

The European Union Public Licence (EURL)

Patrice-Emmanuel Schmitz^a

(a) Legal expert – www.Joinup.eu.

DOI: [10.5033/ifosslr.v5i2.91](https://doi.org/10.5033/ifosslr.v5i2.91)

Abstract

This paper details the origin and main characteristics of the European Union Public Licence (EURL), an OSI-approved free or open source software licence, copyrighted by the European Union. It focuses on the new version 1.2 of the EURL that has been drafted in 2013, which the European Commission reports will be published before the end of the year. However, comments are relevant for the version 1.1 as well. What makes the EURL unique is its multilingual working value, specific warranties, references to the Court of Justice of the European Union and its provisions related to licence compatibility, making its copyleft “variable” for facilitating interoperability. The operation of this copyleft component of the licence is probably its most specific aspect, sometimes wrongly understood as a possibility for “relicensing”. This is therefore especially developed in this paper.

Keywords

Law; information technology; Free and Open Source Software; licence; EURL; European Union

Origin of the EURL

From 2001 to 2005 the European Commission (EC) started focusing, in particular through programmes dedicated to reinforcing interoperability and the development of the Information Society, on the advantages of adopting the free/open source (FOSS) model for sharing software resources of public bodies within the EU, based on the use of open standards. One of the conclusions was: “the Commission should lead by example, distribute its own produced software and then encourage the public sector in Member States to do the same”.¹ As a license is needed for any software sharing (redistribution for reusing, adapting, etc.), the EC set certain requirements for this licence. It must:

1. grant all Free (or Open Source) software freedoms;

¹ “Report on Open Source Licensing of software developed by The European Commission”, 16th December 2004, online at <https://joinup.ec.europa.eu/elibrary/case/open-source-licensing-software-developed-european-commission-applied-circa-solution-20>

2. ensure protection from exclusive software appropriation (therefore be a “share alike” or “copyleft” licence);
3. have working value in all official EU languages (so there is no need for sworn translators in Court and related institutions for translations);
4. conform with European copyright law and terminology;
5. include the “communication to the public” right, including Web distribution / Software as a Service - SaaS (in such case, the software is not distributed as a downloaded package or as a CD-Rom, but as a Cloud Computing application that remote users access via Internet)²;
6. clarify the applicable law and competent court, as requested by EU institutions;
7. approach warranties and liability in conformity with “Case law” (a general exclusion of liability is not valid before most European courts); and
8. not be too long, too complex, but be comprehensive and pragmatic.

Preliminary studies carried out in 2004³, showed that no existing licence was found that complied with at least four of the key requirements (N° 3, 4, 6 and 7). Therefore, the decision to write an “EURL” was taken. After a public consultation that provided substantial improvements⁴, about 50 persons contributed to the writing of the EURL. The work from the original team was complemented by contributions from IPR lawyers from 22 Member States.

As a result, the EURL version 1.1 was certified / approved by the Open Source Initiative (OSI) in March 2009. Since then, and particularly during the last few months, the use of the EURL for licensing projects has been growing strongly. The first yearly end-of-year evaluation (2012) counted about 500 projects (some of them with up to 100 licensed files), and new projects are published every week. The European Parliament has selected the EURL for the distribution of its first FOSS project.⁵

The European Commission considers that the EURL is not a “vanity licence” (where the main motivation of the author is just to forge “its own” licence and attach its name to it⁶), but answers to a number of relevant issues, starting from the fact that governments and public sector organisations in general in Europe are often legally obliged to use instruments with a working value in their local language (requirement N° 3). At least three additional points were also important to clarify (N°4, 6, 7): terminology, applicable law / competent court, and warranty and liability disclaimers.

Other points in the list are not unique to the EURL, even if the coverage of SaaS is not frequent (the OSL 3.0⁷ and the GNU Affero General Public License built on the GPL v3⁸ presents similar characteristics on this specific point). Some licences (in particular the very permissive ones, like

² The first EURL version (1.0) was less explicit on this specific point. It has been clarified in version 1.1.

³ The Report on Open Source Licensing of software developed by the European Commission, op cit, considered the OSL as the best choice and proposed two options: adapting the OSL or writing a specific licence. The Commission opted for writing the EURL.

⁴ Report on outcomes of public consultation about the EURL, 30th November 2005, online at https://joinup.ec.europa.eu/sites/default/files/GPOSS_adv-06_V11%2019%20Jan%2006.doc

⁵ Online at <http://www.at4am.org/eurl/>.

⁶ This “reproach” has been addressed to institutions like the NASA, after evaluation of the impact of the NOSA (Nasa Open Source Agreement): see comments from the US Department of Defence – <http://opensource.com/life/11/3/nasa-concludes-first-open-source-summit-aims-make-openness-default>.

⁷ Online at <http://opensource.org/licenses/OSL-3.0>.

⁸ Online at <http://opensource.org/licenses/AGPL-3.0>.

the BSD or the MIT) are much shorter, but it is commonly acknowledged that the EURL is concise and comprehensive compared with some other “copyleft” licences⁹.

EURL and Licence Proliferation

In the early days of free/open source software (meaning until the year 2000) the GNU/GPL v2 licence and its LGPL “library” variant were adopted by some 90% of all FOSS projects. Since then, the number and the frequency of use of other licences have increased strongly. OSI has officially “approved” more than 60 licences¹⁰. The GPLv3 and AGPLv3 introduced in 2007 have not replaced the previous GPLv2, which still seems to be the most used licence (e.g. by GNU/Linux). Some important business projects are driven by foundations (Apache, Mozilla etc.) promoting other FOSS licences. Such licence proliferation may be considered unfortunate, because it has made the work of developers more complex as it raises the potential of legal incompatibility at the time of distribution, when an ICT solution includes multiple FOSS components. However, it looks to be a definitive fact: nearly every week, new licences are drafted¹¹. The task of “OSI licence reviewers” as described in their code of conduct¹² seems endless, even considering that only a small part of all licences that could be OSD-compliant have been submitted to them: they do not currently approve many new licences (CeCILL 2.1 is a unique exception in 2013 so far).

To compensate the issue of licence proliferation, the EURL has chosen a system to ensure legal interoperability between potentially incompatible licenses (see below). The EURL has also inspired other (non-EU) governments (e.g. Quebec in Canada), which have asked permission to adapt the EURL so as to use it as a template for their own needs (changing names and jurisdiction only). In such case, maintaining the same list of compatible licences may strongly reduce the impact of such licence proliferation.

Similarly, the new (2013) OSI approved version 2.1 of the CeCILL licence¹³ (used by the French administration) now includes the EURL and the GPL/AGPLv3 as downstream compatible licences, which looks positive for developers from both communities.

The EURL Used as a “Reference”

Another point of interest for the EURL is to be part of the European Interoperability Framework (EIF) and to be used as a reference, especially in public software requirements and procurement agreements¹⁴. In line with the EU ministerial declarations¹⁵ on the opportunity to reduce

9 In particular, some analysts were disappointed by the complexity of the GPLv3 that is nearly three times longer. For example, Ernest Park noted: The EURL v1.1 is a legal instrument, simple and clear to interpret, with less baggage than the GPL v3 in “Freedom and choice in open source licensing – Comparing the EURL v1.1 and the GPL v3”, online at <http://www.linux.com/news/biz-os/legal/18749-freedom-and-choice-in-open-source-licensing-comparing-the-eurl-v11-and-the-gpl-v3>.

10 <http://opensource.org/licenses/alphabetical>

11 <http://projects.opensource.org/pipermail/license-discuss/>

12 <http://opensource.org/codeofconduct/licensing>

13 <http://opensource.org/licenses/CECILL-2.1>

14 See the Guide for the procurement of standard-based ICT / Elements of Good Practice, European Economics 23, March 2012, available online at <http://cordis.europa.eu/fp7/ict/ssai/docs/study-action23/d3-guidelines-finaldraft2012-03-22.pdf> and the ISA standard “Sharing and reusing clauses”, available online at http://joinup.ec.europa.eu/elibrary/document/isa_share_reuse_d_2-1-standard-sharing-and-re-using-clauses-contracts.

15 In 2005, the Manchester eGovernment ministerial declaration stated: “Member States will, during the period 2006-2010, share technologies, where appropriate develop common solutions and work towards interface harmonisation of existing solutions” (in the field of eProcurement), and in 2009, the Malmö ministerial declaration on eGovernment stated: “The Open Source model could be promoted for use in eGovernment projects.” (online at

development costs by sharing and reusing software, contracting authorities should obtain from their suppliers the right, not only to use but also preserve their rights to redistribute the developed software in the future, as the case may be (e.g. in case the development is successful, interesting for other stakeholders, and if a sharing decision is taken by the authorities).

Therefore, suppliers must not only assign or license to the contracting authority the IPR of the solution (including the software code), but must also guarantee that it can be legally (re)distributed to third parties by the contracting authority, without any copyright infringement issue or licence conflicts (if several components of the solution were distributed under non-compatible FOSS licences) and royalty free (e.g. to cater for the situation in which some proprietary standard or patents were implemented).

An example¹⁶ of such provision is:

“The supplier will grant that the purchasing authority has the right to distribute the delivered application under the European Union Public Licence (EURLv1.1 or later) or any licence(s) providing the rights stated in the article 2 of the EURL.”

Such reference to the EURL is especially convenient due to its multi-lingual validity: it can be part of public procurement specifications written in any language of the EU.

Rights Granted by the EURL v1.1

The rights granted (by article 2 of the EURL) to the recipients of the covered Work are unmodified through all versions of the license. They are those required by the Open Source Definition (OSD): to use, reproduce, modify, communicate, and re-distribute the work.

In addition, it is stated (in article 2) that the exercise of such rights and the use of necessary licensor patents (if any) must be royalty free (RF). This means that one may sell software or works covered by the EURL (e.g. for a lump sum representing a contribution to the development costs of a standard or of a software solution, or a fixed maintenance fee for support services etc.), but once this is done, any further exploitation of the covered Work itself (but not necessary a larger work using the EURL software) cannot be subject to the payment of royalties (e.g. a fee – even small or reasonable – per use or per user).

This principle of RF licensing, expressly formulated in article 2 of the EURL, is in fact fundamental to any FOSS licensing. When freedom is granted to all possible recipients in the world to exploit and make derivative works and to redistribute such works to anyone, the control of the use for the charging of royalties becomes impossible to implement in practice. As highlighted in a 2012 workshop organised by the EC¹⁷, this principle has yet to be integrated by many institutions, like the standards developing organisations (SDO), for whom development costs need to be covered, which is economically legitimate. However, in so far as SDOs want their specifications to be implemented by FOSS communities and also recover their costs by adopting a FRAND (fair reasonable and non discriminating) licensing policy, they do and will have to

<http://ec.europa.eu/digital-agenda/sites/digital-agenda/files/ministerial-declaration-on-egovernment-malmo.pdf>)

16 This example is given in Appendix 1 of the Guide accompanying the European Commission communication “Against lock-in: building open ICT systems by making better use of standards in public procurement” - COM(2013) 455 / SWD(2013) 224, 25th June 2013.

17 DG Enterprise workshop, 22nd November 2012 “Implementing FRAND standards in Open Source: Business as usual or mission impossible?”, reported online at http://ec.europa.eu/enterprise/sectors/ict/standards/extended/event_open_source_en.htm .

imagine alternative solutions: global agreements with FOSS representative foundations or a dual licensing policy (RF for FOSS implementations, reasonable royalties possible in other cases and where infrastructure or base technology is concerned). This would not be discriminatory against non-FOSS (or proprietary, or infrastructure) implementations, as FOSS is not a group, a product or a technology, but a legal regime that anyone may adopt for distributing software¹⁸.

What Made the EURL v1.1 Specific?

The EURL is specific and different from other FOSS licences on a number of points:

- **Multilingualism**
This point is the most visible: like many other European Union legal instruments, the EURL is available in 22 languages. Gaelic and Croatian version have yet to be published.
- **Terminology**
The EURL is drafted to work under European Law, even if it may be used outside the European Union and submitted to third country courts. Relevant provisions include the copyright terminology (the “communication to the public”), the limitation of liability clause, and the reference to European treaties.
- **Warranty**
The covered work is licensed without warranty, except one important one: the original licensor and every subsequent contributor warrant that they are the authors (or licensees with sufficient rights) of their own contributions. This reinforces the security offered by the licence (regarding possible copyright infringements) and is in the end the type of requirement that one finds in most reasonable contributor agreements.
- **Reference to the European Court**
Taking advantage of the treaties (TFEU), the EURL will benefit from interpretation by a unique jurisdiction in case of litigation: the Court of Justice of the European Union. In addition, the 28 Member States jurisdictions may address questions and be answered by a single European Court.
- **Variable “Copyleft”**
The EURL is “copyleft” on source code and binaries, but its share-alike effect is, in the interest of license interoperability, variable¹⁹ in the case of combined derivative works, as discussed below. The fundamental purpose is to avoid exclusive appropriation of the covered software.

Changes Planned in the EURL v1.2

As from 2012, the EC’s objective is to reinforce its legal toolkit (including the EURL) for more software sharing, reuse and interoperability. Two key objectives are increased compatibility, and

18 See Italian Supreme Court decision (Corte Costituzionale, 22nd March 2010) <http://www.cortecostituzionale.it/actionSchedaPronuncia.do?anno=2010&numero=122>, commented by Carlo Piana in IFOSSLR Vol 2, No 1 (2010), DOI: 10.5033/iffoslr.v2i1.38.

19 The notion of « variable copyleft » was coined for the EURL by Rowan Wilson (Oxford University) <http://www.oss-watch.ac.uk/resources/eurl>.

updating to the European legal framework. On the one hand, if “copyleft” aims to protect against appropriation, licence conflicts may also create legal barriers between FOSS communities. Therefore the EURL includes an appendix of “compatible licences” providing interoperability with a list of similar licences (based on a 2006 study²⁰). However, by 2012 this list was outdated. On the other hand, several changes in the European legal framework (the entry in force of the Lisbon Treaty in 2009) meant that a new version of the EURL was considered, version 1.2²¹, and a working draft was submitted to public consultation²² as from mid-December 2012. The publication of the new version was planned originally by June 2013, but due to translation and organisational reasons it is now promised by the end of 2013.

The EURL v1.2 will present the following main differences with the previous v 1.1:

- Terminology is adapted in consideration of the Lisbon Treaty (this concerns the name of EU institutions and the references to the TFEU) in articles 14, 15;
- The licence now covers “the Work” – in version 1.1, it referred to both “the Work” and to “the Software”, which was confusing. The Work can be software, but also any other kind of copyrighted work: copyrighted data, specifications, documentation etc. – the modification is done in the introduction and in articles 1, 4 and 7;
- The scope of possible “additional agreements” is enlarged (i.e. they may cover jurisdiction and any other provisions, in so far as the granted rights are not restricted) – in article 9;
- The list of compatible licences is extended to licences published after the initial EURL: GNU GPLv3, AGPLv3, MPLv2, LGPL v2.1 and v3, CeCILL v2.1. The list is extended to the EURL itself (all versions as from v1.1) – in the Appendix of compatible licences according to article 5 EURL.

Software covered “by the EURL” will be automatically covered by the new license version, but licensing of software covered by “the EURL v1.1 only” will not. After publication of the EURL v1.2, it will still be possible to distribute code under the EURL v1.1 only.

As reported above, the modifications introduced by v1.2 will stay quite limited. The main change is the extension and management of the list of compatible licences. In particular, the extension to the GNU GPL v3, and to the GNU AGPL v3 were welcomed without restriction by communities²³ and on the public discussion forum implemented on Joinup.eu²⁴ Compatibility with the GPL v3 looks to be the solution for a question of principle (more than for a practical issue, as no real interoperability problem was reported so far): the EURL v1.1 has been presented by external analysts²⁵ as “incompatible by design” with the GPL v3, which was not the aim of the European Commission. Still today, as reported by Professor Moglen, there are bitter regrets that the European Commission declined (in 2006) to participate in the making of the GPLv3, on the

20 Fabian Bastin and Philippe Laurent, «Study on the compatibility mechanism of the EURL »

<https://joinup.ec.europa.eu/system/files/doc/Doc3ef5.pdf>

21 The draft EURL v1.2, with changes highlighted, is published on the European Commissions' Joinup site :

<https://joinup.ec.europa.eu/sites/default/files/EURL%20v1.2%20-%20Draft%20EN%20v15%20Mar%202013.pdf>

22 The contributions to the public consultation are published: <https://joinup.ec.europa.eu/community/eurl/topic/public-consultation-draft-eurl-v12>

23 Comments done on LWN network are illustrative <http://lwn.net/Articles/529737/#Comments>

24 See e.g. <https://joinup.ec.europa.eu/community/eurl/topic/public-consultation-draft-eurl-v12>

25 Ernest Park, Freedom and Choice in Open Source Licensing: Comparing the EURL v1.1 and the GPL v3

<http://www.openlogic.com/wazi/bid/187980/Freedom-and-Choice-in-Open-Source-Licensing-Comparing-the-EURL-v1-1-and-the-GPL-v3>

ground that it could only participate in government-to-government processes²⁶. This may be the reason why, while accepting compatibility with the GPLv2 and a “de facto” compatibility with the GPLv3, the FSF still considers the EURL v1.1 as globally incompatible with the GPL, which is the most confusing and contradictory²⁷. The publication of v1.2 should at least close this controversial chapter, and –hopefully – pave the way for greater understanding.

How is the EURL's Variable Copyleft Implemented?

Interoperability (at licence level) is the possibility to reuse the covered code in other projects, possibly in combination with code(s) covered by other licences, while keeping the freedom to distribute the resulting combination, even when considered as a derivative or composed work under copyright law.

Interoperability is a non-issue with permissive licences (as the BSD, the MIT) because – with the exception of respecting copyright notices - they establish no conditions for copying, merging and/or redistributing the covered code, even inside the software code of proprietary or non-free applications.

Interoperability becomes an issue with “share-alike” (or “Copyleft”) licenses, when a binding condition of the licence is to keep the covered code and its evolutions under inherited FOSS conditions, in order to avoid its exclusive appropriation under non-free licensing. Just how far this share-alike obligation is applied with respect to derivative or composed works based on the original covered work is often debated, in particular with regard to GPL licenses.²⁸

The EURL is one of these share-alike licences, and the following question is posed: how strong is the EURL copyleft? In other words, to what extent must any re-distribution of the work or any combination of software with the work be done under the same EURL licence, according to its share-alike terms? And therefore, is the work protected from subsequent distribution under other licensing terms, which could lead to appropriation for the benefit of a third party software vendor?

In Europe, there are still some doubts whether the concept of “strong copyleft”, whereby simply linking²⁹ the code covered by a “copyleft” licence with another source code automatically may extend the coverage of the licence to this other code, would be generally considered effective (in any EU member state and whatever the licence, GPL, EURL or any other could be). There are specific exceptions for interoperability implemented by Directive 91/250 on the legal protection of computer programs. In May 2012, the Court of Justice of the European Union interpreted Directive 91/250, “as meaning that neither the functionality of a computer program nor the programming language and the format of data files used in a computer program in order to exploit certain of its functions constitute a form of expression of that program and, as such, are not

26 Eben Moglen, An introduction to the most used FOSS licence, the GPL license, in :European Parliament – Legal aspects of free and open source software – compilation of briefing notes / p. 12

<http://www.europarl.europa.eu/document/activities/cont/201307/20130708ATT69346/20130708ATT69346EN.pdf>

27 FSF declares EURL 1.1 compatible with GPLv2 and details how – via CeCILL – the EURLv1.1 is also compatible with the GPLv3, but still categorises the EURL as GPL-incompatible (!) <http://www.gnu.org/licenses/license-list.html#GPLIncompatibleLicenses>

28 Malcolm Bain, “Software interactions and the GNU General Public Licence” IFOSSLR 2010 p. 165 <http://www.ifosslr.org/ifosslr/article/view/44/74>

29 Linking makes two software programs work in a single application without merging their source code. Generally speaking, static linking combines components through compilation, copying them into the target application and producing a merged object file that is a stand-alone executable., and dynamic linking combines components when the application is loaded (load time) or during execution (run time).

protected by copyright in computer programs for the purposes of that directive".³⁰ Although this judgement was not taken in the framework of free software distribution, it might have repercussions in this field, too. More particularly, it raises the question if, by licensing his/her work, a copyright holder can restrict or impose conditions on legitimate licensees from reproducing and distributing (under any other licensing terms of their choice, FOSS or non-FOSS) the specific portions of the code that are strictly necessary for linking / implementing interoperability between the licensed program and other works, such as data formats or APIs (application programming interfaces). The question is most probably still open³¹ and hopefully the Court will have the chance to clarify this matter in future case-law.

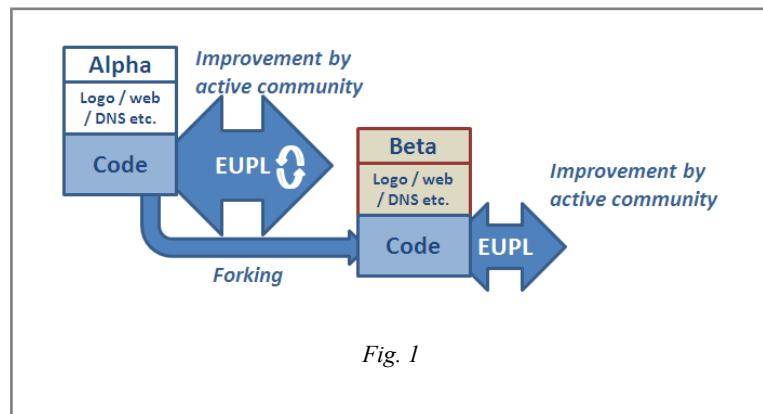
In order to understand the way the "variable copyleft" of the EURL operates, we can analyse three cases:

- a) The normal case where a copy or a derivative work is distributed
- b) The exception for interoperability, implemented by article 5 of the EURL
- c) Inside the exception, the case where a more permissive licence could be applied

a) The Normal Copyleft Reuse Under the EURL

With the aforementioned reservations, we can state that the EURL "copyleft" is as strong as possible, on source code and binaries of copies and all derivative works of the covered work (derivative within the meaning applied by a court interpreting the license), with defined interoperability exceptions.

Let's first consider the normal case with regard to the distribution of code under the EURL:



- A project "ALPHA" is more than just its software code: it is an organisation, owned by a person or a body, with an active community of developers, a web site, DNS, logo etc. Globally, this project "ALPHA" can never be "re-licensed" outside the will of its original

³⁰ Decision online at <http://curia.europa.eu/juris/document/document.jsf?text=&docid=122362&pageIndex=0&doclang=en&mode=req&dir=&occ=first&part=1&cid=564907>

³¹ For a specific discussion on linking (in the context of the GPL), Malcolm Bain, op cit. <http://www.ifosslr.org/ifosslr/article/view/44/74>

licensor (who is free, as the 100% copyright owner, to provide exceptions or to distribute the project software under various licences, which is called dual or multiple licensing).

- Re-distribution of the code of project "ALPHA", covered by the EURL, is possible inside the context of another project (e.g. "BETA"), meaning after copying and pasting ALPHA's code into BETA's software, as is or with modifications, and thus making BETA's code a derivative work according to applicable copyright law. -In all events, this redistribution must be done under the same EURL licence.

This is sometimes known as a "forking", when BETA takes all or most of ALPHA's code and distributes it as another owner (of the derivative work), brand name, logo, web site etc. Forking, as such, is rare, at least when the original licensor organises an active community around its ALPHA project: if this is the case, all improvements will normally be done and shared within the ALPHA project, without needing to create any new BETA project.

Forking may occur for 1) licensing / philosophical reasons or 2) for functional / technical reasons:

- 1) A first example, is the case where ALPHA's licensor has lost its independence (e.g. is purchased by a proprietary vendor), and the community decides to re-launch the project to preserve EURL licensing (though this is not likely to happen if the licensor is a public sector body);
- 2) A second example is the case where ALPHA's licensor does not want to integrate/support certain new functions. For example, an Indian government wants to localise/adapt software distributed by the European Parliament in local Indian languages, but the EP does not want to be involved in this process. However, the new Indian project code must also be distributed under the EURL.

The hypothesis where a portion of the covered code is merged in another project is similar: as a derivative, this project must be covered by the EURL, in case it is distributed.

b) Exception to the “Normal Copyleft” (Reuse in Other Copyleft Works)

The third paragraph of Article 5 EURL was written for interoperability reasons: to allow the covered code to be integrated with other projects' code covered by other stronger or weaker copyleft licences. “Integration” is here in its widest sense where a derivative work is produced, including mixing significant parts of the code as resulting from a copy/paste operation as described above. In other cases where the resulting work is not derivative, but a simple compilation or aggregate, this paragraph will not apply. This third paragraph reads as follows:

“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.”

In the EURL v1.2, the list of compatible licences published in the Appendix contains the following licenses (v 1.2 additions to the Appendix of v.1.1 are in bold):

- *GNU General Public License (GPL) v. 2, v. 3*

- *GNU Affero General Public License (AGPL) v. 3*
- *Open Software License (OSL) v. 2.1, v. 3.0*
- *Eclipse Public License (EPL) v. 1.0*
- *CeCILL v. 2.0, v. 2.1*
- *Mozilla Public Licence (MPL) v. 2*
- *GNU Lesser General Public Licence (LGPL) v. 2.1, v. 3*
- *Creative Commons Attribution-ShareAlike v. 3.0 Unported (CC BY-SA 3.0) - for works other than software*
- *European Union Public Licence (EURL), any version as from 1.1*

The presence in the list of weaker copyleft licences (the LGPL and the MPL) may look unnecessary for most cases of integration: these licenses permit larger works incorporating their code to be distributed under the EURL. However, in case the EURL covered code is actually mixed into the component covered by the LGPL or the MPL, the inclusion of these licenses in the Appendix is useful.

The interoperability exception of the EURL will allow licensees running a third project “DELTA”, to reuse files or source code covered by one of the above compatible licences in the DELTA project code, and to insert or merge the EURL covered code in DELTA's code: they can licence the resulting work as a whole under the compatible licence.

Notwithstanding the permission to authorise “this distribution and/or communication” of the new DELTA project software “as a whole” under a compatible licence, the EURL's provisions still apply to the part of the code that was originally covered by the EURL. In particular, it maintains the attribution obligations (article 5: the Licensee shall keep intact all copyright, patent or trademarks notices and all notices that refer to the Licence and to the disclaimer of warranties).

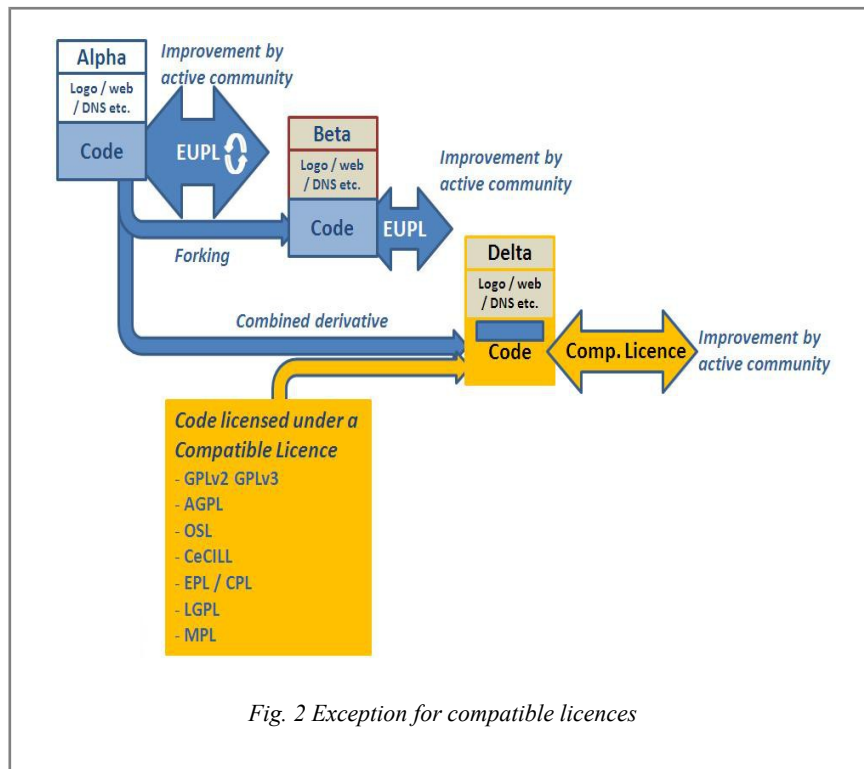


Fig. 2 Exception for compatible licences

Conditions for such “variable copyleft” are as follows:

1. Software code covered by the EURL is combined in/with another, different work.
2. The combination (larger work) forms a derivative work (i.e. the covered code was copied and pasted in another file, covered by another licence) or is otherwise covered by the other license, so that the resulting merged code must be licensed as a whole under the new license. Keeping distinct licences, like for the various parts of an aggregate work, is not an option (because the codes are mixed).
3. The other work, with which the code covered by the EURL is merged, is licensed under a Compatible Licence (according to the list).
4. The same compatible licence (according to the list) is used to license the new larger work “as a whole”.

The exception for compatible licence described above to relicensing some or all of the EURL covered “code” should not be understood as the possibility to “relicense” a “**project**”³². As said above, this is obviously not the case: the reuse of some EURL code in the project “DELTA” will not impact the project “ALPHA”, but just the code that instance of the EURL code that has been reused.

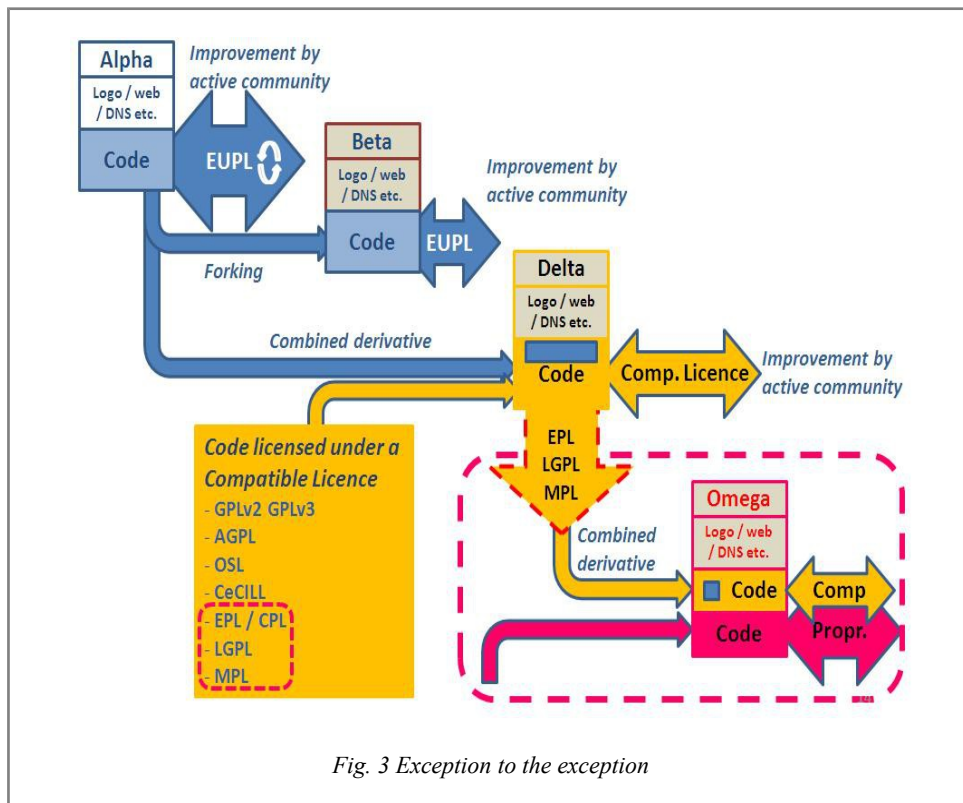
32 For example, the too brief formulation used by the Free Software Foundation may induce recipients in error: “The EURL allows relicensing to GPLv2, because that is listed as one of the alternative licenses that users may convert to”, at <http://www.gnu.org/licenses/license-list.en.html#GPLIncompatibleLicenses>

Is there any risk to see someone licensing some trivial code (some kind of copyrightable wrapper, but without real functional added value) under a compatible licence for creating a formal larger work and licensing it under this compatible licence? No cases were reported in five years EURL distribution (2007-2012). It is not the way FOSS operates. Making trivial forking is losing time and reputation. A forked work is sustainable only when a working community takes it over and improves it substantially.

The presence of the EURL itself in the list of compatible licences is not trivial. It was important to avoid introducing more licence proliferation issues by making the two versions 1.1 and 1.2 of the EURL incompatible (an issue that was not avoided when introducing the GNU GPL v3 which is incompatible with the GPLv2). Therefore all projects licensed under “the EURL v1.1 only” will be authorised to integrate code that will be developed under the EURL v1.2. The reciprocal situation is not automatic, but – due to the reversibility – it is assumed that exceptions will be easy to obtain, if and when needed.

c) Exception to the Exception

Because three of the listed compatible licences are more moderately “copyleft” (or only at file level) it may be that some code covered by the EURL could also be reused in a third generation project that takes up the mixed code (EURL + EPL, for example) and redistributes it – in binary executable form – under a non-FOSS licence.



This is possible where:

1. the "daughter project" DELTA is (by decision of its own licensor and because code under these licences - EPL, LGPL or MPL - was reused) distributed under one of these more moderately copyleft licences listed as Compatible;
2. that some code from DELTA is combined or forked in a "grand-daughter" project "OMEGA", and;
3. that the OMEGA licensor decides to distribute its executable version under proprietary terms, something that is permitted, subject to certain conditions, by the moderate copyleft licenses.

Even in such a case (that has never occurred in real world so far) the portions of the DELTA code present in OMEGA will stay covered by their primary licence (EPL, LGPL or MPL) and – as stated before – the EURL covered code that is present in these files will stay covered by the EURL provisions and marked with its specific attributions. These files must stay FOSS and publicly available as source code, but the copyleft is generally limited at file level according to the provision of these licences (meaning without any copyleft impact on the rest of the OMEGA project).

This possible exception has made some analysts to declare that the EURL *"gives recipients ways to relicense the work under the terms of other selected licenses, and some of those only provide a weaker copyleft. Thus, developers can't rely on this license to provide a strong copyleft"*³³.

This point is of course – at least theoretically – founded. But we have to see it in a context, and temper it:

- The term "relicense the work" is somewhat ambiguous, as previously stated.
- Some compatible licences provide a "weaker" copyleft (LGPL, MPL, EPL): it does not mean that these licences are weak or permissive: they are copyleft, but at file level, without the aim of propagating the licence coverage by e.g. dynamic linking.
- The strict focus on "strong copyleft" may not be ideal in an interoperable world, where multiple FOSS licences coexist. Furthermore, the notion of "strong copyleft" is still unclear, debated and has not received confirmation from European case law.

Conclusion

The above series of examples illustrate that the EURL achieves several of its aims:

- It intends to protect effectively the covered code and derivative works from exclusive appropriation by a third party;
- It makes some part the covered code reusable in OTHER free software projects under other licenses (without re-licensing the original project); and
- It does not prevent the reuse of some code of these other projects by the software industry

³³ FSF – op. cit.

with a variety of business models.

The EURL is not a goal in itself. It is just a tool for facilitating the distribution of copyrighted work by new categories of FOSS licensors, including (but not exclusively) the European public sector authorities and enterprises. The tool results from searching for the best possible compromise between copyleft (with the aim to avoid exclusive appropriation of the covered work) and legal interoperability (with the aim to see the work widely used in many other projects and to facilitate the distribution of these projects).

The tool is not perfect, it has been adapted twice (v1.0 in 2007, v1.1 in 2009) and changes that are expected in 2013 (v1.2) are reinforcing interoperability aspects.

In the future, it may be that the evolution will continue with new versions of the EURL, or that a better legal instrument will eventually be adopted and implemented to replace software licences. Two directions look possible in the current context:

- More copyleft licences follow the EURL example and develop their list of compatible licence³⁴;
- The relevant licence steward organisations (despite all their differences) unify their efforts and produce a multilingual working, worldwide valid, comprehensive licence with the aim to conciliate the requirements from most FOSS stakeholders. In such case, the EURL may completely disappear or become a facet in a wider project.

However, predicting the future is hazardous; as everyone knows that yesterday is history, today is a gift, tomorrow is a mystery...

References

- The EURL v1.1 and v1.2 – text of the licence, in 22 languages – Joinup.eu.
<https://joinup.ec.europa.eu/software/page/eurl>
- Guidelines for using the EURL <https://joinup.ec.europa.eu/software/page/eurl/eurl-guidelines>
- Guidelines on public procurement of Open Source Software
<https://joinup.ec.europa.eu/elibrary/document/guideline-public-procurement-open-source-software>
- Spanish Royal decree 4/2010 (English version) see in particular article 16
http://administracionelectronica.gob.es/recursos/pae_000002017.pdf
- The EURL in Italy: www.eurl.it
- “Experience of introducing the EURL in ISTAT” (Carlo Vacari 2010) presentation slides (in Italian) : <http://fr.slideshare.net/vaccaricarlo/introduzione-eurl-in-istat>
- Malta public sector software distribution policy

³⁴ This is the way followed by the new version 2.1 of the CeCILL licence, which is now downstream compatible with the GPL, AGPL and EURL - <http://www.cecill.info/index.en.html>

https://www.mita.gov.mt/MediaCenter/PDFs/1_GMICT_P_0097_Open_Source_Software_v2.0.pdf

- Guide for the procurement of standard-based ICT / Elements of Good Practice – (European Economics 23 March 2012) - <http://cordis.europa.eu/fp7/ict/ssai/docs/study-action23/d3-guidelines-finaldraft2012-03-22.pdf>
- ISA standard “Sharing and reusing clauses”
http://joinup.ec.europa.eu/elibrary/document/isa_share_reuse_d_2-1-standard-sharing-and-re-using-clauses-contracts

About the author

Patrice-Emmanuel Schmitz is a legal expert at www.Joinup.eu.

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 5, Issue 2 (December 2013). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Schmitz, Patrice-Emmanuel (2013) 'The European Union Public Licence (EURL)', *International Free and Open Source Software Law Review*, 5(2), pp 121 – 136
DOI: [10.5033/ifosslr.v5i2.91](https://doi.org/10.5033/ifosslr.v5i2.91)

Copyright © 2013 Patrice-Emmanuel Schmitz.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND available at <http://creativecommons.org/licenses/by-nd/2.0/uk/>

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.

