Managing Software Intellectual Assets In Cloud Computing

Part 1

By Dwight Olson and Stephan Peters

Stephan Peters, LES Germany, and Dwight Olson, LES (USA & Canada), discuss how technology escrow, a common technique for avoiding quarrels between licensor's and licensee's of software, is morphing and getting global attention to help make win-win situations in the cloud! This is the first article in a two part series on the emerging issues of having software and data in the cloud. These two articles begin to address the inherent complexity of managing the risk of software assets including data in the cloud. Part one will look at software programs and systems, then part two will focus on data. This article assumes the reader is generally familiar with classical software escrow and cloud computing concepts including SAAS, PAAS and IAAS.

Nowadays, software is highly sophisticated and may be protected by many forms of Intellectual Property (IP) and contain many forms of Intellectual Assets (IA). It is almost ubiquitous as it impacts and improves countless aspects of our professional, corporate, and personal lives. Software creators typically devote significant time and financial resources in an applications pre-commercial phase and those investments often continue long after the software's market launch and through its life cycle. It is thus a no-brainer that owners of the software strive to protect the valuable components in that software. Just as for any commercialization, the rule, 'Put protection first!' applies.

As discussed in a previous *les Nouvelles* article regarding software, one could consider "intellectual property to comprise patents, trademarks, copyrights and industrial designs because typically these four intellectual properties have Federal legislation with a public registry to govern, protect, and permit value propositions for the owners."[2] The software content or components called IA would be the trade secrets, know-how, or the codified, tangible descriptions of specific knowledge which may not have specific Federal legislation to protect commercialization.[3]

Independent of the kind of IP and IA protections that apply, the classical software model comprises one creator/licensor, an application, a license and a user who installs (the object code of) the software on his proprietary hardware. The licensor controls the origin of the software (its source code) and provides maintenance to the user, who in turn owns and controls the complete infrastructure on which the application runs. Moving ahead into the cloud, this simple setup is changing completely as additional parties are introduced and power and control shift.

■ Dwight Olson,
V3Data, Principal,
San Diego, CA, USA
E-mail: dolson@V3Data.com

■ Stephan Peters,
Deposix Software Escrow,
Founder & CEO,
Munich, Germany
E-mail: stephan.peters@deposix.com

Having established that, what about digital *data* housed in the myriad of electronic databases across the globe? Data may not be protected as an IP, but it certainly is a valuable intellectual asset. In the past, data was considered owned by the licensees—at least that's what they would contend—and historically most corporate data was controlled by the software in protected information technology (IT) departments; governance was done by IT personnel who were employees of the company and typically charged with the responsibilities of maintaining and securing that data. Most data resided behind the walls of the company's data center where security, retention and backup were of primary concern. That is, keeping data safe from migration outside the company, keeping the data only as long as required for legal reasons (or if vital digital records, forever), ensuring data consis-

1. Software as a service, platform as a service, infrastructure as a service.

2. "Intellectual Asset Identification, The First Step in an Intellectual Property Management Program," by Dave Tyrrell and Gary Floyd, Vertex Intellectual Property Strategies Inc., 2011, *www.vertexips.com/information/articles/identification.html.*

3. Software Inventory Valuation Part 3—A TSV (OV) *les Nouvelles*, June 2009.

tency (its validity, accuracy, usability and integrity) and keeping the data backed up in case of an IT data center disaster. These were the nightmares of the IT department. To complicate these four nightmares nowadays, software and data are becoming valuable assets and as such may be generating revenue directly or indirectly for the company. In the cloud we might find licensors, licensees and other stakeholders fighting for revenue without consideration of any rules of the road.

Data and software migrating to the cloud is moving responsibility for governance out of the hands of the corporation. So, with data and software in the cloud, where is it really and who is responsible for governance? More about data in Part 2.

The First Challenge of Governance in the Cloud

How exactly are we going to manage the intellectual assets represented by software applications in the cloud? A little background here would be helpful. Yes, for the licensor unwarranted copies may exist, but this is not a new issue for the owner of software, even if in the cloud. Digital rights management systems have been around for quite some time and will operate just fine in the cloud. However, for the Licensee, corporate concern is about continued use. This implies access to the licensor's design documents, trade secrets, know-how, and source code. And why are they so important? Recall that software application source code provides access to all the designs, know-how and other intelligence incorporated into a given software application product. Think of it as accessing all the information about a patent at the PTO[5], but without a patent in place to protect the licensor.

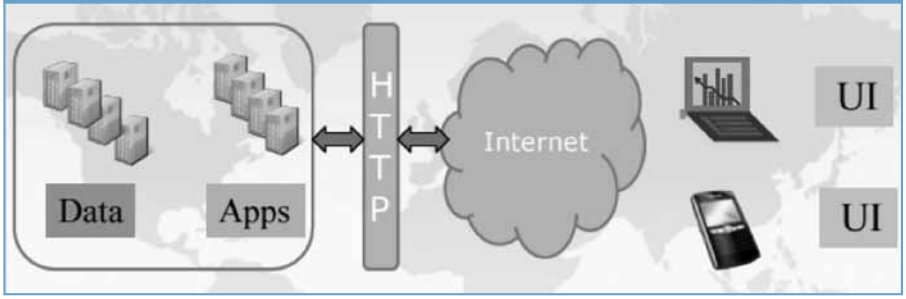
***An example**. When software biotec creators design software for a new functionality such as to add a new 3-D field in the company's genome's database and manage that field via a browser, they must write specific instructions for the computer servers and browsers and then coordinate with the database manager to modify the database schematics. Creators may use any of the manifold of existing computer based programming languages and API's that provide for 3-D modeling in HTML 5 for transport to browsers. The result is the so-called source code, and anyone being capable of 'reading' it, that is understanding the computer program modeling language, could extract the designs, know-how and expertise which the original creators put into it (thus the effort by the creator to keep it confidential). The computer-based programming language and API interfaces 'translate' the source code into computer-readable code called object code. This process is called 'compilation' and as a result creates the "executable" programs (you know them on a PC as the so-called '.exe' and '.cgi ' files). These files control the computer and interface the database system. So in order to implement the new 3-D field, the creator needs the source code to make the program changes. The creator also needs to coordinate with the database manager for access to the database schematic to make the database changes, then the creator needs the data extracted, schematics updated, and data restored together with the installation of the new software. Without the source code the software and data could only be used as it was. There would be no 3-D field.*

Ownership or rights and access to the source code provide the ability to modify and support use, continuation and commercialization of software. To complicate matters, today's software and data are becoming inextricably linked. And as seen in the figure below, software and data and their physical infrastructure (hardware) may no longer be physically together or managed by the same organizations.

It's Easy Being the Licensor… Not!

Historically some licensors have relied on copyright protection or to possibly file a patent for the software. Yet, it is not always that easy. While copyright protec-

Cloud Computing[4]



4. From *LES 2010 Annual Meeting Workshop 2-B Cloud Computing 101: Business Models, Technologies and Licensing Issues*, by Richard Greeley, Director, Licensing, Microsoft Corporation; Stacy Snowman, Partner, DLA Piper; Tyron Stading, Founder, CTO, Innography.

5. Patent and Trademark Office.

tion and patents for software are generally available across most industrialized countries and jurisdictions, typically only certain parts of today's comprehensive software programs out there stand a chance of reaching that intended shelter. Yes, copyright protection is typically universal and used to protect specific components of the software, such as the deliverable .exe programs, "how to" documentation and to aid in the enforcement of un-warranted copies. Patents for computer-implemented inventions offer a broader protection for a product or process regardless of the software language they are written in. Patents can be obtained generally for novel software-based inventions which, for example, guide a satellite in orbit, manage more telephone calls through narrower bandwidth, or make a computer run faster through more efficient memory usage. But any competitor, who finds a different way to achieve the same objective, perhaps even a better way, will not have to explain himself to the patent holder. Furthermore, and this is true for other content such as books, music or videos as well—software is digital and travels very easy on the Internet and between computers and countries in the cloud. Nowadays, there are very few physical restrictions on its dissemination or boundaries. In many ways, it is this last aspect only that enables cloud computing.

But, here is the heart of the issue for the licensor. Historically, even when considering copyright and international patent protection for one's software, most licensors will never allow their source code, know-how, and design document contents to be disclosed, either to the general public (through copyright registry disclosure) or to licensees who desire access to source code for risk management. Why? Because trade secrets and know-how, often considered the most valuable assets of a company, could possibly be extracted and re-used in any number of creative and legal methods. The result could mean dire consequences for a vast majority of existing creators and licensors.

It's Easy Being the Licensee… Not!

So while the licensor/creator has every reason to keep his trade secrets and know-how secret, that is not the position of cloud customers, cloud caretakers, and licensees (let's call them the body of users). From their perspective, risk management and protection is just as crucial for continued revenue. This may be further complicated if the software application is managing the data "owned" by one of the body of users.

Is the body of users to be totally dependent on the good will of the licensor to allow them to exploit both the data and software's immediate benefit and its long-term potential? NOT! If only it would be that easy! The issue of software continuity for the body of users is far more complex. Here is an example. Often times in the corporate world, corporate licensees have to invest millions in new and improved software and it's inevitable implementation into an existing IT landscape. Often new software, even updates, must be integrated and databases reinstalled. This requires additional costs for individual adaptations, interface programming, new hardware, modifications to surrounding IT infrastructure, time and effort to analyze and adapt obsolete or incompatible internal business processes, and then comes training for the employees. Yes, the body of users each has a strong interest in mitigating the risks involved and in protecting their investments in any software and recapturing existing data. But stuff happens! Licensors and even some cloud providers could default on maintenance or other critical deliverables and sometimes they even go bankrupt. Then what?

Thinking About the Problem

For those readers who have not had to think about access to trade secrets such as source code at the corporate level, recall the time when you decided to upgrade from an older model of a PC with Microsoft XP to a newer model PC with Microsoft Vista (now Windows 7). Remember? Some of your XP programs would not install on Vista, you needed to go to the supplier of the software and purchase an upgrade. Remember! Not fun! What would you have done if the supplier was not there for the upgrade? What would you have done with your data files? In the corporate world, the company needs to get access to the creator and/or the source code if the licensor is gone.

Depending on the licensing process, the parties who did not start addressing this conflict of interest early in the negotiations encountered a very costly situation. The quarrel over the source code has been recorded as an insurmountable stumbling block in more than one case in the past 30 years! Just recently, Marshall Phelps, past corporate counsel for Microsoft, was speaking at an LES event on doing business in China; he recalled the Chinese government wanted the source code for the Microsoft Operating System. I believe the words were: "NO WAY, they are some of the most valuable assets of the company!"

(Note: The particular case of open source (OS) will not be discussed here—while the authors recognize the value and acceptance of the many OS business models being used globally, we predict that there will always be proprietary software in the global community including the OS collaboration world.)

This conflict of interest becomes apparent when the licensor prefers to keep his source code secret and not disclose it to anyone; but the body of users seeks to get hold of the source code in case the licensor defaults on obligations. Both sides have legitimate interests. If both sides insist, they would never sign an agreement, but luckily source code escrow was invented in the early 1980's[6] to address this specific problem. Historically, without source code escrow the deal may have gone south or the creator may give up its assets.

But, the issue and solution is far more complicated in the cloud. Why? In the cloud, who is responsible for mitigating the risk for future source code access? Who, of the body of users, has any contract/license agreement at all with the creator of the licensor? Who and where are the creator(s)? Do the SAAS players license the software to the PAAS or IAAS players, or do they just have an agreement to use the platform or infrastructure? Is the SAAS player the creator or is there another creator/licensor? These were the nightmares of the corporate IT department. Who of the body of users is thinking of these issues?

In order to begin to solve this or any other cloud issue, it may be wise to select a team of players who can begin to provide some guidance. Possibly, IP counsel with international bankruptcy experience, a trusted neutral third party with international and technical escrow expertise, and the body of user's counsels would be a good start. They must construct solutions to mitigate risks for all parties' including the licensor. Sometimes a starting solution is to use the current escrow vehicle to address the software continuity dilemma and morph as needed.

Historically, after 1986, the software escrow vehicle was a three-party constellation in which the escrow company served as a trusted bailee or SPE[7] and held the IA in digital secure custody; similar in concept to a notaire, bank, or legal services firm which serves as trustee in physical properties and securities. But in the cloud, the trusted bailee must have the competency to understand the total situation and to help implement a solution complementary to the underlying agreement. The team must take into account all points of view and mitigate the risks including technical and value solutions. Someone needs to give an opinion as to whether the escrow and contents are valuable, appropriate, and fit for the purpose. In the cloud this will require the legal community, technical escrow companies and consultants to work in concert to solve the dilemma. How does one create instruments and verify, validate, and value the contents that will help solve access and continuity? And be able to address the issues of the data (part 2)?

For the technical escrow companies, these organizations need to have internal processes with verification and validation expertise to help counsel with the deposit components, and then be ready to accommodate regular updates, electronic or media, and possibly have an international transfer capability or partner to house the escrows in an appropriate country.

Remember in the United States, Congress amended the Bankruptcy Code, 365n, to permit an executory contract/license to continue after bankruptcy as well as any escrow agreement that was supplementary to the license. So what licenses exist in the cloud? What country is the licensor in? If the provider goes bankrupt, who is responsible?

Based on the business deal, the countries involved, the parties involved, and the risks to be mitigated, the technology escrow/bailee/SPV company and counsels may need to morph software escrow and adapt new solutions to the special situations of the international cloud. For example, Data Securities International operated prior to the 365n statue change for executory contracts. They worked with U.S. counsels to prepare instruments that could begin to solve the executor status issues. Using a structure like a 'letter of credit" or 2 x 2 party agreement structure' could act as a beginning. Though this structure was never tested in court, because 365n was amended, at all times where a license agreement existed with an escrow agreement, the source code deposit was delivered to the licensee.

In one situation, however, when the escrow was supplementary to an investment collateral agreement; instead of providing the escrow contents, it provided the investor with enough information to convince the judge to render the decision that the debtor could not just walk away with the source code. Because of the escrow, the investor was able to gain an equity position in the new company.

Once the contract structures are worked out, specific release conditions defined, SPV defined and

6. *http://en.wikipedia.org/wiki/Data_Securities_International.*

7. See *les Nouvelles* March 2008 Leveraging Software: "lenders will prefer to have the software assets reside in a wholly owned subsidiary that has a totally separate operation from the software owner. Referred to as a Special Purpose Vehicle or Entity ("SPE"), these holding companies serve to protect the lender against the possibility that the software owner will file for bankruptcy protection. If there is not a SPE already in place, setting one up should be one of the first steps taken."

implemented, content and ownership transfer process agreed upon, verification, validation, and valuation determined, only then can the software owner securely transmit the initial components to the first independent software escrow/bailee/SPV company for digital safe keeping. In some cases this may require a secure and encrypted transfer between independent software escrow companies of the design documents, know how, trade secrets and source code internationally to provide legal jurisdictions. This may also require the same or another independent escrow company to accommodate the data and database for additional escrow protections for the database owners. However, without someone providing an opinion as to whether the escrow is worth its salt, verification, validation and/or valuation must be undertaken.

Yes, another major expense of a source code (trade secret) escrow registry is in the verification, validation and/or valuation procedures besides the contract formulations. Having some degree of guarantee is of the utmost importance. If you are going to put it together, you want to make sure what is in the escrow registry is what is useful, informative, and of value. This was not a trivial situation for the first technical escrow companies, yet they managed to put together some sophisticated standard verification processes. These processes are still in use today by most of the technical escrow companies. Yes, verification can even be done for the cloud.

Today, There is a Lot More Risk in the Right to Use Software in the Cloud

Why validation and valuation processes for the IA? Well, currently verification deals with continuity of the software application for use. If the deal is only for software continuity in the cloud, then "for use" verification may be all that is required.

Note: "For use" verification processes typically only deal with the ability to recompile the source code into executable code that will permit the licensee to update and make changes to the existing program. However, "For collateral agreements, development agreements and the like" other processes need to be developed to determine if "all" components are in escrow that provide the other party with the ability to fulfill the intent of the agreement. It may be collateral where a bank needs to collapse on the asset if a loan was not repaid.

But many escrow situations are centric to development agreements, collateral agreements, reseller agreements, co-operative agreements, and the like. Thus data and other components of the intellectual assets may be required to be opinioned in addition to the source code. Of course, the software escrow registry holding company must be able to verify that the deposit can re-build the application, but what else? The new software escrow registry must also be able to technically validate any open source issues (Is the escrow full of open source?), database schematic issues, and database retention issues. Then possibly they may be required to provide an opinion as to the value of the software contents. Is the software ownership transferring between countries? Are there tax implications? What else might be needed for different uses of the software? These are just a few of the questions that will need to be answered.

Let's Close Part 1 With the Advantages of Software Escrow of the Past

A correctly administered software escrow registry and agreement can be beneficial to all parties, and the most obvious advantages being:

1. *The critical IP/IA of the software developer is properly protected; there is no immediate discloser of source code.*
2. *The licensor is securing a revenue stream by licensing his software to that particular customer, which otherwise might not have happened.*
3. *At the same time, the licensee's investment (into this software and into his IT infrastructure in general) is protected by an appropriate risk management tool.*
4. *Licensee's need for continuity (of IT operations) is being addressed.*

Apart from these obvious benefits, software escrow offers additional advantages that may not be so evident for the casual global observer. There are many others reasons.

First, facilitating deal closures: for the licensor, using a professional escrow service builds trust in the marketplace and thus serves as an effective sales tool. The licensor is sending out a positive market signal about his own solidity and is openly addressing the risk management needs of his customers.

Second, fostering IP creation: escrow generally fosters the creation of IP, more specifically by supporting the development of software, through offering more attractive market conditions—due to the additional security, licensees are more likely to buy the license from the developer.

Third, offering additional financing options: when looking for financing—a critical process for every software developer—software escrow offers the benefit of reducing risks to prospective investors/shareholders. A potential investor performing a due

diligence will carefully analyse the company's IP assets and assess their individual risk. If the licensor can prove that his software is held in custody with a professional escrow service, this will add significant value to his business. And furthermore, software developers can offer to add the potential investor to its escrow as beneficiary, thereby granting them access to what typically is the major 'asset' in any (new) technology start-up.

Fourth, improving stability in the EU for Basel II and Solvency II ratings: Escrow may help licensees to obtain a better rating under the Basel II or Solvency II schemes by reducing their overall operational IT risk, *e.g.* failure of their critical IT systems due to a potential default of their licensor. As a result, the licensee may obtain access to cheaper credit offerings.

Fifth, building an audit trail: From the moment of signing an escrow agreement, all events such as modifications concerning the source code are seamlessly documented and a professionally managed audit trail accumulates. The licensee will benefit from an audit trail as it allows him to roll back to older versions at any time. But also the licensor will benefit from older versions of his source code staying in the depot of an independent trustee. In case of an IP violation, the escrow company could always prove the exact date when the licensor developed his IP, a possibly crucial aspect when it comes to patents, or cases of industry espionage or disgruntled employees which involve the IP incorporated in the source code.

In closing, it is worth noting that the United States is the only country in the world that offers patent protection under its patent law to those who are the 'first to invent.' All other countries are on a 'first to file' basis. Given U.S. patent legislation, an audit trail provides a valuable additional backup to the licensor when planning to file for a U.S. patent. In this way, the licensor will be able to prove 'first to invent.'

Software escrow, though not as widely known outside of the United States, Germany and Great Britain, has escrow companies (modeled after Data Securities International, Inc.[8] the first software escrow company that started in 1982) that can begin to provide solid solutions for controlled access to the verified, validated and possibly valued source code! ■

8. See *http://en.wikipedia.org/wiki/Data_Securities_International?oldid=0.*