

# POURING NEW WINE INTO OLD WINESKINS: WHY “ON PREMISE” SOFTWARE SOURCE CODE ESCROW ARRANGEMENTS ARE ILL- SUITED FOR REMOTELY HOSTED “OFF PREMISE” SOFTWARE AS A SERVICE LICENSE AGREEMENTS

KINGSLEY OSEI\*

|  |     |
|--|-----|
| I. INTRODUCTION .....  | 384 |
| II. OLD WINESKINS .....  | 386 |
| A. On Premise Software Escrow Arrangements .....                         | 386 |
| B. The Utility of Source Code Escrow Arrangements .....                  | 387 |
| C. Are Source Code Escrow Arrangements Worth The<br>Trouble?.....        | 388 |
| III. SAAS LICENSES AND UNIQUE ISSUES PERTAINING TO THEIR<br>ESCROW ..... | 390 |
| A. Evolution of SaaS Licenses .....                                      | 390 |
| B. SaaS Licensing and Pricing Models .....                               | 392 |
| C. Disruptive Risks Inherent In SaaS Licensing .....                     | 392 |
| D. Special Bankruptcy Considerations .....                               | 393 |
| 1. SaaS Licensing Agreements as Executory Contracts .....                | 393 |
| 2. Section 365(n) to the Rescue? .....                                   | 396 |
| IV. NEW WINESKINS.....   | 397 |
| A. SaaS Escrow Arrangements.....   | 397 |
| 1. Appropriate Escrow Release Triggers for SaaS Licenses ...             | 397 |
| 2. Addressing Licensing and Subscription Issues.....                     | 398 |
| 3. Dealing with Licensor’s Bankruptcy .....                              | 398 |
| B. Alternatives to Escrow Arrangements.....                              | 400 |
| 1. Service Level Agreements and Failover Guarantees.....                 | 400 |
| 2. Disaster Recovery Plans .....   | 401 |
| 3. SaaS Application Interoperability and Portability .....               | 402 |
| V. CONCLUSION.....   | 402 |

---

\* Kingsley S. Osei (Kingsley.osei@suny.edu) is Contracts Counsel for the State University of New York System Administration in Albany, New York.

## I. INTRODUCTION

Many colleges and universities are adopting the “Software as a Service” (“SaaS”) licensing model, a variant of cloud computing,<sup>1</sup> to reduce their information technology (“IT”) acquisition costs and to create efficient on-demand IT systems that are delivered as a service rather than as a product.<sup>2</sup>

---

1. Cloud computing is defined by the National Institute of Standards as “[A] model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” See Peter Mell & Tim Grance, *The NIST Definition of Cloud Computing*, Special Publication 800-145, NAT’L INSTIT. OF STANDS. AND TECH., 2 (Sept. 2011), <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.

Cloud computing exists in different iterations such as: Platform as a Service (“PaaS”), Infrastructure as a Service (“IaaS”), and Software as a Service (“SaaS”). See ORACLE, *Achieving the Cloud Computing Vision*, 5 (Oct. 2010), <http://www.oracle.com/technetwork/topics/entarch/architectural-strategies-for-cloud—128191.pdf>.

In PaaS, the provider allows the consumer to deploy the consumer’s own or acquired software applications using programming tools supported by the provider into the provider’s cloud infrastructure. *Id.* The consumer does not control the underlying cloud infrastructure including the network, servers, etc., but has control over its deployed applications. Examples of this model are Amazon’s “Elastic Compute Cloud” or “EC2 Utility Cloud Platform” and Google’s “App Engine.” *Id.*

In IaaS, the consumer is allowed to deploy into the cloud infrastructure using the provider’s processing, storage, networks, and other fundamental computing resources and is able to deploy and run arbitrary software, which can include operating systems and applications. *Id.* The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications, etc. *Id.*

In SaaS, the licensee or subscriber uses the provider’s applications running on a cloud infrastructure. *Id.* The applications are accessible from various client devices through the internet. *Id.* The licensee or subscriber does not manage or control the underlying cloud infrastructure including networks, servers, operating systems, storage, etc. *Id.*

2. Notwithstanding the widely held notion of software as a commodity, such categorization is still contentious, given the ramification of the application of UCC warranties. In *Triangle Underwriters, Inc. v. Honeywell, Inc.*, a contract for the development of a computer system with custom-designed software, installation, training, and maintenance was considered a sale of a “good.” 457 F.Supp. 765, 769 (E.D.N.Y. 1978), *aff’d in part, rev’d in part and remanded on other grounds*, 604 F.2d 737 (2d Cir. 1979). See also *Confer Plastics, Inc. v. Hunkar Lab., Inc.*, 964 F. Supp. 73, 77 (W.D.N.Y. 1997) (“Under the law of New York, the transaction at issue constitutes a sale of goods.” The court rationalized that the sale of computer software, though an intangible item, was more readily characterized as “goods” than “services.”); *Architectronics, Inc. v. Control Sys., Inc.*, 935 F.Supp. 425, 432 (S.D.N.Y. 1996).

In *Communications Group, Inc. v. Warner Communications, Inc.*, software was considered a good even though a finished software product may reflect a substantial investment of programming services. 527 N.Y.S.2d 341 (N.Y. Civ. Ct. 1988).

A computer software package agreement including a license for limited use of copyrightable information, installation, and a warranty service package was an agreement for the sale of goods under the New York UCC. *Id.* Similarly, a contract to

SaaS licensing offers enormous benefits to colleges and universities. Some of the benefits include: (1) ubiquity of access from fixed or mobile devices; and (2) “pay as you go” expense payments without massive upfront hardware and software installation costs. SaaS also enables licensees to leverage the flexibility and scalability of the internet.<sup>3</sup> A recent survey<sup>4</sup> found that about thirty-four percent of colleges and universities in the United States have, or are in the process of implementing, a SaaS or cloud computing solution.<sup>5</sup> Another report by an on-demand financial management, human capital management, and cloud computing software vendor lists Brown, Cornell, Carnegie Mellon, and Georgetown as major clients that are migrating their human resources, payroll, financial, and other administrative IT systems from locally run computing resources to the cloud.<sup>6</sup> A less obvious, but equally significant trend is the ongoing migration by many colleges and universities of their student email, calendaring and social networking applications to Yahoo, Google, and Microsoft, among many others.<sup>7</sup> “Google Apps for Education” and

---

develop specifications for the delivery, modification, installation, testing, licensing, and implementation of a computer software system for international bank management was a “good” under New York law. See *Schroders, Inc. v. Hogan Sys., Inc.*, 522 N.Y.S.2d 404 (N.Y. Sup. Ct. 1987). See also RIVERBED, *Data Storage in the Cloud: Can You Not Afford It?*, [http://docs.media.bitpipe.com/io\\_10x/io\\_100958/item\\_431220/Data%20Storage%20in%20the%20Cloud\\_Can%20you%20afford%20WW%20ROI%20Whitepaper.pdf](http://docs.media.bitpipe.com/io_10x/io_100958/item_431220/Data%20Storage%20in%20the%20Cloud_Can%20you%20afford%20WW%20ROI%20Whitepaper.pdf).

3. Mell & Grance, *supra* note 1; see also Gianpaolo Carraro & Fred Chong, *Software as a Service (SaaS): An Enterprise Perspective* (Oct. 2006), <http://msdn.microsoft.com/en-us/library/aa905332.aspx> (last visited Feb. 11, 2013).

4. CDW-G is a subsidiary of CDW Corp., a reseller of hardware, software, and supplies that is devoted to government clients in the United States.

5. See David Nagel, *Campus IT Plans for Increased Cloud Adoption* (Oct. 26, 2011), <http://campustechnology.com/articles/2011/05/26/campus-it-plans-for-increased-cloud-adoption.aspx> (last visited Feb. 11, 2013).

6. See Workday and Brown University, *A Unified Approach Brings New Insight*, WORKDAY, <http://www.workday.com/Documents/pdf/case-studies/workday-brown-university-case-study.pdf>; Workday and Cornell University, *Efficiency Meets Individuality*, WORKDAY, <http://www.workday.com/Documents/pdf/case-studies/workday-cornell-case-study.pdf>; Workday and Carnegie Mellon University, *Customer Profile*, WORKDAY, [http://www.workday.com/customers/by\\_industry/education\\_and\\_government/carnegie\\_mellon\\_university.php](http://www.workday.com/customers/by_industry/education_and_government/carnegie_mellon_university.php); Workday and Georgetown University, *Unified Administrative Operations*, WORKDAY, <http://www.workday.com/Documents/pdf/customer-profiles/customer-profile-georgetown-university.pdf>.

7. Also known by various monikers such as “Cloud-sourcing, Outsourcing, Consortial sourcing, Institutional sourcing, Collaborative sourcing . . . technologies and IT services are being delivered to colleges and universities in a myriad of ways. Whereas in the past the role of the IT organization was to provide IT services to the campus community—known (now) as insourcing—over time that role has subtly but concretely changed. IT leaders today must not only provide but also decide: which tools and services should they continue to supply, which are better delivered by others, and perhaps most critically, which methods from among the bewildering array of alternative sourcing strategies will best serve their faculty, staff, and students.” Edward

“Microsoft Live@EDU” rebranded as “Microsoft Office 365 for Education” are online hosted, co-branded communication and collaboration services designed for students, faculty, and alumni to provide cloud based email, document sharing and storage, as well as enterprise class tools.<sup>8</sup>

This article examines the inadequacies of the on premise/product oriented software escrow protections frequently relied upon by colleges and universities in their SaaS acquisitions when a software licensor or SaaS provider files for bankruptcy, ceases to support or maintain the software application, or experiences a disruptive application access event that prevents the licensee’s use of the licensed software.<sup>9</sup> Significantly, this article will offer alternative considerations that SaaS licensees may take into account to optimally address an application access disruption event, including, but not limited to, portability and disaster recovery issues.

## II. OLD WINESKINS

### A. On Premise Software Escrow Arrangements

On premise software, as distinguished from SaaS, is software licensed by physical delivery and installation on the servers or computer systems at a licensee’s place of business.<sup>10</sup> In this model, the licensor is often required to deposit its proprietary source code and related documentation with a neutral third party or escrow agent for the benefit of the licensee, should the licensor file for bankruptcy, fail to support the software, or cease to do business, among other release events.<sup>11</sup> This corollary escrow deposit arrangement offers the licensee a secure, confidential, and dependable mechanism to access the licensed software source code.<sup>12</sup> From a

---

Mahon et al., *Alternative IT Sourcing Strategies: Six Views*, 15 (2011), <http://net.educause.edu/ir/library/pdf/ERM1140.pdf>.

8. Microsoft in Education, *Microsoft Live@edu*, MICROSOFT, <http://www.microsoft.com/education/en-us/solutions/Pages/liveedu.aspx> (last visited Feb. 11, 2013).

9. See RICHARD RAYSMAN & PETER BROWN, *COMPUTER LAW: DRAFTING AND NEGOTIATING FORMS AND AGREEMENTS* 7-58 (Release 54, 2012). See also Mary A. Moy, *The Intellectual Property Bankruptcy Protection Act: An Unbalanced Solution to the International Software Licensing Dilemma*, 11 U. PA. J. INT’L BUS. L. 151, 172 (1989).

10. H. WARD CLASSEN, *A PRACTICAL GUIDE FOR SOFTWARE LICENSES FOR LICENSORS AND LICENSEES* 143 (3d ed. 2008).

11. See RAYSMAN & BROWN, *supra* note 9, at 7-58; see also Periklis A. Pappous, *The Software Escrow: The Court Favorite and Bankruptcy Law*, 1 SANTA CLARA COMPUTER & HIGH TECH. L.J. 309, 309 (1985).

12. A “source code is the set of instructions as written by the computer programmer in the appropriate programming (computer) language. . . . which are used to direct computer functions.” RAYSMAN & BROWN, *supra* note 9, at 1-18.

See also MTH 568 - Computational Science, *Source Code, Compiling, and Executable Code*, UNIVERSITY AT BUFFALO, [http://www.nsm.buffalo.edu/courses/mth568/www/intro\\_to\\_c.html](http://www.nsm.buffalo.edu/courses/mth568/www/intro_to_c.html) (last visited Feb. 11, 2013) (“[The program] that is

licensee's perspective, a good source code escrow deposit arrangement should be accompanied by documentation that describes the source code's structure or dataflow to enable the programmer to decipher, enhance, or modify the software program to ensure its continuous use by the licensee.<sup>13</sup> An effective escrow deposit arrangement should also incorporate a verification mechanism that will enable the licensee to periodically verify that the materials on deposit are completed, contain updated versions of the software, and capable of being compiled into an object code version of that software.<sup>14</sup> The source code or software license agreement should also have appropriate release events<sup>15</sup> suited to the business or operational objectives for which the software was licensed.<sup>16</sup>

### B. The Utility of Source Code Escrow Arrangements

As the foregoing clarifies, source code arrangements offer continuity of access to the licensee should any of the above listed release events occur.<sup>17</sup> The scope of the licensor's duties and licensee's rights for the release of the code is often encapsulated in licensing or escrow agreements. Essentially, a software source code deposit arrangement is a risk aversion and business continuity mechanism that permits a licensee to survive disruptions in use of the software if the licensor cannot provide, maintain, and update the software in accordance with the terms of the licensing agreement.<sup>18</sup>

---

written by a human using a text editor . . . This code is easily written and read by humans and can be translated into a code that can be executed by a computer. Source code is converted to *executable code* (which is useful to the computer) by a process called *compiling*. . . The source code can be transformed into executable code by using the GNU C compiler (gcc). The executable or machine code produced is named whatever you wish using the '-o' flag of the gcc compiler. For convenience, the executable is usually named the same name as the source code (without the '.c' suffix). The executable code can then be run on the machine by typing ./ (to tell the shell that the program is located in the current directory) followed by the name of the executable code").

13. RAYSMAN & BROWN, *supra* note 9, at 7-58.

14. *Id.*

15. Pappous, *supra* note 11, at 309; *see also* Shawn Helms & Alfred Cheng, *Source Code Escrow: Are You Just Following the Herd?* (Feb. 25, 2008), [http://www.jonesday.com/files/Publication/09e73d7d-7240-450f-ae18-07970d03977f/Presentation/PublicationAttachment/7cbcbfeb-8cf4-4056-9c89-11c07b903a73/JD\\_Source%20Code%20Escrow.pdf](http://www.jonesday.com/files/Publication/09e73d7d-7240-450f-ae18-07970d03977f/Presentation/PublicationAttachment/7cbcbfeb-8cf4-4056-9c89-11c07b903a73/JD_Source%20Code%20Escrow.pdf).

16. *See* RAYSMAN & BROWN, *supra* note 9, at 7-58.

17. *Id.*

18. *Id.* *See also* John Boruka, *Trust But Verify: How Verification Services Can Put Teeth in Your Technology Escrow Agreement*, IRON MOUNTAIN, <https://secured.ironmountain.com/Knowledge-Center/Reference-Library/View-by-Document-Type/White-Papers-Briefs/Sponsored/Trust-But-Verify.aspx> (last visited Feb. 11, 2013).

### C. Are Source Code Escrow Arrangements Worth The Trouble?

Despite its enormous benefits, anecdotal evidence indicates that software escrow arrangements are often fraught with problems and unmet expectations. Even though most software source codes are annotated<sup>19</sup> to enable a programmer to “decipher” and make the software useful for the licensee, licensees often run into the arduous reality that “decoding” reams of computer programming language and documentation written by others and further, to compile and transform executable source code into machine readable object code, is not always easy.<sup>20</sup> Often times, source code release event(s) stipulated in licensing agreements do not correlate with the objectives of the parties in “escrowing” the source code. Though a licensee’s objective in escrowing a source code may be to ensure the software’s continued maintenance, it is not uncommon to find deposit provisions that are devoid of such provisions, but replete with other needless release events such as bankruptcy, ceasing to do business, etc.<sup>21</sup> For instance, if a licensee’s objective in escrowing the source code is not ongoing maintenance of the software because of its in-house capabilities, it will be foolhardy to premise the release of the source code on the licensor’s failure to maintain the software.<sup>22</sup> Further, incorporating a release event such as “ceasing to do business” may be moot since the licensor may technically have “ceased to do business” and there may be a continuity of software maintenance undertaken by an acquiring company.<sup>23</sup> Other practical considerations may make the release of the software irrelevant. A licensor in the last throes of shutting down its business or on the verge of bankruptcy may not have updated the source code in escrow due to staffing issues. It may not have fixed bugs to ensure compatibility with other system upgrades or added functionalities required for the licensee’s evolving or changing business needs.<sup>24</sup>

---

19. Annotation of software is mainly used for expanding code documentation and offering commentary on how the software code is written. “The source code will incorporate the programmer’s comments to provide a guide as to the specific function being performed by the program at its various steps. These [source code] comments can be read by a programmer literate in the particular computer language to understand the program and the techniques used by the original programmer in solving the particular problem.” See RAYSMAN & BROWN, *supra* note 9, at 1-19.

20. J.T. Westermeier, *Strategies Relating to the U.S. Bankruptcy Laws and Information Technology*, FINNEGAN HENDERSON FARABOW GARRETT & DUNNER LLP (Jan. 2010), <http://www.finnegan.com/resources/articles/articlesdetail.aspx?news=6db94cca-c0d5-47e3-8a72-01d660722f74>.

21. See Linda Markus Daniels, *Does Your Source Code Escrow Agreement Achieve Its Objectives?*, [http://www.innovasafe.com/pdf/Does%20Your%20Source%20Code%20Escrow%20Agreement%20Achieve%20Its%20Objectives\\_Linda%20Markus%20Daniels.pdf](http://www.innovasafe.com/pdf/Does%20Your%20Source%20Code%20Escrow%20Agreement%20Achieve%20Its%20Objectives_Linda%20Markus%20Daniels.pdf).

22. *Id.*

23. *Id.*

24. See RAYSMAN & BROWN, *supra* note 9, at 7-62 & 7-63.

It is no wonder that statistically, only a small percentage of software source code escrows are ever released.<sup>25</sup> According to Iron Mountain,<sup>26</sup> a provider of information storage and management solutions, between 1990 and 1999, only ninety-six escrow accounts out of more than forty-five thousand escrowed software accounts with the company were released.<sup>27</sup> This low release rate perhaps reflects the rarity of the occurrence of release events. This may then beg the question whether the return on the expense of putting escrow arrangements in place is matched by perceived or anticipated risks that escrow arrangements seeks to insure. Also, it is not uncommon to find escrowed source codes to be outdated, defective, or failing to meet ongoing requirements. Again, according to Iron Mountain, 97.4% of all analyzed escrow material deposits are incomplete and 74% have required additional input from developers in order to be compiled. Even if the software is updated, the question remains whether a licensee has in-house resources or capability to utilize the code upon release.<sup>28</sup> This situation is often complicated by prohibitions against soliciting licensor's employees upon termination of the licensing agreement, cutting off a valuable use and continuity resource necessary to maintain and support the software.

Even though an escrow agreement may provide clear triggering events for the release of source code,<sup>29</sup> a licensor may still have the ability to block release to the licensee if the agreement provides for licensor's consent or fiat to release the source code. A delay in release of the software source code at the behest of an economically challenged licensor may further prevent requisite updates to the software.<sup>30</sup>

Furthermore, as will be discussed later in this article, if bankruptcy is the major and sometimes only, triggering condition authorizing release of the source code from escrow, the bankruptcy court's broad powers over the vendor-debtor estate makes removal from escrow after bankruptcy a risky proposition.<sup>31</sup> The vendor trustee may be able to reject such a contract as "executory" and obtain a court order to remove the source code copy from escrow leaving the user without the anticipated access to source code.<sup>32</sup>

---

25. See Jonathan L. Mezrich, *Source Code Escrow: An Exercise in Futility*, 5 MARQ. INTELL. PROP. L. REV. 117, 121–22 (2001). See also Helms & Cheng, *supra* note 15.

26. Boston, Massachusetts based information destruction and data backup services reputed to be the largest provider of source code escrow services in the United States.

27. See Helms & Cheng, *supra* note 15.

28. *Id.*

29. *Id.*

30. *Id.*

31. See RAYSMAN & BROWN, *supra* note 9, at 7-29 to -32.3.

32. *Id.*

### III. SAAS LICENSES AND UNIQUE ISSUES PERTAINING TO THEIR ESCROW

#### A. Evolution of SaaS Licenses

As described above, SaaS is computer software deployed offsite and predominantly licensed as a service over the internet.<sup>33</sup> SaaS was first defined by an article titled, “Strategic Backgrounder: Software as a Service,” published by the Software and Information Industry Association (SIIA) e-Business Division.<sup>34</sup> The intent of this article was to dispel confusion between SaaS and similar licensing models:

In the software as a service model, the application, or service, is deployed from a centralized data center across a network—Internet, Intranet, LAN, or VPN—providing access and use on a recurring fee basis. Users “rent,” “subscribe to,” “are assigned,” or “are granted access to” the applications from a central provider. Business models vary according to the level to which the software is streamlined, to lower price and increase efficiency, or value-added through customization to further improves digitized business processes.<sup>35</sup>

Historically, the SaaS licensing model is the progeny of earlier client server legacy applications<sup>36</sup> marketed by Application Service Providers (ASPs) in the 1980s.<sup>37</sup> This ASP application model, which was billed as an alternative to on-premise software,<sup>38</sup> did not live up to expectations because it was in the most part, cumbersome to use legacy applications not designed to leverage the scalability of the internet.<sup>39</sup> As the number of ASPs

---

33. See *supra* note 1 and accompanying text.

34. “The Software & Information Industry Association is the principal trade association for the software and digital content industry. SIIA provides global services in government relations, business development, corporate education and intellectual property protection to the leading companies that are setting the pace for the digital age.” SOFTWARE & INFORMATION INDUSTRY ASSOCIATION, <http://www.siiia.net> (last visited Feb. 11, 2013).

35. Fred Hoch et al., *Software as a Service: Strategic Backgrounder*, SOFTWARE & INFORMATION INDUSTRY ASSOCIATION, 4 (Feb. 2001), <http://www.siiia.net/estore/ssb-01.pdf>.

36. A legacy application is “[a]n application that was written for an earlier operating system or hardware platform. For example, mainframe applications were legacy apps when the world embraced client/server networks. Windows 3.1 applications were legacy apps when Windows 95 was introduced. . . . Any business software that is not Internet enabled in some form is sometimes considered a legacy application.” *PC Magazine Encyclopedia: legacy application*, PC MAGAZINE, [http://www.pcmag.com/encyclopedia\\_term/0,2542,t=legacy+application&i=46019,00.asp#fbid=MEKI50BJ2vY](http://www.pcmag.com/encyclopedia_term/0,2542,t=legacy+application&i=46019,00.asp#fbid=MEKI50BJ2vY) (last visited Feb. 11, 2013).

37. See Service-now.com, *A Brief history of SaaS: Modernizing Enterprise Software*, 3–4, <http://www.computerworld.com/pdfs/Service-Now-BriefhistoryofSaaS.pdf> (last visited Feb. 13, 2013).

38. See CLASSEN, *supra* note 10, at 143.

39. See THINKstrategies, Inc., *A Whitepaper for SaaS Customers and Vendors*:

proliferated, the cost and challenge of delivering these client server applications became cumbersome, time consuming, and expensive. These inherent limitations and failure to deliver the promised efficiencies and comparative cost advantage to on-premise software led to the demise of these earlier client server legacy models.<sup>40</sup>

The advent of the internet saw the revival of on-demand software licensing and other externally delivered IT platform or infrastructural services. According to a recent study conducted by International Data Corporation (“IDC”),<sup>41</sup> a provider of market intelligence and advisory services to information technology and telecommunications companies, SaaS had worldwide sales of \$13.1 billion in 2009 and forecasts that sales of SaaS could reach \$40.5 billion by 2014.<sup>42</sup> IDC further predicts that by 2014, about 34% of all new business software licenses will be SaaS models, and SaaS delivery will constitute about 14.5% of worldwide software sales.<sup>43</sup> Put succinctly by the Vice President for SaaS and Cloud Services Research at IDC:<sup>44</sup>

“[E]nterprise IT plans are rapidly shifting to accommodate the growing choices for sourcing most or all IT software functions, from business applications to software development and testing, to service and desktop management, as SaaS services become available from established vendors and new models for accessing functionality in the cloud creates lower-cost options and more tailored models for consuming IT services.”<sup>45</sup>

As pointed out earlier, SaaS models seek to save the licensee the up-front installation and maintenance costs<sup>46</sup> associated with on-premise software licensing whilst allowing the licensee “to take advantage of the benefits of centralization through a single-instance, multi-tenant architecture, and to provide a feature-rich experience competitive with

---

*Leveraging Escrow Agreements to Safeguard Your Business from the Risks Associated with Software-as-a-Service*, IRON MOUNTAIN, 5 (2007), <http://ironmountain.com/resources/services/doc/us-tes-wp-leveraging-escrow-saas-think.pdf>.

40. *Id.* at 4.

41. See INTERNATIONAL DATA CORPORATION, <http://www.idc.com> (last visited Feb. 11, 2013).

42. *SaaS Revenue to Grow Five Times Faster than Traditional Packaged Software Through 2014, IDC Finds*, BUSINESS WIRE (July 26, 2010), <http://www.businesswire.com/news/home/20100726005135/en/SaaS-Revenue-Grow-Times-Faster-Traditional-Packaged>.

43. *Id.* Another study by Gartner predicts worldwide revenue for SaaS delivery to grow from 2008 to 2013 by 19.4% overall. See Alexander Benlian et al., *The Risks of Sourcing Software as a Service—An Empirical Analysis of Adopters and Non Adopters*, INSTIT. FOR INFO. SYS., 2, <http://is2.lse.ac.uk/asp/aspecis/20100026.pdf>.

44. See Analyst Profile: Robert P. Mahowald, INT’L DATA CORP., <http://www.idc.com/getdoc.jsp?containerId=PRF000230> (last visited Feb. 11, 2013).

45. BUSINESS WIRE, *supra* note 42.

46. See Carraro & Chong, *supra* note 3.

comparable on-premise applications.”<sup>47</sup>

### B. SaaS Licensing and Pricing Models

Whether offered directly by a vendor or through an aggregator,<sup>48</sup> SaaS applications are often sold or licensed by subscription, with licensee, or customers paying as they use the software.<sup>49</sup> It is noteworthy that within the SaaS subscription-based licensing genre, there are sub-licensing categories including: (a) pure subscriptions-based models, where a monthly payment is calculated on the software actually used, considering the actual number of users; (b) usage-based models, where payment is determined by application usage and is typically related to peak or near peak levels of usage; (c) transaction-based models, where the licensor or provider sometimes charges customers for each business transaction; and (d) value-based or shared risk or revenue models, which are based on the provision of whatever software is needed to achieve business goals, and payment linked to the achievement of those goals.<sup>50</sup> The latter is a fixed-fee model where “users generally pay a predetermined monthly fee based on number of users supported, which application modules are rented and service and support levels specified by the customer.”<sup>51</sup>

### C. Disruptive Risks Inherent In SaaS Licensing

Because they are mainly licensed by subscription, hosted offsite, and delivered through the internet, the crucial risk factors inherent in SaaS licensing are material application access interruptions, telecommunication and power outages, or considerable downtimes due to technical

---

47. *Id.* It is also instructive to note that whilst multi-tenancy is the norm in most SaaS offerings, a new generation of SaaS licensing, which offers “single tenancy” architecture, is being explored. The “[n]ext generation SaaS providers are delivering single-tenant SaaS while keeping costs low through data center automation and virtualization. Each customer receives a unique instance of the application and database. Single-tenant SaaS applications are fully supported by the application experts—the software vendor. Fully automated upgrades are scheduled throughout the year and remain invisible to the end user. Single-tenancy also gives the customer additional opportunity for extensive customization, data security compliance peace of mind, more deployment options, and preserved customizations through upgrades while providing the same user benefits that are driving the popularity of SaaS.” *See Service-now.com, supra* note 37, at 4.

48. An aggregator is a software intermediary that bundles offerings from different vendors and offers them as part of a unified application platform. *See Carraro & Chong, supra* note 3.

49. *See CLASSEN, supra* note 10, at 143; *see also* Tommy van de Zande & Slinger Jansen, *Business Continuity Solutions for SaaS Customers*, UTRECHT UNIVERSITY DEPARTMENT OF INFORMATION AND COMPUTING SCIENCES, [http://slingerjansen.files.wordpress.com/2009/04/businesscontinuitysolutions\\_2-1.pdf](http://slingerjansen.files.wordpress.com/2009/04/businesscontinuitysolutions_2-1.pdf).

50. *See* Hoch, *supra* note 35, at 11.

51. *Id.*

difficulties.<sup>52</sup> A recent example of such an incident occurred on April 21, 2011, when Amazon's Elastic Compute Cloud ("Amazon EC2"), which is the online retailer's utility computing platform and also, a significant cloud services provider, suffered an outage that wiped out some customer application data. Without the escrow of the data and back-up, some developers who hosted their application software on Amazon EC2 irrecoverably lost data and were not able to restart their applications when the system was restored on April 23, 2011.<sup>53</sup>

Because in SaaS licensing the computer hardware, servers, storage devices, and cloud infrastructure belong to the licensor and licensee data may be hosted offsite, structuring an escrow arrangement merely to access source code to recreate the functionality of the software may be of limited practical utility to a licensee.<sup>54</sup> Moreover, SaaS solutions may have a multi-tenant infrastructure serving more than one licensee or subscriber at the same time.<sup>55</sup> This creates unique back-up and data recovery challenges if a release condition should occur.<sup>56</sup> SaaS licensing therefore presents unique business continuity and operational challenges for which the adoption of an escrow model, which is best suited for on-premise and product oriented licensing application and merely allows the licensee to access and restore the functioning of proprietary software, may not be appropriate for the significant purposes it is designed to serve for the licensee.<sup>57</sup>

#### D. Special Bankruptcy Considerations

##### 1. SaaS Licensing Agreements as Executory Contracts

Aside from the disruptive operational considerations discussed above, the bankruptcy of the licensor may also present special continuity of access and intellectual property matters worth mentioning. Prior to the addition of section 365(n) to the United States Bankruptcy Code,<sup>58</sup> a Chapter 11 bankruptcy trustee<sup>59</sup> for a licensor ("licensor"), as a debtor in possession,<sup>60</sup>

---

52. See van de Zande & Jansen, *supra* note 49.

53. See Rich Miller, *Amazon EC2 Outage Wipes Out Data*, DATA CENTER KNOWLEDGE (Oct. 2, 2007), <http://www.datacenterknowledge.com/archives/2007/10/02/amazon-ec2-outage-wipes-out-data> (last visited Feb. 11, 2013).

54. See Rebecca Anderson, *Key Issues in SaaS Contracts*, KEMP LITTLE, LLP (June 2010), <http://www.kemplittle.com/Publications/item.aspx?ListName=Short%20Lines&ID=52#.URnRX1qb-4Q>; see also *Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region*, AMAZON WEB SERVICES (April 29, 2011), <http://aws.amazon.com/message/65648>.

55. See ORACLE, *supra* note 1.

56. THINKstrategies, Inc., *supra* note 39, at 7.

57. See van de Zande & Jansen, *supra* note 49, at 10.

58. 11 U.S.C.A. § 365 (West 2012).

59. ADMIN. OFF. OF THE US COURTS, *The U.S. trustee or bankruptcy*

---

*administrator, Reorganization Under the Bankruptcy Code*, <http://www.uscourts.gov/FederalCourts/Bankruptcy/BankruptcyBasics/Chapter11.aspx> (last visited Feb. 11, 2013) (citations omitted).

The U.S. trustee plays a major role in monitoring the progress of a chapter 11 case and supervising its administration. The U.S. trustee is responsible for monitoring the debtor in possession's operation of the business and the submission of operating reports and fees. Additionally, the U.S. trustee monitors applications for compensation and reimbursement by professionals, plans and disclosure statements filed with the court, and creditors' committees. The U.S. trustee conducts a meeting of the creditors, often referred to as the "section 341 meeting," in a chapter 11 case. The U.S. trustee and creditors may question the debtor under oath at the section 341 meeting concerning the debtor's acts, conduct, property, and the administration of the case.

The U.S. trustee also imposes certain requirements on the debtor in possession concerning matters such as reporting its monthly income and operating expenses, establishing new bank accounts, and paying current employee withholding and other taxes. By law, the debtor in possession must pay a quarterly fee to the U.S. trustee for each quarter of a year until the case is converted or dismissed. The amount of the fee, which may range from \$250 to \$10,000, depends on the amount of the debtor's disbursements during each quarter. Should a debtor in possession fail to comply with the reporting requirements of the U.S. trustee or orders of the bankruptcy court, or fail to take the appropriate steps to bring the case to confirmation, the U.S. trustee may file a motion with the court to have the debtor's chapter 11 case converted to another chapter of the Bankruptcy Code or to have the case dismissed.

60. ADMIN. OFF. OF THE US COURTS, *The Chapter 11 Debtor in Possession, Reorganization under the Bankruptcy Code*, UNITED STATES COURTS, <http://www.uscourts.gov/FederalCourts/Bankruptcy/BankruptcyBasics/Chapter11.aspx> (last visited Feb. 11, 2013) (citations omitted).

Chapter 11 is typically used to reorganize a business, which may be a corporation, sole proprietorship, or partnership. A corporation exists separate and apart from its owners, the stockholders. The chapter 11 bankruptcy case of a corporation (corporation as debtor) does not put the personal assets of the stockholders at risk other than the value of their investment in the company's stock. A sole proprietorship (owner as debtor), on the other hand, does not have an identity separate and distinct from its owner(s). Accordingly, a bankruptcy case involving a sole proprietorship includes both the business and personal assets of the owners-debtors. Like a corporation, a partnership exists separate and apart from its partners. In a partnership bankruptcy case (partnership as debtor), however, the partners' personal assets may, in some cases, be used to pay creditors in the bankruptcy case or the partners, themselves, may be forced to file for bankruptcy protection.

Section 1107 of the Bankruptcy Code places the debtor in possession in the position of a fiduciary, with the rights and powers of a chapter 11 trustee, and it requires the debtor to perform of all but the investigative functions and duties of a trustee. These duties, set forth in the Bankruptcy Code and Federal Rules of Bankruptcy Procedure, include accounting for property, examining and objecting to claims, and filing informational reports as required by the court and the U.S. trustee or bankruptcy administrator (discussed below), such as monthly operating reports. The debtor in

could pursuant to section 365(a) assume or reject<sup>61</sup> a software licensing agreement with a licensee as an executory contract.<sup>62</sup> If with the bankruptcy court's approval, the licensor rejected the contract, the licensee was deemed to be in breach of the contract. The licensee's substantive relief was to seek damages as an unsecured creditor of the licensor and is entitled to damages for a pre-petition claim in bankruptcy payable at whatever fractional rate the court determines for such a claim.<sup>63</sup> If the licensed software had a source code in escrow, the licensee's ability to access the source code was curtailed by its inability to sue for access because the Bankruptcy Code barred the specific performance of a rejected executory intellectual property agreement.<sup>64</sup>

---

possession also has many of the other powers and duties of a trustee, including the right, with the court's approval, to employ attorneys, accountants, appraisers, auctioneers, or other professional persons to assist the debtor during its bankruptcy case. Other responsibilities include filing tax returns and reports which are either necessary or ordered by the court after confirmation, such as a final accounting. The U.S. trustee is responsible for monitoring the compliance of the debtor in possession with the reporting requirements.

Railroad reorganizations have specific requirements under subsection IV of chapter 11, which will not be addressed here. In addition, stock and commodity brokers are prohibited from filing under chapter 11 and are restricted to chapter 7.

61. 11 U.S.C.A. § 365(a) (West 2012).

62. "Executory contract" is not defined in the Bankruptcy Code, but the most widely accepted definition, which was also adopted in *Lubrizol Enters. Inc. v. Richmond Metal Finishers, Inc.*, 756 F.2d 1043 (4th Cir. 1985), is the one postulated by Professor Vern Countryman: that "under which the obligation of both the bankrupt and the other party to the contract are so far unperformed that the failure of either to complete the performance would constitute a material breach excusing the performance of the other." (quoting Vern Countryman, *Executory Contracts in Bankruptcy*, 57 MINN. L. REV. 439, 460 (1973)). See also *Gloria Mfg. Corp. v. Int'l Ladies Garment Workers Union*, 734 F.2d 1020, 1022 (4th Cir. 1984); *Fenia Cattle Co. v. Silver*, 625 F.2d 290, 292 (9th Cir. 1980).

63. The 4th Circuit in the seminal case of *Lubrizol Enters. Inc. v. Richmond Metal Finishers, Inc.*, 756 F.2d 1043 (4th Cir. 1985), said in dicta that "[u]nder 11 U.S.C. § 365(g), Lubrizol would be entitled to treat rejection as a breach and seek a money damages remedy; however, it could not seek to retain its contract rights in the technology by specific performance even if that remedy would ordinarily be available upon breach of this type of contract." *Id.* at 1048. See also Jennifer S. Bisk, *Software Licenses Through the Bankruptcy Looking Glass: Drafting Individually Negotiated Software Licenses that Protect the Client's Interests in Bankruptcy*, 17 FORDHAM INTELL. PROP. MEDIA & ENT. L.J. 611, 630 (2006); Daniel T. Brooks, *Intellectual Property Bankruptcy Protection Act of 1988*, 272 PLI/Pat 575, 607 (1989).

64. 11 U.S.C.A. § 365(n) (West 2012); See also Moy, *supra* note 9, at 169; 11 U.S.C.A. § 542(a) (West 2012) ("Except as provided in subsection (c) or (d) of this section, an entity, other than a custodian, in possession, custody, or control, during the case, of property that the trustee may use, sell, or lease under section 363 of this title, or that the debtor may exempt under section 522 of this title, shall deliver to the trustee, and account for, such property or the value of such property, unless such property is of inconsequential value or benefit to the estate.").

## 2. Section 365(n) to the Rescue?

The affirmation of the above inequities of section 365(a) wrought on the licensee by the Fourth Circuit in *Lubrizol v. Richmond Metal Finishers*<sup>65</sup> and the refusal of the Supreme Court to grant certiorari led Congress to pass section 365(n), to strike a balance between licensor rehabilitation in bankruptcy and the need to preserve a licensee's contractual rights to the software or other "qualified intellectual property."<sup>66</sup>

Section 365(n) provides in relevant part that:

(1) If the trustee rejects an executory contract under which the debtor is a licensor of a right to intellectual property, the licensee under such contract may elect—(A) to treat such contract as terminated. . . (B) to retain its rights . . . to such contract, to such intellectual property (including any embodiment of such intellectual property to the extent protected by applicable non-bankruptcy law).<sup>67</sup>

For the purposes section 365(n), the subcategories of intellectual property are: trade secrets; inventions, processes, design, or plants protected under Title 35 (the U.S. Patent Act); patent application; plant variety; work of authorship protected under Title 17 (the U.S. Copyright Act); or mask work protected under Chapter 9 of Title 17 to the extent protected by applicable non-bankruptcy law.<sup>68</sup> For the purposes of section 365(n), a trademark is not delineated as a category of intellectual property for which a licensee can have recourse.<sup>69</sup> Since on-premise software is protected intellectual property under copyright law,<sup>70</sup> it is undeniable that it will be accorded protection under section 365(n).<sup>71</sup> The same cannot be

---

65. 756 F.2d 1043 (4th Cir. 1985).

66. The subcategories of intellectual property that are protected under section 365(n) are: trade secrets; inventions, processes, design, or plant protected under title 35 (the US Patent Act); patent application; plant variety; work of authorship protected under title 17 (the US Copyright Act) or mask work protected under chapter 9 of Title 17 to the extent protected by applicable non-bankruptcy law. Clearly excluded as an "intellectual property" under section 365(n) is trademarks.

67. 11 U.S.C.A. § 365(n) (West 2012).

68. 11 U.S.C.A. § 101(35A) (West 2012).

69. In a note in the legislative history, Congress said, "[T]he bill does not address the rejection of executory trademark, trade name or service mark licenses by debtors-licensors . . . such contracts raise issues beyond the scope of this legislation. In particular, trademark, trade name and service mark licensing relationships depend to a large extent on control of the quality of the products or services sold by the licensee. Since these matters could not be addressed without more extensive study, it was determined to postpone congressional action in this area to allow the development of equitable treatment of this situation by bankruptcy courts." See S. REP. NO. 100-505, at 6 (1988).

70. 17 U.S.C.A § 102(a) (West 2012).

71. See 17 U.S.C.A §§ 101, 102, 117 (West 2012). See also *Apple Computer, Inc. v. Formula Int'l, Inc.*, 725 F.2d 521, 523 (9th Cir. 1984); *Moy, supra* note 9, at 163; *Schlumberger Res. Mgmt. Data Serv. v. Cellnet Data Sys., Inc.*, 327 F.3d 242 (3d

unequivocally said of a SaaS licensee. The confluence of remote delivery, pay-as-you-go pricing structure, and orientation as a service as opposed to a product,<sup>72</sup> arguably, confer inferior proprietary status to SaaS licenses from a licensee's perspective. Unsurprisingly, it has been argued that SaaS should be considered a service and thus outside the purview of section 365(n).<sup>73</sup> If the licensing agreement grants access to the software by means of a service contract, the customer will not necessarily receive a license to the underlying intellectual property, and will lack the right to retain its use of the software.<sup>74</sup> It has also been argued that because SaaS licenses are mainly structured as service contracts, and "the license to access and use the application is plainly secondary, a court very well may consider the license to be *de minimus* to the overall agreement and characterize the . . . agreement as something other than an intellectual property license."<sup>75</sup> It therefore stands to reason that "a licensor in bankruptcy or its trustee could reject the . . . agreement and the licensee would have no protection under §365(n)."<sup>76</sup> A licensee is however, not without recourse. As will be suggested below, there are creative ways of circumventing such limitations, and a licensee can take other protective steps, such as structuring SaaS licensing and escrow arrangements to provide continued rights to use the software.

#### IV. NEW WINESKINS

##### A. SaaS Escrow Arrangements

If the parties agree to place the SaaS source code into escrow with a third party vendor, the licensee should consider the following steps to optimize its utilization if the need should arise.

##### 1. Appropriate Escrow Release Triggers for SaaS Licenses

The typical source code release triggers such as bankruptcy, ceasing to do business, failing to support the software that are common in on-premise software escrow arrangements, and which enable the licensee to recreate the software application within its own computer system, may not sufficiently address the crucial risk factors that are inextricably inherent in

---

Cir. 2003).

72. See THINKstrategies, Inc., *supra* note 39, at 5.

73. See CLASSEN *supra* note 10, at 143.

74. *Id.*

75. Elaine M. Laflamme & Noel D. Humphreys, *Protecting Against Bankruptcy of an ASP*, N.Y. L. J. (Feb. 26, 2002) (emphasis added). See also Richard M. Cieri & Michelle M. Morgan, *Licensing Intellectual Property and Technology from the Financially-Troubled or Startup Company: Pre-bankruptcy Strategies to Minimize the Risk in a Licensee's Intellectual Property and Technology Investment*, 55 BUS. LAW. 1649, 1685 (1999–2000).

76. *Id.* (footnote omitted).

the utilization of a SaaS infrastructure.<sup>77</sup> Such significant risk factors include, but are not limited to: (1) unexpected service interruptions; (2) downtime due to power outages; or (3) loss of functionality of a remotely hosted application.<sup>78</sup> Aside from the loss of access, there is the concomitant loss of proprietary data hosted offsite on licensor's servers or storage devices.<sup>79</sup> It is therefore crucial that in addition to specifying typical release events suited to on-premise software licenses such as bankruptcy, ceasing to do business, failing to maintain software, etc., the SaaS license or source code agreement additionally should include these SaaS operational downtime related release events that will enable the licensee to access the source code and stored proprietary data.<sup>80</sup>

## 2. Addressing Licensing and Subscription Issues

The licensee may also insist on contractual provisions that separate its subscription of SaaS services from the licensing of the underlying software. Such separation, whilst assuring the licensee the operational and economic advantages of a "pay as you go" model, will also ensure that the licensee's rights to the software as an intellectual property inure to its benefit under section 365(n), if the licensor should file for bankruptcy.<sup>81</sup>

## 3. Dealing with Licensor's Bankruptcy

The SaaS licensing agreement and attendant escrow agreement must affirmatively characterize SaaS software as "intellectual property" subject to section 365(n) to enable the licensee to definitively exercise its rights under section 365(n). Such contractual affirmation may circumvent or ameliorate any industry or customary precept that seeks to characterize SaaS license as a mere accessory to the provision of a service and therefore not an intellectual property right within the purview or protection of section 365(n).<sup>82</sup>

Also, since the licensee may be obligated to make royalty payments under the contract if the licensor should file for bankruptcy, it may be useful to include contractual terms which separate the SaaS licensing fees from payments for the licensors affirmative obligations under the agreement such as maintenance, support, consultation, further development, indemnification, etc.<sup>83</sup> In other words, "a licensee would be well advised to have the license agreement clearly delineate any royalties or license fees that are payable with respect to the licensed intellectual

---

77. See van de Zande & Jansen, *supra* note 49.

78. *Id.*

79. *Id.*

80. See generally THINKSTRATEGIES, INC., *supra* note 39.

81. See Laflamme & Humphreys, *supra* note 75.

82. See CLASSEN *supra* note 10, at 146.

83. See Westermeier, *supra* note 20, at 5.

property from other fees that the licensee may be required to pay for maintenance, training or other services. Without this delineation, the licensee may be required to pay amounts for services that it will no longer be entitled to receive from the licensor . . .”<sup>84</sup> because it cannot be compelled to provide such services if it rejects under section 365(n). On the other hand, if the licensor accepts the license agreement in bankruptcy, the

[l]icense agreements need to contemplate specifically the post-bankruptcy rights that the licensee may retain. The continuing obligations of the licensee that survive the licensor’s bankruptcy rights need to be detailed. For example, it may be appropriate to adjust the payment terms to reflect the responsibilities of the parties in the event that the licensee elects to retain the licensed software and related source code. Typically, license fees, renewal fees, maintenance fees, and so forth implicitly include royalties for the use of the licensed intellectual property, but the royalties are not expressly denominated as such. Licensors need to ensure a continuing revenue stream from the retaining licensee. It is recommended that license agreements reflect the royalty payments that the licensee has to pay the licensor in such a situation in which the licensee is granted the right to retain the licensed software albeit without any licensor support. This is because, if the licensor is not providing maintenance, then there would probably be no continuing obligation to pay the licensor for the retained use of the software and related source code if the licensee’s only existing obligation, for example, was to pay maintenance fees.<sup>85</sup>

---

84. See Michael R. Egger, *A Practitioner’s Guide to Section 365(n) of the U.S. Bankruptcy Code*, FENWICK WEST LLP (Apr. 29, 2003), [http://www.martindale.com/intellectual-property-law/article\\_fenwick-west-llp\\_2520.htm](http://www.martindale.com/intellectual-property-law/article_fenwick-west-llp_2520.htm).

85. See Westermeier, *supra* note 20, at 3. An example of licensing terms that may achieve that objective:

If the Licensor rejects the License Agreement under Section 365(n) of the Bankruptcy Code, the Licensee may elect to (i) treat the Agreement as terminated pursuant to Section XXX (Termination) of this License Agreement or (ii) retain Licensee’s rights under the License Agreement, including, without limitation, the right and license to use, adapt and modify the Licensed Software and related Source Code for the full term of the License Agreement and obtain a complete and current copy of the source code corresponding to the licensed software used by Licensee from the Source Code Escrow Agent or, in the event a complete and current copy of the Source Code is not provided to the Source Code Escrow Agent, then directly from Licensor.

In consideration of obtaining a copy of the Source Code under the provisions of this Agreement, Licensee agrees to pay Licensor, in lieu of any other fees, an annual royalty in the amount of \$\_\_\_\_\_ commencing upon Licensor’s receipt of the Source Code and continually thereafter on the

In light of the foreclosure of licensor's obligation to affirmatively provide services related to the software upon licensor's rejection of the licensing agreement in a Chapter 11 bankruptcy proceeding, and because the licensees retain only rights that exist at the time of licensor's bankruptcy, the licensee can optimize the residual utility of the software it inherits from the bankrupt licensor by requiring the licensor to maintain, support and update both the software in escrow or in possession of the licensor throughout the term of the agreement. In this regard, the licensee should ensure that the license agreement expressly includes ongoing source code and maintenance rights and obligations.<sup>86</sup>

It is also instructive to note that the statutory protections accorded to the licensee by section 365(n) are meaningless if upon rejection of the license agreement by the licensor's trustee, the licensee does not take affirmative steps to exercise its retention rights under section 365(n). Failing to take such affirmative steps may result in the relegation of its claims to a mere pre-petition status as an unsecured creditor.<sup>87</sup> The licensee may however avoid this, if it is affirmatively provided in the licensing agreement that its failure to assert its rights of retention should not be construed by the courts as termination of contract pursuant to section 365(n)(1)(A).<sup>88</sup>

## B. Alternatives to Escrow Arrangements

In lieu of or in addition to escrow arrangements, the parties may consider service level agreements, incorporate disaster recovery plans, or SaaS portability or failover provisions to assure continue access to the SaaS application in the SaaS license agreement.

### 1. Service Level Agreements and Failover Guarantees

Service Level Agreements ("SLAs") set measurable network availability or uptime commitments of a SaaS system.<sup>89</sup> The SLA sets contractual

---

anniversary of such receipt for as long as Licensee continues to use the Licensed Software, or any derivative thereof. *Id.*

86. See Ron Meisler et al., *Rejection of Intellectual Property License Agreements Under Section 365(n) of the Bankruptcy Code: Still Hazy After All These Years*, 19 NORTON J. OF BANKR. L. & PRAC. 163, 170–71 (2010).

87. See *In re EI Int'l*, 123 B.R. 64, 68 (Bankr. D. Idaho 1991) (where the licensor contracted to supply customized software to the licensee, but after the licensor filed for bankruptcy, the licensee elected to retain its rights after the licensor rejected the contract under section 365 of the bankruptcy code. The license claim for \$3,631,533 in contract damages was rejected by the court, which ruled that if the licensee was entitled to damages; it would be what it was entitled to as a unsecured prepetition claim for breach of contract by the licensor.).

88. See Alan S. Wernick, *The Software Bankruptcy Trap*, COMPUTER LAW STRATEGIST 3 (April 1991), <http://www.innovasafe.com/pdf/wernick%20bankruptcy.pdf>. See also *In re EI Int'l*, 123 B.R. at 66.

89. See Pankesh Patel et al., *Service Level Agreement in Cloud Computing*, [http://knoesis.wright.edu/library/download/OOPSLA\\_cloud\\_wsla\\_v3.pdf](http://knoesis.wright.edu/library/download/OOPSLA_cloud_wsla_v3.pdf).

expectations between the SaaS service provider and a licensee and specifies in measurable terms, what services and guarantees the cloud provider will provide.<sup>90</sup> In SLAs, particularly, the uptime commitment refers to the obligation of the licensor to use commercially reasonable efforts to ensure that its hosted system or application is available to the licensee a certain percentage of the time. It is not uncommon to find service level or uptime commitment (excluding scheduled maintenance) of 99.9% of network availability in licensing agreements.<sup>91</sup> To be meaningful to the licensee, an SLA should require the licensor to use commercially reasonable efforts to correct any material problems in the services, including any failure to satisfy the uptime commitment.<sup>92</sup> If the licensor fails to satisfy the uptime commitment for a given month, the licensee should be entitled specific service credit equal to pre-set percentages of the monthly fees for the services for stated uptimes. The SLA should also require the licensor to proactively manage and monitor the application server hardware devices and software to ensure optimal performance and reliability as well as to detect abnormal events or exceeded utilization or performance thresholds. The licensor should also be required to operate, monitor, and administer all servers, applications, and networks supporting the services. Maintenance for scheduled outages, if necessary, should be conducted at a time and in a manner that minimizes adverse impacts on SaaS access.

## 2. Disaster Recovery Plans

Another viable alternative or complementary arrangement to escrowing SaaS applications is a disaster recovery plan. Disaster Recovery Plans (“DRP”):

[Apply] to major, usually physical disruptions to service that deny access to the primary facility infrastructure for an extended period. A DRP is an information system-focused plan designed to restore operability of the target system, application, or computer facility infrastructure at an alternate site after an emergency. The DRP may be supported by multiple information system contingency plans to address recovery of impacted individual systems once the alternate facility has been established. A DRP may support a [Business Continuity Plan (“BCP”)] or [Continuity of Operations Plan (“COOP”)] by

---

90. Lee Badger et al., *US Government Cloud Computing Technology Roadmap Volume I Release 1.0 (Draft)*, Special Publication 500-293 (Draft), NAT’L INSTIT. OF STANDS. AND TECH., 17 (Nov. 2011).

91. *99.9% Uptime SLA*, BLACKBOARD, <http://www.blackboard.com/Services/Managed-Hosting/99-9-Uptime-SLA.aspx> (last visited Feb. 11, 2013).

92. See Wayne Jansen & Timothy Grance, *Guidelines on Security and Privacy in Public Cloud Computing*, Special Publication 800-144, NAT’L INSTIT. OF STANDS. AND TECH., 7–8 (Dec. 2011), <http://csrc.nist.gov/publications/nistpubs/800-144/SP800-144.pdf>.

recovering supporting systems for mission/business processes or mission essential functions at an alternate location. The DRP only addresses information system disruptions that require relocation.<sup>93</sup>

Relocation should be from the supplier's primary facility to a standby facility, which is geographically remote from that of its primary facility. The supplier should be required to test at least annually the resilience of the back-up and data return arrangements put in place under this plan, and the customer should request to see the results of those tests.<sup>94</sup>

### 3. SaaS Application Interoperability and Portability

SaaS portability refers to the ability to move SaaS applications between vendors with minimal integration issues.

Data portability is the ability of cloud consumers to copy data objects into or out of a cloud or to use a disk for bulk data transfer. Service interoperability is the ability of cloud consumers to use their data and services across multiple cloud providers with a unified management interface. System portability allows the migration of a fully-stopped virtual machine instance or a machine image from one provider to another provider, or migrate applications and services and their contents from one service provider to another.<sup>95</sup>

With the consequences of losing access to a business critical application magnified if it occurs in conjunction with the loss of remotely hosted data, it is imperative that in addition to or in lieu of requiring the escrow of licensor's proprietary software, the licensee may require the licensor to provide the ability to either extract and migrate data and application to either the licensee on a on premise software solution or to a new vendor or the licensor's backup infrastructure to enable the licensee to resume vital operations.

## V. CONCLUSION

The technological transformation of software licensing from on-premise to externally hosted models has taken a big leap ahead of settled legal principles that govern licensees' access to escrowed source code should access disruptions occur.<sup>96</sup> The natural considerations that underlie the

---

93. Marianne Swanson et al., *Contingency Planning Guide for Federal Information Systems*, Special Publication 800-34 Rev. 1, NAT'L INSTIT. OF STANDS. AND TECH., 11 (May 2010), [http://csrc.nist.gov/publications/nistpubs/800-34-rev1/sp800-34-rev1\\_errata-Nov11-2010.pdf](http://csrc.nist.gov/publications/nistpubs/800-34-rev1/sp800-34-rev1_errata-Nov11-2010.pdf).

94. *Id.*

95. Fang Liu et al., *NIST Cloud Computing Reference Architecture*, Special Publication 500-292, NAT'L INSTIT. OF STANDS. AND TECH., 15 (Sept. 2011).

96. See CLASSEN, *supra* note 10, at 146.

structuring of an on-premise software source code agreement that enables a licensor to access source code to recreate the functionality of software on licensee computer systems are inadequate to address the disruptive risks attendant to the licensing of SaaS licenses.<sup>97</sup> Though source code release events such as bankruptcy, failure to maintain, ceasing to do business, etc., are still relevant to licensing, its remote delivery and offsite storage of licensee data requires consideration of other release factors in escrow arrangements. Colleges and universities must consider other critical release factors such as prolonged power outages, system failure, or downtimes as additional triggers for the release of the source code and related documentation to enable them to recreate the functionality of an externally hosted system. The SaaS licensing agreement and escrow agreement should also consider the escrow of licensee data that will be stored on the licensor's externally situated servers and devices to enable the licensee to access these expeditiously in conjunction with the source code.<sup>98</sup> It is also clear that because SaaS software may not be deemed to be intellectual property, it is prudent to provide contractual provisions and language that affirmatively characterize SaaS as intellectual property in order to enable the licensee to unequivocally rely on the useful, but somewhat limited protections accorded to a licensee if the licensor in bankruptcy should terminate the licensing agreement. Colleges and universities that license SaaS should proactively consider the practical steps outlined in this article to fully exploit the benefits of their SaaS acquisitions. "Neither do men put new wine into old bottles: else the bottles break, and the wine runneth out, and the bottles perish: but they put new wine into new bottles, and both are preserved."<sup>99</sup>

---

97. See van de Zande & Jansen, *supra* note 49.

98. *Id.*

99. *Matthew* 9:17 (King James).