Continuity in the Cloud: new practical solutions required

An inventory from a Dutch perspective

September 2013

Ernst-Jan Louwers, attorney-at-law
at Louwers IP|Technology Advocaten

Copyright © 2013
All Rights Reserved
Table of content

BIOGRAPHICAL SUMMARY....................................................... 3
INTRODUCTION ........................................................................... 4
ESCROW AGREEMENT AND BANKRUPTCY .............................. 6
TRADITIONAL AND CLOUD ESCROW ................................. 8
CONTINUATION HOSTING SERVICES AGREEMENT ....... 9
SPLIT COPYRIGHT ................................................................. 10
USUFRUCT .................................................................................. 12
DATA ............................................................................................ 13
FOUNDATION AND MIRROR ENVIRONMENT ............... 13
CONCLUSION ............................................................................... 15
ANNEX: METHOD AND LEVEL OF RELEVANCE OF
THE APPLICATION OR SERVICE .............................................. 17
BIOGRAPHICAL SUMMARY

Mr. Ernst-Jan Louwers, LLM (1959), attorney-at-law ('advocaat')
Firm: Louwers IP|Technology Advocaten
Address: Ten Hagestraat 5, P.O. Box 440, 5600 AK
Eindhoven, the Netherlands
Phone: +31 40 2393200; fax: +31 40 2393201
E-Mail: louwers@louwersadvocaten.nl
Internet: www.louwersadvocaten.nl

Primary Areas of Practice: IP/advertising, IT/internet, privacy

Education: Leiden University and specialised in Intellectual Property Law at University of Amsterdam

Recognition: Chambers guide: "He's someone you can trust. He is thorough, fast and practical. He is extremely accessible and does not stick his nose in the air." He is based in Eindhoven and concentrates on IT, IP, competition law and matters relating to the Internet and ecommerce law."

Membership in Associations, Committees, etc.: Complaints and Appeals Board for .nl domain names (C&AB), arbitrator at WIPO, board member to Brainport Centre for Technology & Law, member of the Cyberspace Team of Marques, European Community Trademark Association ECTA, International Trademark Association INTA, Dutch Information Technology Law Association (NVvIR), iTechLaw.org and Dutch Copyright Association and IT specialists association Dutch Association for Information Technology Lawyers (www.vira.nl).
INTRODUCTION

This paper is written from a Dutch legal perspective and is intended for discussion purposes. It was written for the ItechLaw conference in Amsterdam in October 2013.

Software normally presents itself in two capacities, the machine language (object code) and the human-perceivable source code. Traditionally the software is installed locally on the user’s system. In order to ensure the possibility to maintain and adapt the software, for instance bug fixing, the sources are deposited at an independent escrow agent who would be entitled to release the sources to a legitimate user upon certain triggering events, such as bankruptcy of the supplier. If the bankruptcy is already there, any actions would be too late.

The information to be deposited with an escrow agent may include but the source code of the software and the necessary documentation. This is information that is necessary for anyone other than the programmer of the software itself to

---

1 An elaborate discussion of the backgrounds and ins and outs of escrow agreements can be found in chapter 9 of Part II of Overeenkomsten inzake informatietechnologie, P.C. Blok and others, SDU 2010; Part II was written by Ernst-Jan Louwers, Marianne Korpershoek and Tom de Wit.

2 Pres. Rb ’s-Hertogenbosch 26 January 1993, CR 1993/6, p.256 and Hof ’s-Hertogenbosch 7 February 1994, CR 1994/2, p. 50 (Breedveld & Schröder/Mr Keizers q.q.): in the absence of an escrow agreement the claim of a customer on the bankruptcy receiver to release the source code – without compensation for the supplier – was refused. The court implicitly indicated that this could have been different if the software had been developed in commission and for the account of the customer and could not be deployed elsewhere; this was not the case in this procedure.
understand that software. The holder of the rights to the software or the supplier will not be prepared to release such information to the user because the source code gives the possibility to modify or (further) develop the software.

On the other hand, the user has a legitimate interest to have a copy of the sources for continuity purposes in certain events such as bankruptcy of the supplier and poor performance under the maintenance agreement. For such events, a source code escrow is arranged to ensure the continuity of the software. Without specific agreement and transfer of ownership to the user or an escrow agent of (at least) a carrier with the source code and documentation, the user can be faced with the situation that he cannot properly maintain, repair the software or ensure the uninterrupted availability of the software.

The necessity of escrow does not exist in all cases. In case of open source software the sources are available anyhow under the applicable open source license.\(^3\)

The escrow agreement must be regarded as a compromise between the interests of the supplier not to give away its trade secrets and IP rights in the source code and, nonetheless, provide the user certain continuity warranties in certain disaster events, such as the supplier’s bankruptcy. The deposit of the source code with an escrow agent protects the supplier against a too rapid disclosure of the source code to the user. And the user has the certainty of access to the source code under certain conditions.

Nowadays, however, software applications no longer only run locally. Models like Software as a Service (‘SaaS’), Platform as a Service (‘PaaS’) or Infrastructure as a Service (‘IaaS’) are widely used whereby the software and the data no longer reside on the user’s own local system but on the SaaS service provider’s computer system and own or third party data warehouse. The software solutions and data are accessible over the internet.

**ESCROW AGREEMENT AND BANKRUPTCY**

The changes in technology call for new solutions to warrant continuity of the software and protect the user from losing its data in certain events, especially in case of bankruptcy of the supplier. Other events, such as the ill performance by the supplier under the maintenance agreement or ceasing support and further development of the software, can be arranged in agreements without much headache. Properly arranging the release of the sources and documentation upon bankruptcy may be more complicated from a legal point of view. Therefore, this paper focuses on that situation.

Especially in case of bankruptcy a problem may arise with regard to the release of the source code and its use by the lawful user of the software. For example, a bankruptcy receiver must operate in the interests of the joint creditors and may oppose the release of the sources to and use by the customer (without proper compensation) even in case an escrow agreement is in place.

It is expressly noted, however, that under current Dutch law the copyrights may not always be affected by the bankruptcy of the
owner of the copyrights if he is the actual or fictitious maker of the copyright protected material, i.e. in this case the software.\(^4\) This exception may be expected to be abolished shortly in view of the contemplated amendment of the DCA. For this reason and other reasons we recommend not to rely on this exception.\(^5\)

Since the so-called Nebula judgment of the Dutch Supreme Court, it is even questionable whether an escrow agreement is sufficient to ensure that the receiver in the bankruptcy of the supplier cannot oppose the release to and the use of the sources by the customer.\(^6\) Even though this judgement did not relate to software licenses and escrow, the judgement may nonetheless affect the position of the user in case of bankruptcy of the supplier.

The Supreme Court held that the fact that the (continued) existence of a reciprocal agreement as such is not affected by the bankruptcy of one of the contracting parties, does not mean that the creditor of such an agreement can exercise its rights as if no bankruptcy was available. The Supreme Court adds that this also applies to cases where the bankrupt is not required to perform a specific performance under the agreement concluded

\(^4\) Section 2 subsection 3 Dutch Copyright Act (Auteurswet), further DCA

\(^5\) For further reading reference is made to http://www.dickvanengelen.nl/files/Artikelen/2013_Van-Engelen_Verouderde_auteursrechtelijk_privileges_NJB_2013.pdf

between the parties but simply needs to tolerate the use of property belonging to the bankrupt party.

In Dutch legal literature it is defended that the scope of the judgment extends not only to tangible goods, but also to intangibles such as IP rights and licenses. If we apply the judgement to software given in escrow, this could imply that the licensed user could be forbidden to use the source code even upon release by the escrow agent. There is no specific case law yet in which an escrow agreement is put to the test under the judgement mentioned.

The user’s position also is tricky, if the supplier as licensor and the user as licensee have agreed that in the event of bankruptcy, the licensee may obtain the source code directly from the licensor without the intervention of an escrow agent. In case the supplier goes bankrupt, the receiver may arguably decide to deny his cooperation and refuse to release the sources if that is in the best interest of the estate.

**TRADITIONAL AND CLOUD ESCROW**

As already noted in the introduction, traditional escrow arrangements are often not sufficient anymore. In particular for the following reasons:

a) under Dutch law said Nebula judgment (the right of a receiver to refuse his cooperation and block the release of the sources) in certain cases;

b) technological developments, which also imply that only the source code does not provide sufficient guarantees any longer.
In connection with SaaS and Cloud solutions, proper arrangements to ensure the continuity of the SaaS software are highly recommendable but source code escrow is not always enough. If only the source code of the software is secured, the timely continued availability of the SaaS software and data is not sufficiently guaranteed.

Because both the software and the data of the user in a Cloud service are not under the user’s custody (as opposed to traditional locally installed software), access to the source code version of the software only is not sufficient to warrant continuity. For the user it is important to ensure more or less uninterrupted access to its own data and the Cloud services. The sources and technical documentation with respect to the software and the Cloud services would allow specialists to ensure the SaaS application to work, fix bugs, and perform updates.

Obviously, the level of measures ensuring continuity of software applications, services and access to data also depends on the kind of service, software and data involved. The table attached to this paper illustrates the levels.

**CONTINUATION HOSTING SERVICES AGREEMENT**

A measure that is also offered to ensure continuation after bankruptcy of the supplier of SaaS services, is making arrangements with the relevant hosting service provider. These arrangements are designed to ensure that the software continues to run. Users then agree with the hosting provider to take over the payment obligations in the event of an emergency, including bankruptcy.
In the case of bankruptcy it is very questionable whether such an agreement with the hosting provider would be sufficient. It would only solve the actual access to the application and data as such but not the access to the source code. Moreover, the receiver might be able to prohibit the further use of the software by withdrawing the license agreement if this would be in the best interest of the joint creditors.

**SPLIT COPYRIGHT**

An option regularly suggested as an alternative to a license is the partial transfer of the copyrights to software.\(^7\) The purpose of this partial transfer is to ensure continuity for the user. The user will own a part of the entire copyright over which the trustee in the bankruptcy of the supplier will have no control.

The splitting of copyright raises at least the following questions, especially in view of cloud computing where many users use the same software:

- Can the split copyright be transferred to an unlimited number of persons?
- How can the same piece be transferred repeatedly?
- Does this create a joint copyright to the part or the entire copyright?
- What rights must be considered not transferred after the transfer (and might fall into the hands of the trustee after bankruptcy)?

\(^7\) Article 2 DCA, expressly allows partial transfer. See for instance http://www.cordemeyerslager.nl/escrowproblemen-opgelost-met-afgesplitst-auteursrecht.
• Does the transfer of this split copyright also cover access to the source code?

These questions are further discussed below.

Pursuant to article 2 DCA both complete and partial transfer of copyright is possible (for example, the copyright confined to national borders, types of publication etc.). The supplier of the software - and owner of the copyright - can accordingly transfer all or part of the rights to the software to third parties.

This might solve the 'problem' for the licensee of locally installed software in the event of bankruptcy of the supplier since the licensee would then also be an owner of his part as installed on his system and not only licensee. 8

Which part is transferred?

An important question is what part of the copyright needs to be transferred. This should then in any case be the right to use and adapt the software for maintenance.

In our view, however, the problem here is that in case of SaaS software it is not possible to transfer a piece of copyright, namely the right to adapt the software, to multiple users, since each user uses the very same copy of the software which is available on the server of the SaaS supplier.

8 This view seems to be strengthened by the the ECJ in its UsedSoft judgement; http://www.ippt.eu/files/2012/IPPT20120703_ECJ_UsedSoft_v_Ora cle.pdf; see also http://www.dickvanengelen.nl/home/artikelen/usedsoft-v-oracle-the-ecj-quietly-reveals-a-new-european-property-right-in-bits-bytes
It is doubtful whether such ‘transfer’ would not simply be interpreted as a license rather than a transfer of ownership title.

Furthermore, another and maybe the most important disadvantage is that most suppliers will be very reluctant to say the least to transfer part of the copyrights to the users.

Joint copyright?

In case of cloud computing where many users use the very same software on the same system, splitting the copyright could create multiple copyright owners of the same part of the copyright and thus create a joint copyright.

A solution could be found in transferring part of the copyright (the right to make available and adapt the software) to a separate foundation or user association. This way these rights are in one hand to the benefit of all users. But even then, we doubt whether that will be a solution acceptable to the supplier-owner of the IP rights.

USUFRUCT

Instead of a license the user can be granted a right of usufruct. The advantage of usufruct is that it is a limited right which can be held against third parties acquiring the basic IP right. The downside is that this solution might be judged as improper use of the right and that the usufruct is invalidated. If the licensee has a valid right of usufruct he can invoke the usufruct against the IP owner and other parties, including the receiver and successors of the original owner.
Also for other reasons the right of usufruct is not always attractive for the owner of the IP rights. One of the reasons is that transfer of the right of usufruct can most probably not be excluded under Dutch law. That way the owner could be confronted with a successor of the initial holder of the right of usufruct without his consent.

DATA

In many cases the cloud provider / SaaS supplier uses external data centers to host its own services to its customers. Upon bankruptcy of the supplier the immediate reaction of the data center would normally be to terminate its services without delay.

A simple solution to prevent this and buy time, is to pay the data center services a couple of months in advance. No doubt, the supplier will in that case ask the customer to make this advance payment as well. This can be seen as a kind of insurance for continued access to the data and SaaS solution.

Also in this case the receiver in the bankruptcy of the supplier could prohibit the further use of the software on the basis of the license agreement.

IP ALLOCATION AND FOUNDATION

To prevent the intellectual property rights to software from falling in the hands of a receiver, the IP rights can be transferred in its entirety to a legal entity other than the

---

supplier that bears the operational risks. The supplier should ensure that the intellectual property rights are allocated to a separate entity (e.g. the holding company or even better a separate IP company). To ensure the validity of the transfer such transfer should always take place against a realistic price, both for tax reasons and for reasons of civil laws and bankruptcy laws. This separate entity should then - as far as possible – remain free from risks which could lead to bankruptcy.

In case of bankruptcy of the supplier (the operating company), the IP rights would not fall in the bankruptcy estate. On its turn, the IP company or holding company grants a license to the supplier enabling the supplier to market, (sub)license and modify the software.

The IP company or holding company should at all times remain free from liabilities and should therefore not engage in license agreements, warranties or other commitments towards the supplier’s customers.

**Special purpose foundation**

Although the rights to the software are thus sheltered in the separate company, this does not take away the necessity of further warranties to ensure the continued and uninterrupted availability of the software.

---

10 Please note that under Dutch law in some cases the copyrights may not be affected by the bankruptcy of the maker of the copyrighted material such as software. We recommend, however, not to rely on this exception. Upon transfer the copyrights will fall in the bankruptcy estate of the entity that has acquired the copyrights.
A sound solution for safeguarding the continuity of cloud solutions could be to set up a special purpose foundation that preferably includes a server with the SaaS application and user data as well as the source code and technical documentation. Preferably, the foundation operates a permanent real-time or almost real-time mirror environment enabling the user to switch to this fallback service whenever necessary. This solution may at the same time also serve as backup facility of the user data.

The IP owner, the user and the foundation then agree that in the event of an emergency situation, the foundation will take over the cloud services, including running and maintaining the SaaS software, as long as necessary. As soon as the holding company, also being the IP owner, has set up a new operating company replacing the bankrupt supplier or has sold of the IP rights to a third party, the new supplier can take over the cloud services again.

**CONCLUSION**

There are many ways to safeguard the position of the legitimate user of computer software. Cloud solutions, SaaS services and legal developments force us to re-invent existing escrow and systems to warrant continuity. In our increasingly borderless online world, lawyers are challenged to come up with new solutions to replace existing escrow and other continuity solutions.

The combination of transfer of the IP rights to a separate IP company or holding company not bearing risks, and the establishment of a separate special purpose foundation with
escrow-warranty commitments with the users as beneficiaries, has been applied more often by us. In some cases even together with a full, (almost) real-time backup and fallback system available operated by the foundation.

Many of the solutions discussed in this paper will never be accepted by the Microsofts and Oracles of this world. But it may help smaller software suppliers to ensure continuity and thus enhance trust in their services.
## ANNEX: METHOD AND LEVEL OF RELEVANCE OF THE APPLICATION OR SERVICE

<table>
<thead>
<tr>
<th>Level</th>
<th>Tolerance</th>
<th>Methods</th>
<th>Legal</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-critical application/service</td>
<td>Some interruption acceptable</td>
<td>• Deposit of sources at escrow agent may be sufficient</td>
<td>• Escrow agreement</td>
<td>• Application not available</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• SLA provisions with respect to reinstallation and going live</td>
<td>• Reinstallation and going live can take some time</td>
</tr>
<tr>
<td>Critical application/service</td>
<td>Must be available again immediately</td>
<td>• Deposit at escrow agent/special purpose foundation</td>
<td>• Advanced escrow-warranty agreement through special purpose entity</td>
<td>• Expensive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Hosting provider</td>
<td>(foundation; ‘stichting’)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Consultancy</td>
<td>• Foundation to take over services temporarily</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Fully redundant system / realtime mirror operated by special purpose foundation</td>
<td>• Transfer IP rights to the software to holding or other separate entity</td>
<td></td>
</tr>
<tr>
<td>Non-critical or critical</td>
<td>Plan B</td>
<td>• Reverse backup of data at customer or customer’s data center</td>
<td>• Agreement with cloud provider for the reverse backup in cloud agreement</td>
<td>• Requires a lot of attention from customer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Fallback system</td>
<td>• Agreement with consultant for the establishment of the fallback system</td>
<td>• Requires regular 'maintenance' such as checking reverse data backup and testing the fallback system</td>
</tr>
</tbody>
</table>