

RESEARCH OUTPUTS / RÉSULTATS DE RECHERCHE

Report on study of the compatibility mechanism of the EUPL

BASTIN, Fabian; Laurent, Philippe

Publication date:
2006

Document Version
Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (HARVARD):
BASTIN, F & Laurent, P 2006, *Report on study of the compatibility mechanism of the EUPL: Advice report, European Commission, 11th september 2006.*

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Advice Report

EUROPEAN COMMISSION
Enterprise Directorate General

Report on

Study of the compatibility mechanism

of the

EUPL

(European Union Public Licence)

v. 1.0

11th September 2006

Prepared by:



Fabian BASTIN

Employed as post-doctoral researcher at the CERFACS (European Centre for Research and Advanced Training in Scientific Computing – Toulouse, France).

Philippe LAURENT

Researcher at the CRID (Research Centre in Computers and Law – FUNDP Namur)
Lawyer at the Brussels Bar.

Content of the report

| | |
|---|------------------|
| <u>CONTENT OF THE REPORT</u> | <u>2</u> |
| <u>1. EXECUTIVE SUMMARY</u> | <u>3</u> |
| <u>2. INTRODUCTION AND BACKGROUND</u> | <u>5</u> |
| <u>3. THE “COMPATIBILITY MECHANISM”</u> | <u>6</u> |
| 3.1. THE COMPATIBILITY ISSUE AS A DRAWBACK OF THE COPYLEFT EFFECT | 6 |
| 3.2. UPSTREAM AND DOWNSTREAM COMPATIBILITY ISSUES | 8 |
| 3.3. TERMINOLOGY | 9 |
| 3.4. STAKES OF THE OF THE EUPL’S COMPATIBILITY | 10 |
| 3.5. ALTERNATIVE SOLUTIONS TO THE COMPATIBILITY ISSUE | 10 |
| NON-COPYLEFT SCHEME | 10 |
| DUAL-LICENSING | 11 |
| INCLUSION OF A COMPATIBILITY CLAUSE | 12 |
| 3.6. COMPATIBILITY CLAUSES: A COMPARATIVE ANALYSIS | 12 |
| CECILL LICENCE (VERSION 2) | 12 |
| GPL 3 (DISCUSSION DRAFT 2 OF VERSION 3, 27 JULY 2006) | 13 |
| 3.7. THE EUPL COMPATIBILITY CLAUSE AND ITS EFFECTS | 14 |
| <u>4. THE “COMPATIBILITY” LIST</u> | <u>17</u> |
| 4.1. CRITERIA OF SELECTION | 17 |
| 4.2. SELECTION OF THE LICENCES FOR THE FIRST VERSION OF THE LIST | 19 |
| <u>5. PROPOSAL OF A FIRST LIST</u> | <u>21</u> |
| <u>6. SELECTED BIBLIOGRAPHY</u> | <u>21</u> |

1. Executive Summary

The European Commission (DG “Enterprise and Industry”, responsible for the implementation of the IDABC programme) has elaborated a Free/Open Source Software licence called EUPL v.1.0. This licence was principally drafted to apply to software of the European Community and, most probably, to software of other (European or national) administrations as well. The use of this licence by other users is also likely to occur.

Taking under consideration this particular context, the compatibility problems that the EUPL faces, like other copyleft licences, have been addressed by the addition of a compatibility clause. This clause provides for the possibility to relicense a work based upon the EUPLed software under other copyleft licences that are selected according to certain criteria and listed in a “Compatible Licences” appendix.

This compatibility clause allows a certain flexibility as regards software development, as it broadens the possibilities to merge code under the EUPL with other Free/Open Source Software. However, this implies that the EUPL will, in these cases, step aside for the application of these other licences to the further version of the software.

The compatibility clause has been revised in order to improve its effectiveness as regards the EUPL’s objectives and to avoid conflicts between the different obligations of the “migrant” (namely the person who is, at the same time, licensee under EUPL and licensor under the compatible licence). The proposed wording is the following:

“ If the Licensee Distributes and/or Communicates Derivative Works or copies thereof based upon both the Original Work and another work licensed under a Compatible Licence, this Distribution and/or Communication can be done under the terms of this Compatible Licence. For the sake of this clause, “Compatible Licence” refers to the licences listed in the appendix attached to this Licence. Should the Licensee’s obligations under the Compatible Licence conflict with his/her obligations under the present Licence, the obligations of the Compatible Licence shall prevail. ”

Selection criteria have been proposed in order to select a first list of licences to which the EUPL is made compatible by way of the compatibility clause:

1. the licence should be recognized as a FOSS licence (by the acceptance as such by either the FSF or the OSI);
2. the licence must be strongly copyleft (at least as regards the source code); **and**
3. the licence must be of practical use, that is
 - a large number of software rely on it, **or**
 - it governs **either**
 - at least one major software having a large number of user in the field where it applies, **either**
 - a project developed inside a public administration of a Member State of the European Community, **or**
 - a project partially or totally funded by the European Community or one of its Member States.

Other criteria have been added in order to update the list: a licence should be added in the list of compatible licences if

1. it meets the three abovementioned criteria; **and**
2. either a public administration of a Member State of the European Community or developers partially or totally funded by the European Community or one of its Member States make use of this licence (as licensee or as licensor) for existing code and express the willingness **either**
 - to include this code, or part of it, with or without modification, inside EUPLed code, **or**
 - to include EUPLed, with or without modification, code inside the code under this licence.

By “existing code”, we mean that the code should exist before considering its combination, possibly in a modified form, with EUPLed code, in order to avoid unnecessary extensions of the list.

A first proposal of compatible licences (chosen according to the first selection criteria) gathers the following licences:

- General Public License (GPL) v. 2
- Open Software License (OSL) v. 2.1, v. 3.0
- Common Public License v. 1.0
- Eclipse Public License v. 1.0
- Cecill v. 2.0

2. Introduction and Background

The European Commission (DG “Enterprise and Industry”, responsible for the implementation of the IDABC programme) has initiated discussion on the creation of an Open Source Software Licence. In this context, a draft “European Union Public Licence” (EURL) has previously been elaborated¹. The copyrights to this licence belong to the European Community.

The EURL licence has been created for the need of distribution of software developed under the IDA and IDABC programmes (such as CIRCA, IPM, eLINK,...). However, it would not be excluded that the EURL licence could be taken under consideration for the distribution, as the case may be, of other software.

Following a public consultation, it became evident that the EURL licence does correspond to the needs of a significant group of potential IDA/IDABC applications users, developers or improvers. This group would mainly be established inside the public sector and would be looking for a legal instrument that could be accepted in their own language by their competent authorities in case they would improve and redistribute IDA/IDABC software applications to their members. This could also be the case of the European Commission itself as well as of other European Institutions or bodies as far as the IDA/IDABC software applications are concerned. Nevertheless, the use of the EURL licence would not be, in any case, mandatory.

The EURL licence is a “copyleft” licence and imposes the same terms in case of redistribution. The aim of the “copyleft” effect introduced in the EURL licence is to avoid appropriation of the licensed work by a third party who could sell it as a proprietary work, so avoiding at the same time the situation where the European Commission could be in a position to purchase under the terms of a proprietary licence a new version of a IDA/IDABC software originally released by the European Commission under the EURL licence.

The main drawback of “copyleft” licenses is that they are generally not mutually compatible. Nevertheless, the EURL licence intends to deal with this issue by way of a compatibility clause allowing, in certain circumstances, the redistribution of derived works under other licences that are chosen and gathered in an exhaustive list attached to the EURL.

The question of compatibility of licences has already been examined in two reports requested by the European Commission. The first one – on Open Source Licensing of software developed by the European Commission – (hereunder referred to as “the first report”)² dealt with the issue in general terms. The second one – on the outcomes of the public consultation about the EURL licence – (hereunder referred to as “the second report”)³ focused more particularly on the EURL licence.

¹ The current draft version of the EURL is v.1.0 of 17 May 2006.

² See Advice Report of 16 December 2004 on Open Source Licensing of software developed by the European Commission (applied to the CIRCA solution), prepared on the request of the European Commission by the CRID and UNISYS with the participation of Séverine Dusollier (CRID-FUNDP Namur), Philippe Laurent (CRID-FUNDP Namur) and Patrice-Emmanuel Schmitz (Unisys Belgium), <http://ec.europa.eu/idabc/en/document/5425/5585#eurl>,

³ See Advice Report of 30th November 2005 on the outcomes of the public consultation about the EURL licence, prepared by UNISYS with the participation of Patrice-Emmanuel Schmitz (Unisys Belgium), Rishab Aiyer Ghosh (Merit), Séverine Dusollier (CRID-FUNDP Namur), Philippe Laurent (CRID-FUNDP Namur) and Jean-Paul Triaille (Dewolf & Partners-Lawyer) - <http://ec.europa.eu/idabc/en/document/5425/5585#eurl>

The aim of this report is to further the analysis of the compatibility issue and its handling in the frame of the EUPL by way of a compatibility clause, and to provide a first selection of licences (chosen according to specific and motivated criteria) to be listed in the EUPL compatibility appendix.

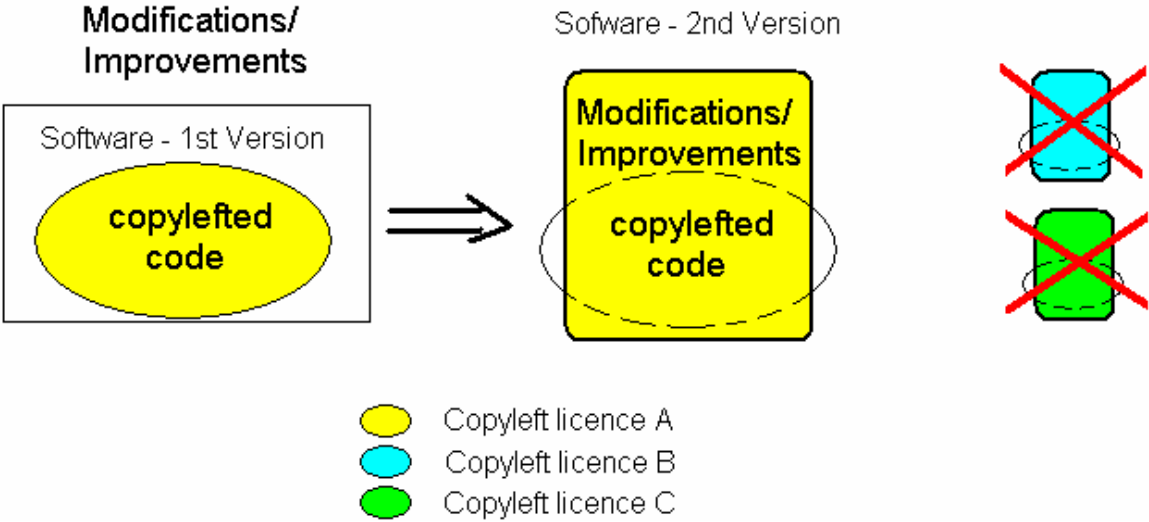
3. The “compatibility mechanism”

3.1. The compatibility issue as a drawback of the copyleft effect

The compatibility issue could be depicted as being a drawback of the “copyleft effect”.

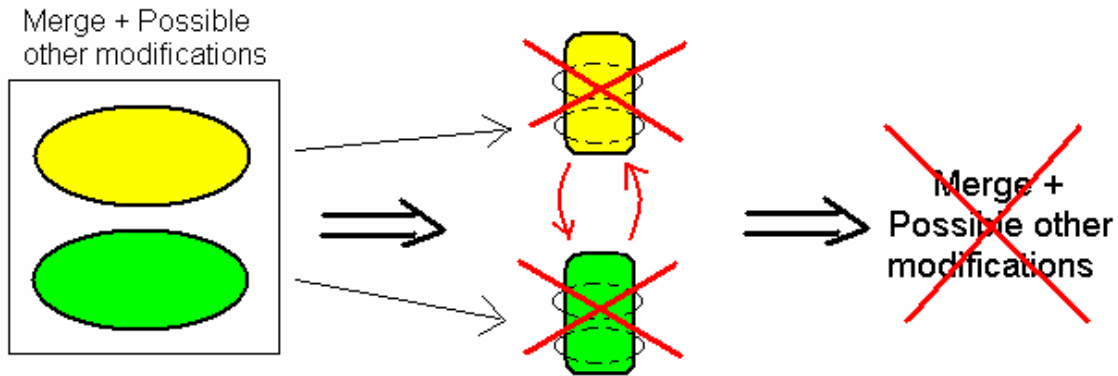
Copyleft licences are Free/Open Source Software (hereunder FOSS) licences allowing the use, modification and distribution of software on the condition that its modifications (the improved software) are distributed under the same terms and conditions⁴. The **aim** of the copyleft scheme is to ensure that a FOSS and its derivatives remain free/open source: the **result** is that the project usually remains under the same licence. One must stress the difference between the objective and the result: whereas the aim does not necessarily impede flexibility as regards the choice of FOSS licence, the result does usually so.

Practically speaking, it means that the licensee has usually no choice as regards the licensing of his derivative work: it must be distributed under the same licence.



Another important consequence is that the licensee may not be allowed to mix different codes licensed under different copyleft licences. Indeed each one of these latter will usually impose the redistribution of the result only under itself.

⁴ For a deeper analysis of the copyleft effect, and the difference between copyleft and non-copyleft licences, please refer to the first and second reports.



The question of merging or combining code together is not always clear, since code can be combined mainly:

- by the integration of one code in another code in the same file,
- by separating them in different files but generating a single executable on basis of these files, or ;
- by linking (separately) compiled codes at the execution time (such an operation being called dynamic linkage).

The extent of the copyleft effect is dependent on the licence. Some licences restrict the effect on a file basis (this is the case of the Mozilla Public License - hereunder “MPL”- and the Common Development and Distribution License - hereunder “CDDL” - for instance). In that case, we will speak of “**weak copyleft**”. Some licences impose the redistribution of the source code under the same licence, while the executable code may be governed by another licence (as long as the source code is available and remains under the original licence): the executable version of the derivative work can then be proprietary (this is the case of the MPL, the CDDL, and the Eclipse Public License - hereunder “EPL”). In that case, we will refer to “**source only copyleft**” licences. Most licences do not restrict the copyleft effect to modifications on a file basis, and the executable is to be considered as a derivative work. In that case, we will speak of “**strong copyleft**”. Finally, the question of dynamic linkage is trickier, and may depend on the applicable national legislation and/or the interpretation of the latter.⁵

In terms of compatibility, one says that copyleft licences are usually not compatible amongst themselves. Code under licence A usually cannot be licensed under licence B, nor being merged with code under licence B.

The compatibility issue is amplified by the fact that the number of different copyleft licences, applying to different software, has greatly increased during the last decade: the more copyleft licences exist, the more compatibility problems are likely to occur, and as a result, the more open source software development is hindered or, at least, slowed.

One of the current aims of FOSS licences drafters is therefore to ensure a better compatibility amongst the FOSS licenses, in order to ascertain that, on the one hand, FOSS remain

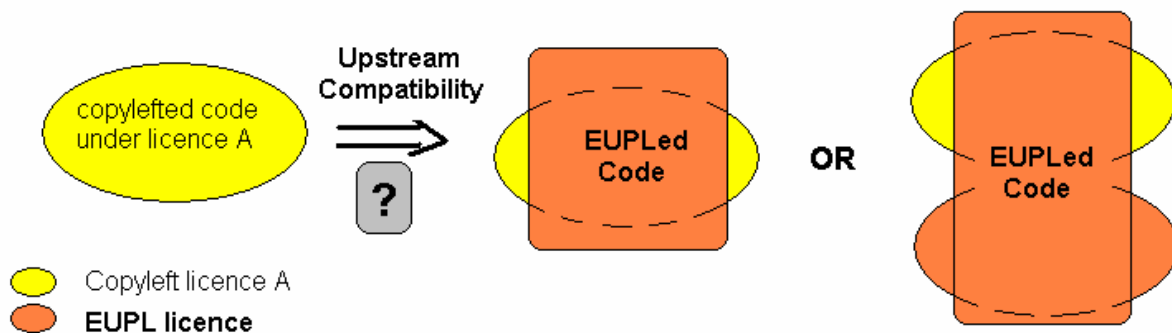
⁵ Determining what precisely is a « derivative » undergoing the licences copyleft effects is a tricky issue that cannot be discussed in the present report. Please refer to the first and second reports, as well as to their bibliographies. **In this report, we will start from the presupposition that the copyleft effect is operating.** In the further analysis about the selection of licences, we will therefore propose to qualify or discard a licence according to its copyleft effect, independently of what is provided and/or interpreted about linkage.

free/open source and that, on the other hand, the code to which they apply can be combined with each other. The FOSS community is also asking to address the licences proliferation issue by avoiding the creation of new licences if any of the existing ones already meets the specific objectives of the licensor.

3.2. Upstream and downstream compatibility issues

When creating a new licence, namely the EUPL, the compatibility issue has to be addressed from two points of views: the upstream, and the downstream perspectives.

In an **upstream perspective**, the question to be assessed is whether code distributed under other FOSS licences may be licensed under EUPL, or incorporated in projects licensed under EUPL (merged with code under EUPL).



This “upstream compatibility” depends mostly on the terms and conditions of the upstream licence (licence A). If this latter is a copyleft licence, it is most likely that licence A allows only the redistribution under the same licence. In this case, nothing in the drafting of the EUPL will change the fact that the code under licence A may not be licensed under EUPL or merged with EUPLed code⁶ and relicensed under EUPL. A few copyleft licences however encompass compatibility clauses allowing either (1) to “change” or “custom” certain terms and conditions of the licence, or (2) to license the modified software under one or several other licence(s)⁷. While to date this practice is more the exception than the standard case, it receives an increasing attention due to the exposed compatibility issue.

In the first situation, the drafting of the EUPL could influence the upstream compatibility: this drafting should be the same as the drafting of licence A and provide for the modalities that are allowed in the compatibility clause (if provided). In other words, it should be possible to imagine adapting the drafting of the EUPL to the compatibility conditions of the upstream licence.

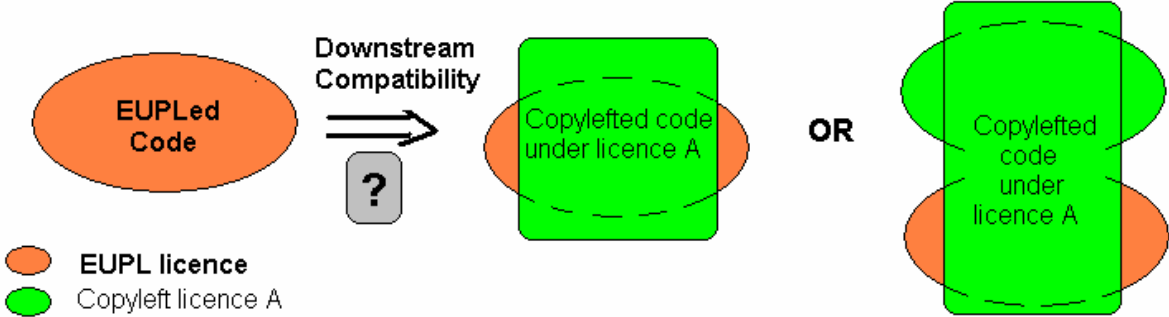
Given the actual drafting of the EUPL, its background and the aims it has to fulfil, it is unlikely that the EUPL will ever benefit from this kind of “upstream compatibility”.

⁶ Except sometimes by dynamic linkage, the library remaining covered itself by its original licence.

⁷ Cfr. *Infra*.

In the second case, the presence of the EUPL amongst the elected licences would ensure compatibility. In this respect, it should be up to the Community to contact the drafters of such copyleft licence in order to insure that the EUPL appears amongst the elected licences.

In a **downstream perspective**, the question to be assessed is whether code distributed under EUPL may be licensed under other licences, or incorporated in projects licensed under other licences (merged with code licensed under other licences).



On the contrary to the “upstream compatibility” issue, this “downstream compatibility” depends mostly on the terms and conditions of the EUPL. Indeed, as in any other copyleft licence, the copyleft effect of the EUPL licence is obtained by the contractual clauses embedded in the licence. Therefore, one who drafts the EUPL masters its copyleft effect and may attenuate, complete or “reshape” the latter. This is the aim of the compatibility clause in article 5 of the current EUPL version.

In summary, the compatibility clause in the EUPL does not address the “upstream compatibility” issue, but only the “downstream compatibility” issue, so we will hereafter use the term “compatibility” as synonym of “downstream compatibility”, unless otherwise stated.

3.3. Terminology

The term “Compatibility” means the possibility either to create software on basis of a combination of software distributed under different licences, or, when open source code is used in software, to distribute this latter under another licence.

One must pay attention to the fact that compatibility is a tricky word as regards the relations between open source licences since, in that context, **it does not encompass reciprocity**. We have noticed here above that one could distinguish a “downstream compatibility” and an “upstream compatibility”: in order to assess the compatibility of a licence, one must respect the way of the “stream”.

The compatibility of licence A towards licence B means that code under licence A may be licensed under licence B or merged with code under licence B to create a whole licensed under licence B. This effect depends on the copyleft effect (the wording of the copyleft and/or compatibility clauses) of licence A.

The fact that licence A is compatible with licence B does not entail that the opposite is true. In order to assess the compatibility of licence B towards licence A, one must analyse the copyleft and/or compatibility clauses of licence B.

3.4. Stakes of the of the EUPL's compatibility

CIRCA, IPM or eLINK are Community's software that will be released under the EUPL. One of the aims of this source opening is allowing national administrations to build upon this software, to develop it according to their needs and specifications and to grant all the users the benefit of those improvements. This adaptation and development of the Commission's software is likely to happen through mergers between this latter and the national administrations' own software, and/or with other FOSS as well. This possibility to merge CIRCA, IPM and eLINK with other FOSS code depends on the compatible character of the EUPL.

New FOSS licences in general, and new copyleft licences in particular, are usually not welcomed by the open source community, especially given the actual compatibility problems. However the second report has clearly shown that the EUPL would fill the needs of significant groups of users, mainly established inside the public sector. In order to reconcile these positions, it is important to create a licence that meets the expectations of the targeted users group, while being welcomed by the general open source community. Providing for a compatibility clause seems the more practical and effective way to strike this balance.

Nevertheless, given the non-reciprocal character of compatibility, one must not forget the implications of creating a "compatible licence". Whether its positive aspects are to grant flexibility in software development, and allow the merging of codes from different origins, its negative aspects entail a loss of control on the downstream developments. Indeed, insuring compatibility equals to accepting the fact that the project "switches" from the EUPL to another licence (to which the EUPL has been made compatible), which is itself potentially incompatible with the EUPL. In that sense, including a compatibility clause is an act of denial by which its drafter bears the risks that the licence could actually not apply to a further version of the software. As a consequence, such a compatibility clause also raises the issue of **forking**, which means that the same project could lead to creation of various development branches, the interactions between the latter being restricted as long as contributions are made on a copyleft basis. While forking would promote the diffusion of previously EUPLed code and the improvements of the initial project, various (but not all) actors inside the FOSS community consider this as a bad practice.

3.5. Alternative solutions to the compatibility issue

Confronted to the compatibility issue, one could find several solutions in order to try and avoid its drawbacks.

Non-copyleft scheme

The first solution in order to avoid the problem would be to choose or create a non-copyleft licence. The major drawback is obviously the fact that the aims of the copyleft effects are not reached in any respect. The code can be, in particular, used inside a proprietary derivative product, putting away from the initial licensor the possibility to freely benefit from code improvements (a famous example is the SSH software, whose version 2.0 is proprietary, and whose version 1.2.12 gave birth to a popular open-source fork, called OpenSSH).

Dual-licensing

Another solution could be seen in the “dual-licensing” practice. In order to facilitate merging and diffusion of the code, one may license the same code under two or more licences, at least one of them being copyleft. The licensee may therefore use the licence that he prefers, or which is imposed by the circumstances. If only copyleft licences are used, this solution implies that the licensor has no particular objectives except to make sure that the software will remain Free/Open source, no matter which licence will finally prevail (this particular regime governs, for instance, parts of the Linux kernel, with the dual-licence GPL/OSL).

This solution does not however completely address the compatibility problem, as the different licences that will be offered to the choice of the licensee are not likely to be compatible amongst themselves. One could therefore blame this type of dual-licensing practice for the creation of a fork at the mere origin of the licensing scheme.

When it implies the use of pre-existing and third parties’ licences, this practice does not usually allow any control neither on the licence nor on the future versions of the project, except by the addition of separated copyright assignment clauses. These clauses are not *per se* part of the licence, but are a common practice in large projects, where one or a few official maintainers is responsible for the acceptance or rejection of contributions to the official and promoted software version, or for the election of official developers. The FSF for instance requires that contributors to official FSF projects (most – not all – of the GNU projects) grant a copyright assignment⁸. The same is true for OpenOffice.org, since contributors are required to sign a Joint Copyright Assignment if they want their contributions to be incorporated in official releases.

The use of dual-licensing is also a common practice to allow software exploitation in both FOSS and commercial circuits, a practice sometimes criticized by the FOSS community. A famous example is the Berkeley DB, developed by Sleepycat Software Inc., a company bought by Oracle in 2005. The policy is here clear: “*If you do not want to release the source code for your application, you may purchase a license from Sleepycat Software*”⁹. Välimäki points that for such a commercial purpose, “*legally, dual licensing requires the use of a license with a strong copyleft clause and that all legal rights to use and distribute the software are managed by a single entity*”¹⁰. Note that other licences encourage commercial exploitation without making use of the dual-licensing model. In particular the Common Public License and Eclipse Public License have a special clause allowing the use of a commercial/proprietary license when the program is distributed in object code. These licences remain however copyleft by requiring the source code, governed by the original licence, to be obtainable at reasonable cost and manner.

⁸ <http://www.gnu.org/licenses/why-assign.html>. The European FSF branch in particular invites European contributors to fill the Fiduciary Licence Agreement (<http://www.france.fsfeurope.org/projects/fla/>).

⁹ <http://www.sleepycat.com/company/licensing.html>

¹⁰ VÄLIMÄKI, M., “*Dual Licensing in Open Source Software Industry*”, *Systèmes d’Information et Management* 8(1), 2003, pp. 63-75.

Inclusion of a compatibility clause

The creation of a new copyleft licence, which is made compatible by way of a compatibility clause, allows the drafter to remain in control of the licence, and to adapt the licence to its project and the circumstances of the project¹¹. In that respect, the application of the compatibility clause (entailing the licensing of a further version of the software under (an)other licence(s)) should only happen when the circumstances require it: most of the time, this switch of licence is imposed by the merging of the software with a code already licensed under another FOSS licence.

One could imagine a multitude of compatibility clauses, as these clauses are only the result of the autonomy of the will of the parties: the licensor is free to draft the licence as he likes¹² (and, amongst others, to provide whatever the compatibility clause he wants) and the licensee is free to agree with those terms.

3.6. Compatibility clauses: a comparative analysis

In order to assess the compatibility clause of the EURL, one could first analyse the compatibility clauses of the Cecill licence (version 2) and the new draft of the GPL 3 (discussion draft 2 of version 3, 27 July 2006¹³)

Cecill licence (version 2)

In terms of compatibility, the main novelty of the Cecill licence¹⁴ is the introduction of a clause allowing the distribution under the GPL instead of the Cecill. Its article 5.3.4 reads as follows:

“5.3.4. COMPATIBILITY WITH THE GPL LICENSE

In the event that the Modified or unmodified Software is included in a code that is subject to the provisions of the GPL License, the Licensee is authorized to redistribute the whole under the GPL License.

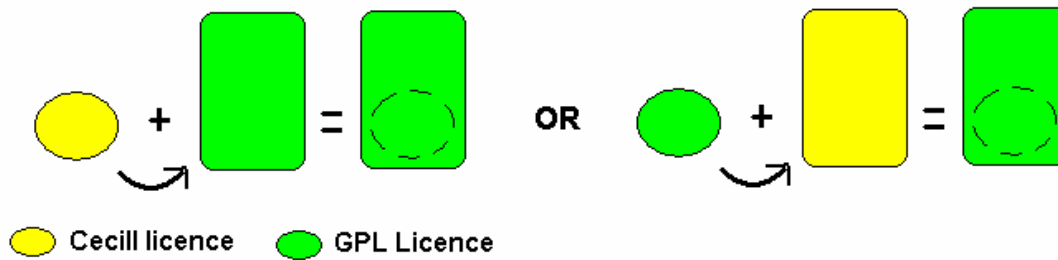
In the event that the Modified Software includes a code that is subject to the provisions of the GPL License, the Licensee is authorized to redistribute the Modified Software under the GPL License.”

¹¹ As regards the issue of the « choice » of an open source licence, please refer to the first report.

¹² This is a general principle of contract law: we do not address mandatory law in this report.

¹³ See <http://gplv3.fsf.org/gpl-draft-2006-07-27.html>. This licence was not a released version at the drafting moment of this report.

¹⁴ Two other licences, called Cecill-B and Cecill-C have been written to more or less mimic the BSD licence and LGPL, while being drafted under the French legislation. The Cecill-B licence not being copyleft, and the Cecill-C addressing modules and dynamically linked code, they will not be considered in this report and the use of the “Cecill” term will refer to the first, strongly copylefted, form.



Such a practice has been approved by the Free Software Foundation (FSF), which officially recognises the Cecill as GPL-compatible. The main goal of the Cecill was indeed to deliver an open-source GPL-compatible licence, which is drafted under the French legislation, and available in French as well as English for internationalisation reasons.

The compatibility clause of the Cecill is restrictive in the sense the licence, in its current version, does only provide for compatibility with the GPL. Situations however may arise in which software used by European public administration could be covered by possible other various licences.

GPL 3 (discussion draft 2 of version 3, 27 July 2006)

This current version of the discussion draft of the GPL3 encompasses a quite complex system aiming at making the GPL more compatible.

Its Section 5, b reads as follows:

*“You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License must apply, **unmodified except as permitted by section 7 below**, to the whole of the work, and all its parts, regardless of how they are packaged.”*

The copyleft clauses impose the redistribution of the work under the GPL unmodified except when the modifications are allowed by Section 7.

Section 7 (whose title was “Compatibility Clause” in version 1 and replaced by “Additional Terms” in version 2) provides that the **original author** may add clauses, which grant “additional permissions” or, on the contrary, have a narrowing effect, by adding “additional requirements”.

According to section 7, §a, the additional permissions may make exceptions to one or more requirements of the GPL. A totally different licence encompassing a compatibility clause allowing the relicensing under the GPL without any other surviving requirement is to be treated as a list of additional permissions. Accordingly, the new LGPL will be considered as a GPL with additional permissions as regards the linkage with libraries¹⁵.

Only several determined types of additional requirements (listed in section 7, §b) may be added.

¹⁵ See <http://gplv3.fsf.org/lgpl-draft-2006-07-27.html>

The **licensee** (and/or licensor of modified versions) may remove any additional permission (even covering code not added by him/her). However, he cannot remove any additional requirement authorized in section 7, §b.

The licensee, when conveying the work (hence being now himself licensor) may only place additional permissions, or additional requirements as allowed by subsection 7b, *on material he added* to a GPLed work, for which he has or can give appropriate copyright permission.

One must stress that this system addresses mainly the **upstream** compatibility issue. The “additional permissions” perspective plays like a funnel, which selects code that may, at the end of the day, be licensed under the GPL 3 in its strict sense. The “additional requirements” perspective is more to be analysed as a flexible copyleft-licensing scheme allowing the addition of determined clauses to a basic and un-modifiable trunk. In other words, the GPL 3 is made “customisable” to a certain extent.

However, the fact that the licensee may place additional permissions or requirements on material he added allows certain flexibility when combining a GPLed code with code under some other copyleft licences, which may be welcomed on a downstream perspective. In that case, the modified code will be submitted to two different licences, depending on the parts of the code (material) that is considered. For instance, one could imagine combining GPL code with “material” under the Affero General Public Licence (hereunder “AGPL”), as the only difference between GPL v.2 and the AGPL is an additional requirement which is allowed under section 7, §b, n°4.

3.7. The EUPL compatibility clause and its effects

Article 5 of the EUPL licence introduces a “compatibility” clause in accordance to which:

“If the Licensee Distributes and/or Communicates copies of the Original Works or Derivative Works based upon both the Original Work and another work licensed under a Compatible Licence, this Distribution and/or Communication can be done under the terms of this Compatible Licence. For the sake of this clause, “Compatible Licence” refers to the licences listed in the appendix attached to this Licence.”

One must insist on the fact that the terms “Compatible Licence” are given a particular (and contractual) meaning in the frame of the EUPL licence. Indeed, these Compatible Licences are actually licences **to which the EUPL is compatible** (see section 2.3). In other words, this clause addresses the **downstream** compatibility of the EUPL, resulting in the relicensing of the derivative work under another licence. On the contrary, creating an “upstream” list of “compatible licence” (which would step aside for the EUPL in case of combination of code) would not be possible and would almost certainly conflict with the terms and conditions of those licences (unless those listed licences would actually allow such compatibility).

The purpose of this article was to allow the combination between EUPLed code and code licensed under certain other FOSS licences (which requires the redistribution of the result under itself – namely **copyleft licences**). In other words, this article was aiming at permitting

some combinations that would otherwise not be possible without this provision. In that situation, the EUPL would step aside for the elected “compatible licences”.¹⁶

The issue of the election of the listed licences is addressed at section 3 of the present report.

The current wording of the clause does not seem to be perfectly clear and its wanted effects seem not to be achieved. Revising it should therefore be considered.

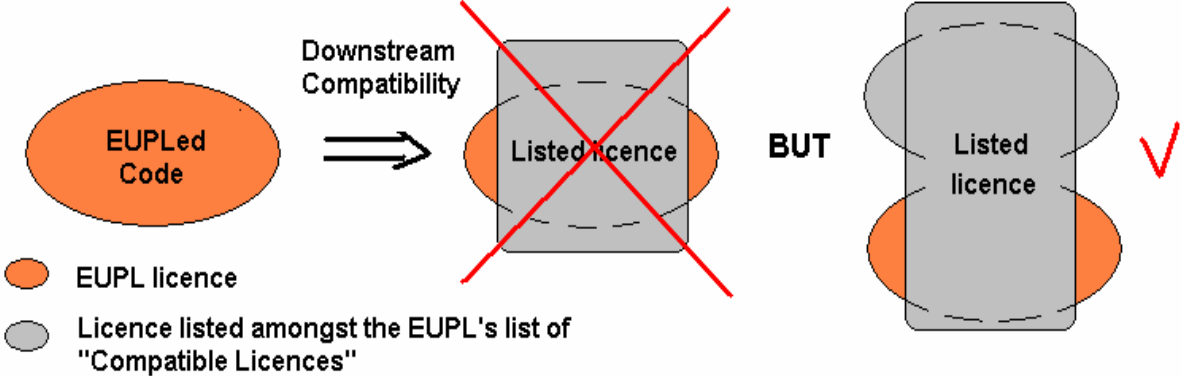
“Original Work” is defined in article 1 as “*the software distributed and/or communicated by the Licensor under this Licence, [...]*”. The Licensee could therefore not possibly Distribute and/or Communicate copies of the Original Works based upon the Original Work and another work licensed *under a compatible licence* (as this interpretation would conflict with the definition of “Original Work”) – unless one considers that the Licensee becomes Licensor under the EUPL, which is not the case as he will precisely be licensor *under the compatible licence*.

One must therefore read the clause as permitting the Licensee to Distribute and/or communicate copies of the Original Work under the terms of “this” Compatible License (but then, which Compatible License is it? The use of the demonstrative adjective seems awkward). Furthermore, this effect would however not be advisable, as it would allow a switch to a Compatible License without necessity.

We would therefore advise the following wording:

“ If the Licensee Distributes and/or Communicates Derivative Works or copies thereof based upon both the Original Work and another work licensed under a Compatible Licence, this Distribution and/or Communication can be done under the terms of this Compatible Licence. For the sake of this clause, “Compatible Licence” refers to the licences listed in the appendix attached to this Licence.”

This effect could be schematised as follows:



When the compatibility clause of article 5 is applied and the Distribution and/or Communication of a derivative work are “done under the terms of a Compatible Licence”:

¹⁶ On this regards, and about the precise aim of the clause, please refer to pp 13-15 of the second report, and more precisely to the downstream compatibility figure as provided at p. 15.

- the **Licensor** remains bound towards the Licensee by all its obligations under the EUPL (and amongst others, by art 2 and 3);

- the **Licensee** is still bound by the licence (as the party who has contracted with the Licensor). On this regard, the “Licensee” is still bound by his obligations as

- “Licensee” (“Licensee” or “You”¹⁷ – see for instance art.5, 9, 10 or 11);
- “Party” (see for instance art. 13); and even as
- “Contributor”¹⁸ (see for instance art. 6).

However, when Distributing and/or Communicating the work under a Compatible Licence (in application of article 5) the Licensee will become licensor solely under this Compatible Licence, and not under the EUPL licence¹⁹.

Therefore, we have to stress the specificity of this situation in which one (hereunder referred to as the “Migrant”) remains licensee under the EUPL and becomes, at the same time, licensor under a Compatible Licence (in other words, when the Licensee makes application of the compatibility clause in article 5 of the EUPL).

Firstly, situations may arise in which the Migrant’s acts will be restricted by the fact he should respect the application of both licences on a particular point. For example, according to the “provision of source code” clause of the EUPL, “*when Distributing and/or Communicating copies of the Work, the Licensee will provide a machine-readable copy of the Source Code or indicates a repository where this Source will be easily and freely available for as long as the Licensee continues to Distribute and/or Communicate the Work*”. However, some copyleft licences provide for (normally slight) different obligations as regards the availability of the source code. The Migrant should therefore pay attention to comply with obligations under both licences.

Some contractual conflicts could also arise from this situation: one could indeed imagine that the obligations of the Migrant as licensee under EUPL conflicts with his obligation as licensor under the Compatible License (even after the compatibility clause has been applied). For example, according to the “Attribution right” clause of the EUPL, the Migrant has to “*keep intact all copyright, patent or trademarks notices and all notices that refer to the Licence and to the disclaimer of warranties*”. However, when relicensing his derivative work under another licence, the Migrant should actually not deceive the licensee by keeping intact the notice referring to the licensing under EUPL. One could moreover wonder which disclaimer of warranties has to be placed on the software, the one of the EUPL or the one of the Compatible Licence, or both...

Therefore, in order to prevent those situations where the Migrant should refrain from using the compatibility clause because of conflicting obligations as licensee under EUPL and

¹⁷ Article 1 of the EUPL defines the Licensee or “You” as “*any physical or legal person who makes any usage of the Software under the terms of the Licence*”.

¹⁸ Article 1 of the EUPL defines the Contributor as “*any physical or legal person who modifies the Work under the licence, or otherwise contributes to the creation of a Derivative Work*”.

¹⁹ Article 1 of the EUPL defines the Licensor as “*the physical or legal person that distributes and/or communicates the Work under the [EUPL] Licence*”.

licensor under the compatible licence, we would advise to complement the compatibility clause, with a sentence such as:

“Should the Licensee’s obligations under the Compatible Licence conflict with his/her obligations under the present Licence, the obligations of the Compatible Licence shall prevail.”

4. The “compatibility” list

The “compatibility” clause of the EUPL licence addresses the “downstream” aspect of the above-mentioned compatibility issue. This clause permits to consider the EUPL as “compatible” to other licences, which are gathered in a list that is attached to the EUPL in the form of an appendix.

The aim of this section is to determine selection criteria and to apply them in order to come out with a first list of licences as well as with guidelines to the updating of this list.

4.1. Criteria of selection

As underlined in the introduction as well as in the first and second report, the EUPL has predominantly been created for the need of distribution of software developed under the IDA and IDABC programmes (such as CIRCA, IPM, eLINK,...).

The significant group of potential users is mainly established in the public sector: the main users of the programmes and the EUPL licence will be the European public administrations.

4.1.1. Copyleft as a general criterion

Given the criteria that have been withheld hereunder, a distinction must be made between the selection of licences for the first version of the list and the updating of the list.

One criterion may however be discussed at this point. As already stressed, the purpose of the clause is to allow mergers between EUPLed code and other copylefted code. Indeed, compatibility problems mainly occur from the conflict between the copyleft effects of two different copylefted licences. The first selection criterion that should always apply is therefore that the chosen licences must in any case be copyleft.

Furthermore, the copyleft effect of the elected licences should be similar to the EUPL’s copyleft and should fulfil the same functions. One of the considerations that were taken into account when deciding to choose for a copyleft licence was to ensure that improved (derivative) source code may be reused by the administrations. Accordingly, the objective is to keep open as much source code as possible. The EUPL’s copyleft effect is therefore not restricted on a “file” basis (cfr. Supra: the elected compatible licences should be strong copylefted - at least, with respect to the source code). On the other hand, more freedom could be left as regards the executable code. The elected licences could therefore be “source only copylefted”, in the sense given in Section 2.1.

4.1.2. First version of the list

The number of copylefted licences is quite large, but some of them are either obsolete, being replaced by newer licences, or of no practical use in the FOSS community. Intel Company itself, for instance, has disclaimed the Intel Public License, while the IBM Public License has been superseded by the Common Public License and the Eclipse Public License. Only currently and actually used and promoted licences should be considered.

More generally, it seems reasonable to only select a subgroup of existing licences on the basis of their importance and popularity. Some committees have been created to study the problem of FOSS licences proliferation. One of such committees has been appointed inside the OSI, and gathers internationally renowned specialists²⁰. This committee has also established a subgroup of most important and popular FOSS licences, which will be withheld as a basis for the selection of the listed licences.

Importance and popularity nevertheless are probably impossible to determine on a purely academic basis, so the proposed criteria mainly rely on the perception of what should be respected in order to correspond to the initial EUPL objectives. This perception could be guided, either by subjective opinions or by the pursued objectives, which are quite specific in the case of the EUPL.

The proposed criteria for acceptance of a licence could be in this last case:

1. the licence must be recognized as a FOSS licence, a point that we formally characterised by the acceptance as such by either the FSF or the OSI;
2. the licence must be strongly copyleft (at least as regards the source code); **and**
3. the licence must be of practical use, that is
 - a large number of software rely on it, **or**
 - it governs **either**
 - at least one major software having a large number of user in the field where it applies, **either**
 - a project developed inside a public administration of a Member State of the European Community, **or**
 - a project partially or totally funded by the European Community or one of its Member States.

Given the reluctance of the open community toward proliferation of FOSS licences, and – for some of their members – towards the forking problems (cfr. supra), the list of compatible licences should be as small as possible. This consideration should be kept in mind when applying the selection criteria.

Various licences moreover offer the licensor the possibility to authorize the distribution and communication under a further version of licence, possibly not yet defined. Since the additional permissions and/or requirements are not known until the new versions are published, and since they could violate some purposes of the EUPL, we advice to only include in the list specific known versions of the licences.²¹

²⁰ <http://www.opensource.org/docs/policy/licenseproliferation.php>

²¹ Such a concern has been previously raised by important actors in the computing science community, in particular Sun, which has written the Common Development and Distribution License on the basis on the

4.1.3. Further versions of the list

In order to respond to the proliferation issue, as already written, the first list should be kept as small as possible, but this implies that a public administration could use a licence not present in the list. Since the main goal of the EUPL is to promote the diffusion of FOSS software between European public administrations, we suggest to add a licence in the list of compatible licences if

1. it meets the three criteria proposed in Section 3.1.2; **and**
2. either a public administration of a Member State of the European Community or developers partially or totally funded by the European Community or one of its Member States make use of this licence (as licensee or as licensor) for existing code and express the willingness **either**
 - to include this code, or part of it, with or without modification, inside EUPLed code, **or**
 - to include EUPLed, with or without modification, code inside the code under this licence.

By “existing code”, we mean that the code should exist before considering its combination, possibly in a modified form, with EUPLed code, in order to avoid unnecessary extensions of the list.

The European Commission, as copyright holder of the EUPL text, is the only organism allowed to modify the list of compatible licences. Once a licence has been added to the list, it could not be removed from it²².

4.2. Selection of the licences for the first version of the list

As previously stated, the OSI has chartered a committee to address the issue of licences proliferation. This committee has recently drafted a first classification of licences in order to give a practical answer to the licence proliferation issue, by classifying recommended licences and those whose usage should be avoided²³. The list of recommended licences, that were selected because these licences are popular and widely used or support projects with strong communities of users, is (we have written in italic licences that present a strong copyleft. Weak copyleft should not be desirable for the purposes of the Community since it allows, to a certain extent, the creation of proprietary derivative works):

- Apache License, 2.0²⁴
- New BSD license²⁵
- *GNU General Public License (GPL)*²⁶
- *GNU Library or "Lesser" General Public License (LGPL)*²⁷

Mozilla Public License. The Description and Rationale document indicates that it fixed the “Effect of New Versions problem” (http://www.sun.com/cddl/CDDL_why_details.html).

²² See second report, p. 11.

²³ <http://crynwr.com/cgi-bin/ezmlm-cgi?10:mmp:74>

²⁴ <http://www.apache.org/licenses/LICENSE-2.0>

²⁵ <http://www.opensource.org/licenses/bsd-license.php>

²⁶ <http://www.gnu.org/licenses/gpl.html>

²⁷ <http://www.gnu.org/licenses/lgpl.html>

- MIT license²⁸
- Mozilla Public License 1.1 (MPL)²⁹
- Common Development and Distribution License³⁰
- *Common Public License*³¹
- *Eclipse Public License*³²

The OSI also lists three special purposes licences:

- Educational Community License³³ (special purpose: only suitable for educational establishments)
- NASA³⁴ (special purpose: for use by an agency of the federal US government, which has special concerns regarding some issues such as copyright protection, copyright notices, disclaimer of warranty and indemnification, and choice of law)
- Open Group Test Suite³⁵ (special purpose: only suitable for tests or test suites)

These three licences seem too specific and addressing cases of no direct impact for the Administrations of the European Community. They will therefore not be considered in the rest of the study.

This list could represent an interesting starting point for discussion, but it gathers both copylefted and less restrictive licenses, so a selection has to be made on basis of the second criterion.

Moreover, since the considered users group are mainly European administrations, FOSS licences that have been developed or used inside European-funded projects or institutions of a Member State should also be considered. Therefore, we would add the Cecill to the list, even if it is currently ignored (but not rejected) by the OSI. Since this licence is recognised by the FSF, it meets criterion 1.

The LGPL should however not be withheld, since it corresponds to a GPL that allows linkage with code under incompatible licences and that specifically targets the coverage of libraries³⁶. As the LGPL explicitly allows linkage with any incompatible licence, including EUPL, there is no compatibility issue to solve as long as such a linkage is not interpreted as the production of derivative work in view of the EUPL.

Other lists of licences could be considered. In particular, some software is governed by the GPL with an additional FOSS exception clause for linkage: the combination with external modules governed by FOSS, but possibly GPL-incompatible licenses, is explicitly permitted. This is for instance the case with MySQL AB, a popular database management system, whose

²⁸ <http://www.opensource.org/licenses/mit-license.php>

²⁹ <http://www.mozilla.org/MPL/MPL-1.1.html>

³⁰ <http://www.sun.com/cddl/cddl.html>

³¹ <http://www.opensource.org/licenses/cpl1.0.php>

³² <http://www.eclipse.org/legal/epl-v10.html>

³³ <http://www.opensource.org/licenses/ec11.php>

³⁴ <http://www.opensource.org/licenses/nasa1.3.php>

³⁵ <http://www.opensource.org/licenses/opengroup.php>

³⁶ Libraries are defined in the LGPL as “a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables”.

GPLed version includes a FOSS exception list³⁷. Such a list primarily targets the licences of libraries, which are effectively linked with the software, and it usually exceeds the list of licences recommended by the OSI. This reflects that in practice the OSI list could be too short. The goal of the compatibility clause is not however to address linkage, which could be allowed by the EUPL even for incompatible licences (and not by the GPL, for instance), depending of the definition of derivative works withheld under the applicable law. Moreover, considering the aims of the compatibility clause and the fact that the use of the EUPL should be promoted in any situation that not strictly requires the application of this provision, the list should be kept as short as possible, and the introduction of new licences in the list should be avoided whenever feasible.

5. Proposal of a First list

Given the analysis carried on here above, a first proposal of a first compatibility list would be the copylefted OSI-recommended licences with addition of the Cecill licence v2.0.

The list should additionally always mention which version(s) of the licences is (are) currently considered as compatible with the EUPL.

According to the proposed criteria and the underlying considerations as depicted in this report, the first proposal of the compatibility list would therefore be the following:

Appendix

Compatibility list – EUPL V.1.0 – date xxx/xxx/2006

The following licences are “compatible licences” under article 5 of the EUPL:

- General Public License (GPL) v. 2
- Open Software License (OSL) v. 2.1, v. 3.0
- Common Public License v. 1.0
- Eclipse Public License v. 1.0
- Cecill v. 2.0

Previous versions of the Open Software License could possibly be included, but the latter are no longer recommended by its author, Lawrence Rosen, and are not widely used to our knowledge.

6. Selected bibliography

BEARDWOOD, J.P., ALLEYNE, A.C, “The price of Binary Freedom: The Challenge of Open Hybrid Software”, *Computer Law Review International*, 4/2006, pp. 97-104.

³⁷ <http://www.mysql.com/company/legal/licensing/foss-exception.html>

LAURENT, PH., “Logiciels libres et droit d’auteur: naissance, titularité et exercice des droits patrimoniaux”, in *Les logiciels libres face au droit*, cahier du C.R.I.D. n°25, Bruxelles, Bruylant, 2005, p. 74-75.

LAURENT, PH., “Un logiciel de la Communauté européenne en *open source* ? Le choix crucial d’une licence libre”, *Revue du Droit des Technologies de l’Information* 23, 2005, pp. 22-40.

RAYMOND, E.S., *The Jargon File* (version 4.4.7), <http://www.catb.org/jargon>.

ROSEN, L., “*Open Source Licensing: Software Freedom and Intellectual Property Law*”, Upper Saddle River (New Jersey), Prentice Hall, 2004, p. 241-253.

VÄLIMÄKI, M., “*Dual Licensing in Open Source Software Industry*”, *Systèmes d’Information et Management* 8(1), 2003, pp. 63-75.

VAN HEKESSEN, A., HOLLEBEEK, N., TJONG TJIN TAI, E., “Licentievormen van open source software”, in *Open Source Software: Een verkenning naar de juridische aspecten van open source software*, NVvIR, Elsevier juridisch, 2006, pp. 99-100.