

PARTIE I

INTRODUCTION GÉNÉRALE : REPÈRES POUR COMPRENDRE LE MOUVEMENT DU LOGICIEL LIBRE

Yorick COOL et Philippe LAURENT*

CHAPITRE 1 : HISTORIQUE DU MOUVEMENT LIBRE

Section 1 : Naissance du mouvement

1. Il est courant d'entendre dire que Richard M. Stallman est l'inventeur, le père ou le parrain des logiciels libres. Ainsi que l'a rappelé le professeur Montero dans l'avant-propos du présent ouvrage, une telle assertion méconnaît complètement l'histoire de l'informatique, qui est parsemée de logiciels libres depuis sa naissance. Toutefois, il est vrai que l'importance de Stallman peut difficilement être exagérée dans la mesure où il est celui qui a tenté d'organiser le mouvement en tant que communauté consciente d'elle-même, et qu'il en a été le premier théoricien. C'est un personnage charismatique et excentrique², controversé mais respecté pour ses contributions tant informatiques qu'organisationnelles et théoriques à la communauté libre. Malgré les critiques qu'on peut lui adresser, il est indéniable que sans lui, le mouvement du logiciel libre n'aurait pas la même physionomie aujourd'hui.

2. Depuis la fin des années 70, Stallman travaillait au Laboratoire d'intelligence d'artificielle du Massachusetts Institute of Technology (M.I.T.),

* Les auteurs tiennent tous deux à remercier Fabian BASTIN et Frédéric BURLET pour leur aide précieuse et leurs éclaircissements techniques lors de la rédaction de cet article.

² Outre son apparence hirsute somme toute classique, Stallman est connu pour se déguiser en « Saint IGNUcius », en s'affublant d'une robe de prêtre et d'un vieux plateau de disque dur comme couvre-chef. <http://www.stallman.org/saint.html>. Ce type de comportement à vocation clairement humoristique lui vaut pourtant des critiques d'une certaine partie de la communauté libre, notamment pour les répercussions négatives qu'il pourrait avoir en termes d'image du mouvement auprès des entreprises. Pour des éléments biographiques sur Stallman, voy. S. WILLIAMS, *Free as in freedom : Richard Stallman's crusade for free software* disponible à l'adresse <http://www.faiozilla.org/toc.html> et les éléments biographiques fournis par Stallman lui-même <http://www.stallman.org/#serious>.

un des hauts lieux de la culture *hacker*³, dont une des habitudes était de partager librement logiciels et codes sources, afin que tout un chacun puisse améliorer les logiciels des autres à l'envi.

La petite histoire veut que c'est lors de l'arrivée d'une nouvelle imprimante Xerox que Stallman débuta sa carrière de prosélyte du logiciel libre⁴. L'appareil en question constituait un progrès énorme comparé aux machines de l'époque parce qu'il réduisait grandement les temps d'impression. Il comportait cependant un défaut de taille: il était très sensible aux bourrages papier, ce qui engendrait de nombreuses pertes de temps, faisant ainsi perdre le principal bénéfice de la nouvelle machine. Stallman décida d'implémenter une solution qu'il avait déjà utilisée pour des imprimantes plus anciennes, à savoir, ajouter au pilote de l'imprimante une fonction qui prévenait l'utilisateur lorsqu'il y avait un bourrage. Ainsi, le problème de bourrage serait plus vite réglé, et l'on profiterait mieux de l'appareil. Stallman se heurta toutefois à un problème : les codes sources du pilote n'étaient pas fournis avec l'imprimante, et Xerox refusait de les lui communiquer.

Cet évènement fut, semble-t-il, la goutte d'eau qui fit déborder le vase pour Stallman. Jusqu'alors, le monde universitaire, et le M.I.T. en particulier, avait pu ignorer superbement le fait que les logiciels étaient de plus en plus fréquemment exploités de manière propriétaire, c'est-à-dire sans que l'utilisateur puisse prendre connaissance du code source, copier, modifier ou redistribuer le logiciel. Mais cet incident bénin fit prendre conscience à Stallman du fait que les logiciels propriétaires étaient de plus en plus répandus, et devenaient incontournables. Alors que l'exploitation propriétaire avait longtemps été exceptionnelle, elle était en train de devenir la règle. Or pour un *hacker* comme Stallman, habitué à la liberté totale en matière de logiciels, une telle évolution était une dérive néfaste car limitant indûment la liberté des utilisateurs de

³ Le terme *hacker* désigne une personne particulièrement douée en informatique et qui connaît bien les arcanes de la programmation. Le grand public et les médias confondent fréquemment ce terme avec celui de *cracker*, désignant une personne souhaitant utiliser ses connaissances informatiques pour nuire ou obtenir un profit. La confusion est tellement répandue qu'on peut considérer que le mot en est venu à couvrir les deux sens. Le problème vient évidemment du fait que la première acception a une connotation positive et est considérée comme un compliment, alors que la seconde est connotée négativement et est utilisée péjorativement. Voy. notamment l'encyclopédie en ligne <http://fr.wikipedia.org/wiki/Hacking>; E. S. RAYMOND, *How to become a hacker*, http://www.catb.org/~esr/faqs/hacker-howto.html#what_is; du même auteur, *A brief history of hackerdom*, <http://www.catb.org/~esr/writings/cathedral-bazaar/hacker-history/>.

⁴ Stallman ne se souvient apparemment plus de l'année de cet incident, ce qui paraît paradoxal dans la mesure où il n'a de cesse de souligner son importance. Selon toute vraisemblance, il a eu lieu en 1983. Voy. E. S. RAYMOND, *A fan of freedom: thoughts on the biography of RMS*, <http://www.catb.org/~esr/writings/rms-bio.html>.

prendre connaissance de l'information et de l'utiliser afin de faire progresser la science⁵.

C'est à partir de ce moment que Stallman initia sa réflexion sur les logiciels libres et décida de lutter contre les logiciels propriétaires, qui sont selon lui « l'ennemi »⁶. Pour ce faire, il quitta le M.I.T. en janvier 1984 et lança le projet GNU. Celui-ci a pour but, encore à ce jour, de développer un système d'exploitation complètement libre. GNU signifie « Gnu's Not Unix », en référence au système d'exploitation phare des années 70 et 80 qui eut beaucoup de succès chez les *hackers* de par ses qualités techniques. Le projet déboucha sur toute une série d'utilitaires qui, associés plus tard au noyau Linux, deviendraient le porte-étendard de la cause des logiciels libres. Le projet continue à ce jour, et travaille notamment sur un noyau de système d'exploitation baptisé HURD.

3. En 1985, Stallman fonda la Free Software Foundation (F.S.F.), une A.S.B.L. ayant pour but de promouvoir les logiciels libres et d'en développer. En accomplissant le premier rôle, la F.S.F. est devenue un acteur de poids dans le monde du logiciel libre. Elle est aussi controversée tant pour ses idées que pour ses méthodes.

Du point de vue idéologique, la F.S.F. reprend sans compromis les idées libertaires de Stallman et d'une bonne partie des *hackers* des années 1970. Elle fait de la liberté du code une question éthique et politique. De son point de vue, l'appropriation privative d'un logiciel et de son code relève du vol. Le mode de distribution propriétaire crée des conséquences sociales néfastes qu'il faut éradiquer. Selon la F.S.F., il rétribue l'égoïsme et la cupidité, décourage le partage, l'avancée des connaissances, etc. De nombreuses voix s'élèvent pour critiquer ce discours comme étant trop radical et manichéen. Il est aussi souvent soutenu qu'un tel discours porte plus préjudice au mouvement libre qu'autre chose, parce qu'il le décrédibilise et le rend inattractif pour le monde de l'entreprise.

Les méthodes de la F.S.F. sont aussi contestées sur plusieurs plans. Tout d'abord, on lui reproche souvent son dogmatisme. Celui-ci se manifeste non seulement dans ses idées, mais également dans sa façon de les appliquer. Ainsi a-t-elle pour habitude de se servir de son autorité pour qualifier de non-libres des licences alternatives à la General Public licence (G.P.L.) (*infra*, n° 6), sans expliquer en quoi la licence n'est pas libre. Certains expliquent cette méthode par la volonté de la F.S.F. de voir la G.P.L. s'imposer au détriment des autres

⁵ Il est fréquent de tracer une filiation entre ce type de raisonnement et les origines universitaires des *hackers*. Voy. E. S. RAYMOND, *Homesteading the noosphere*, <http://www.catb.org/~esr/writings/cathedral-bazaar/homesteading/>.

⁶ Voy. par exemple <http://www.gnu.org/philosophy/free-software-for-freedom.html>.

licences libres. Dans le même ordre d'idées, il est permis de relever une contradiction entre le discours officiel encensant la liberté des utilisateurs et les nombreuses restrictions aux libertés de l'utilisateur que la F.S.F. tente d'imposer, afin de voir la G.P.L. prédominer⁷. Enfin, la F.S.F. intervient régulièrement pour peser dans des litiges – essentiellement aux Etats-Unis – et faire prévaloir des règlements de litiges extra-judiciaires. Certains de ses détracteurs prétendent qu'il s'agit en fait de manœuvres visant à éviter que certaines interprétations que la F.S.F. fait de la G.P.L. ne soient rejetées par les tribunaux, et donc à maintenir le monopole interprétatif de la F.S.F.

Que ces critiques soient fondées ou non, il est indéniable que Stallman et sa F.S.F. ont beaucoup apporté à la communauté libre. Si les logiciels libres existent depuis les débuts de l'informatique, c'est à Stallman que l'on doit d'avoir soulevé la question de leur particularité et d'avoir imprimé au mouvement une dynamique sans laquelle il n'aurait probablement pas la même ampleur aujourd'hui.

Section 2 : La juridicisation

4. Dès le départ du mouvement, les questions juridiques étaient au centre des débats puisque la distinction entre logiciels libres et logiciels propriétaires ne repose que sur une différence de droits concédés aux utilisateurs. On comprend de ce fait que la réflexion sur les licences remonte à la même époque que la fondation de la F.S.F.

La première question juridique que semble s'être posée la F.S.F. a été : comment empêcher que quelqu'un s'approprie un logiciel libre et l'exploite de manière propriétaire⁸ ? En effet, si l'on se contente de concéder à un licencié les droits d'utiliser, modifier, reproduire et distribuer un logiciel, rien n'empêche celui-ci de distribuer le logiciel à des tiers sans leur communiquer les codes sources et sans leur permettre de faire autre chose qu'utiliser le logiciel. C'est un des effets

⁷ L'exemple le plus frappant en la matière est celui des librairies (sur cette notion, *infra*, n° 136 et s.). La F.S.F. considère que la G.P.L. interdit que des librairies non-G.P.L. soient liées à un logiciel sous G.P.L. Une telle restriction limite la liberté tant des utilisateurs que des développeurs, sans que l'on en voie véritablement le bénéfice pour la communauté libre, puisqu'il s'agit en fait de tenter d'imposer la G.P.L. aux auteurs de librairies pour logiciels sous G.P.L. Pour un approfondissement de la problématique, voy. *infra*, n° 136 et s.

⁸ L'importance de cette question aux yeux de Stallman se comprend d'autant plus qu'il avait été victime d'une telle appropriation. Stallman avait écrit le logiciel Emacs. Steve Gosling, lui, avait écrit une version différente du même logiciel, diffusée elle aussi sous licence libre. Mais par la suite, les détenteurs des droits sur cette version du logiciel ont cherché à empêcher Stallman de réutiliser le code de Gosling. On trouvera un exposé plus complet de cet épisode – que nous avons largement simplifié – à l'adresse http://www.free-soft.org/gpl_history/.

des licences non-copyleftées que de permettre à un licencié d'exploiter le logiciel – modifié ou non – de manière propriétaire.

Les défenseurs les plus acharnés des logiciels libres, tels que Stallman et la F.S.F., considèrent qu'un tel résultat est négatif, puisqu'ils s'opposent farouchement à l'exploitation propriétaire de logiciels et ne veulent pas que l'industrie propriétaire profite de leurs avancées. C'est pourquoi ils ont forgé la notion de copyleft. Le terme, inventé en 1984 ou 1985 par Don Hopkins⁹, joue sur l'opposition avec le terme « copyright » et la polysémie de « left », signifiant non seulement « gauche », mais aussi « laissé »¹⁰. Il désigne le mécanisme par lequel les droits sont concédés à condition que le licencié ne limite pas les droits des tiers, notamment en distribuant le logiciel sous une licence plus restrictive des droits de l'utilisateur. En d'autres termes, si le licencié redistribue le logiciel, il doit le faire sous la même licence que celle dont lui-même bénéficie. A défaut, il commet à la fois une violation contractuelle et un acte de contrefaçon. Grâce à ce mécanisme, une fois que du code est diffusé sous une licence libre, un licencié ne peut plus jamais exploiter ce code de manière propriétaire. Voilà l'essence du copyleft.

5. Toutefois, il existe toute une gradation dans la mise en œuvre du copyleft. Ainsi peut-on dire, à la suite de F. Bastin¹¹, que les licences copyleftées se situent sur un spectre allant de ce qu'on pourrait appeler un copyleft faible à un copyleft fort. Ainsi, plus on avance sur ce spectre, plus l'obligation de redistribution libre est exigeante. Des exemples peuvent éclairer le propos.

La Mozilla Public License (M.P.L.) est une licence qui oblige à redistribuer le logiciel et ses modifications sous M.P.L. Toutefois, si un licencié intègre du code sous M.P.L. dans un ensemble logiciel plus vaste dont certaines parties sont originales, il ne doit pas fournir le code source des parties entièrement originales (art. 3.7). On peut dans ce cas parler de copyleft faible, puisque les ajouts ne sont pas couverts par le copyleft, seul le code originellement copylefté devant rester libre.

Selon l'article 1, c) de l'Open Source License (O.S.L.) par contre, tout logiciel contenant une portion de code sous O.S.L. devra être diffusé sous la même licence (s'il est diffusé). Cependant, l'interprétation courante de l'O.S.L. admet généralement que l'utilisation dynamique d'une librairie sous licence O.S.L. par un logiciel n'implique pas nécessairement que ce dernier doive également être diffusé sous O.S.L. Cette licence utilise donc un copyleft plus fort que la M.P.L.,

⁹ R. STALLMAN, *The GNU Project*, <http://www.fsf.org/gnu/thegnuproject.html>.

¹⁰ S. DUSOLLIER, « *Open source et copyleft : une remise en cause de la figure de l'auteur* », in *Autour de la figure de l'auteur*, Bruxelles, Bruylant, à paraître.

¹¹ <http://www.fundp.ac.be/~fbastin/computing/hackers/crid.pdf>.

mais plus faible que la G.P.L. qui considère que même un logiciel chargeant une librairie sous G.P.L. doit être soumis à cette licence. La G.P.L. est ainsi l'archétype du copyleft fort

On voit que le copyleft peut se manifester de plusieurs manières et être plus ou moins exigeant. Il faut d'ailleurs souligner que plus le copyleft est exigeant, plus on a la certitude que le logiciel et ses améliorations resteront entièrement libres. Mais malgré les différences d'application, l'idée reste toujours la même : la redistribution d'un logiciel libre ne peut se faire que sous la licence originale.

6. La première licence à consciemment incorporer ces principes fut la GNU Emacs¹² General Public License, rédigée par Stallman en 1985¹³. Elle marque un tournant majeur dans l'histoire des licences de logiciels libres. Jusqu'alors, les licences libres étaient totalement informelles, quand même elles étaient écrites. La plupart se résumaient à une ligne disant que l'on pouvait faire ce qu'on voulait du logiciel sans autre précision, ou encore que « vous pouvez copier ce logiciel en tout ou en partie, tant que vous n'en tirez pas un bénéfice et tant que vous ne prétendez pas l'avoir écrit ». Ecrites par des informaticiens pour refléter la culture *hacker* (voy. *supra*, n° 2), ces licences sommaires manquaient cruellement de précision ou de rigueur juridique. Au contraire, la GNU Emacs G.P.L. est structurée et emploie à bon escient des concepts juridiques. A cet égard, on peut dire qu'elle a joué le rôle de déclencheur de la réflexion juridique autour des logiciels libres.

En 1986, John Gilmore, un autre informaticien de renom et futur défenseur des logiciels libres, suggéra à Stallman de retirer de la licence toutes les mentions explicites du logiciel Emacs afin de créer une licence libre générique que la F.S.F. pourrait employer pour tous les produits GNU. Suivant ce conseil, Stallman révisa la GNU Emacs G.P.L. en profondeur et publia, en 1986, la version 1.0 de la General Public License : la G.P.L., licence utilisée pour près de 70% des logiciels libres¹⁴, était née.

En 1991, la F.S.F. publia la G.P.L. version 2.0, une révision en fait mineure de la première version. C'est le texte que nous connaissons aujourd'hui. Elle publia également la Library General Public License (L.G.P.L.) destinée avant tout aux librairies. Elle est identique à la G.P.L. à ceci près qu'elle permet au licencié de linker un logiciel propriétaire avec la librairie sous L.G.P.L. Elle a été légèrement révisée en 1999, et renommée Lesser General Public License, afin que son utilisation ne soit pas limitée aux librairies.

¹² Emacs est un éditeur de texte aux fonctionnalités très avancées qui a beaucoup fait pour la réputation informatique de Stallman.

¹³ S. WILLIAMS, *op. cit.*

¹⁴ Chiffre basé sur la répartition des projets par licence sur <http://freshmeat.net/stats/#license>.

Section 3 : Addition, multiplication et divisions

7. 1991 est une année charnière à un autre égard : c'est alors que Linus Torvalds, un étudiant finlandais, commença à son tour à élaborer un système d'exploitation libre, baptisé Linux¹⁵. Projet modeste à ses débuts, il connut un succès foudroyant. En effet, dès que possible, Torvalds publia les résultats de son travail et les soumit à la critique des utilisateurs du web. Plus encore, il encouragea les utilisateurs à lui soumettre toutes les modifications qu'ils jugeaient utiles, Torvalds décidant alors lesquelles seraient gardées. Ce mode de fonctionnement, encore en vigueur aujourd'hui, montre à quel point internet a été crucial au développement de Linux et des logiciels libres en général.

D'un point de vue technique, il paraît intéressant de souligner que le résultat du développement initié par Torvalds n'est en fait pas un système d'exploitation en tant que tel, mais un *kernel*, c'est-à-dire le noyau du système d'exploitation, la partie communiquant avec le *hardware* de l'ordinateur. Torvalds est donc le chef du développement du linux kernel, mais non du système d'exploitation Linux. Celui-ci se compose du linux kernel et de toute une gamme d'outils divers. Cet ensemble constitue ce qu'on appelle une « distribution linux ». Il est possible de coupler le linux kernel à beaucoup de logiciels et d'outils différents, ce qui explique le foisonnement de distributions différentes que l'on trouve sur le marché. Citons Red Hat, Debian, Mandrake, SuSe, Slackware, etc. Toutes ces distributions sont différentes en ce qu'elles ne constituent pas le même ensemble logiciel, mais sont néanmoins toutes compatibles car toutes sont organisées autour du linux kernel¹⁶.

Un autre point commun se dégage de toutes les distributions linux : toutes recourent massivement aux outils développés par le projet de GNU de Stallman. On peut ainsi dire que le linux kernel est un substitut au noyau HURD, initialement prévu pour être le noyau système d'exploitation GNU. Cet état de fait a conduit Stallman et la F.S.F. à exiger que le système d'exploitation Linux soit désigné par l'appellation « GNU/Linux », parce que sans le projet GNU, il ne serait pas viable. Par la suite, nous utiliserons les deux appellations indifféremment.

8. L'addition du linux kernel à l'environnement GNU a été cruciale pour la communauté du libre. Le produit qui en est résulté, le système d'exploitation GNU/Linux, a connu un succès tel que la visibilité des logiciels libres en a été largement accrue. Sa réputation de qualité et de sécurité a séduit de nombreuses

¹⁵ Le nom est une contraction de Linus et Unix.

¹⁶ La question est en réalité plus complexe, puisque la compatibilité dépend également des bibliothèques utilisées, de l'architecture informatique etc. On peut ainsi dire que le fait de partager le même noyau est une condition nécessaire, mais non suffisante pour assurer la compatibilité.

entreprises. Le nombre d'utilisateurs de Linux s'est multiplié exponentiellement et le grand public a découvert Linux notamment grâce au soutien que de gros fournisseurs de matériel informatique (IBM, Sun...) ont commencé à lui apporter à la fin des années 90. De ce fait, les mots « logiciels libres » et « open source » ont été sur toutes les lèvres pendant un temps.

9. Outre les qualités intrinsèques de Linux et le bouche-à-oreille, l'une des raisons principales de la popularité des logiciels libres depuis la fin des années 90 a été le développement du mouvement open-source. Celui-ci est né en 1998, à la suite de l'initiative des responsables de la distribution Debian qui souhaitaient définir plus clairement ce qu'ils pouvaient inclure dans la distribution en respectant « l'esprit du libre. ». Leur réflexion a débouché sur la création de l'Open Source Initiative (O.S.I.), organisme indépendant de Debian, et à la rédaction de l'Open Source Definition (O.S.D.). L'O.S.I. a vite attiré les membres de la communauté libre souhaitant se distancier de Stallman et de la F.S.F.¹⁷. Considérant que l'argumentaire essentiellement éthique de Stallman n'était pas à même de convaincre les entreprises de soutenir les logiciels libres, ils ont développé un discours plus axé sur les qualités techniques et les avantages économiques des logiciels libres. Au passage, le terme « open source » a été inventé afin de remplacer l'expression « free software » (logiciels libres). Ce dernier terme présente en effet une ambiguïté sur le sens du mot « free », qui peut signifier tant « gratuit » que « libre », alors que la distribution de logiciels libres n'est pas nécessairement gratuite. La confusion possible a été considérée par les fondateurs de l'O.S.I. comme de nature à détourner les entreprises des logiciels libres, d'où l'emploi d'« open source ».

Si le marché informatique a réagi très favorablement à la stratégie open-source, ce ne fut pas le cas de Stallman et de la F.S.F. En premier lieu, ils reprochent à l'O.S.I. de faire l'impasse sur les enjeux éthiques de la question des logiciels libres. En effet, pour Stallman et la F.S.F., le choix du logiciel libre est avant tout une question éthique. Les autres critiques découlent de ce reproche fondamental. Ainsi dans la guerre terminologique « free software » contre « open source », Stallman et la F.S.F. considèrent que le terme « open source » au lieu de souligner que les logiciels libres sont avant tout des logiciels que l'on peut utiliser, copier, modifier et distribuer librement, met l'emphase sur la disponibilité du code source. Or pour eux, la question du code source est une question accessoire à la problématique plus fondamentale des libertés concédées aux utilisateurs de logiciels libres. De la même manière, ils reprochent au mouvement open source d'être plus conciliant lorsque quelqu'un souhaite

¹⁷ A noter que de manière quelque peu paradoxale, les mainteneurs de la distribution Debian se sont depuis rapprochés de la F.S.F., et la distribution est considérée comme étant celle qui respecte le mieux les principes de la F.S.F.

mélanger logiciels libres et logiciels propriétaires, là où la F.S.F. tend à s'opposer à toute combinaison de ce type.

Au fond, les conséquences pratiques de la divergence entre les mouvements « free software » et « open source » sont minimales. Il est en tout cas bien entendu qu'il n'y a pas de différence fondamentale entre un logiciel libre et un logiciel open source. Il s'agit avant tout de désaccords « stratégiques » : les deux mouvements souhaitent que les logiciels libres se développent, mais les moyens qu'ils souhaitent mettre en œuvre à cette fin sont différents. Dans son discours, la tendance « free software » met en avant les enjeux éthiques, là où la tendance « open source » souligne avant tout les avantages économiques et techniques des logiciels libres. On pourrait en quelque sorte qualifier la première tendance d'« idéologique » et la seconde tendance de « réaliste ». Pour certains, ces divisions sont contre-productives dans la mesure où elles nuisent à la cohérence du discours de la communauté libre, alors que pour d'autres elles sont un symptôme de démocratie et de transparence. Ce qui est certain, c'est qu'elles peuvent entraîner une certaine confusion pour le profane, qui se demandera souvent ce qui sépare un logiciel libre d'un logiciel open source. C'est pourquoi il nous paraît important de retracer ces aspects historiques et terminologiques. Par la suite, nous utiliserons les termes « logiciels libres » et « open source » de manière indifférente, sauf indication contraire.

Section 4 : La « culture » libre et ses dérivés

10. Le mot est ressassé, il existe toute une « communauté » autour des logiciels libres. Le terme désigne non seulement tous les développeurs de logiciels libres, mais également leurs utilisateurs. Ceux-ci sont en effet fréquemment très actifs, qu'ils agissent dans les grandes associations que sont la F.S.F., l'O.S.I., l'Association pour la Promotion et Recherche en Informatique Libre (A.P.R.I.L.), l'Association Francophone des Utilisateurs de Linux et des logiciels libres (A.F.U.L.), ou l'Association Electronique Libre (A.E.L.), ou qu'ils s'impliquent dans les Linux User Groups (L.U.G.). Ces derniers ont pour but de partager les expériences des utilisateurs de Linux, d'initier les débutants, etc. Malgré les occasionnelles chamailleries entre tendance « free software » et tendance « open source », l'ambiance est généralement détendue et bon enfant, et la communauté est fort soudée.

11. Les idées au départ du logiciel libre font désormais des émules dans d'autres domaines. L'exemple le plus frappant nous semble être wikipédia, l'encyclopédie collaborative en ligne. Il s'agit d'un site à vocation encyclopédique où tout un chacun peut modifier un article encyclopédique ou en soumettre un nouveau. Les résultats atteints sont impressionnants tant en termes

de qualité que de quantité, puisque la version en anglais comprend plus de trois-cent neuf mille articles, et la version française près de quarante-cinq mille. A titre de comparaison, l'encyclopédie Britannica tourne autour de quarante-mille articles. L'art aussi s'inspire du logiciel libre, avec les licences Art Libre et Creative Commons. Bien que ces différentes manifestations de « l'esprit du libre » ne seront pas examinés dans le cadre du présent ouvrage, il est évident qu'il y a une parenté entre elles et les logiciels libres.

CHAPITRE 2 : ECONOMIE DU LIBRE

Section 1: Free as in « freedom », not as in « free » beer

12. Cette célèbre comparaison, que nous devons à Richard Stallman en personne, semble constituer un parfait point de départ pour cette brève analyse de l'économie des logiciels libres. On associe en effet trop souvent, et à tort, gratuité et logiciels libres. Pourtant le terme « Free » dans l'expression « free software » signifie bien « libre » et non « gratuit »¹⁸.

Cet amalgame classique provient en premier lieu d'une confusion entre exemplaire et licence. Alors que le premier peut être vendu en échange d'un prix, la seconde sera concédée gratuitement.

Il est clairement stipulé à l'article 1 de la G.P.L., par exemple, que le distributeur peut se faire rémunérer pour l'« acte physique de transférer une copie ». Stallman lui-même se faisait rémunérer près de 150 \$ par bande distribuée contenant les premiers « packages GNU ». Actuellement, il est courant de vendre des produits comprenant une distribution sur plusieurs CD-Rom ainsi que des manuels d'installation et d'utilisation. Il est également envisageable de faire payer les utilisateurs pour le téléchargement de copies.

Cependant, une fois en possession d'une copie, l'acheteur peut, en vertu de la licence libre et gratuite, utiliser, reproduire et redistribuer le programme autant de fois qu'il le désire sans être obligé de s'acquitter de « droits d'auteur ». Vu cette particularité des logiciels libres, il est clair que les distributeurs de logiciels libres ont intérêt à ce que leur packages payants présentent une certaine valeur ajoutée (manuels, conseils d'utilisation, éventuelles garanties, services après vente...) et un prix compétitif. En pratique, la vente de copies de programmes ne

¹⁸ « To understand the concept, you should think of « free » as in « free speech » not as in « free beer » », <http://www.fsf.org/philosophy/free-sw.html>.

représente qu'un très faible pourcentage des activités rentables relatives aux logiciels libres.

13. Si ce système de licences ultra-permissives et gratuites n'est pas très porteur en matière de vente de produits, il permet par contre une totale liberté sur le plan de la commercialisation de services¹⁹. Un deuxième amalgame doit en effet être évité : celui que l'on fait entre « libre » et « non-commercial ».

Si l'idée n'est pas de gagner de l'argent par le biais de licences d'utilisation, les informaticiens gardent l'opportunité de vendre leurs services ayant trait aux logiciels libres (installation, personnalisation, maintenance...). En effet, le monde du logiciel libre est avant tout constitué d'informaticiens. Lorsque l'on parle de libre copie, de libre distribution et d'accès aux sources, il s'agit souvent de parties de programmes disséminées à travers l'Internet et inutilisables par le commun des mortels. En outre, tout comme en matière de logiciels propriétaires, si certains logiciels ou certaines distributions sont conçus afin de cibler directement les utilisateurs finaux, développer un « environnement » dépassant le stade du simple ordinateur familial uni-poste requiert généralement l'intervention de personnes qualifiées. De plus, une fois l'environnement opérationnel établi, la maintenance ainsi que les mises à jour devront également être effectuées par des professionnels.

Même si la liberté de faire commerce avec les logiciels libres n'est pas reprise en tant que telle dans l'énumération de la F.S.F., elle semble constituer la clé de voûte du business-plan libre.

Section 2 : Une industrie de services

14. En faisant prévaloir la vente de services sur la vente de produits, le monde du libre ne fait finalement que respecter la nature du marché de l'informatique. Dans le marché des logiciels propriétaires, le pourcentage d'informaticiens employés pour développer des logiciels destinés à la grande distribution est dérisoire²⁰. La grande majorité d'entre eux a en effet pour tâche de conseiller leurs clients ou employeurs, leur proposer des solutions adaptées, constituer des environnements informatiques correspondant à leurs attentes et, enfin, entretenir et développer ces environnements.

¹⁹ Nombre de ces services consisteront en des modifications substantielles apportées aux programmes ou en leur personnalisation, actes pour lesquels l'autorisation de l'auteur sous forme de licence est nécessaire. Voy. *infra*, n° 94 et s.

²⁰ E. S. RAYMOND, *The Magic Cauldron*, juin 1999, <http://www.catb.org/~esr/writings/magic-cauldron/magic-cauldron.html>.

15. Ces services constituent également la base de l'économie du libre. Bon nombre d'entreprises de distribution proposent à leurs clients des produits standards sous forme de produits comprenant CDs, manuels et contrat d'assistance par « helplines », rendant par ce fait les logiciels libres accessibles aux usagers finaux. Par ailleurs, des entreprises spécialisées en services logiciels libres se développent de plus en plus, proposant la création d'infrastructures informatiques sur mesure, l'installation, la maintenance et la sécurisation de réseaux, le développement de nouvelles applications spécifiques à certains secteurs. Ces services ciblent bien évidemment les petites, moyennes ou grandes entreprises, mais se destinent également de plus en plus aux gouvernements.

16. Mais les activités d'entreprise ne s'arrêtent pas là. Les entreprises les plus reconnues, telles que Red Hat, SuSE ou Mandrake par exemple, fournissent des services d'enseignement, qui s'accompagnent généralement de remises de diplômes ou « certificats ». Outre le fait que ce système est en soi une source de revenus, il permet de souligner qu'une connaissance spécifique en matière de logiciels libres constitue un atout supplémentaire sur le marché. Qui plus est, la certification s'étend au-delà de la qualification du personnel, les entreprises se lançant maintenant dans la certification du hardware et de sa performance lorsqu'il tourne sous environnement libre.

17. Par ailleurs, si les adeptes du libre ne font pas de profit en se reposant sur leurs droits d'auteur, et s'ils bannissent la brevetabilité des logiciels, la marque est certainement leur droit intellectuel de prédilection²¹. La renommée et la reconnaissance étant sans doute des éléments clés du succès dans le marché du libre, les entreprises les plus prospères seront celles qui réussiront à associer ces qualités à leur marque. Outre sa valeur intrinsèque et l'accroissement de capital qu'elle représente, une marque forte constituera un instrument commercial majeur, ouvrant la porte à l'octroi de licences, aux franchises et aux accords de distribution qui vont généralement de pair.

Section 3 : L'incitation au développement de logiciels libres

18. Dans une industrie entièrement basée sur le service, on se demande bien évidemment quel est l'intérêt des acteurs de faire autre chose que fournir des services et notamment de développer les logiciels libres auxquels ils ont trait.

²¹ Certaines licences libres réservent explicitement l'usage de marques à leurs titulaires. Voir par exemple l'article 6 de la licence Apache version 2.0. Voir également le site du projet Mozilla (<http://www.mozilla.org/foundation/licensing.html>), qui insiste sur le respect de ses marques.

Une première réponse s'appuie sur des raisonnements qui échappent à toute logique économique. Bénévolat, plaisir de la programmation, désir de se rendre utile ou de se faire accepter dans le milieu... bon nombre de motifs sociaux ou moraux sont avancés pour expliquer l'engouement des programmeurs. Malgré la noblesse de ces causes, elles sortent de l'objet de la présente section. D'autre part, il faut souligner que beaucoup d'entreprises, insensibles à ce genre d'incitations, s'adonnent cependant – et paient pour ce faire des informaticiens – au développement et à la création de nouvelles applications.

19. Les programmeurs peuvent trouver un intérêt économique dans la participation à des projets libres à plusieurs égards. Sur le marché de l'emploi, le fait d'avoir pris part au développement de logiciels libres constitue certainement un avantage. En effet, cette expérience est généralement appréciée non seulement pour les connaissances informatiques acquises, mais également dans la mesure où elle reflète des capacités spécifiques de collaboration et de travail de groupe. La participation à un projet libre constitue dès lors un moyen de différenciation entre développeurs.

20. Le développement de logiciels libres est également profitable aux entreprises, qui établissent de véritables « stratégies open source ». Certaines d'entre elles ouvrent et libèrent leurs codes afin de profiter des améliorations et corrections apportées par le mouvement libre, tout en continuant la commercialisation du produit grâce à leur expertise unique du produit en question²². Des entreprises de hardware ont parfois tout avantage à promouvoir le développement de logiciels libres qui fonctionnent sur leurs machines, ce qui augmentera la valeur de ces dernières, d'autant qu'elles pourront être livrées avec des logiciels gratuits²³.

Certaines entreprises, comme MySQL par exemple, jouent sur les deux tableaux, distribuant leurs logiciels sous licence copyleftée d'une part, et sous licence payante pour les développeurs qui désireraient incorporer leur programme dans un ensemble distribué selon le modèle propriétaire, d'autre part. Alors que la première distribution profite au mouvement libre tout en ayant un effet publicitaire, la seconde permet de rentabiliser l'investissement.

Enfin, d'une façon générale, les sociétés de services ont également tout intérêt à initier de nouveaux produits, soit parce qu'ils porteront leur marque, soit parce qu'ils constitueront l'empreinte reconnaissable de leur expertise, ou encore, parce qu'ils généreront de plus amples possibilités de services à pourvoir.

²² J. CREMER et A. GAUDEUL, « *L'économie du logiciel libre* », 10 septembre 2003, <http://www.jm.u-psud.fr/~adis/rubriques/p/carrytic/cremer2.pdf>.

²³ IBM ou SUN, par exemple, ont très bien assimilé cette stratégie.

CHAPITRE 3 : QUELQUES LICENCES

21. Tout au long de ce cahier, les auteurs seront amenés à citer et à analyser dans leurs contributions certaines clauses contractuelles extraites de différentes licences libres. Aussi avons-nous cru utile de présenter dans cette section quelques licences, sélectionnées en fonction de la fréquence de leur utilisation, leur qualité représentative ou au contraire leurs spécificités atypiques. Le but n'est évidemment pas d'analyser dans le détail chacune de ces licences, mais plutôt de donner un aperçu de leurs principales caractéristiques, ainsi que de fournir des informations générales quant à leurs auteurs et à l'esprit dans lequel elles virent le jour.

GNU General Public License (G.P.L.) version 2.0, juin 1991

22. A tout seigneur tout honneur, commençons par la licence la plus connue, la plus utilisée, la plus controversée et présentée comme étant l'une des plus virales (*infra*, n° 135 et s.). La licence GNU General Public License²⁴ est la licence créée, utilisée et promue par la Free Software Foundation.

La G.P.L. fut rédigée par Richard Stallman ; elle consiste en fait en une mise en oeuvre des droits d'auteur qui correspond à la philosophie et à l'idéal de ce dernier. Tout en laissant à tous l'opportunité d'accéder au code source des programmes, ainsi que la possibilité d'exécuter, d'utiliser, d'adapter, de transformer et de redistribuer ces derniers, la G.P.L. empêche quiconque de restreindre ces libertés sur le programme. Dès lors, non seulement la G.P.L. est une des premières licences libres, mais elle est également la première application du système « *copyleft* ». Linux est le logiciel le plus connu distribué sous G.P.L.

La G.P.L. commence par un long préambule qui, s'il relève surtout du discours militant, a cependant le mérite de décrire l'esprit dans lequel les termes de la licence devront être interprétés.

S'en suivent les clauses classiques en matière de logiciel libre. La G.P.L. octroie au licencié l'autorisation d'utiliser, copier, modifier et distribuer le programme à condition de respecter les notices de copyright, de non-garantie et d'exonération de responsabilité. La G.P.L. prévoit aussi l'obligation du licencié de fournir, en cas de redistribution, le code source ou, à tout le moins, d'en garantir l'accès. Copyleft oblige, elle impose également au licencié de redistribuer le programme, modifié ou non, sous la même licence.

²⁴ Disponible sur le site de la F.S.F. à l'adresse <http://www.gnu.org/licenses/gpl.html>.

Enfin, on y trouve par la suite des clauses relatives aux sanctions en cas de non respect de la licence, à la formation du contrat, à son interprétation et son application. Conformément à la tradition informatique, la licence se termine par une clause de non-garantie et d'exonération de responsabilité.

GNU Lesser General Public License (L.G.P.L.) version 2.1, février 1999

23. La L.G.P.L.²⁵ est une licence bien plus récente que la G.P.L. et qui fut rédigée par Stallman et Eben Moglen. Cette nouvelle licence cible plus particulièrement les créateurs de librairies informatiques et constitue une alternative copyleft moins « virale » que la G.P.L.

En vertu d'une interprétation très extensive de la G.P.L., la licence imposerait que tout programme qui serait lié dynamiquement à une librairie informatique distribuée sous cette licence devrait également être soumis aux termes de la G.P.L., ce dernier devant également être considéré comme une « œuvre basée » sur cette librairie selon les termes de l'article 2 de la G.P.L. (*infra*, n° 136 et s.). Vu que l'effet copyleft ainsi décrit s'effectue par simple « linkage », il a été comparé à une « contamination virale » ou, en d'autres termes, à une propagation par simple contact.

La L.G.P.L. est rédigée de manière à exclure cet effet spécial de copyleft et de permettre, ainsi, que des programmes puissent être dynamiquement liés avec une librairie sous L.G.P.L. tout en laissant aux auteurs de ces programmes, à certaines conditions, la possibilité de distribuer leur œuvre sous n'importe quelle licence.

La raison avancée pour la création de cette G.P.L. « atrophiée » diffère selon les camps et la façon dont le copyleft s'appréhende. Le préambule de la L.G.P.L., qui reflète la position de la F.S.F., avance que la distribution sous cette licence encourage la propagation de l'utilisation de certaines librairies, ce qui en fait des standards *de facto*. La L.G.P.L. servirait dès lors à promouvoir davantage le libre et à essayer d'atteindre la standardisation. Les dénonciateurs de l'effet viral de la G.P.L. présentent cela sous un tout autre angle, avançant que cette contamination dissuaderait beaucoup de programmeurs de créer des logiciels liés à des librairies sous G.P.L. Dès lors, la F.S.F. se serait rendue compte elle-même de l'effet négatif que pouvait avoir un copyleft extrême.

²⁵ Disponible sur le site de la F.S.F. à l'adresse <http://www.gnu.org/licenses/gpl.html>.

La L.G.P.L. se présente globalement comme une G.P.L. qui compterait parmi ses clauses une importante exception de « linkage » (article 6 de la L.G.P.L.).

Berkeley Software Distribution license (B.S.D. license)

24. L'histoire de la Berkeley Software Distribution remonte aux années 70, à l'ère du développement des premiers véritables systèmes d'exploitation. Déjà quinze ans avant les premiers pas de Gnu/linux, les étudiants de l'université de Berkeley développaient de leur côté un système d'exploitation sur base de code Unix. Actuellement, même s'il est vrai que les systèmes d'exploitation B.S.D. ne sont pas monnaie courante, les différentes évolutions du projet, telles que FreeBSD, OpenBSD ou NetBSD gardent une bonne réputation auprès des spécialistes en serveurs Intranet ou Extranet. Le programme Sendmail est un des logiciels les plus connus distribués sous licence B.S.D.

La licence B.S.D.²⁶ se démarque par sa permissivité et sa brièveté. Elle permet la redistribution et l'usage du programme, sous forme de code source ou binaire, avec ou sans modification, à condition que les notices de copyright et les termes et conditions de la licence soient conservées et apparaissent sur la distribution. Sa dernière clause prévoit que les noms des contributeurs ou de leur organisation ne pourront servir à promouvoir les produits dérivés. La licence se termine par une clause de non-garantie et d'exonération de responsabilité.

La licence B.S.D. constitue l'exemple classique de la licence non-copyleft. Du moment que les conditions citées ci-dessus sont respectées, le licencié peut très bien inclure du code du programme distribué sous B.S.D. dans un logiciel qu'il distribue sous licence propriétaire.

²⁶ Disponible à <http://opensource.org/licenses/bsd-license.php>. Les versions de cette licence ne sont pas précisées officiellement. Cependant il faut noter que le 22 juillet 1999, William Hoskins, le directeur de l' « Office of technology licensing » de Berkeley, déclara la suppression de la clause dite de « publicité » (voir <ftp://ftp.cs.berkeley.edu/pub/4bsd/README.Impt.License.Change>). La licence originale est parfois nommée la « 4-clause B.S.D. » ou la vieille B.S.D., alors que la B.S.D. revue est appelée « 3-clause B.S.D. », nouvelle B.S.D. ou B.S.D. révisée. Dans le cadre de ce cahier, c'est cette dernière version qui sera utilisée.

Q Public License (Q.P.L.) version. 2.0

25. La Q.P.L.²⁷ est la licence rédigée par Trolltech, une société qui développe, entre autres, la librairie informatique Qt. Même si cette dernière est actuellement distribuée sous G.P.L., la Q.P.L. est toujours appliquée à d'autres programmes.

Tout en étant une licence libre, la Q.P.L. se démarque par ses clauses atypiques qui, en définitive, en font une licence assez restrictive. La clause la plus inhabituelle est certainement celle de l'article 3 qui autorise la distribution de modifications apportées au programme à condition que celles-ci soient séparées du programme, c'est-à-dire fournies sous forme de patch par exemple. Si les modifications sont distribuées sous la licence Q.P.L., seul l'auteur originel se voit octroyer le droit d'intégrer ces modifications dans de futures versions, à condition que ces versions soient distribuées sous la même licence (ce qui constitue une sorte d'effet copyleft vis-à-vis de l'auteur originel) en sus de toute autre licence.

L'article 6 est également et particulièrement remarquable dans le sens où il pose des conditions au fait de « linker » un programme avec l'original ou avec la version modifiée du logiciel sous Q.P.L.²⁸.

Open Software License (O.S.L.) version 2.1

26. Rédigée par Lawrence E. Rosen, l'O.S.L.²⁹ est une licence libre et copyleft que l'on pourrait présenter comme étant la licence de l'O.S.I.

L'O.S.L. se démarque des autres licences par son langage et sa propension à l'exhaustivité. Les droits accordés reprennent les catégories légales du droit américain. Tous les droits intellectuels relatifs aux programmes sont abordés, que ce soit le droit d'auteur, les brevets ou les marques. En outre, l'O.S.L. présente toute une série de clauses classiques, généralement présentes dans tout contrat mais souvent absentes en matière de logiciels libres, telles qu'une clause

²⁷ Disponible à <http://www.trolltech.com/licenses/Q.P.L..html>.

²⁸ L'article 6.a prévoit que les sources des logiciels liés au programme sous licence doivent être disponibles. L'article 6.b prévoit que les éléments liés au programme sous licence doivent être distribués sous des termes et conditions permettant la modification et la redistribution. Trolltech, dans sa version annotée de la licence, explique clairement que cette clause a pour but de forcer les développeurs à ne distribuer leurs applications faisant appel à la librairie que sous licence open source. Voir <http://www.trolltech.com/licenses/Q.P.L.-annotated.html>. Nous aurons l'occasion de voir par la suite que l'existence de cette prérogative dans le chef des développeurs de bibliothèques ou autres éléments de linkage est loin d'être toujours garantie.

²⁹ Disponible à <http://www.opensource.org/licenses/osl-2.1.php>

de garantie d'éviction ou une clause déterminant la loi applicable et la juridiction compétente.

Enfin, l'O.S.L. contient une clause (article 10) tout à fait inédite sur la résolution immédiate du contrat pour le cas où le licencié intenterait une action contre le donneur de licence sur base de la violation d'un brevet que ce dernier aurait commise.

En définitive, l'O.S.L. est intéressante dans la mesure où elle est sans doute la plus complète d'un point de vue juridique.

Mozilla Public License (M.P.L.) version 1.1

27. Mozilla est le successeur de Netscape, et consiste en un projet spécialisé dans le développement d'outils Internet « grand public ». En bref, le projet Mozilla est avant tout connu pour ses « browser » et ses applications de messagerie électronique³⁰.

Mozilla a créé sa propre licence copyleft : la M.P.L.³¹. Cette licence est originale à plusieurs égards. Elle sépare développeur initial, à savoir l'auteur du code originaire du programme, et contributeurs. Cette distinction dévoile toute sa portée dans la mesure où la licence prévoit expressément, en son dernier article, la possibilité du développeur initial de distribuer son œuvre simultanément sous une autre licence de son choix.

La M.P.L. se distingue également par son système de copyleft faible, et non viral, qui fait la distinction entre œuvre modifiée et œuvre plus large. La M.P.L. oblige le contributeur à redistribuer le logiciel et ses modifications sous M.P.L. Par contre, si un licencié intègre du code sous M.P.L. dans un ensemble logiciel plus vaste dont certaines parties sont originales, il peut octroyer la licence de son choix sur ces parties entièrement originales (art. 3.7). La définition du terme « modification » à l'article 1.1 permet de mieux clarifier la distinction entre une œuvre modifiée et l'ensemble plus vaste. Si un logiciel se compose de plusieurs fichiers, une modification constitue tout ajout ou retrait à un fichier originaire ou l'insertion d'un fichier reprenant partie du code originaire. Ainsi, *a contrario*, si le contributeur insère au programme un nouveau fichier entièrement original, il pourra distribuer cette partie entièrement originale selon ses propres conditions.

³⁰ Ces produits concurrencent dès lors directement les produits propriétaires équivalents tels qu'Internet Explorer ou Outlook

³¹ Disponible à <http://www.mozilla.org/M.P.L./M.P.L.-1.1.html>.

Apache License, version 2.0, janvier 2004

28. L'Apache Software Foundation se définit comme une fondation fournissant de l'aide à la communauté Apache de développement de logiciels libres et visant la création de logiciels de pointe³². Ce but est certainement atteint en matière de serveurs web : d'après l'étude réalisée par Netcraft, les serveurs apaches seraient largement en tête du marché³³.

La licence Apache³⁴ est une licence libre non-copyleft que l'on pourrait comparer à une licence B.S.D. évoluée et précisée. Son article 6 est particulièrement intéressant dans la mesure où il souligne clairement l'importance de la marque en matière de logiciels libres.

CHAPITRE 4 : DERNIÈRES CLARIFICATIONS

29. Il reste encore à clarifier quelques points qui font souvent l'objet de malentendus. Ces points sont les caractéristiques des logiciels libres, leur rapport au droit d'auteur, et la nature contractuelle des licences.

Section 1 : Caractéristiques des logiciels libres

30. On attribue souvent un certain nombre de caractéristiques aux logiciels libres, que ce soit pour les encenser ou les critiquer. On entend ainsi dire qu'ils ne sont destinés qu'aux techniciens, qu'ils sont plus sûrs, moins vulnérables aux virus, ou moins sujets aux bugs. Que faut-il penser de telles affirmations ? Elles ne sont ni vraies ni fausses. Elles ont pour point commun d'être indécidables *a priori*. Un logiciel libre est un logiciel dont la licence confère à ses utilisateurs le droit de l'utiliser, de le modifier, de le copier et de le distribuer. La distinction entre logiciel libre et logiciel propriétaire est donc une pure question de régime juridique, et n'a pas, en soi, de conséquences techniques automatiques.

31. L'impression que les logiciels libres sont destinés aux techniciens provient de deux éléments. Tout d'abord, les premiers logiciels libres célèbres étaient effectivement des outils orientés vers la programmation et la gestion de réseaux, n'intéressant que les spécialistes. Toutefois, ceci ne découlait pas d'une propriété intrinsèque des logiciels libres, et dès lors que les logiciels pour

³² Voir www.apache.org.

³³ « Webserver Survey » effectué par Netcraft, disponible à l'adresse http://news.netcraft.com/archives/web_server_survey.html.

³⁴ Disponible à <http://www.apache.org/licenses/LICENSE-2.0.html>.

informaticiens furent au point, on vit apparaître des logiciels à usage généraliste : suites bureautiques (Open Office), logiciels de retouche photo (The Gimp), browsers (Mozilla), etc. Ensuite, on considère parfois que l'accès aux sources n'a d'intérêt direct que pour les informaticiens à même de les comprendre. Si cela est vrai, cela n'empêche en rien un utilisateur profane d'employer des logiciels libres. Certes, il n'exercera pas tous les droits dont il dispose, mais dans la mesure où ces droits n'entraînent pas de contrepartie significative, cela n'est pas un problème. Encore une fois, les particularités juridiques des logiciels libres ne les cantonnent certainement pas aux informaticiens. De ce fait, le seul critère pour savoir si un logiciel libre peut être utilisé par un profane est, comme pour tout logiciel, le public ciblé par ses auteurs. Dans la mesure où les applications disponibles sont de plus en plus diversifiées à ce niveau, il n'y a rien qui restreint l'utilisation de logiciels libres par des non-techniciens.

32. De la même manière, rien ne garantit *a priori* la sécurité, la résistance aux virus ou l'absence de bugs dans un logiciel libre. Certes, l'accessibilité du code source par la communauté des utilisateurs du logiciel facilite l'identification de failles de sécurité ou d'erreurs dans le code, mais ce n'est pas une condition suffisante pour garantir l'élimination de tout risque : pour que l'ouverture du code puisse avoir de tels effets positifs, il est aussi nécessaire que la communauté soit suffisamment grande et impliquée dans le feedback³⁵. Le caractère libre d'un logiciel peut améliorer sa qualité, mais cette amélioration n'est pas garantie. Tout comme dans le monde propriétaire, l'analyse qualitative doit se faire au cas par cas.

Section 2 : Le rapport entre les logiciels libres et le droit d'auteur

33. L'on entend parfois dire que les logiciels libres sont « contre » le droit d'auteur, ou « anti-droit d'auteur ». Ces formules tentent de capturer l'idée que les licences de logiciels libres récuse le droit d'auteur ou échappent à son emprise. Le public est parfois renforcé dans cette impression par l'utilisation du terme « copyleft », antithèse supposée du « copyright ». La réalité est toute autre. Loin de nier le droit d'auteur, les licences de logiciels libres s'appuient sur lui pour produire leurs effets. En effet, c'est parce que l'auteur d'un logiciel peut déterminer ce que les tiers peuvent faire avec son œuvre qu'il est à même d'imposer certaines conditions et restrictions à l'usage du logiciel. Ainsi, sans droit d'auteur, il serait impossible d'imposer l'obligation de redistribution sous licence libre qui caractérise le copyleft ou de réclamer la publication des sources

³⁵ Voy. E. S. RAYMOND, *The cathedral and the bazaar*, <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>.

lors de la rediffusion. Dans un système sans droit d'auteur, rien de tout cela ne serait possible et n'importe qui pourrait redistribuer commercialement un logiciel en se faisant passer pour son auteur ou sans en communiquer les sources, choses que le mouvement libre réproouve plus que tout. Les licences de logiciels libres, loin de s'opposer au droit d'auteur, ont donc besoin de lui et sont bien plus une manifestation de sa flexibilité que de son obsolescence³⁶.

Section 3 : La nature des licences de logiciels libres

34. On entend parfois défendre l'idée que les licences de logiciels libres ne sont pas des contrats. L'origine de cette thèse est confuse. Elle semble se nourrir de certaines considérations émises par le professeur Eben Moglen³⁷, avocat de la F.S.F., à propos de la force obligatoire de la G.P.L., et d'une volonté de faire échapper ces licences au droit des contrats, afin d'en assurer la validité. Toutefois, les raisonnements du professeur Moglen se font en droit américain et semblent inacceptables en droit belge³⁸, et il ne suffit pas d'avoir un intérêt pratique à une interprétation juridique pour justifier cette dernière. Encore faut-il qu'elle soit rigoureuse et correcte.

35. Le problème fondamental avec la thèse de la licence qui ne serait pas un contrat est qu'on ne voit pas ce que la licence pourrait être d'autre. Si l'on considère qu'une licence libre a des effets juridiques concrets en tant que licence de logiciel, il faut admettre qu'elle fait naître des obligations. Or les sources d'obligations sont en nombre limité : les délits et quasi-délits, les contrats et quasi-contrats, l'engagement par voie unilatérale, la théorie de l'apparence, la loi³⁹. L'on ne voit pas laquelle de ces sources pourrait fonder les obligations des parties si ce n'est le contrat. Il ne s'agit en aucun cas d'un délit ou quasi-délit, d'un quasi-contrat, de l'apparence ou de la loi.

La formulation, que l'on entend parfois, selon laquelle les licences de logiciels libres seraient des autorisations générales d'utiliser un logiciel à toutes fins

³⁶ Ceci est d'ailleurs reconnu par le préambule de la G.P.L. qui précise que la protection des droits de l'utilisateur se fait en deux étapes, l'obtention d'un droit d'auteur et la concession d'une licence accordant certains droits.

³⁷ *Free software matters : Enforcing the G.P.L.,I*, disponible à l'adresse <http://moglen.law.columbia.edu/publications/lu-12.html>.

³⁸ Il n'est d'ailleurs pas certain qu'ils soient valables en droit américain.

³⁹ Certains auteurs considèrent que les catégories des délits, quasi-délits et quasi-contrats peuvent être subsumées sous la catégorie d'obligations nées du fait de la loi et ne distinguent donc plus que les obligations nées du fait de la loi ou du contrat. Ce raisonnement paraît difficilement tenable dans la mesure où le contrat n'a lui aussi de force obligatoire dans un système juridique donné que du fait de la loi, et donc qu'on devrait alors dire que la seule source d'obligations est la loi.

pourrait laisser à croire qu'en fait elles se fondent sur un engagement par voie unilatérale de la part de l'auteur. Toutefois, l'hypothèse ne tient pas. Tout d'abord, elle impliquerait que l'auteur puisse retirer son autorisation pour le futur comme il l'entend et ainsi priver les utilisateurs de tout droit sur le logiciel, puisque ce que la volonté seule peut créer, la volonté seule doit pouvoir le défaire⁴⁰. Or une telle situation créerait une insécurité juridique intenable pour l'utilisateur. Ensuite, l'hypothèse de l'engagement par voie unilatérale ne permet pas d'expliquer que dans toute licence libre, même la moins contraignante pour l'utilisateur, il existe des obligations à charge de l'utilisateur (*infra*, n° 214-218). Or un engagement par voie unilatérale est évidemment inapte à imposer des obligations à des tiers. Pour que ceux-ci soient tenus d'obligations, il est indispensable qu'il y ait un contrat. La figure du contrat est la seule à même d'expliquer que tant le donneur de licence que le licencié sont tenus d'obligations. Elle est donc incontournable⁴¹.

⁴⁰ Certes, il y a des conditions à respecter pour se dégager d'un engagement pris par voie unilatérale. Il n'empêche qu'il est toujours possible de s'en dégager, et ce, sans avoir besoin de l'accord de quiconque. Voy. M. COIPEL, « La théorie de l'engagement par voie unilatérale et son intérêt particulier en droit des sociétés », in *Mélanges offerts à Pierre Van Ommeslaghe*, Bruxelles, Bruylant, 2000, p. 38.

⁴¹ En ce sens, J.-B. SOUFRON, *La G.P.L. est-elle un contrat en droit français ?*, http://soufron.free.fr/soufron-spip/article.php3?id_article=35.