

3.1. Choisir la licence Open Source adaptée au besoin

Publié sur Guide Open Source (<http://guideopensource.info>)

3.1. Choisir la licence Open Source adaptée au besoin

Par le groupe de travail Syntec Informatique-FniLL

Créé le 10/04/2010 - 21:56

Soumis par [Benjamin Jean](#) [1] le lun, 10/04/2010 - 21:56

3.1.1. Tactique et technique dans le choix des licences Open Source

[Back to Top](#)

3.1.1.1. Stratégie dans le choix de la licence

Le choix de la licence traduit des intérêts juridiques, techniques et commerciaux. Le choix d'une licence Open Source suit un processus particulièrement simple à décrire :

La société doit adopter la licence la plus adaptée à son projet. L'existence de centaines de licences laisse présager un nombre tout aussi grand de situations particulières les ayant justifiées. Leur multiplicité préjudicie à leur compréhension, alors qu'il n'y a en réalité que quelques obligations qui différencient les licences les unes des autres¹ : les clauses copyleft, les clauses relatives aux usages par la communauté, les clauses relatives aux brevets, [DRM](#) [2], Tivoization², la langue utilisée, etc.

La société dresse donc, dans un premier temps, un **état des lieux** qui permet de constater que certaines licences ne peuvent pas être réutilisées en l'état en dehors de projets particuliers, que d'autres ne sont manifestement pas adaptées à ses intentions et, pour finir, que seules quelques-unes répondent en totalité ou partie à ses attentes.

La société peut encore faire le choix de la multilicence (décrite supra). On parle de multilicence lorsqu'une seule et même création est soumise à différentes licences : qu'elles soient toutes Open Source ou qu'au moins l'une d'elle le soit. Multiplier les licences Open Source peut parfois offrir des situations véritablement avantageuses : assurer une compatibilité au bénéfice de plusieurs licences, bénéficier de la renommée d'une licence, optimiser la cession de droits, etc.

Exemple : Le navigateur web Firefox, sous licence [MPL](#) [3]/[LGPL](#)/[GPL](#)³, est un très bon exemple de la souplesse qu'offrent les licences. En effet, le logiciel est construit de façon modulaire⁴ et gagne à conserver une licence qui intègre ce particularisme⁵. Cependant, afin de rassurer les développeurs et les communautés sur leur faculté de disposer du code pour d'autres projets, les licences LGPL et GPL s'ajoutent à la [MPL](#) [3]. De cette façon, le logiciel libre Firefox dispose d'une licence qui lui permet d'être compatible avec les licences les plus populaires tout en bénéficiant des développements sous la licence la plus adaptée aux projets de Mozilla – notamment une portée réduite qui amenuise l'aspect contraignant de son copyleft⁶.

Dans un second temps, une fois ce premier état des lieux réalisé, la société doit se poser un certain nombre de questions indirectement liées au projet : la licence doit-elle être connue et reconnue par les communautés ? Doit-on s'inspirer des choix opérés pour des projets concurrents analogues ? Certaines licences doivent-elles être proscrites ? Toutes ces questions ne trouvent de réponses qu'au cas par cas, en fonction de l'examen et de l'approche choisie par la société.

Incontestablement, une licence disposant d'une large notoriété participe avantageusement à la communication qui suivra la libération du logiciel. En revanche, le choix d'une licence similaire à un projet concurrent est un pari dont il faut bien mesurer les aléas : les contributions pouvant librement circuler d'un projet à l'autre, celui qui réunit la meilleure communauté risque de l'emporter et de détourner à son profit les contributeurs de son concurrent.

3.1. Choisir la licence Open Source adaptée au besoin

Publié sur Guide Open Source (<http://guideopensource.info>)

A l'inverse, bénéficiaire de l'expérience et de l'expertise déployée par une société concurrente peut justifier un tel choix.

- [1](#). Voir notamment Patrice-Emmanuel SCHMITZ, " Licence compatibility lists: could they compensate proliferation by developing circles of trust?" , à l'occasion d'[EOLE 2010](#) [4], au Parlement Européen
- [2](#). Néologisme définissant la création d'un système qui inclut des logiciels libres, mais utilise le matériel électronique pour interdire aux utilisateurs d'y exécuter des [versions](#) [5] modifiées. Source [Wikipedia](#) [6]
- [3](#). Après avoir été quelque temps uniquement sous [MPL](#) [3], il a été estimé que cette seule licence n'était pas suffisante à motiver les contributions.
- [4](#). Chaque nouveau [composant](#) [7] apporte une nouvelle fonctionnalité autour d'un noyau. De nombreux projets Open Source adoptent cette structure qui permet notamment de réparer efficacement les tâches de développement.
- [5](#). En effet, la [MPL](#) [3] permet l'adjonction de greffon (plug-in) ou toute autre extension au logiciel tant que chaque fichier contenant du code sous [MPL](#) [3] soit exclusivement sous [MPL](#) [3]. Cette tolérance permet une plus large diffusion du logiciel.
- [6](#). La conséquence logique d'une telle multilicence est d'ailleurs que le copyleft final est celui le moins contraignant de l'ensemble des licences.

[Back to Top](#)

3.1.1.2. Personnalisation de la licence

Dans l'hypothèse où un logiciel est distribué sous une licence libre déjà connue il peut être opportun d'ajouter une interprétation relativement aux clauses des licences Open Source. Lorsque l'un des termes de la licence est source d'interprétation, celui qui choisit la licence peut lever l'ambiguïté qu'elle pourrait engendrer en donnant une portée bien définie à cette clause.

Ainsi, il y a création d'un faisceau d'indices qui pourrait donner des indications aux juges lors d'un litige dont la résolution passe par l'interprétation de la licence. Ainsi le juge pourrait valider l'interprétation définie par l'entreprise qui recourt à la licence. De nombreuses clauses imprécises ou équivoques peuvent être interprétées de la sorte lors du choix de la licence.

L'usage des exceptions est une autre technique plus radicale qui consiste à modifier la licence de base en ajoutant, dans une clause jointe à la licence ou insérée dans les en-têtes, une spécificité qui déroge aux termes initiaux avec pour effet de rendre la licence finale plus ou moins contraignante. Il est à noter qu'en l'absence de stipulation contraire, une clause additionnelle permissive pourra être supprimée par tout licencié au moment de la redistribution d'une copie du logiciel¹.

La maîtrise de plus en plus fine de la pratique contractuelle liée aux licences Open Source conduit aujourd'hui à conseiller une pratique réfléchie et adaptée de ces exceptions et interprétations. Les premières confèrent plus de souplesse aux licences², les secondes assurent une plus grande sécurité juridique. Un bon usage de celles-ci peut permettre de prévoir et de résoudre a priori la plupart des situations préjudiciables : problèmes d'incompatibilité, failles au sein des licences, etc.

- [1](#). Ceci vaut bien sûr pour les licences dites permissives, mais aussi pour toute [licence copyleft](#) [8] puisque le copyleft ne perpétue que les termes stricto sensu de la licence.
- [2](#). La troisième version de la [GNU GPL](#) [9] invite même dans un article dédié à user de cette faculté sur ses propres contributions (en distinguant d'une part les permissions additionnelles qui peuvent être ajoutées sans limitation et les termes supplétifs à tirer parmi une liste de sept types de clauses).

[Back to Top](#)

3.1. Choisir la licence Open Source adaptée au besoin

Publié sur Guide Open Source (<http://guideopensource.info>)

3.1.2. Types de licences Open Source

Les développements qui suivent visent à étudier de manière générale les licences *Open Source* par typologie et d'effectuer un zoom sur la [GNU](#) [10] *General Public License*.

[Back to Top](#)

3.1.2.1. Typologies des licences

La rédaction des licences Open Source n'étant pas toujours l'œuvre de juristes et émanant de systèmes juridiques hétérogènes, on s'aperçoit que certaines clauses sont inutiles au regard des règles de droit d'un pays, alors que d'autres doivent être interprétées pour être contextualisées.

Parmi les licences Open Source et afin d'avoir une vision plus globale du système, plusieurs classifications peuvent être croisées¹.

1 Classification classique : [licence copyleft](#) [8] versus [permissive](#)²

L'utilisation du terme **copyleft** s'entend des licences qui rendent persistantes les libertés consenties en astreignant les utilisateurs subséquents à concéder systématiquement les mêmes libertés. Concrètement, une cession non exclusive est demandée, les contributeurs restant titulaires de leurs droits et libres d'exploiter leur contribution par ailleurs.

On parle indifféremment de réciprocité ou de copyleft. Dans cette situation, c'est l'intérêt de l'utilisateur final qui prévaut sur la liberté de celui qui diffuse l'œuvre.

Deux mécanismes sont aujourd'hui utilisés :

- imposer l'utilisation d'une licence particulière (à l'instar de la [GNU GPL](#) [9], à son propre compte)
- obliger à conférer les mêmes libertés (comme le fait la [LAL](#) [11] : les droits cédés devront se retrouver dans la licence finale — qu'il s'agisse de la [LAL](#) [12] ou de toute autre). Il en résulte une relation de confiance qui sécurise et favorise les collaborations entre professionnels.

À l'inverse, on parle de licence **permissive** lorsque seules les obligations de celui qui reçoit l'œuvre — comme celles en matière de brevets pour la licence Apache — doivent être transmises, laissant le contributeur libre d'en ajouter d'autres lors du transfert aux utilisateurs ultérieurs (le logiciel redistribué perd souvent les libertés qui lui étaient attachées). Ces licences sont traditionnellement assimilées à des renonciations et le statut des œuvres est souvent proche de celui des œuvres tombées dans le domaine public, puisqu'elles n'imposent en règle générale que le respect de la paternité (avec les habituelles clauses d'exclusion ou limitation de garanties et de responsabilité). Cette relative liberté les fait coexister sans anicroche puisqu'il est très simple pour ces licences d'être compatibles en perpétuant simplement les obligations initiales.

2 Classification historique

• **Les licences [GNU](#) [10] / philosophiques** : il s'agit des licences publiées par la Free Software Foundation et plus généralement toutes celles qui partagent son esprit et sa philosophie. Les plus utilisées sont celles de la [FSF](#) [13] : la première de la famille est la [GNU](#) [10] *General Public License* — publiée en 1989, modifiée en 1991 et 2007, sa petite sœur est la [GNU](#) [10] *Library General Public License* — renommée *lesser GPL* — et leur cousine à destination de la [documentation](#) [14], la [GNU](#) [10] *Free Documentation* [14] licence. Il y a deux nouvelles entrantes : la [GNU](#) [10] *Affero GPL* depuis fin 2007 et la *Simpler FDL* qui devrait suivre prochainement. À l'origine, le langage de ces licences, très proche de celui des développeurs, est empreint d'une intention forte et d'une portée parfois complexe à définir.

3.1. Choisir la licence Open Source adaptée au besoin

Publié sur Guide Open Source (<http://guideopensource.info>)

La réécriture récente des licences ajoute à cette famille de licences des [versions](#) [5] beaucoup plus juridiques et complexes. Elles s'opposent à toute réappropriation du code grâce à leur copyleft qui impose que tout logiciel dérivé — basé sur, ou constituant un tout avec le logiciel — soit lui-même soumis à cette licence. Les sociétés intéressées par l'alternative du « Libre » hésitent souvent à recourir à l'utilisation de ces licences aux implications extensives et parfois incertaines.

- **Les licences académiques / universitaires.** Elles sont en large partie à l'origine du développement de l'infrastructure d'Internet. Pour exemple, le système de nom de domaine BIND, le protocole TCP-IP et Sendmail sont tous des standards de facto, issus de ces licences permissives. On y retrouve l'idée d'un partage des connaissances « sans condition » issu des universités américaines. Elles sont fréquemment formulées d'une façon courte et claire. Elles consistent généralement en l'énumération de la totalité des droits conférés, une obligation de respecter la paternité de l'œuvre et une exclusion de responsabilité et de garantie. Un bon exemple est la licence [BSD](#) [15] — pour Berkeley Software Distribution.

- **Les licences communautaires.** Elles sont principalement issues de projets libres qui, devenus populaires, choisissent de rédiger et d'utiliser leur propre licence. Très spécifiques puisqu'intimement liées à un projet et à son déroulement, elles sont en principe peu juridiques et trop souvent susceptibles d'interprétations hasardeuses. Les deux principales sont la licence Artistic et la licence Apache. Ce sont essentiellement des licences permissives, mais leur spécificité les rend difficiles — voire impossibles — à concilier avec la plupart des licences copyleft.

- **Les licences institutionnelles :** Elles furent introduites par des sociétés intéressées par le développement coopératif de leur produit selon le modèle de l'Open Innovation. La première, la plus symptomatique est la Mozilla Public License ([MPL](#) [3]), rédigée par la firme Netscape pour la libération du code de son navigateur. Ces licences sont précises et très complètes, ont une étendue définie et emblématique du mouvement Open Source.

3 Par domaine

Les licences Open Source sont aujourd'hui utilisées dans de nombreux domaines : les logiciels bien sûr, mais aussi les encyclopédies (on pense à Wikipedia), les livres, la musique et bientôt tout type d'œuvres. La majeure partie des licences Open Source trouve leur fondement dans une application particulière pour un domaine artistique bien déterminé. De ce fait, il est très délicat et déconseillé d'utiliser des licences rédigées pour un domaine dans un autre.

Par exemple, le formalisme extrêmement contraignant de la [GNU GPL](#) [9] n'est pas du tout adapté à la musique (le texte entier de la licence doit être attaché ; et imaginez qu'il faille distribuer le master et les sources en plus du [CD](#) [16]...).

Inversement, des licences comme les Creative Commons ont été écrites pour la musique, le film, les livres, etc., mais pas pour le logiciel. Très logiquement, licencier un logiciel sous Creative Commons — par exemple une [CC](#) [17]-By-SA pour prendre une licence Open Source assez similaire à la [GNU GPL](#) [9] — n'est pas conseillé puisque tous les aspects propres au logiciel sont inexistantes : la distribution du [code source](#) [18] n'étant notamment pas envisagée.

Enfin, quelques licences, plutôt rares, ont été rédigées avec l'ambition de s'étendre à l'ensemble des créations couvertes par le droit d'auteur, voire toute la propriété littéraire et artistique : l'[OSL](#) [19] et la [LAL](#) [12] sont deux très bons exemples avec des origines bien différentes.

Il est néanmoins nécessaire dans certaines situations, notamment en présence d'œuvres dites multimédias — regroupant de nombreux types d'œuvres — de réfléchir à l'opportunité d'utiliser une licence pour le tout dans un objectif d'uniformisation ou de privilégier une approche modulaire (en utilisant une licence pour chaque type d'œuvre).

Il n'y a encore une fois ni de bonnes ni de mauvaises réponses : tout dépend de l'espèce.

3.1. Choisir la licence Open Source adaptée au besoin

Publié sur Guide Open Source (<http://guideopensource.info>)

4 Par liberté

Sous la coprésidence de Valérie-Laure BENABOU et Joëlle FARCHY la commission spécialisée du [CSPLA](#) [20] portant sur la mise à disposition ouverte des œuvres de l'esprit a publié en juin 2007 un rapport reprenant une distinction issue des travaux de thèse sur les œuvres libres de Mélanie CLEMENT-FONTAINE. On y retrouve la classification selon le degré de libertés conférées :

- Les licences qui offrent une liberté pérenne. Il s'agit des licences disposant d'un copyleft : l'œuvre et ses dérivées sont libres ;
 - Les licences qui offrent une liberté fragile. Il s'agit ici des licences permissives qui autorisent la propriété par un tiers d'une œuvre dérivée — les contributeurs n'ont donc aucune garantie de pouvoir à leur tour bénéficier des contributions ultérieures ;
 - Les licences qui offrent une liberté asymétrique. Il s'agit ici des licences qui créent un déséquilibre au profit de celui qui a initialement le choix de la licence : avec l'exemple de la licence [CC](#) [17]-By-NC qui interdit l'exploitation commerciale de l'œuvre Open Source, mais présente une souplesse supérieure dans la réutilisation.
-
- [1.](#) Une description des familles de licences Open Source et de leurs grandes caractéristiques figurent dans un tableau reproduit en Annexe 3.
 - [2.](#) Voir : (petit) Guide à l'usage des licences libres, description des différents types de licences libres et des conséquences qui leur sont attachées, Intervention de Benjamin JEAN, lors de la Matinée Juridique du Syntec Informatique du 14 mars 2008.

[Back to Top](#)

3.1.2.2. L'exemple de la GNU GPL

Pour comprendre l'évolution des licences Open Source, il faut suivre l'évolution de la [GNU](#) [10] General Public Licence [1](#).

S'appliquant évidemment au logiciel et à la [documentation](#) [14] technique l'accompagnant, la licence GPL fixe les conditions de diffusion : les logiciels sous licence GPL peuvent librement être copiés, adaptés et distribués de quelque manière que ce soit, à la seule condition de rester soumis aux mêmes termes. La question de l'interprétation de la [GNU GPL](#) [9] v2 pose problème en ce qu'elle émane d'un système de droit anglo-saxon, mais la troisième version se détache de ce droit au profit d'une terminologie contractuelle plus conforme au droit français. L'objectif est de créer un « fonds commun auquel chacun peut ajouter, mais duquel personne ne peut retrancher », comme aime à le répéter Eben Moglen, l'avocat de la [FSF](#) [13] et directeur de la [SFLC](#) [21].

- Le [code source](#) [18] du logiciel diffusé sous licence GPL est communiqué à l'utilisateur. Mais une condition s'impose : le [code source](#) [18], auquel l'utilisateur a eu accès, doit être également transféré en cas de diffusion. Cet utilisateur ne saurait de sa seule initiative, et donc sans l'autorisation du créateur du logiciel, distribuer celui-ci sans son [code source](#) [18]. En d'autres termes, il ne saurait introduire, dans la chaîne de diffusion du logiciel, une forme d'appropriation de facto en se réservant le [code source](#) [18]. Il irait ainsi contre la volonté du premier programmeur et ne respecterait pas les termes de la licence.
- Compte tenu de la possibilité qui lui est faite d'accéder librement au [code source](#) [18], l'utilisateur peut modifier celui-ci, voire modifier le logiciel dans son entier. C'est ici une considérable exception apportée à l'interdiction de la décompilation traditionnelle, comme nous le verrons plus loin, interdiction visant précisément à empêcher toute modification du [code source](#) [18]. Par contre, cette possibilité d'examen et de modification du [code source](#) [18] implique une obligation : l'utilisateur doit préciser dans une notice les adaptations dont il est l'auteur, la date de ses apports, et son identité. Chaque programmeur successif participe à l'enrichissement du logiciel, et par principe son droit au

3.1. Choisir la licence Open Source adaptée au besoin

Publié sur Guide Open Source (<http://guideopensource.info>)

nom sera respecté.

- Un logiciel ainsi modifié peut également être distribué, mais cette nouvelle diffusion doit également respecter les stipulations de la licence GPL : elle doit être diffusée sous les mêmes termes : une fois qu'un logiciel a été cédé en vertu de ses stipulations, il est interdit de réintroduire un quelconque élément « propriétaire », c'est-à-dire qu'on ne peut se réserver le résultat d'une modification, ni la diffuser à titre onéreux, ni même introduire des portions de code faisant l'objet d'une réservation (par le brevet ou le droit d'auteur) dans les adaptations successives, dès lors que l'on distribue ces dernières.

Cette licence se veut la plus ouverte, mais comporte bel et bien des obligations pour l'utilisateur :

- Obligation, pour toute redistribution et toute modification du logiciel initial, d'adopter la licence GPL et de la diffuser en même temps que le logiciel ;
- Obligation pour un programmeur travaillant à partir d'un logiciel libre cédé sous cette licence, de soumettre sa création dérivée à la licence GPL ;
- Obligation de porter à la connaissance des tiers et de tout cessionnaire le nom des programmeurs qui sont déjà intervenus sur le [code source](#) [18] ;
- Obligation d'informer tout cessionnaire ultérieur du logiciel ou de ses modifications des règles de la licence GPL.

Deux vagues successives (la première le 29 juin, et la seconde le 9 novembre 2007), ont respectivement donné lieu à publication des nouvelles [versions](#) [5]2 des [GNU GPL](#) [9] (v3), [GNU LGPL](#) [22] (v3) et de la [GNU](#) [10] Affero GPL (v3) (dite aussi AGPL).

Sur la forme, les trois licences s'affichent avec une compatibilité parfaite au profit de la [GNU GPL](#) [9] et s'offrent une modularité qui permet de faciliter leur utilisation conjointe. Si les améliorations ont pesé sur la lisibilité des licences, la [FSF](#) [13] a fait attention à conserver leur similitude. Ainsi, la [GNU LGPL](#) [22] apparaît finalement comme une exception à la [GNU GPL](#) [9] (en se concentrant sur les droits supplémentaires accordés, et renvoyant à la [GNU GPL](#) [9] pour le reste de ses termes). De même, l' Affero GPL ne se distingue de la [GNU GPL](#) [9] que par une clause qui lui permet de s'adapter aux spécificités des logiciels voués à être utilisés via le réseau sans avoir à être distribués. Ces derniers devront alors permettre aux utilisateurs de bénéficier de la version du logiciel utilisée par ledit service, comblant ainsi ce que l'on avait dénommé l'« [ASP](#) [23] loophole ». Enfin, un travail de rédaction a aussi permis d'internationaliser les licences et d'assurer leur validité pour l'ensemble des systèmes juridiques.

- Sur le fond, les changements sont nombreux et ne peuvent qu'être listés ;
- Une « obligation de cohérence » interdit à l'utilisateur de limiter par d'autres voies, ou droits exclusifs concurrents, les libertés qu'il concède par la licence : que ce soit à l'égard des Mesures Techniques de Protection, des limitations matérielles (Tivoïsation) ou encore des brevets ;
- Un encadrement des accords parallèles portant sur des brevets³ agit tant du côté du promettant (qui s'obligerait à ne pas opposer ses brevets) qu'à celui qui en bénéficierait ;
- Une modularité est incorporée afin de permettre d'intégrer d'autres briques logicielles sous licence à minima libre (disposition profitant à la compatibilité avec des licences comme LaTeX, etc.) ;
- Des adaptations technologiques (comme la distribution par peer to peer).

Garante du rôle qu'elle s'est donnée, la Free Software Foundation assure par ces mises à jour la pérennité des libertés qu'elle prône, sans aucune concession.

- [1.](#) Voir la version originale sur <http://www.gnu.org/copyleft/gpl.html> [24] et l'analyse de ses différentes [versions](#) [5] sur <http://www.vvlibri.org/>
- [2.](#) Voir la « [Rétrospective 2007 sur le logiciel libre et les sujets afférents](#) [25] » publiée par l'April
- [3.](#) En réponse à l'accord Novell/Microsoft

3.1. Choisir la licence Open Source adaptée au besoin

Publié sur Guide Open Source (<http://guideopensource.info>)

[Back to Top](#)

3.1.3. Enjeux de la compatibilité entre licences

Sur cette question, voir notamment : Benjamin JEAN, « Option Libre : compatibilité entre contrats », sous la direction de Michel VIVANT, ERCIM, 2006

[Back to Top](#)

3.1.3.1. Licence Unique

La question « particulière » est celle de la compatibilité des licences. Si la démarche qui précède permet de s'orienter vers le choix optimal d'une licence, celle-ci peut se révéler impossible à mettre en œuvre si le produit « final » intègre des composants soumis à d'autres licences, aux termes incompatibles lors de la distribution ou de l'utilisation du logiciel.

Il n'existe pas de facto de licences dominantes : toutes ont la même force obligatoire et le licencié doit renoncer à exploiter une création lorsqu'il ne peut pas respecter l'ensemble de ses engagements.

La seule condition pour distribuer sous licence unique un projet comprenant des composants sous d'autres licences¹ est que cette distribution soit permise par l'ensemble des autres licences.

Après avoir été un frein majeur aux échanges entre projets libres pendant de longues années, le problème de l'incompatibilité entre licences semble se résorber progressivement grâce aux licences « de nouvelles générations » :

- La situation la plus commode réside en la compatibilité expresse : soit la licence prévoit que le logiciel peut indifféremment être redistribué selon ses propres termes ou selon ceux d'une licence déterminée², soit la compatibilité est limitée aux seules situations où l'ajout d'un [composant](#) [7] sous une autre licence Open Source impose de redistribuer le tout sous cette unique autre licence³.
- En l'absence d'une telle compatibilité, il convient de procéder à une analyse logique du contexte : l'enjeu consiste à gérer les flux de droits et d'obligations de la société pour savoir ce qu'elle peut faire ou ne pas faire : compatibilité induite. Lorsque l'on souhaite céder une contribution pour laquelle on est seulement licencié, la question se résume à s'assurer que l'on est bien en mesure de céder tous les droits que la licence confère et à veiller que l'on n'oblige pas moins l'utilisateur final que l'on est soi-même obligé⁴.

La dernière version de la [GNU GPL](#) [9] (V 3) s'appuie sur ce principe pour accueillir en son sein une modularité qui lui permet d'assurer la compatibilité à l'égard d'autres licences Open Source auparavant incompatibles⁵ : en permettant, sur chaque nouvelle contribution, d'une part, l'ajout de « permissions additionnelles » et, d'autre part, l'ajustement par l'utilisation de certains « termes supplétifs » limitativement autorisés. Le problème n'est néanmoins pas tout à fait résolu puisqu'il faudra ensuite confronter chaque licence aux différentes clauses, au cas par cas, afin d'être certain de leur compatibilité⁶.

Cette réflexion peut se traduire mathématiquement grâce à la théorie des ensembles sous la forme qui suit⁷ : la licence B est dite compatible (c'est-à-dire qu'elle peut être utilisée pour redistribuer le logiciel en respectant les termes de la licence originale A) si et seulement si l'ensemble des droits de la licence B est inclus dans la licence A et que l'ensemble des obligations de la licence A est inclus dans la licence B. Ainsi, une licence est compatible avec elle-même, une [licence copyleft](#) [8] n'est pas forcément compatible avec une licence permissive et une licence permissive ne l'est forcément pas à l'égard d'une [licence copyleft](#) [8] .

Attention, la version des licences constitue dans certains cas un élément déterminant :

3.1. Choisir la licence Open Source adaptée au besoin

Publié sur Guide Open Source (<http://guideopensource.info>)

même si la plupart permettent une compatibilité ascendante au bénéfice des nouvelles [versions](#) [5] de la licence, certaines autorisent aussi aux auteurs de licences Open Source de figer la licence à une version spécifique⁸. Il y a alors un copyleft en deux temps : le premier caractérisant l'étendue de la licence et le second déterminant la version sur chaque contribution.

La situation la plus confortable consiste à rester dans la même « famille de licences » : il en existe plusieurs et celles-ci couvrent en théorie un large panel de choix tout en assurant une compatibilité parfaite entre elles⁹.

- ¹. Par exemple, distribuer sous [GNU GPL](#) [9] v3 une solution qui comporte des composants sous licence [BSD](#) [15] et Apache.
- ². C'est notamment ce que faisait la LGPL v2 au bénéfice de la [GNU LGPL](#) [22].
- ³. Ce que l'on retrouve notamment dans l'European Public Licence au profit de cinq licences que sont l'[OSL](#) [19] v. [26] 2.1 et v. [26] 3.0, la CPL v. [26] 1.0, l'[EPL](#) [27] v. [26] 1.0, la [CeCILL](#) [28] v. [26] 2.0 et la [GNU GPL](#) [9] v. [26] 2.0
- ⁴. Cette compatibilité est aujourd'hui de mieux en mieux expliquée, comme le démontre un papier récent de la Software Freedom Law Center qui détaille la procédure permettant de conserver la licence [BSD](#) [15] sur du code distribué sous Licence [GNU GPL](#) [9].
- ⁵. Comme les licences plus permissives comme les licences Apache, LaTeX, etc.
- ⁶. Avec le risque de divergence d'interprétation, comme c'était déjà le cas les licences GPL et Apache.
- ⁷. Sur les questions de compatibilité entre licences, voir Benjamin JEAN, Option Libre : « compatibilité entre contrats », Op . Cit., 2006.
- ⁸. Si on prend l'exemple de la [GNU GPL](#) [9] et du noyau Linux : toutes les contributions de Linus TORVALD sont distribuées sous [GNU GPL](#) [9] « version 2 seulement », ainsi, même si d'autres parties autorisent la distribution sous [GNU GPL](#) [9] version 3, le noyau Linux ne pourra pas être distribué sous la troisième version de la [GNU GPL](#) [9] tant que les composants sous GPL v2 n'auront pas été réécrites ou que leur auteur aura consenti à modifier la licence !
- ⁹. La plus connue est certainement la famille des licences [GNU](#) [10] (GPL, LGPL, AGPL, etc.), mais bien d'autres peuvent être citées pour l'exemple : [CeCILL](#) [28] (-A, -B, -C), [OSL](#) [19] ([OSL](#) [19], AFL), etc.

[Back to Top](#)

3.1.3.2. Cas d'incompatibilité et solutions alternatives

Certains conflits ne peuvent contractuellement être résolus. Si l'œuvre litigieuse s'avère trop complexe pour être réécrite, il faudra alors renoncer à la distribution ou à l'utilisation du tout¹³⁴.

Il reste néanmoins quelques autres solutions : contacter le ou les auteurs d'un [composant](#) [7] litigieux pour leur demander une dérogation, une exception ou un changement de licence ; acheter une licence commerciale à ces derniers ; se faire céder la titularité de leurs droits² ou contourner matériellement le problème posé par la licence.

Le premier expédient possède l'avantage de la simplicité, mais son succès dépendra du nombre d'auteurs du [composant](#) [7], de leur personnalité et de ce que l'entreprise est prête à investir dans le projet. Les deux autres mesures dépendront aussi totalement de la volonté des titulaires de droits. Le dernier procédé se révèle le plus précaire puisqu'il nécessite une connaissance parfaite des licences afin d'estimer ce qui peut matériellement être permis et ce qui ne le serait pas (certaines formes de distribution³, certains modes de distribution⁴ et certaines constructions du logiciel⁵ permettent par exemple d'alléger les contraintes).

En résumé, le processus visant à vérifier la compatibilité expresse entre une licence Open Source A (d'origine) et une licence B (licence finale) est le suivant :

- Considérons que les droits ne posent pas de problème puisqu'à minima la Licence A est libre.

3.1. Choisir la licence Open Source adaptée au besoin

Publié sur Guide Open Source (<http://guideopensource.info>)

- Penchons-nous sur les obligations :
 1. la licence B contient toutes les obligations de la licence A - modulation des variations éventuellement tolérées par la licence,
 2. la licence A n'oblige pas à la redistribution sous une licence particulière OU elle permet expressément la redistribution sous la Licence B (ou une licence qui lui soit compatible).
- Considérons une [licence copyleft](#) [8] comme une licence qui a comme obligation de conserver les droits lors de la redistribution, alors : si la licence A est une licence Open Source copyleft , la licence B devra aussi être libre et copyleft .

Plus généralement, la condition de compatibilité est donc atteinte si l'ensemble des droits accordés par la licence absorbante B est inclus dans l'ensemble des droits conférés par la [licence compatible](#) [29] A et que l'ensemble des obligations imposées par la licence absorbante B.

- [1.](#) Sauf bien sûr à trouver un remplacement pour la partie litigieuse.
- [2.](#) On utilise souvent l'appellation américaine de Copyright assignment pour désigner ces cessions de droits. Il en existe de multiples sortes : La cession pure et simple sans contrepartie (par exemple comme condition à l'intégration d'une contribution dans un projet) ou les cessions finalisée (c'est-à-dire que l'utilisation par le cessionnaire est clairement délimitée par la cession : par exemple, toute distribution dont la contribution doit être réalisée sous licence open source).
- [3.](#) Par exemple exécutable, tant que les sources des modifications sont disponibles par ailleurs.
- [4.](#) On pense notamment aux logiciels dont seul le client est distribué alors que le serveur est hébergé chez l'éditeur.
- [5.](#) Par exemple la construction modulaire pour l'utilisation de composants sous licence du type [MPL](#) [3], voire LGPL.

[Back to Top](#)

URL source: <http://guideopensource.info/guide/choisir-licence-open-source-adaptee-besoin>

Liens:

- [1] <http://guideopensource.info/user/1>
- [2] <http://guideopensource.info/glossary/term/30>
- [3] <http://guideopensource.info/glossary/term/49>
- [4] <http://www.eolevent.eu>
- [5] <http://guideopensource.info/glossary/term/14>
- [6] <http://fr.wikipedia.org/wiki/Tivoisation>
- [7] <http://guideopensource.info/glossary/term/3>
- [8] <http://guideopensource.info/glossary/term/11>
- [9] <http://guideopensource.info/glossary/term/33>
- [10] <http://guideopensource.info/glossary/term/31>
- [11] <http://ar.tlibre.org/>
- [12] <http://guideopensource.info/glossary/term/48>
- [13] <http://guideopensource.info/glossary/term/43>
- [14] <http://guideopensource.info/glossary/term/6>
- [15] <http://guideopensource.info/glossary/term/17>
- [16] <http://guideopensource.info/glossary/term/20>
- [17] <http://guideopensource.info/glossary/term/19>
- [18] <http://guideopensource.info/glossary/term/4>
- [19] <http://guideopensource.info/glossary/term/58>
- [20] <http://guideopensource.info/glossary/term/27>
- [21] <http://guideopensource.info/glossary/term/63>

3.1. Choisir la licence Open Source adaptée au besoin

Publié sur Guide Open Source (<http://guideopensource.info>)

[22] <http://guideopensource.info/glossary/term/35>

[23] <http://guideopensource.info/glossary/term/16>

[24] <http://www.gnu.org/copyleft/gpl.html>

[25] <http://www.april.org/articles/divers/retrospective-2007.html#ToC23>

[26] <http://guideopensource.info/glossary/term/70>

[27] <http://guideopensource.info/glossary/term/39>

[28] <http://guideopensource.info/glossary/term/22>

[29] <http://guideopensource.info/glossary/term/10>