS is always more important than how you deliver it.

## About Freeform Dynamics



Freeform Dynamics is a research and analysis firm. We track and report on the business impact of developments in the IT and communications sectors.

As part of this, we use an innovative research methodology to gather feedback directly from those involved in IT strategy, planning, procurement and implementation. Our output is therefore grounded in real-world practicality for use by mainstream business and IT professionals.

For further information or to subscribe to the Freeform Dynamics free research service, please visit [www.freeformdynamics.com](http://www.freeformdynamics.com) or contact us via [info@freeformdynamics.com](mailto:info@freeformdynamics.com).

### Terms of Use

This document is Copyright 2013 Freeform Dynamics Ltd. It may be freely duplicated and distributed in its entirety on an individual one to one basis, either electronically or in hard copy form. It may not, however, be disassembled or modified in any way as part of the duplication process. Hosting of the document for download on the Web and/or mass distribution of the document by any means is prohibited unless express permission is obtained from Freeform Dynamics Ltd.

This document is provided for your general information and use only. Neither Freeform Dynamics Ltd nor any third parties provide any warranty or guarantee as to the suitability of the information provided within it for any particular purpose.