

- **Un logiciel est libre** si et seulement si sa licence garantit les quatre libertés fondamentales :

**la liberté** d'utiliser le logiciel, **la liberté** de copier le logiciel, **la liberté** d'étudier le logiciel, **la liberté** de modifier le logiciel et de redistribuer les versions modifiées.

- **Le copyleft est** une méthode générale pour rendre **libre** un programme (ou toute autre œuvre) et obliger toutes les versions modifiées ou étendues de ce programme à être libres également. **Le copyleft s'applique** uniquement quand une personne veut redistribuer le programme. On est autorisé à faire des versions modifiées privées, sans aucune obligation de divulguer les modifications effectuées sur le programme s'il n'est distribué à personne. Le copyleft s'applique uniquement au programme et non à ses sorties. Par exemple, un portail Web utilisant une version modifiée privée d'un **CMS** sous GPL ne sera pas obligé de livrer ses sources. Cette situation est corrigée par l'[Afferro General Public License](#), une version de la GPL 3 qui ajoute un paragraphe pour accorder aux utilisateurs d'un programme accédé par un réseau les mêmes droits que les utilisateurs d'un programme installé localement.

[GNU General Public License \(GPL\), version 3 \(#GNUGPL3\) \(#GNUGPLv3\)](#)

C'est la dernière version de la licence publique générale GNU (GNU GPL) : il s'agit d'une licence de logiciel libre et d'un copyleft. Nous la recommandons pour la plupart des logiciels. Veuillez noter que la GPLv3 n'est pas, en soi, compatible avec la GPLv2. Cependant, la plupart des logiciels publiés sous GPLv2 autorisent d'utiliser aussi les termes de versions ultérieures de la GPL. Si c'est bien le cas, vous pouvez utiliser le code sous GPLv3 pour faire la combinaison désirée. Pour en savoir plus sur la compatibilité entre les licences GNU

[GNU General Public License \(GPL\), version 2 \(#GPLv2\)](#)

C'est la version précédente de la licence publique générale GNU (GNU GPL) : une licence de logiciel libre et un copyleft. Nous recommandons la [dernière version](#) pour la plupart des logiciels. Veuillez noter que la GPLv2 n'est pas, en soi, compatible avec la GPLv3. Cependant, la plupart des logiciels publiés sous GPLv2 autorisent d'utiliser aussi les termes de versions ultérieures de la GPL. Si c'est bien le cas, vous pouvez utiliser le code sous GPLv3 pour faire la combinaison désirée. Pour en savoir plus sur la compatibilité entre les licences GNU

[GNU Lesser General Public License \(LGPL\), version 3 \(#LGPL\) \(#LGPLv3\)](#)

C'est la dernière version de la licence publique générale GNU amoindrie (GNU LGPL) : une licence de logiciel libre, mais qui n'est pas un copyleft fort, car elle permet de lier des modules non libres. Elle est compatible avec la GPLv3. Nous la recommandons dans des [circonstances spéciales seulement](#). Veuillez noter que la LGPLv3 n'est pas en soi compatible avec la GPLv2. Cependant, la plupart des logiciels publiés sous GPLv2 autorisent d'utiliser aussi les termes de versions plus récentes de la GPL. Si c'est bien le cas, vous pouvez utiliser le code sous GPLv3 pour faire la combinaison désirée.

[GNU Lesser General Public License \(LGPL\), version 2.1 \(#LGPLv2.1\)](#)

C'est la version précédente de la licence publique générale GNU amoindrie (GNU LGPL) : il s'agit d'une licence de logiciel libre, mais pas aussi forte en tant que copyleft, car elle permet l'intégration de modules non libres. Elle est compatible avec les versions 2 et 3 de la GNU GPL. Nous recommandons généralement la [dernière version de la LGPL](#), [uniquement dans des cas spécifiques](#).

[GNU Affero General Public License \(AGPL\), version 3 \(#AGPL\) \(#AGPLv3.0\)](#)

La licence publique générale GNU Affero (GNU AGPL) est une licence de logiciel libre avec copyleft. Ses termes sont en fait ceux de la GPLv3, avec un paragraphe supplémentaire dans l'article 13, qui autorise les utilisateurs interagissant avec le logiciel sous licence sur un réseau à recevoir les sources du programme. Nous recommandons aux développeurs d'envisager l'utilisation de la GNU AGPL pour tout logiciel qui sera exécuté sur un réseau. Veuillez noter que la GNU AGPL n'est pas compatible avec la GPLv2. Elle est aussi techniquement incompatible avec la GPLv3, ce au sens strict : vous ne pouvez pas prendre de code publié sous la GNU AGPL et le transmettre ou le modifier sous les termes de la GPLv3, ou vice versa. Cependant, vous êtes autorisé à combiner des modules ou des fichiers sources séparés, publiés sous ces deux licences à la fois, dans un même projet, ce qui donnera à beaucoup de programmeurs toutes les permissions dont ils ont besoin pour faire les programmes qu'ils veulent.

[GNU All-Permissive License \(#GNUAllPermissive\)](#)

C'est une licence de logiciel libre permissive (voire laxiste) [lax, permissive] compatible avec la GNU GPL, que nous recommandons pour les fichiers README et autres petits fichiers d'aide des paquets GNU. Tous les développeurs peuvent l'utiliser dans des conditions similaires. Les anciennes versions de cette licence n'avaient pas la deuxième phrase avec la déclaration expresse d'absence de garantie. Cette même analyse s'applique aux deux versions.

[Licence Apache version 2.0 \(#apache2\)](#)

C'est une licence de logiciel libre, compatible avec la version 3 de la GNU GPL. Veuillez noter que cette licence n'est pas compatible avec la GPL version 2, car certaines de ses conditions ne figurent pas dans cette version de la GPL. Cela inclut des clauses de fin de brevet et d'indemnisation. La clause de fin de brevet est une bonne chose, c'est pourquoi nous recommandons la licence Apache 2.0 pour les programmes de taille importante, de préférence aux autres licences permissives (laxistes).

[Licence BSD modifiée \(#ModifiedBSD\)](#)

Il s'agit de la licence BSD d'origine, mais privée de sa clause publicitaire. C'est une licence de logiciel libre permissive (voire laxiste) sans copyleft, compatible avec la GNU GPL. Cette licence est quelquefois appelée « licence BSD à trois clauses ». La licence BSD modifiée n'est pas mauvaise, en tant que licence permissive (voire laxiste). Cependant, recommander la « licence BSD » présente un risque, même dans des cas spéciaux comme les petits programmes, car il pourrait facilement y avoir confusion, ce qui conduirait à utiliser la [licence BSD](#) d'origine, pleine de défauts. Pour éviter ce risque, suggérez plutôt l'emploi de la licence X11. La licence X11 et la licence BSD modifiée sont plus ou moins équivalentes. Toutefois, la licence Apache 2.0 est meilleure pour les programmes de taille importante car elle évite les entourloupes relatives aux brevets.

[CC0](#)

La CC0 est une licence Creative Commons permettant de verser une œuvre au domaine public. L'œuvre ainsi publiée est mise dans le domaine public dans les limites les plus larges permises par la loi. Si ce n'est pas possible pour une quelconque raison, la CC0 fournit aussi une licence permissive (voire laxiste) comme substitut. Le œuvres couvertes par cette licence laxiste sont compatibles avec la GNU GPL, tout comme celles du domaine public.

Si vous souhaitez mettre votre œuvre dans le domaine public, nous vous recommandons d'utiliser la CC0.

[Licence BSD clarifiée \(#clearbsd\)](#)

C'est une licence de logiciel libre, compatible avec la GPLv2 et la GPLv3. Elle est basée sur la [licence BSD modifiée](#), et ajoute un terme déclarant explicitement qu'elle ne vous octroie aucune licence de brevet. C'est pourquoi nous vous encourageons à faire attention quand vous utilisez des logiciels sous cette licence ; vous devez d'abord vous demander si le donneur de licence pourrait vouloir vous poursuivre pour violation de brevet. Si le développeur refuse de vous donner des licences de brevets dans le but de vous piéger, il serait sage d'éviter le programme.

[Licence FreeBSD \(#FreeBSD\)](#)

Il s'agit de la licence BSD d'origine dont la clause publicitaire et une autre clause ont été supprimées (elle est parfois appelée « licence BSD à deux clauses »). C'est une licence de logiciel libre permissive (voire laxiste) sans copyleft, compatible avec la GNU GPL.

[Mozilla Public License \(MPL\), version 2.0 \(#MPL-2.0\)](#)

C'est une licence de logiciel libre. Le paragraphe 3.3 assure sa compatibilité indirecte avec la GNU GPL version 2.0, la GNU LGPL version 2.1, la GNU AGPL version 3, ainsi que toutes les versions ultérieures de ces licences. Quand vous recevez un programme sous MPL version 2.0, vous pouvez faire une « création plus vaste » [Larger Work] qui combine ce programme avec du code régi par une des licences GNU précitées. Quand vous le faites, le paragraphe 3.3 vous donne la permission de distribuer le programme régi par la MPL sous les termes de la même licence GNU, à une condition : vous devez vous assurer que les fichiers initialement sous MPL restent disponibles sous les termes de la MPL.

Autrement dit, quand vous faites une combinaison de cette sorte, les fichiers qui étaient initialement sous MPL auront deux licences : la MPL et la licence GNU. Au final, la création plus vaste, globalement, sera couverte par la licence GNU. Les gens à qui vous fournirez cette combinaison auront la possibilité d'utiliser n'importe quel fichier initialement régi par la MPL sous les termes de cette licence, ou bien de distribuer tout ou partie de la création plus vaste sous les termes des licences GNU sans restriction supplémentaire.

Il est important de comprendre que la condition de distribution des fichiers couverts par la MPL s'applique seulement à l'auteur et distributeur initial de la création plus vaste. Si elle s'appliquait aussi aux destinataires, ce serait une restriction supplémentaire incompatible avec la GPL et l'AGPL. Cela dit, quand vous contribuez à un projet existant, [nous vous recommandons habituellement de garder la même licence pour les modifications](#), même quand vous n'y êtes pas obligé. Si vous recevez un programme régi par une licence GNU, ou quelques fichiers sont également sous MPL, vous ne devez enlever la MPL de ces fichiers que lorsqu'il y a pour cela une solide justification.

Vérifiez les avis de licence des logiciels sous MPL avant de faire une création plus vaste de cette manière. Ceux qui publient le programme original sous la MPL 2.0 peuvent choisir de ne pas faire usage de cette compatibilité en ajoutant une phrase dans les avis de licence, disant que « cette création est incompatible avec les licences secondaires » [the work is "Incompatible With Secondary Licenses"]. Tout logiciel qui inclut cet avis n'est pas compatible avec la GPL et l'AGPL.

Les logiciels couverts par les versions précédentes de la MPL peuvent être mis à jour vers la version 2.0, mais tout logiciel qui n'est pas déjà disponible sous l'une des licences GNU listées doit être signalé comme incompatible avec les licences secondaires. Cela signifie qu'un logiciel disponible uniquement sous les versions précédentes de la MPL reste incompatible avec la GPL et l'AGPL.

[Licence X11 \(#X11license\)](#)

Il s'agit d'une licence de logiciel libre permissive (voire laxiste) sans copyleft, compatible avec la GNU GPL. Les anciennes versions de XFree86 utilisaient la même licence, et quelques variantes actuelles de XFree86 l'utilisent également. Les licences ultérieures de XFree86 sont distribuées sous la licence [XFree86 1.1](#).

Cette licence est parfois appelée « licence du MIT » mais ce terme est trompeur : le MIT a publié ses logiciels sous diverses licences.

Cette licence est bien adaptée aux petits programmes. Un programme plus grand devrait normalement être placé sous copyleft, mais si vous tenez à lui donner une licence permissive (laxiste) nous recommandons la licence Apache 2.0 car elle protège l'utilisateur contre les entourloupes relatives aux brevets.

[Licence Expat \(#Expat\)](#)

Une licence de logiciel libre permissive (voire laxiste) sans copyleft, compatible avec la GNU GPL. D'une manière ambiguë, on l'appelle parfois « la licence du MIT ».

[Open Software License, toutes les versions jusqu'à 3.0 \(#OSL\)](#)

L'Open Software License est une licence de logiciel libre. Elle est incompatible avec la GNU GPL de plusieurs manières.

**Les versions récentes** de l'Open Software License ont une clause qui requiert de la part des distributeurs d'essayer d'obtenir un agrément explicite à la licence. Cela signifie que distribuer des logiciels sous licences OSL sur des sites FTP ordinaires, envoyer des correctifs sur une liste de diffusion ou stocker le logiciel dans un système de contrôle de version ordinaire, est une violation de la licence et vous rend possible de révocation de cette licence. Par conséquent, l'Open Software License rend très difficile de développer des logiciels en utilisant les outils courants pour les logiciels libres. Pour cette raison, et parce qu'elle est incompatible avec la GPL, nous recommandons qu'aucune version de l'OSL ne soit

utilisée pour quelque logiciel que ce soit. Nous vous demandons instamment de ne pas utiliser l'OSL pour les logiciels que vous écrivez. Cependant, il n'y a aucune raison de ne pas exécuter les programmes qui ont été publiés sous cette licence.

**Définition de l'« Open Source », version 1.0** « Open Source » implique plus que la simple diffusion du code source. La licence d'un programme « open-source » doit **correspondre aux critères suivants** :

- 1. Libre redistribution.** La licence ne doit pas empêcher de vendre ou de donner le logiciel en tant que composant d'une distribution d'un ensemble contenant des programmes de diverses origines. La licence ne doit pas exiger que cette vente soit soumise à l'acquiescement de droits d'auteur ou de royalties. (voir justification)
- 2. Code source.** Le programme doit inclure le code source, et la distribution sous forme de code source comme sous forme compilée doit être autorisée. Quand une forme d'un produit n'est pas distribuée avec le code source correspondant, il doit exister un moyen clairement indiqué de télécharger le code source, depuis l'Internet, sans frais supplémentaires. Le code source est la forme la plus adéquate pour qu'un programmeur modifie le programme. Il n'est pas autorisé de proposer un code source rendu difficile à comprendre. Il n'est pas autorisé de proposer des formes intermédiaires, comme ce qu'engendre un préprocesseur ou un traducteur automatique. (voir justification)
- 3. Travaux dérivés.** La licence doit autoriser les modifications et les travaux dérivés, et leur distribution sous les mêmes conditions que celles qu'autorise la licence du programme original. (voir justification)
- 4. Intégrité du code source de l'auteur.** La licence ne peut restreindre la redistribution du code source sous forme modifiée que si elle autorise la distribution de fichiers « patch » aux côtés du code source dans le but de modifier le programme au moment de la construction. La licence doit explicitement permettre la distribution de logiciel construit à partir du code source modifié. La licence peut exiger que les travaux dérivés portent un nom différent ou un numéro de version distinct de ceux du logiciel original. (voir justification)
- 5. Pas de discrimination entre les personnes ou les groupes.** La licence ne doit opérer aucune discrimination à l'encontre de personnes ou de groupes de personnes. (voir justification)
- 6. Pas de discrimination entre les domaines d'application.** La licence ne doit pas limiter le champ d'application du programme. Par exemple, elle ne doit pas interdire l'utilisation du programme pour faire des affaires ou dans le cadre de la recherche génétique. (voir justification)
- 7. Distribution de la licence.** Les droits attachés au programme doivent s'appliquer à tous ceux à qui le programme est redistribué sans que ces parties ne doivent remplir les conditions d'une licence supplémentaire. (voir justification)
- 8. La licence ne doit pas être spécifique à un produit.** Les droits attachés au programme ne doivent pas dépendre du fait que le programme fait partie d'une distribution logicielle spécifique. Si le programme est extrait de cette distribution et utilisé ou distribué selon les conditions de la licence du programme, toutes les parties auxquelles le programme est redistribué doivent bénéficier des droits accordés lorsque le programme est au sein de la distribution originale de logiciels. (voir justification)
- 9. La licence ne doit pas contaminer d'autres logiciels.** La licence ne doit pas apposer de restrictions sur d'autres logiciels distribués avec le programme qu'elle couvre. Par exemple, la licence ne doit pas exiger que tous les programmes distribués grâce au même médium soient des logiciels « open-source ». (voir justification)
- 10. Exemples de licences.** Les licences suivantes sont des exemples de licences que nous considérons conformes à la définition de l'Open Source : GNU GPL, BSD, X Consortium, et Artistic. C'est aussi le cas de la MPL.

dential to DFSG, except for addition of MPL and QPL to clause 10.

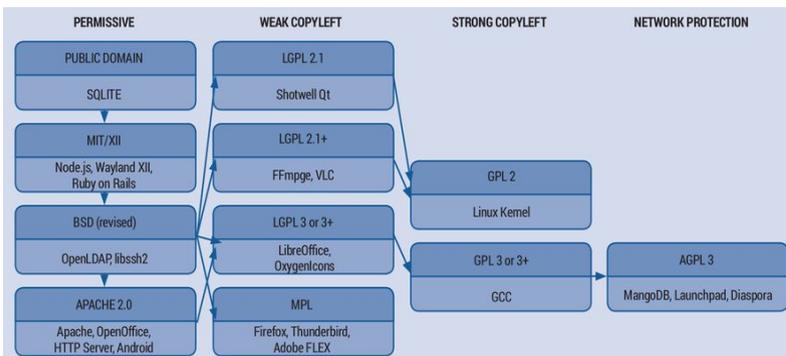
1.1 -- added LGPL to clause 10.

1.2 -- added public-domain to clause 10.

1.3 -- retitled clause 10 and split off the license list, adding material on procedures.

-> **Le droit d'auteur** est l'ensemble des droits dont dispose un **auteur** ou ses **ayants droit (héritiers, sociétés de production)** sur des œuvres de l'esprit originales et des droits corrélatifs du public à l'utilisation et à la réutilisation de ces œuvres sous certaines conditions. Il est composé de deux types de droits :

- **Le droit moral**, qui reconnaît notamment à l'auteur la **paternité de l'œuvre** et le respect de son **intégrité**. Dans certains pays, dont la France, il est perpétuel, inaliénable et imprescriptible
- **Les droits patrimoniaux**, qui confèrent un **monopole** d'exploitation économique sur l'œuvre, pour une **durée variable (selon les pays ou cas)** au terme de laquelle l'œuvre entre dans le « **domaine public** ». Le droit d'auteur aux États-Unis est l'ensemble des droits exclusifs dont dispose un **auteur** sur les **œuvres** de l'esprit aux **États-Unis**. Le droit d'auteur aux États-Unis fait partie du **droit fédéral (en)**; il est protégé par la **constitution des États-Unis** dans l'article I, section 8, clause 8, aussi connu comme la « Clause du Droit d'Auteur », laquelle dispose que : « Le Congrès aura le pouvoir ... Afin de promouvoir le progrès des sciences et des arts utiles, en assurant pour un temps limité, aux auteurs et inventeurs le droit exclusif à leurs écrits et découvertes respectifs. » Cette clause est à la base des **législations** américaines sur le **droit d'auteur** (« Science, Auteurs, Écrits ») et de celles sur les **brevets** (« Techniques, Inventeurs, Découvertes »). Un brevet confère un monopole sur une invention, par la voie de deux droits: Droit de Faire Art. L. 611-1 CPI: "Le brevet confère à son titulaire ou à ses ayants cause un droit exclusif d'exploitation" Droit d'Interdire à tout tiers: la fabrication, l'offre, la mise dans le commerce, l'utilisation ou l'importation ou la détention du produit objet du brevet l'utilisation du procédé, objet du brevet, ... l'offre, la mise dans le commerce ou l'utilisation ou l'importation ou la détention... du produit obtenu directement par le procédé objet du brevet L'atteinte à ces droits constitue le délit de contrefaçon.



-> **Durée des droits (France)** : individuelle: 70 ans après la mort ; collaboration: 70 ans après la mort du dernier auteur ; collective: 70 ans après la publication

**CAA/CLA** Copyright Assignment Agreement (CAA) cession agreement where a copyright holder surrender all their copyright sanctioned rights on some work to another party Contribution License Agreement (CLA) agreement where a copyright holder gives a license (usually non-revocable, possibly exclusive) to enforce specific copyright sanctioned rights to another party on paper, CAA are more powerful than CLA; but they only go as far as the legal system allows them, in most of Europe moral rights cannot be surrendered CLAs can be so broad to bede facto equivalent to CAAs key copyright right for policy reasons: the ability to relicense If a vendor participating in a FOSS project has, alone, the ability to relicense, strategy considerations based solely on the chosen FOSS license are completely moot.

**Academix Licenses** : Historically relevant subset of lax permissive licenses / The simplest licenses: very few restrictions / Reserving only attribution (keep names and copyright notice) / Available for all uses, including use in proprietary software / Originally written for and popularized by universities. **Examples: MIT, BSD, ISC**

**Lax permissive licenses** Include explicit grant of patent license (in modern variants) /

Available for almost all uses, including use in proprietary products. **Examples: Apache License**

**Reciprocal licenses** Require that derivative work maintains the same license / Usually require binary distributions to be accompanied by complete and corresponding source code / Also known as "strong copyleft" or just "copyleft" / Historically called "viral licenses", as a denigration tactic : If reciprocally licensed code is incorporated, then the application is "infected" and must be released as a whole under the same license **Examples: GPL, AGPL CC BY-SA (for non-software works)**

**Scope-limited reciprocal licenses** : Like reciprocal licenses, but with limits on the scope of which parts of a derived work fall under the license terms : changes to the main work falls under the license terms // additional works that happen to be used with/added to/embedded with the main work do not / They vary in the way the scope of the main work is limited / According to the denigratory analogy: "virality" is limited to the main work / Also known as: "weak copyleft" **Examples: MPL, CDDL, LGPL**

**8 business modèles à connaître**

**Advertising(publicité)**: Les pubs sont affichées directement (dans le logiciel ou sur son installateur) ou indirectement (sur le site ou les notices) ce qui génère du revenu. Le propriétaire du logiciel établit des contrats soit directement soit via un réseau de publicité, peut-être en utilisant un SDK. Dans la variation mobile de ce modèle, le gérant de l'appStore est aussi impliqué.

**Auxiliary Services**: Le revenu est généré par des services payants fournis avec le softwar plutôt que juste le software seul.

Ces services peuvent inclure l'implémentation, le support, maintenance, consultation, formation, traduction en d'autres langages ou même fourni des fichiers compilés en échange de frais (plutôt que je juste le code source).

**Corporate development and distribution (Développement interne en société et distribution)**: les organisations (commerciale ou non profit) paient des devs à pleins temps ou temps partiel pour personnaliser et améliorer le FOSS pour leur propre besoins dans l'organisations(société) Ensuite, ils distribuent leur changement à la communauté FOSS pour continuer à améliorer le produit FOSS. Pour les vendeurs hardwares, cela peut être un moyen de lancer des ventes ou réduire les coûts de maintenance et production. Il se différencie du modèle où les changements sont faits en internes et ne sont pas partagés !!

**Crowdfunding (cagnotte)**: le propriétaire du projet ou même un dev peut proposer une implémentation et un budget nécessaire à sa mise en oeuvre. Les personnes intéressées peuvent contribuer comme elles veulent. Si le budget attendu est atteint, alors le dev commit son boulot pour conclure le "projet".

**Double licence / Vente exceptionnelle** : Le produit est sous licence FOSS mais si quelqu'un veut l'utiliser sans adhérer aux termes de la licence FOSS, il peut payer.

Abonnements et donations : Tout est dit

**SaaS with distribution of server software** : Le coeur du software est sur un serveur. C'est un cloud. Il y a une partie gratuite ou payante selon l'offre. L'entreprise gère ses abonnements comme elle le veut.

**Update subscription** : Les users doivent payer annuellement ou à chaque MAJ, correction de bugs, etc.

**Les deux modèles les plus importants**

**Vente des données utilisateurs**: Collecter les données d'utilisations, le feedback des users, leurs préférences, etc... les devs peuvent utiliser ça pour développer de meilleurs app et features. Les apps FOSS peuvent utiliser un "framework de pubs".

**Certification du software** : Le software peut être fourni sous une licence FOSS mais nécessiter une certification pour utiliser une marque déposée (pas sûr de la traduction..).

Dans une des particularités de ce business model, l'OEM (Original Equipment Manufacturers) peut utiliser le software certifié dans ses appareils hardwares. Dans une autre version

encore, les proprios du projet enregistre un nom de marque (exemple : Moodle) dans toute la planète et ensuite crée un schéma basé sur la confiance ET les royalties où seuls les partenaires de licences peuvent utiliser le software ou nom de projet dans leur "PROMOTIONS AND SERVICE DESCRIPTIONS".