

Executing an IP Protection Strategy in a SaaS ENVIRONMENT

A closer look at the Software as a Service model and how technology escrow can help protect your subscription investment and, potentially, your business.

BY FRANK BRUNO



SAAS IS A HOT TOPIC THESE DAYS AND the market is taking off. SaaS—or Software as a Service—is a software distribution model in which applications are hosted by a vendor, or service provider, and made available to customers via a network, typically the Internet. Rising at a compound annual growth rate of 28 percent, according to a study by market analyst IDC, a subsidiary of International Data Group, predicts the SaaS market will reach \$8 billion by 2007 as the role of on-demand, and hosted application management expands, and adoption accelerates.¹ With the rapid growth and implementation of SaaS, the risks involved in letting someone else host your software applications, and your proprietary information, must be assessed.

The SaaS Story

The names have changed since the early part of the decade. What were once known as Application Service Providers (ASPs) are now called providers of hosted solutions, SaaS, On-Demand Software, and managed services. Additionally, the emergence of a new generation of net-native, or Web-enabled, software solutions supports the rise of the new SaaS providers.

In 1999, salesforce.com pioneered the concept of delivering enterprise applications via a simple Web

site. Recognizing the inherent inefficiencies of the traditional software market, salesforce.com has evolved and now features customer relationship management, and sales force automation applications. In addition, NetSuite, Inc., offers enterprise resource planning applications. Providers of SaaS are able to minimize the time, effort and costs that users typically have for installing applications and keeping them running.

A recent survey by IT research and consulting service Cutter Consortium showed SaaS gaining attention, and winning customer adoption because of its attractive combination of accelerated deployment capabilities, and real cost savings. “According to those in our survey who are currently using SaaS, it is meeting their expectations by generating a greater [return on investment] ROI than traditional software packages, while lowering staff support requirements, and improving application reliability and performance,” said Jeffrey Kaplan, senior consultant with Cutter Consortium’s sourcing and vendor relationships team.² Other highly sought after benefits provided by SaaS include the ability to accelerate the software development process, and shift the focus of IT staff to more strategic projects.

A survey of 200 industry professionals, who attended a recent SaaS summit, showed about 40 percent of the respondents saying they were exploring the option of moving over to SaaS, while 35 percent said they had already initiated the process. Companies of all sizes are taking advantage of SaaS solutions. The most popular types of business applications currently sought by SaaS customers are Customer Relationship Management (CRM), Sales Force Automation (SFA), Enterprise Resource Planning (ERP), Human Resource Management (HRM), and Supply Chain Management (SCM).

Assessing the Risks of SaaS

As companies embrace SaaS, however, it is critical to have clear business continuity, and disaster recovery plans in place. As with anything, there are glitches with on-demand models. SaaS providers have been known to go offline with outages lasting several hours, leaving customers in the dark. Customers also need to make sure Web-based security measures are in place to protect the company’s data, which is now online. Customers must guard themselves, not only against temporary outages, but against more severe disruptions; such as if the SaaS provider suddenly disappeared, or if it were no longer providing an adequate level of service delivery.

Six Steps to SaaS Safety

1. Can the SaaS vendor provide the support and reliability that our business demands?

From the perspective of the SaaS vendor, an important key to their success is the ability to convince potential customers that their data, and the ability to use it, are safely entrusted with the SaaS provider.

2. What proactive measures can SaaS providers take to address and allay fears, and move business forward?

In a SaaS environment, the SaaS provider owns and operates the software application, maintains the servers that run the application, and employs the people needed to maintain the application, making this a more affordable option than the traditional software model. Instead of purchasing the software outright via a license, users can take advantage of a subscription, or pay-as-you-go model, and also eliminate some of the costs associated with IT infrastructure and staffing.

3. Can technology escrow provide the same benefits to users, as well as providers of SaaS services?

The answer is yes. In fact, setting up an escrow account becomes even

About the Author

FRANK BRUNO is a senior business strategist in the division of Intellectual Property Management Services at Iron Mountain, headquartered in Boston, Massachusetts. Bruno consults with corporations, contract managers, developers, and intellectual property (IP) law professionals throughout the United States concerning IP management best practices, including technology escrow matters. He has spoken at many professional and industry events, most recently at the NCMA World Congress 2006. Send comments about this article to cm@ncmahq.org.

more critical when using SaaS, since loss of support by the SaaS provider means not only the loss of the application functionality, but access to all of the proprietary data along with it.

Technology escrow and verification services have long been used in traditional software licensing to protect the licensee. Under this scenario, an escrow agent holds a copy of the software source code in escrow, releasing it to the licensee in the event certain release conditions are met, such as if a developer goes out of business or fails to support the product.

4. What differences exist between the intellectual property protected in a traditional licensed software agreement and that of a SaaS provider?

For starters, the deposit content list will differ for a SaaS model. In addition to requiring the source code itself, the user should ask for a copy of the object code, or “the executable,” as well as instructions on building the application production environment, such as hardware or third-party tools, and, of course, the most recent version of the data contained in the application, at a minimum, to get back up and running.

Source code is the human readable version of the software. It is compiled into object code (also known as the “executable”). You want both in a SaaS situation because the licensee does not even have object code—they have simply subscribed to the application as a service served up through the web. If the vendor goes away, the licensee has nothing. This is essentially a disaster. Therefore, it is imperative to gain access to the object code, a description of the live production environment (what equipment and other tools are necessary to run the executable) and of course, the data. You would want the object code first and then the source code to continue to maintain the software.

For the same reason that a developer will not automatically provide source code when they license their software, do not expect a copy of the

“According to those in our survey who are currently using SaaS, it is meeting their expectations by generating a greater ROI [return on investment] than traditional software packages, while lowering staff support requirements, and improving application reliability and performance.”

– Jeffrey Kaplan
Senior Consultant, Sourcing and Vendor Relationships Team, Cutter Consortium

object code from the SaaS provider. Distributing the executable could cause pirating which could compromise the SaaS provider’s revenue stream. In other words, object code is the equivalent to source code when it comes to a SaaS provider.

5. Consider negotiating for a “demand release”

A demand release is a condition that guarantees that the user receives the object code, related components and data, upon sole request without delay or objection from the developer. Access to the source code can be set up in a separate account with standard release conditions, since recreating the application development environment would be of secondary importance to the disaster at hand. The demand release negates the ramifications of going, for example, from 10 to 15 days without access to your customer or financial data.

6. Conduct a disaster recovery test

Further, when setting up an SaaS escrow account, a thorough disaster recovery test should be performed on the front end to validate the build instructions to ensure a quick, and successful redeployment of the application into production, should a demand release occur. With respect to the data, many companies do not feel that it is necessary to escrow

data, since it already belongs to them. However, considering the Recovery Time Objective (RTO), and the Recovery Point Objective (RPO) by application, you may determine that the frequency of back-ups might be greater for mission-critical applications that do not run on your infrastructure.

Since time is of the essence in a SaaS business relationship, resolving these issues up front could mean the difference between being back online within days, rather than weeks—or possibly not at all. **CM**

Endnotes

1. Jessica Sebor, “SaaS Will Increase Serving Size,” *CRM magazine’s eweekly* (March 21, 2006), destinationCRM.com. Accessed at <http://destinationcrm.com/articles/default.asp?ArticleID=5927&Keywords=SaaS>. Erin Traudt and Amy Konary, “Top 10 Predictions of 2006: Software as a Service, March 2006. IDC market study available for purchase at <http://www.idc.com/getdoc.jsp?containerId=34872>.
2. Kaplan, Jeffrey, “SaaS Survey Shows New Model Becoming Mainstream,” *Cutter Consortium Executive Update*, Vol. 6, No. 22, December 2005. Available from Cutter Consortium :: Special Offer: SaaS Survey Shows New Model Becoming Mainstream.
3. “SaaS is OK, but not yet Sassy,” Red Herring, March 2, 2006. <http://www.redherring.com/Article.aspx?a=15941&hed=SaaS+is+OK%2C+but++Not+Yet+Sassy>