



Ministerie van Volksgezondheid,
Welzijn en Sport

Open Source Licenses: An In-Depth Analysis

Maurice Hendriks

In cooperation with all contributors

Gepubliceerd op 23 januari 2026

Creative Commons Attribution 4.0 International



Content

Introduction	2
The Dutch Copyright Law Exception	3
General License Types	4
Permissive	4
Reciprocal / Weak Copyleft	4
Restricted / Strong Copyleft	5
Sidenotes	5
Public Domain	6
An In-Depth Analysis	7
Permissive	8
Weak Copyleft	17
Strong Copyleft	26
OSPO Reviewed License Checklist	31
Recommendations	31
💡 Strong copyleft and EU-law	31
Contributors	32

The use of GenAI

Generative AI (GenAI) has been used in the preparation of this document in four ways:

- GenAI reviewed the texts for flow, language, and spelling.
- GenAI was asked to provide additional suggestions for all texts.
- GenAI was consulted for brainstorming on specific points.
- GenAI was used for the Dutch to English translations.

Before seeking input from GenAI, all texts are first written by the lead author or contributors themselves. This ensures that no information is included that the author(s) lack background knowledge of or whose implications are unclear. Writing the initial drafts manually also helps make the author's intent clear to GenAI.

Furthermore, this document is being developed publicly. The required knowledge on open source, legislation, and government policy is generally non-confidential and freely reusable.

Where GenAI was consulted on specific texts, an explanation is provided in the Rationale.

This approach ensures that the use of GenAI is responsible and considered a valuable contribution to developing the most practical tool possible.

Introduction

According to copyright law, the creator of a creative work automatically owns the intellectual property rights (IP-rights) to that work—unless, explicit agreements state otherwise. Source code is also considered a creative work, so it's essential to properly document who is legally the owner of the delivered source code. Only the IP-right holder can decide how (and if) the source code may be published.

Some of the limitations copyright law imposes on the work can be loosened by providing a license—under specific terms and conditions—to individuals or groups. By doing so, others can obtain the rights to use, modify or redistribute it. An open source license (OSS) is a standardized contract that establishes these duties and rights. By using an OSS-license, legal ambiguities are avoided, and they make explicit what users can or cannot do with the source code. Moreover, by using an OSS-license, IP-right holders can express the core values of

open source set forth by the Open Source Initiative.

- The right of freely distribute the software
- The freedom to modify or create derivative works from it.
- The exclusion of responsibility and liability for the misuse of the source.
- The protection of original authors' integrity.
- The protection of digital rights.

An OSS-license manages (to a certain extent) various aspects of IP-rights and liability. Without attaching an open source license to source code, the code is not legally considered open source. Openly published source code can still be studied, but cannot be distributed or adapted without the explicit permission of the IP-right holder.

The goal and effect of an OSS-license is to make software freely available to everyone, enabling others to build upon it, fix bugs, add new functionality, or reuse parts in different applications.

The Dutch Copyright Law Exception

As described above, source code published without an open source license cannot be reused by others. An important exception is found in Dutch copyright law Article 15b, which specifically covers publications by the Dutch government. It states that published government works can be reused and redistributed by others unless specific reservations are made. Without such a reservation, the reuse of government-published works is not considered copyright infringement.

Although the government retains the IP rights, these rights cannot be enforced without adding a reservation. When necessary, such a reservation can be as simple as this:

```
/*  
  All rights reserved  
  Copyright Ministry of Health, Welfare and Sport  
*/
```

It is recommended to place this reservation in the header of each source code file, together with the copyright notice. This way, reuse of the source code without permission is prohibited.

For Dutch government agencies, using an open source license does not broaden the usage rights—as it does for other authors—but instead imposes a (limited) restriction on reuse. The

license specifies the terms and conditions under which others can reuse, modify, or redistribute the works.

General License Types

Open source licenses mainly differ in the degree of reciprocity they require—a principle known as copyleft. This term is intentionally chosen as the opposite of traditional, restrictive copyright. The type of reciprocity determines the obligations for reuse, redistribution, and adaptation of the source code. There are three well-known categories of open source licenses.

Permissive

Permissive licenses impose only a few requirements on the reuse of source code. You may freely modify the source code, use it, and integrate it into proprietary software without the obligation to publish your changes as open source. Often, there is a requirement to include a reference to the original author and a copy of the original OSS license.

Some licenses, such as Apache 2.0, also require you to document all changes made, while others, like MIT-0, do not.

Well-known examples of permissive licenses include MIT, MIT-0, BSD, and Apache 2.0.

Proprietary software

Proprietary software is software for which the ownership rights lie with the creator, publisher, or another rights holder. This party determines what users may and may not do with the software. Usually, it is not permitted to modify, copy, or freely distribute the software. In some cases—such as software protected by a strict user agreement (EULA) or by patents—users may only be allowed to use the software in specific ways. Source

Reciprocal / Weak Copyleft

These licenses impose stronger demands: if you change the source code, you are obligated to published the changes under the same open source license. The obligation only takes effect once the modified version, or a product derived from it, is distributed to third parties outside

the organization. Only the changed components themselves fall under these obligations, not the whole software solution in which the components are integrated.

This makes weak copyleft licenses useable in both open as closed source software, as long as the open source components themselves follow the terms and conditions of the respective license.

Well-known examples of weak-copyleft licenses include Mozilla Public License (MPL v2.0), European Union Public License (EUPL v1.2).

Restricted / Strong Copyleft

Strong copyleft licenses add an additional inheritance effect. This means that when you use components licensed under such terms (e.g., GPLv3) in your software, the entire software must be published under the same license.

For example, suppose you are using three components with different open source licenses: A: MIT license B: EUPL license C: GPLv3 license

The combined software (A + B + C) must be published under the GPLv3 license. Components A and B retain their original licenses and can be reused independently under those licenses, but the final combined software must comply with the GPLv3.

Therefore, strong copyleft licenses are incompatible with closed source software.

A well-known example of a strong copyleft license is the GNU General Public License v3 (GPLv3).

A notable derivative is the Lesser GPL (LGPLv3), which is less strict. When you use an LGPL-licensed component as a separate library (soft-linked), the rest of the software does not need to be published under the LGPLv3.

Another derivative is the Affero GPL v3 (AGPLv3), which extends the obligations of the GPLv3 to software offered as a service (SaaS).

Sidenotes

An OSS license in the Restricted / Strong Copyleft category requires more legal and practical knowledge to apply correctly. When using source code under this type of license, you must



comply with stricter conditions. Failure to properly adhere to these conditions puts you at risk of being challenged or even sued by the original authors for copyright infringement.

Applying an OSS license also means that, as a creator, you relinquish control over how your source code may be used in the future. Your code could just as well contribute to the development of life-saving medicines as to the construction of weapons systems. OSS licenses do not make distinctions based on moral or ethical use.

It is also important to understand that an OSS license does not impose an obligation to publish. What the license enforces is the right to free distribution—but this right only comes into effect once the software or source code is actually distributed. As long as the software is used exclusively within an organization or a closed collaboration (such as a foundation) and is not actively shared with third parties, there is legally no distribution. This makes it possible to collaborate internally on OSS projects without the obligation to make the code publicly available.

However, once the source code or final product is distributed outside the organization, every contributor gains the right to further share that version of the code under the terms of the OSS license.

Conversely, source code can be made public without an OSS license attached. In that case, copyright remains fully in effect (except for government works). The code may be viewed but cannot be used, modified, or distributed without permission. Public disclosure in this case means transparency only—not reuse.

Public Domain

As described earlier, an open source license lifts many of the restrictions that normally arise from copyright law (except in the case of government works). This is done in the form of a user license. It is important to note that the open source license does not repeal copyright itself—copyright remains fully in force. The creator remains the legal owner but grants others certain usage rights under specific conditions.

If you want to completely waive your copyright, this can only be done by dedicating the work to the public domain. In that case, the source code is no longer legally owned by anyone and effectively belongs to everyone. There are then no restrictions on use, modification, or distribution, because no one has the legal right to challenge these actions.



Of all open source licenses, the BSD-0 license comes closest to the public domain. This extremely permissive license does not even require attribution to the author. However, formally, the code still remains under copyright in this case.

It is good to know that not all countries legally allow the waiver of copyright on a work. In practice, this rarely causes problems as long as the original author does not assert their rights. Nevertheless, it is important to be aware of the legal context, especially if you work with or publish under a license that implies a waiver of rights.

In some situations, it may be safer to release source code under an explicit open source license. This way, clearer agreements exist regarding use, reuse, and liability—which often provides more legal certainty than relying on the public domain.

An In-Depth Analysis

The classification into permissive, weak copyleft, and strong copyleft is a greatly simplified representation of the differences between license types. To apply licenses correctly, it is important to have a more detailed understanding of the differences between the main licenses or at least to understand what specific licenses themselves entail.

However, before conducting a thorough analysis, there are some principles to keep in mind when choosing a license:

- If the source code to be distributed is intended to be integrated into a large number of works and requires no maintenance, or if it can be given to anyone to be integrated even into programs that will become the exclusive property of third parties who can resell them under their own terms, a permissive license is the best choice. MIT or BSD licenses are preferable to putting the code in the public domain.
- If the source code to be distributed demonstrates specific expertise and you want to know about and benefit from any improvements made by third parties, you should choose a copyleft license. This is particularly justified for source code funded by public money, to avoid having to purchase improved versions of the program.
- Most licenses do not consider remote access (SaaS or Software as a service) to be a distribution, even though this mode of distribution is becoming very common.

The different degrees of copyleft will be compared below in the in-depth analysis.

Permissive

The MIT, BSD*, and Apache v2.0 licenses are permissive open-source licenses without strong copyleft effects, but they differ in scope, specific rights, and protections. The MIT License originates from the Massachusetts Institute of Technology and is known for its simplicity and flexible, form-free application. The BSD* license descends from the Berkeley Software Distribution and applies strictly to software, imposing as few obligations as possible. The Apache v2.0 license was developed by the Apache Software Foundation and, in addition to permissive conditions, offers extra legal assurances such as explicit patent protection and rules for contributions and sublicensing.

Feature	MIT	BSD*	Apache v2.0
Scope	Software + accompanying documentation as a whole	Software only (source and binary form)	Software, documentation, and configuration independently
Contributions & sublicensing	Sublicensing explicitly allowed; relicensing not	Sublicensing implicit; relicensing not	Sublicensing explicit; contributions automatically under Apache v2.0; relicensing not
Patent protection	None	None	Royalty-free patent license for contributions
SaaS	SaaS not considered distribution	SaaS not considered distribution	SaaS not considered distribution
Liability	As-is	As-is	As-is, with nuance for local law and third-party commercial warranties

Feature	MIT	BSD*	Apache v2.0
Characteristic	Maximum simplicity and freedom	Minimal coverage, strictly limited	Comprehensive protection and legal certainty

The MIT License is the simplest and most flexible: it covers software and its accompanying documentation as a whole, explicitly allows sublicensing, and imposes few additional obligations. BSD* is more limited and applies only to software; documentation and configuration are excluded, and sublicensing is only implicitly covered. Apache v2.0 is the most extensive: it covers software, documentation, and configuration independently, automatically licenses contributions under the same terms, and provides explicit protection against patent claims.

What all three have in common is that SaaS (Software as a Service) is not considered distribution, and relicensing of the original work is not allowed. Liability is limited in all three by an as-is clause, with Apache v2.0 allowing some flexibility for local legislation and third-party commercial warranties.

In short: MIT for maximum simplicity and freedom, BSD* for minimal software-only coverage, and Apache v2.0 for legal certainty, contribution protection, and broad file-type coverage.

The BSD Variants

There are several variants of the BSD license. The differences mainly concern how attribution of authors and copyright is handled. In all other respects, the BSD variants are the same. That's why in comparisons, the BSD license is referred to as BSD*.

Variant	Additional conditions / notes
BSD-0	Does not require any form of attribution.
BSD-1	Requires preservation of the original authors, copyright notices, and license terms in the source code.

Variant	Additional conditions / notes
BSD-2	Same as BSD-1, but adds that this inclusion also applies to the final software product.
BSD-3	Same as BSD-2, but adds that the names of authors and contributors may not be used in endorsements or promotional expressions without explicit permission.
BSD-4	Same as BSD-3, but with the literal addition: This product includes software developed by [the organization].

Scope

MIT	BSD*	Apache v2.0
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software") [...]	Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met [...]	"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
The MIT License applies to software and its accompanying documentation; standalone documentation or configuration is not covered.	The BSD* License applies only to software in source and binary form.	The Apache v2.0 License allows each component (software, documentation, configuration) to be licensed and distributed independently.

The BSD* license has the narrowest scope—it applies exclusively to software, excluding documentation and configuration. The MIT license is broader than BSD*, as it covers both software and its accompanying documentation, though only when part of the software package;

standalone documentation or configuration are still excluded. The Apache v2.0 license has the broadest scope. It encompasses software, documentation, and configuration, all of which can be used and distributed independently. This makes Apache 2.0 more flexible for projects that contain multiple types of files.

Degree of Copyleft

MIT	BSD*	Apache v2.0
[...] to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions [...]	Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met [...]	“Work” shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
Distribution of the complete bundle, whether source code, documentation, or configuration.	Distribution as source code or binaries.	Distribution as object form or source code.

The MIT license is the most extensive and form-agnostic in this respect; it makes no distinction between distributing software as source code, standalone binaries, or as a complete package including documentation and configuration. It is intentionally phrased broadly so that all forms of distribution are covered. In contrast, the BSD* and Apache v2.0 licenses explicitly define the forms in which software may be distributed, limiting them to source code and binaries. Apache v2.0 further adds object form as an allowed distribution format.

Distribution and SaaS

The MIT, BSD*, and Apache v2.0 licenses all lack a specific clause regarding SaaS. When software is offered exclusively as SaaS, none of these three licenses consider that to count as distribution. This means that the obligations imposed by the licenses—such as including copyright notices, disclaimers, and the license text upon distribution—do not automatically apply in the case of SaaS, since no copy of the software is being transferred.

Patents

MIT	BSD*	Apache v2.0
None	None	<p>Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that</p> <p>Apache v2.0 automatically grants a royalty-free patent license for using the work when, without this patent grant, the work could not be used.</p>

It is important to note that MIT and BSD* provide no protection against potential patent infringement. Anyone using software under MIT or BSD* runs the risk of inadvertently infringing third-party patents. Apache 2.0 offers additional assurance in this regard, because each contribution from a licensor comes with an automatic royalty-free patent license, as long as it is necessary to use the work.

Relicensing and Compatibility

MIT	BSD*	Apache v2.0
[...] including without limitation the rights to [...] sublicense [...]	None	Subject to the terms and conditions of this License, each Contributor [...] sublicense [...]

All three licenses impose no restrictions on the context in which the work may be used. For example, a derivative work may be published under a different open-source license or even included in a proprietary project. The original work, however, always remains available under its original license.

It is important to note that BSD* does not explicitly mention this right of sublicensing; it is inferred from the general intent of the license to maximize freedom. This interpretation is supported by the BSD* assessment by the community and organizations such as the Open Source Initiative (OSI) and SPDX. For those seeking unambiguous certainty regarding sublicensing, MIT or Apache 2.0 is the safer choice.

None of these three licenses allow relicensing of the original work. You may not redistribute the work unchanged under a license other than the original license.

In general, open-source licenses are not automatically compatible with derivative works under a more permissive license. Concretely, this means that software under Apache v2.0 cannot be included in a derivative work that is published under MIT or BSD*. The reverse is possible. Overall, these three permissive licenses are generally compatible with stricter licenses, whether weak or strong copyleft.

One important exception is compatibility with GPL v2.0: Apache v2.0 is not compatible with GPL v2.0, but it is compatible with GPL v3.0. This is because GPL v2.0 does not allow the



imposition of additional restrictions. GPL v2.0 lacks a patent clause, while Apache v2.0 includes an explicit patent clause. That additional restriction makes Apache v2.0 incompatible with GPL v2.0. The MIT and BSD* are both fully compatible with both the GPL v2.0 and v3.0.

To avoid license lock-in, a Contributor License Agreement (CLA) can be useful. At the same time, the fear of license lock-in is somewhat less relevant when choosing a permissive license; by selecting MIT, BSD*, or Apache v2.0, you are intentionally releasing your source code as freely as possible.

Later License Versions

The MIT, BSD*, and Apache v2.0 licenses all lack a specific clause regarding new versions of the license. This means that the original work, without modifications, cannot simply be released under a new version of the respective license. You are therefore bound by the exact version of the license under which the work was originally published.

Chain of Authorship



MIT	BSD*	Apache v2.0
Afwezig	Afwezig	<p>Art. 5 Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.</p> <p>Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.</p> <p>By submitting a request to include a contribution under Apache v2.0, the contributor declares that this is done under the same terms as this license. Exceptions are only permitted with the explicit consent of the original licensor.</p>

The MIT and BSD* licenses say nothing about the chain of authorship; Apache v2.0, on the other hand, does. Licenses such as MPL and EUPL go even further by explicitly stating that the contributor must actually have the IP-rights to include the work in the respective project. Apache v2.0 only requires that a contribution fall under the same terms and implicitly assumes that the contributor possesses the necessary IP-rights.

In all three cases, it can be useful to include a Developer Certificate of Origin (DCO). This

explicitly confirms that the contributor owns the IP-rights to their contribution and provides additional legal certainty for the project.

Disclaimer of Warranty and Limitation of Liability

MIT	BSD*	Apache v2.0
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, [...]	THIS SOFTWARE IS PROVIDED BY [Name of Organization] "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES [...] ARE DISCLAIMED. IN NO EVENT SHALL [Name of Organisation] BE LIABLE [...].	Art. 7 [...] Licensor provides the Work [...] on an "AS IS" BASIS, WITHOUT WARRANTIES [...]. Art. 8 In no event [...] unless required by applicable law [...] shall any Contributor be liable to You for damages

All three licenses include the standard as-is clause, which fully disclaims liability. Apache v2.0 is slightly more nuanced: it allows for exceptions where local law does not permit complete exclusion of liability. Thus, the liability disclaimer under Apache v2.0 is not absolute, but in practice it is broadly comparable to that of the MIT and BSD* licenses.

Permissive licenses generally aim to protect the developer or the licensor with a total exclusion of liability. However, in many EU Member States, such total exclusions are not fully enforceable. This means that statutory product liability protections cannot be waived or overridden by contractual terms, even if the license explicitly seeks to exclude all liability. So, these licenses do not protect open source software developers when they have the explicit intention to do harm. No liability exclusion can exempt licensors from the new obligations under recent regulations. These obligations mainly concern cybersecurity. They apply when open source software is placed on the market or commercially offered. See Chapter II of the EU Cyber Resilience Act (CRA).

Additional Agreements

MIT	BSD*	Apache v2.0
Afwezig	Afwezig	Art. 9 You may choose to offer and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. [...] You may act only on Your own behalf and on Your sole responsibility

Only under Apache v2.0 may users or service providers offer additional services or warranties, and always solely under their own name and responsibility. Any liability arising from such offerings rests entirely with them, not with the original licensor or previous contributors. In this way, the standard liability disclaimer of the licensor remains intact, even when third parties provide commercial guarantees.

Inability to Comply

MIT	BSD*	Apache v2.0
None	None	None

Weak Copyleft

The European Union Public License (EUPL) v1.2 and the Mozilla Public License (MPL) v2.0 are both open source licenses with weak to moderate copyleft effects but differ in scope, legal context, and practical application. The EUPL was developed by the European Commission and explicitly aligns with EU legislation. The MPL originates from the Mozilla Foundation.

Feature	EUPL v1.2	MPL v2.0
Scope	Broader than just source code; includes software and related works	Limited to source code and files within the project
Copyleft	Applied at the derivative-work level	Applied at the file level
SaaS / Distribution	SaaS explicitly covered	SaaS not covered
Patents	Royalty-free patent license	Royalty-free patent license, with limitations if code is removed
Relicensing	Possible, flexible, and compatible with many licenses	Possible, but less flexible; specific compatibility rules apply
Copyright chain / contributors	Declaration of rights transfer required	Declaration of rights transfer required
Additional agreements / liability	Functionally similar	Functionally similar
Force majeure / governing law	Follows default mandatory law	Contains explicit instructions for governing law and force majeure

In summary, the key differences are the degree of copyleft and the scope concerning SaaS.

Including SaaS is important because most cloud services from companies like Google (Drive), Microsoft (OneDrive), Apple, and other GAFAM members distribute software as a service without providing access to the source code. Neither the MPL v2.0 nor the GPL v2.0 require developers to share the source code for software offered as a SaaS solution.

Scope

EUPL	MPL
Introduction: This European Union Public Licence (the 'EUPL') applies to the Work (as defined below) which is provided under the terms of this Licence.	Art. 1.13: Source Code Form: means the form of the work preferred for making modifications.
The EUPL can be applied to a wide range of works: source code, documents, datasets, and other digital assets.	The MPL is primarily focused on source code and does not cover other digital assets such as documentation or datasets.

The MPL can only be used for source code. The EUPL can be applied to 'works' in a broader sense; this includes source code, as well as documents, datasets, and other digital assets.

Degree of Copyleft

EUPL	MPL
Art. 1: The Original Work: the work or software distributed or communicated by the Licensor under this Licence, available as Source Code and also as Executable Code as the case may be.	Art. 3.1: All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License.
The EUPL applies copyleft at the derivative work level: any work derived from EUPL-licensed code must be distributed under the EUPL or a compatible license.	The MPL applies copyleft at the file level: only the files covered by the MPL must be made available in source code form.

Under the MPL, a component can contain both MPL-licensed files and proprietary code.

Under the EUPL, copyleft applies to the entire derivative work; extensions or additions must therefore also be released under a compatible license.

Example: Consider an open-source encryption library (such as Mbed-TLS or OpenSSL). These contain several algorithms (A, B, and C), each in their own (set of) files. - Under the MPL, you

can add a proprietary algorithm D in a new set of file(s). - Under the EUPL, such an addition is considered part of the derivative work, so algorithm D must also be released under the EUPL (or a compatible license).

Distribution and SaaS

EUPL	MPL
Art. 1: 'Distribution' or 'Communication': any act of selling, giving, lending, renting, distributing, communicating, transmitting, or otherwise making available, online or offline, copies of the Work or providing access to its essential functionalities at the disposal of any other natural or legal person.	Art. 3.1: All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. [...] Art 3.2: If You distribute Covered Software in Executable Form [...]
SaaS use falls under the concept of distribution; no separate license like the AGPL is required.	The MPL regulates the distribution of source code but does not include an explicit provision for SaaS. Online-only availability does not automatically trigger licensing obligations.

The EUPL is explicitly designed for cloud/SaaS applications; the MPL is limited to traditional source code distribution.

In case of the EUPL, Would the original unmodified source code already be publically available, then there is no obligation to again distribute the unmodified source code.

Patents

EUPL	MPL
Art. 2: The Licensor hereby grants You a worldwide, royalty-free, non-exclusive, sublicensable licence to do the following, for the duration of copyright vested in the Original Work.	Art 1.10: ‘Patent Claims’ of a Contributor means any patent claim(s), including without limitation [...] that would be infringed, but for the grant of the License, by the making, using, selling [...] of either its Contributions or its Contributor Version. Art. 2.3: [...] Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor [...] under Patent Claims infringed by Covered Software in the absence of its Contributions.

Both licenses automatically grant a royalty-free patent license for the use of the work, when such a patent license is necessary to make use of the work. However, the MPL introduces an important nuance. If parts of the original code are removed, the patent license for those removed portions lapses. If you later add similar functionality that resembles what was removed, you could still infringe on the original contributor’s patent.

Relicensing and Compatibility

EUPL	MPL
<p>Art. 5: If the Licensee Distributes or Communicates Derivative Works or copies thereof based upon both the Work and another work licensed under a Compatible Licence, this Distribution or Communication can be done under the terms of this Compatible Licence. [...] 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.</p> <p>The EUPL allows derivative works to be distributed under a compatible open source license. These compatible licenses—including the GPL, LGPL, and MPL 2.0—are listed in the annex of the EUPL. This is fundamentally different from the GPL, which does not allow relicensing under other licenses (except for the AGPL, which is stricter regarding SaaS use). The second part of the article means that you can re-release EUPL-licensed work under a stricter or milder copyleft license: a) Distribution under the MPL weakens the copyleft effect to the file level. b) Distribution under the GPL strengthens the copyleft effect to the entire work.</p>	<p>Art 2.4: No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).</p> <p>If someone decides to attach a different license when distributing the original MPL code—such as a newer MPL version or one of the compatible licenses—this does not change the freedoms originally granted by the initial licensor to the source code.</p>

The broad compatibility of both the EUPL and the MPL provides meaningful flexibility when creating or distributing derivative works. However, this flexibility operates differently in each license. The EUPL allows relicensing for Derivative Works or for Combinations with components under a license listed under Compatible Licence. The MPL, by contrast, uses file-level copyleft: MPL-covered files must remain under the MPL, but they may be combined

with code under any other license within a Larger Work. Relicensing an MPL-covered file itself is only permitted in the cases explicitly defined by the MPL. One such case is distribution under a later version of the MPL. Another is distribution under a license listed under Secondary License, which is allowed by default unless the file is marked Incompatible With Secondary Licenses. This ensures that the MPL is compatible with strong copyleft licenses.

In neither the EUPL nor the MPL does this flexibility permit relicensing the original, unmodified work solely to change its primary license. Doing so would exceed the permissions granted and would constitute copyright infringement.

Later License Versions

EUPL	MPL
Art. 5: If the Licensee distributes or communicates copies of the Original Works or Derivative Works, this Distribution or Communication will be done under the terms of this Licence or of a later version of this Licence unless the Original Work is expressly distributed only under this version of the Licence — for example by communicating ‘EUPL v. 1.2 only’. The Licensee (becoming Licensor) cannot offer or impose any additional terms or conditions on the Work or Derivative Work that alter or restrict the terms of the Licence..	Art. 10.2: You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

Both the EUPL and the MPL allow works to be distributed under later versions of the license, unless the original author explicitly excludes this. This prevents license lock-in: future changes to the license, for example due to updated European legislation, automatically apply to works without version restrictions.

Not all open source licenses permit new versions to apply automatically. It is also not allowed to retroactively fix works published under EUPL v1.2 or MPL v2.0 to those specific versions if no such version restriction was set at the time of publication.

Chain of Authorship

EUPL	MPL
Art. 6: Each Contributor warrants that the copyright in the modifications he/she brings to the Work are owned by him/her or licensed to him/her and that he/she has the power and authority to grant the Licence. [...] Each time You accept the Licence, the original Licensor and subsequent Contributors grant You a licence to their contributions to the Work, under the terms of this Licence.	Art. 2.5: Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

The provisions in both the EUPL and the MPL serve the same purpose and are comparable to a Developer Certificate of Origin (DCO). Each contributor implicitly declares that their contribution is lawful and that they have the authority to publish it under the respective license. This protects subsequent license holders from liability for any copyright errors or infringements made by earlier contributors.

Disclaimer of Warranty and Limitation of Liability

EUPL	MPL
Art. 7: [...] This disclaimer of warranty is an essential part of the Licence and a condition for the grant of any rights to the Work. Art. 8: However, the Licensor will be liable under statutory product liability laws as far such laws apply to the Work.	Art. 6: Covered Software is provided under this License on an "as is" basis, without warranty of any kind [...] Art. 7: any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You

As with almost all open-source licenses, both the EUPL and the MPL exclude warranties and liability.

In the case of the EUPL, this exclusion is not absolute: mandatory legal provisions (such as product liability) still apply.

Additional Agreements

EUPL	MPL
Art. 9: While distributing the Work, You may choose to conclude an additional agreement, defining obligations or services consistent with this Licence. [...]	Art. 3.5: You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations [...] You must make it absolutely clear that any such warranty [...] is offered by You alone.

Under both the EUPL and the MPL, users or service providers may offer additional services or warranties only on their own behalf and under their own responsibility. Any liability arising from these services rests entirely with them and not with the original licensor or previous contributors. This ensures that the standard exclusion of liability for the licensor remains in effect, even if third parties provide commercial guarantees.

Inability to Comply

EUPL	MPL
None	Art. 4: If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply [...] to the maximum extent possible; and (b) describe the limitations [...]

The MPL contains explicit instructions for situations in which compliance with the license is legally impossible. The EUPL does not include such a provision, but of course also adheres to mandatory legal requirements.

Strong Copyleft

The GNU General Public License version 3.0 (GPLv3) and the GNU Affero General Public License version 3.0 (AGPLv3) are both strong copyleft licenses developed by the Free Software Foundation. The main difference between these licenses is that the AGPLv3 specifies that providing software over a network—such as in Software-as-a-Service (SaaS) scenarios—is considered a form of distribution. As a result, the provider must also make the source code available in such cases. The GPLv3 does not include this provision, so SaaS usage falls outside the scope of the license.

Since both licenses are almost identical, the following analysis primarily focuses on the GPLv3. Where there are relevant differences between the AGPLv3 and the GPLv3, these will be highlighted.

Scope

Preamble the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program - to make sure it remains free software for all its users. **Definition** “The Program” refers to any copyrightable work licensed under this License. **Artikel 1. Source Code** The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

Although the (A)GPL, in the definition of “The Program,” formally refers to any copyrightable work, the rest of the license makes clear that it is primarily designed for software. The (A)GPL operates with concepts such as source code, object code, system libraries, and standard interfaces, and it includes provisions regarding installation information and other software-specific aspects.

These concepts have little independent meaning outside the context of computer programs. Therefore, it can be said that the (A)GPL is, in practical and legal terms, a software license, even though the broad definition theoretically allows other types of works to be placed under the license. digital assets.

Degree of Copyleft

5.c. Conveying Modified Source Versions You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply [...] to the whole of the work, and all its parts, regardless of how they are packaged. [...]

This article describes the inheritable (copyleft) effect of the (A)GPL. It requires that any work incorporating GPL code must be distributed as a whole under the (A)GPL. As a result, the entire derivative work inherits the GPL terms. It is precisely this inheritable effect that makes the (A)GPL a strong copyleft license: reuse is permitted, but only while preserving the same freedoms for all subsequent users.

Distribution and SaaS

0. Definitions To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

The (A)GPL does not consider offering software over a computer network (SaaS) to be distribution of a copy. The obligations under the GPL generally only take effect when the work is actually distributed. In this respect, the AGPL does not change that definition.

Instead, the AGPL adds a separate requirement for remote use:

13. Remote Network Interaction; Use with the GNU General Public License. Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software.

This requirement ensures that for modified software offered over a network (SaaS), users still gain access to the source code, even if they do not receive a copy of the program.

It is important to note that this requirement only applies to modified software. If a program is offered as SaaS without changes, the AGPL does not require additional source code distribution, since that source code should already be accessible through the original author or distributor.

Patents

11. Patents. Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

Like the MPL and the EUPL, the (A)GPL automatically grants a royalty-free patent license to the extent necessary to use a contribution lawfully. This license applies to the so-called essential patent claims: patents owned or controlled by the contributor that would be infringed by actions the license permits, such as making, using, selling, or distributing that contributor's contribution.

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

It is important to understand that this does not mean that all relevant patents are automatically royalty-free. Only the contributor's own essential patents are covered by this automatic license. Patents held by third parties, or patents that only become relevant due to subsequent modifications by others, are not included. This is precisely what the above article establishes.

Furthermore, the (A)GPL specifies that a patent license granted to one recipient automatically extends to all other recipients of the work and derivative works. The (A)GPL also prohibits discriminatory patent licenses: a patent license may not restrict the rights explicitly granted by the GPL, nor make them conditional on terms that undermine these rights. This prevents a company from indirectly controlling who may use, distribute, or modify (A)GPL code through patents or commercial arrangements.

Finally, it is important to note that the copyleft or inheritance effect of the (A)GPL applies only to copyright. When (A)GPL code is incorporated into a larger work, the entire work containing the (A)GPL contribution automatically inherits the (A)GPL copyright terms. If that same work also contains parts covered by patents, those patents are not affected by the (A)GPL terms. Users of the combined work are therefore not automatically protected against patent infringement.

Relicensing and Compatibility

2. Basic Permissions. Sublicensing is not allowed; section 10 makes it unnecessary. **5.c. Conveying Modified Source Versions** [...] This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

13. Use with the GNU Affero General Public License. Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

The (A)GPL explicitly does not allow a work to be relicensed under another license. In addition, (A)GPL licenses are generally not compatible with other open-source licenses. The only exception is that GPL code can be combined with and relicensed under the AGPL, because the AGPL contains all of the obligations of the GPL, supplemented with an additional requirement for network use, such as in software-as-a-service (SaaS) scenarios.

In the GPL license text, this is explicitly titled **Use with the GNU Affero General Public License**. and in the AGPL it is titled **Remote Network Interaction; Use with the GNU General Public License**. This keeps the articles aligned in numbering for easy cross-reference between the GPL and AGPL.

Later License Versions

14. Revised Versions of this License. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the 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.

The (A)GPL is not automatically compatible with newer versions of the (A)GPL. To explicitly allow the use of a later version, the original author must clearly indicate this by stating in the

license text that later versions of the (A)GPL also apply. If no specific version is mentioned, users are free to choose whichever version of the (A)GPL they wish to apply.

Chain of Authorship

11. Patents. A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based.

This clause functions similarly to a Developer Certificate of Origin (DCO). Each contributor implicitly affirms that their contributions are lawful and that they have the authority to release them under the applicable license. In this way, it prevents downstream licensees from being held liable for potential copyright errors or infringements of previous contributors.

Disclaimer of Warranty and Limitation of Liability

15. Disclaimer of Warranty. THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, [...]. **16. Limitation of Liability.** IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, [...].

In the original license, this is written entirely in uppercase.

As with almost all open-source licenses, the (A)GPL explicitly disclaims warranties and liability, meaning that users use and redistribute the program entirely at their own risk.

Additional Agreements

12. No Surrender of Others’ Freedom. If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. [...] the only way you could satisfy

both those [other] terms and this License would be to refrain entirely from conveying the Program.

This article reinforces the inheritance effect of the (A)GPL. If, for any reason, you cannot comply with the conditions of this license, you are not permitted to use or distribute the work. In other words, the source code shared under the (A)GPL may not be used at all. The only exceptions are those described in Articles 15, 16, and 17, which relate to warranty and liability.

Inability to Comply

17. Interpretation of Sections 15 and 16. If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

Under the (A)GPL, the exclusion of warranty and liability is not absolute. If local law does not fully allow the provisions in Sections 15 and 16, the applicable statutory rules that most closely align with these sections will apply. However, parties are free to accept warranty or liability, for a fee, when using or distributing the program.

OSPO Reviewed License Checklist

To simplify the evaluation and use of open-source software (OSS) licenses, an OSPO Reviewed License checklist can be helpful. This checklist provides an overview of commonly used OSS licenses and indicates whether they are approved for use within your organization. It helps establish a clear reference framework for your organization regarding the use of OSS licenses.

Recommendations

Strong copyleft and EU-law

One of the most prominent aspect of strong copyleft is the opinion of the Free Software Foundation that linking (statically or even dynamically) a (A)GPL covered work with any other

program creates a combined derivative globally covered by the (A)GPL. This opinion is at the origin of qualifying the (A)GPL as “viral”. Under the EU law (Directive 2009/24/EC; On the legal protection of computer programs) this opinion looks invalid, but at the contrary of the EUPL, the application of the (A)GPL is not specifically covered by the EU law. This means that—due to the absence of explicit grounding in EU law and the lack of settled European case law—the lawfulness of this aspect of the (A)GPL is legally uncertain.

So there are two reasons why the EUPL should be considered a moderately (weak) copyleft license: 1. According to European law, which still applies to the EUPL, linking multiple programs to make them interoperable should always be permitted, regardless of any license. This exception to copyright stems from recitals 10 and 15 of the Directive 2009/24/EC. This is the main reason why the EUPL is not to be considered as a “viral licence”: each component made interoperable through linking retains its original license. 2. Regarding derivative works where source code covered by different licenses is not simply linked but truly merged, the EUPL allows such a combined work to be distributed under a compatible license, and the list includes the “weak copyleft” MPL & LGPL.


For more information: Why the EUPL is NOT a Viral Licence?

Contributors

Thanks to everyone mentioned by name here, as well as all the contributors who wish to remain anonymous.

- Maurice Hendriks (Main author; Dutch Ministry of Health, Welfare and Sport)
- Jonas van den Bogaard (Alliander)
- Patrice-Emmanuel Schmitz (External European Commission advisor)

This text is available under the CC-BY-4.0

 Collaborate on HackMD