

EUPL Interoperability

Which FOSS components in EUPL solutions? *By Patrice-Emmanuel Schmitz*

At a time a growing number of European governments intend to encourage the distribution of their software under the European Union Public Licence v1.1 (EUPL), it is useful to keep in mind the implications of this strategy and to develop e-Government solutions based on components licensed under compatible conditions. How is the EUPL interoperable with copyleft FOSS (free or open source) licences¹ and which component can you link or merge in a solution distributed under the EUPL?

The free licences categories

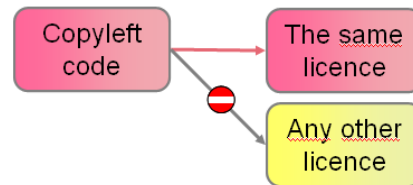
Historically, the licences for distributing free or open source software (FOSS) where divided in two families: **permissive** and **copyleft**. To make short a long story, the most permissive licences tolerate to merge, combine or improve the covered code and to re-distribute it under any licence (including non-free or “proprietary”). The BSD is the most famous example of permissive licence.



At the contrary, the copyleft licences impose the use of the same licence as soon the distributed work includes (= is derivative, is merged, modified at source

¹ Except when quoting American texts or sources, we use here the UK writing “Licence” and not the American writing “License”.

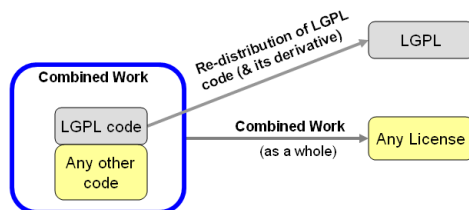
code level with etc.) the covered code. The Gnu GPLv2 is the most famous example of these licences.



Copyleft is perceived as a barrier for using components as “software building blocks” (this is the case of software libraries), because developers may want to distribute the combined result under the licence of their choice. In most cases, simply using, aggregating or combining various components will not generate derivative works according to copyright law (and doing so is authorised by all FOSS licences, even copyleft). However, there are risks that some integrations (when code is modified, merged) are derivative works. The uncertainty provides from the lack of specific case law in this field.

P-E Schmitz is lawyer and legal expert for www.osor.eu, the European Union collaborative open source observatory and software forge. This article presents the author personal opinions and does not commit the European Commission.

Therefore a third category was promoted² and was presented as “**Weak copyleft**” (WC). The Gnu LGPL is the best example of these copyleft licences, which are aimed to be more interoperable with others. The work that is assembled with the covered building blocks (in fact a combined-derivative work) can be distributed under any licence, but the LGPL building blocks “taken alone” will remain distributed as free software under the LGPL, including their possible improvements, if any.



Around the year 2000, the pioneer FOSS licence Gnu GPLv2 was used in 85% of all FOSS projects and it was considered as “good for the community” to deal with only one copyleft licence. This opinion was motivated by the fact that strong copyleft licences are – by design – legally incompatible. A developer could of course technically combine pieces of code covered by incompatible licences, but the distribution of the resulting derivative work would be prohibited according to copyright law, because of the licence conflict.

Today, the phenomenon of licence proliferation has made the situation a little bit more complex: the Open Source Initiative maintains a list of about 60 licences, but receives hundred some each years and up to 1.800 different models of free software licences have been identified by experts³. Many of these

² By the Free software foundation, representing the most “purist” trend in the free software philosophical movement

³ After counting 1,800 free software licences used in hundreds of thousands of projects, the Black Duck company patented (Patent US

recent licences are copyleft (including the new Gnu GPLv3 introduced in June 2007, which is not compatible with the previous GPLv2). This generate uncertainties for licensing large open source /free software solutions (i.e. in eGovernment) that are combined works based on the integration of multiple components. Some auditors said that one half of the projects they audit suffer from some sort of licensing compatibility issues, with about on fifth suffering from serious problems⁴.

Therefore the challenge is not anymore avoiding or reducing licence proliferation (this is a definitive fact). It is to implement interoperability between strong copyleft licences.

The new “IC” category

When needing and creating a new licence having equal legal value in 22 languages, a terminology compliant with European legal environment, limitations of liability or warranty valid in all Member States and convenient jurisdiction - applicable law provisions, the European Commission decided to insert copyleft conditions in its EURL, while avoiding the incompatibility with the most used other copyleft licences.

This makes the EURL a representative of a fourth category of licence, that are “**interoperable – copyleft**” (IC). It is therefore useful to define how this EURL interoperability works, at a time where new government policies adopt the EURL for distributing their software⁵. The

7,552,093 B2) the technology for controlling the use of open source licensing in a multi-source development process.
<http://www.earthtimes.org/articles/show/black-duck-software-awarded-patent,1147065.shtml>

⁴ <http://oss-ipr-database.com/>

⁵ In Spain, the Royal decree 4/2010
http://www.csae.mpr.es/csi/pg5e41_ingles.html;

European Union Vice President Neelie Kroes, in charge of the digital agenda, refers to the EUPL for easy licensing under the EU legal framework⁶. Procurement authorities, who are aware of incompatibility risks, specify that solutions delivered by their ICT providers “must be easily distributable by them – as a whole – under the EUPL”⁷.

EUPL “downstream” compatibility

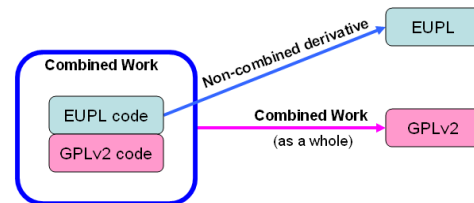
First of all, the EUPL is a copyleft licence (just like the GPL): if you create source code and decide to distribute it under the EUPL, this will stay “forever”. When a GPL code cannot be distributed under the EUPL, the EUPL code cannot be distributed under the GPL (or any other licence) for the same reason.

However, the EUPL has a “downstream”⁸ **compatibility list** including five other licences: the GNU GPLv.2, OSLv2.1, OSLv3, Eclipse v1.0 (plus the superseded CPLv1.0) and the CeCILLv2.0. This list makes the EUPL interoperable: what is allowed by interoperability is to produce a **derivative work** when **combining** code licensed under the EUPL with other code licensed under a **compatible licence** (GPLv2 in the example below) and to distribute the

resulting derivative solution “as a whole” under the compatible licence.

Please note that the primary licence applicable to the EUPL code “taken alone”, or extracted from the combined solution when it is still possible⁹ will not change, including the possible improvements done. Therefore, if some contributor corrects a bug in a software licensed under the EUPL, localises it in another language or create a new functionality by adding his or her own written code, this contributor is not authorised to switch to a compatible licence and to distribute the new derivative work under the GPL (or under the Eclipse, OSL, CeCILL).

Illustration:



The permission to redistribute the combined work under the GPLv2 (as illustrated above) is resulting from EUPL compatibility clause (a section of article 5), stating:

*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*

In Malta, the 1st June 2010 policy http://ictpolicies.gov.mt/docs/GMICT_P_0097_Open_Source_Software_v1.0.pdf ;

In the Netherlands, the [NOIV Licence wiki](#) (in Dutch);

In Estonia, the national interoperability framework http://www.riso.ee/en/pub/2009it/pdf/Yearbook_ENG.pdf .

⁶ Neelie Kroes speech on Youtube: <http://www.youtube.com/watch?v=ok100U4Fq3Y&NR=1>

⁷ Provisions introduced by Member States ICT agencies in their ICT specifications (i.e. by the Red.es government ICT agency in Spain).

⁸ “Downstream compatibility » refers to licences that may be used for derivative works; “Upstream compatibility” refers to the licences of covered components that could be included in a work licensed under the EUPL.

⁹ It is possible when identified components are still “linked” in a derivative work. It may not be possible when a derivative work really merges codes.

Licence conflict with his/her obligations under this Licence, the obligations of the Compatible Licence shall prevail

In the EUPL, interoperability can be summarized as follows: “licences are interoperable when covered components can be linked in a combined and derivative work (a larger work), which can be distributed “as a whole” under one of their licences.” Outside this scope, interoperability does not need to modify the licensing conditions of each covered component, taken alone (this is important, as some of the compatible licences may have – on many points - different provisions than the EUPL).

The EUPL interoperability mechanism is quite comparable to the way the Gnu LGPL is applicable (nothing was invented, and many other licences are interoperable as well), but it is associated with a much stronger protection against appropriation because of the limitative list of other licences (that are all copyleft) which can be used for distributing a combined work. This is the reason why the EUPL can be categorized as “**Interoperable Copyleft**”.

“Upstream” compatibility

When the “downstream” compatibility of the EUPL is explicitly fixed in its appendix, the “upstream” compatibility is not fixed. However, it is extremely important, as it determines the possibility to reuse existing components for developing and distributing the combined software, which is one of the main use cases of the EUPL. This compatibility is not obvious to clarify, as it was observed that it is not depending on the EUPL, but on the terms and conditions of the upstream licence, and – in specific cases – on the exception lists published by some licensors. For this

reason, the preliminary study done at the time the EUPL was still a draft did not provide precise answers on this point¹⁰.

Let’s fix the scope: first, the question is missing practical interest for covered components “taken alone” (there is no reason why a EUPL licensor would change their primary FOSS licensing conditions). The question is mostly interesting for new combined works that could be considered as **derivative works**, according to the applicable copyright law (integrating or linking multiple covered components in a single new solution). This is also where the risk of licence conflict exists. As mentioned higher, recent European public sector calls for tenders request from providers (their developers and contractors) the facility to distribute the provided application “as a whole” under the EUPL as the single licence applicable to the solution.

Secondly, it may be that the source codes of some used components are requested to be available from a repository under the terms of their primary licences, when applicable. This is not a blocking issue. What is important, legitimate and reasonable is to be authorized for distributing at least the **combined object code – the solution binaries**, even linked statically or imbedded and merged – as a whole, in a consistent package and under a single licence.

When can we ensure that is such distribution possible, and which covered components can be combined by developers or by contractors answering such calls?

As a preliminary remark, it is crucial that the contract awarding authority **specifies clearly its future licensing intentions**

¹⁰ Bastin F. & Laurent Ph. - “Study on the compatibility mechanism of the EUPL v1.0” p. 8. - IDABC 2006.
<http://ec.europa.eu/idabc/servlets/Doc?id=27472>

regarding the source and object code in the call for tenders and in the contract with the team of developers. If the tendering specifications simply request “open source” or “free software components” for example, it may be that the delivered application will be usable internally, but that any redistribution, even as executable binaries (i.e. to other organisations or through www.OSOR.eu) will be prohibited, due to copyleft licence incompatibilities. As reported in the IDABC Guideline on public procurement of Open Source Software¹¹, it is recommended that the specifications include the following provision (in case the EUPL is preferred):

“The contractor will ensure that <the contract awarding body> may distribute the software application as a whole, with its object code, source code and modifications, to any party of its choice, under terms and conditions of the <EUPL> licence.”

In order to enlarge the possibilities of using copyleft components, it is recommended to add:

The above provision does not make obstacle to the use of existing open source components, when the primary licence of these component “taken alone” (not included / linked in the software application) request the availability of their source code under their primary licence.

This second provision will make compliant all components which primary licence must stay applicable on source code, while binaries could be distributed under the EUPL.

Concretely, which FOSS components could be used?

If the licence selected for distributing the combined work is the EUPL, developers may use:

1. All the original code that they write (or they previously own, including their proprietary code) for developing the solution. Licensing this code under the EUPL presents the advantage of simplifying the legal framework for providers also: it establishes their right to reuse / improve the same code for other clients or applications¹².
2. Components licensed under a “permissive” licence. This group includes the Berkeley Software Distribution (BSD) licence, the X11 MIT license, the Apache 2.0 software license, etc.).
3. Components licensed under a so-called “weak copyleft” licence (i.e. the Gnu LGPLv2 or v3, the Mozilla Public license MPL 1.1 or the Artistic 2.0). These licences are recommended and frequently used for components (of libraries) that are aimed to be compiled or aggregated in combined works. These licences may impose specific conditions (as the free availability of the source code) and the OSI-approved EUPL is compliant with these conditions.
4. Component licensed under a copyleft licence when the licensor has published a “FOSS exception list”. When it is not done at licence level (we have seen that the EUPL has its own compatibility list), it can be nicely done by a licensor. For example, Oracle Corp., distributing the MySQL client libraries under the GPLv2, has included non-compatible copyleft licences

¹¹ Ghosh – R.A. and all. Guidelines on public procurement of Open Source software (June 2010 update)
<http://www.osor.eu/studies/expert-guidance/OSS-procurement-guideline%20-final.pdf> .

¹² People working in the software industry know how far it is sensitive to reuse developments that were done for (and paid by) a specific client. FOSS licensing solves this by establishing distribution rights for both licensors and recipients.

(including the EUPL and the GPLv3) in its FOSS exception list¹³.

5. Components licensed under some other copyleft licences, depending on their provisions and other legal considerations. As it is not possible to screen all existing licences, let's explore two groups: the licences that are already included in the EUPL compatibility list and other copyleft licences.

a) Licences those are included in the EUPL Compatibility list

- **ECLIPSE public license (EPL v 1.0)**

Section 3 A of EPL allows “contributors to distribute the Program in object code form under its own licence agreement”, provided that the new licence (The EUPL in our example) complies with the EPL v1.0 conditions, effectively disclaims warranties, exclude liability, does not imply additional obligations for the EPL contributors and states that the source code is available.

The provisions of the EUPL complies with the all the above points. A contributor (this includes developers elaborating a wider, combined solution) may therefore re-license the binaries generated from EPL code under the EUPL (while the EPL source code “taken alone” would have to remain under the EPL)¹⁴.

¹³ <http://www.mysql.com/about/legal/licensing/foss-exception/>

¹⁴ According to Mike Milinkovich (author of ECLIPSE), this feature allows the many commercial vendors who ship products based on Eclipse to offer their customers a single commercial licence for their products. Mike Milinkovich consider that this approach meets the requirements of European government agencies regarding the distribution as a whole” under the EUPL.

- **CPL v1.0**

This licence is identical to (and is now superseded by) the ECLIPSE (see the above section that applies to CPL as well).

- **OSL (v2 and v3)**

Provision 1 a) of OSL allows recipients “to reproduce the Original Work in copies, either alone or as part of a collective work”. The OSL establish a distinction between collective works (this notion can be assimilated to combined works) and other derivative works based on the original works. The OSL author, Laurence Rosen, confirmed that “the OSL currently allows OSL and ***any other*** components to be combined into a collective work, and the entire combination can be licensed under any licence. The OSL components, however, remain under the OSL”¹⁵The copyleft character of the OSL makes no obstacle to interoperability and a combined solution including OSL components can be distributed “as a whole” under the EUPL (same mechanism as for the Gnu LGPL).

- **CeCILL v2**

The French licence CeCILL (which has two authentic linguistic versions (English and French) and is used mostly in some French administrations) has an interoperability provision, but it is in direction of the Gnu GPL only (V2 or V3 depending on the case). The EUPL was not considered at the time of writing CeCILL. As long this is the case, contracting authorities should only authorize their developers and contractors to reuse CeCILL components in combined applications if they accept to distribute this work

¹⁵ Laurence Rosen, e-mail to the author – 10 May 2010.

under the CeCILL, GPLv2 or GPLv3 licences.

According to CeCILL Article 5.3.4:

The Licensee can include the Modified or unmodified Software in a code that is subject to the provisions of one of the versions of the GNU GPL, and distribute that entire code under the terms of the same version of the GNU GPL.”

- **GPLv2**

The Gnu GPLv2 has no interoperability provisions. Therefore, contracting authorities should not authorize their developers and contractors to make the integration of GPLv2 components in combined application that they intend to distribute under the EUPL, if this may produce a derivative work of this covered component.

However, the risk disappears if the component licensor inserts the EUPL in a FOSS exception list.

In all cases, the distribution of the larger work will also be possible under GPLv2, which is an alternative (accepted by the EUPL compatibility list). If such alternative is accepted by the licensor, any “GPL” (without version number¹⁶) or GPLv2 components will be legally compatible for building the solution.

¹⁶ GPL version nr. is often omitted by licensors, or they states “GPLv2 or later” making any GPL version applicable.

Discussion between the lawyer and the pragmatist

Pragmatist: What could be the consequence if a work integrating code covered by GPLv2 or CeCILL (these licences **are** in the EUPL compatibility list, but **are not interoperable**) is distributed under the EUPL?

Lawyer: Copyright is exclusive: once an author has chosen a strong copyleft licence, his/her willingness must be respected absolutely. The author has the right to oppose to any other, parallel or alternative distribution.

Pragmatist: Don't you believe that the advantages of interoperability strip by far a strict reading of the licences, and that when a copyleft licence is included in the EUPL downstream compatibility list (which is the case of the Gnu GPLv2) it produces automatically a "de facto" impact on upstream compatibility?

Lawyer: Could you explain?

Pragmatist: Imagine that an author or authority X distributes "as a whole" under the EUPL a CD with the binaries of a combined work including components covered by a copyleft licence A (i.e. GPLv2) that is in the compatibility list. In addition, this author complies with all EUPL requirements and source code is available / downloadable with the relevant primary licensing conditions (= GPLv2 for the relevant covered code, EUPL for other code). Could a person or organization Y sue the authority X for copyright violation?

Lawyer: Strictly legally, yes (no doubts)!

Pragmatist: But what would be the interest for doing so, because the source code is available under the "right" licence and the compatibility clause of the EUPL authorises immediately this person Y to distribute the solution as a whole (all binaries and the EUPL code as well) under the same licence A (GPLv2)? The developers' community working with licence A (GPLv2 in my example) will benefit from all the work done by X (including improvements to the component covered by licence A and other developments covered by the EUPL). Therefore the distribution of the combined solution under the EUPL will have positive impacts on free software improvement and shared usage.

Lawyer: So you think that the "downstream" compatibility of the EUPL has an impact on the "upstream" compatibility because it removes any interest in copyright litigation. My friend, you want to apply the old Roman law principle: "*Nemini actionem competere, nisi ei cuius interest*".

Pragmatist: Who! ... (long silence). I do assume so. I admit that it would be different if I redistribute the combined work under a different licence (not the EUPL but licence B or C or D assuming they are also in the compatibility list). In such case I would need to check compatibility with the licence A.

Lawyer: I fear that the original author may have a real interest to sue you if he estimates that on some points, the EUPL provides different or less explicit or less detailed protections than his compatible licence A (i.e. regarding specific cases of usage – like "tivoization", digital rights management, patents etc.).

Pragmatist: No, no, the EUPL will not deprive the users of a compatible licence from any protection, because the compatibility provisions of the EUPL states: "*Should the Licensee's obligations under the Compatible Licence conflict with his/her obligations under this Licence, the obligations of the Compatible Licence shall prevail.*" Therefore, no conflict may occur with a licence that is included in the EUPL compatibility list.

Lawyer: Uh? Give me some time to look at these EUPL provisions again...

The lawyer's point of view must prioritise a strict reading of the licences and the application of copyright law rather than the business advantages (i.e.: "no GPL components in a combined work distributed under the EUPL", except with a FOSS exception list).

Pragmatist's view is that The EUPL approach should be considered when reviewing other existing licences: facing licence proliferation, interoperability is a better answer than maintaining barriers between incompatible licences.

b) Copyleft licences that are not in the EUPL Compatibility list

In general, all components covered only by a **strong copyleft licence that is not included in the EUPL compatibility list**, should **not be integrated or statically linked** (in a way it creates a derivative) as a part of a combined work distributed under the EUPL.

It is important to remind that the simple “usage” of a FOSS component, even covered by a copyleft licence, does not produce a derivative according to the applicable copyright law. Aggregating, linking and combining may not produce any derivative.

This is – of course – to confirm or infirm by screening the provisions of the relevant licence (does it distinguish code and object, and allow at least to distribute binaries as a whole under the EUPL? In such case, the component would be compliant).

Asking to the licence author in case of doubt is recommended: licensors distributing under a copyleft licence may probably welcome the opportunity to add the EUPL in their FOSS exception list.

This is simpler than enlarging compatibility at licence level (i.e. maintaining a compatibility list in the licence). In the case of the EUPL, it has been recommended to keep the list of EUPL compatible licences as small as possible, and enlargement criteria were defined¹⁷:

1. the compatible 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.



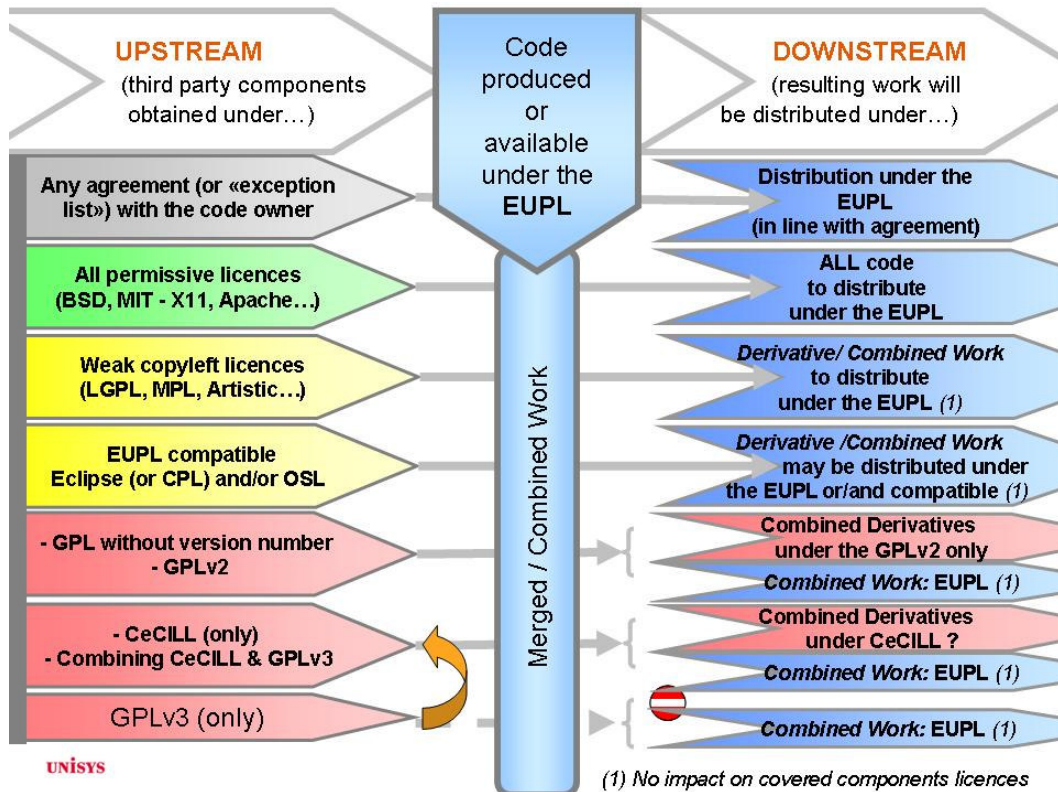
It would be possible and advisable to extend the EUPL compatibility list to new licences complying with the above

¹⁷ Bastin F. & Laurent Ph. - “Study on the compatibility mechanism of the EUPL v1.0”, p. 18 IDABC 2006
<http://ec.europa.eu/idabc/servlets/Doc?id=27472>.

criteria.¹⁸ The question is pertinent for the GPLv3, which was launched after the publication of the EUPL compatibility list in January 2007. The GPLv3 probably complies with criteria 1 & 2. Many projects refer to “GPL” without mentioning the version. V2 and V3 are very different and are not compatible: however, both GPLv2 (#9) and V3 (#14) states “If Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation”. Re-licensing EUPL work under the GPLv3 “after combining it with a CeCILL component” is possible but

Is not the most simple/elegant. An extension of the compatibility list is a formal modification of the EUPL licence (it would most probably be needed to create a EUPL v1.2), which requires substantial efforts due to the necessity to update all existing documents mentioning the version 1.1. It would also be opportune to collect the feedback of the current EUPL community before extending the list and to follow a new OSI approval process. In the meantime, any licensor using the EUPL may publish, as far as needed, a FOSS exception list including the GPLv3 (this exception being valid for software licensed by this licensor).

EUPL Interoperability summary



18

Conclusion

The EUPL, which allows for easy licensing of open source software under an EU legal framework and has official value in 22 languages, is one of the most interoperable licences. It is representative of a new category of “Interoperable copyleft” licenses. The EUPL ensures that the distributed software will not be “appropriated” exclusively by third parties, while respecting the primary licences of the used components: it is at the same time copyleft and interoperable.

The EUPL is particularly appropriate for licensing combined ICT solutions. In this framework, many FOSS components can be used and the solution can be distributed “as a whole” under the EUPL if the component is covered by a permissive licence, by a weaker copyleft licence, by some copyleft licences which are in the compatibility list and by some other copyleft licences, depending on a screening of their provisions.

In case of conflict with another copyleft licence (which is not the most frequent, as the simple fact of aggregating, and combining does not produce a derivative work), obtaining a FOSS exception for integrating this component in the EUPL larger work is the recommended way of obtaining legal interoperability

The most recommended propagation way for combined applications is the distribution of executable binaries under the EUPL, combined with the availability of the source code under the EUPL and/or under the primary licence of each component “taken alone” (even modified or linked) in so far this primary licence imposes to do so (knowing that all copyright and licensing attributions must be respected).