

## Software as a service

From Wikipedia, the free encyclopedia

**Software as a service** (**SaaS**, typically pronounced [sæs]), sometimes referred to as "software on demand," is software that is deployed over the internet and/or is deployed to run behind a firewall on a local area network or personal computer. With SaaS, a provider licenses an application to customers either as a service on demand, through a subscription, in a "pay-as-you-go" model, or (increasingly) at no charge when there is opportunity to generate revenue from streams other than the user, such as from advertisement or user list sales. This approach to application delivery is part of the utility computing model where all of the technology is in the "cloud" accessed over the Internet as a service.

### Advantages

- Accessible from anywhere with an internet connection
- No local server installation
- Pay per use or subscription based payment methods
- Rapid scalability
- System maintenance (backup, updates, security, etc) often included in service
- Possible security improvements, although users with high security requirements (e.g., large corporations) may find SaaS a security concern
- Reliability

SaaS has become a common model for many business applications including accounting, collaboration, customer relationship management (CRM), enterprise resource planning (ERP), invoicing, human resource management (HRM), content management (CM), and service desk management.<sup>[1][2]</sup>

### Contents

- 1 History
- 2 Philosophy
- 3 Key characteristics
- 4 Benefits
- 5 Implementation
- 6 SaaS and SOA
- 7 Adoption
  - 7.1 Drivers
  - 7.2 Sales channels
  - 7.3 Pricing models
  - 7.4 User satisfaction
- 8 Criticism
- 9 See also
- 10 References

### History

Software as a service's acronym, SaaS, first appears in an article called "Strategic Backgrounder: Software as a Service." It was published in February 2001 by the Software & Information Industry's (SIIA) eBusiness Division. This 18 page document is one of the most complete essays pertaining to SaaS available today. SIIA developed the backgrounder to analyze the current state of the SaaS market and its near term prospects, and to provide insight for its members who may be profoundly impacted by changes implied in the SaaS mode. Software as a service is essentially an extension of the idea of the Application Service Provider (ASP) model.<sup>[3]</sup>

One of the first SaaS applications was SiteEasy, a web-site-in-a-box for small businesses, that launched in 1998 at Siteeasy.com. Developed by Atlanta-based firm WebTransit (now defunct) by Gary Troutman and Drew Wilkins, SiteEasy was sold on a subscription basis to its first customer in the Fall, 1998.<sup>[4]</sup>

Loudcloud, founded in 1999 by Marc Andreessen, was one of the first to attempt to commercialize Software as a Service computing with an Infrastructure as a Service model.<sup>[5]</sup> The technology also was called "ASP's" or Application Service Providers, under the terminology of the day. Loudcloud changed its name to Opsware after selling the operations side of the business to EDS in 2002, and HP acquired Opsware in 2007<sup>[6]</sup> and EDS in 2008.<sup>[7]</sup> HP Software Division now offers the SaaS originally developed by Loudcloud as HP Software-as-a-Service.

## Philosophy

Software and business professionals generally associate the term SaaS with business software, and as a possibly lower-cost way for businesses to use software as needed rather than license every application on every device. With a well-designed implementation and properly priced licenses, on-demand SaaS provides license benefits without the associated complexity and the potential high cost to equip devices with applications they may not need.

Though many Unix applications already work using the SaaS model, EULA applications did not have this flexibility before SaaS. A licensed copy of a word processor, for example, had to reside on each machine in order to create a document on that machine. The equipped program has no intrinsic value loaded on a computer that is turned off for the night. The same employee would need another fully paid license to write or edit a report at home on their own computer, while the work license is inoperative.

## Key characteristics

SaaS characteristics include:<sup>[8]</sup>

- Network-based access to, and management of, commercially available software
- Activities managed from central locations rather than at each customer's site, enabling customers to access applications remotely via the Web
- Application delivery typically closer to a one-to-many model (single instance, multi-tenant architecture) than to a one-to-one model, including architecture, pricing, partnering, and management characteristics
- Centralized feature updating, which obviates the need for end-users to download patches and upgrades.
- Frequent integration into a larger network of communicating software—either as part of a mashup or a plugin to a platform as a service

(Service oriented architecture is naturally more complex than traditional models of software deployment.)

SaaS providers generally price applications on a per-user basis and/or per business basis, sometimes with a relatively small minimum number of users and often with additional fees for extra bandwidth and storage. SaaS revenue streams to the vendor are therefore lower initially than traditional software license fees, but are also recurring, and therefore viewed as more predictable, much like maintenance fees for licensed software.

Some SaaS applications are free to the user, with revenue being derived from alternate sources such as advertising, or upgrade fees for enhanced functionality (often referred to as "freemium"). Examples of free SaaS applications include large players such as Gmail and Google Docs, as well as smaller providers like Wave Accounting (free accounting) and Freshbooks (freemium time tracking and invoicing).

In addition to characteristics mentioned above, SaaS sometimes provides:

- More feature requests from users, since there is frequently no marginal cost for requesting new features<sup>[citation needed]</sup>
- Faster new feature releases, since the entire community of users benefits<sup>[citation needed]</sup>
- Embodiment of recognized best practices, since the user community drives the software publisher to support best practice<sup>[citation needed]</sup>

## Benefits

Some of the claimed benefits for SaaS are:

- Capital expenditure is reduced by not having to purchase servers or full copies of software. This is counterbalanced by the increased revenue cost of paying for the use of the SaaS
- Faster implementation. In some cases the customer can deploy SaaS more quickly as no local installation is required.
- Depending on the user, it may remove a non-core activity (deployment and support of the software and its associated infrastructure) freeing up time to focus on core business activities
- Reduced need to predict scale of demand and infrastructure investment up front
- Possible improvements to reliability if the SaaS provider's infrastructure is more redundant or has higher availability than the user would otherwise have

## Implementation

One of the companies that sells that kind of service classifies SaaS into four "maturity levels," whose key attributes are [configurability](#), [multi-tenant](#) efficiency, and [scalability](#).<sup>[9]</sup> Each level is distinguished from the previous one by the addition of one of those three attributes:

- Level 1 - Ad-hoc/custom: Each customer has a customized version of the hosted application that runs as its own instance on the host's servers. Migrating a traditional non-networked or [client-server](#) application to this level of SaaS typically requires the least development effort, and reduces operating costs by consolidating server hardware and administration.
- Level 2 - Configurable: This adds greater program flexibility through configurable [metadata](#), so many customers use separate instances of the same application code. This lets the vendor meet different customer needs through detailed configuration options, while simplifying common [code base](#) maintenance and updating.
- Level 3 - Configurable, multi-tenant-efficient: This adds multi-tenancy to the second level, so a single program instance serves all customers. This enables more efficient server resource use without apparent difference to the end user, but ultimately faces scalability limits.
- Level 4 - Scalable, configurable, multi-tenant-efficient: The fourth and final SaaS maturity level adds scalability through a [multitier architecture](#) that supports a [load-balanced](#) farm of identical application instances that run on a variable number of servers. The provider can adjust system capacity to match demand by adding or removing servers without further altering the [software architecture](#).

SaaS architectures may also use virtualization, either in addition to multi-tenancy, or in place of it.<sup>[10]</sup> A principal virtualization benefit is that it can increase system capacity without additional programming. On the other hand, much programming may be required to construct a more efficient multi-tenant application. Combining multi-tenancy and virtualization provides still greater flexibility to tune the system for optimal performance.<sup>[11]</sup> In addition to full [operating system-level virtualization](#), other virtualization techniques applied to SaaS include [application virtualization](#) and [virtual appliances](#).

SaaS application development may use various types of [software components](#) and [frameworks](#). These tools can reduce time-to-market and the cost of converting a traditional [on-premise software](#) product or building and deploying a new SaaS solution. Examples include components for subscription management, [grid computing](#) software, [web application frameworks](#), and complete SaaS platform products.<sup>[12]</sup>

## SaaS and SOA

Much like other software, SaaS can also take advantage of [Service Oriented Architecture](#) to let software applications communicate with each other. Each software service can act as a service provider, exposing its functionality to other applications via public brokers, and can also act as a service requester, incorporating data and functionality from

other services. [Enterprise Resource Planning](#) (ERP) Software providers leverage SOA in building their SaaS offerings.

Software as a secure service (SaSS) is a variation of SaaS that provides security in the link to the service, content storage on the service, and non-proprietary format data backups and restores of data stored on the service.<sup>[citation needed]</sup>

## Adoption

### Drivers

A traditional rationale for outsourcing IT systems involves applying [economies of scale](#) to application operation, i.e., an outside service provider can offer better, cheaper, more reliable applications. SaaS-based application use has grown dramatically. A [Gartner](#) survey in July 2009 found that customers are "somewhat satisfied".<sup>[13]</sup> Several important changes to the way people work have facilitated this rapid acceptance:

- Fast, low-cost broadband is available.<sup>[14]</sup>
- Computers have become widespread—most information workers have at least basic computer skills.
- Computing has become a commodity. In the past, corporate mainframes were jealously guarded as strategic advantages. More recently, applications were viewed as strategic. Today, people know it's the business processes and the data itself (customer records, workflows, pricing information) that matters. Computing and application licenses are cost centers, and as such, they're suitable for cost reduction and outsourcing. The adoption of SaaS could also drive Internet-scale to become a commodity.
- Insourcing IT systems requires expensive overhead including salaries, health care, liability, and physical building space.
- Applications have tended to standardize. With notable, industry-specific exceptions, most people spend most of their time using standardized applications.<sup>[citation needed]</sup> An expense-reporting page, an applicant screening tool, a spreadsheet, or an e-mail system are all sufficiently ubiquitous and well understood that most users can switch from one system to another easily. This is evident from the number of web-based calendaring, spreadsheet, and e-mail systems that have emerged in recent years.
- Parametric applications are usable. In older applications, one could often only change a workflow by modifying the code. In more recent applications, particularly web-based ones, significantly new applications can be created from parameters and macros. This allows organizations to create different kinds of business logic on a common application platform. Many SaaS providers allow a wide range of customization within a basic set of functions.
- A specialized software provider can target global markets. A company that made [software for human resource management](#) at boutique hotels might once have had a hard time finding enough of a market to sell its applications. But a hosted application can instantly reach the entire market, making specialization within a vertical market not only possible, but preferable. This in turn means SaaS providers can often deliver products that meet specific market needs better than traditional "shrinkwrap" applications.
- Web systems demonstrate reliability. Despite sporadic outages and slow-downs, most people are willing to use the public Internet, the Hypertext Transfer Protocol and the TCP/IP stack to deliver business functions to end users.
- Security is sufficiently well trusted and transparent. With the broad adoption of [SSL](#), organizations have a way of reaching their applications without the complexity and burden of end-user configurations or VPNs.
- Enablement technology (tools, libraries, etc.) is available. According to IDC,<sup>[8]</sup> organizations developing enablement technology that allow other vendors to quickly build SaaS applications will play an important role in driving the adoption of SaaS. Because of SaaS' relative infancy, many companies have either built enablement tools or platforms or are in the process of engineering enablement tools or platforms. A Saugatuck study shows the industry will most likely converge to three or four enablers that will act as [SaaS Integration](#) Platforms (SIPs).<sup>[15]</sup>
- Wide-area network bandwidth has grown drastically, following Moore's Law (more than 100% increase each 24 months), and is about to reach slow local networks bandwidths. Added to network quality improvement, this has driven people and companies to trustfully access remote locations and applications with low latencies and acceptable speeds.
- SaaS has "democratized" software, allowing small and medium businesses to access functionality formerly the domain of large enterprises. Many analytical software tools have been released as SaaS applications on a monthly subscription basis.

- SaaS facilitates data aggregation. Instead of collecting data from multiple data sources with different database schemas, all data for all customers is stored in a single database schema (i.e., multi-tenant). This simplifies running queries across customers, mining data, and looking for trends.
- The rise of third-party [SaaS data escrow](#) services has reduced some security concerns by allowing application data to be held with an independent third party.

## Sales channels

With products focus on the [mid market](#), direct selling can become an expensive undertaking. SaaS companies seek alternatives by selling through [value-added resellers](#) (VARs), [Managed Service Providers](#)(MSPs), [Master Managed Service Providers](#) (MMSPs), and similar alliance partners. However, since SaaS is not only a different delivery mechanism, but a different business model and different technology, selling through channels has its own challenges.

## Pricing models

SaaS applications provide the opportunity to implement pricing models that establish and maintain recurring revenue streams. Most SaaS vendors charge a monthly hosting or subscription fee. Opportunities also exist to charge per transaction, event, or other unit of value. These alternative pricing models exist because customers "lease" the software from the vendors and the vendors can view all transactional activity.

## User satisfaction

Gartner's 2008 survey of 333 enterprises in the US and UK found a low level of approval from customers, describing overall satisfaction levels as "lukewarm." Respondents who decided against SaaS cited high service cost, integration difficulty, and technical requirements.<sup>[13]</sup> A recent report from Forrester, "The ROI of Software-As-A-Service," examined a range of companies that chose SaaS solutions and found that SaaS does result in long-term value. Companies interviewed for the report cited several reasons for their ROI of SaaS:

- Rapid deployment
- Increased user adoption
- Reduced support needs
- Lower implementation and upgrade costs

## Criticism

[Richard Stallman](#) strongly criticizes SaaS. According to Stallman, using SaaS can cause as much harm as [proprietary software](#), since users can't modify the particular software they use, thus, they can't control their own computing.<sup>[16]</sup>

## See also

- [Application service provider](#)
- [Cloud computing](#)
- [Computer user satisfaction](#)
- [Mobile enterprise](#)
- [Multitenancy](#)
- [O3Spaces](#)
- [Online office suite](#)
- [Platform as a service](#)
- [Service bureau](#)
- [Software plus services](#)
- [Supply chain](#)
- [Twinfield](#)
- [Utility computing](#)

## References

1. <sup>^</sup> Biddick, Michael (Jan. 16, 2010). ["Why You Need A SaaS Strategy"](#). Retrieved 2010-03-10.
2. <sup>^</sup> [SaaS Cloud Computing Vendors](#)
3. <sup>^</sup> [Software as a Service: Strategic Backgrounder](#), Software & Information Industry Association, Feb 2001, retrieved May 2010
4. <sup>^</sup> Archive.org crawled site in 1999. <http://web.archive.org/web/19991007185122/http://siteeasy.com/>
5. <sup>^</sup> [Computeractive.co.uk](#)
6. <sup>^</sup> [Crunchbase.com](#). "Opsware."
7. <sup>^</sup> ["HP Closes EDS Acquisition" news release](#)
8. <sup>^</sup> <sup>a</sup> <sup>b</sup> Traudt, Erin; Amy Konary (June 2005). "2005 Software as a Service Taxonomy and Research Guide". IDC. pp. 7.
9. <sup>^</sup> ["Architecture strategies for catching the long tail"](#). April 2006. Retrieved 2008-05-24.
10. <sup>^</sup> Wainwright, Phil (October 2007). ["Workstream prefers virtualization to multi-tenancy"](#). Retrieved 2008-05-24.
11. <sup>^</sup> Chong, Fred (October 2006). ["Multi-tenancy and Virtualization"](#). Retrieved 2008-05-24.
12. <sup>^</sup> Schuller, Sinclair (March 2007). ["Repealing the SaaS Tax"](#). Retrieved 2008-05-24.
13. <sup>^</sup> <sup>a</sup> <sup>b</sup> [Gartner Survey Shows Many Users are Underwhelmed by Their Experiences of SaaS](#), Gartner.com, 2009-07-08. Retrieved 2009-09-16
14. <sup>^</sup> [Institute of Directors - Retrieved 8 April 2010](#)
15. <sup>^</sup> ["SaaS 2.0: Saugatuck Study Shows Rapid SaaS Evolution to Business Platforms"](#). April 2006. Archived from [the original](#) on October 29, 2006. Retrieved 2006-09-01.
16. <sup>^</sup> [Who does that server really serve? Retrieved 2010-04-03.](#)

Retrieved from "[http://en.wikipedia.org/wiki/Software\\_as\\_a\\_service](http://en.wikipedia.org/wiki/Software_as_a_service)"

Categories: [Business models](#) | [Cloud applications](#) | [Software distribution](#) | [Services management and marketing](#) | [Service-oriented \(business computing\)](#) | [Software industry](#) | [Cloud computing](#)

- This page was last modified on 27 January 2011 at 19:55.
- Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. See [Terms of Use](#) for details.  
Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.