

Les pages man de MySQL ne sont plus libres (bye bye GPL)

Edit : en fait [c'était un bug](#) (mais un bug troublant quand même). Fausse alerte donc, le contenu du billet ci-dessous n'est plus pertinent.

Quand on saute d'une version 5.5.30 à la 5.5.31, il ne se passe généralement pas grand chose mis à part quelques bugs fixés.

Sauf qu'ici on a décidé *en catimini* de changer la licence des [pages man](#) de [MySQL](#) et ce n'est pas très *fair-play* de la part de son propriétaire [Oracle](#).

Heureusement qu'on a le fork [MariaDB](#), d'où est d'ailleurs issue cette traduction.



Les pages man de MySQL changent discrètement de licence pour ne plus être

sous GPL

[MySQL man pages silently relicensed away from GPL](#)

Colin Charles – 18 juin 2013 – MariaDB Blog

(Traduction : Lgodard, ehsavoie, Asta, rou, BastienLQ)

On nous a récemment rapporté que la licence des pages man de MySQL a été modifiée. Le changement a été fait *en douce* lors du passage de MySQL 5.5.30 à MySQL 5.5.31. Cela affecte toutes les pages du répertoire man/ du code source.

On peut voir que ces changements sont intervenus durant cette courte période (5.5.30->5.5.31). Les anciennes pages du manuel étaient publiées sous la licence suivante :

Cette documentation est un logiciel libre : vous pouvez la redistribuer et/ou la modifier uniquement sous les termes de la licence GNU GPLv2 telle que publiée par la Free Software Foundation.

Les nouvelles pages man (5.5.31 et ultérieures – toujours en vigueur pour 5.5.32) sont sous la licence suivante :

Ce logiciel et la documentation rattachée sont soumis à un accord de licence contenant des restrictions sur leur utilisation et leur divulgation et sont protégés par les lois sur la propriété intellectuelle. Sauf autorisation expresse dans cet accord de licence ou par la loi, vous ne pouvez pas utiliser, copier, reproduire, traduire, diffuser, modifier, changer de licence, transmettre, distribuer, exposer, publier ou montrer une quelconque partie, quelque soit sa forme ou les moyens utilisés. Il est interdit de faire de la rétroingénierie, désassembler ou décompiler ce logiciel, sauf si la loi le requiert pour l'interopérabilité..

Ce n'est clairement pas une marque d'affection pour MySQL de la part d'Oracle. La nouvelle licence est également devenue

bien plus longue (en clair ce n'est pas sous licence GPL). Tandis que ce qui suit a été récupéré de l'outil resolveip, le copyright est le même pour toutes les pages man. Vous pouvez comparer [la page man de la version 5.5.30](#) et [celle de la version 5.5.31](#).



Quand le politique se met au service du privé pour que le public arrête le libre !

« L'affaire OpenJustitia » qui se déroule actuellement en Suisse est un cas très intéressant.

Comme on peut le lire sur le [site](#) du projet, **OpenJustitia** est « un ensemble de logiciels spécifiques pour les tribunaux. Le [Tribunal fédéral](#) a développé ces derniers de sa propre main et les a personnalisés à ses propres besoins. OpenJustitia permet notamment une recherche efficace dans les décisions du tribunal. »

Il a donc été développé en interne et, comme son nom le suggère, il est **libre** (sous licence GNU GPL v3) et a d'ailleurs reçu un prix dernièrement aux [CH Open Source Awards 2012](#).



Nous voici donc en présence d'un logiciel libre métier développé et mutualisé par l'administration. D'ailleurs le canton de Vaud a d'ores et déjà signé une convention de collaboration avec le Tribunal fédéral.

C'est exactement ce que prône en France une association comme l'[ADULLACT](#) avec la fameuse citation de son président François Elie : « l'argent public ne doit payer qu'une fois ».

Sauf qu'un parti politique (et derrière lui un éditeur de logiciels propriétaires) ne l'entendent pas de cette oreille, comme nous le rapporte l'[ICTjournal](#).

Pour ce qui concerne l'éditeur, c'est (plus que) maladroit mais (malheureusement) compréhensible :

L'entreprise bernoise Weblaw, editrice de logiciels de tribunaux propriétaires, estime que le Tribunal fédéral et sa solution font de l'ombre aux fournisseurs privés de logiciels. Le Tribunal fédéral doit-il s'occuper de droit ou de logiciels ?

Mais ce qui l'est moins c'est de voir l'[UDC](#) lui emboîter le pas et ne pas saisir l'intérêt, voire le bon sens, à utiliser du logiciel libre dans les institutions publiques :

Le [Conseil fédéral](#) doit examiner, à la demande de l'UDC, si le Tribunal fédéral a le droit de d'agir comme fournisseur du logiciel open-source Openjustitia. En agissant de la sorte,

ce dernier délivrerait des services non liés à ses compétences judiciaires.

Il est « totalement absurde » que le Tribunal fédéral fonctionne comme distributeur de logiciels, a déclaré Martin Baltisser, secrétaire général de l'UDC. Selon lui, d'une part le Tribunal fédéral n'aurait aucun intérêt prépondérant à agir en tant que fournisseur de logiciels, d'autre part il serait également dépourvu de base légale. Selon la Constitution et la Loi sur les finances de la Confédération, l'Etat ne peut intervenir au niveau commercial uniquement s'il n'existe pas d'offre privée et qu'une loi l'y autorise. Le Tribunal fédéral réplique qu'il ne réalise « aucun service commercial », comme le projet est open source, le logiciel est mis à disposition gratuitement.

Comme on peut le voir ci-dessous, on en a même parlé le 20 octobre dernier [à la RTS](#) mais, triste classique, en occultant complètement le *libre* pour n'évoquer que le *gratuit* :

D'autres voix se font heureusement entendre, comme celle de l'élu des Verts [François Marthaler](#) qui conteste, à juste titre et avec vigueur, cette demande de clarification de l'UDC sur son [blog](#) :

Je veux bien croire que la situation économique de Weblaw soit menacée. Mais je ne peux pas imaginer que les pouvoirs publics se trouvent empêchés de développer des solutions plus performantes et surtout moins onéreuses, dans l'intérêt de tous les contribuables et du bon fonctionnement de l'Etat. Plus encore que les coûts du développement initial du logiciel, ce qui est en jeu, c'est la maintenance et l'évolution du système au profit de l'administration, des justiciables et, finalement, des contribuables.

Sans le dire, Weblaw s'attaque au modèle économique des logiciels libres (open source). Un modèle dans lequel le

prestataire ne peut prétendre encaisser plus que la réelle valeur ajoutée au produit et pas une rente de situation. Que se serait-il passé si une société privée avait conçu le logiciel OpenJustitia et avait décidé de le mettre sous licence GNU/GPL ? Rien ! L'UDC n'aurait pas pu invoquer le « moins d'Etat » pour défendre les intérêts privés de cette petite société.

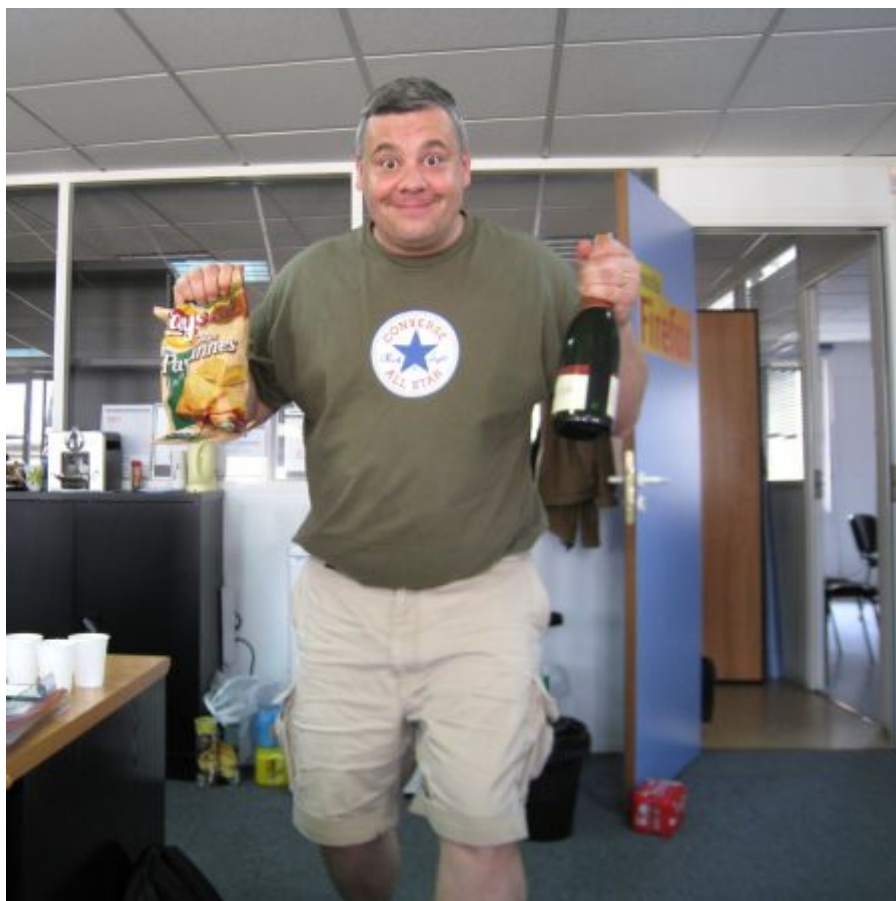
Espérons que comme le dit le dicton : les chiens aboient, la caravane passe...

La nouvelle version 2 de la Mozilla Public License tend vers l'unité

Le 3 janvier dernier la [Fondation Mozilla](#) annonçait la sortie officielle de la [version 2](#) de la [Mozilla Public License](#).

C'est un évènement à saluer [comme il se doit](#) car cette nouvelle version rapproche les deux grandes familles de licences libres logicielles que sont les [copyleft](#) (comme la [GPL](#)) et les licences dites [permissives](#) (comme la [BSD](#) ou la [MIT](#)).

Pour résumer, on pourrait dire que la différence entre une licence libre avec copyleft et une licence libre qui en serait dépourvue, les premières forcent le code dérivé à rester libre alors que les secondes autorisent le code dérivé à être intégré à un produit fermé, non libre^[1].



Mozilla peut-elle apporter l'unité à l'open source ?

[Can Mozilla Unify Open Source?](#)

*Simon Phipps – 6 janvier 2012 – ComputerWorld.uk
(Traduction Framalang : Don Rico e_Jim et Goofy)*

La nouvelle licence open source de Mozilla est bien plus qu'un simple ravalement de façade. Elle pourrait créer de nouvelles possibilités pour l'unité de la communauté du Libre.

La première semaine de janvier 2012 marque un jalon discret mais important dans le mouvement de l'open source, grâce à la [publication d'une deuxième version de la Mozilla Public License \(MPLv2\)](#) et sa [validation en tant que licence libre officielle](#) au sens de l'[Open Source Initiative \(OSI\)](#). Quand bien même beaucoup n'y voient qu'un énième détail juridique, cette publication est importante à deux titres : le procédé par lequel on l'a élaborée, et l'objectif pour lequel on l'a

créée. Il s'agit d'une licence qui a pour but l'unité.

Rédaction et révision de cette licence se sont déroulées selon un processus très ouvert, dans lequel [Luis Villa](#) a joué un rôle prépondérant. Organisé en majeure partie dans des forums publics, le débat a conduit à de nombreuses modifications du texte. Luis est entré en contact très tôt avec l'OSI, a intégré les retours du groupe de révision des licences, puis obtenu sans mal l'approbation du conseil d'administration.

D'autres articles sur cette nouvelle licence se sont concentrés sur les modifications de la partie « patent peace » (*NdT: la paix des brevets*) et autres ajustements des clauses (adieu, Netscape !), mais le changement le plus important apporté par la version 2 de la licence Mozilla est à mon sens l'inclusion d'une compatibilité particulière avec la [GPL \(GNU General Public License\)](#). Par le passé, le projet Mozilla jonglait avec un système complexe et peu clair de triple licence afin de composer avec les univers des licences copyleft et non copyleft. De manière générale, les autres utilisateurs de la MPL (et ses nombreux clones rebaptisés) ne prenaient pas cette peine, et par conséquent certains [codebases](#) se sont retrouvés exclus de toute collaboration possible avec l'immense univers des logiciels placés sous licence GPL.

Selon un procédé inédit que la Commission européenne a inauguré pour la [Licence Publique de l'Union européenne \(EUPL\)](#), la MPLv2 inclut des clauses permettant à un projet de stipuler, de façon optionnelle et explicite, sa compatibilité avec d'autres licences, en particulier celles de la famille GPL. À mes yeux, la MPLv2 représente une mise à jour d'envergure de la famille précédente des v1.x, justement grâce à cette compatibilité explicite avec la GPL, laquelle offre pour la première fois une passerelle praticable entre les paradigmes [permissifs](#) et [copyleft](#). Elle ne satisfera pas [les puristes des deux mondes](#), mais propose avec pragmatisme une nouvelle solution aux projets open source appuyés par des

entreprises. Celles-ci pourront disposer d'une communauté qui produit du code sous licence permissive tout en fournissant à cette même communauté un moyen d'entretenir des relations avec d'autres communautés travaillant sur du code sous licence copyleft.

Avec le déclin continu du business model de la double licence (ce que d'aucuns nomment « exceptions commerciales au copyleft »), il devient de plus en plus évident que les licences permissives sont importantes pour les entreprises commerciales qui contribuent à l'open source. De la même façon, l'écosystème GPL ne disparaîtra pas, aussi les conceptions qui reposent sur une opposition idéologique – y compris celles qui prônent l'élimination de tout code sous GPL – sont néfastes pour toutes les entreprises

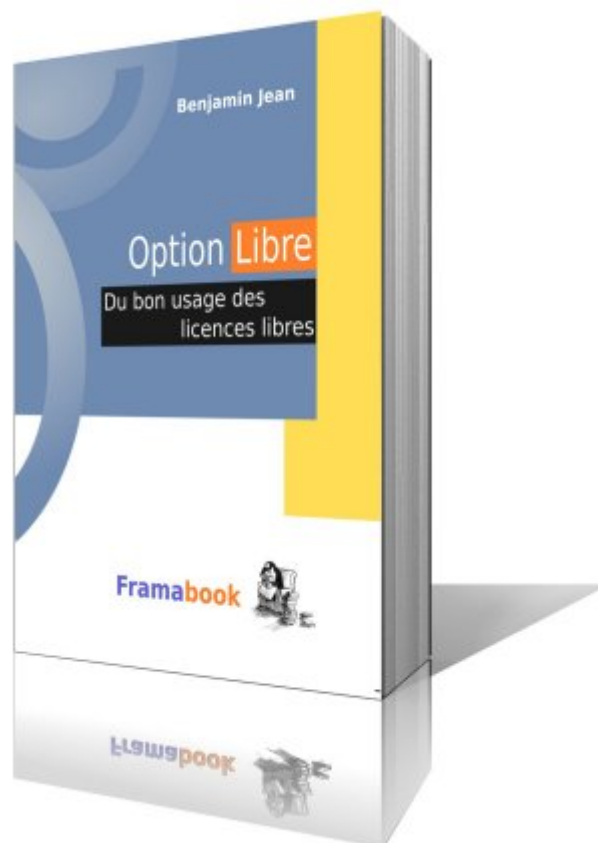
Je salue l'arrivée de la MPLv2, un pas en avant vers l'unification de la cause commune de nombreux développeurs open source. Bravo, Mozilla !

Notes

[1] Crédit photo : [Tristan Nitot](#) (Creative Commons By-Nc-Sa)

Option Libre, nouveau framabook pour tout savoir sur les licences libres

En cette nouvelle année 2012, Framasoft est fier et heureux de vous annoncer la sortie d'un nouveau **livre** framabook que nous attendions depuis longtemps : [Option Libre. du bon usage des licences libres](#), par notre ami et fidèle collaborateur **Benjamin Jean**.



Nous l'attendions depuis longtemps mais peut-être vous aussi. Ce livre que nous espérons précieux et utile vient en effet combler une lacune, celle de **donner une information claire, riche et précise sur les différentes licences libres**.

Afin de **faire son choix et/ou les utiliser en toute connaissance de cause**.

Nous aurons l'occasion d'en parler plus en détails dans de futurs billets, mais en attendant place à la **préface** de Michel Vivant, professeur à Sciences Po., que nous vous suggérons fortement de lire car, au delà de donner envie de parcourir l'ouvrage, elle est au demeurant fort bien écrite : « Le Libre n'est ni objet d'effarouchement ni objet de militantisme. **Le Libre est tout simplement.** ».

Et surtout [place au livre](#) qui, comme tout volume de la collection framabook, est un livre **sous licence libre**, disponible aussi bien à la vente, [20 € sur notre boutique EnVenteLibre](#) (merci pour votre soutien), qu'en [libre](#)

[téléchargement](#) sur le site du projet.

Préface

C'est à bien des titres que je connais M. Jean. Tout d'abord, il y a quelques années comme étudiant du master de « droit des créations immatérielles » que j'avais fondé puis ensuite pour l'avoir recruté comme chargé de mission à Sciences Po ; pour l'avoir eu encore à mes côtés pour organiser diverses manifestations dont un prestigieux cycle de conférences sur la propriété intellectuelle à la Cour de cassation ou comme membre d'un think tank sur le cloud computing...

À chaque fois, j'ai pu apprécier ses qualités humaines comme ses qualités intellectuelles. Benjamin Jean sait mêler la rigueur, la finesse d'analyse, le sens des nuances et de l'à-propos à une réelle modestie et une constante gentillesse, ce qui est rare, sans oublier, sur un autre registre, une grande passion pour les sujets qu'il aborde.

C'est tout cela qu'on retrouve dans les pages qu'il nous livre ici, et que j'ai pu, pour beaucoup d'entre elles, déjà découvrir en « bonnes feuilles ».

Passion bien sûr car, très tôt passionné d'informatique, Benjamin Jean s'est aussi vite découvert une passion pour cet « autre modèle » qu'est le « libre » qu'il nous décrit dans sa véracité et sa complexité. Car le libre n'est pas sous sa plume cet objet d'effarouchement qu'il est pour bien des juristes traditionalistes qui n'aiment pas sortir du cadre qu'ils connaissent. Il n'est pas non plus cet objet de militantisme (et pourtant Benjamin Jean est un militant) qu'il est pour bien des défenseurs du libre. Le libre est tout simplement.

Il est, il existe comme un phénomène installé dans le paysage des pratiques, né de la pratique et en réaction contre les pratiques établies, répondant à diverses aspirations dont la

volonté de partager la création faite (et le bénéfice qui en découle, point nécessairement le bénéfice économique) n'est pas la moindre. C'est le mérite de Benjamin Jean de présenter ainsi cette « Option libre » comme un phénomène qui mérite considération, appelle compréhension et décryptage, et ne se réduit pas à quelques modèles (quelques caricatures chez ses adversaires) simples.

La présentation des diverses écoles de pensée qui font qu'en réalité il n'y a pas un libre mais des libres ou, si l'on préfère, de multiples voies pour organiser cette liberté (car c'est bien d'une liberté organisée qu'il s'agit), est sans doute, sur le terrain spéculatif, une des parties les plus intéressantes de l'ouvrage. D'autant que – notre auteur le montre bien – aujourd'hui, au-delà des mouvements qui ont initialement fait naître le Libre, chacun a sa raison qui le pousse à opter pour celui-ci, jusqu'aux grands acteurs du privé qui jouent ou jouaient traditionnellement la carte propriétaire : « Les individus, les associations et autres organismes à but non lucratif peuvent avoir un rapport au Libre complexe qui mêle philosophie, idéologie, et des arguments plus raisonnés comme le coût ou l'indépendance qu'offrent les solutions logicielles libres » écrit ainsi Benjamin Jean, avant de souligner comment entreprises et administrations ont elles aussi adopté le Libre comme élément de leur stratégie.

Voilà peut-être qui pourra sembler bien spéculatif. C'est pourtant certainement la condition préalable sans laquelle il n'est guère possible de se retrouver véritablement au sein du monde du Libre. Mais, pour qui veut des choses plus concrètes, on ne peut ignorer cet autre moment fort de l'ouvrage, qui prolonge les réflexions, quand Benjamin Jean nous propose « repères » et « réflexes » à avoir. La « proposition d'une grille de lecture » des licences, avec la question délicate de leur compatibilité, les pages sur le choix des licences ou sur la manière de concevoir un projet pensé sous le signe du Libre

sont des pages précieuses, certainement à lire et à relire.

Ainsi, celui qui cherche des « recettes » comme celui qui veut prendre du recul trouvera son bonheur dans cet ouvrage.

Libre à chacun de s'y plonger ou non. Option libre évidemment...

Mais ce serait un bien mauvais usage de la liberté que de ne pas profiter de cette heureuse Option Libre.

Michel Vivant

Professeur à l'École de droit de Sciences Po., Paris

-> [Option Libre. du bon usage des licences libres](#)

Le logiciel libre pourrait-il exister sans le copyright ? La réponse de Stallman

Dans l'un de nos récents framabooks, le professeur néerlandais Joost Smiers se pose la question suivante : [Et si nous supprimions carrément le copyright ?](#)



Pour en conclure que les cartes seraient évidemment redistribuées mais que le monde ne s'arrêterait pas de tourner. Et force est de reconnaître qu'il n'est pas le seul à envisager cette radicale solution.

Or, apparent paradoxe, il se trouve que, juridiquement parlant, les licences des logiciels libres sont adossées au

copyright. Elles le respectent pour mieux, en quelque sorte, le retourner en leur faveur, à fortiori lorsque ces licences sont également [copyleft](#), comme la plus célèbre d'entre elles, la licence [GNU GPL](#).

C'est au père de cette dernière, [Richard M. Stallman](#)^[1], que [Glyn Moody](#) s'est adressé pour lui demander ce qu'il pense du copyright et de l'avenir du logiciel libre si le copyright n'existait plus.

Le logiciel libre pourrait-il exister sans le copyright ?

[Could Free Software Exist Without Copyright?](#)

Glyn Moody – 9 juillet 2010 – ComputerWorldUK

(Traduction Framalang : Vincent, Barbidule, Toufalk, Pablo, Goofy, et Petrus6)

Il y a quelques jours, j'écrivais sur la manière dont la licence GNU GPL de Richard Stallman utilise le copyright afin de garantir que les utilisateurs de la licence partagent le code qu'ils distribuent. S'ils ne le font pas, ils sont en violation de la GPL, et perdent donc leur protection contre les actions en violation de copyright.

Cela est bel et bon, mais comme beaucoup l'ont fait remarquer, cela a pour conséquence paradoxale que la licence GNU GPL dépend du copyright, un monopole intellectuel, pour promouvoir la liberté intellectuelle. De plus, cela semble condamner le logiciel libre à un sorte de symbiose avec le copyright, en le contraignant à défendre ce monopole sans lequel la GPL ne serait pas aussi puissante.

Voilà une perspective bien sûr légèrement dérangeante, et je m'étais donc dit il y a peu que je soulèverais la question avec RMS lui-même, puisqu'il avait forcément conscience du problème et qu'il avait peut-être une solution (ce que

j'espérais). Ces derniers mois la question s'est posée à plusieurs reprises, j'ai donc pensé que ça valait le coup de publier ses réponses à mes interrogations, pour donner un éclairage sur ce débat crucial.

D'abord, je lui ai demandé comment nous devrions réformer le copyright, puisque c'est un monopole intellectuel dont abusent les éditeurs, mais que la GNU GPL en dépend.

Voici la réponse de Stallman :

« Pour la plupart des œuvres, je pense que le copyright pourrait être acceptable s'il était plus court (je propose 10 ans), s'il permettait une redistribution de copies verbatim non commerciale, et si les « remix » modifiant l'œuvre étaient clairement considérés comme un usage légitime (« [fair use](#) »).

Cependant, je pense que les logiciels et toutes les œuvres ayant une utilité concrète doivent être libres. »

Il a poursuivi :

« Je serais heureux que le copyright sur les logiciels soit aboli si c'était fait d'une manière telle que la liberté des logiciels soit garantie. Après tout, le but du copyleft est justement d'atteindre cet objectif pour les dérivés de certains programmes. Si tous les logiciels étaient libres, nous n'aurions pas besoin du copyleft.

Cependant, menée de la mauvaise manière, l'abolition du copyright pourrait n'avoir aucun effet sur les logiciels privés (car plus que le copyright, c'est le [CLUF](#), Contrat de licence utilisateur final, et le caractère secret du code qui créent des restrictions), elle viendrait simplement remettre en cause la pratique du copyleft. Naturellement, dans ce cas je m'y opposerais.

En d'autres termes, je me sens plus concerné par l'effet de la

loi sur les libertés des utilisateurs que par ce qui pourrait arriver au copyright en tant que tel. »

Je lui ai alors demandé comment l'abolition du copyright pourrait être menée pour que le logiciel libre soit encore possible.

« Il faudrait éliminer le copyright sur les logiciels, déclarer les CLUF juridiquement nuls et adopter des mesures de protection du consommateur qui imposent la distribution du code source aux utilisateurs et l'interdiction de la [tivoisation](#). »

Stallman a expliqué ce qu'il entendait par « tivoisation » il y a quelques années, quand la GNU GPL v3 était en cours d'élaboration. C'est le fait d'élaborer une machine telle que, si l'utilisateur installe une version modifiée d'un programme, la machine refuse de l'exécuter.

Ce nom vient de [Tivo](#), le premier produit dont j'ai su qu'il faisait ça. Le Tivo contient des logiciels libres sous GPL v2, dont le code source est fourni. L'utilisateur du Tivo peut donc modifier le programme, le compiler et installer la version modifiée sur sa machine. Celle-ci refusera cependant de fonctionner car elle aura décelé qu'il s'agit d'une version modifiée. Cela signifie qu'en théorie, l'utilisateur a la liberté numéro 1 (NdT : la liberté d'étudier le fonctionnement du programme), mais en réalité, il ne l'a plus, ce n'est qu'un leurre. Cette pratique est systématique, et elle constitue une menace générale pour la liberté de l'utilisateur.

Nous avons donc décidé d'empêcher cela, en modifiant les conditions de distribution des binaires. Nous avons précisé que si vous distribuez les binaires dans un produit, ou pour être utilisés dans un produit, alors vous devez fournir tout ce dont l'utilisateur a besoin pour installer sa propre

version modifiée et faire que le produit fonctionne de la même façon, sous réserve que les changements effectués dans le code ne modifient pas la fonction. L'important est que non seulement l'utilisateur puisse être capable d'installer et faire fonctionner une version modifiée, mais encore que celle-ci doit être capable de faire la même chose que l'original.

Il faut noter que Stallman ne croit pas nécessaire d'abolir complètement le copyright, il propose juste de le restreindre un peu ; la durée exacte pouvant faire l'objet de débat :

« Ma proposition de faire durer le copyright 10 ans à partir de la date de publication se veut conservatrice. Je pense que 5 ans suffisent et je n'ai rien contre une période plus courte encore mais je ne me battraï pas pour ça. »

D'après Stallman, l'avantage de cette « modeste proposition » est qu'elle ne requiert pas de grandes modifications législatives. Alors que ce serait bien le cas pour les mesures de protection de l'utilisateur qui seraient nécessaires afin de préserver la liberté du logiciel si le copyright venait à disparaître.

Il est plus facile de faire pression pour ramener le copyright à ses conditions d'origine (14 ans pour les nouvelles oeuvres avec, en option, 14 années supplémentaires) plutôt que pour l'abolir complètement. C'est ironiquement une approche très pragmatique, alors même que Stallman est souvent accusé du contraire.

Notes

[1] Crédit photo : [Sebastian Oliva](#) (Creative Commons By-Sa)

Geektionerd : Quoi d'neuf chez Microsoft

Et voici Microsoft qui refait parler de lui.

Après [François Fillon chez Microsoft](#), c'est Steve Ballmer à l'Elysée pour y recevoir la Légion d'honneur. « J'admire cette entreprise qui comprend les valeurs humanistes de la France et de l'Europe » a alors [déclaré](#) Nicolas Sarkozy pour l'occasion, n'ayant visiblement pas peur de perdre quelques électeurs libristes en route.

Dans ces valeurs humanistes il n'y a toujours pas le logiciel libre, dont la licence GPL est [explicitement exclu](#) du prochain Marketplace dédié au Windows Phone 7.

Un sujet de plus à évoquer lorsque [Framasoft se rendra lui aussi chez Microsoft](#) au début du mois prochain.

QUOI D'NEUF CHEZ MICROSOFT

La célèbre entreprise lance son « application store » pour Windows Phone, dont la licence interdit très explicitement les logiciels libres. . . .

Pas de logiciel libre pour le Windows Phone...



En même temps, pas de client non plus...

Un partout, la ball(mer) au centre.

Et comme chez nous, en France, on sait reconnaître le talent, c'est bien sûr le moment de remettre la légion d'honneur au PDG, Steve Ballmer.

Le roi de France Nicolas I a félicité Bill II pour son entreprise qui, je cite, « comprend les valeurs humanistes de la France et de l'Europe »...

...ce dernier a encouragé son Altesse dans sa volonté de créer un « Internet civilisé » et de lutter contre la cybercriminalité*.



Aaaah, quand deux idéologies axées sur la loi, du pognon et la soit de pouvoir, se rencontrent, ça fait toujours ton sur ton...

* En référence probable aux dangereux criminels qui avaient piraté le Facebook de sa Seigneurie.

Enfin, sachez que le grand manitou du réseau Framasoft*, j'ai nommé aka, a été invité chez Microsoft France pour débattre avec eux. . . Je propose un entraînement intensif :



Pourquoi rejeter les logiciels propriétaires ?

Parce qu'ils privent les utilisateurs de toute liberté !

Allez, ton crochet du droit !

Et on dit « privé », Richard...

Et si on t'offre des petits fours et une X-Box ?

Allez Alexis !
Donne tout ce que t'as !

Incorruptible !

Alexis gee

Crédit : [Simon Gee Giraudot](#) (Creative Commons By-Sa)

À qui la faute si les logiciels libres sous licence GPL sont éjectés de l'App Store ?

On peut reprocher beaucoup de choses à Microsoft mais jamais on n'a vu un logiciel libre empêché de tourner sous Windows parce que la licence du premier était incompatible avec le contrat d'utilisation du second (sinon un projet comme Framasoft n'aurait d'ailleurs pas pu voir le jour).



Il semblerait qu'il en soit autrement avec Apple et son [App Store](#), la plateforme de téléchargement d'applications pour les appareils mobiles fonctionnant sous [iOS](#) (iPod, iPhone et iPad).

C'est que révèle la récente « affaire VLC », [célèbre logiciel libre](#) de lecture vidéo, dont la licence semble tant et si bien poser problème à Apple qu'il a été brutalement et unilatéralement décidé de le retirer de l'App Store.

Et tout le monde se retrouve perdant, à commencer par l'utilisateur qui ne pourra plus jouir de cette excellente application sur son iPad & co^[1].

Derrière ce malheureux épisode ([qui n'est pas le premier du genre](#)) se cache une question en apparence relativement simple : logiciels libres et App Store peuvent-ils cohabiter ? Un logiciel, dont les libertés d'usage, de copie, de modification et de distribution, sont garanties par sa licence, peut-il se retrouver dans un espace dont le contrat

stipule un nombre limité de copies sur un nombre limité de machines ? Et comme la réponse est en l'occurrence négative, le risque est réel de voir les logiciels libres totalement écartés de l'App Store, et par extension des nombreuses autres plateformes privées qui poussent comme des champignons actuellement.

Mais attention, il s'avère que dans le détail c'est plus complexe que cela. Et c'est pourquoi nous avons pris la peine d'ajouter notre grain de sel au débat en traduisant l'article ci-dessous qui résume assez bien à nos yeux la situation. C'est complexe mais ça n'en est pas moins intéressant voire enrichissant car les particularités de la situation éclairent et illustrent de nombreux aspects du logiciel libre.

Il convient tout d'abord de préciser que ce n'est pas le logiciel libre en général mais le logiciel libre sous [licence GPL](#) (et son fameux [copyleft](#)) qui est pointé du doigt ici. Ensuite il y a l'existence d'un troisième larron en la personne de la société Applidium à qui l'on doit le portage de VLC dans iOS et sa présence dans l'App Store (dans un premier temps accepté puis aujourd'hui refusé). App Store dont les règles d'utilisation définies par Apple sont floues et changeantes. Enfin c'est bien moins la « communauté VLC » dans son ensemble que l'un de ses développeurs qui est impliqué dans cette histoire.

Choisir de placer son logiciel sous licence libre, a fortiori sous licence GPL, n'est pas un acte anodin. La liberté « en assurance-vie » que vous offrez là à vos utilisateurs peut s'opposer parfois frontalement à d'autres logiques et objectifs.

Tout le monde se retrouve perdant, mais le logiciel libre le serait davantage encore s'il devait céder en y perdant son âme.

La GPL, l'App Store et vous

[The GPL, the App Store, and you](#)

Richard Gaywood – 9 janvier 2011 – Tuaw.com

(Traduction Framalang : Olivier Rosseler et PaulK)

Mon collègue Chris a rédigé un article sur [l'éviction de l'App Store](#) du célèbre logiciel de lecture vidéo VLC, après la plainte d'un développeur du projet sur la base d'une violation de la GNU Public License (l'application a depuis été [mise à disposition sur Cydia](#) pour les appareils jailbreakés).

Les réactions de la blogosphère Mac ont été virulentes après cette décision, et Chris s'est admirablement fait l'avocat des plaignants. Mais il ne faut pas oublier qu'une histoire a toujours plusieurs versions (et que la vérité se trouve souvent à mi-chemin entre les deux). Aujourd'hui, je me fais avocat de la défense : pourquoi Rémi Denis-Courmont était parfaitement dans son bon droit. Mais pour mieux étayer mon argumentation, je vais revenir un peu sur l'historique de l'Open Source. Alors accrochez-vous !

Commençons par le commencement : pourquoi VLC a été retiré de l'App Store

L'histoire a [déjà été traitée](#), mais voici quelques faits que vous devez connaître. VLC (hébergé à l'adresse <http://videolan.org>) est un lecteur audio et vidéo complet, multi-plateforme qui a maintes fois fait ses preuves. VLC est publié sous la [version 2](#) de la licence GNU Public License (GPL). Une entreprise nommée Applidium, sans lien avec le projet, a utilisé le code source de VLC pour [en faire une application pour iOS gratuite](#), afin que les utilisateurs d'iPhone et d'iPad puissent s'en servir pour lire plus de formats de fichiers que leur appareil n'en supporte nativement, comme les fichiers AVI et MKV. En accord avec la GPL, Applidium a libéré le code source de la version modifiée

de VLC.

L'accueil reçu par ce projet sur les listes de diffusion des développeurs de VLC fut [contrasté](#). Alors que certains développeurs n'y voyaient pas d'inconvénient, le portage de leur travail (censé être ouvert) sur une plateforme, iOS, connue pour sa fermeture et ses restrictions sur la ré-utilisation du code des applications distribuées sur l'App Store, gênait les autres. En point d'orgue de cette histoire, l'un des développeurs, Rémi Denis-Courmont (auteur d'une bonne partie du code de VLC) demanda à Apple de retirer l'application VLC pour iOS de l'App Store pour violation de la GPL, à laquelle est soumise sa contribution au projet.

Finalement, quelques mois plus tard, il semblerait qu'Apple ait obtempéré sans poser de question puisque l'application [a été retirée](#). Denis-Courmont [s'interroge](#) malgré tout sur les délais. Si c'était effectivement la réponse d'Apple à sa demande, l'entreprise aurait pu agir bien plus tôt.

Et donc, qui est dans le vrai ?

Une courte histoire des licences Open Source

Tout d'abord, à quoi sert l'Open Source ? Parfois, des programmeurs initient des projets sur leur temps libre (ou en tant qu'universitaire, ne subissent pas les pressions commerciales, tous les systèmes d'exploitation basés sur Unix trouvent leurs racines dans l'Open Source et dans les universités, ce qui n'est pas une coïncidence), puis ils deviennent trop gros pour s'en sortir seuls. Une aide extérieure est la bienvenue, par exemple pour réparer les bogues ou pour ajouter de nouvelles fonctionnalités, et grâce à Internet les volontaires sont nombreux. Ils mettent donc le code sous une licence Open Source afin d'en publier la source et de recruter des collaborateurs.

Dit ainsi, ça paraît assez utopique, mais le fait est que de nombreux logiciels de qualité sont nés ainsi, grâce à l'Open

Source : le compilateur C GCC, le noyau [Mach](#), le serveur web Apache HTTPd, l'interface de commande bash, les langages de programmation Perl et Python et le moteur de rendu Webkit utilisé par Safari. Ces quelques exemples ainsi que [des centaines d'autres](#) sont tous Open Source et font tous partie de Mac OS X. Apple [utilise](#) donc clairement l'Open Source.

Au-delà des logiciels, nous devons Wikipedia à la famille des licences [Creative Commons](#), créées pour étendre les idées des logiciels Open Source à d'autres activités créatives. On peut même dire que l'Open Source reproduit approximativement les processus scientifiques où les chercheurs publient leurs méthodes et leurs découvertes librement afin que d'autres chercheurs puissent s'appuyer sur leurs travaux. Ce sont là des concepts très importants à mes yeux puisque je suis titulaire d'un doctorat en informatique. Je pense que l'on peut s'accorder sur le fait que les idées derrière les logiciels libres nous sont bénéfiques à tous.

Prenons l'exemple d'Alice, une programmeuse qui a écrit un petit utilitaire. Un autre programmeur, Pierre, apprécie beaucoup son programme, mais il voudrait le modifier pour qu'il réponde mieux à ses besoins. Alice décide alors de dévoiler le code source de son programme pour que Pierre puisse y apporter des changements.

La licence la plus simple qu'Alice puisse appliquer à son code est celle du domaine public. Cela signifie que l'auteur a volontairement abandonné tous ses droits sur le code et que chacun peut en faire ce qu'il veut. Pierre peut faire ses modifications et garder la nouvelle version pour lui-même, ou ouvrir sa boutique et la vendre. Il peut même en parler à son patron, Paul, chez Microsoft, et peut vendre l'outil d'Alice dans la prochaine version de Windows, sans qu'elle ne touche jamais un centime. Si vous vous dites que c'est pas très juste pour Alice, vous n'êtes pas les seuls. Gardez cet exemple à l'esprit.

Alice a également le choix entre des familles de licences similaires : la licence [MIT](#), la licence [BSD](#) et la licence [Apache](#). Grâce à elles, Alice conserve son droit d'auteur sur le code, mais elles octroient également à quiconque le télécharge le droit légal de le modifier et d'en distribuer la version modifiée. Différentes variantes de ces licences imposent également à Pierre de mentionner le nom d'Alice quelque part dans le logiciel, à l'instar de la licence Creative Commons Attribution. Mais rien ne l'empêche de le vendre sans rien verser à Alice en échange.

La GPL et les autres licences copyleft

En réponse à ce problème, la Fondation pour le Logiciel Libre (FSF) a créé la [GNU Public License](#). Elle est aussi appelée [copyleft](#) pour bien marquer la rupture qu'elle représente par rapport aux autres licences dont il est question ici. Philosophiquement, la GPL a été créée pour assurer aux utilisateurs de logiciels sous GPL les quatre libertés suivantes :

- La liberté d'exécuter le logiciel, pour n'importe quel usage ;
- La liberté d'étudier le fonctionnement d'un programme et de l'adapter à ses besoins, ce qui passe par l'accès aux codes sources ;
- La liberté de redistribuer des copies ;
- La liberté de faire bénéficier à la communauté des versions modifiées.

Évidemment, la licence en elle-même est un charabia juridique assez dense, dont le but est d'assurer une validité légale à cette philosophie, mais ces quatre libertés en sont vraiment le cœur. Vous remarquerez que la deuxième et la quatrième libertés imposent que le code source de tout programme sous GPL soit mis à disposition en plus de l'exécutable, que l'on utilise vraiment. C'est la raison pour laquelle de nombreuses entreprises qui font usage de code sous GPL dans leurs

produits hébergent des pages entières de code source, comme le fait [Netgear](#) par exemple. Alors, évidemment, leurs concurrents peuvent analyser le fonctionnement exact de leurs produits, pourtant ils estiment que cet inconvénient est largement compensé par les avantages qu'il y a à s'appuyer sur des produits Open Source.

Si on reprend notre histoire d'Alice et Pierre, voyons ce qu'il se passe maintenant. Alice rend son code source public. Pierre rédige son patch et décide sournoisement de vendre cette nouvelle version du programme d'Alice. Mais Alice n'est pas bête et voit bien ce qu'il se trame. Elle pose alors un ultimatum à Pierre : soit il rend public le code source de sa version modifiée , soit ils se retrouveront devant un juge pour rupture du contrat grâce auquel il a obtenu la version d'Alice (c'est à dire, la GPL). Bob s'incline évidemment et publie son code. Aucun cas n'est allé jusqu'au tribunal dans la vraie vie. Pour beaucoup, cela prouve que la GPL est valide et incontournable : personne n'ose la contester.

L'App Store et la GPL peuvent-ils co-exister ?

Le conflit le plus évident entre l'App Store et la GPL réside dans la troisième liberté ? « La liberté de redistribuer des copies ».

Ce sujet a été largement débattu, chaque camp avançant ses arguments. Malheureusement, la résultat est que : premièrement, c'est un terrain plutôt flou et deuxièmement, il ne sera jamais certain qu'une des deux partise a plus raison que l'autre sans décision de justice. Il est peu probable que l'on en arrive à une telle situation car Apple (en tant que distributeur de logiciel pour tout l'App Store) n'ira probablement pas se battre en justice pour un logiciel Open Source. Concrètement, tant que le contraire n'est pas prouvé, l'App Store et la GPL sont incompatibles. C'est sans doute l'avis d'Apple en tout cas, ils n'auraient sûrement pas supprimé VLC (et d'autres logiciels Open Source comme [GNU Go](#))

de l'App Store dans le cas contraire.

Les membres de la Fondation pour le Logiciel Libre (FSF), auteurs de la GPL [ne pensent pas non plus](#) que l'on puisse concilier la GPL et l'App Store. Leur principal argument réside dans la clause suivante de la GPL v2 :

À chaque fois que vous redistribuez un programme (ou n'importe quelle réalisation basée sur le programme), l'utilisateur final reçoit automatiquement une licence du détenteur originel du programme l'autorisant à copier, distribuer et modifier le programme sujet à ces termes et conditions. Vous ne devez imposer à l'utilisateur final aucune autre restriction à l'exercice des droits garantis par la licence.

Regardons maintenant les [Règles d'utilisation](#) des produits App Store :

(i) Vous êtes autorisé à télécharger et synchroniser un Produit à des fins personnelles et non commerciales sur tout produit de la marque Apple tournant sur iOS (« Produit iOS ») que vous possédez ou contrôlez.

(ii) Si vous êtes une société commerciale ou un établissement scolaire, vous êtes autorisé à télécharger et synchroniser un Produit destiné à être utilisé soit (a) par une seule personne sur un ou plusieurs Produits iOS que vous possédez ou que vous contrôlez, soit (b) par plusieurs personnes sur un Produits iOS partagé dont vous êtes propriétaire ou que vous contrôlez. Par exemple, un seul salarié peut utiliser le Produit aussi bien sur son iPhone que sur son iPad, ou encore, plusieurs étudiants peuvent utiliser le Produit en série sur un seul iPad situé dans un centre de ressources ou une bibliothèque.

(iii) Vous pourrez simultanément stocker des Produits App Store à partir d'un nombre maximum de cinq Comptes différents

sur des Produits iOS compatibles, tels qu'un iPad, un iPod ou un iPhone.

(iv) Vous pourrez procéder à la synchronisation manuelle de Produits App Store à partir d'un ou plusieurs appareils autorisés par iTunes vers des Produits iOS munis d'un mode de synchronisation manuel, à condition que le Produit App Store soit associé à un Compte existant sur l'appareil principal autorisé par iTunes, étant précisé que l'appareil principal est celui qui a été synchronisé en premier avec le Produit iOS ou celui que vous avez ultérieurement désigné comme tel en utilisant l'application iTunes.

Le conseil Légal de la FSF considère ces termes comme des restrictions à l'utilisation que peut faire l'utilisateur des logiciels obtenus sur l'App Store, c'est donc directement une atteinte à la GPL. Il importe peu que le port de VLC sur iOS soit libre. En fait, il aurait été plus simple d'imaginer qu'il ne l'était pas.

Imaginez, si Applidium avait fait payer 5\$ pour leur port de VLC, tout en distribuant le code source complet sur leur site web. Rien dans la GPL n'empêche les développeurs de facturer leurs logiciels dérivés, c'est donc convenable. Mike achète une copie du logiciel pour 5\$ et veut donner une copie à son ami Steve (c'est son droit, garanti par la GPL), mais il ne peut pas. Steve peut acheter sa propre copie à 5\$, mais ne peut pas l'avoir par Mike ; il doit aller voir Apple.

Dans la réalité, VLC était gratuit, ce n'est donc pas gênant pour Steve de ne pas pouvoir dupliquer la copie de Mike ; mais ce ne sera pas forcément le cas avec d'autres logiciels et dans tous les cas, ça ne change rien, étant donné que les termes de la GPL n'acceptent pas plus que l'on restreigne la liberté de partager le logiciel parce que le logiciel dérivé se trouve être gratuit.

Notez que la FSF [considère](#) également le fait de ne pouvoir

installer la même application que sur cinq iOS en même temps comme une autre restriction aux droits de l'utilisateur ; cela semble être une erreur, puisque ce terme du contrat n'apparaît que sur la partie dédiée au contenu iTunes du Contrat de l'utilisateur et ne s'applique donc pas aux applications de l'iOS. Il est également possible que le Contrat de l'utilisateur pour iTunes ait été modifié après la publication de l'article de la FSF.

Qu'en est-il des autres programmes sous GPL présents dans l'App Store ?

On croit souvent, à tort, que lorsqu'un code est publié sous GPL, il ne peut plus l'être sous une autre licence. Ce n'est pas vrai. Si tous les ayants-droits, c'est-à-dire toutes les personnes ayant contribué au code, se mettent d'accord, le logiciel peut être placé sous une double licence.

Prenez l'exemple de Java. Java est placé sous GPL et Apple a pourtant utilisé la source de Sun pour la modifier (afin que la machine virtuelle Java sous OS X soit mieux intégrée à cet environnement) sans pour autant distribuer le code source de la version modifiée. Comment ? Simplement parce qu'ils ont utilisé le code source de Sun sous une autre licence, ce que Sun pouvait proposer car (à l'époque) le code de Java leur appartenait. Libre à eux donc de l'offrir au monde entier sous GPL tout en le proposant à Apple sous une autre licence.

Mais pour les grands projets, comme VLC, qui acceptent des correctifs publics de la communauté depuis longtemps, il n'est pas envisageable d'obtenir la permission rétroactive de tous les contributeurs pour appliquer une licence double. D'autres projets, en revanche, sont proposés sous GPL mais n'ont jamais accepté de correctifs extérieurs, ce qui signifie que leur code est toujours sous le contrôle d'une seule et même personne ou entreprise. On peut citer l'exemple de Doom, qui est sur l'App Store. [BeTrains](#) en est un autre exemple et les développeurs [ont récemment décrit](#) comment ils arrivent à

concilier GPL et App Store.

Doit-on modifier la GPL pour la rendre compatible avec l'App Store ?

Cette question ne relève plus du problème légal, mais plutôt de la politique des logiciels. Les défenseurs de la GPL adoptent souvent une attitude méfiante, ce qui est, à mes yeux, souvent perçu comme du fanatisme en dehors de la communauté Open Source. Mais pourtant, les violations de la GPL sont malheureusement courantes. Par exemple, une étude récente de Matthew Garret montre que parmi [des centaines de tablettes Android](#), la quasi-totalité violent la GPL. En d'autres termes, les systèmes de tous ces fabricants s'appuient à 99% sur du code libre mais contiennent 1% de leurs propres modifications. Ensuite ils les vendent comme s'ils étaient à l'origine de la totalité du système.

Il va sans dire que les développeurs qui ont participé à ces projets sur leur temps libre sont indignés par ces pratiques, c'est pourquoi ils condamnent aussi facilement tout ce qui ressemble de près ou de loin à une violation de la GPL. C'est souvent pris pour du fanatisme ou même de l'hostilité, mais au fond, c'est juste l'expression du dégoût de groupes de bénévoles face aux pratiques d'entreprises qui tentent de se faire de l'argent facile avec leur travail.

Sans oublier cette peur que tout signe d'indulgence fasse basculer les choses du mauvais côté, que cela encourage plus encore les entreprises à violer la GPL, et que la FSF, avec ses maigres ressources, n'ait plus les moyens d'identifier et de menacer les entreprises qui se comportent mal pour qu'elles respectent la communauté sur laquelle elles s'appuient.

Les règles de l'App Store doivent-elles changer pour être compatibles avec la GPL ?

C'est ce qu'[aimeraient](#) certains développeurs de VLC, comme Ross Finlayson. Plutôt que de voir VLC banni de l'App Store,

ils préféreraient que ses règles soient modifiées afin de créer une exception pour les applications sous GPL. Rien n'est impossible, mais Apple (pourtant bien enclin à piocher largement dans l'Open Source pour créer Mac OS X) n'est pas vraiment réputé pour son ouverture aux compromis lorsqu'on touche à l'App Store. Il y a très peu de chance qu'un tel changement arrive, surtout pour des applications gratuites qui ne rapportent rien à Apple.

Est-ce que VLC pourrait être publié sous une autre licence ?

Actuellement, il serait presque impossible que VLC puisse paraître sous une autre licence, même réservée uniquement à Applidium, car chaque développeur ayant travaillé sur le projet devrait donner son accord, ce qui n'est pas gagné pour certains, dont Denis-Courmont.

Si VLC n'avait jamais été publié sous GPL, ou plutôt, s'ils n'avaient jamais accepté de contributions extérieures placées sous GPL, alors il n'y aurait pas de problème. Mais sans la forte protection apportée par la GPL, ils n'auraient certainement pas attiré autant de contributeurs. Impossible de dire si le projet aurait abouti ou pas.

Conclusion

Lorsqu'un projet choisit une licence comme la GPL ou tout autre licence copyleft, en opposition à des licences plus ouvertes, comme la MIT ou la BSD, les développeurs veulent s'assurer que toutes les œuvres dérivées seront libres. Cette protection a cependant un prix. En choisissant la GPL, l'équipe de développement originelle de VLC a fait la promesse à tous ses futurs collaborateurs que leur travail resterait libre, pour toujours. Ce que dit ouvertement la FSF et ce que concède implicitement Apple, c'est que l'App Store ne reconnaît cette définition de la liberté. Aucune des deux organisations n'est réputée pour accepter des compromis, donc,

jusqu'à ce que l'une d'entre eux cède, les applications construites sur du code sous GPL, comme VLC, n'honoreront pas l'App Store de leur présence. C'est dommage, c'est sûr, mais pour les deux camps c'est une question de principes bien plus importants que quelques applications.

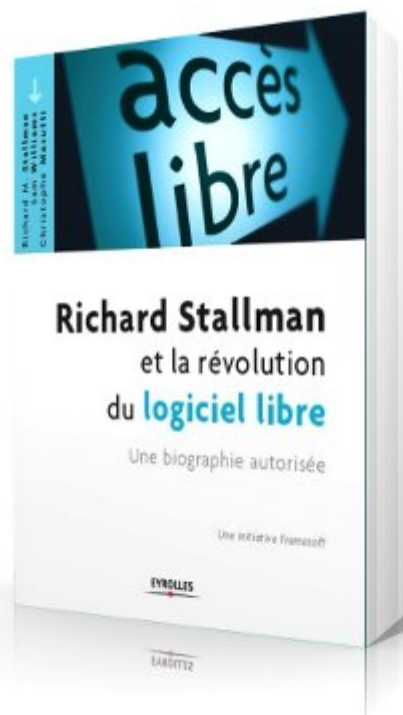
Je finirai en m'adressant aux commentateurs : ce débat englobe bien d'autres aspects, certains que j'ai juste abordés, d'autres que j'ai complètement omis, parfois par souci de clarté ou encore par manque de compétence. Je vous demanderai donc simplement d'être indulgents car c'est un sujet complexe et même un article de cette longueur ne permet pas d'en faire le tour. Mais je vous en prie, débattuez-en !

Notes

[1] Crédit photo : [Chris Willis](#) (Creative Commons By)

**Il était une fois un livre
sur et avec Richard Stallman**

Le livre [Richard Stallman et la révolution du logiciel libre – Une biographie autorisée](#), qui est sorti le 21 janvier aux éditions Eyrolles, possède trois auteurs.



Il y a bien entendu Sam Williams, auteur de la version d'origine en anglais. Mais si nous n'étions qu'en face de sa simple traduction, il demeurerait alors le seul auteur de l'ouvrage. Or deux autres noms apparaissent : Richard Stallman lui-même et [Christophe Masutti](#) de Framasoft.

Pourquoi cette originalité et pourquoi méritent-ils tous deux de venir s'associer à Sam Williams ? Vous le saurez en parcourant l'histoire peu commune de ce livre, telle qu'elle a été vécue par Christophe.

Il nous propose une très jolie formule pour expliquer les intentions et les apports de Stallman : il a souhaité « [hacker](#) » le livre.

Avec le même état d'esprit, ou plutôt la même [éthique](#), que lorsqu'il se trouvait, plusieurs dizaines d'années auparavant, jeune programmeur parmi les siens au département de recherche en intelligence artificielle du MIT.

Il était une fois un livre sur et avec

Richard Stallman

Christophe Masutti – janvier 2010 – GNU Free Documentation License



Tout a commencé en mars 2007, lorsque Alexis Kauffmann a posté un [message](#) sur le [forum](#) du réseau [Framasoft](#), invitant les volontaires à participer à la traduction du livre de Sam Williams, [Free as in Freedom: Richard Stallman's Crusade for Free Software](#), publié en 2002 chez O'Reilly sous la licence libre [GNU Free Documentation License](#).

Framasoft a une certaine habitude des traductions et s'est même constitué avec le temps une équipe entièrement dédiée à cela, le groupe [Framalang](#). Il se trouve qu'à cette époque Framalang ne chôlait pas et nous ne souhaitons pas leur rajouter une charge de travail supplémentaire avec un livre de quelque 300 pages !

Nous avons donc fait le choix de proposer directement le projet dans un wiki public, et pas n'importe lequel d'entre eux : [Wikisource](#), la bibliothèque libre du réseau [Wikimedia](#). Lors d'une [conférence](#) tenue aux [Rencontres mondiales du logiciel libre 2009](#) de Nantes, Alexis donne plus de détails sur le mode opératoire : quelqu'un avait déjà traduit la préface et le premier chapitre du livre sur un site personnel, ce qui nous a servi de base pour créer la structure de l'ensemble du projet [sur Wikisource](#). L'objectif était bien entendu d'arriver à nos fins mais il s'agissait également d'une expérience, celle d'une ambitieuse traduction collaborative et spontanée ouverte à tous, exactement comme on crée, modifie et améliore les articles de Wikipédia.

Un an plus tard, le bilan était contrasté. Nous avons bénéficié de l'exposition de Wikisource et de nombreuses personnes étaient venues participer. Mais quantitativement il restait encore beaucoup à faire et qualitativement ce qui

avait été fait manquait singulièrement de cohérence et d'harmonie (ne serait-ce que pour se mettre d'accord sur la traduction de tel ou tel terme important et récurrent). Et puis nous butions sur des questions aussi élémentaires que la décision de « clore » un chapitre traduit, ce qui sur un wiki aussi fréquenté ne va pas de soi.

Ne nous en déplaise, il fallait mettre un peu de [« cathédrale dans le bazar »](#), c'est-à-dire un peu de verticalité dans l'horizontalité du projet. Alexis a alors pris la décision de [relancer le projet sur le blog](#) de Framasoft, en invitant les bonnes volontés à se regrouper sur une [liste de discussion](#) dédiée et créée pour l'occasion. Pour ma part, je pris l'initiative d'animer cette liste qui compta rapidement une bonne dizaine d'inscrits (dans le livre vous trouverez en préambule les remerciements à cette liste nominative de participants).

Notre première décision consista à créer ailleurs un deuxième wiki, mais cette fois-ci loin des regards du réseau Wikimedia. Il ne s'agissait pas de jouer les cachottiers, mais nous en étions arrivés à la conclusion qu'il n'était plus possible de continuer à travailler sur Wikisource, dans la mesure où à tout moment une personne externe à la liste pouvait s'en venir tout « bouleverser » (c'est, j'en conviens fort bien, ce qui fait tout le charme de Wikipédia, mais à cette période d'un projet si spécifique nous souhaitions avant toute chose être efficaces et coordonnés dans notre travail).

Nous nous trouvions cependant face à un dilemme : d'un côté la traduction d'origine sur Wikisource restait bien entendu ouverte et continuait sa vie de texte wiki (bien que fort peu active, puisque la liste avait capté une bonne part de l'énergie disponible) et de l'autre côté, le travail sur notre nouveau wiki prenait forme et la traduction avançait plutôt bien. Une fois terminée, notre intention était de reverser la traduction de *Free as in Freedom* dans Wikisource, quitte à « écraser » les contributions effectuées dans l'intervalle

(ces contributions peuvent néanmoins être réhabilitées grâce à l'historique des modifications). Aujourd'hui, on peut donc trouver sur Wikisource cette [traduction française](#) de *Free as in Freedom* publiée par Sam Williams en 2002. Modulo le fait que quelqu'un est peut-être venu en déplacer un mot il y a une heure ☐

Notre traduction avançait donc plutôt bien jusqu'à obtenir une forme convenable à la relecture en novembre 2008, avec en prime la décision définitive d'en faire un volume de plus de la collection de livres libres [Framabook](#).

Une petite parenthèse est ici nécessaire. En effet, nous travaillions depuis peu en étroite collaboration avec l'équipe de [La Poule ou l'Oeuf](#), qui est une chaîne éditoriale en ligne pour la production de livres, pensés comme unités d'une collection, permettant un rendu final de type TeX, PDF, ePub ou HTML. Ce livre était aussi pour nous l'occasion d'implémenter dans le système non seulement l'ouvrage mais aussi la maquette générale de notre collection Framabook. Nous sommes très heureux du résultat car non seulement la Poule ou l'Oeuf a servi pour la mise en page du livre publié chez Eyrolles, mais ce sont désormais tous les framabooks qui vont bénéficier de la puissance de cet outil et des compétences de l'équipe de la Poule ou l'Oeuf.

Parenthèse fermée. Un mois plus tard, en décembre 2008, j'écrivis à Sam Williams pour lui demander une préface. Il accepta, tout en me précisant qu'il aurait bien aimé que Richard Stallman eût participé aux éventuelles modifications de l'ouvrage en anglais mais que finalement cela ne s'était pas produit. À ce moment-là, je ne compris guère l'allusion qui trouva son explication quelques jours plus tard.

Nous réfléchissions également aux illustrations possibles de l'ouvrage. Il y avait une belle [photo](#) de Richard Stallman dans le livre d'origine chez O'Reilly, tirée du [site web personnel](#) de Richard. Je contacte donc ce dernier, non seulement pour

lui demander l'autorisation d'utiliser sa photo, mais aussi pour l'informer que nous comptons publier la traduction en français (une [traduction en italien](#) avait été publiée en 2003).

C'est là que tout a basculé (positivement).

Il faut savoir que le livre *Free as in Freedom* n'a jamais obtenu l'appui formel de Richard Stallman. Pire : Richard aurait déjà affirmé qu'il était hors de question de venir lui demander un autographe sur le livre. On peut légitimement se demander pourquoi... Certes le travail de Sam Williams est d'abord un travail de journaliste, et il dresse un portrait sans concession de la personnalité de Richard Stallman : introverti, controversé, irascible et intransigeant. Tous ceux qui se sont rendus au moins une fois à l'une de ses conférences et qui sont au courant de ses activités, ont une bonne idée de ce que je veux dire.

Mais ce n'est pas sur ce point que Richard Stallman était en désaccord avec l'ouvrage : il y avait des erreurs manifestes voire des interprétations faussées dans les faits concrets relatés dans l'ouvrage. En somme, un travail mené un peu trop vite et sans assez de recul. Par ailleurs, un programmeur de l'envergure de Richard Stallman met un point d'honneur à vouloir reformuler avec exactitude les approximations, même lorsqu'elles ont été commises volontairement dans le but de rendre l'ouvrage plus accessible. Il n'en demeure pas moins que sous le prétexte de l'accessibilité, certaines approximations transformaient carrément le propos ou les événements. La manière dont Richard a agit sur le texte est relatée dans sa préface et lorsque le propos relève notamment de l'interprétation personnelle de Sam Williams, les ajouts de Richard sont clairement identifiés dans le livre par des encarts et des notes de bas de page. Les lecteurs pourront donc se faire une bonne idée des transformations ainsi opérées, quitte à aller voir et comparer avec l'original de Sam Williams.

Je prends un exemple : lorsque Sam Williams relate la tension entre [Eric S. Raymond](#) et Richard Stallman, on comprend que Raymond accuse Richard d'être la principale cause du manque de réactivité du [projet Hurd](#) (le projet de noyau du [système GNU](#)), et que cette accusation est fondée (on se doute néanmoins que Raymond n'a pas bien digéré la fin de non recevoir pour les modifications de l'éditeur [Emacs](#) qu'il voulait imposer). Pour Williams, et aussi pour Raymond, c'est le « micro-management » à la Stallman, c'est à dire sans concession, qui a freiné Hurd, avec pour conséquence la popularisation du [noyau Linux](#), qui obéit, lui, à un schéma de développement beaucoup plus ouvert. Il serait pourtant simpliste de se contenter de cette interprétation. Richard l'explique autrement, tant en montrant que Hurd n'est pas une mince affaire qu'en montrant aussi que le noyau Linux n'est pas la panacée du point de vue technique comme du point de vue éthique (le plus important à ses yeux).

Bref, suite à mon courriel, Richard me répondit qu'il désirait apporter quelques précisions sur l'épisode [LMI](#) et [Symbolics](#), deux entreprises qui débauchèrent le gros de l'équipe de [hackers](#) du [MIT](#) au début des années 1980. Cet épisode était très important, mais il ne touchait en gros qu'une dizaine de paragraphes dans l'ouvrage. Lorsque j'en fis référence à l'équipe de notre liste de discussion, tout le monde approuva l'idée.

Pourtant, au fil des échanges, Richard me confia qu'il n'avait jamais lu le livre de Sam Williams, et qu'en lisant les chapitres en anglais que je lui envoyais (repris depuis le site personnel de Sam Williams), il ressentait fortement le besoin de le réécrire.

Et tout l'art du hacker se révéla.

Alors que je lui suggérais d'écrire lui-même son autobiographie (d'un certain côté, j'anticipais sur mes craintes : la retraduction de l'ensemble à partir de toutes ces nouvelles modifications !), il se contenta de me renvoyer

des chapitres réécrits, sur lesquels je faisais un [« diff »](#) (une [commande Unix](#) permettant d'afficher les différences entre deux fichiers) pour pouvoir implémenter ces modifications dans la traduction française.

Après tout, qu'est-ce qu'un hacker ? Le lecteur en trouvera une bonne définition historique en annexe de l'ouvrage. L'essentiel est de comprendre que « hacker » signifie surtout améliorer, et qu'un bon hacker qui se respecte ne supporte pas l'imperfection. En toute logique, Richard ressentait tout simplement l'envie irrésistible de « hacker » le livre de Sam Williams. Qui d'autre sinon lui ?

J'énonce tout ceci avec le recul que me permet la parution de l'ouvrage. Dans les faits, je recevais plusieurs courriels par semaine contenant les modifications de Richard. Je les implémentais après les avoir lues, demandé des précisions, et argumenté lorsqu'elles étaient discutables. Bref, un travail soutenu qui nous amena Richard et moi jusqu'au début de l'été 2009.

Un mois auparavant, Alexis avait rencontré [Muriel Shan Sei Fan](#), directrice de la collection *Accès Libre* aux éditions Eyrolles. Et entre la poire et le fromage, il évoqua « l'aventure » de cette traduction qu'il continuait à suivre attentivement. Muriel trouva le projet tant est si bien intéressant qu'elle nous proposa de le publier chez eux.

Nous acceptâmes, mais ce serait vous mentir que d'affirmer que ce fut une décision facile et unanime dans l'équipe. En effet, nous avons, et nous avons encore, nos habitudes chez [InLibroVeritas](#), l'éditeur si particulier et attachant de tous les autres framabooks, avec qui nous travaillons main dans la main depuis des années pour défendre et faire la promotion du logiciel libre et sa culture.

Plusieurs arguments ont cependant pesé dans la balance. Tout d'abord nous n'avions plus affaire cette fois à un livre sur

un logiciel libre particulier ([Thunderbird](#), [OpenOffice.org](#), [LaTeX](#), [Ubuntu](#)...). Nous étions face à un livre mutant, une traduction devenue « biographie autorisée » car modifiée et enrichie pour l'occasion par son sujet même. Et pas n'importe quel sujet : la plus illustre des personnalités de la jeune histoire du logiciel libre. Cela méritait assurément de rechercher la plus grande exposition possible. Outre sa capacité de diffusion et distribution, Eyrolles nous offrait aussi son expertise et expérience. Le livre avait été traduit et une première fois relu, mais nous étions néanmoins conscients de sa perfectibilité de par les conditions mêmes de sa réalisation mentionnées plus haut (sachant de plus qu'à ce moment précis de l'histoire Richard n'en avait toujours pas fini avec ses propres modifications). Eyrolles a ainsi retravaillé le texte et mis à disposition du projet plusieurs personnes non seulement pour effectuer de nouvelles relectures mais parfois même pour retraduire certains passages. J'ajoute que nous apprécions la collection pionnière [Accès Libre](#) qui abrite en son sein de nombreux ouvrages de qualité sur le logiciel libre. Et enfin dernier argument et non des moindres : sous notre impulsion, Eyrolles allait pour la première fois publier d'emblée un livre sous licence libre, avec tous les avantages et inconvénients que cela suppose.

Nous nous rencontrâmes in the real life, Muriel, Richard, Alexis et moi, au cours d'un déjeuner en marge des Rencontres mondiales du logiciel libre de Nantes en juillet 2009. Nous discutâmes des modalités de publication et surtout, justement, de la question de la licence. L'ouvrage d'origine étant sous licence GNU Free Documentation License (à cause d'un Stallman insistant, Sam Williams s'en explique à la fin de son livre), sa traduction ne pouvait que l'être également. Or publier sous licence libre n'était pas dans les habitudes d'Eyrolles et cela ne rentrait pas forcément dans les « cases » de ses contrats types (rien que le fait d'interdire la classique interdiction de photocopier était une nouveauté). De plus nous connaissons les positions sans concession de Stallman dès que

l'on touche à ces questions de licence. Mais nous avons néanmoins réussi sans trop de mal à nous mettre d'accord, et il faut rendre ici hommage aux éditions Eyrolles d'avoir su s'adapter et innover.

La dernière ligne droite fut en tout cas aussi passionnante que stressante, avec ses nombreux va-et-vient entre Richard (apportant ses dernières modifications), Eyrolles (éditant à la volée l'ensemble de l'ouvrage), La Poule (obligée à mettre en forme un texte sans cesse en mouvement) et moi (dispersé un peu partout). Toujours est-il que vers la fin décembre 2009, ouf, nous étions prêts et le projet bouclé. Nous méritions tous ce beau cadeau de Noël que nous nous offrions là ☐

De leur côté, Richard Stallman et [John Sullivan](#) vont très prochainement publier en anglais le livre dans sa nouvelle version, aux [éditions internes](#) à la [Free Software Foundation](#), ajoutant ainsi une dimension supplémentaire au projet. Non seulement nous touchons les lecteurs francophones, mais le monde anglophone pourra aussi se délecter de ce « hack biographique ». Grâce à la licence libre (et aux efforts de quelques uns), le livre, parti des États-Unis, revient à la maison après un détour par la France qui l'aura transformé !

Pour moi, ce livre n'est pas seulement une biographie, même s'il en a l'apparence et la saveur. Il s'agit d'une histoire, celle du mouvement pour le logiciel libre, qui a influencé profondément l'histoire générale de l'informatique depuis la fin des années 1960. On considère généralement cette histoire à travers celle de l'industrie logicielle ou des composants d'ordinateurs. Mais il manque souvent une approche en termes de pratiques d'ingénierie et de circulation de l'information. Le logiciel libre constitue en cela une belle illustration de l'ingénierie logicielle, qui avance non seulement par projet, mais aussi parce qu'elle est fondamentalement un débat permanent sur la disponibilité et le partage de l'information. Lorsque le partage d'idées est impossible (notamment à cause des brevets) et lorsque les développeurs et les utilisateurs

sont restreints dans leurs libertés, alors c'est la société toute entière qui pâti de la pénurie de code et de libertés.

Tous les métiers ont leur déontologie. Les informaticiens ont une éthique et ceux qui la distillent sont les hackers. Par delà ses actes et ses programmes, l'un des principaux mérites de Richard Stallman est d'avoir réussi à concentrer et matérialiser cette éthique dans une simple licence (la fameuse [GNU General Public License](#)), qui va non seulement fonder, défendre et diffuser le logiciel libre mais qui constitue aujourd'hui une référence et une source d'inspiration pour d'autres mouvements émancipateurs. En fait, les programmes ont toujours été libres, et c'est un non-sens éthique qui les a rendu privés à un moment donné. L'histoire de l'informatique est heureusement loin d'être terminée.

Celle de ce livre touche par contre à sa fin, puisqu'il sera officiellement publié le 21 janvier prochain sous le titre *Richard Stallman et la révolution du logiciel libre – Une biographie autorisée*. Je remercie chaleureusement tous ceux que cette aventure a mis sur notre chemin. Toutes ces rencontres qui font aussi le sel d'un tel projet. À Framasoft, nous sommes fiers d'avoir pu le mener à son terme. Et malgré le labeur qui n'a pas manqué, ce fut un réel plaisir. Plaisir que nous espérons désormais partager avec le lecteur...

Cette histoire touche donc à sa fin, disais-je, mais, licence libre oblige, rien ne dit qu'il ne subisse pas à l'avenir d'autres métamorphoses. Ne serait-ce que parce que Richard est heureusement toujours parmi nous et qu'il lui reste encore certainement de belles pages à écrire dans le livre de sa vie.

Viol de la licence GPL : Microsoft s'excuse et fait profil bas

La semaine dernière Microsoft a dû retirer de ses serveurs un logiciel qui permet d'installer Windows 7 à partir d'une image ISO, après qu'un blogueur a découvert que le logiciel était une modification dissimulée d'un logiciel libre sous licence GPL dont Microsoft hébergeait lui-même les sources.



On attendait une réaction publique et officielle, la voici sous la plume de Peter Galli, rien moins que le « Open Source Community Manager for Microsoft's Platform Strategy Group »^[1].

Mise à jour de l'outil USB/DVD de Windows 7

[Update on the Windows 7 USB/DVD Tool](#)

*Peter Galli – 13 novembre 2009 – Port 25
(Traduction Framalang : Gilles)*

Comme vous avez pu le lire, et tel que cela a été indiqué ici, nous avons mené une enquête concernant l'outil de d'installation de Windows 7 depuis une clé USB, qui pouvait contenir du code sous licence GPL V2. le WUDT (NDT : Windows 7 USB/DVD Download Tool) est un outil gratuit, offert par le Microsoft Store, qui permettait à nos clients de créer une clé USB amorçable ou un DVD de sauvegarde à partir de l'édition électronique de Windows 7, disponible sous forme d'image ISO.

Après avoir analysé le code en question, nous sommes désormais en mesure de confirmer que tel était bien le cas, bien que cela n'ait pas été fait de manière délibérée. Bien que nous ayons confié cet outil à un développeur externe, nous partageons la responsabilité de cette situation; ce code est passé au travers des mailles de notre processus de revue de code. Nous avons également procédé à une revue d'autres portions de code disponibles dans le Microsoft Store, et il s'agit du seul incident que nous avons pu y découvrir.

Lorsqu'il est porté à notre attention qu'un composant de Microsoft contient du code développé par un tiers, notre objectif est de respecter les termes contractuels selon lesquels ce code est partagé. En conséquence, nous publierons le code source, ainsi que les fichiers binaires, de cet outil dès la semaine prochaine, sous les termes de la licence General Public Licence V2 (GPL V2), et prenons également des mesures pour améliorer nos futures revues de code, grâce à cette expérience

Nous nous excusons auprès de nos clients pour tout désagrément que cela a pu leur causer.

Notes

[1] Crédit photo : [Miss Rogue](#) (Creative Commons By-Sa)

Pourquoi j'utilise la licence GPL ou les états d'âme d'un

développeur

Voici une traduction intéressante à plus d'un titre.



D'abord parce qu'elle donne la parole à un développeur de logiciel libre (dont nous n'oublions pas ce que nous leur devons, c'est-à-dire tout). En l'occurrence il s'agit de [Zed Shaw](#), bien connu dans la communauté [Python](#) et RoR ([Ruby on Rails](#)).

Ensuite parce qu'elle évoque la classique opposition à l'intérieur même des licences libres entre celles de type [GPL](#) et celle de type [BSD](#). C'est toute la question du [copyleft](#) et de la viralité de la licence. Le copyleft est une apparente restriction, puisqu'il impose de diffuser le code modifié sous la même licence. Mais peut-être que cette contrainte est paradoxalement [une garantie supplémentaire de liberté pour l'utilisateur](#) ?

Enfin parce qu'elle aborde de nombreuses problématiques liées à l'open source et au monde de l'entreprise. Le fait de cacher à ses clients que l'on a utilisé du logiciel libre dans les solutions que l'on propose est une attitude malheureusement très fréquente (permettant de faire croire à moindre frais que l'on est hyper-compétent et que l'on a travaillé dur sur le projet). Il y a aussi toute la logique des startups, dont on attend un retour sur investissement presque immédiat, et qui du coup font de la rétention de code pourtant libre au départ.

Et puis il y a la question morale de la reconnaissance et celle très concrète du gagne-pain du développeur, frustré d'être ainsi ignoré et constatant non sans une certaine irritation que d'autres en profitent à sa place.

Voilà donc, entres autres arguments, pourquoi l'article ci-dessous mérite attention. Fatigué qu'une énième personne vienne lui demander pourquoi son code n'est pas sous licence BSD, Zed Shaw a décidé de réagir avec ce plaidoyer pugnace pour la licence GPL.

« Je veux que les gens apprécient le travail que j'ai fait et la valeur de ce que j'ai réalisé. Pas qu'ils passent en criant « pigeon » au volant de leurs voitures de luxe. »

Remarque : L'auteur concentre dans son titre les trois licences de la famille : la [GNU/GPL](#), la [LGPL](#) et l'[AGPL](#)^[1].

Pourquoi je (A/L)GPLise

[Why I \(A/L\)GPL](#)

*Copyright Zed A. Shaw – 13 juillet 2009 – Blog personnel
Avec l'aimable autorisation de l'auteur
(Traduction Framalang : Poupoul2 et Cheval boiteux –
Remerciements à Léviathan)*

Dans le monde de Python, la GPL est fréquemment critiquée, la plupart préférant utiliser une licence plus permissive telle que BSD, MIT ou [Python](#). Il est donc compréhensible que des gens soient en colère parce que j'ai placé [Lamson](#) (NdT : *Serveur SMTP développé en Python*) sous licence GPL. Nombreux sont ceux qui détestent cette licence, pensant qu'elle contrevient à l'esprit de Python.

J'aimerais cependant expliquer pourquoi **moi** j'utilise la GPL, après des décennies d'écriture de logiciels open source et quelques projets reconnus. Ce sont mes raisons de l'utiliser, elles ne s'appliquent qu'à moi et à ce que je souhaite faire

de mon logiciel dorénavant. Vous êtes libres de vos opinions et choix, et j'espère que vous respecterez les miens.

C'est le droit de l'auteur

Je crois que ce qu'un auteur souhaite faire avec son travail est son droit le plus strict. Il l'a écrit, alors que tout le monde lui expliquait que ça ne fonctionnerait pas. Il s'est enchaîné à ce travail plutôt que de sortir avec des amis. Il en a corrigé les bogues, écrit sa documentation de façon à ce que tout le monde puisse l'utiliser. Il a peut-être même passé du temps à en assurer la promotion et à aider les gens. Tout cela gratuitement, et pour des raisons qui lui appartiennent.

Lorsque vous dénigrez un projet ou une personne pour son choix de licence, vous êtes un **gigantesque** abruti. L'auteur a travaillé sur ce projet, pas vous. Au minimum, vous insultez les croyances de cette personne dans le logiciel libre et open source, mais aussi son sentiment que le logiciel libre et open source fait progresser notre culture.

Au pire, vous êtes un gros et ingrat malotru, parce que quelqu'un vous donne un logiciel, alors que vous insultez son travail non pas sur des critères techniques, mais sur une licence que vous désapprouvez.

Je me fiche totalement de la licence que les autres utilisent pour leurs logiciels, c'est **leur** logiciel et râler à cause de la licence qu'ils ont choisi est injurieux. Ils l'ont écrit, y ont sué sang et eau, et vous devriez être reconnaissant d'avoir le privilège de simplement y avoir accès.

Voici la première raison pour laquelle j'utilise la GPL :

Parce que c'est mon choix, et si vous n'êtes pas d'accord avec ça, n'utilisez pas mon logiciel. C'est aussi simple que ça.

Je ne veux plus être ignoré

J'ai écrit [Mongrel](#) (*NdT : Serveur et librairie HTTP pour Ruby on Rails*) et puis je l'ai donné, avec l'espoir que ça en aiderait d'autres et que le donner me rapporterait d'une manière ou d'une autre. Peut-être un boulot, ou un peu de respect, ou encore ma propre entreprise créant d'autres logiciels tels que celui-ci.

Mongrel a eu un grand succès, et de très nombreuses entreprises gagnent beaucoup d'argent avec. Non seulement permet-il à Rails de fonctionner, mais également à quasiment tous les frameworks Web Ruby, tel que les serveurs web Ruby, et a même été porté dans d'autres langages. Mongrel est et a été un super projet et j'en suis réellement fier.

Laissez moi vous expliquer à quel point Mongrel est avancé. Vous rappelez vous de la nouvelle attaque récemment publiée sur Apache appelée [Slowloris](#) (*NdT : attaques par Déni de Service via des requêtes HTTP partielles*) ? J'avais en fait prévu cette attaque, et écrit Mongrel de manière à ce qu'il y soit résistant (autant que Ruby me l'ait permis). Je l'avais appelé la « trickle attack » (*NdT : attaque par écoulement*), et j'en avais même fait la démonstration. C'était en 2004. Il y a **cinq** ans. Je l'avais même ajouté comme attaque à [RFuzz](#) en 2005.

Malheureusement, le succès de Mongrel ne m'est pas revenu. Bien que tout le monde utilise mon logiciel, la grande majorité des entreprises utilisatrices étaient des startups, et la dernière chose que les startups admettent c'est qu'elles ne possèdent pas leur propriété intellectuelle. Elles souhaitent que tout le monde, et particulièrement les sociétés de capital risque et les investisseurs, croit qu'elles sont des génies qui ont « innové » dans tout ce qu'elles font fonctionner.

Lorsque je regardais autour de moi, les entreprises n'avaient

aucun problème à admettre qu'elles utilisaient Ruby on Rails, mais elles n'en disaient pas plus, ce qui signifiait que lorsque j'essayais de trouver du travail, il m'était impossible d'expliquer l'ampleur de l'impact de Mongrel. Pour elles, il ne s'agissait que d'un simple serveur Web que leur administrateur système devait utiliser, Ruby on Rails était le véritable pourvoyeur d'argent.

Voici la seconde raison pour laquelle j'utilise la GPL :

À cause de l'expérience Mongrel, j'ai presque **besoin** que les entreprises soient obligées d'admettre qu'elles utilisent mon logiciel. Je préférerais à la limite que personne n'utilise mon logiciel, plutôt que de me trouver dans une situation où tout le monde l'utilise et personne ne l'admet.

Pire, tout le monde l'utilise, et en même temps me dit que je ne sais pas développer.

Le capital risque a changé l'industrie

L'industrie du logiciel a changé depuis l'an 2000. Je mettrais en fait le curseur à l'entrée en bourse de Google, mais je ne peux pas dire exactement à quel moment. Ce que je peux dire, c'est que les méthodes actuelles de progression des entreprises technologiques induisent l'exploitation du code source plutôt que la contribution.

Je ne peux pas réellement en vouloir à ces startups, elles sont conçues ainsi. Une entreprise adossée à un capital-risqueur aura presque **toujours** un objectif de gain à court terme, dans l'espoir d'apporter plus tard un joli retour sur investissement. Pour un capital-risqueur, un retour sur 20 ans n'est pas acceptable, particulièrement si vous prévoyez de faire cela sans offre et des stocks options disponibles pour tout le monde.

Pour une entreprise financée par un capital-risqueur, il ne

s'agit toujours que d'augmenter des revenus pour donner l'illusion de la croissance, afin que les actionnaires investissent et mènent le prix de l'action aussi haut que possible. Il **n'est pas** dans leur intérêt de **ne pas** faire d'offre. Ils veulent de l'argent maintenant, autant que possible, et ne sont pas intéressés par l'investissement technologique ou humain.

Ce qu'elles veulent, ce sont des tonnes de technologie gratuite qu'elles peuvent cacher aux investisseurs. Elles sont ravis que cette technologie soit maintenue par des gens qui ne l'ont pas écrite, pour que les employés ne puissent pas réclamer leur dû plus tard.

Cette gestion à courte vue, combinée au financement limité de départ, signifie que ces entreprises exploitent l'open source. Elles l'utiliseront, gagneront leur argent, et partiront lorsque ceci sera fait. Ce qui, en fait, est parfaitement logique, parce que ainsi vont les choses, et honnêtement, si vous êtes une entreprise qui cherche à gagner de l'argent de cette manière, c'est ce que je vous suggèrerais de faire.

Pourtant, le contrat tacite entre les entreprises et les développeurs open source est désormais révolu. Je n'ai aucune raison de leur offrir un usage sans restriction de mon logiciel, puisqu'ils ne sont intéressés que par la transformation de mon logiciel en offre alléchante de mise sur le marché à court terme, à deux ou cinq ans.

Voici la troisième raison pour laquelle j'utilise la GPL :

Les entreprises du secteur technologique sont désormais conçues pour être créées et détruites rapidement au bénéfice de gros capital-risqueur, tout en maintenant des coûts très bas. Cela signifie qu'elles n'ont aucune prime économique à donner, aussi n'ai-je aucune prime sociale à leur « donner » mon logiciel.

Les développeurs sont des plagiaires

Ne nous voilons pas la face, tout le monde a besoin de manger et de payer son loyer. Pour beaucoup de gens, leurs besoins en argent sont modestes, et si vous travaillez sur quelque chose, vous espérez en retirer quelque chose qui vous aidera à satisfaire ces besoins. Il se peut que ce ne soit pas de l'argent. Peut-être est ce que ce sera du respect, ou des honneurs, mais vous espérez réellement quelque chose de votre travail.

Nous sommes d'accord. Aussi je trouve étrange un message twitter tel que celui-ci :

[ericholscher](#) @zedshaw Ce serait chouette si l'interface/code était sous licence BSD. Très utile au travail. L'interface REST marche aussi très bien.

Éric est un garçon très sympathique, aussi je ne vais pas m'acharner sur lui, mais si je traduis le message, ça pourrait donner ça : « Hey ! Ton logiciel est fantastique ! Je peux l'avoir gratuitement, pour pouvoir l'utiliser au boulot, plaire à mon patron et me faire de l'argent ? »

Honnêtement, combien d'administrateurs disent à leur patron que ce qu'ils utilisent provient du logiciel libre ? Combien d'entre vous disent aux investisseurs que toute votre infrastructure logicielle est basée sur un truc qu'un autre gars a écrit en plusieurs mois ? Combien d'entre vous vont voir leur dirigeants en disant : « Vous savez, il y a ce gars, Zed, qui a écrit le logiciel que nous utilisons, pourquoi ne pas l'embaucher comme consultant ? »

Vous ne le faites pas. Aucun d'entre vous. Vous prenez le logiciel, et vous l'utilisez tel Excalibur terrassant le dragon, et vous en ramassez ensuite tous les lauriers. Vous ne rendez rien, et en fait, j'ai croisé une grande majorité d'entre vous qui assèment constamment que je ne sais pas

développer, histoire de plus encore protéger son cul.

Il n'est désormais plus pardonnable et acceptable que vous tous plagiez le travail que vous utilisez. Comme vos employeurs ont besoin de l'illusion de « l'innovation », vous avez besoin de l'illusion d'être intelligent. Ce qui vous amène à mentir, et à voler tout ce qui vous tombe sous la main, comme si cela vous appartenait.

Ok, mais cela signifie que vous avez cassé le contrat tacite avec les développeurs et défenseurs de logiciels libres et open source. Aussi longtemps que vous me rendez ma paternité et que vous assurez la promotion de mon travail, je vous donnerai mon logiciel. Puisqu'aucun d'entre vous ne le fait, je n'ai aucune motivation à vous donner mon logiciel, de la même manière que je n'ai pas besoin de me conformer au contrat passé avec l'entreprise nourrie au capital-risque.

Voici la quatrième raison pour laquelle j'utilise la GPL :

J'utilise la GPL pour que vous restiez honnête. Vous **devez** désormais dire à vos patrons que vous utilisez mon boulot. Et ils s'en pisseront dessus de trouille. Parfait. Parce que j'ai aussi une solution pour ça.

Si c'est bien, payez pour l'avoir

J'aime travailler sur [Lamson](#), parce que faire des application de messagerie est tellement plus drôle que des applications Web. Lorsque je m'assois pour construire une application de courriel, ça ne nécessite qu'un ensemble de technologies et c'est terminé. Si j'ai besoin de faire une application Web, cela implique du design, des templates, du javascript, des bases de données, et plein d'autres machins.

L'autre raison pour laquelle j'aime écrire des logiciels de courriels est que **personne d'autre ne le fait**. Vous tous êtes de grosses buses, parce que même avec un projet tel que

Lamson, vous êtes encore tous effrayés par le gros monstre SMTP. Le jour où j'ai dit que je pourrais faire du Mail over REST, vous avez presque fait dans votre slip. Oui, parce que ça aurait été **tellement plus simple**.

Et j'aime donner Lamson, parce que c'est une partie de moi-même. C'est ce parfait mélange de camelote, bidouillages techniques, marketing et écriture que j'aime, et dans lequel j'excelle. Même si j'en arrive à faire des montagnes d'argent en construisant des choses avec Lamson, Lamson restera libre pour que je puisse rester honnête.

Mais j'écris aussi mes propres logiciels et projets avec Lamson. Je l'utiliserai pour créer autant d'incroyables services de courriel que je pourrais. Certains gratuits, certains payés par la pub, certains payants, mais tous utiliseront Lamson pour botter des culs avec du courriel.

Maintenant, compte tenu du fait qu'il n'y a aucune motivation pour une entreprise à admettre qu'elle utilise mon travail, et qu'il n'y a aucune motivation pour un développeur à me rendre mon travail, posez vous cette simple question :

Pourquoi diable est ce que je ne facturerais pas des gens qui ne sont pas capables d'utiliser correctement la GPL ?

Je sais, je sais, les conservateurs du logiciel libre pensent que cela fait de moi un traître, mais réfléchissez-y. Vous avez le logiciel, sous licence GPL, et aussi longtemps que vous êtes un projet open source taillé dans la pierre qui publie son code, vous êtes libre de l'utiliser et d'en faire ce que vous voulez.

J'aime l'open source, mais qu'en est-il des entreprises qui l'utilisent ? Les entreprises vont devoir payer à présent. C'est ainsi que fonctionne l'économie. Si c'est assez bon pour que vous l'utilisiez, alors c'est assez bon pour que vous le payiez.

Voici la cinquième raison pour laquelle j'utilise la GPL :

Je serai toujours un développeur open source, mais très franchement, nous crevons parce que des entreprises qui utilisent nos logiciels ne rendent rien. L'ironie de la situation est que pour pouvoir accroître ma motivation à faire de l'open source, j'ai besoin de me faire payer.

Je ne demanderai évidemment jamais rien à un projet open source puisqu'ils honorent le contrat tacite : si je donne, vous donnez.

Mais les jours des entreprises-minutes et des développeurs ingrats qui gagnent de l'argent sur mon dos avec mon logiciel sont terminés. Ma nouvelle devise est :

L'open source à l'open source, le business au business.

Si vous faites de l'open source, vous êtes mon héros et je vous soutiens. Mais si vous êtes une entreprise, alors parlons affaires.

Finalement, la valeur

Au final, je voudrais également donner une petite raison pour laquelle j'utilise la GPL : La valeur perçue. Lorsque les gens peuvent obtenir quelque chose gratuitement sans même avoir besoin d'y penser, ils lui donnent même moins de valeur que quelque chose pour lequel ils paient même pour une somme dérisoire. C'est juste la manière d'être des gens, et il n'y a pas grand chose à faire contre ça.

Ma dernière raison d'utiliser la GPL est que je crois que mes projets ont de la valeur, et je veux que les gens qui les utilisent perçoivent cette valeur. Ils ont même tellement de valeur que je veux les associer à un document légal complexe et totalement non testé comme preuve de leur utilité. Si je voulais faire simple, je les placerais simplement sous licence

BSD et tout le monde les utiliseraient.

Je veux que les gens apprécient le travail que j'ai fait et la valeur de ce que j'ai réalisé.

Pas qu'ils passent en criant « pigeon » au volant de leurs voitures de luxe.

Notes

[1] Crédit photo : [Phillie Casablanca](#) (Creative Commons By)