Solutions for Today's Enterprise

Options and Alternatives
for Effective Selection,
Deployment and Operations

From Premise to SaaS to SaaS²

Michael McCloskey
Chief Executive Officer
FrontRange Solutions

Table of Contents

1 Executive Summary

A company deploying software today has several operational paradigms from which to choose. The criteria for making such an important decision depends on many factors including such things as availability of IT personnel, ability to manage the infrastructure and operations, availability of capital vs. operating funds, integration and customization capabilities and more.

While some in the media would have everyone believe that SaaS — Software-as-a-Service — is the dominant model in today's software world, the fact of the matter is that the traditional premise deployment model, in which software is purchased outright through a perpetual license, remains the most prevalent framework in current usage.

The reality is that each model — premise, SaaS, and now SaaS^2 (Solutions-as-a-Service, or "SaaS Squared") — has particular strengths and weaknesses, making it all the more important that a company carefully consider its specific needs and requirements when evaluating the various options before it. These options, it should be noted, also include hybrid solutions – blended premise, SaaS and SaaS^2 – which bring with them their own unique set of challenges and benefits.

It is imperative that an enterprise develop well-defined criteria with which to evaluate the paradigm that fits best. This paper offers an explanation of the various deployment approaches, and an over-view of some of the more important criteria that should be considered as companies evaluate the most appropriate software solution for their particular environment.

2   Deployment Models Overview

The software deployment options from which a company may choose can broadly be placed into four categories: Premise, ASP/MSP, SaaS, and SaaS[2].

2.1   Premise

The most common model is the "premise" model whereby a company purchases the software upfront, and then has IT professionals (internal and/or external) install, set-up, and configure the software to run on the company's servers or desktop PCs. This involves purchasing a "perpetual" license, with ownership of the software belonging to the enterprise. This has been a tried and tested method that has worked for years. As we will see later in this document, this method is the most prevalent in usage today and a good choice for many customers.

2.2   ASP and MSP

During the late '90s and early 2000s, another model emerged called the "Application Service Provider," or ASP. ASPs, sometimes called MSPs (Managed Service Providers), primarily purchased the software from a premise vendor and then in turn simply hosted the application. The ASP/MSP "rented" the software to customers, and customers were able to avoid upfront licensing fees. However, most of these solutions were not designed with an Internet architecture that was truly multi-tenant, nor did they deliver true cost savings. Customers using an ASP/MSP model often had the choice of "out-of-the-box" functionality with little customization capabilities (and faster implementation than premise solutions), or greater customization capability to meet their specific needs (implementation similar to premise solutions). A benefit of this model was reduced capital outlays, but because all administration was simply transferred to the service provider, there was actually little benefit or cost savings. The ASP/MSP model as described above was largely unsuccessful as it simply shifted costs from the user to the service provider. The ASP is largely a footnote in history and not a true "SaaS" solution in our definition.

2.3   SaaS

The next stage in the evolution of software delivery was SaaS, or Software-as-a-Service. SaaS software provided a new design paradigm, which was to serve many customers on a common infrastructure. The software was designed and architected with the Internet as a foundation. It includes a multi-tenant architecture which is designed to serve multiple customers thereby enabling greater efficiency and cost savings to both the supplier and the customer. In this paradigm, the software is not purchased up front but rather paid for on an annual or subscription basis based on the number of users or other usage criteria. The SaaS supplier of the software manages many of the processes formerly executed by the internal IT organization, including such things as security, fail-over, redundancy, upgrades, and other IT operations. This approach has gained some popularity due to such factors as the perception that deployments will occur more rapidly, that less effort will be required by internal IT organizations to operate the application, and that potentially lower costs of operations will be realized.

2.4 SaaS^2 - Solutions as a Service

An emerging paradigm that builds on the traditional SaaS model is what we at FrontRange term "Solutions as a Service," or SaaS^2. SaaS^2 provides all the benefits of traditional SaaS applications and adds incremental customer benefits not included in typical SaaS offerings. SaaS^2 solutions are built on an Internet-architected platform, support multiple applications, can operate in hybrid environments (premise and SaaS), and are capable of providing both premise and software-as-a-service applications. SaaS^2 solutions are architected as a platform and have the ability to provide multiple applications on their foundation. The applications are robust and include such benefits as a richer user experience, the ability to simultaneously manage best practices as well as complex workflows and configurations, provide strong service and support offerings, have the ability to migrate to premise and hybrid environments, and offer a range of strong service and support offerings. SaaS^2 vendors provide strong domain expertise to deliver best practices based solutions, and are capable of providing end-to-end management of the solution, including full configuration, administration and management. By providing both strong out-of-the-box functionality as well as full management of the solution, SaaS^2 can offer greater value than traditional SaaS options.

While "SaaS" is a popular buzzword and the impression one would get from the media is that it has taken over the software world, it is important to remember that the premise model remains by far the dominant delivery approach today, by a wide margin, and continues to be the right solution for many companies. For an enterprise, making the right choice as to which software or solutions to deploy requires thoughtful consideration. All models have advantages and drawbacks, and inevitably the correct answer depends upon the company's specific circumstances and requirements.

The Models: Premise, ASP, SaaS, and SaaS^2

	Premise	ASP/MSP	SaaS	SaaS^2
Ownership	Customer Procures	Buys and Resells	Service Provider	Solution Provider
Cost Model	License/Maintenance	Rental	Rental	Rental
Implementation	Customer/Vendor/3rd Party	ASP	SaaS/Customer/3rd	SaaS^2/Customer/3rd
Configurability	High	Low	Low/Medium	High
Upgrades	By Customer	By ASP	Service Provider	Solution Provider
User Experience	Customizable	Static	Rigid	Highly Configurable
Infrastructure	Customer	ASP	SaaS	SaaS^2

Figure 1 – Deployment options require careful consideration

3 Deployment Models

3.1 Approaches

As mentioned earlier, there are several operational models available for software deployment in today's enterprises. SaaS has steadily increased its market share in recent years relative to premise-based solutions, although the latter remains as the market leader. Building on the traditional SaaS model is "Solutions-as-a-Service" (SaaS^2), which provides a platform with multiple applications, takes SaaS to a level of richer functionality, integration, and configuration capability, and deliver services and support by domain experts that offer proven and best practices. Importantly, SaaS^2 solution providers are able to offer both SaaS and premise solutions and operate in hybrid environments that allow enterprises to mix and match applications to meet their specific circumstances.

3.2 Premise-based

Software has traditionally been distributed and installed at a customer location. As the software was typically installed inside the organization, it has come to be known as "premise-based." If the application was desktop-based, such as Microsoft Office, it was typically individually installed on PCs throughout the enterprise, and updating the software was often the responsibility of the internal IT staff. For other applications, the programs were often installed on a server and delivered to users. Ultimately, these
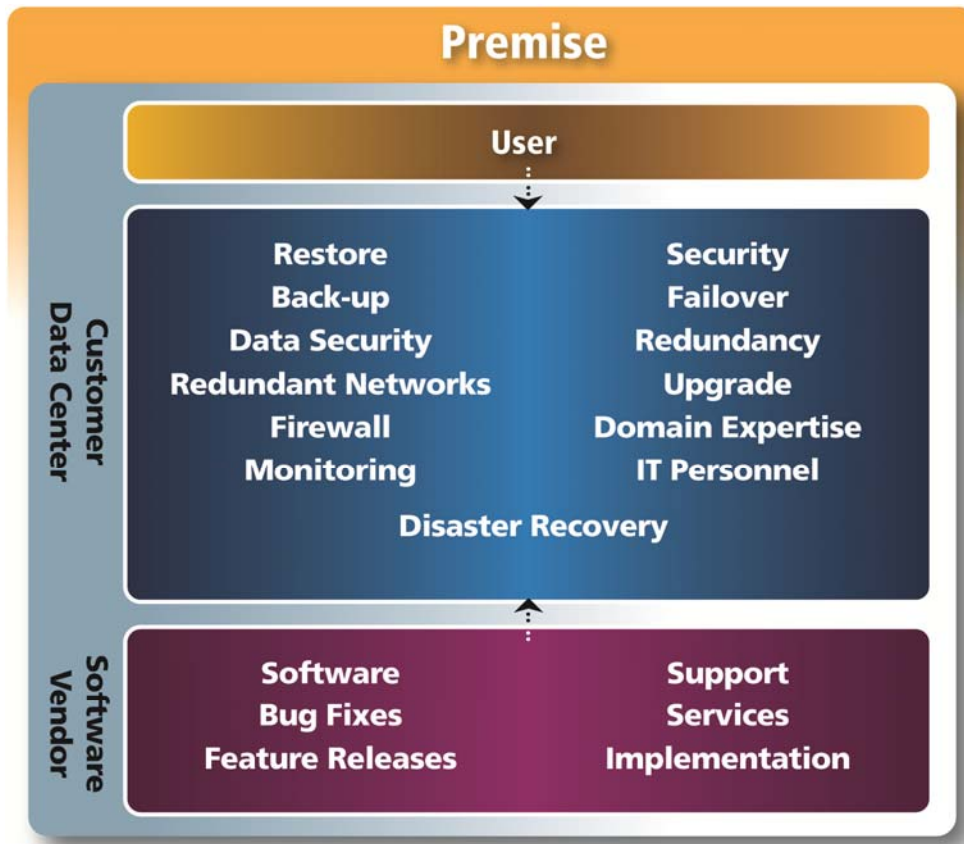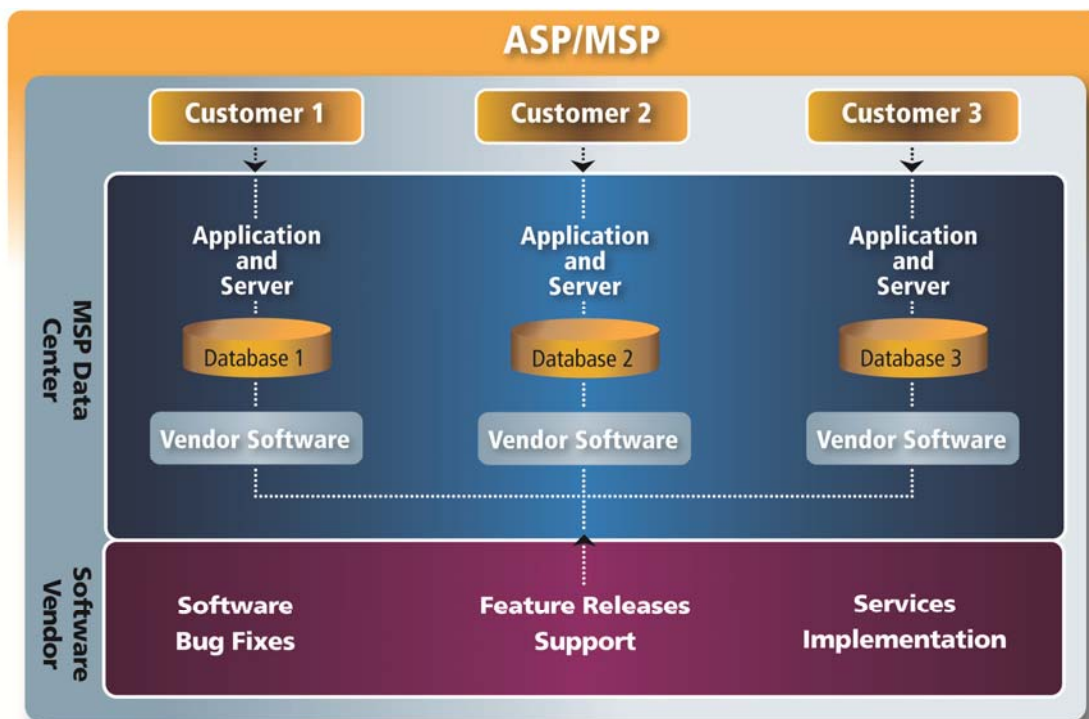
Figure 2 – Premise-based alternative

applications were managed by internal IT organizations with the support of the software vendor. The IT organization would take responsibility for the delivery of the application to users, as well as the requisite security, failover, and operational requirements.

3.3   ASP/MSP (Hosted Premise Solutions)

During the late 90's and early part of this century, however, the premise model has been challenged, as technology has advanced and offered new opportunities for growth, cost-efficiency, and security. As the desire to outsource IT advanced, and the downturn in the earlier part of the decade took hold, the concept of having an application hosted by a third party began to gain traction. This next step in the evolution of software delivery was the "Application Service Provider" or ASP model. ASP's, sometimes referred to as MSP's (managed service providers) "hosted" a premise solution and delivered the application over the Internet. One benefit of this model was reduced capital and upfront costs for the customer, but because all administration was done by the service provider, the cost of maintaining the system and making changes was very high.

Figure 3 – As depicted above, hosted premise-based models are not true SaaS

The ASP model, however, primarily shifted the capital and management costs from inside the company to an outside provider. With no major gain in efficiency, the ASP/MSP model, based on hosting a "premise solution", proved ineffective for both the provider and the customer. Many of these businesses failed as their customers did not receive a true "Internet based" applications, cost savings, nor the flexibility and integration capabilities that they desired. We do not consider the ASP or hosted premise model to be an

effective nor true SaaS offering and would caution customers to carefully consider a true Internet architected solution with SAS 70 capability when considering software as a service.

3.4 Software as a Service

When the Internet began to penetrate the modern organization, it extended the local area network outside the enterprise and provided access to computers literally around the globe. As this access became ubiquitous, pervasive, and exponentially faster, far-flung servers on the network allowed people to run applications remotely. The web browser emerged as the tool to access these programs, and the increasing functionality of the browser led to a richer user experience that was similar to what people were used to with their Windows software. This new software deployment model, known as Software-as-a-Service (SaaS), allowed people to pay for the use of their software on a subscription or rental basis.

**Figure 4 – SaaS models provide real alternatives yet there
is room to continue to improve to new levels of value**

SaaS applications initially tackled business functions, and began in part with less mission-critical applications, such as, sales force management. With Internet connectivity assured — and IT support resources limited — salespersons often worked remotely in a field office or from their homes. What's more, from a company perspective, if the sales force management application was down, the company was unlikely to need to stop operations. Saleforce.com, now with over $1 billion in annual revenue, was

an early champion of this application and the SaaS concept. In time, more and more business applications — such as project planning and accounting — would become available in the SaaS model.

Software-as-a-Service has evolved as a viable alternative to the traditional perpetual license/premised-based model over the past 10 years. A number of technical factors have helped to accelerate this transition, the most important of which is the pervasive availability of the Internet, new software designs, and increased computing power. These capabilities, combined with increased bandwidth speeds, the widespread acceptance of the web browser on every desktop or laptop, and improvements in the browser experience, have all come together to provide a new "global platform" from which applications can be run.

Today's financial realities have forced many firms to place greater scrutiny on their costs, investments, and resources. Spending on IT climbed relatively steadily through the Y2K timeframe, and then slightly beyond, until severe cutbacks by the telecommunications, financial services, and technology firms in the wake of the dotcom crash and downturn in the economy. It was during this correction that many CIOs realized their cost structure was far more fixed than earlier thought, and despite trying to reduce expenditures — including through the shedding of tens of thousands of workers — their large-scale deployments of applications were actually increasing in total costs. With this realization, CIOs wanted to have a more flexible cost structure. The SaaS model offered a means to achieve this objective.

For the early SaaS vendors, server hardware price points began to drop dramatically through the dotcom bust, which bottomed in 2003. The cost, however, of managing many servers continued to be high. Historically, companies wrote their enterprise software to "take over the entire server" for an organization. SaaS vendors began pioneering the concept of "single instance, multi-tenant." In this model, the vendor was able to provide a single application which was capable of serving hundreds (or thousands) of companies at one time. This allows a single hardware infrastructure to be more heavily utilized, bringing down the cost per user for the vendor and ultimately the customer as well. In addition, SaaS providers were able to focus development on a single core code stream, thereby incorporating greater functionality. In this model, users are maintained on the most current versions of the software, automatically receive increasingly new and powerful features, and do not have to be concerned with upgrades.

The evolution from ASP/MSP hosted premise application providers to SaaS offered true economic advantages, in some cases, to the customer. The SaaS provider was able to achieve real economies of scale that could be passed on to the customer, and the customer was able to reduce costs. There are tradeoffs, of course: many early SaaS applications, for example, forced upgrades or maintained all data in a monolithic data model. Many were also inflexible and difficult to integrate with existing applications. While some vendors have solved this problem, it is important for enterprises to fully understand their unique requirements in evaluating a potential solution.

3.5 SaaS[2] and the Cloud

The newest evolution in the marketplace, promulgated by FrontRange Solutions, is Solutions-as-a-Service, which we refer to as SaaS[2]. This model provides all of the traditional benefits offered by SaaS providers with some additional benefits that can include the following:

- True "Internet Based Cloud Platform"
- Provide a Platform with Multiple Applications for a more complete solution
- Embedded best practices for increased standardization and consistency
- Strong domain expertise delivered by industry experts
- Ability to offer the full range of capabilities, from complete self administration and control by the user to 100% managed by the solution provider
- Comprehensive services, integration and customization capabilities to meet specific requirements
- Strong training, services and support capabilities
- Flexible and simple pricing models
- Provide both premise and SaaS solutions
- Ability to operate in hybrid (premise and SaaS) environments
- Provide Migration from SaaS[2] to Premise solutions

SaaS² Isolated Multi-Tenant Cloud Solution

Customer 1 · Customer 2 · Customer 3

SaaS² Solutions as a Service

Application Schema 1 · Application Schema 2 · Application Schema 3

Database 1 · Database 2 · Database 3

SaaS² Solution

SaaS² Cloud Infrastructure

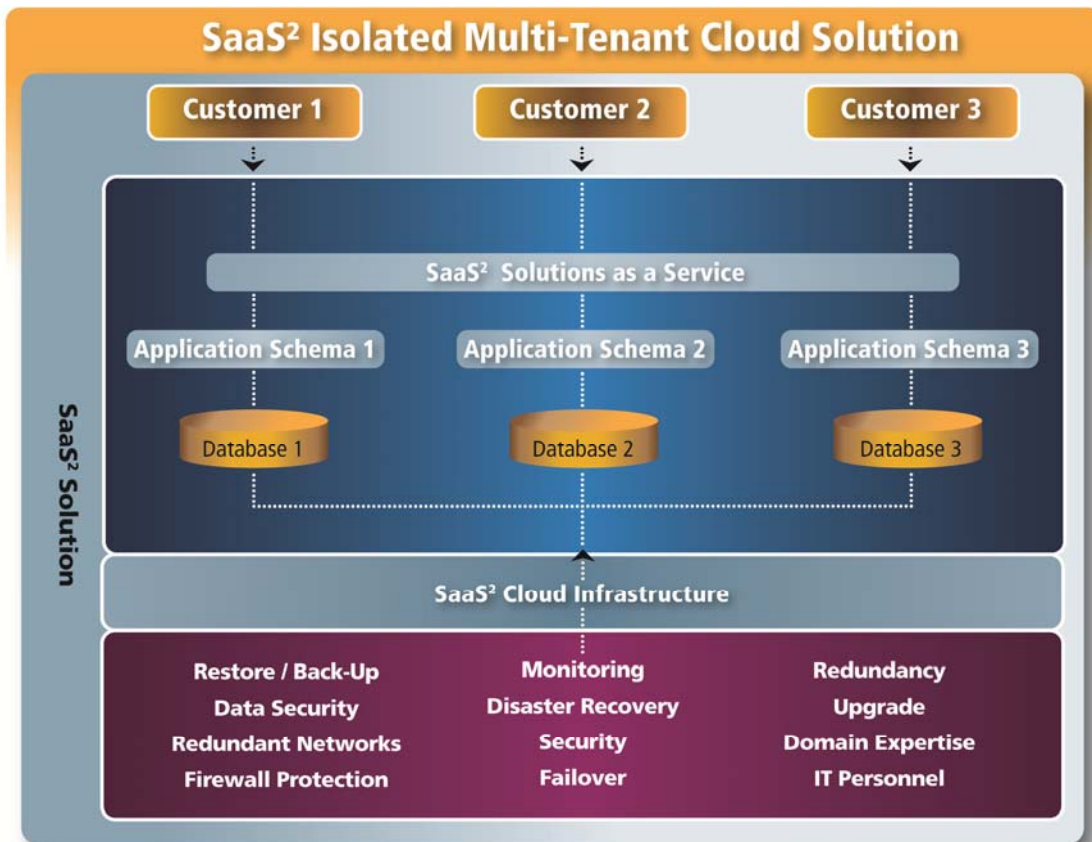| Restore / Back-Up | Monitoring | Redundancy |
Data Security	Disaster Recovery	Upgrade
Redundant Networks	Security	Domain Expertise
Firewall Protection	Failover	IT Personnel

Figure 5 – The SaaS² model provides a complete solutions delivery model enabling optimal value

The ability to manage the entire solution and make changes and enhancements to a customer environment ensures that a company is receiving the exact solution they need, while at the same time significantly reducing risk. The SaaS[2] model goes beyond simply providing an application by offering a platform of related applications. These solutions are capable of converting to premise at a later time in the event that the customer desires to bring the application in-house. SaaS[2] offerings have strong user experiences for faster adoption, provide greater workflow and configuration capabilities, and can have superior automation for increased productivity. SaaS[2] —"Solutions as a Service"— offers more capabilities along with greater value than traditional SaaS application providers.

In addition, a true SaaS[2] provider has the capability to run the solution from a SaaS, cloud, or premise environment. This affords customers choices for the future as business needs change. For example, a company may initially want SaaS, then decide additional services are necessary, or it may become necessary to run the software on an internal cloud or premise environment. Designed correctly, this transition provides the ultimate in customer flexibility. By offering the precise combination of internal versus service provider responsibilities to sustainably meet customer needs, the SaaS[2] provider moves the relationship beyond that of being a mere application provider to that of becoming a true business partner.

3.6   Comparisons

With the two major deployment models — premise-based and SaaS — appearing to dominate the market, it is helpful to look at them side-by-side to consider their relative strengths and weaknesses.

3.6.1   Key SaaS and Premise Differences

When comparing premise-based software and SaaS, consideration must be given to each deployment method's specific impact and requirements. A premise-based solution typically is installed on local servers, and configured and deployed at a customer's facility. It generally serves the application over the LAN, which is often much faster than an Internet connection. Access to the LAN from the outside by remote workers is generally available, but at considerably slower speeds than those experienced by workers inside the organizational walls, as with a typical premise-based solution.

SaaS solutions generally are accessed remotely, with the infrastructure (servers, network connectivity, redundancy, failover, etc.) managed by the SaaS provider. Additionally, the responsibility for the security of the application's data — as well as its potential vulnerability — falls on the SaaS provider. A user's primary requirement is to have Internet connectivity and sufficient bandwidth to be able to run the application with adequate performance.

SaaS applications differ from premise-based software in the following ways:

- Applications require Internet access for the user, who accesses the application from a centrally managed remote server.
- Remote server provides the application to hundreds or thousands of companies at the same time — not just to one company.
- There is no requirement that any specific software be installed inside the organization to access the application other than a web browser.
- Software is updated and maintained by the SaaS Company, removing any burden on the customer.
- Customers typically pay for SaaS software by renting it for a monthly, per-user price. This shifts the cost burden from a large initial outlay to a more manageable payment schedule.

3.6.2   Example: Microsoft Outlook / Exchange and Gmail

A very simple comparison between a more traditional application and a SaaS application is that of Microsoft Outlook/Exchange and Google's Gmail. While the functional lines between these two offerings seem to be increasingly blurring, their current, respective mainstream deployments offer a revealing base for comparison.

Microsoft Outlook runs on a desktop or laptop computer, and connects to a backend server known as Microsoft Exchange. The program typically "downloads" email messages periodically to a local hard drive, so that a user can access his email and attachments even when not connected to the Internet, such as, for example, on an airplane flight. Outlook is very easy to use, and has rich, powerful features; hundreds of millions of people are familiar with its use in a corporate environment. The program also has an extremely large support ecosystem, with devices such as the Blackberry PDA designed to complement the offering and provide road warriors with yet another level of mobility.

Microsoft Outlook is typically included in the larger Microsoft Office suite, which means that — even though the application suite is one of the most widely used in the world — organizations may be paying for functionality they do not use. Further, Outlook must be installed and configured on each PC or laptop, typically by an IT professional. And the Exchange server, which provides the messages to Outlook, has to have its own dedicated in-house server — again, installed and managed by IT professionals. The total cost of the Exchange Server and its associated hardware can be $10,000 or more. For a small business with a dozen or so employees, the cost per-person of such a configuration can be material.

Gmail, on the other hand, is an application hosted by Google that can be accessed from a web browser. No IT professional is required for an installation, as there is none. Nor is there any requirement for a server to be installed within the client company. Gmail, however, operates best when an Internet

connection is working. While it's possible to "download" Gmail to a mobile device, Google still relies on persistent Internet connectivity to send and receive messages.

Gmail is not generally regarded as powerful as Outlook, and is often slower when it comes to uploading attachments. Outlook, combined with Exchange, is able to move messages along the Internet network in an organization at speeds that are often ten times faster than the Internet connection.

This is not to say which is better, only to illustrate one must carefully assess the correct choice based on specific needs and circumstances.

4 Key Considerations Overview

For organizations with a capable IT department and available capital for IT investment, purchasing software outright and deploying it across the entire enterprise — the perpetual license/premise-based model — may provide productivity and cost benefits that can set the firm apart from competitors.

For companies with limited IT resources, on the other hand, or a desire to focus resources in other areas considered more strategic, the concept of having access to powerful software applications that are managed and maintained by a third party can be extremely attractive. As Internet applications have improved and gained maturity, Software-as-a-Service has emerged as a viable way of providing application functionality to a user.

For Fortune 500 global corporations or small agile manufacturing firms, key questions and considerations should be addressed when evaluating which software model works best. These issues include:

- application's importance to the organization's daily operation
- the firm's internal IT capabilities
- geographic support requirements
- the availability of capital to the firm
- the speed within which an application must be up and running
- customization requirements expected for a particular application
- the level of internal expertise required to operate the application
- ability to manage upgrades and deploy new features
- the overall size of the potential user base within the organization
- typical transaction volumes to be supported by the application
- data and security requirements
- delivery model desired – internal vs. external – to enable focus on core competency
- switching costs
- IT staff's ability to implement security in-house along the lines of SAS 70
- the degree to which application must integrate with existing IT system elements

5 Key Considerations

5.1 Internal IT Resources and Core Competencies

Any enterprise's decision to deploy a new application necessarily begins with a serious evaluation of the resources needed.

Critical issues which should be addressed include: What resources are required for successful project completion and operation? What are the other competing projects and where does software deployment fit in terms of prioritization? Is the solution part of delivering the enterprise's core competency? The central role that IT plays in enterprises, combined with the "lean staffing mode" so often seen in the present economy, fosters a cautious perspective towards any expansion of IT workload.

While lower impact on IT staff is a key factor in choosing SaaS, it is critical in most cases that IT leadership take on at least an advisory role in any SaaS decision, given the leadership's subject matter expertise and essential judgment in guiding the enterprise's information future.

5.2 Security

Every IT expert (and to some extent most people today) understands that in addition to its many benefits, today's Internet environment also includes a broad range of threats, including identity theft, viruses, malware, and denial of service attacks, to name just a few.

While all SaaS solutions rely on Internet connectivity, the "pooled" nature of their subscription base allows a degree of investment in their "hardening" such that they are considered as secure as — or even more secure than — the on-premise solutions used by the largest enterprises. Nonetheless, for many organizations, some information is seen as too precious to expose it to the Internet.

One way of characterizing the risk of a given data deployment is by ranking its security motive:

- **Sovereignty:** Government matters, ranging from military and statecraft to pending legal decisions, have strict compliance requirements and severe or even criminal penalties for violations.
- **Privacy:** Individual and organizational projects often depend on private information, such as trade secrets, inventions, and more, for their success.
- **Confidentiality:** Parties holding information belonging to others can have a contractual duty to avoid its disclosure.

Data security decisions always include a judgment component. It is important to note that the higher the data security concerns rank (on the above list), the more compelling the need to isolate the data from

potential antagonists. SaaS solutions are written to standards considered appropriate to confidential data. Any suggestion that SaaS solutions are not appropriate for privacy- or sovereignty-oriented data security criteria is subject to question at this time.

5.3   Integration with On-Premise Applications

Premise solutions generally have strong and mature integration capabilities. SaaS solutions, by design, historically tend to favor simpler client-side interfaces. However "open" they may be in design, it is important to recognize that many SaaS solutions may not readily match the customizations found at a given subscriber premise.

It is a basic design fact that SaaS applications are fully able to interconnect and exchange data with their web-based clients. For SaaS solutions, the threshold of integration at the basic operational level isn't in question; rather, the issue is how well the SaaS context can "plug and play" with respect to the rest of the enterprise software mix.

On the other hand, customization can also be part of the integration issue's solution: since SaaS interfaces tend to be standardized, a SaaS customization (browser proxy/gateway or plug-in) could be used at a given premise to bridge local information systems with one or more SaaS data sets. The disadvantage of this approach is the time and cost for developing such a customization, directly in conflict with the "quick and affordable" benefits of SaaS. SaaS providers, generally quite aware of this concern, have software libraries intended to simplify this type of data integration.

5.4   Distributed Facilities and Bandwidth Availability

A practical concern for any premise or SaaS application's deployment is the degree to which the remote offices and other facilities critical to the application's use are connected, as measured in terms of bandwidth *actually available for the application*.

The technology side of this question is aimed at an organization's IT department; its answer should be based on a recent network analysis. What may be expected from moving a given application off-premise onto SaaS will depend on the starting migration point:

- **On-premise only, single facility:** bandwidth requirements will increase with the seat count and activity level.
- **On-premise only, multiple facilities:** bandwidth requirements may actually decrease, since remote sites will contend less for access to the main facility.

The actual distribution of SaaS clients will drive changes in the pattern of use. While the business side of this question focuses on cost optimization, satisfaction results when both sides — technical and business — meet to explore the details and find equilibrium.

5.5 Costs

One of the biggest differences between the traditional license model and SaaS is how each approach's costs are spread out across the product's lifetime.

With a perpetual license model, a company must outlay an initial amount to purchase the product license, which is then typically used by many workers in the company. Additional monies are often required to purchase a server to run the software, as well as to have IT staff install and set up the software as needed. Once the application is up and running, further costs accrue for maintenance, support, product enhancements, and the internal IT personnel who are tasked with managing it all.

The advantage of the perpetual model is, to some extent, the fixed nature of these costs. Once paid for, a system's maintenance figures remain relatively constant, thus enabling the organization to have largely "capped" its cost to a known quantity. Very large organizations are often able to negotiate an attractive license fee, whereby they can effectively serve a high number of users for a specific price; consequently, the incremental cost of supporting new users is marginal.

With a SaaS model, the initial outlays are less than those for the perpetual license model for several reasons. First, instead of purchasing the software outright, the company commits to using the software for a minimum period of time — for example, 12 months — in a relationship that is more like a rental. Second, rather than purchasing a certain number of users or capacity, the SaaS model allows an organization to add new licenses as needed, so they may start small and expand later, steadily increasing the number of users and the associated costs as their organizational needs expand. Lastly, because SaaS software is deployed over the web, the IT department may be much less involved, with its attendant lower costs.

When considering premise vs. SaaS solutions, organizations should carefully consider the cost portfolio over an extended number of years. This should include as many hard costs as possible, including such items as licensing and maintenance vs. ongoing costs of renting, costs associated with maintaining the system, configuration and deployment costs, server and infrastructure costs, and personnel costs. Access to capital vs. ongoing operating costs will also help determine which approach may be more appropriate for a given situation. In all cases, it is advisable to build a cost comparison matrix as an aid in evaluating all financial factors.

5.6 Expected Term-of-Use

Ideally, the purpose of an application, and particularly its projected lifetime use, should figure into a SaaS vs. on-premise deployment decision. Most managers, when allocating resources for a given application, will consider the application's role in the enterprise — standards are different for enduring, longer-term requirements than for projects that are ephemeral or speculative in nature.

Today's SaaS solutions are typically directed to basic business functions, so that a SaaS choice tends to serve a longer-term agenda. This is not to suggest that SaaS has no place in the prototyping/rapid application development arena, but it is fair to say that the more speculative aspects of SaaS use relate to feasibility evaluations and testing scenarios.

This latter point again relates to the importance of assessing the SaaS provider, and the range of options available for continued use of a given SaaS application should the provider leave the market.

5.7 Customization Level

The economy-of-scale arguments supporting commodity oriented solutions such as SaaS can be compelling, but such arguments may become weaker once customization enters into a given subscriber's requirements list. In practice, this is where the hard choices come into play – in the end, the SaaS vs. on-premise decision hinges on product fit in actual use, with the challenge growing most serious when none of the available options presents an ideal fit. While the prospects for such situations grow fewer as SaaS offerings mature and broaden, the ideal of a "perfect fit" SaaS solution can be hard to find for organizations with specialized needs.

In these cases, it becomes necessary to shift the due diligence from fact finding to fact *facing*, as the organization takes on the often painful question of to what degree a "customized" application solution is actually needed. This feature assessment must first evaluate whether the enterprise's core needs are being met in the provider's current packages, and if not, whether they are in fact necessary or whether the anticipated time and cost to enhance the current packages will be sufficient to meet those needs.

In some more elaborate cases, even if the SaaS feature analysis indicates that the existing application doesn't include the entire feature set needed, it may still be the chosen option as it is the closest fit available. For such cases, a hybrid SaaS scenario may be the near-optimal choice: adjusting the mix of service choices to reach the organization's objectives.

Factors to consider when the options for meeting core requirements diminish include:
- **A la carte SaaS:** use of the "low-hanging fruit" from a given SaaS bundle
- **Multiple SaaS Services:** use of multiple SaaS services to accomplish objectives
- **Customization:** use of on-premise elements to fill gaps.

5.8   User Experience, Training Considerations, and User Adoption

The user experience is dependent upon the manner in which the User Interface addresses the goals and objectives of the solution's users. The UI must be attractive, easy to use, and contextual, while ultimately providing high levels of automation that simplify and accelerate task completion, and making the job of the user more productive and satisfying.

Training is a critical success factor for any solution. Users must be thoroughly trained so that they get the most out of the solution. Today, training typically consumes a significant portion of the HR budget. While funding for training has declined sharply in the present recession, affordable, excellent training remains one of the most important benefits promised by SaaS. Yet, promises must be tested, making a careful comparison of the relative costs in training personnel to use SaaS and/or on-Premise solutions essential.

A compelling user experience and well-trained users are critical to increasing application adoption and usage. The success of the solution and satisfaction of those using the solution will increase user adoption and result in more satisfied clients.

6 FrontRange - Solutions as a Service (SaaS²) and Cloud

FrontRange believes it is important to offer a range of premise and SaaS² applications so that customers have full flexibility in satisfying their needs. Solutions-as-a-Service (SaaS²) provides all of the traditional benefits of SaaS and premise providers with some important additional benefits, including:

- Provide both premise and SaaS² solutions
- Deployment of embedded best practices for increased standardization
- Strong domain expertise by industry experts
- Perform major administrative and workflow maintenance for reduced costs
- Provide comprehensive services and customization capabilities to meet specific requirements
- Deliver strong training, support and consulting capabilities
- Capability to define, implement, and manage the entire solution
- Ability to migrate the SaaS² product in house in the future
- Provide a platform for multiple applications
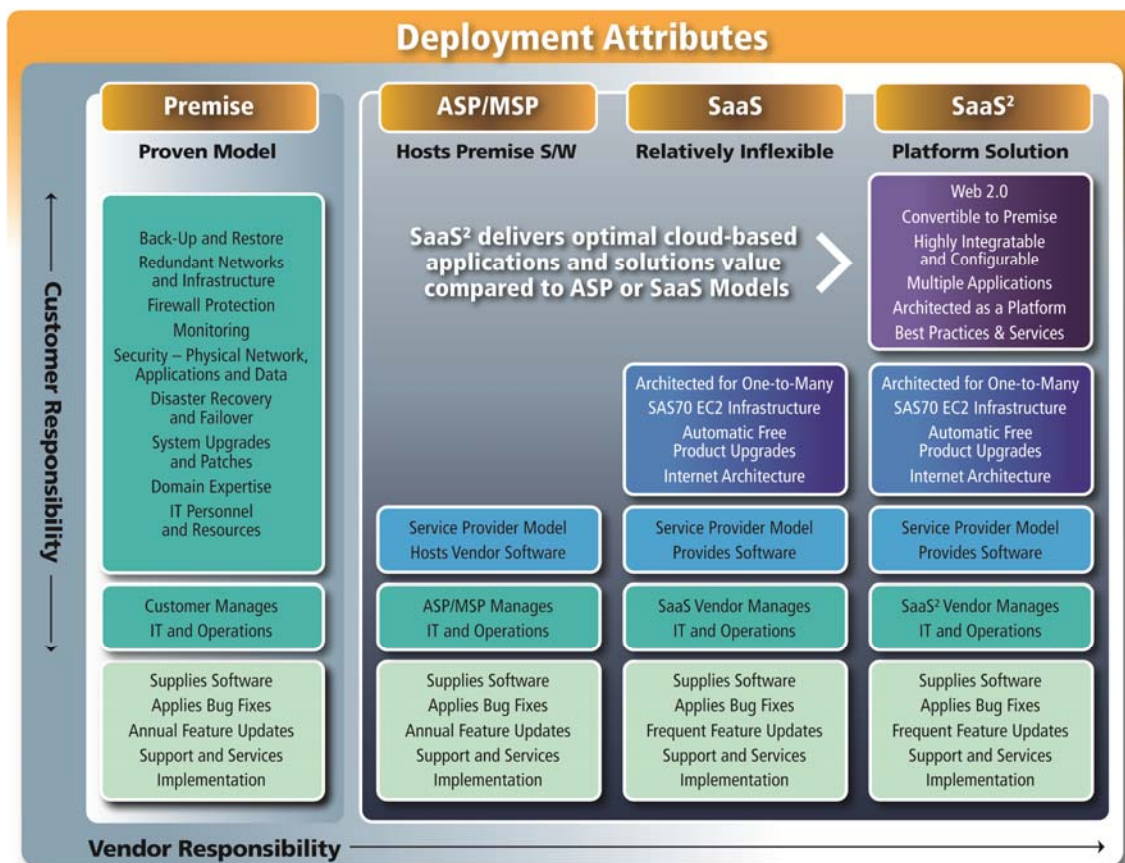- Flexible pricing models
- Ability to operate in a hybrid environment

**Figure 6 —Premise and cloud-enabled solutions both provide viable
options today depending upon organizational needs and resources**

FrontRange believes the ability to manage the entire solution and make changes and enhancements to a customer environment ensures that a company is receiving the exact solution they are looking for while at the same time significantly reducing risk. The relationship with the customer and solution provider moves beyond providing the application to becoming a true business partner. A SaaS^2 provider is capable of complete administration of the application, support for users, deploying industry experts for training and education, and configuring and managing the complete lifecycle of the solution.

FrontRange - Pioneering Solutions as a Service (SaaS^2)

FrontRange Solutions provides customers with a comprehensive portfolio of solutions that include Premise and SaaS^2 Solutions. Customers have the flexibility of choosing the type of solution they prefer, as well as to mix and match environments for hybrid deployments based on their specific needs and requirements. When evaluating a solution provider, it is a strong benefit to consider a provider that offers a combination of Premise, SaaS and SaaS^2 solutions.

7 Decision Matrix

As we've seen, deciding what is the best path forward for enterprise software deployment is a non-trivial task. In some ways, buyers are now provided with more options than ever before.

No one solution is right for every organization. Each organization must identify the key considerations that should be evaluated and establish weightings on key decision points in advance of any decision.

The figure below, in the context of the broader discussion of this white paper, is intended to help companies wrestling with the selection decision by highlighting some of the key factors involved and, depending on the requirements or resources of the firm, to direct a decision toward a premise, SaaS, or hybrid solution. Applying weightings to each factor yields a score that offers a directional indication of which approach may be a better match, given current and future business realities.

Selection Guide SaaS Versus Premise

	SaaS											Premise
IT Staffing: (Capabilities/Availability)	LOW	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	HIGH
Geographic Dispersion Of Users	HIGH	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	LOW
Capital Availability	LOW	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	HIGH
Speed Required to Deliver Completed App	RAPID	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	NORMAL
Customization Requirements	LOW	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	HIGH
Number of potential users	LOW	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	HIGH
Pervasive Internet Access	YES	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	NO
Bandwidth Availability	HIGH	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	LOW
Integration with other Apps	LOW	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	HIGH
Expected Usage For Application	SHORT	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	LONG

About FrontRange Solutions

FrontRange is a leading provider of powerful and affordable premise and on-demand based IT Service Management, IT Asset Management, and Customer Service solutions. Our solutions enable IT and Services Transformation by providing Enterprise-class capabilities that deliver fast time-to-benefit, high ease-of-use, and rapid return-on-investment. With an award-winning tradition and recognized as a leader by industry analysts, FrontRange's products and solutions are used by over 13,000 customers in more than 80 verticals and 45 countries to quickly improve interactions with external and internal clients and achieve better business results. For more information, call 800.776.7889, visit www.frontrange.com, or follow us at 'FrontRangeSols' on Twitter.

For more information visit www.frontrange.com or call:

North America
United States   800.443.5457

EMEA		*APAC*	
United Kingdom	*+44 118 951 8000*	*Melbourne*	*+61 3 9823 6292*
France	*+33 13 926 5555*	*Sydney*	*+61 2 8080 3300*
Germany	*+49 89 31 8830*	*China*	*+86 10 6581 4196*
Italy	*+39 03 654 48253*	*India*	*+91 22 3088 1119*
Poland	*+49 89 31 8830*	*New Zealand*	*+64 9 359 7402*
Russia	*+7 495 710 9930*	*Singapore*	*+65 6829 2147*
Spain	*+34 91 550 1646*		
South Africa	*+27 11 325 5600*		