



Technology Strategy and Management

Cloud Computing and SaaS as New Computing Platforms

To become an industry platform, vendors must open their infrastructure technology to other product companies.

THERE IS NO doubt that software as a service (SaaS) as well as the more general infrastructure technology that facilitates this type of software delivery and pricing—cloud computing—are becoming new platforms for enterprise and personal computing.^a They compete with traditional desktop or handheld computers (including smartphones) that run applications directly on the devices. We can see all the platform concepts discussed in my previous *Communications* column (“The Evolution of Platform Thinking,” January 2010): direct and indirect network effects at the ecosystem level as well as firms offering infrastructure in addition to products, open versus closed systems, and conflicts of interest emerging between platform leaders and their complementors.

But SaaS and cloud computing rise to the level of an industry platform only when firms open their technology to other industry players, including complementors and potential competitors, rather than simply using the Web as an alternative delivery and pricing mechanism for what used to be pack-



aged software products. Whether SaaS and the cloud competition are “winner take all” markets (like Microsoft in desktop operating systems or the VHS format in home VCRs) or “winner take most” (such as Google in Internet search) remains to be seen. But we can analyze this question through the lens of platform dynamics.

The ideas of SaaS and cloud comput-

ing have emerged gradually.^b In fact, delivering software applications over a network is an old idea but, in the past,

^b The analysis here draws heavily on R. Bhattacharjee, “An Analysis of the Cloud Computing Platforms” (Cambridge, MA: Unpublished Master’s Thesis in System Design and Management, Massachusetts Institute of Technology, June 2009). Also see B. Hayes, “Cloud Computing,” *Commun. ACM* 51, 7 (July 2008), 9–11.

^a This column is based on portions of my forthcoming book, *Staying Power: Six Enduring Principles for Managing Strategy and Innovation in an Uncertain World* (Oxford University Press, 2010).

has not reached the level of an industry platform. The concept goes back to time-sharing in the 1960s and 1970s, as well as application hosting in the 1980s and 1990s. Then we saw an increasing number of firms in the 1990s and 2000s deliver what used to be packaged software applications from a new platform—the Web—and usually for free. These applications ranged from email to calendars, groupware, online publishing, simple word processing, and many other common consumer and even business applications. Advances in networks as well as virtualization technology have made Web delivery possible regardless of the type of computer the user purchased. But, again, only when vendors open their SaaS or cloud infrastructure technology to other product companies do we have an industry platform.

For example, Salesforce.com created a customer relationship management (CRM) product and configured it not as packaged software but as software delivered over servers and accessed through a browser. When it did this, it created its own in-house platform for delivering the software as a service to its customers. But then it created AppExchange as an open integration platform for other application companies that built products utilizing some features in the Salesforce CRM product. When it did this, Salesforce.com created a new industry platform, or rather a platform wannabe because there are competitors. Salesforce then extended the open platform concept with Force.com, a development and deployment environment using Salesforce's SaaS infrastructure. Amazon (Elastic Compute Cloud, known as EC2) and Google (Google App Engine) also have opened up their SaaS or cloud infrastructures to host outside applications as well as their own productized online services.

By the end of 2008, Amazon was already the most popular general-purpose cloud platform, with over 400,000 developers registered to use its Web services.^c Amazon has become so attractive because it has a rich infrastructure to support online retailing operations and has made these services available to its cloud users—data storage, com-

puting resources, messaging, content management, and billing.

Network Effects

SaaS and cloud computing platforms exhibit *direct* network effects to the extent they have specific application programming interfaces (APIs) or Web services that encourage application developers to tailor their applications or that make it difficult for users of these applications to switch platforms. The direct network effects do not seem as powerful as between Windows and applications written for PCs, or between particular smartphone operating systems like Symbian, Blackberry, or Palm, and applications written for those environments.

The SaaS and cloud programming interfaces and technical standards for exchanging data and logic are usually simple and standardized, relying on the Internet HTTP protocol. But some APIs and Web services are specific to individual SaaS/cloud platforms. For example, many real estate companies or retail shops have built applications that incorporate Google Maps—tying the applications to Google's platform. Other companies have built e-commerce applications using facilities for handling payments provided by Amazon—tying the applications to Amazon's platform.

Cloud and SaaS platforms exhibit *indirect* network effects to the extent that the popularity of one platform over another with developers makes the platform more attractive to other developers or users. As more applications appear on a particular platform, they attract more application developers in a positive feedback loop. SaaS/cloud platform competitors also can try to attract end users by making use of their platforms free, perhaps with funding

The product firms seem to offer SaaS and the cloud as another mode of delivery and pricing.

from advertisements. The application companies generally pay a fee depending on usage, data storage, or some other criteria. Platform vendors can charge high or low fees to attract developers, or make some aspects of their platforms free. This is another version of the “free, but not free” strategy that we have seen in PC and Internet software.

Some firms, such as Salesforce.com (with Force.com and VisulaForce) and Bungee Labs, have taken the SaaS and cloud platform ideas further than just providing an environment to launch applications. They also provide services and program development tools that help companies build new applications within these competing platform environments. Developers can also usually integrate with Web services from other vendors, even though some Web services or APIs may be specific to the platform. This is another version of an “open, but not open,” or “closed, but not closed” strategy.

In short, we see SaaS and cloud platforms appearing in multiple levels: First, we see the general technology of the Internet and virtualization making SaaS technically possible. Then we see companies utilizing this technology to offer SaaS or cloud versions of their products. Finally, we see some firms not only offering SaaS versions of their products (now Web-based services) but opening up their technology to allow other application developers to build and launch applications from these platforms.

SaaS or cloud platforms also appear to be efficient for both users and vendors. Multiple customers can use the same facilities and thereby increase utilization rates for the hardware and the networks. For example, Amazon and Google have enormous data centers that they do not fully utilize. They can launch their own products (automated services) while also hosting applications from other companies, without sacrificing security of the different “tenants.” Hosts such as Amazon, Google, and Salesforce.com generally guarantee security for their hosting customers through detailed service level agreements (SLAs).

Effects for Traditional Software Vendors

On the other hand, there are some negative consequences of the SaaS and cloud platforms for traditional software

^c See <http://gigaom.com/2008/10/09/amazon-cuts-prices-on-s3/>, cited in Bhattacharjee.

product companies and users. Most software product companies today offer Web-based hosted versions of their applications. These in-house SaaS platforms only account for a few percent of sales at big vendors such as SAP, Oracle, or Microsoft, but the demand is rising. SaaS and cloud platforms are especially popular among startup enterprise software companies.^d

The product firms seem to offer SaaS and the cloud as another mode of delivery and pricing. But, in this model, customers usually do not pay separate maintenance and product license fees, which account for about two-thirds of the revenues of the major product firms. Nor do the customers have to deal with customizing versions of the software or migrating patches—which traditionally have generated service revenues for the product companies and account for most of the other one-third of revenues for these firms. That work is all done for them via the SaaS/cloud infrastructure and the one price. There are sometimes issues of performance of applications over a shared, dispersed network. Some enterprise customers are also concerned about security of their data, proprietary knowledge in their applications somehow leaking to competitors, or the SaaS/cloud platform failing.

We also have potential conflicts of interest when application software companies move their products to a SaaS or cloud platform infrastructure for delivery and pricing with their users but also open up their platforms to other application companies whose products are potentially complementary (for example, Salesforce.com). There is less conflict when we have pure infrastructure provisioning from companies that have excess computing capacity on the Web but are not specifically application product vendors (for example, Amazon).

But when a company tries to play both sides of this market, conflicts can occur. For example, Google's App Engine now includes Salesforce.com's APIs as part of its platform. A company can write an application, launch it on Google's App Engine, and use features from Google (such as search or Google Maps) as well as features from Salesforce's CRM prod-

SaaS and the cloud replace some traditional software products but will not eliminate them anytime soon.

uct. However, if Google decides to build its own CRM product, then we have a potential conflict of interest. Salesforce will have to rely on Google to maintain neutrality.

Microsoft, the largest software product company, for years has been preparing (albeit reluctantly) for SaaS and cloud computing as an alternative to traditional packaged software. It created Windows Live and Office Live and has more than a decade of experience with the online MSN network, which delivers content as well as software products and product upgrades. Now, Microsoft has created a cloud platform called Windows Azure that will compete with Amazon and Google. Early indications are that Azure will be relatively neutral to the extent that application developers should be able to use various programming languages and not just the .NET environment. Developers should also be able to incorporate features from other Web services platforms.

But Microsoft is also packaging its own online services and products into Azure and clearly gives preference to its own products and services. For example, everything on Windows Live and Office Live will be available as well as Microsoft SQL Server services, Microsoft CRM services, .NET services, and Sharepoint Services. Microsoft is therefore making it possible for customers to use various Microsoft products as Web services rather than buying the packaged software. Customers should be able to integrate the Microsoft services with products of other vendors, but exactly how open the new platform will be remains to be seen. The other issue for Microsoft is that usage of the Azure cloud will reduce demand for Windows

desktop and servers. Microsoft expects customers as well as some application companies will build applications using the Azure Web services and running the whole system on the Azure platform rather than buying more PCs or servers bundled with Windows.

Conclusion

SaaS and the cloud are clearly new platforms for computing. They replace some traditional software products but will not eliminate them anytime soon. While it is relatively easy for a software product company to create a hosted version of its products, delivering these products over an outside SaaS platform like Amazon, Google, AppExchange, or Windows Azure requires rewriting at least some and maybe most of the code to use the different interfaces and available services.

Although the SaaS/cloud delivery and pricing model has many advantages, it also has disadvantages for product vendors and users. The transition, therefore, is likely to be gradual and partial, as companies create new versions of products that seem well suited to the new platform delivery and pricing models. Similarly, users have many customized applications and data stored in proprietary databases. They would all have trouble switching to a SaaS/cloud platform quickly but surely can do so gradually if the economics make more sense.

Finally, as long as the SaaS/cloud vendors maintain some differentiation among their platform offerings, direct network efforts are not too powerful, and switching is not too difficult or expensive for application developers or users, then we will probably continue to see multiple SaaS/cloud platforms coexist.^e This is what we have experienced in video game consoles (with Sony, Microsoft, and Nintendo) and smartphone operating systems and handsets (with RIM, Apple, Nokia, Palm, Microsoft, and Google). ■

e For the conditions that make and do not make "winner take all" markets, see T. Eisenmann, G. Parker, and M.W. Van Alstyne, "Strategies for Two-Sided Markets," *Harvard Business Review* 84, 10 (Oct. 2006), 92–101.

Michael Cusumano (cusumano@mit.edu) is Sloan Management Review Distinguished Professor of Management and Engineering Systems at the MIT Sloan School of Management and School of Engineering in Cambridge, MA.

Copyright held by author.

d M.A. Cusumano, "The Changing Software Business: Moving from Products to Services," *IEEE Computer* 41, 1 (Jan. 2008), 20–27.