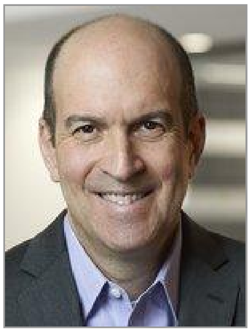
LAW360

Portfolio Media. Inc. | 111 West 19th Street, 5th floor | New York, NY 10011 | www.law360.com
Phone: +1 646 783 7100 | Fax: +1 646 783 7161 | customerservice@law360.com

Source Code Escrow Agreements Are Reaching For The Cloud

By **Mark Kesslen and Leah Satlin** (February 28, 2020, 3:22 PM EST)

Source code escrow agreements have long been accepted by software providers in traditional on-premises software sales. But how often do we see on-premises software licenses today? An overwhelming number of vital business functions are now offered through cloud applications, including software-as-a-service solutions.



Mark Kesslen

When it comes to SaaS, the customer is often at a greater risk of losing access to the solution than it would be with traditional software, and yet the traditional source code escrow model is not sufficient to mitigate that risk. As tech transactions practitioners who negotiate SaaS agreements on a near-daily basis, we are seeing, in real time, a rapidly changing market in which SaaS customers are demanding source code escrow agreements, and a growing number of SaaS providers are capitulating.

Leah Satlin

So, how does it work, and how are the risks and costs allocated between the parties?

To understand the new escrow model, one needs understand the traditional on-premises escrow model. Source code escrow offers buyers a contingency plan in the event the provider goes out of business or no longer offers maintenance and support for certain software programs that buyers may consider mission critical to their businesses.

When a business becomes dependent on certain software to maintain operations, a source code escrow provision in its software license agreement (and separate three-party source code escrow agreement among the customer, provider and escrow agent) is considered an essential safety net for business continuity. This model has become so commonplace in the market that buyers expect it and, more often than not, software providers offer it up front, as a standard provision in their agreements. This leads to smoother negotiations and establishes trust between vendor and customer.

On-premises software operates in the customer’s own live environment, and the customer’s data is stored on its internal systems and backup systems. The software’s availability depends on the availability of the customer’s own system. If the software provider decides to discontinue the software, declares bankruptcy, or ceases operations, there is generally no immediate concern to the customer, because the software can continue to run on the customer’s system.

In such cases, the customer would invoke its rights under its traditional source code escrow agreement, obtain access to the source code and other materials and recreate (or engage a service provider to recreate) the development environment for the software, which would allow the customer to continue to use, maintain and update the software with little downtime, if any.

Because there is little or no threat of immediate, substantial business interruption, traditional source code escrow agreements often contain release conditions that require some time to pass before the escrow agent is permitted to release the materials to the customer (e.g., the provider must cease operations for a period of 60 days or more).

With SaaS, on the other hand, the software code, infrastructure, data and storage exist in a production environment outside of the customer’s premises. The availability of the software often depends not only on the SaaS provider, but on its third-party hosting/cloud providers. Outages, and inaccessibility to data, lasting mere minutes could result in substantial business interruption for the customer. In such a scenario, a traditional source code escrow agreement is of little or no use to the customer.

Instead, what is required is a far broader scope of escrow materials and services to aid the customer in case of an outage, including a copy of the customer’s data stored in a secure back up data center, back up hosting, highly detailed documentation containing build instructions for recreating (or engaging a third-party vendor to recreate) the application and production environment, and, of course, the source code and object code.

Recognizing this issue, source code escrow service providers now offer SaaS risk management services. These escrow providers have created programs to ensure SaaS application continuity and data accessibility, by offering capabilities such as copying the SaaS application (and all of the customer’s data) to a second server located at a secure data center, and even hosting a standby recovery environment on which to seamlessly run the application in the case of mission critical applications where the customer cannot afford even a few minutes of downtime. This concept isn’t new; escrow companies have offered SaaS escrow services for over a decade.

SaaS providers, however, have resisted the inclusion of escrow provisions in their subscription agreements, though the tide appears to be turning recently. SaaS providers had previously taken the position that they made no continuity guarantees and have relied upon business continuity and disaster recovery policies to assuage customers (even though such business continuity and disaster recovery policies often apply to the provider’s business, not the customer’s business).

However, as businesses are becoming more sophisticated about cloud solutions, and more experienced in onboarding SaaS applications, we are seeing more demand for SaaS escrow provisions in subscription agreements.

There are many things to consider in negotiating SaaS escrow provisions. The scope of the escrow materials is an important issue, because the customer wants the deposited materials to include everything necessary to reproduce the development environment and run the application, and yet it is not always clear what that means. The level of detail that providers may have to give in their documentation to enable the customer or a third party to recompile the executable code may be far above and beyond what the provider typically states in its customer-facing documentation, which could lead to extra costs incurred by the provider.

In some cases, to fully transition the solution to another environment, the customer may need access to third-party ancillary software or data sources that support the SaaS application, and providers must consider whether it is even within the provider’s rights or ability to place into escrow.

Customers tend to want to use escrow service providers who can maintain mirrored applications that can be instantaneously activated and hosted by the escrow agent, the customer or the customer’s third-party service provider — effectively serving as a business continuity site. Such escrow programs can be expensive, so the issue of cost allocation is another point of negotiation in SaaS escrow provisions as they become more prevalent in the market.

In addition, one of the biggest points of contention between providers and customers with respect to SaaS escrow is the escrow-release conditions. As discussed above, under the traditional software model the release of source code generally requires the vendor to cease all operations for a significant time period (e.g. 30-60 days), file a petition in bankruptcy or any other proceeding relating to insolvency, liquidation or assignment for the benefit of creditors, or officially discontinue the software and/or support for it.

With respect to the SaaS escrow model, savvy customers understand the need for release conditions addressing the urgency associated with downtime. However, providers do not want to release their applications and all of the intellectual property related thereto so easily.

Therefore, SaaS escrow release conditions often dovetail with the applicable service-level agreements. If an SLA provides for reasonable remedies for short periods of unplanned downtime, then the SaaS provider can argue that the escrow should only be triggered by longer periods of unplanned downtime or chronic failures.

Although there are several points of negotiation in these SaaS escrow provisions, providers are more and more frequently accepting the reality of SaaS escrow, including them in their form subscription agreements to appeal to prospective customers wary of business continuity risks.

From the customers' standpoint, they must assess (1) whether the application is mission critical; (2) the cost from both a financial and reputational perspective of going down; (3) the availability of substitute applications; (4) the transition time to a substitute application; and (5) the stability and reliability of the vendor.

Even as SaaS escrow provisions become customary in vendor agreements, the question of their effectiveness remains. While the escrow can give customers comfort when taking on the risk of onboarding an SaaS solution, the actual, practical transitioning of the application, data center and hosting environment in the event of a release condition may be more catastrophic than the downtime itself. It's time for customers to make sure they "cover their SaaS."

Mark Kesslen is a partner and chair of the intellectual property section of the tech group at Lowenstein Sandler LLP.

Leah Satlin is counsel at the firm.

The opinions expressed are those of the author(s) and do not necessarily reflect the views of the firm, its clients, or Portfolio Media Inc., or any of its or their respective affiliates. This article is for general information purposes and is not intended to be and should not be taken as legal advice.