

Si on arrêterait d'utiliser les licences libres ? (au profit du domaine public)

L'un des auteurs [que l'on traduit le plus](#) sur le Framablog, Glyn Moody, choisit ici de mettre les pieds dans le libre plat.

Et si on n'utilisait plus les licences libres, qui ne sont pas sans poser problèmes, en plaçant directement le code dans le domaine public ?

Les avantages pourraient finalement dépasser les inconvénients !

Remarque : Sur le même thème on pourra également lire (ou acheter) notre framabook [Un monde sans copyright... et sans monopole](#). Sans oublier notre auteur Pouhiou qui place directement [ses romans](#) dans le domaine public et s'en explique ici dans [un fort intéressant dialogue](#) avec Lionel Maurel aka Calimaq.



Pourquoi il est temps d'arrêter d'utiliser les licences libres

[Why it's time to stop using open source licences by Glyn Moody](#)

Glyn Moody – 13 février 2013 – The H Open

(Traduction : Tr4sK, aKa, Sphinx, Isdf, Penguin, ProgVal, lamessen, Shanx, Amargein, ronane, MFolschette, Isser)

Les logiciels libres reposent sur un paradoxe. Afin que les utilisateurs puissent être libres, les licences libres utilisent quelque chose remettant en cause la liberté : le copyright. Ce dernier est un monopole intellectuel se basant sur la restriction de la liberté des gens à partager, la liberté est donc restreinte et non pas étendue. Quand Richard Stallman, en 1985, a créé la *GNU Emacs General Public License*, cela représentait un bidouillage brillant, maintenant il est peut-être temps de passer à autre chose.

Nous y sommes déjà et des éléments le montrent. Il y a 18 mois, les gens ont commencé [à remarquer](#) le déclin des licences copyleft vers des licences plus permissives comme la licence Apache ou BSD. Plus récemment, la croissance de GitHub a attiré l'attention, montrant également que de plus en plus de gens n'utilisent plus de licences sur GitHub (ce qui peut s'avérer [problématique](#) d'une certaine manière).

Je ne pense pas que le déclin des licences copyleft soit la preuve d'un échec, bien au contraire ! Je l'[écrivais](#) dans mon édito précédent, le logiciel libre a au fond gagné, prenant le pouvoir dans la plupart des secteurs clés de l'informatique. De la même manière, le passage à des licences permissives n'a été rendu possible que grâce au succès du copyleft : les idées participant à la création collaborative et à la contribution à un projet que l'on utilise sont maintenant généralisées. Il n'y a donc plus besoin de licences copyleft « fortes » pour faire respecter ces valeurs, elles font désormais partie de l'ADN des codeurs. Ian Skerrett le [déclarait](#) ainsi en 2011 :

« On n'a plus besoin de s'assurer que les développeurs ou les entreprises soient honnêtes. Ils contribuent aux projets open source parce que cela les aide dans leur travail. S'il fallait s'assurer que les développeurs sont honnêtes, pourquoi y aurait-il autant de projets Apache réussis ? Prenons l'exemple du projet Eclipse qui utilise un copyleft faible. Je ne connais que très peu de contributions ayant été intégrées à Eclipse parce que le développeur avait été forcé de contribuer à cause des obligations de licence. Les gens contribuent aux projets qu'ils utilisent parce qu'ils ont saisi le bénéfice qu'ils pouvaient recevoir grâce à leur contribution. »

C'est aussi pourquoi nous ne devons pas nous inquiéter du *cloud computing* – parfois présentée comme la tueuse des licences libres – puisqu'il n'y a pas de distribution obligeant à publier les éventuelles modifications du code. Mais une fois de plus, nous n'avons pas besoin de cette obligation : si une entreprise d'informatique de *cloud computing* veut pleinement tirer les bénéfices du logiciel libre, elle y contribuera de toute façon. Si elle ne le fait pas, elle n'a rien compris.

Quelle licence devons-nous donc adopter si on n'utilise pas une licence copyleft ? Apache ? BSD ? Pourquoi pas **aucune** licence du tout , c'est à dire mettre le logiciel dans le domaine public ? Après tout, c'est la conclusion logique du mouvement vers des licences de plus en plus permissives – une qui permet tout.

Compte tenu des discussions passionnées qui ont tendance à se produire lorsqu'on émet l'idée qu'il y a un mouvement des licences copyleft traditionnelles vers des licences légèrement plus permissives, je soupçonne que l'idée d'évoluer vers une licence complètement permissive sera choquante pour certains. À l'évidence, cela semblerait impossible, car conduisant « certainement » à l'effondrement du logiciel libre lui-même

si celle-ci était largement adopté.

Un [article](#) intéressant de [Clark Asay](#), maître de conférences à la *Penn State Univerity Dickinson School of Law* (et aussi frère de [Matt Asay](#), personnalité connue de tous dans le logiciel libre) étudie cette idée en profondeur et présente quelques arguments convaincants sur le fait que placer les logiciels libres dans le domaine public fonctionne et s'avère bénéfique.

Asay (Clark et non Matt) remarque qu'il y a un coût à utiliser des licences libres en termes de conformité. Les entreprises dépensent énormément d'argent en se préoccupant de faire les choses bien, mais les programmeurs, eux aussi, perdent du temps à s'occuper de détails légaux alors qu'ils pourraient être en train d'écrire davantage de lignes de code. En particulier, l'incompatibilité entre les nombreuses licences et leurs variantes est une barrière importante à de plus larges réutilisations et collaborations. Ces problèmes signifient que les logiciels libres ne sont pas utilisés aussi largement et efficacement qu'ils le pourraient, commercialement ou non. Ces difficultés peuvent expliquer en partie le glissement vers des licences plus « permissives », guidé par le désir d'éviter justement ces problèmes.

Asay se met alors à examiner les deux objections majeures à rendre le code disponible librement, sans aucune licence. La première tient au fait que les entreprises risquent de récupérer du code puis de l'enfermer, ce qui selon lui a peu de chance de se passer puisque, en faisant ainsi, elles écarteraient beaucoup des bénéfices que seuls les logiciels libres offrent :

« Si une société devait prendre la responsabilité d'un projet et le rendre fermé, elle n'obtiendrait certainement pas le travail bénévole que les contributeurs du monde entier ont envie d'offrir aux projets disposant de licences ouvertes. Sans ce travail bénévole, les sociétés perdraient un des

avantages les plus significatifs des modèles ouverts d'innovation, et ce travail bénévole resterait probablement fidèle à la version ouverte du projet. C'est pourquoi les sociétés encouragent d'ores et déjà à ouvrir le plus de projets possibles et à y contribuer. Car en faisant ainsi, cela va attirer une force de travail bénévole et déclenchera une innovation correspondant mieux à leurs besoins et stratégies.

Est-ce que la réciprocité (c'est-à-dire l'obligation de contribuer au projet en contre-partie) permet d'éviter la désertion des contributeurs individuels ? Il semble peu probable que, dans la plupart de cas, les contributeurs individuels aient le temps, l'intérêt et les ressources pour s'emparer d'un projet non-réciproque et d'en créer un équivalent fermé. La littérature suggère que les buts espérés par les individus qui contribuent à des projets aux licences ouvertes, ont peu à voir avec un avantage financier direct. Leurs intérêts pour la contribution résident plutôt dans la créativité, l'amélioration de sa réputation, et les bénéfices financiers indirects. Bien qu'il soit toujours possible pour des contributeurs de s'emparer d'un projet ouvert et de le rendre fermé pour l'intégrer dans leurs propres produits (et ainsi contrevenir à ce modèle ouvert d'innovation), les mêmes raisons qui suggèrent que les sociétés ne le feront pas s'appliquent également aux contributeurs individuels. Les contributeurs individuels sont encore moins à même d'abandonner les buts qu'ils espèrent atteindre dans la contribution à des projets ouverts, en plus de n'avoir que des ressources limitées pour réussir à fermer et maintenir un projet. »

L'autre objection majeure qu'il met en avant est que si le logiciel est placé dans le domaine public, il n'existe aucune exigence pour que tous les programmeurs reçoivent la reconnaissance qu'ils méritent pour leur contribution à un projet. Cette reconnaissance peut être importante en raison de

l'estime des pairs dont ils jouissent en retour et peut même aboutir à des compensations économiques sous la forme d'offres d'emploi et d'augmentations de salaire.

Attribution et réputation

Je suis d'accord pour dire qu'attribuer le mérite d'un projet est important, et cela de [manière cruciale](#). En fait, je pense que cela représente la clé du problème concernant les produits numériques partagés sur Internet, étant donné que les bénéfices économiques dépendent de cela. Toutefois, forcer cette attribution grâce à des licences ne représente peut-être pas le meilleur moyen. Asay l'explique ainsi :

« Dans une large mesure, l'approche de la gestion de la propriété intellectuelle choisie par les modèles d'innovation ouverts (c'est-à-dire les licences classiques des projets open source) échoue à remplir ses missions. Par exemple, le respect des licences attribution-only (mention de l'auteur original du projet) provoque l'enfouissement d'une telle reconnaissance dans les documents légaux ; cela fait que la reconnaissance d'une telle paternité devient minime. Cette réalité montre que ceux qui contribuent en utilisant des licences attribution-only, bien qu'étant motivés par une certaine forme de reconnaissance, obtiendront un tout autre type de reconnaissance que celui engendré par la propriété intellectuelle. Dans le monde du logiciel libre et open source, des outils comme GitHub, utilisé largement comme outil social de développement, peuvent fournir plus efficacement la reconnaissance que cherchent les programmeurs. Le fait que de plus en plus de contributions à des logiciels effectuées via GitHub se font sans avertissement de non-respect de la propriété intellectuelle ou des clauses de licences montre que le « prix » d'un tel avertissement à cause d'une documentation légale obscure n'en est pas un, du moins pour ceux qui contribuent. »

En effet, les personnes et les entreprises ont déjà commencé à utiliser les [profils GitHub](#) comme un moyen d'afficher et de mesurer la capacité à coder. Je soupçonne que cela deviendra bientôt la norme, avec des moyens plus formels d'établir sa réputation dans le monde du développement, apparaissant aux côtés de ceux informellement dictés par les normes sociales au sein de la communauté du logiciel libre.

Après avoir abordé les deux principales objections de se passer de licences *open source* traditionnelles, Asay examine ensuite ce que le domaine public signifierait en pratique :

« Une approche du style domaine public devrait remplacer efficacement les droits d'auteurs automatiques, supprimer tous les droits liés aux brevets (les deux respectant les droits des brevets déjà obtenus ainsi que de façon prospective), et renoncer à tous les recours possibles venant avec eux. Les droits liés au secret commercial, le cas échéant, devraient être supprimés dès que le titulaire des droits a publié le logiciel ou le contenu au public. On peut dire que renoncer à tout droit de marques est non seulement inutile mais déconseillé, car les autres pourraient utiliser les marques pour embrouiller les consommateurs quant à la source du logiciel ou du contenu. En effet, c'est exactement pourquoi la licence CC (Creative Commons), qui inclut une [licence dédiée au domaine public](#) dans son répertoire légal, exclut expressément les droits de marque dans son outil. »

Ce dernier point sur les marques est important, bien qu'il puisse sembler étrange de prétendre que tous les monopoles intellectuels et marques doivent être conservés, parce qu'ils servent un but très différent de celui du droit d'auteur ou des brevets. Les marques sont conçues pour protéger les consommateurs contre la fraude, plutôt que de chercher à exclure les concurrents, même si c'est la façon dont elles sont souvent utilisées aujourd'hui.

Mais pour les projets *open source*, les marques sont purement une question de réputation – c'est à dire qu'elles deviennent des garanties de qualité lorsqu'elles sont appliquées à un programme. N'importe qui peut prendre le code et l'utiliser et l'adapter de différentes façons. Mais il ne pourra pas utiliser la marque du projet, car cela impliquerait qu'il s'agit d'une version officielle « approuvée ». Ce qui serait évidemment problématique pour une variété de raisons, à commencer par celle de la sécurité.

Asay aborde également une objection importante dans sa thèse selon laquelle placer un logiciel dans le domaine public serait la meilleure façon de le distribuer : si c'était le cas, pourquoi tout le monde s'en tiendrait à la licence GPL ou Apache ? Comme il le souligne :

« Dans le monde du logiciel libre et open source, par exemple, il n'existe aucun outil reconnu ou largement utilisé dévoué au domaine public. A la place, l'Open Source Initiative et la Free Software Foundation – les deux principales organisations de défense des logiciels libres dans le monde – refusent ou approuvent les licences ouvertes utilisées dans la communauté. S'il est vrai que divers projets pourraient tout simplement ignorer ces licences recommandées et d'adopter une approche par domaine public – certaines ayant essayé de faire exactement cela – cette approche suppose que les organisateurs de ces projets comprennent comment le faire. »

Le fait est qu'il est actuellement assez difficile de placer un logiciel dans le domaine public, premièrement parce qu'il y a un biais culturel contre une telle attitude, même au sein de la communauté du logiciel libre, et deuxièmement parce que légalement c'est un processus délicat. En effet, Asay suggère que nous avons besoin d'une nouvelle législation – ce qu'il appelle *Public Domain Act (NdT, en français : « Loi du Domaine Public »)* – pour rendre ce processus plus facile. Il faudra

évidemment prendre en compte les différents systèmes de copyright ayant cours dans le monde – par exemple, ceux qui mettent plus l'accent sur les « droits moraux ».

Il y a quelques années, j'ai [demandé à Richard Stallman](#) ses points de vue sur la manière dont le copyright devrait être réformé, particulièrement concernant l'aspect logiciel. Voici ce qu'il a répondu :

« Pour la plupart des types de travaux, je pense que le droit d'auteur serait acceptable si nous l'avions (1) fait plus court (je suggère 10 ans), (2) permis la redistribution de copies dans un but non commercial et sans modification, (3) défini comme fair use les réutilisations sous forme de remix. Cependant, je pense que les logiciels et autres œuvres d'usage pratique doivent être libres. »

Notez qu'il pense, lui aussi, que le logiciel devrait être exempt de tout copyright – en d'autres mots, dans le domaine public – mais il a ajouté quelques mises en garde :

« Je serais heureux de voir l'abolition du copyright dans le logiciel si c'était fait de façon à s'assurer que le logiciel est libre. Après tout, l'objectif du copyleft est d'atteindre ce but pour les dérivés de certains programmes. Si tous les logiciels étaient libres, le copyleft ne serait plus nécessaire.

Cependant, l'abolition du copyright peut aussi être faite de manière erronée et pourrait n'avoir aucun effet sur les logiciels propriétaires typiques (qui sont restreints par des CLUF, Contrats de Licence Utilisateur Final, et le secret du code source plus que par le copyright), et ne ferait que porter atteinte à l'utilisation du copyleft. J'y serais naturellement opposé. »

Placer les logiciels libres dans le domaine public serait

équivalent à abolir le droit d'auteur pour ces programmes tout en laissant le code propriétaire intact. Est-ce vraiment un problème ? Personnellement, je ne le pense pas, pour les raisons que j'ai mentionnées : toute entreprise prenant du code dans le domaine public et se l'appropriant perd tous les avantages de son ouverture. Il est vrai qu'il reste des programmes hérités des maisons de logiciels à l'ancienne qui ont toujours été propriétaires, mais leurs existence n'affecte pas vraiment le monde plus large du logiciel libre, qui est maintenant arrivé à l'indépendance et à l'autosuffisance. Ce que Microsoft et ses semblables font en ce moment est quelque peu hors de propos.

Bien sûr, le passage au domaine public ne signifiera pas la disparition des licences libres actuelles – elles seront toujours là pour ceux qui souhaitent les utiliser. Comme toujours, le choix et la liberté personnelle sont capitaux. Mais j'espère que les gens y réfléchiront à deux fois avant d'introduire de nouvelles licences, ou même avant d'en mettre à jour d'anciennes. En particulier, j'espère qu'il n'y aura jamais de GNU GPL version 4. Au contraire, nous devons parachever la [révolution](#) que Richard Stallman a commencée il y a près de trois décennies en rendant le logiciel libre véritablement libre, en le plaçant dans le domaine public et en brisant les chaînes qui le lient encore à ce monopole vieux de trois cents ans nommé copyright.