

14-04-2025

Open Source Licences Used in GÉANT

Authors: Branko Marović (UoB/AMRES), Kiril Kjiroski (UKIM/MARNET)


This work is licensed under Creative Commons Attribution-ShareAlike 4.0 International. To view a copy of this licence, visit <https://creativecommons.org/licenses/by-sa/4.0/> 

Table of Contents

Executive Summary	4
1 Introduction	4
1.1 Licence Compatibility	4
1.2 Relicensing and Sublicensing	6
1.3 Copyright, Patents and Trademarks	7
2 Low Risk Licences	8
2.1 Apache-2.0	8
2.2 Bouncy Castle License	8
2.3 BSD-2-Clause	8
2.4 BSD-3-Clause	9
2.5 BSD-4-Clause	9
2.6 BSL-1.0	10
2.7 CC0-1.0 / WTFPL / Unlicense	10
2.8 CC-BY-4.0	10
2.9 CC-BY-SA-4.0	11
2.10 CDDL-1.0	11
2.11 CDDL-1.1	11
2.12 Golang BSD + Patents	12
2.13 ISC / OBSD	12
2.14 MIT / X11	13
2.15 NUnit	13
2.16 OpenSSL	13
2.17 Public Domain	14
2.18 Python-2.0	14
2.19 Zlib	14
3 Medium Risk Licences	15
3.1 Artistic-1.0	15
3.2 Artistic-2.0	15
3.3 EPL-1.0	16
3.4 EPL-2.0	16
3.5 EUPL-1.2	17
3.6 GPL-2.0-with-classpath-exception	17
3.7 LGPL-2.0	17
3.8 LGPL-2.1	18
3.9 LGPL-3.0	18
3.10 MPL-1.1	19
3.11 MPL-2.0	19

4	High Risk Licences	20
	4.1 AGPL-3.0	20
	4.2 GPL-1.0	20
	4.3 GPL-2.0	20
	4.4 GPL-3.0	21
5	Glossary 21	
6	Licence Cheat Sheet	23

Executive Summary

This document provides a comprehensive overview of the main open source licences for GÉANT Software Developments teams. It highlights the importance of licence compliance and offers recommendations for selecting the appropriate licence for GÉANT projects by describing each licence's specific requirements, compatibility and patent provisions. By following the recommendations and guidelines outlined in this document, GÉANT software development teams can ensure that their projects remain compliant with OSS licences and contribute to the open source community. Ensuring compliance with these licences and understanding their interactions facilitates better governance and reduces legal risks. For further assistance and guidance on OSS and licensing in GÉANT software developments, please contact the GÉANT OSLS team at osls@geant.org.

1 Introduction

Open source licences can be classified into several categories. Below is a brief list of these categories:

Public Domain is the most permissive model and is not technically a licence. It is not always possible to waive all rights under the laws and regulations of some countries. In such cases, the CC0 public domain dedication can replace the public domain for practical purposes. Software in the public domain carries no potential liability. Its modified code can be licensed under any terms, making it usable in both open source and commercial software by allowing derived works to be distributed without disclosing any source code.

Permissive Licences impose minimal requirements on how software may be modified or redistributed. Almost two-thirds of open source software in circulation uses this type of licence.

Copyleft Licences allow the modification of code and the distribution of new works based on it, provided the requirements for redistribution under the same conditions are met.

- **Weak Copyleft Licences:** These have a library or file scope.
- **Strong Copyleft Licences:** These often require releasing the entire project or product under a licence that is the same as or similar to the one used in the original works.
- **Network Protective Licences:** Under these, distribution over a network is considered a form of distribution, thus requiring modified code to be made available to external users.

Every licence listed below is classified into one of the above categories, under the title containing the licence name. For easier use with Mend reports, the licences are presented alphabetically after being grouped into IPR risk categories. Mend defines three risk categories for software licences: Low Risk, Medium Risk and High Risk. These categories broadly correspond to Permissive, Weak Copyleft and Strong Copyleft licences. However, this classification is not strict (e.g. Artistic 2.0 is a permissive licence but belongs to Mend's Medium Risk category). In this document, licences are listed alphabetically within each risk category. Only licences detected by Mend in GÉANT software projects are included.

1.1 Licence Compatibility

The most frequently used licences in GÉANT software projects and their relationships are shown in Figure 1.

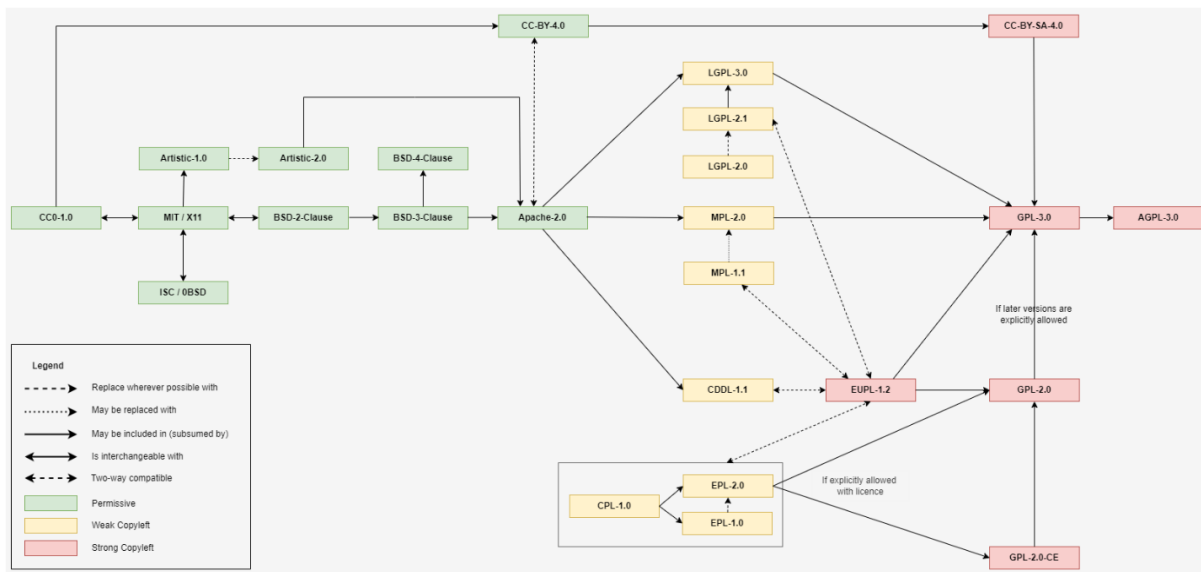


Figure 1: Relationships between the most frequently used licences in GÉANT projects

In Figure 1, different colours indicate categories of software licences. Licences marked in green are permissive, those in yellow are weak copyleft and those in red are strong copyleft. Relationships between licences are illustrated using different arrow styles:

- A **one-way solid arrow** means the destination licence subsumes the origin licence. In some cases, as noted in Figure 1, additional conditions may apply.
- A **one-way dashed arrow** indicates that there is a newer version of the licence, suggesting the author should replace the older version in their code.
- A **one-way dotted arrow** signifies that the older licence can be replaced with the new version by an author using and modifying the code or mixing it with their own, provided the original code has not been distributed with additional licences.
- A **two-way solid arrow** means both licences are interchangeable.
- A **two-way dashed arrow** signifies that authors can license combined work under either of the licences. For better future compatibility, it is recommended to publish the combined work under both.

In this document, the statement “Licence A is subsumed by Licence B” means that code under Licence A can be included in work licensed under Licence B, represented in Figure 1 by an arrow from A to B. This is equivalent to stating “Licence B subsumes Licence A.” When Licence A is subsumed by Licence B, a copy of Licence A must still be included with all distributions of the combined program. “Interchangeable” means that code under Licences A and B can be licensed under either one. “Licence A is compatible with Licence B” means it is permitted to publish or distribute combined work under a licence, typically at least one of the input licences. In other words, when licences are compatible, it is legally possible to combine or merge multiple pieces of software. These nuances matter when selecting a licence for the combined program.

Open source software licences typically require the licence to be kept with the code it covers, usually by including its original text or a reference to it. Strictly speaking, the licence of the combined program absorbs (i.e. subsumes), rightfully replaces or is interchangeable with the licences of all its parts. Below is a summary of the procedure for licensing a combined program.

To determine which licences can be used for a combined program, authors should start with a list of all the licences used in their work. They can then remove any licence subsumed by another on the list, meaning the

code under it can be included in software under the other licence. A licence is subsumed by another when compliance with the former implies compliance with the latter. For example:

- LGPL 2.1 is subsumed by LGPL 3.0 and both are subsumed by GPL 3.0 and AGPL 3.0.
- LGPL 2.0 and GPL 2.0 are NOT automatically subsumed by LGPL 3.0 and GPL 3.0.
- The Apache 2.0 licence is subsumed by any GNU licence, version 3.0 or later.
- Mozilla Public Licence 2.0 is subsumed by GNU GPL, version 2.0, 3.0 or later.
- BSD, Expat, MIT/X11 and ISC licences and CC0 public domain dedication are subsumed by the Apache 2.0 licence.
- Expat, MIT/X11 and ISC licences are interchangeable with the BSD 2-clause licence.
- BSD 2-clause is subsumed by BSD 3-clause.

In this document, the phrase “generally considered” is used to indicate that there is no explicit statement of the relationship in the licence text, but it is inferred from the licence clauses and lacks legal precedent.

The recommendations in this guide aim to minimise the risks of incompatibility and maximise the ability to reuse and combine covered source code and its licence(s). This follows the principle that source code developed or modified with public funding should remain public. The EUPL 1.2 licence, like other copyleft licences, preserves the public nature of the covered code while offering maximum compatibility with other licences. It therefore combines many advantages of strong copyleft without the restrictive nature of viral clauses, allowing free combination and linking of differently licensed components and covering all distribution modes including remote distribution (SaaS).

Works analysed by Mend sometimes have dual licences. Dual and multi-licensing involve releasing the same code under multiple licences, allowing users to choose which to apply provided the project complies with the requirements of all offered licences. This increases compatibility options for users and enhances flexibility when selecting components or licensing their own code. Conversely, when different components within a project use different licences, the project should be described as using “multiple licences” rather than being multi-licensed. Mend, however, displays a list of licences in both cases.

In dual licensing, copyleft licences are often presented as one option because they impose strict open source conditions on users’ code. This drives users unwilling to comply with these conditions to select the alternative, typically a commercial licence. Offering a permissive licence instead would remove the incentive to choose the commercial option. This approach, known as “Choice of Licence,” enables casual or small-scale users to use the software free of charge, while enterprise users or those incorporating the software into products must choose the commercial licence.

Multiple permissive licences may also be offered in multi-licensing arrangements to ensure compatibility with projects under various licences, as no single licence may suit all. For example, since GPL v2 cannot be combined with Apache 2.0, authors may dual-license their project under both to allow use under either.

1.2 Relicensing and Sublicensing

Relicensing is the act of changing the current licence to another. This is possible if the original licence is permissive (thus includes the right to change the licence), allows relicensing to a secondary licence named in its terms or if all contributors agree to the change. It includes two scenarios:

- Switching to another copyleft licence where the original one explicitly allows relicensing under it (e.g. CeCILL permits relicensing to GPL v2 or later).

- Changing the licence entirely, which requires explicit written permission from all contributors. Permissive licences often do not explicitly mention relicensing or sublicensing, but their permissive terms effectively allow it.

Sublicensing allows the licensee (the individual or organisation granted the licence) to pass on certain rights, such as changing the licence, to a third party. This can be essential for broader distribution, integration or software use, particularly in commercial or collaborative projects. Sublicensing depends heavily on the terms of the original licence. Common use cases include:

- Commercial distribution, where companies sublicense software to customers as part of a larger product or service offering.
- Partnerships, where organisations sublicense software to partners for collaboration or integration into joint projects.
- Custom solutions, where developers sublicense components of permissively licensed software for client-specific applications or bespoke solutions.

Permissive licences typically grant sublicensing rights explicitly, though with varying conditions. Therefore, projects under permissive licences can be relicensed, changing the licence of the original or modified code. This is typically done to apply a commercial licence, which some may disguise by applying an open source looking “source-available” licence, often referred to as “fauxpen.” This may be undesirable if the original intent was to keep the software free. Copyleft licences generally prohibit sublicensing for the code within their scope by requiring redistribution under the same terms.

1.3 Copyright, Patents and Trademarks

Preserving original copyrights and notices is a key legal and ethical requirement when using, modifying or distributing software. These notices ensure proper recognition of the original creators and maintain the integrity of the licence terms.

Patents and patent rights are sometimes addressed in licences through explicit patent grants, as seen in LGPL, GPL, AGPL, Apache 2.0, EPL and EUPL. Some licences do not mention patent rights explicitly, but their language may imply such grants. Patent-related concerns may arise when derived works are distributed under a different licence along with permissively licensed original parts. Licence texts may include patent-related clauses such as:

- Defensive termination, which automatically revokes patent rights if the licensee initiates patent litigation against the authors or contributors.
- Broad patent grants, covering all current and future patents owned by contributors that could be infringed by the entire codebase and through modifications or combinations, and not just by specific (existing or new) contributions.

Most open source licences exclude trademark rights unless explicitly stated. Trademarks of the original work are typically preserved through applicable trademark law and supported by the licence’s requirement to include copyright notices. However, some licences regulate trademark use explicitly:

- Apache 2.0: “... does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor...”
- BSD 3- and 4-clause: the non-endorsement clause prohibits use of project and contributor names without written permission.
- Artistic 2.0: “... does not grant you the right to use any trademark, service mark, tradename, or logo of the Copyright Holder.”

2 Low Risk Licences

2.1 Apache-2.0

Permissive

Description: The Apache 2.0 is a widely used licence that allows software use, distribution and modification for any purpose. Derivative works can have stricter licences but unaltered components must use the Apache 2.0 licence. Strong OSI community support. Unlike MIT, recipients receive a patent licence. Applicable law and competent court are unspecified.

Requirements and mixing: The original copyright notice should be included in a separate file and should copy the NOTICE with attribution notes (if present) from the original library. The source code should include the licence text and a notice of significant changes in files that were changed. There is no obligation to disclose the affected source code but modification notifications must be included. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: The licence is subsumed by GPL 3.0, LGPL 3.0 and CDDL. It is two-way compatible with MIT, BSD (2- and 3-clause) and MPL 2.0. It is linking-permissive.

Patent rights: Patent rights to the software user or developer are explicitly granted.

Remarks: The Apache licence does not grant trademark rights, implying that users cannot use the project's name or associated trademarks without explicit permission.

2.2 Bouncy Castle License

Permissive

Description: The Bouncy Castle License should be read in the same way as the MIT licence. It is specific to the Bouncy Castle Crypto APIs, a widely used cryptography library in Java and C#. The Bouncy Castle License is designed to be business-friendly while promoting the use of strong cryptography.

Requirements: The source code should include notice files and the original licence. Users must retain and add new attributions for copyright, authorship and recognition. Modified files must be marked as such to maintain transparency. Users are allowed to sublicense the software and its modifications.

Compatibility, linking and mixing: Code under MIT, BSD (2- and 3-clause), Apache 2.0 and MPL 2.0 can be combined with code under the Bouncy Castle License and published under any of the input licences. Code under the Bouncy Castle License can be included in projects licensed under the GPL and LGPL.

Patent rights: Patent rights are not mentioned.

2.3 BSD-2-Clause

Permissive

Description: BSD licences are a family of permissive licences with minimal restrictions on the redistribution of covered software. The BSD 2-clause licence is the simplest BSD licence and the most compatible with other licences.

Requirements: The original copyright notice should be added to the licence text. The source code should include the licence text. The source code does not have to include a notice of significant changes. There is no obligation to disclose affected source code. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: It is subsumed by the entire GPL family (including LGPL and AGPL licences) and is linking-permissive. The BSD 2-clause licence is interchangeable with MIT and ISC licences. MPL and Apache 2.0 licences are two-way compatible with the BSD 2-clause licence. The BSD 2-clause licence is subsumed by BSD 3-clause and 4-clause licences.

Patent rights: Patent rights are not mentioned.

Remarks: There is no prohibition on using the name of the project to promote work.

2.4 BSD-3-Clause

Permissive

Description: In addition to the conditions in the BSD 2-clause licence, the author's name may not be used to endorse or promote products derived from this software without specific prior written permission.

Requirements: The original copyright notice should be included in the licence text. The source code should contain the licence text but does not need to include a notice of significant changes. There is no obligation to disclose affected source code. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: The licence is subsumed by GPL and LGPL licences. ISC, MIT, MPL and Apache 2.0 are two-way compatible with BSD 3-clause. BSD 3-clause is subsumed by BSD 4-clause. It is linking-permissive.

Patent rights: Patent rights are not mentioned.

Remarks: Non-endorsement clause: the name of the project or its contributors may not be used without written permission.

2.5 BSD-4-Clause

Permissive

Description: In addition to the conditions in the BSD 3-clause licence, neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

Requirements: The original copyright notice should be added to the licence text. The source code should include the licence text. The source code should include an acknowledgment statement. It does not have to include a notice of significant changes. There is no obligation to disclose affected source code. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: BSD 4-clause is not GPL compatible and may cause issues when combined with other licences. Due to the advertising clause, BSD 4-clause is not compatible with other licences, especially those with more restrictive terms that prohibit the additional requirement of the advertising clause. Nonetheless, it subsumes other BSD licences.

Patent rights: Patent rights are not mentioned.

2.6 BSL-1.0

Permissive

Description: The Boost Software License (BSL) imposes minimal restrictions on code use and redistribution. Initially created for the Boost C++ Libraries, BSL encourages open collaboration in a business-friendly and flexible manner.

Requirements: The source code should include the original copyright notice. This requirement does not apply when distributing compiled code. The licence text should also be included but a notice of significant changes is not necessary. There is no obligation to disclose affected source code. Users have the right to sublicense the software and its modifications, providing additional flexibility in licensing for downstream recipients.

Compatibility, linking and mixing: BSL is subsumed by the entire GPL family (including LGPL and AGPL licences) and is linking-permissive. BSL is two-way compatible with MIT, BSD (2- and 3-clause), Apache 2.0 and MPL. It can be included in projects licensed under Apache License 2.0.

Patent rights: Patent rights are not mentioned but are considered to be implicitly granted.

2.7 CC0-1.0 / WTFPL / Unlicense

Public Domain

Description: These licences effectively dedicate a work to the public domain to the fullest extent permitted by law. They also provide lax licences to cover cases where a simple dedication is inadequate. Only Unlicense is an OSI-approved licence.

Requirements: No copyright notice or licence text is required (CC0 suggests attribution on moral grounds). The source code does not need to include a notice of significant changes. There is no obligation to disclose modified source code. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: As public domain dedication tools or waivers, these licences can be inherently subsumed by all licences and works and are linking-permissive.

Patent rights: Patent rights are not mentioned (in the case of CC0, they are not granted).

2.8 CC-BY-4.0

Permissive

Description: Permissive licence that provides flexibility for both commercial and non-commercial use, making it suitable for projects and initiatives that value openness, collaboration and the free exchange of ideas and content. However, CC BY 4.0 is not intended to be used for software.

Requirements: Original copyright notice should be included in the licence text. The source code should include the licence text or URL linking to it. There is no requirement to include a notice of significant changes. There is no obligation to disclose affected source code. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: The licence is subsumed by all GPL and LGPL licences. ISC, BSD, MIT, MPL and Apache 2.0 are two-way compatible with CC BY 4.0. It is linking-permissive.

Patent rights: Patent rights are not granted.

2.9 CC-BY-SA-4.0

Strong Copyleft

Description: The Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) licence is a widely used open access licence for creative works such as text and images, allowing creators to share their works while retaining attribution rights and copyleft-style control. It is **not intended for software** but is suitable for documentation, allowing sharing and modification under clear attribution and share-alike terms. This distinction ensures documentation remains independent of software licensing considerations, ensuring clarity and legal compliance.

Requirements: The original copyright notice should be added to the licence text. The source code should include the licence text or a URL link to it. Any modifications should be noted and previous notices retained. There is no obligation to disclose affected source code. Sublicensing of relevant code is not allowed.

Compatibility, linking and mixing: This licence is subsumed by GPL 3.0 but not by GPL 2.0. CC BY-SA 4.0 subsumes CC BY 4.0, CC BY-NC-SA 4.0 and other CC 4.0 licences. It is not linking-permissive and is typically used for data, databases and other content. If content licensed under this licence is part of a software package, the licence does not extend to the software as a whole. Modified content must remain under the same licence.

Patent rights: Patent rights are not granted.

2.10 CDDL-1.0

Weak Copyleft

Description: The Common Development and Distribution License (CDDL) is derived from the Mozilla Public License 1.1. CDDL is used primarily for Oracle-related projects, allowing both commercial and non-commercial use.

Requirements: The original copyright notice, licence text and a list of conditions must be retained in the software documentation or other materials provided with the distribution. Both the original source code and modified derivatives must be made available. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: CDDL 1.0 is considered incompatible with GPL. CDDL is generally considered two-way compatible with 2-clause and 3-clause BSD, MIT and Apache 2.0 licences. It is linking-permissive.

Patent rights: The licence includes a patent grant clause, providing a limited patent licence to the patents owned by contributors.

2.11 CDDL-1.1

Weak Copyleft

Description: CDDL 1.1 includes corrections to address conflicts with European copyright law found in CDDL 1.0 and allows individual developers to use the CDDL for their work. It is considered a weak copyleft licence, with copyleft applying at the file level.

Requirements: The original copyright notice should be added to the licence text. The source code should also include the licence text. While it is not mandatory to include a notice of significant changes, you must disclose affected source code when distributing the work. Developers should ensure that derivative work is licensed under the same version of the licence.

Compatibility, linking and mixing: CDDL 1.1 is not compatible with the entire GPL family (including LGPL and AGPL licences). It is generally considered two-way compatible with 2-clause and 3-clause BSD, MIT, MPL 2.0 and Apache 2.0. It is permissible to combine CDDL-licensed code with software under other licences to create an aggregate “larger work,” which can be kept proprietary or released under a different licence. It is linking-permissive.

Patent rights: Patent rights are explicitly granted. Contains a patent infringement termination clause.

Remarks: It is recommended to use dynamic linking in programming languages that also support static linking.

2.12 Golang BSD + Patents

Permissive

Description: The Golang BSD + Patents licence is based on the BSD 3-clause licence and includes explicit patent rights.

Requirements: The original copyright notice should be added to the licence text. The source code should include the licence text but does not require a notice of significant changes. There is no obligation to disclose modified source code. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: The licence is subsumed by all GPL and LGPL licences. Code under BSD 2-clause, BSD 3-clause, MIT and Apache 2.0 licences can be combined with code under Golang BSD + Patents under either of the input licences. It is two-way compatible with MPL. It is linking-permissive.

Patent rights: The licence includes a patent grant.

Remarks: It is essentially the BSD 3-clause licence with the added condition that users cannot sue Google for licence infringement.

2.13 ISC / OBSD

Permissive

Description: The ISC licence is a “stripped-down” version of the MIT and simplified 2-clause BSD licences, removing extraneous language. Despite its name, the OBSD licence is a simplified version of the ISC licence, eliminating even the requirement to include a copyright notice. All other requirements, compatibility and mixing rules also apply to OBSD.

Requirements: The original copyright notice should be added to the licence text (OBSD does not require it). The source code should include the licence text but does not need to include a notice of significant changes. There is no obligation to disclose affected source code. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: It is linking-permissive. The ISC licence is generally considered interchangeable with other permissive licences like MIT and BSD 2-clause. It is two-way compatible with the BSD 3-clause. ISC-licensed code can be included in projects licensed under GPL, LGPL, MPL, Apache 2.0 and CDDL.

Patent rights: Patent rights are not mentioned.

2.14 MIT / X11

Permissive

Description: The MIT licence is highly compatible with other permissive licences.

Requirements: The original copyright notice should be added to the licence text. The source code should include the licence text but not a notice of significant changes. There is no obligation to disclose modified source code. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: The MIT licence is linking-permissive. Code under 2-clause BSD and ISC licences can be freely combined with MIT-licensed code and published under each of these licences. BSD 3-clause, MPL and Apache 2.0 licences are two-way compatible with the MIT licence. MIT-licensed code can be used in GPL, LGPL and CDDL-licensed projects.

Patent rights: Patent rights are not mentioned but are implicit in the USA due to “without restriction.”

Remarks: It allows the use or mention of trademarks except for X11, which forbids usage of the X Consortium name without prior written authorisation.

2.15 NUnit

Permissive

Description: NUnit is a unit-testing framework for all .Net languages. Newer versions (starting from 3.0) are released under the MIT licence while older versions use the zlib/libpng licence. The licence allows the use of NUnit in free and commercial applications and libraries without restrictions.

Requirements: By association with the MIT licence, the original copyright notice should be added to the licence text. The source code should include the licence text but not a notice of significant changes. There is no obligation to disclose modified source code. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: NUnit is linking-permissive by association with the MIT licence. Code under BSD 2-clause, ISC and other MIT-licensed code may be freely combined and published under either of the input licences. BSD 3-clause, MPL and Apache 2.0 licences are two-way compatible with the NUnit licence. NUnit-licensed code can be used in GPL, LGPL and CDDL-licensed projects.

Patent rights: By association with the MIT licence, patent rights are not mentioned but are implicit in the USA due to “without restriction.”

2.16 OpenSSL

Permissive

Description: The OpenSSL License is the original licence used for the OpenSSL project before version 3.0. The last version of this licence is OpenSSL License 1.1. It combines texts from the Apache 1.0 licence (OpenSSL part) and the BSD 4-clause licence (Original SSLeay part).

Requirements: Redistributions of source code must retain the copyright notice, the list of conditions and the disclaimer of warranty. The source code should include the licence text but not a notice of significant changes. There is no obligation to disclose modified source code. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: The OpenSSL licence is linking-permissive. It requires the phrase “this product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit,” making it similar to the Apache 1.0 licence and incompatible with all GPL family licences. The original SSLeay License includes the phrase “This product includes cryptographic software written by Eric Young (eay@cryptsoft.com),” a characteristic of the BSD 4-clause licence, making it incompatible with other licences, especially those with more restrictive terms prohibiting the additional advertising clause requirement. It subsumes BSD 2-clause and 3-clause licences. All inherited issues have been resolved in newer versions (3.0 and greater) of OpenSSL software, which use the Apache 2.0 licence.

Patent rights: Patent rights are not mentioned.

2.17 Public Domain

Public Domain

Description: Public domain is not a licence per se, but a “public domain dedication” means the creator relinquishes all rights to a work, placing it in the public domain. This allows anyone to freely use, modify, distribute and adapt the work without restrictions or licensing conditions. Since public domain is usually implicit, it is often formalised by applying the CC0 dedication.

Requirements: None.

Compatibility, linking and mixing: There are no conflicts or restrictions related to licence compatibility when using or combining public domain works with works under any other licence, whether open source or proprietary. This means that public domain is subsumed by any licence.

Patent rights: Unlicense and WTFPL, as public-domain equivalent formalisations, grant all rights, functioning as unconditional waivers.

2.18 Python-2.0

Permissive

Description: Its primary use is for the distribution of Python project software and documentation.

Requirements: The source code should include the original copyright notice and licence text. The source code should include notices of significant changes. There is no obligation to disclose modified source code. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: Python licences are subsumed by GPL and LGPL **except** for versions **Python 1.6b1 through 2.0 and 2.1**. Code under MIT, Apache 2.0, MPL and BSD (2- and 3-clause) licences can be combined with Python 2.0-licensed code since they are considered two-way compatible. The Python 2.0 licence is subsumed by the CDDL 1.1 licence. It is linking-permissive.

Patent rights: Patent rights are not mentioned.

2.19 Zlib

Permissive

Description: The Zlib licence is used for a variety of software libraries and applications, named after the zlib compression library and by the libpng library for working with PNG image files.

Requirements: The source code should include the original copyright notice for source code only. It should include the licence text for source code only. It should include notice of significant changes. There is no obligation to disclose affected source code. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: Code under MIT and BSD licences (2- and 3-clause), Apache 2.0 and MPL can be combined with code under the Zlib licence (two-way compatible). Code under the Zlib licence can be used in GPL, LGPL and CDDL-licensed projects. It is linking-permissive.

Patent rights: Patent rights are not mentioned.

3 Medium Risk Licences

3.1 Artistic-1.0

Permissive

Description: According to the FSF, Artistic 1.0 is not considered a free software licence (as it contains passages lacking clear meaning) and is therefore not compatible with the GPL. It is primarily used for Perl programming language software and related projects (also modified as Artistic Perl 1.0). Authors are recommended to replace Artistic 1.0 with Artistic 2.0 whenever possible due to compatibility issues.

Requirements: The source code should include the original copyright notice and licence text. Any differences should be clearly documented although you do not have to disclose a modification if it can be replaced with the original in the combined work or if it is available to the original work's copyright holder. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: Code under the Apache 2.0, MPL, MIT and BSD licences (2- and 3-clause) may be combined with code under Artistic 1.0 and published under either of the input licences. This is subject to specific Artistic 1.0 requirements such as marking modifications and preserving copyright notices. It is linking-permissive.

Patent rights: Patent rights are not granted.

3.2 Artistic-2.0

Permissive

Description: Artistic 2.0 is compatible with the GPL thanks to the relicensing option in section 4(c)(ii). It is primarily used for Perl programming language software and related projects. Wherever possible, it is recommended to replace Artistic 1.0 with Artistic 2.0 due to compatibility issues. Licensing compatibility is greatly improved with the relicensing clause.

Requirements: The source code should include the original copyright notice and licence text. Differences should be clearly documented but modifications do not need to be disclosed if they can be replaced with the original in the combined work or if they are available to the original work's copyright holder. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: Code under Apache 2.0, MPL, MIT and BSD licences (2- and 3-clause) may be freely combined with code under Artistic 2.0 and published under either of the input licences. Code under Artistic 2.0 can be used in GPL, LGPL and CDDL-licensed projects. It is linking-permissive.

Patent rights: Patent rights are explicitly granted.

3.3 EPL-1.0

Weak Copyleft

Description: The Eclipse Public License (EPL) is like the Common Public License, with the removal of the broader patent retaliation language regarding patent infringement suits specifically against “Contributors to the EPL program.” EPL-licensed code in a program can only be released under the EPL even if the full program is released under a different licence. It should be replaced with EPL 2.0 wherever possible.

Requirements: The source code should include the original copyright notice, licence text and notices of significant changes. You should disclose the affected source code when distributing work. Sublicensing of relevant code is permitted. A distributor can relicense to EPL 2.0.

Compatibility, linking and mixing: EPL 1.0 is not compatible with the entire GPL family (including LGPL and AGPL licences). EPL 1.0-licensed code can be migrated to EPL 2.0 and vice versa. Code under MIT, Apache 2.0, MPL 2.0 and BSD (2- and 3-clause) licences can be combined with EPL 1.0-licensed code although it requires licensing EPL 1.0 parts under the EPL 1.0 licence. It is linking-permissive.

Patent rights: Patent rights are explicitly granted.

Remarks: Unlike EPL 2.0, it is not suitable for scripting languages as it does not require source code to be made available; there is no Secondary License. New York State is defined as the “choice of law.” The affected code is defined as a “module,” not a file.

3.4 EPL-2.0

Weak Copyleft

Description: If an initial contributor releases a specific piece of code and designates GPL 2.0 or later as a secondary licence, it provides explicit compatibility with these GPL versions. However, the EPL 2.0 without this designation is incompatible with the GPL.

Requirements: The source code should include the original copyright notice and licence text. While the source code does not need to include a notice of significant changes, you should disclose the affected source code when distributing the work. Sublicensing of relevant code is permitted. The same or later EPL is applicable for affected code and you can add a secondary licence to enable GPL 2.0+ compatibility, allowing sublicensing of binaries under other terms.

Compatibility, linking and mixing: It is not compatible with the entire GPL family (including LGPL and AGPL licences except for special cases stated in the “Description”). EPL 2.0 and EPL 1.0 are generally considered interchangeable. Code under MIT, Apache 2.0, MPL 2.0 and BSD (2- and 3-clause) licences can be combined with code under EPL 2.0 licence although the EPL 2.0 parts must remain under the EPL 2.0 licence. It is linking-permissive.

Patent rights: Patent rights are explicitly granted.

Remarks: Commercial distributors should defend contributors from any lawsuits or legal damages, which is a unique feature of this licence.

3.5 EUPL-1.2

Strong Copyleft

Description: The European Union Public License 1.2 is designed for software created and distributed by public administrations of European Union Member States. The EUPL aims to promote the sharing and reuse of software developed by public administrations.

Requirements: The source code should include the original copyright notice and licence text. It should include notices of significant changes and you should disclose affected source code when distributing work. Sublicensing of relevant code is permitted.

Compatibility, linking and mixing: It is subsumed by GPL 2.0 or later, AGPL 3.0, CeCILL 2.0, OSL 2.1, OSL 3.0. EUPL 1.2 is explicitly two-way compatible with LGPL 2.1 or later, LiLiQ, CeCILL 2.1, CPL 1.0, EPL 1.0 or later, MPL 1.1 or later and with other EUPL versions. EUPL 1.2 subsumes Apache 2.0, Artistic, BSL 1.0, BSD 2-clause and 3-clause, CDDL 1.0 or later, ISC, MIT, Python 2.0 and Zlib. It is linking-permissive since linking makes no derivatives. When combining EUPL 1.2-licensed code with source code licensed differently, the resulting derivative can be licensed under a subsuming licence.

Patent rights: Patent rights are explicitly granted.

3.6 GPL-2.0-with-classpath-exception

Strong Copyleft

Description: The exception allows you to link your GPL-licensed code with specific non-GPL libraries (like the Java Classpath libraries) without making your code subject to the GPL. The Classpath exception only applies to the specific libraries or circumstances mentioned in the exception itself.

Requirements: The source code should include the original copyright notice and licence text. It should include notices of significant changes including dates. When distributing any derived work, you must disclose the affected source code but this is not required for private use. You cannot sublicense the relevant code.

Compatibility, linking and mixing: It is subsumed by the entire GPL family (including LGPL and AGPL licences). The Classpath exception allows GPL-licensed code to be linked with specific non-GPL libraries without forcing GPL requirements on the entire work. It is linking-permissive only for executables.

Patent rights: Patent rights are not mentioned but are considered to be implicitly granted.

3.7 LGPL-2.0

Weak Copyleft

Description: The Library GPL 2.0 is an older version of the LGPL and is now considered obsolete. It should be replaced with version 2.1. LGPL is primarily used for software libraries, enabling their use in open source and proprietary applications. Under the LGPL, you only need to share modifications to the library and you can dynamically link it in your application without sharing your application's source code.

Requirements: The source code should include the original copyright notice and licence text. Additionally, the source code should include notices of significant changes including dates, and you should disclose the affected source code when distributing work. You cannot sublicense the relevant code. Modifications to LGPL 2.0-licensed libraries must be released under the LGPL 2.0 but the rest of the application can be under any licence.

Compatibility, linking and mixing: LGPL 2.0 is subsumed by GPL 2.0 but not compatible with GPL 3.0 and is linking-permissive. It permits dynamic linking of the library with both open source and proprietary applications. This allows mixing of LGPL 2.0 code with MIT, 2- and 3-clause BSD, Apache 2.0 and CDDL. If the library is statically linked with an application (i.e. the library becomes part of the application), the application must also be made available in an object code or other relevant format, allowing for modification and re-linking of the application.

Patent rights: Patent rights are not mentioned but are considered to be implicitly granted.

Remarks: Use dynamic linking in languages that also have static linking.

3.8 LGPL-2.1

Weak Copyleft

Description: Lesser GPL 2.1 is an older version of the LGPL permitting linking with modules under licences that are not considered free by the FSF. Clarifies linking terms of LGPL 2.0.

Requirements: The source code should include the original copyright notice and licence text. It should include notices of significant changes including dates, and disclose the affected source code when distributing the work. You cannot sublicense the relevant code. Modifications to LGPL 2.1-licensed libraries must be released under LGPL 2.1 but the rest of the application can be under any licence.

Compatibility, linking and mixing: Derivative works of LGPL 2.1 code must be licensed under the same version (or, if allowed by the licensor, a later version of LGPL) or GPL 2.0 or later. LGPL 2.1 allows dynamic linking from code under MIT, BSD (2- and 3-clause), Apache 2.0 and MPL licences. LGPL 2.1 is explicitly two-way compatible with the EUPL 1.2 licence.

Patent rights: Patent rights are not mentioned but are considered to be implicitly granted.

Remarks: Use dynamic linking in languages that also support static linking. Provide copyright and licence information in the UI if copyrights are already shown.

3.9 LGPL-3.0

Weak Copyleft

Description: Lesser GPL 3.0 is the current version of LGPL, permitting linking with non-free modules. It prohibits practices that restrict installing or running modifications.

Requirements: The source code should include the original copyright notice and licence text. It should include notices of significant changes including dates and disclose the affected source code when distributing the work. You cannot sublicense the relevant code. Modifications to LGPL 3.0-licensed libraries must be released under LGPL 3.0 but the rest of the application can be under any licence. If the library is statically linked, object code or a suitable mechanism for re-linking must be provided.

Compatibility, linking and mixing: Derivative works of LGPL 3.0 code must be licensed under LGPL 3.0 or GPL 3.0 or later if stated. LGPL 3.0 allows linking from code under permissive licences (MIT, BSD, Apache 2.0, MPL).

It is not compatible with GPL 2.0 without the “or later” clause. LGPL 3.0 is explicitly two-way compatible with the EUPL 1.2 licence. It supports both static and dynamic linking, provided re-linking is enabled.

Patent rights: Patent rights are explicitly granted under GPL 3.0 terms, which are inherited.

Remarks: Use dynamic linking in languages that also support static linking. Include installation information for consumer devices. Provide copyright and licence in the UI if copyrights are already shown.

3.10 MPL-1.1

Weak Copyleft

Description: Mozilla Public License 1.1 is a weak copyleft licence where copyleft applies at the file level, with some complex restrictions that make it incompatible with GNU GPL. A module covered by GPL and a module covered by MPL cannot legally be linked together although the MPL may include an option to choose another licence.

Requirements: The source code should include the original copyright notice and licence text. It should include notices of significant changes including dates. When distributing combined work, you must disclose the affected source code under the same licence.

Compatibility, linking and mixing: MPL 1.1 is not compatible with the entire GPL family including LGPL and AGPL licences. However, code under the Mozilla 1.1 licence can include code under Apache 2.0, MIT and BSD (2- and 3-clause) licences. It is linking-permissive.

Patent rights: MPL 1.1 includes specific provisions related to patents.

3.11 MPL-2.0

Weak Copyleft

Description: Mozilla Public License 2.0 is a weak copyleft licence applying at the file level, which permits relicensing to GPL except where explicitly denied.

Requirements: Authors must state that the code is under MPL and where to find the licence. The source code should include the original copyright notice but does not have to state significant changes. When distributing combined works, you must disclose the affected source code. You can sublicense binaries of aggregate works. Derivative works can be licensed under any future version of the licence.

Compatibility, linking and mixing: Software under previous versions of MPL can be upgraded to MPL 2.0. However, any software not already available under one of the GNU licences listed in the licence text must be marked as “Incompatible with Secondary Licences” in places where licence use is stated. This is detailed in “Exhibit B” of the licence text. MPL 2.0 is subsumed by GPL 2.0, LGPL 2.1, AGPL 3.0 and all later versions of those licences. You must ensure that files originally under MPL remain available under MPL’s terms as well. MPL 2.0-licensed code can be combined with code under Apache 2.0 and published under either licence. Code distributed under the Mozilla 2.0 licence can include code under ISC, MIT and BSD (2- and 3-clause) licences. It is linking-permissive.

Patent rights: Patent rights are explicitly granted.

Remarks: MPL 2.0 does not allow the use or mention of trademarks.

4 High Risk Licences

4.1 AGPL-3.0

Strong Copyleft, Network Protective

Description: Affero GPL 3.0 licence terms effectively comprise those of GPL 3.0, with an additional provision requiring users interacting with software over a network to be able to receive its source code. AGPL 3.0 and GPL 3.0-licensed code can be combined into a single program under the AGPL 3.0 licence. AGPL is recommended for network-based software but restricts future proprietary versions. If you modify AGPL software used over a network, you must provide access to the modified source code.

Requirements: Source code should include the original copyright notice and licence text. It should include notices of significant changes including dates. You must disclose affected source code when distributing combined work. You cannot sublicense the relevant code.

Compatibility, linking and mixing: AGPL 3.0 subsumes GPL 3.0 and is not linking-permissive.

Patent rights: Patent rights are explicitly granted under GPL 3.0 terms, which are inherited.

Remarks: AGPL's notable feature is the additional clause addressing usage over a network. It explicitly states that if users interact with the software over a network, the server must provide the source code for that version of the program.

4.2 GPL-1.0

Strong Copyleft

Description: This version served as the foundation for subsequent GPL versions but is no longer recommended for use due to various limitations and issues that were addressed in later versions.

Requirements: The source code should include the original copyright notice and licence text. It should include notices of significant changes including dates. Developers should disclose affected source code when distributing derived works, except for private use. Relevant code cannot be sublicensed.

Compatibility, linking and mixing: It is compatible only with GPL 1.0. Code under MIT, ISC and BSD 2-clause and 3-clause licences can be used in GPL 1.0-licensed works. It is not linking-permissive. It is recommended to use newer licences that provide stronger patent protections, clearer terms and better compatibility with other open source licences.

Patent rights: Patent rights are not mentioned.

4.3 GPL-2.0

Strong Copyleft

Description: Focuses on software licensing and distribution. It is considered very US-centric.

Requirements: Source code should include the original copyright notice and licence text. It should include notices of significant changes including dates. You should disclose affected source code when distributing derived works, except for private use. Relevant code cannot be sublicensed.

Compatibility, linking and mixing: It is compatible with GPL 2.0 only. FSF provides for original authors an upgrade option to GPL 3.0 or later for projects licensed under GPL 2.0. GPL 2.0-licensed code can be combined with AGPL 3.0-licensed code under AGPL 3.0. When distributing combined work with LGPL 2.1, you can choose to license it under either GPL 2.0 or LGPL 2.1. When distributing combined work with Apache 2.0, you must license it under GPL 2.0. Code under MIT, ISC and BSD 2-clause and 3-clause licences can be used in GPL 2.0-licensed works. It is not linking-permissive.

Patent rights: Patent rights are not mentioned but are considered to be implicitly granted.

Remarks: Has the IoT gap addressed by GPL 3.0. Interactive software must display copyright notices, warranty information, redistribution permissions and how to view the licence.

4.4 GPL-3.0

Strong Copyleft

Description: One major danger that GPL 3.0 blocks is tivoisation (also known as the IoT gap). Tivoisation means certain “appliances” (which have computers inside) contain GPL-covered software that you cannot effectively change because the appliance shuts down if it detects modified software. GPL 3.0 provides an option for licensors to add or upgrade to any later version of the GPL.

Requirements: The source code should include the original copyright notice and licence text. It should include notices of significant changes including dates. You should disclose affected source code when distributing any derived work, except for private use. Relevant code cannot be sublicensed.

Compatibility, linking and mixing: It is not linking-permissive. Code under GPL 3.0 can be combined with AGPL 3.0-licensed code and the resulting work is licensed under AGPL 3.0. GPL 3.0 is not compatible with GPL 2.0. However, FSF provides an option for projects licensed under GPL 2.0 to upgrade to GPL 3.0 or later. Code under GPL 3.0 can be combined with Apache 2.0 or LGPL 3.0-licensed code and the resulting work is licensed under GPL 3.0. You can combine code under GPL 3.0 with code under MIT, ISC and BSD 2-clause and 3-clause licences but must follow GPL 3.0 terms when distributing the combined work.

Patent rights: Patent rights are explicitly granted.

Remarks: Closes the IoT gap. Include installation information for consumer devices. Interactive programs must use their usual user interface to display copyright notices, warranty disclaimers, state that the program is covered by GPL 3.0, explain that there is no warranty and how to view a copy of the licence.

5 Glossary

Copyleft	A method making a program free, requiring all modified and extended versions to be free as well.
Derivative Work	A new, original product that includes aspects of an existing copyrighted work.
FOSS (Free and Open Source Software)	Software that is both free software and open source, ensuring users have the freedom to use, modify and distribute it.
FSF (Free Software Foundation)	A non-profit organisation that promotes computer user freedom and defends the rights of all free software users.
GPL (General Public License)	A widely used free software licence that guarantees users freedom to run, study, share and modify software.

IPR (Intellectual Property Rights)	Legal rights granted to creators or owners of works, including patents, copyrights and trademarks.
Licensing	The granting of permission to use intellectual property under defined conditions.
OSI (Open Source Initiative)	A non-profit organisation that promotes and protects open source software by certifying licences as Open Source Definition-compliant.
OSS (Open Source Software)	Software for which the source code is freely available and may be redistributed and modified.
Permissive Licence	A type of free software licence with minimal restrictions on software use, modification and distribution.
Software Licence	A legal instrument governing software use or redistribution; the UK spelling for the noun is “licence.” “License” is used within licence names.
Strong Copyleft	A licensing model that requires derivative works to adhere to the original work’s licence terms.
Weak Copyleft	A licensing model that allows linking with other software licensed under different terms, providing more flexibility than strong copyleft licences.

6 Licence Cheat Sheet

Table 1: Software licences that are frequently used in GÉANT projects or are otherwise significant

Licence	Patent Grant	Note
Low-Risk Licences (Permissive)		
Apache-2.0	Yes (defensive, broad)	Permissive and widely used. Grants patent rights for using the software. May require reciprocal grants.
Bouncy Castle License	Not mentioned	Similar to MIT. Primarily used for cryptographic libraries.
BSD 2-Clause	Not mentioned	Similar to MIT. Simple and widely used with minimal requirements.
BSD-3-Clause	Not mentioned	Similar to BSD 2-clause. Widely used. Includes a non-endorsement clause for promotional use.
BSD-4-Clause	Not mentioned	Includes an advertising clause. Less common.
BSL-1.0	Not mentioned, implicitly Yes	Business-friendly. Similar to MIT. Used for Boost C++ libraries.
CC0-1.0 / WTFPL / Unlicense	No for CC0-1.0	All dedicate works to the public domain. No restrictions but only Unlicense is open source.
CC-BY-4.0	No	Attribution licence for creative works. Not intended for software.
CC-BY-SA-4.0	No	Strong copyleft. Attribution and share-alike required. For creative works and documents.
CDDL-1.0	Yes (essential)	Derived from MPL 2.0.
CDDL-1.1	Yes (defensive, essential)	Minor update of CDDL 1.0. Adds patent infringement termination clause.
Golang BSD + Patents	Yes (defensive, broad)	BSD 3-clause with broad patent grant (like Apache 2.0).
ISC / OBSD	Not mentioned	Similar to MIT. Minimal restrictions.
MIT / X11	Not mentioned, implicit in USA	Simple and widely used. Minimal restrictions.
NUnit	Not mentioned, implicit in USA	Minimal restrictions. Used for the NUnit testing framework.

OpenSSL	Not mentioned	Mix of Apache 1.0 and BSD 4-clause. Includes specific requirements for OpenSSL libraries. Grants rights to essential patents.
Public Domain	Not mentioned	Not subject to copyright. No restrictions.
Python-2.0	Not mentioned	Legacy licence for the Python programming language.
Zlib	Not mentioned	Minimal restrictions. Used for the zlib compression library.
Medium-Risk Licences (Mostly Weak Copyleft)		
Artistic-1.0	No	Weak copyleft. Mainly used for Perl.
Artistic-2.0	Yes (defensive, essential)	Update of Artistic 1.0. Compatible with GPL 2.0.
EPL-1.0	Yes (defensive, essential)	Primarily for Eclipse projects. Grants rights to essential patents.
EPL-2.0	Yes (defensive, essential)	Similar to EPL 1.0. Widely used in open source projects.
EUPL-1.2	Yes (defensive)	Compatible with GPL. Multi-lingual. Highly compatible. Grants rights to essential patents.
GPL-2.0-with-classpath-exception	Not mentioned, implicitly Yes	GPL 2.0 with linking exception. Mainly used for Java.
LGPL-2.0	Not mentioned, implicitly Yes	Allows linking with non-GPL software. Ensures open source derivatives without affecting the using code.
LGPL-2.1	Not mentioned, implicitly Yes	Clarifies linking terms. Allows relicensing under GPL 2.0 or later.
LGPL-3.0	Yes (defensive, essential)	Prohibits restrictions on installing or running modified versions.
MPL-1.1	Yes (defensive, essential)	Semi-permissive, file-level. Allows combining with GPL code.
MPL-2.0	Yes (defensive, essential)	Flexible and widely used. File-level copyleft.
High-Risk Licences (Strong Copyleft and Network Protective)		
AGPL-3.0	Yes (defensive, essential)	GPL 3.0 with server-side source code disclosure requirement.
GPL-1.0	Not mentioned	Early version of GPL. Less common.
GPL-2.0	Not mentioned, implicitly Yes	Widely used. Incompatible with GPL 3.0 unless "or later" is included.
GPL-3.0	Yes (defensive, essential)	More explicit terms. Incompatible with GPL 2.0-only.

Additional explanations for patent grants descriptions:

- **Defensive termination** (in Apache 2.0, CDDL, Golang BSD + Patents, Artistic 2.0, EPL, EUPL 1.2, MPL and GPL 3.0 including its derivatives) automatically revokes granted patent rights if the licensee initiates patent litigation against the authors or contributors.
- **Broad** patent grants (like in Apache 2.0, Golang BSD + Patents) cover any patents owned by contributors that are necessary to use by the distributed code, its modifications or combinations with other software. These grants extend to potential future uses or modifications.
- **Essential** patent grants (in CDDL, GPL 3.0 family, EPL, MPL) cover patents that would necessarily be infringed by exercising the rights to use, make or sell the program as distributed. This means the rights under the licence cannot be exercised without infringing those patents. These grants apply only to patents that must be infringed to use the code as-is, limiting the grant strictly to the distributed code. Such grants do not cover patents that might be infringed through modifications or combinations beyond the original distribution.
- The **implicit** patent grant interpretation of GPL 2.0, LGPL 2.0 and LGPL 2.1 by some legal scholars and practitioners, as well as the less contested interpretation for MIT and BSD, arises from licence language about the freedom to use, modify and distribute the software. They argue that these freedoms would be meaningless without the underlying patent rights to exercise them. However, this remains a legal interpretation rather than an explicit provision stated in the licence text. At the same time, the implicit patent grants of BSD are less disputed.