

Redefining software platforms

*How PaaS changes the game
for ISVs*

September, 2009

Presented by

ProcolluX
ventures

<http://www.pcxvs.com>

Table of contents

A new kind of platform	3
Changing ISV business models	4
New economics.....	4
New go-to-market.....	5
New customer relationships	6
Dynamics of PaaS.....	8
Distinctive features of PaaS as a platform	8
Multi-tenancy.....	9
Web community and APIs	9
Continuous innovation.....	9
Meta-customization.....	9
Service delivery infrastructure	10
PaaS portability	10
Interplay with other assets	11
'Metaplatfrom' assets	11
Legacy assets	11
Ecosystem partner strategies.....	11
Platform plays in the PaaS era	12
Ingredients of classic ISV platforms	12
How PaaS redefines software platforms	13
Considerations for ISVs selecting PaaS	15
Seizing the opportunity	16

About the author



Phil Wainewright is one of the world's foremost authorities on emerging trends in business automation. He is a prolific writer with a ZDNet blog on Software as a Service, an eBizQ blog on The Connected Web, and a series of influential analyst reports to his name. He serves as CEO of Procullux Ventures, a London-based strategic consultancy working with leading business automation vendors and their customers.

Software as Services:
<http://blogs.zdnet.com/SAAS/>

The Connected Web:
<http://www.ebizq.net/blogs/connectedweb/>

Procullux Ventures: www.pcxvs.com

Detailed biography: www.philwainewright.com



Published by Procullux Media Limited for general information, this document presents independent research and analysis compiled by Procullux Ventures in good faith from best available sources. Preparation of this report for publication was funded by Intuit, Inc.

A new kind of platform

The rise of software as a service over the past decade has opened up new opportunities for independent software vendors to develop new applications hosted and delivered via the Web. But until recently, any ISV creating a SaaS offering has had to build its own hosting and service delivery infrastructure. That has all changed in the past two years with the rise of platform-as-a-service. PaaS is the online equivalent of conventional computing platforms, providing a ready-made infrastructure on which an ISV can rapidly build and deliver a SaaS application.

*PaaS
supercharges
SaaS, accelerating
the pace of
innovation and
intensifying
competition*

While many ISVs are understandably wary of binding their fate to that of an emerging platform provider, those who have stepped forward to become early adopters of PaaS have experienced dramatic reductions in development costs and timescales. By lowering barriers to entry and foreshortening time-to-market, PaaS supercharges SaaS, accelerating the pace of innovation and intensifying competition.

The advent of PaaS will change the game for ISVs – not only those who choose to introduce SaaS offerings, but also those who remain wedded to conventionally licensed, customer-operated software products. PaaS changes the competitive landscape across a swathe of parameters:

- **Dramatically faster innovation cycles.** PaaS implements the iterative, continuous improvement upgrade model of SaaS, allowing developers to monitor and respond to customer usage and feedback, and rapidly incorporate the latest functionality into their own applications.
- **Hugely lowered price points.** The shared, pay-as-you-go, elastic infrastructure of PaaS slashes developers' costs across multiple dimensions, resulting in massively reduced development and operations costs.
- **Multiplicity of players from lowered barriers to entry.** The low costs to get started on a PaaS provider's infrastructure attract huge numbers of market entrants who would not be able to fund their own infrastructure, significantly increasing innovation and competition.
- **New business models, propositions, partner channels and routes to market.** The 'as-a-service' business model opens up new ways of offering products and bringing them to market, many of them highly disruptive to established models.

The dynamics of PaaS are different than conventional software platforms

To remain in control of their own destiny, it is imperative that ISVs understand and evaluate what PaaS can offer them. Key to that understanding is to recognize that PaaS is more than just another platform. It's a new *kind* of platform. The dynamics of PaaS are different than conventional software platforms. Developers should beware of assessing PaaS alternatives on criteria that are no longer valid when applied to PaaS. This report maps the landscape and introduces a new framework for evaluating PaaS providers.

Changing ISV business models

It's tempting to dismiss PaaS as a passing fad, but ISVs should not underestimate its impact on the competitive landscape they face. With its powerful combination of cloud infrastructure, the as-a-service business model and the global reach of the Web, PaaS radically transforms three fundamental aspects of the software industry's business models:

- Economics
- Go-to-market
- Customer relations

New economics

With PaaS, the technology infrastructure is already up and running, transforming ISV economics

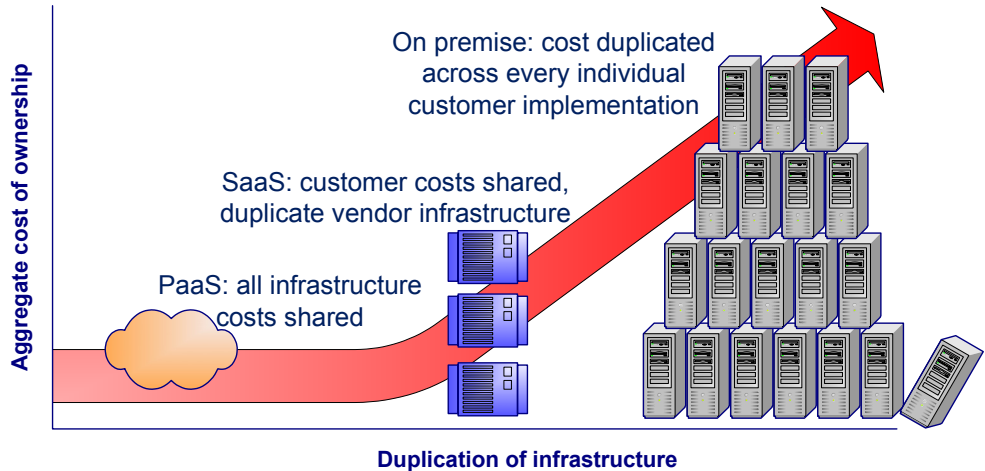
Prior generations of computing platform drove down costs by packaging building-block components into a consistent framework that meant developers could concentrate on application development without first having to create the underlying technology. But developers still had to install and tune the technology, and then support each separate customer implementation. With PaaS, the technology infrastructure is already up and running, and the application is deployed online in a SaaS model, transforming ISV economics.

Lower development costs, with faster returns. Bypassing the need to build, tune and test the underlying infrastructure cuts as much as two years off the development cycle, and reduces implementation timescales by months, so ISVs can start generating revenues faster. Developers on the shared platform benefit from each others' experiences and learning, while instant feedback from customer usage enables continuous improvement of the application.

Changed sales and support models. Online delivery allows high levels of automation and cost savings from web-based sales and support techniques such as self-service administration, social CRM, remote support and web conferencing for demos and sales calls.

Compression of the value chain. The shared infrastructure allows ISVs to operate the applications at a much lower cost than on-premise instances at each individual customer. Squeezing out those duplicated overheads allows providers to collapse their prices without giving up margins.

Fig 1: Sharing infrastructure dramatically cuts software's 'aggregate cost of ownership'



The combined effect of these factors allows developers to deliver finished applications within weeks or months and sell them at significantly lower prices (or higher margins) on a foreshortened sales cycle compared to conventional licensed products. While there are not yet enough metrics available to build a reliable economic model, the anecdotal evidence suggests that proficient developers building on a suitable platform can achieve a tenfold or greater aggregate reduction in costs for typical business applications.

New go-to-market

Developers must come to terms with the as-a-service business model

New computing platforms have traditionally brought software developers into new markets, but PaaS adds the extra dimension of a new business model. Developers not only have to learn how to create software on the platform; they must also come to terms with the as-a-service business model and master emerging web-based sales techniques and channels.

Moving to a pay-as-you-go model. Those accustomed to the upfront, one-time payment model of the packaged software business are rarely prepared for the culture shock of the as-a-service business model. It transforms sales incentive plans, product propositions, billing and collection cycles, customer relationships, cashflows and funding requirements.

Succeeding at trials/freemium, upsells and renewals. By removing the need to install software on the customer's servers, the as-a-service model introduces many new options for incremental selling. Instead of having to sell as much functionality as possible in

a single transaction, a SaaS vendor can sign up prospects to a base level of functionality and then upsell extra capabilities over time. Vendors must work out how to package their application into price points and functionality that make the most of the market opportunity, and learn how to maximize successful trials, 'freemium' seeding, upsells and renewals.

Using web-based sales and service. When an application is being operated and accessed as a web service, buyers tend to have heightened expectations of the vendor's online sales and support capabilities. They expect to find an informative, responsive website, an integrated digital marketing strategy that makes them feel valued as a prospect, and a sales team that's proficient at using web conferencing for sales calls and product demonstrations. Once they become customers, they'll expect responsive, integrated online and offline support, self-service account administration and a proactively nurtured online community for customers and partners.

Leveraging online partnerships. There's a new emerging channel of online partners that SaaS vendors can work with to expand their market reach. They include other SaaS vendors, cloud-savvy IT service companies, providers of online business services and established brands that are extending their reach onto the Web. Making the most of these partnerships means understanding the rules of engagement and business models that work best in the web environment.

The Web extends the reach of ISVs into new geographies and markets but success depends on adapting to the often unfamiliar business dynamics of this new environment.

New customer relationships

PaaS transforms an ISV's relationship with its customers. Instead of a disconnected, 'batch' process – in which a product is created, sold and shipped as a series of discrete, remote acts – delivery as a service is a continuous, ongoing, collaborative interaction with the customer, performed with all the immediacy and responsiveness of the web medium.

Delivery of an operational service, not a toolkit. The old product model of the packaged software industry aims to deliver a multi-purpose toolkit that each customer has to adapt and customize to suit their own technology and business environment. The as-a-service model emphasizes delivery of a working application that is immediately ready to fulfil a business goal, and which can be further refined and customized while it continues to operate. This fosters a culture that's attentive to the business results customers achieve, rather than merely focussing on the breadth of the feature set included in the toolkit.

The as-a-service model emphasizes delivery of a working application that is ready to fulfil a business goal

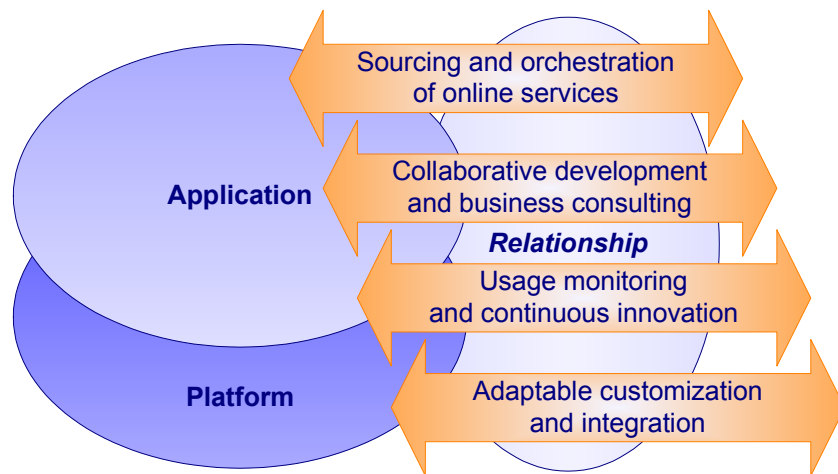
Using rapid, iterative innovation to meet customer business goals.

The always-on connectivity of the Web and the shared infrastructure of the PaaS model allows ISVs to monitor customer usage of the application and respond rapidly to support calls and new feature requests. PaaS extends this by providing a multi-tenant development infrastructure that permits upgradeable customizations, so that customers can work in collaboration with vendors and partners to fine-tune their own implementations of an application to meet individual business needs.

Becoming a trusted advisor/mediator in the on-demand environment.

Every business faces transformative change as a result of the emergence of the Web, and all will look to their online providers as a source of expert advice and assistance in adapting to the challenges of operating online. In addition, application providers are well placed to act as mediators of other online services that interact with or augment their own offerings. ISVs that embrace these business opportunities can move beyond a traditional role as purveyors of software to become trusted providers of an orchestrated portfolio of automated online business processes.

Fig 2: The as-a-service model adds many opportunities for continuous interaction to the vendor-customer relationship.



The impact of PaaS is so much more wide-ranging than earlier computing platforms because it moves application developers up a layer, allowing them to harness the Web itself as an operating platform. As with any platform, their value-add derives from their expertise in understanding how to harness it. But whereas earlier platforms required them to focus on being software engineers, PaaS frees them from worrying about the underlying infrastructure, to concentrate on the business logic and process automation that helps their customers achieve their business goals.

Dynamics of PaaS

Even before the emergence of PaaS, the Web itself has had a significant impact on platform dynamics. Without the Web, it would have proven impossible to organize the degree of global collaboration that has made open source such a successful model for developing infrastructure software. Now open source platforms such as the LAMP stack (Linux-Apache-MySQL-PHP) are serious rivals to long-established proprietary commercial platforms such as Oracle, IBM and Microsoft .NET.

But whereas the open source model taps into the collaborative reach of the Web to share and exchange software code and knowledge, PaaS uses the as-a-service model to make the platform infrastructure itself a shared asset, bringing entirely new dimensions to platform dynamics.

Distinctive features of PaaS as a platform

PaaS is a new breed of platform, distinguished from previous generations by its shared-services model that embodies three separate characteristics of the Web:

*PaaS operates
within the
collaborative global
fabric of the public
Web ... at cloud
scale*

- **Web openness.** PaaS operates within the collaborative global fabric of the public Web, with an API designed to be open to whatever third-party assets and resources may prove useful to platform developers and their clients. Similarly, it draws on the community ethos of the Web, maximizing knowledge sharing and collaboration amongst platform developers.
- **Cloud infrastructure.** PaaS operates at cloud scale, architected so that every individual application shares the entire connected infrastructure, and can draw as much or as little of its resources as are needed from one moment to the next. A single, shared infrastructure ensures that all fixes, upgrades and improvements are immediately available to every platform participant, allowing none to fall behind.
- **As-a-service business model.** Instead of delivering software as if it were a physical product, PaaS provides access to the application infrastructure and charges for measured usage. For ISVs, this pay-as-you-go approach keeps their costs closely aligned with the actual value delivered from day to day.

These characteristics give PaaS several distinctive features, as set out below, that are not found in conventional platforms. Although we can expect to see some feature convergence in the future as

older platforms seek to emulate aspects of PaaS, their on-premise existence cuts them off from the constant evolution that the cloud fosters, condemning them to always follow rather than lead.

Multi-tenancy

PaaS operates as a single, shared resource, in which all participants benefit from every individual test and tweak

Cloud-scale multi-tenancy is the fundamental cornerstone of PaaS, underpinning every other distinctive feature. Conventional platforms are dispersed into a multitude of custom implementations, where each developer's learnings and experiences are confined to their own environment. PaaS operates as a single, shared resource, in which all participants benefit from every individual test and tweak; where enhancements to performance, robustness or security are based on observing aggregate metrics across all use cases; and which presents a single, consistent reference model to which all participants develop.

Web community and APIs

Just like conventional platforms, PaaS defines a consistent API (application programming interface) and nurtures the expertise of a community of developers around it. But whereas a conventional platform vendor is free to define its own API however it pleases, cloud platforms have to respect the wider public Web within which they operate. Their APIs must remain compatible with a global and constantly developing ecosystem of third-party assets and resources, ready to connect on demand to unforeseen services and applications. The community of developers interacts not only with the provider's own APIs but also with many others, and collaborates with peers throughout the Web. This exposure forces PaaS providers to have a much more outward-facing agenda than earlier platform vendors.

Continuous innovation

PaaS adopts the agile, iterative refresh cycle of the SaaS world

In place of the old 18-24 month 'waterfall' cycle of conventional licensed software upgrades, PaaS adopts the agile, iterative refresh cycle of the SaaS world, introducing new features, fixes and enhancements every few weeks, and major functional upgrades at least twice a year. This cycle of continuous innovation is essential to keep pace with the constant evolution of the public Web, where the multiplicity of players ensures that virtually every day produces some new 'must-have' feature – or newly discovered vulnerability.

Meta-customization

Earlier generations of computing platform encouraged developers to extend the core functionality with their own custom software code. Whenever the underlying platform had an upgrade, the custom code would break, requiring time-consuming and costly rewrites. The continuous innovation cycles of PaaS make this totally unworkable, demanding a completely different approach.

In PaaS, the core infrastructure and application objects are protected and all customization is confined to metadata definitions or custom objects that are extensions to the core code, rather than actually changing it. While this requires a new form of self-discipline from developers, who must learn to work within the constraints imposed by each platform, it reduces or eliminates the need to rewrite customizations at each upgrade, protecting the investment that developers make in creating their own intellectual property on the platform.

Service delivery infrastructure

The as-a-service business model requires significant additional infrastructure associated with delivering a service, such as account provisioning, user administration, usage metering and billing, service level monitoring and reporting, sales, support and customer service. Conventional platforms rarely include such infrastructure and it is often omitted from PaaS offerings. It is nevertheless a crucial ingredient for ISV success in the as-a-service world. The most cost-effective means of providing service delivery capabilities is as a multi-tenant service integrated into the PaaS infrastructure.

Service delivery infrastructure is a crucial ingredient for ISV success in the as-a-service world

PaaS portability

With any platform, ISVs inevitably worry about the perils of 'lock-in,' which leaves them dependent on the continued growth and evolution of the underlying platform to protect the commitment and investment they've put into it. Of particular concern with PaaS is their dependence on the continued operation of the platform for their ongoing revenues.

It's not surprising that ISVs look for ways of mitigating that risk. The most obvious route is to select a platform that offers an on-premise alternative, thus giving the option of self-hosting or turning to a third party hoster should something go awry with the provider's own as-a-service offering. Even better, some feel, if the platform is open source, since it is then no longer solely dependent on a single provider for its continued evolution. What ISVs must weigh on the other hand is any dilution of the collective usage, testing and innovation that a single, shared multi-tenant infrastructure continually benefits from. Every separate on-premise instance of a platform weakens the value of that shared body of knowledge.

Every separate on-premise instance weakens the value of that shared body of knowledge

Another way to mitigate risk is to adopt several platforms rather than relying on a single one – in essence, treating the Web itself as the platform and selecting PaaS options that emphasize interaction and federation with other cloud platforms and services. This strategy demands more planning and management, but may provide better long-term risk mitigation than selecting a platform with large numbers of on-premise instances.

Interplay with other assets

Although PaaS operates within the cloud, it must still interact with IT assets that remain on-premise. These fall into two categories.

‘Metaplatfrom’ assets

Client devices, such as desktop PCs and smartphones, aren’t normally regarded as part of the cloud fabric. But as Rubicon’s Michael Mace recently pointed out in his online essay, *Software platforms as a business tool*, cloud-based applications run across both the network and the client, which each form part of a larger whole he termed the ‘metaplatfrom’. Whether or not you agree with this viewpoint, it is certainly the case that PaaS providers are expected to build support for various clients into their platform. In addition to standard browser support, most include toolkits for AJAX clients and often Adobe Flex and/or Microsoft Silverlight. Several also add support for various smartphone client platforms, rather than leaving it to individual developers to engineer their own client libraries.

Legacy assets

No emerging computer platform has ever entirely replaced the previous generation. We are often reminded that mainframes still exist today, performing critical business operations for corporations and governments across the globe. Few businesses are yet ready to give up the significant investments they’ve made in servers and other computing resources that don’t operate in the cloud. It’s up to PaaS providers to offer support for integration to these on-premise resources for as long as they remain in widespread use. In many cases, this is the key towards coaxing customers to take their first steps into the online environment.

It’s up to PaaS providers to support integration to on-premise resources for as long as they remain in widespread use

Support for interactions with each type of on-premise asset, as well as integration to third-party cloud services, is often provided using point-to-point solutions or on-premise adapters. Unfortunately, doing so can easily neutralize many of the advantages of a multi-tenant platform architecture. Where possible, PaaS providers should seek to offer client support and integration services as multi-tenant resources, so that they enjoy the same continuous innovation and meta-customization advantages as the core platform infrastructure.

Ecosystem partner strategies

Market reach has always been a primary consideration when prospective partners are evaluating whether to align with a platform. In a web context, this goes beyond the sheer size of the customer base. Other factors to assess include the provider’s digital and social marketing capabilities, along with its capacity to sustain consistent performance, reliability and continued innovation.

A key dimension is what scope exists to leverage existing customer information assets

ISVs should also evaluate the exposure and contact they will have to potential customers. A key dimension to take into account is what scope exists to leverage existing customer information assets, such as historic business data and core business processes, in applications built on the platform.

When evaluating the effectiveness of the ecosystem itself, the open source movement has set a high benchmark to measure against. Partners will look at the openness of the API and the extent to which the provider is using web collaboration to interact with its developer community and involve them in dialog with customers.

Platform plays in the PaaS era

In computing, the simple Wikipedia definition states that a platform is a computer architecture or software framework that allows software to run. But a successful platform is much more than the technology at its core. The commercial dimension is all-important. To become dominant, an application platform has to:

- Provide a rapid short-cut for developers building a new class of applications
- Support newcomers with help, how-to's, pre-built widgets and infrastructure stacks
- Encourage skill sharing and a sense of community among early adopters
- Offer a ready, fast-growing market for applications and development skills

In short, it must demonstrate ease of adoption, growth momentum and market reach.

Ingredients of classic ISV platforms

Platform-as-a-service incorporates many elements found in classic platforms, whether IBM mainframes, Wintel, Apple, Java or .NET:

- **A deployment platform.** The core component of a platform is a runtime framework and application programming interface (API) that provides a consistent infrastructure to which applications are deployed.
- **Development tools.** A platform provides support for developers in the form of programming languages and related runtime libraries, and sometimes a 'drag and drop' graphical user interface for some development tasks.

- **Platform support.** The platform provider makes expert resources available to help developers take advantage of the platform. Often, 'evangelists' are employed to spread the word about the platform and encourage take-up.
- **A developer ecosystem.** A platform cannot thrive without the enthusiastic support of a community of developers, whose shared skills, experiences and feature extensions enrich its capabilities.
- **A customer base.** Platform providers play a vital role in bringing a customer base to the platform, whether because of their existing standing in the marketplace, or through their efforts to promote and evangelize the platform.

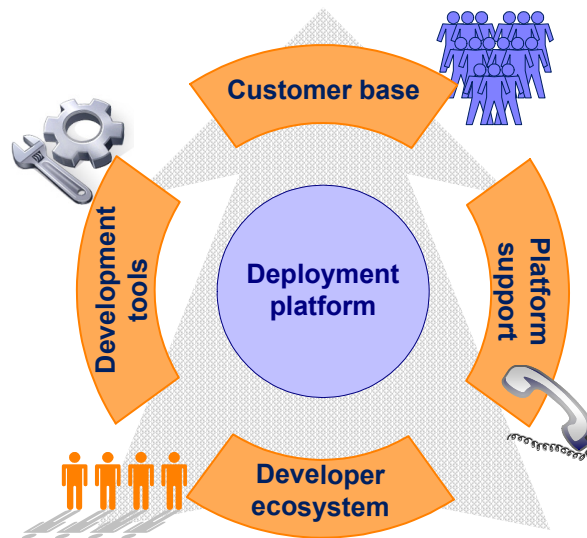


Fig 3: Ingredients of a classic ISV platform include tools, resources, a customer base and the support of developers

The final two elements, although not relevant to the technical definition, have a crucial, symbiotic role in the market success of any platform. Market demand from customers helps encourage developers to adopt the platform, while their efforts to extend the platform's capabilities helps make it more attractive to a broader range of customers. This sets up a 'virtuous cycle' in which developers come to the platform because of its market reach; they extend its feature set; and that in turn brings in a new wave of customers, which sparks the interest of a further tranche of developers, and so on.

How PaaS redefines software platforms

The toolkit and cookbook must cover commercial ingredients as well

PaaS extends the definition of a software platform, adding new, must-have ingredients to the classic recipe. It is no longer enough to give developers a technology toolkit, a solutions cookbook and access to a customer base. PaaS goes beyond technology, introducing an unfamiliar, more complex business model and new

methodologies for development, partner relations, sales and customer service. Meanwhile, the technology itself adds new attributes that support web participation and cloud scale. The toolkit and the cookbook must cover commercial ingredients as well as this broader technology remit, while the platform must cater for both.

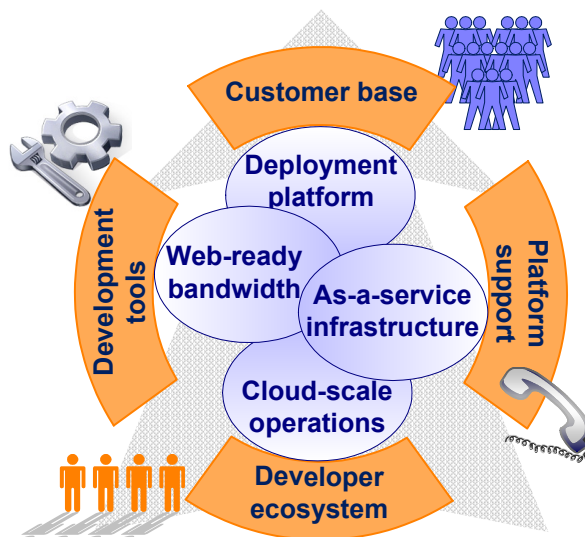
As-a-service infrastructure. Alongside the technology infrastructure that executes the application, PaaS must add the broader business model enablement required to sell, provision and maintain it. This business infrastructure framework needs just as much support, API documentation and developer community interaction as the application infrastructure itself – perhaps more so, since many of the best practices of the as-a-service business model are so much alien territory to those emerging from the conventionally licensed packaged software world.

The platform must remain open and plugged-in to the wider web ecosystem

Web-ready bandwidth. Earlier generations of computing platform sought to 'lock in' partners and customers to the specific technology platform. The Web does not allow PaaS to sit in a walled garden. Although the business model locks partners and customers into day-to-day dependence on their provider's operations, the platform itself must remain open and plugged-in to the wider web ecosystem within which PaaS operates.

Cloud scale operations. A PaaS provider is judged not merely on the excellence of its technology, but also on consistently dependable operation of a high-bandwidth infrastructure. It must operate at cloud scale, and in a fully multi-tenant model that allows developers to continue to take advantage of innovations both to the platform and in the Web at large, while preserving their own intellectual property developed there.

Fig 4: PaaS adds three vital new ingredients to the classic platform mix: web-ready bandwidth, as-a-service infrastructure and cloud-scale operations



Considerations for ISVs selecting PaaS

In this still emerging market, many are not aware of the changed dynamics of PaaS

PaaS options available in the market today don't necessarily reflect the best long-term interests of ISVs. Platform vendors naturally put their own interests first. In this still emerging market, many of them are mistakenly applying principles learnt from previous generations of software platforms, not aware of the changed dynamics of PaaS that have been described in this document.

Similarly, ISVs often approach the choice of platform using criteria that are no longer appropriate to the PaaS environment, or which overlook crucial factors. In particular, it's all too easy to focus exclusively on the development environment itself, or the robustness of the hosting infrastructure, without taking into account how the platform interacts with the broader web context, or what support it provides for pursuing a successful business strategy.

Evaluating providers across all of these parameters is a complex and potentially bewildering task, especially as there is no one-size-fits-all benchmark that matches the circumstances of each and every ISV. To aid decision making, this paper introduces the Procullux ISV PaaS Selection (PIPS) matrix, which identifies three core dimensions on which to evaluate competing platforms when selecting a PaaS provider:

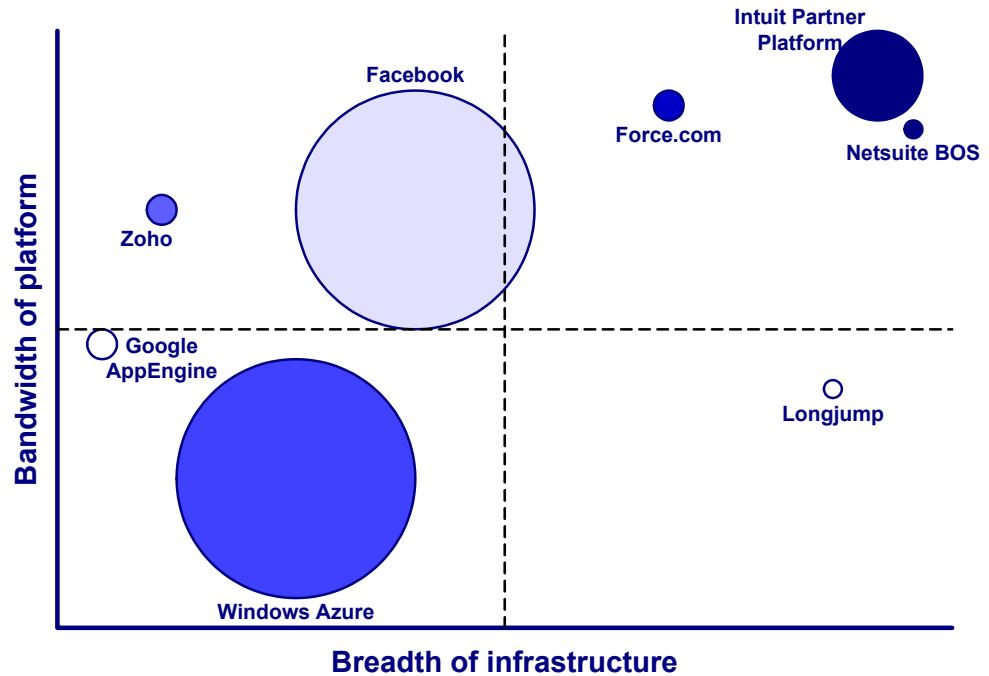
Platform bandwidth. The most PaaS-specific of the three dimensions, this measures the platform's capacity to interact with the web environment, taking into account factors such as the degree of connectivity to other web resources, the openness and adaptability of its APIs, the ability to federate to other platforms, and the extent to which the platform exploits multi-tenancy for communal usage, testing and innovation.

Breadth of infrastructure. This dimension rates the breadth of functionality built into the infrastructure, both as a development platform and as a business platform. To achieve a high rating, a platform must incorporate an extensive service delivery infrastructure.

Customer reach. The final dimension rates the platform's appeal to a defined customer base, taking into account both the size of the installed base and the extent to which the platform leverages existing information assets held by those customers.

The illustration below shows the result of applying the PIPS matrix to eight leading PaaS providers. While the providers in the upper right quadrant offer the fullest PaaS support, each individual ISV may find its best choice falls in a different quadrant, depending on how much they want to leverage factors such as the provider's market reach, its as-a-service infrastructure and connections to third-party services or the availability of on-premise options.

Fig 5: The matrix compares 8 leading PaaS providers. Each circle's size represents the addressable customer base, while the opacity shows how strongly the platform leverages customer information assets. The most connected platforms with the most complete set of development and business tools appear in the upper right quadrant.



Seizing the opportunity

Platform as a service presents huge opportunity for a new generation of independent software vendors. Open to the Web, it gives them more freedom of choice and market reach than earlier software platforms. It lets them focus more than ever before on adding value by offering business expertise in their specialist markets. At the same time, it helps them defend their profitability by driving down development and infrastructure costs.

But like any opportunity, there are risks too. PaaS is relatively new and very little best practice is established. Many providers are feeling their way with as little understanding of the new environment as the ISVs they seek to serve. Understanding why and how it differs from earlier software platforms and working with providers that share an awareness of how to maximize the potential of PaaS is the key to achieving success with applications and services built on these next-generation cloud platforms.