



Global Knowledge®

Expert Reference Series of White Papers

What to Consider When Implementing Software as a Service (SaaS)

What to Consider When Implementing Software as a Service (SaaS)

John Hales, Global Knowledge VMware, SDN, and SoftLayer instructor, A+, Network+, CTT+, MCSE, MCDBA, MOUS, VCP, VCAP, VCI, EMCSA

Introduction

When deciding to deploy your Software as a Service (SaaS) cloud solution you need to decide if you will be a provider of SaaS services, which method to use, and/or if you will be an end user to an existing solution provided by another vendor. In this white paper will first examine your options as a SaaS provider, which include Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).

With deploying SaaS, your customers don't choose any of the server, storage, or networking specifics, and they don't choose an operating system, or even a language to develop new tools in. Rather, they get an application, such as email, a customer relationship management (CRM) application, or a word processor. They can't choose anything other than the settings allowed by the application, possibly including things like the amount of storage desired, default font, or things of an application nature.

You, as the provider, however, will need to consider all these issues, and many more, in developing your application. In order to provide a simple experience for your customers you will need to think about providing all the tools end users will need to use your application. This is a lot of work – probably more so than deploying an existing application on IaaS or PaaS.

In this white paper, you will also receive some guidance around being a SaaS consumer for various aspects of your IT infrastructure needs. We'll talk about the reasons why organizations use various SaaS solutions. We'll also give some examples of common SaaS solutions to get you started, as we have with the [IaaS](#) and [PaaS](#) white papers in this series. Before we get started, though, let's briefly review SaaS cloud computing at a high level to set the stage for our discussions.

When you become a SaaS consumer you'll expect the same level of ease of use, self-service support, etc., that you will want to provide your own customers. Keep in mind that it's possible that you may be both a provider and a consumer, such as providing an email platform and using a SaaS provider for your CRM needs. This is the most familiar scenario to most people today, even if they don't know these applications are SaaS applications. Things like Gmail and MapQuest are ubiquitous today.

Note that SaaS applications can be accessed via a browser where the great majority of the processing is done on the cloud provider's infrastructure, or via an application, such as one that runs on Android or iOS, or even a Windows or PC application, where the processing is shared between the cloud provider's infrastructure and the consumer's device. In either case, data is typically stored on, or at least synchronized with, the cloud provider's infrastructure.

SaaS is at the top of the pyramid of aaS choices, where the customer gets to make very few choices beyond the application settings such as font, text and background color, and content to cache locally as previously mentioned. Be sure to read the terms and conditions and service level agreements (SLAs) provided by the cloud provider to ensure that your needs are met, especially with regard to any backups, uptime guarantees, etc.

The advantage of SaaS from a consumer perspective is that you don't need to worry about any of those decisions — you just use the application and let the provider handle all the details. The downside is that you have no control over most of the configuration, performance, or reliability of your application. You'll need to trust (hopefully, after carefully reviewing the SLA) that your provider has all those issues covered. You'll also need to consider how to get all of your corporate data from the existing application (if any) into the new SaaS application, and more importantly, how it can be pulled back out of the provider's system if the provider does not live up to their promises and/or they go out of business. Remember, you're betting the success of your business and all of your corporate intellectual property on the provider and its security controls, backups, etc., so choose carefully.

In this model, all the underlying infrastructure may be shared across companies, with very little, if any, control over whom the infrastructure is being shared with. Your intellectual property (IP) may exist next to a competitor's, and you are trusting the provider to keep both sets of IP segregated. Now, let's get started looking at the issues and considerations around becoming a SaaS provider yourself.

Five Additional SaaS Implementation Factors

When deciding to deploy your SaaS solutions on either an IaaS or PaaS cloud, there are additional factors you need to consider.

Application Access

Choose how your customers will access your application. This is perhaps the most important factor you'll need to consider.

- If the application is browser-based, will you have various versions of your website that are optimized for mobile and desktop/laptop scenarios?
- How will you keep the content synchronized between the various versions?
- Even in this "simple" scenario, you'll need to decide what browsers to support. Internet Explorer? Safari? Firefox? Chrome? Opera? Others?
- Will the application be HTML5-based or leverage Flash, Java, or other client-side and/or server-side technologies?

You may choose to deploy it as an application instead. In that case, you'll need to consider how your target audience will use the application.

- Will it be on a desktop or a laptop?
- If so, what platform(s) will you support? Mac OS X? Windows? Linux?

Maybe your application will be phone or tablet based instead. Again, you'll have choices to make from a platform perspective. Will your app run on iOS? Windows? Android? Maybe your application will be available on all of the above mobile, desktop, and web-based platforms. The more you support, the broader the base of customers you'll reach, but the greater the complexity in creating and maintain them all and the more staff will be required to make it appear seamless.

Backing up Data

Determine how to back up your customers' data. It is virtually guaranteed that your customers will assume that you are protecting their data. You would be very upset if your Gmail or Hotmail emails were lost or your OneDrive account had files lost. You will need to devise a backup strategy.

Service Outages

Devise a high-availability strategy to handle any outages your provider may experience as well as any outages your own platform may run into. You expect 24/7 access to your data and so will your customers. They won't care that an Internet cable was cut or an earthquake occurred. Some of this may be handled by deploying applications that run locally and then synchronizing the data with your infrastructure, allowing offline access.

Platform Performance

Consider the performance of your platform. Most people will be used to experiencing whatever your app does locally and will expect a similar level of performance from you. Remember, it's not just your app and what it needs to do from a computing-and-storage perspective, but network latency end-to-end. In other words, from the customer to your app and back. You will need to tune and optimize your app locally (i.e., in the data centers you choose to deploy your application) as well as consider where to deploy it.

In general, the closer the proximity to your users, the lower the latency and the better the experience. This may mean that you need to deploy your app in multiple locations around the globe and leverage global load balancers to direct your customers to the closest site. But in doing so, remember any data sovereignty laws, privacy rules, etc., and consider how that impacts your design in this regard, as well as the previous two issues we described (backup and service outages). Determining how to synchronize your data across data centers if you deploy your application infrastructure to more than one is key. One other thing to consider again is the use of local applications (mobile and/or desktop based) to reduce the dependency of your infrastructure always being available at a very low latency.

Security

Lastly, remember that security is critical. If people don't feel that you are protecting their data, Intellectual Property, privacy, etc., you are unlikely to get and keep many customers. You will need to consider the security of data at rest (on your storage platform) as well as in transit (crossing the network, both internally within your app as well as externally between your customer and your app). This extends to any data that is stored on the end user's device (if any). You may have little to no control over that, or it may be managed by others, but you'll need to consider those implications from the start.

Becoming an SaaS Provider on a PaaS Cloud

There are a few unique considerations to PaaS cloud solutions that you'll need to consider as well. Here are a few:

- The most obvious consideration is the platform you wish to use to develop your app compared with the offerings of the vendor(s) you're considering.
- Scalability will be key as you don't know how fast your application will be adopted and the growth rate; you want a provider that can scale quickly to maximize performance as well as to scale back when you don't to minimize costs.
- What is the availability of the SLA for the platform you'll be using? The best app in the world is of no use if you can't access it.

Consider the PaaS components you'll need to implement your solution, including not just the platform itself, but any storage, database, authentication, etc., services that your solution will require. You'll need to determine how to get the components to work together as well, especially if they come from different vendors. Remember that if this data crosses the Internet, you'll need to be aware of any security implications as well as any bandwidth charges imposed by any/all of the vendors involved. All else being equal, you may wish to stick with as few vendors as possible (ideally just one) to minimize these complications.

Implementing SaaS on IaaS

As with PaaS, there are some factors unique to IaaS that you'll need to consider as well. You can refer to the [IaaS cloud white paper](#), but a few factors are highlighted below:

- You will need to choose all your components carefully, but especially networking. Networking is the foundation of your application and thus performance, latency, and scalability need to be key concerns. Also, consider your options for deploying networking components, such as routers, firewalls, and load balancers.
- Storage may also be a key component if your application stores a lot of data and/or requires quick access. Consider the storage options for capacity and performance as well as for backup.
- On the computing side, do you need bare metal for some or all of your solution or are virtual machines (VMs) OK? Do you have some components of your application that are very CPU intensive but not memory intensive or vice-versa? Some providers allow you to choose CPU and memory independently, while others have packages that you must choose from.
- You'll need to plan for disaster recovery and business continuity with minimal to no downtime to your application or you may have very upset customers. Think about it: How do you react when your email application like Gmail, Hotmail, etc., goes down? Your customers will react similarly. They won't care that a disaster happened or your provider was doing maintenance. Plan carefully to keep your application online and minimize any disruption from unforeseen disasters.
- Remember, with a SaaS application, you'll need people to manage the back-end infrastructure, keep it maintained, and upgrade it as needed. But also be prepared to engage with your customers on how to use the application, solve issues, etc. You'll need infrastructure to support that (website, chat, forums, phone numbers, etc.).

Once you have carefully planned out how you will deploy your application, the next consideration is whether or not to deploy the entire solution yourself or whether you leverage other PaaS options to provide some of the ancillary infrastructure such as authentication or backup. One is not necessarily better or worse than the other, but you will need to consider all the necessary components, how each one scales, and the licensing implications as your application scales. However, as with PaaS, keep in mind security implications, bandwidth charges imposed, and the number of vendors involved. Some IaaS vendors also have PaaS offerings such as Microsoft or IBM, while others are primarily focused on IaaS only, such as VMware.

The Pros and Cons of Using SaaS

Let's switch gears from being a SaaS provider to being a SaaS consumer. In the remainder of this white paper, we'll look at the issues around consuming various components of your IT needs by third parties. We'll look at use cases and some common SaaS offerings a little later in this document. The advantages of using SaaS applications in general include the following:

- The first, and probably biggest, advantage is that the SaaS applications are usually cheaper than buying a traditional, locally hosted application up front; rather there is typically either a monthly or annual charge for using the app.
- Second, support is typically included in the monthly fee, whereas with traditional applications there is often either no support, or pay-per-incident support, or an annual maintenance contract is required. This is not universally true – some application vendors pride themselves on including great support for free, but this is becoming less and less common.
- Third, special servers are not required to be purchased and maintained at your location, saving more money. Rather, all the support infrastructure is provided by the SaaS provider (see the previous part of this article for more on that). This saves money on the servers, their power and cooling, support, IT personnel to maintain them, etc. Depending on the application, there may be web access to the application or a client application that will be installed on the end users' devices, but typically these applications are not very large and often not overly resource intensive.
- Fourth, maintenance (upgrades, patches, etc.) are taken care of (on the server side) by the vendor, so you don't need to worry about OS vulnerabilities, maintenance windows to install patches, etc. You may need to upgrade the client occasionally, but many clients will check for upgrades and prompt you to upgrade when you start the app (assuming it's not web-based).

On the other hand, disadvantages include:

- First, and probably most important to consider, is vendor lock-in. Especially, if you don't have local copies of your data, you may not have a way to switch applications and migrate your data. The SaaS provider is not incentivized to help you leave its application, so the provider may not offer an export capability (though another vendor may have an import function of some sort that may be used). Even if there is a way to export your data, there may be hundreds of gigabytes of data or more that will need to be migrated, which can be a time-consuming (and potentially expensive) proposition.
- Second, you are reliant on the vendor staying in business. That is not likely a problem for a large company like Salesforce or Microsoft, but may be a bigger issue for smaller companies. They may provide little or no warning that they are about to go out of business, potentially causing the loss of all your data.

- Third, your data is typically stored in the cloud, so you have no control over backup, location of the data, or security of that data. Some of these may be spelled out in an SLA, but it is important you understand what is and is not defined and any recourse you may have if your data is accessed without your permission, it is moved to a location not legally allowed, etc. For many organizations, this can be as important a consideration as the first two, and maybe even more so.
- Fourth, if you don't upgrade the application frequently, purchasing it up front and using it for a decade would in many cases be much cheaper, though support (if needed) could cut into the savings.
- Fifth, the application may be slower due to network latency than when accessed locally. Some applications cache data to minimize this effect or offer bidirectional synchronization of changes, but it's up to you to see if performance is acceptable or not. Related to this is the need for Internet access—your performance will be the slower of your latency and that of the provider's, but can also be affected by congestion on the Internet. Of course if your Internet connection is offline for whatever reason, you won't have access to the application or your data at all until the connection is online again (unless there is an application and locally cached copies of the data it needs).

Some SaaS Offerings to Get You Started

SaaS applications come in almost every shape and size imaginable. While not endorsing any particular application or group of applications, here are some to get you started (sorted by category):

- Office Applications:
 - Google Docs: docs.google.com
 - Office 365 (subscription) and Office Online (free): <https://www.office.com/>
 - ThinkFree Online Office: <http://member.thinkfree.com/member/goAboutService.action>
 - Zoho Docs: <https://www.zoho.com/mail/online-office-apps.html>
- Email:
 - Gmail: www.gmail.com
 - Hotmail, Outlook, and Live (free Microsoft email accounts): www.hotmail.com, www.outlook.com, www.live.com
 - Mail.com: www.mail.com
 - Yahoo Mail: mail.yahoo.com
 - Zoho Mail: <https://www.zoho.com/mail/>
- CRM:
 - LivePerson: www.liveperson.com
 - Microsoft Dynamics: www.microsoft.com/en-us/dynamics/crm.aspx
 - Salesforce: www.salesforce.com
 - SAP (sales on demand): www.sap.com/pc/tech/cloud/software/cloud-for-sales/overview/index.html
 - Zoho CRM: www.zoho.com/crm
- Accounting:
 - QuickBooks Online: quickbooks.intuit.com
 - KashFlow: www.kashflow.com
 - Xero: www.xero.com
 - Zoho Books: www.zoho.com/books
- Expense Management:
 - Concur: www.concur.com
 - ExpenseCloud: app.trinetexpense.com

- Expensify: www.expensify.com
- Project Management:
 - Basecamp: basecamp.com
 - LiquidPlanner: www.liquidplanner.com
 - Zoho Projects: www.zoho.com/projects
- Web Conferencing:
 - GoToMeeting: www.gotomeeting.com
 - join.me: join.me
 - TeamViewer: www.teamviewer.com
 - WebEx: www.webex.com
- Video Conferencing:
 - Blue Jeans: bluejeans.com
 - Facebook: <https://www.facebook.com/help/439078162792430/>
 - Google+ Hangouts: plus.google.com/hangouts
 - Skype: www.skype.com

Conclusion

As we've seen with the other cloud options, there are many considerations that should be taken into account when deciding on whether to provide SaaS to others or to become a consumer. In fact, for some organizations, you may be both a provider of SaaS applications (CRM, for example) and a consumer (enterprise resource planning [ERP], project management, or accounting, for example). Careful consideration of the issues described herein will become key in having a successful outcome by making you a better provider or consumer.

Learn More

To learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge, check out the complete portfolio of [cloud training courses](#) from Global Knowledge courses.

For more information or to register, visit www.globalknowledge.com or call **1-800-COURSES (1-800-268-7737)** to speak with a Global Knowledge training advisor.

Related White Papers

[What to Consider When Implementing Platform as a Service \(PaaS\)](#)

[What You Need to Know Before Implementing Infrastructure as a Service \(IaaS\)](#)

About the Author

John Hales is a VMware, SDN, and SoftLayer instructor at Global Knowledge. He is a lead instructor for the SoftLayer offerings worldwide. John is also the author of many books, from involved technical books from Sybex to exam preparation books, many quick reference guides from BarCharts, to custom courseware for individual customers. John has various certifications, including VMware VCA-DCV, VCA-DT, VCA-Cloud, VCP, VCP-DT, VCAP-DCA, VCI, and VCI Level 2; the Microsoft MCSE, MCDBA, MOUS, and MCT; the EMC EMCSA (EMC Storage Administrator); and CompTIA A+, Network+, and CTT+. John lives with his wife and children in Sunrise, Florida.