The Software-as-a Service (SaaS) Delivery Stack

A Framework for Delivering Successful SaaS Applications

October 2010
Audience
Executives, founders, and chief technology officers of Software-as-a-Service (SaaS) providers or software companies converting their applications to a SaaS model to deliver their applications over the Internet.

Abstract
SaaS has many advantages to the traditional software delivery model. The recurring revenue stream, simpler maintenance and application updates, and the lower cost of delivery and distribution are especially attractive for both the application provider and the end users. However, SaaS delivery has its own unique challenges in the need for an always-on, always-online infrastructure to assure that the application is available to the client 24 x 7 x 365.

This paper provides a framework – called the SaaS Delivery Stack – to plan and track the myriad of components necessary to deliver an always-on SaaS application. It also presents how SaaS providers leverage a range of data center solutions such as colocation, managed servers, and cloud computing to outsource the delivery stack more cost effectively, with a higher level of reliability and lower risk to the business than building it themselves.

This framework can be used as a tool to help plan and measure the security, scalability and reliability of a SaaS delivered application and the IT infrastructure supporting it.

Introduction
Software-as-a-Service is assuming an ever increasing role in delivering software applications to both businesses and consumers. An entire range of business applications – including sales management, CRM, ERP, Business Intelligence, and even fleet management are delivered over the Internet as SaaS applications. If the application server or Internet connection to that service is down, hundreds or thousands of users can be left without access to their business critical applications.

One can also argue that many of the applications that consumers run from Google searches, E-bay, or hosted e-mail are also running critical SaaS applications that are delivered over the Internet. Whether it is used to market your company; take or process sales; track customers; deliver e-mail, audio, streaming video, data or other media; serve customers; or deliver your entire product— SaaS and Internet-based applications are assuming a paramount role in our day-to-day lives.

SaaS applications are expected to be secure (can’t be hacked, data is secure), reliable (doesn’t go down, if it does there are fail-over scenarios) and scalable (if we add 10,000 users we don’t have to re-architect everything). So, what does it take to assure a continuance of success with Internet facing applications? It takes a set of systems working together. SaaS is a complex web of software, hardware, service and infrastructure that must all work together in a cohesive fabric, in a highly fluid environment.

Different applications have different standards to meet for each of these facets relative to the system components. To maximize each of the three facets (secure, reliable, and scalable) at all levels is prohibitively expensive even for the largest of ventures. The SaaS Delivery Stack can be used as a framework to describe each layer of components and the
trade-offs that can be made in delivering a SaaS application. Together, the framework can act as a road map in developing high-availability SaaS applications and provide the tools to make intelligent trade-offs given the specific application, and its criticality to business continuance.
I. The SaaS Delivery Stack

The SaaS Delivery Stack is a framework used to identify all the elements necessary to deliver a successful experience to your end-users. This framework can be used to identify roles and responsibilities and to scope projects. There are nine layers of components to the SaaS Delivery Stack. These layers are described in Figure 1 below.

![Figure 1: SaaS Delivery Stack](image)

Each layer of the delivery model requires designing, implementing, tuning, management, and monitoring to assure a high degree of security, reliability and scalability in that layer.

A. DEVELOPING A SECURE, RELIABLE AND SCALABLE STACK

The security, reliability and scalability of the end-user experience will be limited by the security, reliability and scalability of the weakest layer in the stack. So, it’s important to understand each layer of the stack. While there are many technologies and services that can increase security, reliability and scalability ultimately much will be determined by the architecture of the entire solution.

It’s the sum of the security, reliability, and scalability for each layer of the SaaS Delivery Stack that ultimately determine your quality of the service experience the SaaS application delivers to the end-user.

B. VALUE ADD AND RISK MANAGEMENT

There is an interesting trade-off in value add and risk management as you move up the stack. The higher layers represent the majority of the value add for most SaaS providers.

The SaaS user experience is driven by the application code and database engines underlying that code. Users have an intimate interaction with the application as it’s presented to their screen and how their data is organized and maintained. While the user experience also includes the availability and speed of the application, the users generally have no idea about the underlying operating system, hardware or data center that the SaaS application is tied to. Unless, of course, the operating system gets a virus, the hardware is too slow, or the data center experiences an outage.
The application code and database also contain higher degrees of risk and fewer opportunities to outsource those portions of the SaaS delivery (See Figure 2 below). For example, there may be 100’s of electricians in most 10 square mile areas who can help with electrical problems. There are fewer experts that can help with Internet connectivity and networking issues. Generally speaking, there are very few people who really understand the application code and data based layers of the delivery stack. Quite often, there may be only 3 to 5 will be the only real experts on the source code or database design for any particular application.

The application and database layers of the Delivery Stack are the crown jewels for a SaaS provider. They deliver the greatest value add for the provider and the highest level of differentiation. They also represent the layers of the stack which need to be controlled and managed closely, as they represent the highest level of risk and are the most difficult to outsource.

On the other hand, the lower portion of the delivery stack, from the operating system down, is easier to outsource and can actually lower the risk when managed by a trusted data center partner.

Fortunately for most SaaS providers, their core strengths are in the application code and database development. Managing the server and storage infrastructure, networking architecture, or the data center details is mired with issues that most SaaS companies don’t deal with on a day-to-day basis. Just as fortunately, there are a number of well respected, well trusted hosting providers in the market that can managed the stack from the operating
system down. These hosting solutions include colocation, managed servers and cloud computing.

C. CONCLUSIONS
SaaS initiatives are expensive but can also be hyper-successful requiring you carefully match the spending on each layer of the SaaS Delivery Stack to the stage of your business. By using this framework to identify where you are today and where you need to be as you grow you can set goals and track progress.

In the following sections, we describe each layer of the stack.
II. End User Experience

The end-user experience is pivotal to successful SaaS applications. Said another way, whether delivering business processes, sales, marketing, support, media, communications, or other applications via the Internet—success requires satisfied end-users. No matter your business or purpose, without the end user, your online service might as well not exist. It is critical for any SaaS business to define a framework built around meeting and exceeding the end-users needs.

The end user experience is the result of how the SaaS delivery stack is implemented. Key factors include:

- **Usability** – Adapting the application to the user behavior is critical in delivering a successful SaaS application. Design considerations include look and feel of the application, the “process flow” and understandability of the screens, and data organization data are critical factors in delivering a successful SaaS application.

- **Security** – Security is a serious concern for many SaaS users. They need to know that their data is secure and can’t be accessed by other users on the system, or hackers reaching over the network. The application and data base design as well as network and physical security are key considerations that drive this part of the user experience.

- **Uptime** – The single biggest concern users have when moving from a client-server application to SaaS is the uptime and availability of the application. As they move from a solution that is always available on their desktop or internal server, they need to be assured that their application is always available and always online and that they can get to it 24 hours a day & 7 days a week.

- **Response Times** – The time it takes for an application to respond is a significant factor in the success of a SaaS application. Users are expecting the same experience (in terms of response time) as they have with their client-server or desktop solution. An underlying server architecture that doesn’t have the horsepower to keep up with user demand can readily sink a SaaS application.

- **Service & Support** – SaaS users may require a Service Level Agreement (SLA) that describes the service uptime and availability your clients can expect. The support experience is also a critical factor in the user satisfaction.

- **Training & Customer On-boarding** – Another critical factor in a successful SaaS end user experience is how quickly and easily the user is able to start using and get up to speed on the application. The more complex the application, the more training resources need to be available to ease the customer’s learning curve.

The majority of these user experience factors depend on the underlying delivery stack implementation – from the application code and database design to the underlying server architecture to the data center that the servers reside in.
III. Application & Database

The application and database contain the user interface and business logic. The application is the execution code that interacts with the database which contains the critical information. The application and database is the unique core of the online solution in the SaaS Delivery Stack, but cannot operate without the other layers of the IDS stack.

Most SaaS providers believe that the greatest value added and the biggest differentiator of their SaaS solution resides with the application and database.

As described in the previous section, the application has the greatest impact on a successful user experience. The user interface, scalability and extensibility are all critically dependent on the application design.

When the developers design the application code and database with the entire SaaS Delivery Stack in mind, the solution can meet the scalability and extensibility requirements that customers demand without needing to be re-architected as the customer base grows.

- **Scalability** - The ability of the application to address a range of users needs from the smallest, casual user to the more intense power users is an important consideration in designing the user experience. Buyers of SaaS applications want to know that they will not “hit their head” on capabilities as their use of the system expands over time.

- **Extensibility** – Including the ability to integrate with other applications and other data sources, and the ability to import and export data are important factors for users selecting a SaaS application.

- **Security** – At the application level, the code should be designed to operate as an enterprise solution with the appropriate security controls and architecture. Security as an after-thought is extremely difficult to patch into the system, therefore it should be a requirement developed and reviewed with security experts as the application is developed.

For these reasons, most SaaS companies consider their application code and database their most important asset – one that is understood by relatively few in-house developers. Control and management of the application provides the greatest value add that many SaaS companies can provide. Outsourcing lower layers of the SaaS Delivery Stack to a trusted partner can free up valuable resources to stay focused on this critical part of the company’s services.
IV. Operating System and Server

Online applications and databases require a server infrastructure to deliver the solution from. Depending on the application, this is typically a set of servers (physical or virtual servers) configured with the right operating systems and middleware required to run the application and database.

SaaS companies often need to decide between running the servers internally or outsourcing some or all of the server management to a trusted provider.

Businesses have a number of outsource options available to deliver and manage the servers for their SaaS application – ranging from basic colocation to managed servers to cloud solutions. server options to run their website or application from the low cost sharing of servers, to providing their own server in colocation, or gaining the flexible service of having a dedicated server provided and managed for them.

- **Basic Colocation** – With basic colocation, you procure, configure, and place the own hardware into a colocation data center that provides the lower layers of the SaaS Delivery Stack and lets you manage the server operation either through direct physical access or remotely over the Internet. With remote management, colocation data centers can provide the “hands and eyes” to help you manage your equipment as required at the data center.

- **Managed Colocation** – With managed colocation, you procure the server hardware, and the managed colocation provider configures and manages the hardware in the colocation environment. This is very similar to the managed servers options below, except you are responsible for the capital and procuring the hardware and warranty agreements for the server infrastructure.

- **Managed Servers** – With managed servers, the data center operator procures, configures and manages all aspects of the server hardware and operating system. Compared to colocation, there is no capital required to purchase the hardware or software – the managed servers are provides as a service with a simple monthly fee. The data center operator performs the hardware maintenance, required upgrades, patches, and responds immediately to all server hardware failures.

- **Public Cloud Computing** – The public cloud is defined as a multi-tenant environment, where you buy a “server slice” in a cloud computing environment that is shared with a number of other clients or tenants. The public cloud runs with a utility model where you can turn-on and turn off computing services and only pay for what you use. However, public cloud computing lacks many of the network and data security solutions that most end users demand from their SaaS providers. Public clouds also suffer from the inability to meet specific compliance requirements such as HIPAA, PCI and Sarbanes-Oxley.

- **Private Cloud Computing** – A private cloud is a single-tenant environment where the hardware, storage and network are dedicated to a specific SaaS company. Most SaaS companies prefer a private cloud over a public cloud specifically for the security, privacy and compliance requirements.
V. Network and Connectivity

A. Network
It’s critical that SaaS applications are protected behind a robust, secure network to assure the security of their client’s data. Through colocation, managed servers, or a dedicated private cloud, SaaS providers can leverage the network capabilities of a managed data center operator.

The network architecture at the data center should be designed to control traffic between the servers and the Internet and reduce or eliminate the threat of outside attacks. There are numerous schemes that should be used to protect the network including: IP block assignment (separate subnet for each customer), redundant Universal Threat Management (UTM) systems, firewalls, intrusion detection, anomaly detection, notifications from CERT, updates of all security patches as released, regular updates of firewall rules, and regular review of network traffic logs and statistics.

B. Internet Connectivity
Internet connectivity is a critical factor in successful SaaS deployment. As users demand always-on connectivity, multiple, redundant Internet connections with automatic failover is typically a requirement for data center connectivity.

While difficult (or expensive) to achieve in-house, most reliable data centers provide redundant Internet connectivity into a redundant, dual network architecture. This assures that there is no single point of failure – from the Internet provider all the way to redundant network interface cards (NICs) on the servers.
VI. Electrical and Physical Data Center Facilities

Electrical, cooling, security and physical facilities across data centers vary widely. As you move from low end data centers to higher tier data centers, the reliability and uptime expectations increase.

Lower end data centers can typically be classified as having an “N” electrical infrastructure, and higher end data centers typically have an “N+1” electrical infrastructure.

“N” is enough to do the job – enough power, cooling, UPS, or internet bandwidth required to support the servers and applications running in the data center. “N+1” is having 1 more of every component so that if (or should I say “when”?) anything fails or needs to be taken offline for repair, there is another component on line to take over for the failed unit.

“N+1” is about redundancy, risk reduction and uptime. The “+1” eliminates single points of failure in critical systems – including HVAC, UPS, generators, and utility power connections into the building. This elimination of risk can even extend to high availability (HA) power delivery to each server through redundant power runs through separate PDUs, UPSs and generators.

The best strategy for any particular SaaS solution depends on the end user’s demand for uptime. If end users can afford to be shut down if power or cooling is lost, then uptime isn’t critical, a non-redundant infrastructure (“N” if you will) can do the job.

In most cases, however, SaaS applications require 7x24 operation and are best housed in an N+1 data center. For example medical records SaaS application that need to be accessed night and day for patient records will require 24x7 uptime and online connectivity.

The final element of the SaaS Delivery Stack is the physical facility. With all of the protection built into the higher layers of the SaaS Delivery Stack, it’s important to pay attention to the physical security requirements of the facility itself – including dual authentication secure access to the data center, physically sound structure (e.g. no windows in the data center), fire suppression, and 24x7 surveillance.
VII. Conclusion

SaaS applications require many pieces to work properly to deliver a successful experience to the end-user. The SaaS Delivery Stack can be a useful framework for planning and tracking the myriad of components necessary to deliver successful SaaS applications.

One the model is in place and understood, it can provide a useful roadmap for planning which components to keep in-house and which to outsource. Most SaaS companies find that the highest value added to their customers is at the application and database level, which is difficult and usually not desirable to outsource. At the same time, many SaaS companies find that managing the server and data center infrastructures are not part of their core competency and find it easier to outsource to a trusted provider.

There are a number of outsourcing solutions available – from colocation to managed servers to cloud computing. The outsourcing model you choose will depend on where you deem your greatest value-add to be, and your comfort in outsourcing different elements of the SaaS Delivery Stack.

About Online Tech Inc. – Online Tech is a premier managed data center operator targeted at meeting the needs of SaaS companies. Our solutions include basic colocation, fully managed colocation, managed servers and private cloud computing. A number of SaaS leaders trust Online Tech’s HIPPA Compliant, SAS 70 certified data centers to ensure their servers are always on, always online, and always safe.

To learn more, visit www.onlinetech.com