

Switzerland

author:[Jaccard,Michel] author:[Ancelle,Juliette]

Introduction to software protection under Swiss Law

Applicable Law

Under Swiss law, the protection of software is principally regulated by the Federal Copyright Act of October 9, 1992¹, last revised on July 1st, 2008 (“SCA”).

Object of Protection

Definition of software

Pursuant to Section 2, paragraph 3 SCA, computer programs are considered to be works under Swiss law, and as such, can be protected by copyright². Computer programs are purposely protected in a separate category of works from literary works³. The notion of computer programs as used in the Copyright Act includes any completed process written in a programming language with the purpose of performing a designated task⁴. The protection of computer programs comprises the protection of the source code and of the object code of the computer program, but it does not cover the ideas or principles on which the software is based (algorithm, formula, etc.)⁵. It is also generally understood that the copyright protection of computer programs does not extend to the development nor the user’s documentation, which can however be copyrighted separately to the extent that it meets the requirement of protection for literary works as stated in Section 2 paragraph 1 SCA⁶.

For the purpose of this Swiss chapter, the terms of “software” or “computer software” will be deemed equivalent to “computer program”, as defined above. We are aware of the distinction drawn between a software and a computer program in relation with Open Source and within numerous Open Source licenses⁷. A debate exists among Swiss scholars regarding this differentiation, the main issue being related to the inclusion of the documentation in the notion of software and therefore in the definition of protected work, with the consequence that the documentation should be covered by the exclusive rental right (see I.4 (b)

¹Swiss Federal Act on copyright and neighboring rights Swiss Copyright Act (or SCA) — RS 231.1.

²Section 2, paragraph 3 SCA: “Computer programs (software) are also considered as works”.

³D. Barrelet / W.Egloff, *Le nouveau droit d’auteur*, Commentaire de la Loi Fédérale sur le droit d’auteur et les droits voisins, 3rd ed., Berne 2008, 2 No. 23; ATF 125 III 263, 269.

⁴Message from the Federal Council, FF 1989 III p. 488 ; V. Salvadé, *La protection du support multimédia au regard du droit d’auteur*, in *Semaine Judiciaire (SJ) 1996*, p. 258.

⁵E. Neff / M. Arn, *Urheberrechtlicher Schutz des Software*, in: *Schweizerisches Immaterialgüter- und Wettbewerbsrecht*, II/2, p. 121ss.

⁶F. Dessemontet, *Le droit d’auteur*, Lausanne 1999, p. 84ss.

⁷See in particular the Open Source Definition of the Open Source Initiative (<http://www.opensource.org/docs/osd>) and the GNU GPL vers. 3 (<http://www.opensource.org/licenses/gpl-3.0.html>).

below). The terms of “software” and “computer program” are however being used interchangeably in the French version of the Copyright Act as well as in the Message of the Federal Council which preceded the adoption of the revised Act, and we will therefore consider both terms to fall under the same definition in relation to Swiss copyright law.

Protected software

In order to be protected under Swiss copyright law, computer software needs to meet the requirement of individual character, which is the basic requirement for each protected work⁸. Such individual character should only be found in the structure of the program, and not in the ideas or the software’s algorithms or logic, which are not protected by copyright. This requirement for computer programs must however be understood as a requirement of novelty or an absence of triviality⁹. Thus, computer software will be deemed to possess sufficient individual character if experts find it original and different from other existing software by the comparison of their source and object codes¹⁰ and to the extent that the author had sufficient leeway at his/her disposal in the writing of the codes¹¹.

Unprotected and public domain software

As a consequence of the requirements set forth in Section 2 of the Copyright Act, Swiss law shall not protect software that does not possess an individual character. Thus, a computer program that professionals will find trivial in relation to preexisting ones, when comparing their codes, and for which the author has not showed any creativity in the writing of the code, shall not be considered as a protected work under copyright law, and can therefore be freely used, to the extent that it is not limited by existing covenants governing it or by the general rules of unfair competition¹². Furthermore, the protection of software developed by a computer is discussed under Swiss law, as some specialized authors consider that the human intervention is too remote to trigger the legal protection of the work¹³.

Under Swiss law, two categories of software will be considered as public domain: (i) software for which the protection has expired through lapse of time (50 years after the death of the author; see I.6 below for more details), and (ii) software

⁸See Section 2, paragraph 1 SCA : “A work is understood, whatever its value or purpose may be, as any creation of the mind, whether literary or artistic, that has an individual character”.

⁹Message from the Federal Council of June 19, 1989, FF1989 III 465, ad p. 508.

¹⁰“AUTO-CAD” case, *Revue Suisse de Propriété Intellectuelle* (RSPI) 1989, p. 58.

¹¹ATF 113 II 196; Message from the Federal Council of June 19, 1989, FF1989 III 465, ad p. 508.

¹²The applicable rules of unfair competition can be found in the Federal Act on Unfair Competition (UCA) of December 19, 1986, RS 241.

¹³F. Dessemontet, *Commentaire romand Propriété Intellectuelle*, Lausanne 2013, LDA 2N 63.

for which the author has voluntarily given up his/her rights and which he/she has offered for the use of the public. This type of public domain software is characterized by the absence of exclusive enforceable rights on the software and the free availability of the computer program, but not necessarily of the source code. Due to the absence of a registration requirement for copyrighted works in Swiss copyright law however, the author of the software who wishes to make it a public domain software must express it clearly, for instance through a notice attached to each copy of the software¹⁴. The public domain software must be distinguished from the Free and Open Source Software (FOSS)¹⁵: in the case of FOSS, the author or co-authors license their rights to the users by giving access to the source code but remain the valid holders of the copyright on the software, whereas public domain software can be validly used, modified, or published without the need for the issuance of a license.

Owner of rights

Author as a natural person

Pursuant to Section 6 SCA, an author is the natural person who created the protected work¹⁶. Thus, in the case of computer software, the programmer who created the software will be considered to be its author in the sense of copyright law, and therefore will own the copyright to the protected work¹⁷. As a result and contrary to what is the case in the USA for example¹⁸, a corporate entity or a legal person cannot be the “original” author of a computer program, even when it has for instance financed the development of the program. In such a case however, the corporate entity may benefit from the mechanism of legal assignment of certain exclusive rights in favor of the employer regarding software created by an employee (see section I.3 below).

Joint work : coauthors

In the development of a computer program, there is often more than one person involved, and therefore more than one author in the sense of copyright law. Most of the time, the developed software will be considered as a joint work, created by several coauthors. Pursuant to Section 7 SCA, a coauthor is a natural person who contributed to the creation of a work by bringing in his/her creative input¹⁹.

¹⁴M. J. Widmer, Open Source Software — Urheberrechtliche Aspekte freier Software, Bern 2003, p. 44.

¹⁵Free and Open Source Software is software subject to an Open Source license; See definition below in II.1.

¹⁶Section 6 SCA : “The author is the natural person who created the work”.

¹⁷Section 9 SCA : “The author has the exclusive right on his/her work and the right to be credited as its author” ; Barrelet / Egloff, Le nouveau droit d’auteur, 17 No. 2.

¹⁸Section 201 (a) of the US Copyright Act of 1976, Pub. L. 94-553, 90 Stat. 2541, as amended through July 1, 2001; Community for Creative Non-Violence v. Reid, 490 U.S. 730, 109 S.Ct. 2166, 104 L.Ed.2d 811 (1989).

¹⁹Section 7, paragraph 1 SCA : “When several persons have contributed in their capacity as authors, the copyright belongs to them jointly” ; Barrelet / Egloff, Le nouveau droit d’auteur, 7 No. 4.

The two main criteria are thus (i) the objective creative collaboration but also (ii) the intent of the authors to create a work together. The coauthor of a protected work must be distinguished from an auxiliary, who simply performs a task he/she has been entrusted with, with no provision for any creative input, and who therefore does not own any rights to the protected work²⁰. In addition, the determining factor that will allow a work to be deemed a joint work and not a derivative work is the accepted and voluntary joint creative collaboration on a common and unique creation within an agreed period of time.

In the case of a joint work, the coauthors are the joint holders of the copyright, with the consequence that they can only use the joint work in common or with each coauthor's agreement and that one coauthor cannot dispose of his/her share without the consent of the others²¹. The coauthors have however a duty to make the use of the work possible and they cannot withhold their authorization of use without cause²². Each coauthor has the standing to represent the joint tenancy of coauthors in case of infringement, but this does not affect the joint ownership of the copyright by all coauthors²³.

The legal regime of the Copyright Act applicable to coauthors has been construed by the Swiss Supreme Court as establishing a "sui generis joint ownership", with the consequence that it does not correspond to any regime of co-ownership already existing under Swiss law²⁴. Aside from the mandatory rules on co-authorship and in particular on the joint ownership of the joint work set out in Section 7 SCA, the coauthors may choose to regulate their relationship via an express agreement, regulating for instance the decision making process or the liability of the coauthors. In the absence of a specific agreement, the coauthors shall form an *unregistered partnership*, regulated by Sections 530 to 551 of the Swiss Code of Obligations (SCO)²⁵, the main consequence being that all decisions are to be made unanimously (Section 534 SCO).

Copyright assignment

Another specificity of computer programs in comparison to other protected works relates to the context of their creation. Most of the time, software is developed within a company by programmers who are employees and have in fact been hired for the purpose of developing software. Thus, taking into consideration the tension between the general principle of designating a natural person only as an author and the economic need of companies investing in the software development, the Swiss legislator has established an automatic legal assignment of rights — absent an agreement to the contrary between the parties — according

²⁰Case 4A_638/2009 "Guide Orange" from the Swiss Supreme Court from April 1st, 2010, paragraph 4.3.

²¹ATF 129 III 715, JdT 2004, p. 271, cons. 3.1.

²²Section 7, paragraph 2 in fine SCA.

²³Section 7, paragraph 3 SCA ; Case 4C.138/2003 "Malbuner II" of the Swiss Supreme Court, sic ! 2004.

²⁴ATF 129 III 715, JdT 2004, p. 271, cons. 3.2.

²⁵Swiss Code of Obligations (or SCO) — RO 27 321.

to which only the employer is entitled to benefit from the rights of use on a software created by the employee as part of his/her work obligations²⁶. The two criteria for the application of this provision are therefore that the programmer developed the software (i) when he/she was *at the employer's service* and (ii) while *performing his/her employment obligations*. This rule thus only applies when an employment relationship in the sense of Swiss labor law exists, and, as already mentioned, can be modified by an agreement between the parties²⁷.

Where no employment relationship exists and the software is developed by a self-employed programmer upon a client's request, the determination of the original ownership of the rights will depend on the interpretation of the intent of the parties. Under Swiss law, to determine whether the copyright on the created work has been assigned or whether the author has only granted a license, one must interpret the intent of the parties according to the theory of purpose²⁸, although in case of doubt on the purpose, a court will favor the granting of a license instead of a full assignment of copyright. Swiss copyright law indeed does not incorporate the concept of "work for hire" as it exists in the USA²⁹, and the ownership over the rights will therefore be determined on a case-by-case basis, interpreting the intent of the parties. In the case where it is established that only a license has been granted, the most problematic question remains the question of exclusivity, and whether the parties intended any license of rights to be exclusive. It is generally admitted that exclusivity has been granted in cases where the software incorporates trade secrets, when its development has only been possible thanks to the client's knowledge or when the source code has been handed over³⁰.

Best practices : In light of the above, we would recommend to always specify in writing in case of outsourced development work : (1) who will be the author of the created software, (2) whether a license will be granted for the created software, and (3) in such a case, whether the license will be exclusive or not.

Scope of protection

Exclusive rights

As is the case for any protected work, the author of computer software benefits

²⁶Section 17 SCA : "The employer is the only one authorized to exercise the exclusive rights of use on software created by the worker in the performance of his/her activities at the employer's service and consistently with his/her contractual obligations"; J.M. Lutz, *Les programmes d'ordinateur*, p. 45s.

²⁷Barrelet / Egloff, *Le nouveau droit d'auteur*, 17 No. 6 ; R. Wyler, *Droit du travail*, Berne 2008, p. 3810-1.

²⁸The theory of purpose (théorie de la finalité) rests upon the purpose of the contract and assumes that, in case of doubt, the transfer of rights intended by the parties extends only to the rights that are necessary to achieve that purpose; K. Troller, *Manuel du droit Suisse des biens immatériels*, vol. II, Bâle 1996, p. 695; Sic! 1997, p. 382.

²⁹§ 101 of the Copyright Act of 1976, Pub. L. 94-553, 90 Stat. 2541, as amended through July 1, 2005.

³⁰J.M. Lutz, *Les programmes d'ordinateur*, p. 187.

from exclusive rights over his/her work. Thus, as provided for in Section 10 SCA, the author of computer software has the exclusive right to decide how, when and whether his/her work shall be used³¹. Section 10 SCA contains an illustrative list of what is covered by this exclusive right of use of the work³². These exclusive rights are the patrimonial rights of the author, as opposed to his/her moral rights, and can be freely assigned or granted to others by the author, usually in exchange for financial compensation. The patrimonial exclusive rights are, however, not limited to lucrative use and also cover not-for-profit use of the work³³. As regards computer programs, the most relevant right is the right to make copies and to distribute them, which includes online distribution according to the prevailing opinion among Swiss authors³⁴.

Specific rights for computer software under Swiss law

In addition to the general patrimonial exclusive rights to which every author is entitled under the Swiss Copyright Act, authors of computer software benefit from two additional exclusive rights: (i) an exclusive rental right³⁵ and (ii) a utilization right³⁶.

Under Swiss law, the right of distribution does not include the exclusive right to rent out the protected work. Thus, only authors of computer software may own an exclusive rental right on their created work³⁷. This provision means that the sale of a copy of the software does not deprive its author of the right to rent it out, i.e. to allow a third party to use it in exchange for financial compensation. Consequently, and contrary to what is the case for other works protected by copyright³⁸, the user who acquired the software is not entitled to rent it out without infringing upon the author's exclusive right, unless an express license has been granted³⁹.

The second exclusive right specific to computer software is a utilization right. Pursuant to Section 12, paragraph 2 SCA, computer software that has been lawfully sold *can be used or sold again*. Swiss copyright law therefore expressly protects the right of the lawful owner of the protected software to use it⁴⁰.

³¹Section 10, paragraph 1 SCA : "The author has the right to decide if, when and how his/her work shall be used".

³²Pursuant to Section 10, paragraph 2 SCA, the exclusive right to use the work includes but is not limited to the right to make copies of the work, the right to distribute the copies, the right to make the work available to the public (...).

³³C. Gasser, *Der Eigengebrauch im Urheberrecht*, p. 39.

³⁴Barrelet / Egloff, *Le nouveau droit d'auteur*, 10 No. 16; R. Hilty, *Sic !* 1997, p. 134 ; E. Neff/ M. Arn, *Urheberrechtlicher Schutz der Software*, in : R. von Büren / L. David (ed.), *Schweizerisches Immaterialgüter- und Wettbewerbsrecht*, vol. II/2, Basel 1998, p. 230ss.

³⁵Section 10, paragraph 3 SCA.

³⁶Section 12, paragraph 2 SCA.

³⁷The Swiss legislator decided to include this additional right to harmonize its legislation with the European Directive on the legal protection of computer programs, in particular its Section 4.

³⁸See Section 13 SCA.

³⁹J.M. Lutz, *Les programmes d'ordinateurs*. p. 189.

⁴⁰Concerning most works protected by copyright, the owner of a copy of the work will only

This use is however limited by the definition of what a lawful utilization of the software is: pursuant to Section 17 of the Ordinance on Copyright and Neighboring Rights⁴¹, lawful utilization of computer software is utilization by the legitimate buyer in compliance with the contractual conditions of the sale. This specific right of utilization for computer software compensates for the fact that the private use exception for copyrighted work does not apply to software⁴².

Moral Rights

The author of a computer program, like any author in the sense of the Swiss Copyright Act, owns moral rights to his/her creation. This notion covers two main exclusive rights: (i) the right to be credited as the author of the work⁴³ and (ii) the right to the respect of the integrity of the work⁴⁴. The moral rights are strictly attached to the personality of the author or of his/her heirs and therefore cannot be alienated nor transferred to a third party by any means⁴⁵. The transfer of the patrimonial rights does not entail the transfer of the moral rights.

Thus, regarding the right to paternity of the work, the author of the work will always be entitled to be credited as the author but will also have the right to choose to remain anonymous. The author can however waive this right, which is often the case for employees, renouncing in their employment contract to exercise their right to be credited for their creation, although they remain the sole author(s) of the work⁴⁶.

As regards the right to the integrity of the work, the author can always authorize third parties to modify his/her work by contract but pursuant to Section 11, paragraph 2 SCA, the author retains the right to oppose any modification that would harm his/her personality. This right can be waived by the author, but will in any case have very limited impact in relation to computer software, since any modification of the code or of the programming language will hardly be found to be harmful to the author's personality⁴⁷.

enjoy the work, i.e. read a book, listen to a record or watch a movie, but with computer software, the legislator wanted to protect a professional use of the protected work.

⁴¹Ordinance on copyright and neighboring rights (CRO) from April 26, 1993 and last revised on July 1, 2008, RS 231.11 ; Barrelet / Egloff, *Le nouveau droit d'auteur*, 12 No.13.

⁴²Section 19, paragraph 4 SCA ; Barrelet / Egloff, *Le nouveau droit d'auteur*, 19 No. 29.

⁴³Section 9 SCA : "The author has the exclusive right on his/her work and the right to be credited as the author".

⁴⁴Section 11 SCA : "The author has the exclusive right to decide : (a) if, when and how his/her work can be modified ; (b) if, when and how his/her work can be used for the creation of a derivative or a collective work."

⁴⁵Case "Baupläne", *Sic !* 2004, p. 299.

⁴⁶R. Wyler, *Droit du travail*, Berne 2008, p. 381 ; V. Salvadé, *La protection du support multimédia au regard du droit d'auteur*, in *SJ* 1996, p. 265.

⁴⁷V. Salvadé, *op. cit.*, p. 265.

Exceptions to the exclusive rights

Two provisions of the Copyright Act are to be mentioned as exceptions to the exclusive rights.

Firstly, Section 21 SCA authorizes the lawful owner of the software to obtain information on the software's interfaces through the reverse engineering of the program's code. The reverse engineering, which can be operated by the owner of the software or by an authorized third party, is however limited by law to the case of obtaining necessary information for the development or use of interoperable software and this information cannot be disclosed to unauthorized third parties⁴⁸. Thus, the reverse engineering of software is only valid under Swiss law when the two following conditions are met: (i) the information on the interface must be necessary for the development of interoperability, and (ii) the reverse engineering must be operated by an authorized person, i.e. the lawful owner of the software or a third party mandated by him/her.

The second relevant exception to the exclusive rights of the author resides in the right of the person authorized to use the software to make a backup copy⁴⁹. This right cannot be limited by contract, but is limited to *one copy only*, and only belongs to the lawful user of the software as described above⁵⁰. Any copy made for any other purpose, even by the lawful user, will be deemed an infringement on the author's exclusive right. The rightful user is also authorized to make temporary copies, insofar as they are technically necessary for the transmission of the work to a third party or for any other lawful utilization of the work, and that they do not have any independent economic significance⁵¹.

Term of protection

Under Swiss law, the general term for protection of copyright for computer software is fifty (50) years after the death of the author⁵². This differs from the longer term of protection for other copyrighted works that lasts for seventy (70) years after the death of the author.

In the case of co-authorship, the protection of the software ends fifty (50) years after the death of the last living coauthor⁵³. If each coauthor's input, however, can be separated from the rest of the work, the protection for each separate input will end fifty (50) years after the death of its author⁵⁴.

As regards to derivative work, the term of protection will differ for the original work and for the additional creation, each part being protected by copyright for

⁴⁸G. Fröhlich-Bleuler, *Pratique Juridique Actuelle (AJP/PJA)* 1995, p. 578.

⁴⁹Section 24, paragraph 2 SCA.

⁵⁰G. Fröhlich-Bleuler, *PJA* 1995, p. 573.

⁵¹Section 24a SCA.

⁵²Section 29, paragraph 2, lit. a SCA ; see also J-M Lutz, *Les programmes d'ordinateurs*, p. 201.

⁵³Section 30, paragraph 1, lit. a SCA.

⁵⁴Section 30, paragraph 2 SCA.

fifty (50) years after the death of its respective author⁵⁵.

Special measures

As a signatory of the WIPO Copyright Treaty (WCT)⁵⁶ and the WIPO Performances and Phonograms Treaty (WPPT)⁵⁷, Switzerland has adopted a provision in the Copyright Act for the protection of effective technological measures⁵⁸. Pursuant to Section 39a SCA, the removal or circumvention of effective technological measures used for the protection of copyrighted works or other protected objects is prohibited. Swiss law however provides for an exception to this rule: circumventing protective technological measures will not be punished by law if it is performed for the sole purpose of allowing an authorized use of the protected work⁵⁹.

The Swiss Copyright Act makes it a criminal offence to violate the provision on the protection of technological measures by circumventing or removing the said effective measures for the purpose of unlawfully using the protected work, but also by selling or offering products allowing such actions (Section 69a SCA). The infringer, who must have acted intentionally to be held criminally liable, will be punished by a fine, or by imprisonment in the case that he/she is a professional infringer. In addition, it must be noted that once an infringement to the rules concerning the protection of technological measures occurs, a presumption exists that it has created a threat to copyright and the author will therefore be entitled to damages without being required to establish any actual infringement to copyright⁶⁰.

Analysis of FOSS under Swiss law

For the purpose of this chapter, the terms below will have the following meanings:

Free and Open Source Software (FOSS)

software subject to an Open Source⁶¹ license granting the right to use, analyze, modify and improve the said software by making its source code freely available to users.

Proprietary software

computer software that is the exclusive property of its developers/publishers

⁵⁵Barrelet / Egloff, Le nouveau droit d'auteur, 30 No. 5.

⁵⁶WIPO Copyright Treaty of December 20, 1996, RS 0.231.151.

⁵⁷WIPO Performances and Phonograms Treaty of December 20, 1996, RS 0.231.171.1.

⁵⁸See Section 11 WCT and 18 WPPT ; See also the Message of the Federal Council for the adoption of the two WIPO treaties and the modification of Swiss copyright law, FF 2006, p. 3297.

⁵⁹M. Jaccard/ J. Heumann, Commentaire romand Propriété Intellectuelle, Lausanne 2013, LDA 39a N 25 ff.

⁶⁰See Section 62, paragraph 1bis SCA ; FF 2006, p. 3297.

⁶¹For a definition of Open Source, please see the "Open Source Definition" by the Open Source Initiative (OSI).

and which can only be copied or used in compliance with the owner's licensing agreements, their source code being rarely made available.

Common legal issues in relation with FOSS

We shall analyze in this section the legal issues regularly discussed in relation to Open Source Software, starting with the question of copyright protection of FOSS and then addressing the questions of enforcement of FOSS licenses and of the consequences of a breach of such licenses under applicable Swiss law.

FOSS under Swiss copyright law

Characterization of FOSS

One major issue in relation with copyright law that takes on an important dimension with respect to Open Source Software is the question of characterization. Indeed, one of the legal difficulties in FOSS consists of determining who is entitled to issue a license or to bring a lawsuit in case of infringement, or in determining the duration of the protection of the software. These issues can be solved by the characterization of the Open Source Software either as a *joint work* or as a *derivative work* as defined in the Copyright Act.

FOSS shall be considered a joint work under Swiss law when two or more persons have worked in common on the development of original and unique software for a determined period of time⁶². Thus, when several developers have worked together for the purpose of creating original software, the result will usually be considered a joint work in the sense of copyright law, each developer being a coauthor.

The question becomes more complicated when unfinished software is created and then made available under an Open Source license for the purpose of being further developed by its users. In such cases, it will be hard to establish that the original developer and the user worked in collaboration with a common purpose, i.e. the creation of original software. Thus, a common creation will rarely be found, and the software which has been modified or improved by the user will most of the time be considered a derivative work in the sense of Section 3 SCA. The creator(s) of this new version of the software will therefore be a new author (or new authors) of a new work, this characterization giving rise to different rights.

Rights and obligations of the original coauthors

The coauthors of a protected work are the joint holders of the copyright pursuant to Section 7 SCA and as such form a sui generis legal tenancy. The co-authorship and its legal consequences will have an impact on two main questions in relation with Open Source: (i) who is entitled to give a license and (ii) who has the power to act in case of infringement?

⁶²Barrelet / Egloff, Le nouveau droit d'auteur, 7 No. 4.

As regards to the first issue, the legal regime that applies to coauthors, which is a regime of joint tenancy imposed by law but not defined by the legislator, implies that no assignment can be awarded nor license granted on the software without the common consent of *all coauthors*⁶³. This system can be highly burdensome, especially in relation with Open Source, where the main purpose is to make software freely available to everyone and to allow users to benefit from further developments. Although the strictness of this rule is partially lessened by the obligation for all coauthors not to withhold their consent without motive⁶⁴, the regime of joint ownership can be extremely limiting, especially in the context of Open Source where it is essential to identify the person entitled to grant the Open Source license, and where the requirement of obtaining all the coauthors' consents would be considered a hurdle to free distribution. The coauthors can, however, derogate to the legal regime with a convention entitling one or several of the coauthors to represent them all for the exercise of rights of use.

Best practices : In order to escape from the burdensome legal regime of coauthorship, we would advise authors of Open Source software to join the Free Software Foundation (FSF) which provides a system of assignment of the author's copyright to the FSF, which will then be in charge of the management of the rights⁶⁵.

Regarding the second issue relating to the ability to bring a claim in case of infringement, Swiss law authorizes any coauthor to act but it will always be on behalf of all the coauthors⁶⁶. This system facilitates the protection of the software by allowing any coauthor to act independently on his/her own and by automatically ensuring that all coauthors will benefit from the claim in case of a positive result. The legal regime however does not create a similar system of solidarity in case of a claim for infringement by a third party against the coauthors: under Swiss law, a claim can only be brought against one coauthor and still be valid, but the coauthor shall only be liable for his/her share of the work — the work being divided in equal shares by the number of coauthors without regard to the actual input of each co-author. Such a system allows a third party to enforce its rights without having to identify all coauthors, but it also protects each coauthor from full liability.

Authors of a derivative work

The characterization of portions of Open Source software as a derivative work will also have an impact on the rights of the author, the main legal issue relating to his/her ability to further distribute FOSS. In the majority of cases, the user

⁶³See I.3 (b) above; G. Fröhlich-Bleuler, *Urheber- und vertragsrechtliche Aspekte der Open Source Software*, in Jörg / Arter, *IT-Verträge*, Bern 2007, p. 194.

⁶⁴Section 7, Paragraph 2 SCA ; M.J. Widmer, *Open Source Software — Urheberrechtliche Aspekte freier Software*, p. 81.

⁶⁵E. Moglen, *Why FSF Gets Copyright Assignment From Contributors*, <http://www.gnu.org/licenses/whyassign.html> ; O. Johnny, M. Miller, M. Webbink, *Copyright in Open Source Software — Understanding the Boundaries*, IFOSS Law Review, Vol. 2, Issue 1.

⁶⁶Section 7, Paragraph 3 SCA.

who further develops Open Source software, either by modifying or improving it, will be considered the author of a derivative work and as such own a copyright on what has been modified — inasmuch as this improvement or modification can be protected by copyright, i.e. is original, which will usually be the case⁶⁷. But if the author wants to distribute the modified software to other users, it is necessary to have the consent of the original author (or coauthors) since the original work is incorporated in the derivative one. For most FOSS licenses, the original author already consents to a further distribution of FOSS by its users, and therefore the consent of the original author will not be an issue (See GPL, Mozilla, etc.). However, it may be the case that the Open Source software has been freely distributed under a certain license that does not allow any further distribution by the end user or any distribution of the modified software⁶⁸, in which case the author of the new derivative work will need to obtain the consent of the original developer(s) to license his/her modified software under a new FOSS license.

Assignement of copyright

As we have seen in section I.3 above, the SCA provides for a special rule regarding computer programs that are created by an employee as part of his/her employment obligations. In such cases, although the programmer remains the author of the protected program, the employer will automatically own all of the exclusive rights of use on such programs. Thus, the employer will be the entity with the authority to validly license the software under an Open Source license, with no need to obtain the employee's consent⁶⁹. The benefit of this rule can be that several developers may create a company together, for which they work and which would automatically hold the right to license the created software, without requiring the conclusion of a convention for the management of the rights of all coauthors for each newly created piece of software, in this way escaping from the burdensome system of co-authorship.

Moral copyrights

FOSS being protected by copyright, the developer is thus an author as we have clearly established, and as such, he/she still owns a moral copyright on the licensed software. Therefore, the modifications or improvements of the software, although authorized by the license, cannot infringe upon his/her moral copyright, and more specifically cannot harm the author's personality⁷⁰. The question of whether a modification of the software will infringe upon the author's personality rights will be established on a case by case basis, and it will

⁶⁷G. Fröhlich-Bleuler, op. cit., p. 194.

⁶⁸See for instance the Apple Public Source License 2.0 or the LaTeX Project Public License (LPPL) (<http://www.opensource.org/licenses/category>).

⁶⁹Section 17 SCA ; G. Fröhlich-Bleuler, op. cit. p. 195.

⁷⁰Section 11, Paragraph 2 SCA.

very rarely be found under Swiss law in the context of computer programs⁷¹. This is even truer in the case of FOSS, where the source code has been disclosed, and therefore the programming language made available to all users for the purpose of being modified. In such a context, it will be extremely difficult to establish any harm to the developer's personality, the only case being when his/her reputation as a programmer would be damaged by the subsequent modifications. Such harm will however be highly difficult to establish in the case of FOSS, and more particularly if the software is made available under a GPL, since this license creates an obligation for subsequent users/modifiers to give explicit notice of their modifications⁷², with the consequence that the changes will not be attributed to the original author and therefore not harm his/her reputation. It is however important to bear in mind that the broad right to modify the original software granted by a FOSS general license can still be limited by the original author's enforcement of his/her moral right⁷³.

Enforcement of FOSS licences

General contract law issues

Based on the above definition, the specificity of FOSS is that the source code of a computer program is made freely available to users through the delivery of a FOSS license. Yet, in order to enforce such a license, the latter needs to have been validly concluded between the concerned parties and its content needs to comply with the local applicable law — which also raises questions as regards to the law applicable to the contract and more general issues of conflict of laws (see IV.1 below).

Assuming that the FOSS license is subject to Swiss law, the main contract law issues in relation with FOSS would therefore relate to the valid conclusion of the said license, the identification of the parties to that license, as well as the validity of its content.

As a preliminary remark, it should be noted that Swiss authors generally agree that a FOSS license should be analyzed in the context of contract law and should not be considered to be a unilateral legal act⁷⁴. Under Swiss law, a contract is defined as a bilateral or multilateral legal act by which the parties exchange the expression of their matching intents (meeting of the minds)⁷⁵. Pursuant to that definition, a first issue to consider is how the FOSS license is validly concluded, i.e. how does the meeting of the minds occur?

⁷¹M.J. Widmer, Open Source Software, p. 134 ; some Swiss authors even deny that any modification of computer software could ever infringe upon the author's moral rights.

⁷²Section 5 GPL v3 : "The work must carry prominent notices stating that you modified it, and giving a relevant date" ; Section 2 GPL v2 : "You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change".

⁷³A. Metzger & T. Jaeger, Open Source Software and German Copyright law, IIC 2001 Heft 01, p.65-6.

⁷⁴G. Fröhlich-Bleurer, op. cit. p. 184.

⁷⁵Section 1 of the Swiss Code of Obligations (SCO).

Conclusion of a binding contract

Under Swiss law, it is generally admitted that a contract can be validly concluded on the Internet, simply by clicking on a “Download” or “I agree” button⁷⁶. Therefore, a FOSS license can be validly concluded between the author and the user when he/she downloads the Open Source software after reading the license to which the use of the software is subject and clicking on the “I agree” button. The FOSS license is thus concluded between the right holder and the user by the delivery of the software to the user or his/her access to it. The question of the conclusion of the license is however more complicated when the software is transferred without the user having knowledge of the FOSS license governing it. It is indeed common that a user acquires Open Source software without coming across the FOSS license under which the rights are granted⁷⁷.

In such a case, Swiss scholars consider that the original author of FOSS and the end user are not bound by a license or any other agreement⁷⁸. The consequence of such a situation is that the user will not benefit from the rights of the FOSS license as long as he/she is not aware of its existence: in other words, the user will not be entitled to modify or improve the software until he/she is bound by the conditions of the FOSS license, with the risk arising that any unauthorized modification of the source code will be considered to constitute copyright infringement. Thus, should no copy of the FOSS license be communicated to the user together with the software, then no valid license can be concluded and therefore no modification of the software will be considered valid and covered by the FOSS license. The content of the FOSS license will be considered as the expression of an offer by the licensor (i.e. the author of the software or the holder of the copyright) to conclude an agreement governing the use of the software. Indeed, under Swiss law, an offer in the sense of contract law does not need to be addressed to a specific person and it can be expressly or tacitly accepted by any recipient⁷⁹. Thus, the user who complies with the provisions and conditions contained in the FOSS license will be considered as having validly accepted the offer and the license will be validly concluded and binding upon the parties⁸⁰. This raises however the question of the duration of the validity of the offer and the possibility for the licensor to withdraw such an offer. According to Swiss scholars, the offer will remain binding upon the licensor as long as the user has in good faith the possibility of validly concluding

⁷⁶A. Cherpillod Giacobino, *Internet dans la conclusion du contrat et les solutions de paiement*, in SJ 2003 II p. 405.

⁷⁷On that matter, see also Section 9 of the GNU GPL v3 that states that “You are not required to accept this license in order to receive or run a copy of the Program”.

⁷⁸G. Fröhlich-Bleuler, *op. cit.* p. 188-9.

⁷⁹Section 1, paragraph 2 Swiss Code of Obligations (SCO) : “This manifestation can be express or implied” ; for a complete discussion, see M. Jaccard, *La conclusion de contrats par ordinateur — Aspects juridiques de l’échange de données informatisées (EDI)*, ASR, Berne 1996.

⁸⁰Under Swiss law, the expression of the party’s intent can result from concrete actions that reveal its intent to be bound by the submitted offer ; P. Engel, *Traité des obligations en droit Suisse*, Berne 1997, p. 130; ATF 123 III 53 JdT 1999 I 179.

the contract by the simple exercise of the conditions of use, and it will therefore be extremely difficult for the licensor to withdraw his/her offer and avoid the conclusion of a license agreement on the Open Source Software⁸¹.

If no license agreement has been concluded between the user and the licensor, the modification of the software by the user could constitute a copyright infringement, which may trigger civil as well as criminal liability. The user's liability, however, will most probably be reduced or at least strongly limited in such circumstances, a Court being likely to take into consideration the concomitant liability of the author who did not properly attach the Open Source license to the distributed software, therefore giving rise to legal uncertainty.

Best Practices : It is always recommended to accompany the software with the FOSS general license under which it is offered to be granted, including licenses that do not require so, in order to avoid unclear situations where it is unsure whether a binding contract exists between the parties, because it will otherwise depend on the interpretation of the user's actions to determine whether he/she intended to be bound or not.

Determination of the parties to the FOSS license

Along with the question of whether a valid FOSS license has been concluded with the transfer of the software, the other issue to be considered in relation to Open Source is to determine who is a party to that license agreement.

The situation is particularly complicated when the user himself/herself starts licensing the modified software and so on. As already mentioned, a FOSS license is always concluded between the *author* of the Open Source software, who is the licensor, and the *user* of the software who is the licensee. When there are several authors of the work however (either coauthors or several authors of a derivative work), the license must be granted by all authors and the conditions of use of the FOSS license towards all of them will be binding upon the user. This situation is generally solved by the license itself which will contain a clause according to which the user, by accepting the conditions of the license, will automatically be bound to all previous authors of the software⁸². This mechanism is authorized under Swiss law and therefore creates a valid contractual relationship between all parties.

Violation of the FOSS license — Breach of contract or Copyright infringement?

The question of determining whether a violation of the FOSS license constitutes a breach of contract or a copyright infringement may have a certain impact in Switzerland as regards procedural issues as well as potential remedies to which

⁸¹G. Fröhlich-Bleuler, op. cit. p. 221-2.

⁸²See Section 10 GNU GPL v3 : "Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License" or Section 6 of the GNU GPL v2 : "Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions".

the authors may be entitled. The protection for Open Source software can be based on both legal grounds, and the claim will depend on the type of violation that has occurred.

Thus, the copyright holder will be entitled to invoke a copyright infringement for the cases in which the user does not give proper credit notice or modifies the software when he has only been given the right to use it, whereas the claim of a user brought in order to obtain the disclosure of the source code will only be based on contract law since it only constitutes a breach of the license agreement and not a copyright infringement. It is however conceivable that an action consists of both a breach of contract and an infringement to copyright, in which case the claimant will be able to invoke both violations and the Court will usually have proper jurisdiction on both.

Some differences however exist between a claim based on copyright law and a claim based on contract law, in particular from a procedural point of view. The Swiss legal system being a federal system, a claim is usually brought in front of a local cantonal court before ending up in front of the Swiss Supreme Court; for copyright claims however, there is only one unique competent jurisdiction at the cantonal level before an appeal can be brought in front of the Swiss Supreme Court, whereas a claim based on contract law may go through several cantonal appeals before ending up in front of the Swiss Supreme Court⁸³.

In addition to this procedural difference, the remedies in contract law and copyright law that would be applicable to a violation of a FOSS license can differ in some respects, reason why the claimant will often use both grounds in front of the court⁸⁴. Indeed, although damages can be asked both in case of copyright infringement and breach of contract, contract law's sole alternative remedy will be limited to injunction for specific performance, whereas the SCA provides for more specific remedies, such as the prohibition of further distribution of the infringing copy and/or its destruction⁸⁵. In case of copyright infringement, the author will also be entitled to ask for a declarative judgment of his/her quality of author⁸⁶, or to ask for an injunction to prevent the infringement from happening or to make it stop in cases where the infringement is still producing effects⁸⁷. It should also be noted that Swiss copyright law additionally makes it a criminal offence to perform most acts of copyright infringement⁸⁸.

⁸³See Section 64 SCA. A unified civil procedure act will come into force on January 1st, 2011 and replace the 27 civil procedure codes currently in force within the country ; this legal modification will however not affect the system of one unique competent jurisdiction as regards to copyright which will remain into place.

⁸⁴M.J. Widmer, Open Source Software, p. 106-7.

⁸⁵Section 97 and 107-109 SCO.

⁸⁶Section 61 SCA.

⁸⁷Section 62 SCA.

⁸⁸Section 67 SCA makes it a criminal offence to perform the following actions without right : (1) use a work under a false designation, (2) disclose a work, (3) modify a work, (4) use a work for the purpose of creating a derivative work, (5) make copies of a work, (6) offers the work to the public or distribute it by any other means, publicly represent or perform a work, broadcast the work (...) rent a software.

Warranty and Liability

As regards FOSS, there are two main aspects to be analyzed in this section: one relates to the warranty granted or more generally excluded by the licensor, and the second relates to the validity of a waiver of the licensor's liability.

Exclusion of warranty

Under Swiss law, the regime applicable to the exclusion of warranty will depend on the characterization of the license that will determine the applicable contract law rules. Swiss contract law sets out general rules applicable to all types of contracts and specific rules regulating specific types of contracts (“contrats nommés”)⁸⁹. License agreements are not specifically regulated under Swiss contract law but are considered as “sui generis contracts”⁹⁰ and are therefore subject to the general rules on contracts. In some instances however, a court may apply some rules of specifically regulated agreements by analogy to a license agreement when it is deemed appropriate⁹¹, and Swiss legal scholars have therefore identified specific rules that may be applicable to FOSS licenses.

Although the Open Source Definition provides in the Section 1 that no royalties or fees should be perceived by the licensor under a FOSS license for the selling or giving away of the Open Source software⁹², the distribution of such software can be included in the sale of an aggregate software which can be done against payment or a fee may be charged for the physical act of the transfer. In such cases, Swiss scholars have found that the rules of sales contract could be applicable to the FOSS license, which would be subject to the limitations of the exclusion of warranty⁹³. Pursuant to Section 199 SCO, a clause excluding all warranties in a sales contract is void in the case of fraud. Thus, a clause such as Section 15 of the GNU GPL v3 or Section 11 of the GNU GPL v2 excluding all kinds of warranty could be found non-valid under Swiss law if the transfer of the Open Source Software incorporates elements that can be assimilated to a sales contract and would therefore not be enforceable in cases of fraud. The invalidity of the clause excluding all warranties will however not affect the validity and enforceability of the rest of the license pursuant to the principle of severability.

On the other hand, should the Open Source software be transferred entirely free of charge, such a transfer could fall under the rules of donation, applicable by analogy as regards to the rules on the exclusion of warranty⁹⁴. Yet, in case of a donation, Section 248, paragraph 2 SCO makes the exclusion of all warranties

⁸⁹P. Engel, *Traité des obligations en droit suisse*, Berne 1997, p. 175.

⁹⁰ATF 96 II 154, ad p. 156 ; P. Engel, *Contrats de droit suisse*, Berne 2000, p. 774.

⁹¹K. Troller, *Manuel du droit suisse des biens immatériels*, p. 722.

⁹²Open Source Definition, Section 1 : “The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale”.

⁹³U. Widmer, *Gutachten betreffend Rechtsfragen bei Beschaffung und Einsatz offener Software in des Schweizerisches Bundesverwaltung (Projekt OPUS)*, p. 54.

⁹⁴See Section 239 SCO.

valid. In that context, the provisions from the GPL mentioned above, excluding all types of warranties would be deemed valid under Swiss law⁹⁵.

Waiver of liability

In addition to the exclusion of warranties, most FOSS licenses contain a provision waiving all liability of the author/licensor for damages (e.g. article 16 GNU GPLv3 & article 11 GNU GPL v2). Some of these provisions reserve however the mandatory provisions of the applicable law⁹⁶. Assuming that Swiss law is applicable to a FOSS license, the waiver of liability would not be valid in case of fraud or gross negligence⁹⁷. Any other exclusion of liability for damages resulting from the use of the Open Source software, which is not covered by the cases of gross negligence and fraud, shall therefore be considered valid⁹⁸.

The copyleft principle

A “copyleft” license is a license by which the licensor grants the right to use, distribute and modify the computer program to a licensee, under the condition that all further distributions of the modified software make the work available in a form that facilitates further modification and use the same license. Not all FOSS licenses are “copyleft” licenses (for instance the BSD and the Apache licenses are not “copyleft”, whereas the GPL and the LGPL are). In the absence of a “copyleft” clause, a user that has modified the Open Source software is entitled to integrate his/her new work in another program and sell it as proprietary software, or simply to refuse access to the source code in further distributions of the work⁹⁹.

One issue related to the “copyleft” principle is the question of the compatibility of various FOSS licenses with each other. A license that does not incorporate a “copyleft” clause can indeed be considered incompatible with other FOSS licenses, which can lead to limitations for the re-distribution of the software by the user¹⁰⁰. The issue can however be solved conventionally by a clause integrated in the Open Source License that will determine which other Open Source licenses are compatible and can therefore be used by the users to license their modified version of the software¹⁰¹.

This principle raises also other legal issues, in particular as regards the integration of a part of Open Source software into proprietary software : is the user

⁹⁵M.J. Widmer, Open Source Software, p. 181.

⁹⁶See for example Section 16 GNU GPL v3 : “In no event unless required by applicable law or agreed to in writing will any copyright holder ?...? be liable to you for damages ?...?”.

⁹⁷See Section 100 SCO ; W. Straub, Informatik Recht : Einführung in Softwareschutz, Projektverträge und Haftung; U. Widmer, Projekt OPUS, p. 54.

⁹⁸M.J. Widmer, Open Source Software, p. 182.

⁹⁹G. Fröhlich-Bleuler, op. cit. p. 207.

¹⁰⁰On that issue, a list of all FOSS licenses compatible with the GPL has been published here : <http://www.gnu.org/licenses/license-list.html#GPLCompatibleLicenses>.

¹⁰¹See G. Fröhlich-Bleuler, Die neue Version des GNU Public License, in Jusletter 21. Juli 2008.

under the obligation to make the full new software available under a FOSS license or is it limited to the part of the code developed under an Open Source license? Swiss scholars agree to say that in such cases, the new software constitutes a derivative work in the sense of copyright law, and therefore, the part of the code originating from Open Source software remains subject to the conditions of the Open Source license that has been granted by the original author.

Remedies — Calculation of damages in case of infringement

One of the remedies in case of breach of the FOSS license being the possibility to claim for damages, the delicate question consists in determining the amount of these damages.

A claim for damages, in cases of copyright infringement as well as in cases of breach of contract, is regulated by the general rules of the Swiss Code of Obligations (SCO). Pursuant to Section 42 SCO, the claimant has the burden to prove the existence of a damage. The amount of the damages corresponds to the difference between the current capital of the claimant and the capital he/she would own if the breach had not occurred¹⁰². In case of infringement to intellectual property rights however, it is not always easy for the claimant to compute the claimed damages, and Swiss law does not require him/her to file a claim stating a precise amount¹⁰³. The claimant must however provide the Court with sufficient elements to calculate the amount of damages to award, based on the full instruction of the case and the review of the evidence.

In cases of copyright infringement, the most common basis for a court in computing damages is the amount of failed opportunities of the claimant due to the copyright infringement¹⁰⁴. It may, however, be extremely difficult to establish that the infringement resulted in the loss of opportunities for the copyright holder, and the amount of damages can also be computed on the basis of the royalties that the right holder would have received, should the protected work have been used under a validly granted license¹⁰⁵. In the absence of any case in relation to Open Source software in Switzerland, it is difficult to determine in advance which method a court would use to compute damages for the case of a breach of a FOSS license. It would however most likely include the judicial costs of the prevailing party, as it is common in Switzerland for the losing party to bear the judicial costs of the adverse party¹⁰⁶.

FOSS case law in Switzerland

As of today, we are not aware of the reporting of any case law on FOSS in Switzerland. A decision was rendered by the Swiss Federal Administrative Tri-

¹⁰²ATF 122 III 463.

¹⁰³K. Troller, *Manuel du droit suisse des biens immatériels*, p.1043.

¹⁰⁴ATF 63 II 277, ad. p. 280.

¹⁰⁵K. Troller, *Manuel du droit suisse des biens immatériels*, p.1046; Y. Benhamou, *Dommages-intérêts suite à la violation de droits de propriété intellectuelle*, Genève 2013, p.125 ff.

¹⁰⁶ATF 81 II 534, JdT 1956 I 269 ; Section 106 of the new unified Civil Procedure Act (CPA).

bunal on July 2, 2009, and later by the Swiss Federal Tribunal on March 11, 2011¹⁰⁷ and involved, among others, Red Hat Limited as a plaintiff in this case against the Federal Bureau for Buildings and Logistic. This case however did not relate to any legal issue in relation with Open Source Software but was in fact about Swiss public procurement issues.

Legal procedures

We will start by analyzing briefly the applicable principles of conflict of laws that would give jurisdiction to a Swiss court and lead to the application of Swiss law. We will then provide the reader with a short overview of possible judicial proceedings in Switzerland in case of infringement or breach of the FOSS license.

Application of Swiss law pursuant to private international law (PIL)

In most cases, infringement and breach of a FOSS license will involve foreign players located in different countries. It is therefore important to determine (i) which court should have jurisdiction over this case and (ii) which substantive law the court will have to apply to the case.

Regarding the competent jurisdiction to hear the case, unless the parties have incorporated a valid choice of court clause in the license agreement, Section 109, paragraph 1 of the Swiss Private International Law Act (PILA) provides that the competent court which should hear intellectual property claims is the Swiss court at the defendant's domicile, or, in the absence of domicile, the court at the place where the protection is sought. This provision designates the competent court to hear cases of *infringement* of intellectual property right and sets as a general rule that the Swiss courts of the domicile of the defendant will have jurisdiction over all cases of infringement, whether there has been an infringement on a Swiss or a foreign right¹⁰⁸. This rule differs in some respect with the conflict rule of the Lugano Convention of September 16, 1988 that sets as the competent jurisdiction to hear infringement cases, the courts of the defendant's domicile or the court of the place where the harmful event occurred¹⁰⁹.

In cases of a contractual breach only of the FOSS license, which does not constitute a copyright infringement, Sections 112 and 113 PILA provide that the court of the domicile or the habitual residence of the defendant, or, if the defendant has no domicile or does not reside in Switzerland, the court of the place of performance of the contract has jurisdiction over the case.

As regards applicable law, the general principle of Swiss rules on conflict of laws is the freedom of the parties to a binding contract to agree upon a choice

¹⁰⁷Case from the Federal Administrative Court (FAC) of July 2, 2009, N B-3402/2009 and case from the Swiss Federal Court of March 11, 2011 No. 2C_783/2010.

¹⁰⁸B. Dutoit, *Droit international privé suisse*, Bâle 2005, 4e édition, p. 348.

¹⁰⁹Section 2 and 5 of the Lugano Convention on jurisdiction and the enforcement of judgments in civil and commercial matters, RS 0.275.11.

of law clause¹¹⁰. This general principle also applies to contracts pertaining to intellectual property, in which a choice of law clause is allowed¹¹¹, but in the absence of which the law of the place where the grantor of the intellectual property right has his/her habitual residence is applicable. In specific cases of contracts pertaining to copyrights, in the absence of a choice of law clause, the applicable law will be the law of the place where the person having control over the server containing information has his/her residence, and not of the place where the actual server is located¹¹². This rule however only applies to the contractual obligations of the parties and a different conflict rule exists for the questions of infringement of an intellectual property right. Pursuant to Section 110 PILA, the law applicable to infringement cases is the law of the State in respect of which intellectual property protection is sought. The parties can however agree upon the application of the law of the forum (*lex fori*) to damages claims *after* the harm has occurred¹¹³. This ability for the parties to choose the *lex fori* is however limited to the question of the computation of damages and does not cover the preliminary question of infringement¹¹⁴.

Overview of the possible proceedings in Switzerland in case of infringement

Civil proceedings

On January 1st, 2011, a new unified civil procedure act came into force in Switzerland, and all cantons are therefore subject to the same procedural rules. The obligation for cantons to have a unique trial court as regards to copyright claims remains in force and has been confirmed by the new unified procedure¹¹⁵. This rule applies to cases on issues of copyright infringement but also on issues of transfer of rights or on licenses to use copyrighted work, and therefore covers claims that could be filed in Switzerland in relation to a FOSS license. Following a decision from the unique cantonal trial court, an appeal will always be possible to the Swiss Supreme Court¹¹⁶.

The filing of provisional measures in relation to copyright claims is possible for the party that shows that a copyright infringement is likely to have occurred or is likely to occur in the near future¹¹⁷. The provisional measures will usually be used for the preservation of evidence, for discovering the origin of counterfeiting goods, but it can also consist in the prohibition to use the contentious work, which will probably be the most useful in software cases¹¹⁸.

¹¹⁰Section 116 PILA ; K. Troller, *Manuel du droit suisse des biens immatériels*, p. 1209.

¹¹¹Section 122, paragraph 2 PILA.

¹¹²B. Dutoit, *Droit international privé suisse*, Bâle 2005, 4e édition, p. 428.

¹¹³Section 110, paragraph 2 PILA.

¹¹⁴B. Dutoit, *Droit international privé suisse*, Bâle 2005, 4e édition, p. 355.

¹¹⁵Section 64 SCA and Section 5 CPA.

¹¹⁶Section 74 of the Federal Act on the Swiss Supreme Court.

¹¹⁷Section 65 SCA ; Decision from the Aargau court from June 5, 2007 in *Sic !* 2008, p. 24.

¹¹⁸Barrelet / Egloff, *Le nouveau droit d'auteur*, 65 No. 5.

Criminal proceedings

Along with civil proceedings that can be initiated in case of breach of a FOSS license, the claimant may also press criminal charges against the potential infringer in front of a criminal court. There will be criminal proceedings only if initiated by the right holder, unless the infringer acted professionally in which case the judicial authorities can charge him/her automatically, and a crime will be found only if the infringer acted *intentionally*¹¹⁹. A new unified criminal procedure code also came into force on January 1st, 2011, and is now applied by all cantons in all criminal cases.

Recommended literature or websites

For more information about Open Source software and applicable Swiss law, we recommend the following specialized literature:

- Mike J. Widmer, Open Source Software — Urheberrechtliche Aspekte freier Software, Stämpfli, Bern 2003.
- Gianni Fröhlich-Bleuler, Urheber - und vertragsrechtliche Aspekte der Open Source Software, in: Jörg/Arter, IT-Verträge, Bern 2007.
- Gianni Fröhlich-Bleuler, Die neue Version des GNU Public License, in Jusletter 21. Juli 2008.
- Ursula Widmer, Gutachten betreffend Rechtsfragen bei Beschaffung und Einsatz offener Software in des Schweizerisches Bundesverwaltung (Projekt OPUS).

¹¹⁹Section 67, 69 and 69a SCA ; Barrelet / Egloff, Le nouveau droit d’auteur, 67 No. 1ss.