EUPL-compatible licenses

This is a global compatibility Matrix between all OSI-approved licenses and the EUPL
Author: Patrice-Emmanuel Schmitz / ©OSOR.eu (European Commission – 2011)

HTML: http://www.osor.eu/eupl/matrix-of-eupl-compatible-licenses
PDF: http://www.osor.eu/eupl/eupl-compatible-licenses 26-07-2011.pdf

Disclaimer:

The purpose of this Matrix is to clarify compatibility with the EUPL, seen as the legal possibility to distribute (under the OSI-approved EUPL) an application that incorporates, or links with, components covered by another Free/Open Source Software (F/OSS) license. Due to numerous possibilities, the Matrix does not cover all real situations (which are not always "clear-cut" and where more than two licenses could be relevant). Since European case law is generally missing, the matrix suggests reasonable guidance without providing a guarantee that this suggestion will always be followed by a judge, as the case may be.

Most F/OSS incompatibilities result from copyleft conflicts (when both licenses impose the reuse of the same license in case of redistribution). The aim and utility of copyleft is to protect the free software world against appropriation. In our vision, it should not make distribution difficult or legally impossible when the work is covered by different F/OSS copyleft licenses. In such a case, because of a weak interest, the risk of litigation is weak also, whilst not zero. Therefore, the Matrix is rather "liberal", based on factual license provisions and on comments provided by license stewards. We consider also that permission to distribute the executable binaries under a single license solves compatibility issues (as soon as distribution is legally possible, there is no need for a single license covering the source code). The Matrix is not influenced by ideology (telling the good and the ugly, urging people to use or to avoid specific licenses). On OSOR.eu user request, the OSOR team supports legal interoperability and the prevention of litigation by kindly requesting clarification or exceptions from the license stewards or from relevant licensors. We welcome comments (especially from license stewards) and relevant case law at: www.osor.eu/legal-questions-1/contact-legal.

Definitions:

- **Compatibility** has two ways:
  - **Upstream**, it allows you to merge a work covered by another F/OSS license into a larger work that you may distribute under the EUPL. This is the main scope of the Matrix below.
  - **Downstream**, it allows you to merge the work received under the EUPL into a larger work that you may distribute under a "compatible" license. This is the scope of the EUPL own compatibility list (EUPL #5 compatibility clause and Appendix that includes GPLv2, OSL, Eclipse, CPL and CeCILL as it is discussed at the end of this matrix).

- **F/OSS components** are those covered by an OSI-Approved license (other licenses are not considered in the Matrix).
- An **Application** is a "**larger work**" combining (by incorporation or by linking) F/OSS covered code or portions thereof with code governed by the EUPL.
  Outside, making legally possible a distribution of the larger application under the EUPL, there is no need and no interest for changing the license of any F/OSS component "taken alone" (even after correcting, modifying, translating it, etc.).
- **Incorporation** is merging all or part of the received component (in a copy-paste sense, when some original code is copied) with other software in the application, which becomes a derivative of the received components according to the applicable copyright law.
- **Linking** makes two components working in a single application without merging their source code. The question is: does it produce a derivative? See discussion on linking at the end of the Matrix.
  - **Static linking** combines components through compilation, copying them into the target application and producing a merged object file that is a stand-alone executable.
  - **Dynamic linking** combines components at the time the application is loaded (load time) or during execution (run time).

Matrix (OSI-approved licenses in alphabetical order)

License of existing F/OSS Component	Distribution of the larger application under the EUPL			Comment and references to F/OSS license provisions
	Incorporation	Static Link	Dynamic Link	
Academic Free License (AFL) v3.0	**OK**	**OK**	**OK**	#1c: you can distribute copies and derivative under any license that does not contradict the terms and conditions, including Licensor's reserved rights and remedies, in this Academic Free License. The OSI-approved EUPL is compliant with this requirement.
Adaptive Public License v1.0	**OK**	**OK**	**OK**	As the license is "adaptive", please read the terms carefully. According to # 3.7 you can combine the covered code with other components "not governed by the APL license provided the APL requirements are fulfilled for the covered portion of the larger work." The OSI-approved EUPL is compliant with this requirement.
Affero General Public License v1.0	**NO**	**NO?**	**OK**	The Affero General Public License is based on the GPL version 2, with one additional section covering Software as a Service (SaaS) on the distribution through web services or computer networks (section 2d). Regarding linking, see comments under the Gnu GPL and discussion at the end of this Matrix. Although legally not compatible (for copyleft reasons) the AGPL is very close to the EUPL (which covers also SaaS). Therefore – at least in case of linking – we recommend asking for an exception with the AGPL licensor. See notes under the GNU Affero General Public License version 3 which has succeeded to the AGPL.

Apache license V1.1, V2	**OK**	**OK**	**OK**	Nothing in the Apache V1.1 and 1.2 prohibits re-licensing of a larger work under the EUPL. They are indeed specific Apache v1.1 requirements (prohibiting some uses of the Apache brand), but these requirements are not "restrictions added by the EUPL licensor" (they are inherited from the original Apache licensing). These requirements are not impacting essential F/OSS freedoms. They are in line and compatible with article 5 of the EUPL.
Apple Public Source license APSL v2.0	**OK**	**OK**	**OK**	#4 Larger Works: You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In each such instance, You must make sure the requirements of this License are fulfilled for the Covered Code or any portion thereof. The OSI-approved EUPL is compliant with this requirement.
Artistic license (Perl foundation)	**OK**	**OK**	**OK**	In case of incorporation, verbatim copies (including bug fixes) must reproduce original copyright notices and associated disclaimers (#3). You can distribute under the EUPL a modified version (i.e. integrated/merged in a larger derivative work) provided that you document how it differs from the Standard Version received. The EUPL is a royalty-free license that permits recipients to receive the source code and to freely copy, modify and redistribute the modified version (requirement #4, ii). These conditions are inherited from the Artistic license and are not “restrictions added by the EUPL licensor”: they are compatible with the EUPL provisions.
Attribution Assurance License (AAL)	**OK**	**OK**	**OK**	This license is adapted from the original BSD license. See comments under BSD.
Boost Software License v1.0	**OK**	**OK**	**OK**	Nothing in this license restricts redistribution under any other license (free, as the EUPL, or even non-free).
BSD license (all versions)	**OK**	**OK**	**OK**	Nothing in the BSD license restricts redistribution under any other license (free, as the EUPL, or even non-free). BSD requests to retain copyright notices and restrict the use of the BSD licensor name to promote derivative products. These inherited conditions / restrictions are not “added” by the EUPL licensor and are in line with the EUPL requirements.
Computer Associate Trusted Open Source License CATOSL v1.1	**OK**	**OK**	**OK**	#3.4 (You) may create a Larger Work by combining the Program with other software code not governed by the terms of this License, and distribute the Larger Work as a single product. In such a case (...) make sure that the requirements of this License are fulfilled for the Program. The OSI-approved EUPL is compliant with this requirement.
Common Development and Distribution License CDDL v1.0	**OK**	**OK**	**OK**	#3.6 You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software. The OSI-approved EUPL is compliant with this requirement.

Common Public Attribution License CPAL v1.0	**OK**	**OK**	**OK**	#3.7 You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code. The OSI-approved EUPL is compliant with this requirement.
CUA Office Public License CUA-OPL v1.0	**OK**	**OK**	**OK**	#3.7 You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code. The OSI-approved EUPL is compliant with this requirement.
CeCILL v2.0	**NO** **(but you may distribute the larger work under CeCILL)**	**?**	**OK**	The CeCILL license is not OSI approved. We cover it here by exception because it is included in the EUPL "downstream" compatibility list (you can distribute under CeCILL a larger derivative work integrating components covered by the EUPL and by CeCILL). CeCILL v2.0 ignores reciprocity with the EUPL. Distribution under the EUPL would be possible only if the licensor has published a "FLOSS exception list" for distributing the larger derivative work under the EUPL. There are no examples of such exceptions (probably because no one requested it so far). As the EUPL has included CeCILL in its own downstream compatibility list, there would be a good chance to obtain such exception if requested, but no case was reported. CeCILL ignores linking. As CeCILL stewards are FSF followers, we could assume that they consider static linking in a similar way as for the GPL: covered by copyleft (without exceptions). The only CeCILL compatibility permission favors the GPL and says "include a code" or "include the software" (#5.3.4). Regarding static linking, this is depending on how far you create a derivative (see discussion at the end of this Matrix).
Common Public License (CPL)	**OK**	**OK**	**OK**	(See the Eclipse Public License, which has replaced the CPL).
Eclipse Public License (EPL) v1.0	**OK (object)** **You may also distribute the larger work under the EPL**	**OK**	**OK**	#3. Any person or entity (called "contributor") may distribute the Program in object code form under the EUPL ("its own license agreement"), because the EUPL complies with the requirements of #3b (disclaimer etc.). While the executable version of the application including EPL covered code will be distributed as a whole under the EUPL, it must be documented that the source code of components covered by the EPL will stay under this license. The EPL is also included in the EUPL downstream compatibility list (EUPL Appendix). Therefore the EUPL is compatible with the EPL: as far as needed, you may distribute under the EPL a larger derivative work integrating components covered by the EUPL and by the EPL

Educational Community License ECL v2.0	**OK**	**OK**	**OK**	Similar to the Apache 1.2. Nothing in the license prohibits re-licensing of a larger work under the EUPL. #4: you may provide ... different license terms ... for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License. The OSI-approved EUPL is compliant with this requirement. Concerning linking #1 clarifies that Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of (the covered software).
Eiffel Forum license v2	**OK**	**OK**	**OK**	This is a very permissive license. Similarities with BSD.
Entessa Public License v1.0	**OK**	**OK**	**OK**	This is a very permissive license. Similarities with BSD. License imposes copyright acknowledgements that are compatible with the EUPL.
EU Data Grid license	**OK**	**OK**	**OK**	Nothing in this license restricts redistribution of larger works under the EUPL. Copyright statement and the permission that "Installation, use, reproduction, display, modification and redistribution of this software, with or without modification, in source and binary forms, are permitted" must be reproduced. The OSI-approved EUPL is compliant with requirements.
European Union Public Licence (EUPL v1.0 & v1.1)	**OK**	**OK**	**OK**	Software that was licensed under EUPL v1.0 is automatically covered by the EUPL v1.1 (this "automatic update" was removed from the EUPL v1.1 and is not applicable anymore if the Commission publishes further versions of the EUPL).
Fair License	**OK**	**OK**	**OK**	The license instrument must be retained with the works, so that any entity that uses the works is notified of this instrument.
Frameworx Open License v1.0	**OK**	**OK**	**OK**	#3a: The other license conditions cannot be less favorable... #3c: They are specific copyright notices to respect. The EUPL is compliant with these requirements.
Gnu Affero Public License v3 (AGPLv3)	**NO**	**NO?**	**OK**	See comments related to the GPLv3. Requesting for an exception should be facilitated by the fact the EUPL covers "software as a service" (SaaS) like the AGPL.

Gnu GPLv2.0	**NO (exceptions exist)** **and you may distribute the larger work under GPLv2**	**NO?**	**OK**	Only if the licensor has published a "FLOSS exception list" for distributing the larger derivative work under other listed FLOSS licenses, and EUPL is included. See for example the MySQL FOSS exception list [1]. Exception will be applicable to all portions of the derivative work that are "independent" (not specifically derived from the program obtained under the GPLv2, which – taken alone - stays under its primary license). The GPLv2 is also included in the EUPL downstream compatibility list (EUPL Appendix). Therefore the EUPL is compatible with the GPLv2: you may distribute under GPLv2 a larger derivative work integrating components covered by the EUPL and by the GPLv2) Linking: FSF followers consider static linking as producing a derivative (this is not confirmed by case law, see "Linking discussion"). We assume that licensors using the GPL share this opinion, for extending the protection of the free software world against appropriation. However, there is no risk of appropriation when the larger work is licensed under the EUPL. We recommend asking for exception.

[1] http://www.mysql.com/about/legal/licensing/foss-exception/

Gnu GPLv3.0	**NO (exceptions exist)** **and it is legally possible to distribute the larger work under GPLv3 via the CeCILL roundabout**	**NO?**	**OK**	Only if the licensor has published a "FLOSS exception list" for distributing the larger work under other listed FLOSS licenses, and EUPL is included. See for example the Sencha exception list[2] where all identifiable sections of the larger work, which are not derived from the GPLv3 covered work, and which can reasonably be considered independent and separate works in themselves, may be distributed subject to the EUPL. The GPLv3 is NOT included in the EUPL downstream compatibility list (Appendix of the EUPL). However, it is "legally possible" (without prejudice of a need for it, or not!) to distribute a larger work integrating components covered by the EUPL and by the GPLv3. For this, you may use the "CeCILL roundabout": 1) publish your own specific contribution under CeCILL, or find convenient code covered by CeCILL; 2) combine with the EUPL component and publish the larger work under CeCILL; 3) add the needed GPLv3 components and publish the larger work under the GPLv3. This does not restrict EUPL licensors to provide a specific exception for integrating their own software in a larger work covered by the GPLv3. Linking: FSF followers consider static linking as producing a derivative (this is not confirmed by case law, see "Linking discussion"). We assume that licensors using the GPL share this opinion, for extending the protection of the free software world against appropriation. However, there is no risk of appropriation when the larger work is licensed under the EUPL. We recommend asking for exception.
Gnu LGPL v2.1	**OK (object)**	**OK**	**OK**	According to the LGPL v2.1, #6, you may produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications. These conditions (including the availability of the source code and all FLOSS freedoms) are fully implemented by the EUPL. While the larger derivative work will be distributed under EUPL, the source code of the used library 'taken alone" (modified or not) stays covered by the LGPL (this must be documented with prominent notices)

[2] http://www.sencha.com/legal/open-source-faq/open-source-license-exception-for-applications/

Gnu LGPL v3.0	**OK (object)**	**OK**	**OK**	According to the LGPL v3, #3, You may convey under the EUPL ("terms of your choice") the object code ("executable binaries") of an application that incorporates material from a header file that is part of the Library. While the larger work (binaries) can be distributed under EUPL, the source code of the used library 'taken alone" (modified or not) stays covered by the LGPL (this must be documented with prominent notices).
IBM Public License v1.0	**OK (object)**	**OK**	**OK**	#3: (You) may choose to distribute the Program in object code form under the EUPL (=own license agreement). The OSI-approved EUPL is compliant with the IPL requirements (i.e. it makes source code available and contains the requested disclaimers). As a consequence, make it clear that the source code of the IPL covered component will stay covered by the IPL.
IPA Font License (IPA)	**NO**	**?**	**OK**	This Japanese license is specific to "Digital Font Programs" (containing, or used to render or display fonts). Derived programs (any modification to covered code) must be licensed under IPA (#3.1.3). Regarding static linking, this is depending on how far you create a derivative (see discussion at the end of this Matrix).
ISC License (ISC)	**OK**	**OK**	**OK**	Nothing in the ISC license (similar to BSD) restricts redistribution under any other license (free, as the EUPL, or even non-free). ISC requests to retain copyright notices. These inherited conditions / restrictions are not "added" by the EUPL licensor and are in line with the EUPL requirements.
LaTeX Project Public License (LPPL 1.3c)	**OK**	**OK**	**OK**	The LaTeX accepts re-distribution under another license: "if for any part of your work you want or need to use *distribution* conditions that differ significantly from those in this license, then do not refer to this license... but, instead, distribute your work under a different license.
Lucent Public License v1.02	**OK**	**OK**	**OK**	#3 you may…"distribute the program under your own license agreement" providing it complies with F/OSS conditions and disclaimers. The OSI-approved EUPL is compliant with these requirements.
Microsoft Public License (Ms-PL)	**OK (object)**	**OK**	**OK**	#3 D: If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license. The OSI-approved EUPL is compliant.
Microsoft Reciprocal License (MS-RL)	**OK (object)**	**OK**	**OK**	#3 E: If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license. The OSI-approved EUPL is compliant.

MirOS License	**OK**	**OK**	**OK**	Nothing in the MirOS license (similar to BSD) restricts redistribution under any other license (free, as the EUPL, or even non-free). The license requests retaining the copyright notices and disclaimer. These inherited conditions / restrictions are not “added” by the EUPL licensor and are in line with the EUPL requirements.
MIT License	**OK**	**OK**	**OK**	Nothing in the MIT license restricts redistribution under any other license (free, as the EUPL, or even non-free). The license requests retaining the copyright notices. This is in line with the EUPL requirements.
Motosoto License	**OK**	**OK**	**OK**	#4 g. You may “distribute Derivative Works as products under any other license you select, with the proviso that the requirements of this License are fulfilled for those portions of the Derivative Works that consist of the Licensed Product or any Modifications thereto.” These inherited conditions / restrictions are not “added” by the EUPL licensor and are in line with the EUPL requirements.
Mozilla Public License 1.1	**OK**	**OK**	**OK**	#1.8. A larger work combines the MPL Covered Code (or portions thereof) with other code not governed by the MPL (i.e. code governed by the EUPL). #3.7. You may distribute a Larger Work as a single product under another license (i.e. the EUPL) provided the requirements of the MPL are fulfilled for the Covered Code. Please document (retaining copyright notices) that the MPL covered code (or portions) stay covered by the MPL. As an OSI-Approved license, the EUPL grants to recipients the same freedoms as the MPL.
Multics License	**OK**	**OK**	**OK**	Nothing in the license restricts redistribution under any other license (free, as the EUPL, or even non-free). The license requests retaining the copyright notices and historical background. This is in line with the EUPL requirements.
NASA Open Source Agreement (NOSA v1.3)	**OK**	**OK**	**OK**	#1 Larger Work" means computer software that combines Subject Software, or portions thereof, with software separate from the Subject Software that is not governed by the terms of this Agreement. #3 I: Larger works may be distributed as a single product “not governed by the terms of this agreement” but “Subject Software, or portions thereof, included in the Larger Work is subject to this Agreement.” These inherited conditions / restrictions are not “added” by the EUPL licensor and are in line with the EUPL requirements.
NTP License	**OK**	**OK**	**OK**	Nothing in the license restricts redistribution under any other license (free, as the EUPL, or even non-free). The license requests retaining the copyright notices, disclaimer and restricts the use of trademarked names for advertising. This is in line with the EUPL requirements.

Naumen Public License	OK	OK	OK	This license is made for a specific licensor (Naumen). Nothing in the license restricts redistribution under any other license (free, as the EUPL, or even non-free). The license requests retaining the copyright notices, disclaimer and restricts the use of trademarked names. This is in line with the EUPL requirements.
Nethack General Public License (NGPL)	NO	?	OK	#2b implies that another license can be used: "...work... that in whole or in part contains or is a derivative of NetHack or any part thereof, to be licensed at no charge to all third parties on terms identical to those contained in this License Agreement (except that you may choose to grant more extensive warranty protection to some or all third parties, at your option)". However the word "identical" seems much stronger than "compliant", which is used by many other licenses. Regarding static linking, this is depending on how far you create a derivative (see discussion at the end of this Matrix).
Nokia Open Source License	OK	OK	OK	#1 c. A larger work "combines Covered Software or portions thereof with code not governed by the terms of this License" #3.7. You may distribute a larger work as a single product "not governed by the terms of this License", provided "the requirements of this License are fulfilled for the Covered Software". The OSI-approved EUPL is compliant with requirements.
Non-Profit Open Software License 3.0	NO	?	OK	See comments related to the OSL: without exception, derivatives must be licensed under NPOSL.
OCLC Research Public License 2.0	NO	?	OK	#2: A. "Combined Work" results from combining and integrating all or parts of the Program with other code. A Combined Work may be thought of as having multiple parents or being result of multiple lines of code development. #3 D. "Combined Works are subject to the terms of this license". At the exception of "Aggregates" (presence without integration on the same medium) the OCLC is not interoperable with other F/OSS licenses. Regarding static linking, this is depending on how far you create a derivative (see discussion at the end of this Matrix).
Open Font License 1.1 (OFL)	OK (object)	OK	OK	The scope of the license is limited to the font software and specific derivatives. The Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the copyright notice and this license. Our interpretation is that a larger work can be distributed under the EUPL, while the specific font software source (even modified) will stay covered by the OFL.
Open Group Test Suite License (OGTSL)	OK (object)	OK	OK	#4 You may distribute the work in object code, provided that you ... (rename non-standard executables). These inherited conditions / restrictions are not "added" by the EUPL licensor and are in line with the EUPL requirements.

Open Software License OSL v3.0	**NO** **but you may distribute the larger work under the OSL**	**OK**	**OK**	The OSL is copyleft (#1.c. states "You may distribute derivative Works under the OSL"). Therefore it is implicitly not allowed to distribute larger derivative works under other licenses (including EUPL, GPL etc.). The OSL is included in the EUPL downstream compatibility list (EUPL Appendix). Therefore the EUPL is compatible with the OSL (you may distribute under the OSL a larger derivative work integrating components covered by the EUPL and by the OSL) As the EUPL has included the OSL in its own compatibility list, there would be no reasons for OSL licensors to refuse an exception, but this must be formally requested. There are no examples of such exceptions (probably because no one requested it so far). Linking: according to Lawrence Rosen (OSL steward) linking components (especially if covered by various OSI approved licenses, considering the market factor impact) does not produce a derivative. We assume that OSL licensors share this view.
PHP License 3.0	**OK (object)**	**OK**	**OK**	Nothing in the license restricts the distribution of larger works in binary form under another license (as the EUPL). Redistributions must retain the copyright notice, disclaimer and conditions (including restrictions on the use of "PHP" in the name of derived products). These inherited conditions / restrictions are not "added" by the EUPL licensor and are in line with the EUPL requirements.
PostgreSQL license	**OK**	**OK**	**OK**	Nothing in the license restricts the distribution of larger works under another license (as the EUPL). Redistributions must retain the copyright notice and disclaimer. These inherited conditions / restrictions are not "added" by the EUPL licensor and are in line with the EUPL requirements
Python 2.0	**OK**	**OK**	**OK**	Nothing in the license restricts the distribution of larger works under another license (as the EUPL). Specific changes in Python software must be documented and redistributions must retain the copyright notice and disclaimer. These inherited conditions / restrictions are not "added" by the EUPL licensor and are in line with the EUPL requirements.
Q Public License (QPL v1.0)	**OK**	**OK**	**OK**	Nothing in the license, which is considered as "non-copyleft" restricts the distribution of larger works under another license (as the EUPL). They are specific requirements (to distribute specific modifications of the covered code through patches). #6 adds conditions in case "applications" links with the covered software (availability of source code, free redistribution…) These inherited conditions / restrictions are not "added" by the EUPL licensor and are in line with the EUPL requirements.

RealNetworks Public Source License (RPSL v1.0)	**OK**	**OK**	**OK**	# 1.6 Derivative Work also includes any work which combines any portion of Covered Code or Modifications with code not otherwise governed by the terms of this License. #4 (You may) “distribute the Derivative Work as an integrated product. In each such instance, You must make sure the requirements of this License are fulfilled for the Covered Code or any portion thereof, including all Modifications.” The OSI-approved EUPL is compliant with this requirement. In addition, the covered source code “taken alone” will stay covered by RSPL. RPSL inherited conditions / restrictions are not “added” by the EUPL licensor and are in line with the EUPL requirements.
Reciprocal Public License RPL v1.5	**NO**	**?**	**OK**	License text states: # 6.0 ...You hereby agree that any... Derivative Works, (meaning “defined under U.S. copyright law”) ... are governed by the terms of this License... (and) must be Deployed (meaning distributed) under the terms of this License or a future version of this License... Regarding static linking, this is depending on how far you create a derivative (see discussion at the end of this Matrix).
Ricoh Source Code Public License (RSCPL v1.0)	**OK**	**OK**	**OK**	#1.6. "Larger Work" means a work which combines Governed Code or portions thereof with code not governed by the terms of this License. #3.7. “You may create a Larger Work by combining Governed Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Governed Code.” The OSI-approved EUPL is compliant with requirements.
Simple Public License v2.0	**OK**	**OK**	**OK**	SimPL is a plain language implementation of GPL 2.0. However it allows relicensing of derived works under “substantially similar terms (such as GPL 2.0), without adding further restrictions to the rights provided”. In our interpretation, “such as” does not mean “only” and the EUPL complies with the requirements.
Sleepycat License	**OK**	**OK**	**OK**	Nothing in this license restricts redistribution under any other license (provide it is F/OSS, as the EUPL). The license requests retaining copyright notices and disclaimer, and to document how to obtain the freely redistributable source code. The OSI-approved EUPL complies with requirements.
Sun Public License	**OK**	**OK**	**OK**	#1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License. #3.7. “You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.” The OSI-approved EUPL is compliant with requirements.

Sybase Open Watcom Public License (Watcom v1.0)	**OK**	**OK**	**OK**	#1.5 "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License. #4 “You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In each such instance, You must make sure the requirements of this License are fulfilled for the Covered Code or any portion thereof”. The OSI-approved EUPL is compliant with requirements.
University of Illinois / NCSA Open Source License	**OK**	**OK**	**OK**	Nothing in this license restricts redistribution under any other license (free, as the EUPL, or even non-free). The license requests retaining the copyright notices, disclaimer and restricts the use of licensor name. The OSI-approved EUPL complies with requirements.
Vovida Software License (VSL v1.0)	**OK**	**OK**	**OK**	This license is for use by a specific licensor (Vovida). Nothing in this license restricts redistribution under any other license (free, as the EUPL, or even non-free). The license requests retaining the copyright notices and disclaimer. It restricts the use of licensor and product names. The OSI-approved EUPL complies with requirements.
W3C license	**OK**	**OK**	**OK**	This license is for use by a specific licensor (W3C). Nothing in this license restricts redistribution under any other license (free, as the EUPL, or even non-free). The license requests retaining the copyright notices and disclaimer, and to notify changes to W3C. These inherited conditions are not “added” by the EUPL licensor and are in line with the EUPL requirements.
X.Net license (Xnet)	**OK**	**OK**	**OK**	Nothing in this license restricts redistribution under any other license (free, as the EUPL, or even non-free). The license requests retaining the copyright notices “in all copies or substantial portions of the software”. These inherited conditions are not “added” by the EUPL licensor and are in line with the EUPL requirements.
Zope Public License (ZPL v2.0)	**OK**	**OK**	**OK**	Nothing in this license restricts redistribution under any other license (free, as the EUPL, or even non-free). The license requests retaining the copyright notices and restricts the use of service marks or trademarks of Zope. These inherited conditions are not “added” by the EUPL licensor and are in line with the EUPL requirements.
Zlib/libpng License	**OK**	**OK**	**OK**	Nothing in this license restricts redistribution under any other license (free, as the EUPL, or even non-free). The license requests retaining the copyright notices. This condition is in line with the EUPL requirements.

OK= Allowed
OK (Object) = Distribution of binaries of the larger work “as a single product” under the EUPL is allowed
NO = Not allowed (however, licensor owning full copyright may provide exceptions)
? = Uncertainties (and no exceptions exist so far)

Discussion on “Linking”

A combined application or work is often implemented by linking various components. From the end user point of view, the application may look like a single program (it may have a single name, unique interface, documentation etc.). From the developer point of view, each component is a separate program, possibly obtained under various primary licenses.

In case these various licenses are “free / open source” and “copyleft”, what kind of linking could you implement to be legally authorised to distribute the application, under a single license or – possibly – under several licenses?

There are two main cases of linking:

- Static linking combines components through compilation, copying them into the target application and producing a merged object file that is a stand-alone executable.
- Dynamic linking uses components at the time the application is loaded (load time) or during execution (run time).

In the above Matrix, we assume that both linking are permitted in two cases: when incorporation (merging codes in a “cut-paste” sense) is authorised and when the distribution of object code under the EUPL is authorised (because this covers static linking as well).

From the legal (copyright) point of view, the question of linking is similar to the incorporation discussion: does linking produce a derivative of the used components, or not?

- If the answer is “NO”, each part of the application can be licensed under its primary license: You may declare “My application is licensed to you under the EUPL, except component X that is licensed under GPL, component Y that is licensed under the EPL, component Z under the MPL etc...

- If the answer is “YES”, the distribution of the new derivative could be legally impossible (under any license) as soon a copyleft conflict exists. Such prohibition may be considered as beneficial (or protective) for the free software world in case the licensor planned to use a proprietary license. On the contrary, the prohibition may look cumbersome and fussy when all relevant licenses are OSI-approved (providing the same freedoms) and copyleft (protecting against appropriation).

In Europe there is no case law solving this question. In absence of case law, we have to be careful. We may assume that a judge will interpret the copyright law with more flexibility (no derivative created by linking) in the case of “copyleft conflict” between two OSI approved licenses, and more strictly if a proprietary license is used (creation of a derivative in such case, because in such litigation the "market-based" factors are looking more important than the pure linking technique) – but this is a pure assumption.

The recent GPLv3 looks receptive to these "market-based" factors when a compilation is not used to limit users' freedom, at least in the specific case of aggregates[3]. We estimate that the requirement that independent sources (compiled together) should not form a larger program is to understand from developers' point of view: no portion of the code is copied into a larger program (on the contrary, due to some interface layer, end users may perceive the independent components as a single application).

Instead of European and US case law[4], we may deal with the main F/OSS opinion makers:

- Lawrence Rosen (IP professor and OSI general counsel) says that the method of linking is mostly irrelevant to the question about whether a piece of software is or is not a derivative work. For him, a derivative is made only in the case of incorporation [in a copy-paste sense] of original code, or in the case of modification, translation or other change in any way for creating the new program.

- Other IP specialists believe that static linking may produce derivative works, while dynamic linking may not, but the question is not "clear-cut". For example, some dynamically linked Linux kernel drivers are distributed under proprietary licenses, and the Linux author (L. Torvalds) agrees that such dynamic linking can create derived works in specific circumstances.

- The Free Software Foundation and their followers, in their desire to extend the free-software world by giving it more tools than the proprietary world, are the most assertive: static linking creates derivatives and executables which uses a dynamically-linked library may also be derivatives, except when separate programs just "communicate" with one another[5]. Does this interest for "protecting and extending free software" still exist when the other license is the EUPL (which is a free, copyleft software license)? The FSF could answer positively, because - at the contrary of the OSI - it has a strong GPL centric policy, arguing that for protecting free software there is little salvation outside their own GPLs (in fact, the GPLv3 and AGPLv3).

We made the assumption that, by selecting a Gnu license, licensors follow the FSF position and want to consider that most cases of static linking create a derivative. In all other cases, the proposed Matrix is based on a "liberal" assumption, in consideration of the common "market approach" shared by all OSI approved licenses, especially when these licenses are copyleft.

---

[3] #5 (in fine): "A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate."

[4] In the *Galoob v. Nintendo* an US court of appeal defined a derivative work as having "'form' or permanence" and noted that "the infringing work must incorporate a portion of the copyrighted work in some form", but there have been no clear court decisions to resolve the case of static/dynamic linking making a derivative.

[5] This is the reason why the FSF created the LGPL (which is nearly identical to the GPL) for adding the permission to allow linking for the purposes of "using the licensed library".

The EUPL "Downstream" compatibility list

The EUPL v1.1 downstream compatibility list was established for the EUPL v1.0 (January 2007). It was drafted based on a September 2006 study of the CRID (Research Centre in Computers and Law – FUNDP Namur)[6]

The purpose of the list is to allow mergers between EUPLed code and other copylefted code (when license terms impose to redistribute any larger, merged work under the same "inherited" licence, what becomes legally impossible in case of copyleft conflicts). Therefore the chosen licences had to be in any case copyleft, and "the copyleft effect of the elected licences should be similar to the EUPL's copyleft and should fulfill the same functions", the authors said [7].

The study proposed the following list:

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

The presence of the strong copyleft (OSI-approved) GPL and OSL makes no discussion. The presence of non-OSI-approved CeCILL was interesting because supported by three important French research centres, strong copyleft and – by the way – permitting re-licensing under all GPL versions (including the GPLv3 that was published in June 2007). This makes the EUPL compatible, directly or indirectly, with all GPL versions.

What looks more questionable is the presence of Eclipse (and of the CPL, but we may forget this one that has now been superseded by the EPL), because according to the Free Software Foundation, the EPL only provides a "weaker copyleft".

However "weaker" does not mean "weak! When analyzing copyleft, the study made three categories[8]:

- **"Weak copyleft"** when the licence restricts the effect on a file basis (this is the case of the Mozilla Public License and the Common Development and Distribution License).
- **"Source only copyleft"** when the licence imposes 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 EPL)
- "**Strong copyleft**" when the licence does not restrict the copyleft effect to modifications on a file basis, and the executable is to be considered as a derivative work.

Therefore Bastin and Laurent, while not categorizing the EPL copyleft as "Strong" considered that it was "similar" to the EUPL copyleft on the source code, which is by far the important thing for open source

[6] Fabian BASTIN (CERFACS) and Philippe LAURENT (CRID) Study of the compatibility mechanism of the EUPL v. 1.0 - September 2006 - http://ec.europa.eu/idabc/servlets/Doc3ef5.pdf?id=27472
[7] Op. Cit., p.17
[8] Op. Cit., p. 7

licensors because it effectively prohibits source code appropriation. For this reason other prominent OSS analyst categorize the EPL as “Strong Copyleft”[9]

In January 2007, the European Commission published the EUPL and its compatibility list based on their study, and did not change the list when publishing the EUPL v1.1 in 2009.

[9] For example the German license center ifrOSS lists the Eclipse Public License in the category “Other Licenses with strong Copyleft Effect”. See http://ifross.org/en/license-center