

## The business value of SaaS

### The marketplace

It's hard to miss: the term *Software as a Service* (SaaS) is everywhere, garnering headlines in the technology and business press, leading to hot IPO's raved about in the financial news, and generally being described as the future of software technology.

With this kind of press, virtually every major software vendor is touting some software application built upon SaaS principles, leading to confusion about the solution model and its advantages. This overview will clearly define the term and describe the short-term and long-term benefits of the solution model and its variants.

### SaaS defined

Simply defined, SaaS is an application delivery model in which the user accesses software over the Internet, from anywhere, at any time. (This is why the solution model is also called "on-demand" by some providers.)

The service provider, not the enterprise using the software, has the ownership of the physical location, hardware assets, and maintenance of the system.

SaaS is usually sold via subscription-based pricing, rather than the "perpetual license" that normally accompanies traditional on-premise software. In true SaaS applications, all users run off a robust infrastructure and a single instance of the software – described in more detail on the next page.

### Software as a Service advantages

There are several obvious financial and resource usage advantages to the SaaS model, as compared to the traditional on-premises software model:

- > **Initial costs:** With the subscription pricing model, the up-front investment in software licenses, operating system licenses, databases, servers, backup equipment, backup power, etc., is eliminated. The service provider bears all of these costs, which are shared across the customer base, and passed along in the subscription pricing.
- > **Operations:** Since the service provider is operating the system, low-value, but mission-critical IT operations are eliminated or reduced: nightly backups, tuning databases, defragging storage systems, applying security patches, etc., are all eliminated.

With SaaS business applications, you no longer need to spend money on costly software upgrades, infrastructure, software and hardware maintenance, and expensive upgrade consulting services.

Initial implementation services are also less than the traditional approach, since you have access to the software when the project starts. No more waiting around for production, development, and test environments to be staged. SaaS vendors already have these environments ready to go.

### A history of SaaS

#### Mid to late 90's:

"Utility computing" becomes the theoretical darling of the technology press. Harking back to the old mainframe "time-sharing" model, this approach to corporate computing is based on the idea that "CPU cycles" are now a commodity, with storage and bandwidth rapidly approaching that mark.

#### 1999-2000:

"Application service providers" or ASPs, become the new darling. In many cases, these were companies that took existing client/server applications and delivered them as services over a VPN or the Internet.

## The technology of SaaS

### The importance of multi-tenancy and agile development

True SaaS applications must be delivered from a “multi-tenant” system, which means that there is a single instance of the software running, and multiple companies use this system as if it were dedicated to their own use.

More importantly, only multi-tenant systems can be combined with new software development models to deliver the long-term advantages of the SaaS model.

The most advanced SaaS providers do away with the traditional version release schedule (version 1.0, 1.1, 2.0, etc.) in favor of an agile development methodology, or “rapid application development” methodology. Software providers that use these new software development techniques allow their customers NEVER to be faced with an upgrade (i.e., they are always using the most current version of the software). Even more importantly, these new features and customizations get automatically folded into the base application, so they are no longer “customizations.”

This combines the operational and financial advantages of SaaS with an entirely new set of advantages, most of them focused on the end users:

- > All users are on the same software, and it’s always the most current version.
- > Since custom feature requests are folded into the (one) base application, they are carried forward as standard features. (Note: typically these new features are surrounded by configuration settings that turn off the new feature by default; it’s only when a user asks for this new feature that it is enabled for a new company.)
- > The end result of this is that the underlying feature-set of a multi-tenant SaaS model gets better every day, and each new deployment of the software dramatically increases the features and functions of the software. This is in direct contrast to more traditional enterprise software models, where companies get frozen onto an obsolete release because of the heavy customization required to implement it in the first place.
- > Crucial technology personnel and resources can now focus on the important aspects of running the business: process improvement, business analytics, and customer service.

Component	On premise (traditional)	SaaS
<b>Pricing model</b>	Perpetual software license (purchase)	Monthly subscription
<b>Hardware</b>	Customer purchases	Included
<b>OS licenses</b>	Customer purchases	Included
<b>Software licenses</b>	Customer purchases	Included
<b>User access</b>	Client software and/or web browser	Web browser
<b>Maintenance fees</b>	18-21% of software cost	Included
<b>IT operations (backups, patches, security, monitoring)</b>	Customer must perform/ sub-contract	Included
<b>Upgrade costs (new infrastructure, new patches to software)</b>	Customer must pay (required every 18-36 months)	Included

### A history of SaaS

#### 2003:

In the IT downturn of 2001-2003, ASP becomes a dirty word - many have gone out of business, leaving companies without their application or data. The new word is now “hosted.” Although technically, this term simply means that the server is sitting in a third party data center and is managed by someone else, it picks up steam and becomes widely associated with all variations on the software as a service model.

#### 2004:

Salesforce.com goes public on NYSE, with its slogan “No Software.” The financial media picks up the drumbeat, and the term “Software as a Service” gains widespread acceptance in the industry.

## SaaS is not the same as hosting

As was stated earlier, virtually every software company in the market today is offering some type of service and calling it SaaS. However, a closer look at these offerings demonstrate that they do not offer the same advantages of a multi-tenant SaaS solution (discussed previously) and, in many cases, have some of the same disadvantages of the legacy on-premises model.

The most common type of service that is mistakenly sold as SaaS is actually a hosted application. Application hosting is a widely available service that is useful in many circumstances. In a hosted application, a data center company provides a dedicated server and related hardware, network connectivity, rack space, power and related services, and sometimes offers technology management services.

In contrast to the multi-tenant SaaS solution model, this is a dedicated set of hardware that is being hosted and managed by a third party. The same IT operations are required as with an on-premises model (backups, applying patches, etc.), but these are now performed by a third party in an outsourced fashion rather than an internal IT department.

Application hosting may be advantageous for a company with limited IT resources, but hosting does not offer any of the long-term advantages of a multi-tenant SaaS solution. Because three parties are involved (customer, software vendor, hosting company), the costs are often higher than running a system internally – though often with guaranteed uptime and high quality IT services. Moreover, because each hosted application is an island unto itself, the model suffers the same challenges surrounding upgrades and customizations as the on-premises model.

For more information or any questions you might have on SaaS, please contact a member of our consulting team at [bakertilly.com](http://bakertilly.com) or 800 362 7301.

## A history of SaaS 2005-2006:

Software as a Service solutions gain a strong foothold, with the conventional wisdom saying that the model best fits small and mid-sized companies. However, larger organizations are utilizing SaaS in surprisingly large numbers. A new generation of software-as-a-service companies begins to emerge, combining new software development methodologies (e.g., agile development, rapid application development) with the SaaS delivery model.