

Table des matières

Les pratiques contractuelles propres à l'utilisation des licences open source.....	3
Comprendre les licences open source.....	3
Grille de lecture d'une licence open source.....	3
Classification des licences open source.....	4
Choisir une licence open source.....	5
Le choix d'une licence adaptée aux besoins spécifiques de la société et du produit.....	5
Le choix stratégique de la licence.....	5
Les ajustements tactiques relatifs au choix de la licence.....	6
L'usage tactique des interprétations et des exceptions.....	6
L'utilisation tactique des Multilicences.....	6
L'influence de la compatibilité dans le choix des licences.....	7
Gérer la compatibilité entre les licences.....	7
Les solutions alternatives en cas d'incompatibilité.....	8
Les pratiques encadrant l'utilisation des licences open source.....	8
Des intérêts divergents relatifs à l'activité de la société.....	8
L'édition open source.....	8
Le service et l'assistance autour de Logiciels Libres.....	10
La place du service juridique dans les pratiques liées à l'utilisation des licences open source.....	10
Le travail au contact des équipes techniques.....	11
Le travail complémentaire lié à l'usage des licences open source.....	11

Les pratiques contractuelles des industries de l'open source

« Hors normes », le Logiciel Libre fut pendant longtemps assimilé à un phénomène curieux, relevant à la fois de l'amateurisme, du mutualisme et de la contestation. Pourtant, la reconnaissance dont il jouit aujourd'hui auprès de l'industrie du logiciel se traduit par un renversement complet de paradigme, abandonnant la culture du monopole au profit de l'*open source* et de l'Open Innovation¹ — les entreprises s'étant dès à présent plongées dans cette pratique bénéficient d'une croissance sans équivalent et leur compréhension du modèle leur assure une avance considérable pour les années à venir².

Fruit d'une démarche empirique, l'ambition de cet article est de livrer quelques clés utiles à l'appréhension des conséquences juridiques majeures qu'emporte un tel bouleversement du modèle de pensée au sein d'une entreprise. Les nouvelles pratiques contractuelles tournent autour de trois objectifs complémentaires : tirer le meilleur parti de l'usage des licences *open source*, organiser une synergie avec les autres droits exclusifs et optimiser le système contractuel afférent. Ce dernier point est particulièrement important puisque, exerçant son « droit de se gouverner soi-même », la société use massivement des contrats (dont la licence n'est qu'une partie) pour former le socle du système qui lui sied (on parle de *business model* ou de méthode commerciale). On songe à une propriété intellectuelle « repensée » tant celle-ci est remodelée – les licences *open source* qui sur lesquelles celle-ci est basée ont notamment l'avantage considérable de s'être adaptées aux évolutions techniques et sociétales afférentes à la création. Par exemple, les œuvres multimédias – pour lesquelles se pose habituellement la question du régime³ – s'intègrent sans difficulté à l'univers des licences *open source* où la multiplicité d'auteurs est la règle.

Pour des raisons diverses est variées éloignées de l'objet de cet article, la sémantique revêt une importance toute particulière dans l'écosystème du logiciel libre⁴. Nous nous efforcerons d'utiliser les significations que nous estimons premières des notions : un logiciel étant *libre* dès lors qu'il offre à l'utilisateur l'ensemble des libertés définies par la FSF⁵ et une licence étant *open source* dès lors qu'elle répond aux dix critères de l'*Open Source Definition*⁶.

Ainsi, les licences libres ne sont finalement que de simples contrats⁷ de cessions non exclusives de droits d'auteur auxquels sont associées quelques préoccupations d'ordre de la propriété matérielle⁸. Loin de nier le droit d'auteur, les auteurs de logiciels libres s'en inspirent pour exploiter d'une façon tout autre leur création, comportement parfaitement compris et exprimé par les rédacteurs de la Licence Art Libre⁹ : « [s]i, en règle générale, l'application du droit d'auteur conduit à restreindre l'accès aux œuvres de l'esprit, la Licence Art Libre, au contraire, le favorise. L'intention est d'autoriser l'utilisation des ressources d'une œuvre ; créer de nouvelles conditions de création pour amplifier les possibilités de création. La Licence Art Libre permet d'avoir jouissance des œuvres tout en reconnaissant les droits et les responsabilités de chacun ». Mieux encore, il est possible d'affirmer que ces licences et les pratiques liées, par leur attachement au droit d'auteur, favorisent une reconnaissance étendue de la qualité d'auteur : tout participant à la création sera soumis à la

1 D'après un terme d'Henry William Chesbrough, voir notamment « Open Innovation: The New Imperative for Creating and Profiting from Technology », Harvard Business School Press, 2003. Il faut noter que les notions d'*open source* et d'*open innovation* sont dorénavant synonymes aux États-Unis.

2 Le cabinet d'étude Gartner allant jusqu'à une affirmation plus radicale : « open source impossible to avoid » (Network World, Septembre 2007). Selon ce même cabinet, 80% des logiciels commerciaux contiendront des briques *open source* en 2011 !

3 Voir à ce sujet le rapport du CSPLA : « Le régime juridique des œuvres multimédia : Droits des auteurs et sécurité juridique des investisseurs », rapport de la Commission sur les aspects juridiques des œuvres multimédias rendu le 26 mai 2005, Valérie-Laure Benabou ; Jean Martin ; Olivier Henrard.

4 L'usage qui épargne toutes les susceptibilités est habituellement de parler de FOSS ou FLOSS (Free Libre open source Software).

5 À savoir : la liberté d'exécuter / utiliser le programme ; la liberté d'étudier et d'adapter le programme ; la liberté de redistribuer des copies ; et la liberté de modifier et de publier ses modifications. Ainsi, un logiciel qui serait dans le domaine et dont les sources seraient accessibles pourrait être valablement qualifié de logiciel libre.

6 Une définition fortement inspirée de la *Debian Free Software Guidelines* et rédigée par Bruce Perens. Les licences sont certifiées *open source* par l'*open source Initiative* au regard de cette définition (il y en a 70 licences certifiées *open source* à ce jour).

7 Certains auteurs américains prétendent le contraire : Jones (P), « The GPL Is a License, not a Contract », Linux Weekly News, 3 décembre 2003. ; a contrario Rosen (Lawrence E.), « Open Source Licensing : Software Freedom and Intellectual Property Law », Upper Saddle River, N.J. : Prentice Hall PTR (2004), pp. 59-66.

8 C'est ici l'aspect « contrat de mise à disposition » (de logiciel, musique, etc.) de la licence. Il se traduit notamment par l'inclusion systématique des clauses de limitation ou exclusion de garantie et de responsabilité.

9 Issue du collectif Copyleft Attitude, la « LAL » fut le fruit d'une rencontre entre juriste, informaticiens du libre et artiste. La première version date de 2000.

licence et assujettira les autres participants de la même façon.

Plusieurs constats peuvent déjà être réalisés : la licence ne s'applique qu'à l'égard des utilisateurs finaux du logiciel ou à toute personne qui sans cette dernière ne pourrait l'utiliser comme elle le souhaite ; le logiciel ne sera libre qu'autant que les libertés sont de surcroît matériellement octroyées¹⁰ ; la non-exclusivité qui régit les licences *open source* permet de cumuler des exploitations différentes sur les mêmes créations¹¹. Ces éléments nous amènent à détailler les pratiques contractuelles propres à l'utilisation des licences *open source* (I) et celles qui encadrent cette utilisation (II).

I. Les pratiques contractuelles propres à l'utilisation des licences *open source*

L'utilisation des licences reste simple et classique, mais nécessite un travail préliminaire essentiel pour comprendre, intégrer et maîtriser toutes les subtilités afférentes. Seul ce travail d'analyse et de veille permettra dans un second temps de choisir la licence *open source* idéale avec une visibilité suffisante pour assurer la pérennité indispensable au développement du logiciel.

A. COMPRENDRE LES LICENCES OPEN SOURCE

Dans les faits — et malgré leur nombre —, il n'y a que très peu de différences entre chaque licence. Une proposition de grille de lecture et quelques tentatives de classifications permettront de bien saisir leur mécanisme.

1. Grille de lecture d'une licence *open source*

Trois éléments caractérisent et distinguent une licence libre d'une autre : ses droits et obligations, son étendue¹² et son élément déclencheur.

Les **droits** et **obligations** sont autant ce qui distingue le plus les licences les unes des autres — selon leurs engagements vis-à-vis des brevets, signes distinctifs, DRM, etc. — que ce qui les réunit — le minimum de liberté étant fixé par leur qualité de libre et/ou d'*open source*. Les premiers sont usuellement des cessions non exclusives de droits patrimoniaux rédigées conformément au formalisme habituel de l'article L. 131-3 du CPI¹³. Les obligations¹⁴ se posent comme des conditions (résolutoires) dont le non-respect emportant généralement la terminaison automatique, immédiate ou différée, de la licence.

L'**étendue** de la licence permet de déterminer dans quelle mesure certaines créations peuvent lui être rattachées (à l'instar des œuvres dérivées ou composites du régime légal) et, dès lors, subir les contraintes de la licence : la GNU *General Public License*¹⁵ s'étend au logiciel comme un tout, à l'exception des briques logicielles indépendantes et séparées (sauf si elles-mêmes distribuées comme un tout avec le reste du logiciel) ; la GNU *Lesser General Public License*¹⁶ allège cette dernière en autorisant sans contrainte les

10 Ce qui n'est par exemple pas le cas si seul le binaire du logiciel est distribué et ne le sera que très partiellement si le code source est illisible, non documenté, voire au final inexploitable — remarque d'autant plus importante que beaucoup de sociétés jouent de cette confusion et font du libre « par démagogie ».

11 L'atteinte au droit de divulgation étant ainsi limitée : l'auteur doit certes divulguer au moins sous la licence indiquée, mais il peut aussi choisir d'autre destin à son œuvre et convenir d'une exploitation alternative.

12 On peut aussi parler de **portée**, pour mettre en évidence le côté intrusif de certaines licences.

13 « La transmission des droits de l'auteur est subordonnée à la condition que chacun des droits cédés fasse l'objet d'une mention distincte dans l'acte de cession et que le domaine d'exploitation des droits cédés soit délimité quant à son étendue et à sa destination, quant au lieu et quant à la durée » — chacune de ces délimitations se faisant de façon très large.

14 Ainsi, peut-on énumérer de façon non limitative : l'obligation de distribuer les œuvres dérivées sous la même licence, l'interdiction d'utiliser ses brevets logiciels pour reprendre le contrôle des œuvres, les habituelles clauses d'exclusion ou limitation de garanties et de responsabilité, les clauses sur les brevets, sur les signes distinctifs, etc.

15 Principale licence libre, dont la première version a été publiée en 1989 par la *Free Software Foundation*. La troisième et dernière version, corédigée par la *Free Software Foundation* et le *Software Freedom Law Center*, date du 29 juin 2007.

16 Petite sœur de la première, la GNU LGPL est une « version amoindrie » de la GNU GPL. La première version date de 1991 et la troisième version a été publiée le 29 juin 2007.

simples utilisations du logiciel ; enfin d'autres licences comme la CeCILL¹⁷ ou la MPL¹⁸ s'étendent aux seuls fichiers contenant du code sous cette licence — ce dernier critère étant incontestablement le moins contraignant pour les développeurs.

Enfin, par **élément déclencheur**, il faut comprendre l'acte qui emporte les effets contraignants de la licence et avant lequel le licencié est libre de ses actes. Ce dernier élément est extrêmement important puisque nombre de sociétés ont adapté leur *business model* à ce critère afin de disposer des outils libres sans reverser leur propre technologie. Par exemple, celui qui modifie un logiciel sous GNU GPL peut le faire sans restriction jusqu'à ce qu'il décide de *distribuer* le logiciel¹⁹, inversement, celui qui utilise un logiciel sous *Reciprocal Public License*²⁰ ou *Open Software License*²¹ (et dernièrement la GNU Affero GPL²²) devra respecter toutes les obligations de la licence dès lors que le logiciel peut être *utilisé* ou *interagit* avec des tiers, directement ou indirectement.

Enfin, une attention particulière doit être portée aux définitions : en raison de leur utilisation internationale²³, les licences cherchent fréquemment à substituer des définitions contractuelles aux termes juridiques habituellement utilisés dans tel ou tel système juridique.

2. Classification des licences open source

Il est intéressant d'opérer un recoupement entre classifications afin de mieux visualiser les distinctions et filiations qui unissent ou séparent les licences. Quatre classifications seront présentées brièvement : *copyleft*/permissif, historique, par domaine et par liberté.

Grâce au développement qui précède, on distingue commodément les licences permissives des licences *copyleft* : les premières (permissives) sont des licences dont seules les obligations sont transmises à toutes les personnes détentrices subséquentes d'une copie de l'œuvre ; les secondes (*copyleft*) sont des licences dont tant les obligations que les droits doivent se retrouver dans toute redistribution ultérieure d'une copie de l'œuvre. L'emploi de licences *copyleft*²⁴ engendre un cercle vertueux par la confiance qu'elles instaurent entre les différents acteurs et contributeurs. Ceci par deux mécanismes différents : la licence impose la redistribution sous une licence particulière²⁵ ; la licence énumère expressément les droits qui doivent être transmis au surplus des obligations – dans ce dernier cas, la redistribution peut alors se faire sous cette même licence ou une licence similaire.

Il est aussi possible d'envisager un découpage historique entre les différentes phases : les licences philosophiques (dimension associées aux licences GNU de la FSF) composent la première vague de licences, suivies des licences académiques (licence ultra permissive comme la BSD ou la MIT), des licences communautaires (souvent très particulières et attachées aux besoins d'un projet communautaire) et enfin les licences institutionnelles (introduites par les premières sociétés à s'intéresser au développement *open source*). Leur rédaction s'est perfectionnée au fil des ans, leur permettant de s'adapter de plus en plus aux systèmes commerciaux mis en place;

17 La licence CeCILL est l'acronyme pour Cea Cnrs Inria Logiciel Libre. Elle fut créée en 2004 par des chercheurs français, afin de faciliter leurs travaux de recherches et de permettre leur capitalisation dans un esprit conforme au Libre. La deuxième et dernière version date de mai 2005.

18 La Mozilla Public License (MPL) est la licence *open source* créée à l'occasion de la libération du code source de Netscape Communicator 5 par Netscape, en mars 1998. L'auteur de cette licence est Mitchell Baker, à l'époque avocate de Netscape Communications Corporation, et aujourd'hui présidente de la Mozilla Corporation.

19 En conséquence de quoi une simple utilisation d'un logiciel en tant que service (SaaS) n'est pas une distribution selon la GNU GPL puisqu'il n'y a pas de transfert de copie.

20 La RPL fut rédigée en 2001 par Scott Shattuck afin de combler la non-effectivité de la GNU GPL pour les SaaS. La version actuelle est la 1.5 et a été publiée le 15 juillet 2007. Elle est volontairement très extensive afin de s'étendre à l'usage par le réseau : sont soumis à la licence tout acte « *other than for Your internal Research and/or Personal Use* ».

21 L'Open Software License (OSL) est une licence issue des travaux de Lawrence Rosen. La version 1.1 date de 2002, elle fut revue en 2003 et la dernière et troisième version date de 2005. Elle emploie le terme d'*external deployment* pour appréhender l'usage par le réseau.

22 La (GNU) Affero GPL a été développée en deux temps : le 1er mars 2002, la société Affero publie une première version, très proche de la GNU GPL qu'elle adapte aux spécificités de l'utilisation par le réseau. Reprise en main par la FSF, la GNU Affero 3.0 a été publiée le 19 novembre 2007.

23 Sur les règles de DIP, voir notamment Hakim HAOUIDEG, *Les logiciels libres et le droit international privé*, In *Les logiciels libres face au droit*, Namur ; Bruxelles : Facultés universitaires Notre-Dame de la paix : Bruylant, 2005.

24 On parle aussi de *réciprocité*.

25 Voir plusieurs licences, comme le fait l'European Public Licence (EURL) au profit de l'OSL, la CPL, l'EPL, la CeCILL et la GNU GPL.

De nombreux domaines sont aujourd'hui concernés par les licences *open source* : les logiciels bien sûr, mais aussi les encyclopédies²⁶, les livres²⁷, la musique²⁸ et bientôt tout type d'œuvres²⁹. La majeure partie des licences libres trouve son fondement dans une application particulière et pour un domaine artistique bien déterminé³⁰ — de ce fait, il est parfois très délicat et déconseillé de les utiliser en dehors³¹. Cette question du domaine, du type d'œuvres, s'applique en pratique lorsque l'on cherche à mesurer l'opportunité de regrouper sous une seule licence l'ensemble des composants d'un produit³² : logiciel, graphisme, scénario, documentation, son, etc. La solution mêlant avantages et inconvénient, tout sera affaire d'espèces³³.

Une dernière classification, issue des travaux de recherche de Mélanie Clément-Fontaine³⁴ distingue les licences en fonction des libertés conférées : les licences qui offrent une liberté pérenne ; les licences qui offrent une liberté fragile ; et enfin les licences qui offrent une liberté asymétrique³⁵.

B. CHOISIR UNE LICENCE OPEN SOURCE

Le choix d'une licence *open source* suit un processus particulièrement simple à décrire, il faut néanmoins ajouter une variable bien plus aléatoire qui consiste à gérer les problématiques de compatibilité vis-à-vis des autres œuvres incluses dans l'œuvre ou comprise dans la portée de la licence.

1. Le choix d'une licence adaptée aux besoins spécifiques de la société et du produit

Le choix de la licence lors du lancement du projet ou de la publication de l'œuvre est d'autant plus important qu'il ne sera ensuite plus modifiable qu'avec l'accord de l'ensemble des contributeurs et cessionnaires de droits.

Le mécanisme de « libération » d'un logiciel — ou de toute autre œuvre — s'avère être une juste combinaison de stratégie et de tactique. L'analyse des utilisations actuelles ou prospectives permet d'opter pour une licence stratégique, les ajustements tactiques adaptent ensuite cette dernière aux attentes propres de la société en modulant les droits et obligations. L'avantage d'une telle technique est de répondre à tout type de besoin avec un minimum de licences. Par ailleurs, l'attention du licencié est retenue sur les modulations ajoutées à la licence utilisée, ce qui simplifie la compréhension de la licence globale.

a) Le choix stratégique de la licence

L'existence de centaines de licences laisse présager un nombre tout aussi grand de situations particulières les ayant justifiées. Comme précisé ci-dessus, ce sont essentiellement les obligations qui différencient les licences : les clauses *copyleft*, les clauses relatives aux usages par le réseau, les clauses relatives aux brevets, DRM, Tivoïzation, la langue utilisée³⁶, etc.

26 On pense bien sûr à Wikipedia, mais de nombreuses autres ont vu le jour.

27 Des sociétés d'édition en faisant d'ailleurs leur crédo, telle la société In Libro Veritas qui ne publie que des ouvrages sous licence libre ou ouverte.

28 De nombreux industriels s'intéressent à cette manne de contenu qu'offre la musique libre (Jamendo, Dogmazic, etc.). C'est ainsi que l'on voit des accords se multiplier entre constructeur et gestionnaire de contenu (comme récemment Jamendo et Archos).

29 D'autres catégories de contenu, comme les données géographiques, se sont aussi tournées vers les licences libres (avec des initiatives comme OpenStreetMap) sans qu'il soit possible d'assurer la validité de ces contrats.

30 À l'exception de quelques rares licences qui ont pour ambition de s'étendre à l'ensemble des créations couvertes par le droit d'auteur, voire toute la propriété littéraire et artistique. L'OSL et la LAL sont deux très bons exemples avec des origines bien différentes.

31 Par exemple, le formalisme extrêmement contraignant de la GNU GPL n'est pas du tout adapté à la musique (il s'agit notamment de l'obligation d'attacher le texte complet de la licence à l'œuvre).

Inversement, des licences comme les *Creative Commons* ont été écrites pour la musique, le film, les livres, etc., mais pas pour le logiciel. Licencié un logiciel sous *Creative Commons* — comme une CC-BY-SA, pour prendre une licence libre assez similaire à la GNU GPL — n'est pas conseillé puisque tous les aspects propres au logiciel sont inexistant : la distribution du code source n'étant notamment pas envisagée.

32 Notamment en présence d'œuvres dites multimédias qui, par définition, regroupent de multiples œuvres différentes.

33 L'usage semble être de limiter le nombre de licence à une ou deux (lorsque le choix de la première dépend d'un autre) se répartissant sur les différents éléments de l'œuvre : sons et images ; logiciels ; compilation de logiciel ; et interface graphique par exemple.

34 Voir notamment Mélanie Clément-Fontaine, *Les Œuvres Libres*, Thèse sous la direction du Professeur Michel Vivant, Montpellier 1, 2006

35 C'est cette classification qui avait été reprise par les travaux publiés en juin 2007 de la commission spécialisée du CSPLA portant sur la mise à disposition ouverte des œuvres de l'esprit coprésidée par Valérie-Laure Bénabou et Joëlle Farchy

36 Le palmarès revenant à l'EUPL : licence rédigée au sein du programme IDABC et approuvée par la Commission Européenne le 9 janvier 2007, la licence *copyleft* se décline actuellement en vingt-trois langues !!!

Dans sa recherche, l'auteur constatera que certaines licences ne peuvent pas être réutilisées en l'état en dehors de projets particuliers³⁷, que d'autres ne seront manifestement pas adaptées à ses intentions et, pour finir, que seules quelques-unes répondront en totalité ou partie à ses attentes. Une fois ce premier état des lieux réalisé, de multiples facteurs indirectement liés au projet devront pris en considération : la licence doit-elle être connue et reconnue par les communautés ? Doit-on s'inspirer des choix opérés pour des projets concurrents analogues ? Certaines licences doivent-elles être proscrites ?

Toutes ses questions ne trouveront de réponses qu'au cas par cas, en fonction de l'examen et de l'approche choisie par l'auteur. Incontestablement, une licence disposant d'une large notoriété participera avantageusement à la communication qui suivra la libération du logiciel³⁸. En revanche, le choix d'une licence similaire à un projet concurrent est un pari dont il faut bien mesurer les aléas : les contributions pouvant librement circuler d'un projet à l'autre, celui qui réunira la meilleure communauté risque de l'emporter³⁹ et de détourner à son profit les contributeurs de son concurrent ; à l'inverse, bénéficier de l'expérience et de l'expertise déployée par une société concurrente peut justifier un tel choix.

Une fois la licence sélectionnée, une bonne dose de tactique permet de la parfaire.

b) Les ajustements tactiques relatifs au choix de la licence

S'il faut user de stratégie pour choisir la licence, c'est la tactique qui déterminera son adéquation avec le projet. Pour ceci, il est nécessaire de comprendre et de maîtriser d'une part les interprétations et exceptions, et, d'autre part, les multilicences.

L'usage tactique des interprétations et des exceptions

Lorsque l'un des termes de la licence est source d'interprétation, celui qui choisit la licence peut lever l'ambiguïté qu'elle pourrait générer en donnant une portée bien définie à cette clause litigieuse. L'interprétation utilisée par Linus Torvalds lors de l'utilisation de la GNU GPL sur le noyau *Linux* est encore un très bon exemple en la matière : ce dernier a souverainement précisé, s'opposant par ce fait au rédacteur de la licence que les programmes qui ne faisaient que graviter autour du noyau⁴⁰ n'étaient pas partis d'un tout dérivé auquel la GNU GPL s'étendrait⁴¹. De nombreuses clauses imprécises ou équivoques peuvent être interprétées de la sorte lors du choix de la licence.

L'usage des exceptions est une technique plus radicale qui consiste à modifier la licence de base en ajoutant, dans une clause jointe à la licence ou insérée dans les en-têtes, une spécificité qui déroge aux termes initiaux : rendant ainsi la licence finale plus ou moins contraignante. Il est à noter qu'en l'absence de stipulation contraire, une clause additionnelle permissive pourra être supprimée par tout licencié au moment de la redistribution d'une copie du logiciel⁴².

La maîtrise de plus en plus fine de la pratique contractuelle liée aux licences *open source* conduit aujourd'hui à conseiller une pratique réfléchie et adaptée de ces exceptions et interprétations. Les premières confèrent plus de souplesse aux licences⁴³, les secondes assurent une plus grande sécurité juridique. Un bon usage de celles-ci peut permettre de prévoir et de résoudre *a priori* la plupart des situations préjudiciables : problèmes d'incompatibilité, failles au sein des licences, etc.

37 La MPL par exemple : elle demande notamment à ce que toute référence à la marque Mozilla soit supprimée.

38 Inversement, une licence moins connue devra être défendue bec et ongle, *a fortiori* si elle a déjà des détracteurs.

39 Étant convenu que le premier arrivé dispose d'un avantage concurrentiel certain : c'est plus difficile d'intéresser une communauté que de la conserver.

40 Qui, techniquement, n'interagissant que par des appels système normaux.

41 On retrouve là une interprétation tenue au sein de l'OSI. Voir à ce sujet Rosen (Lawrence E.), « Derivative Works », 2002 ; « The Unreasonable Fear of Infection », 2001 et Franz (Michael), « Dynamic Linking of Software Components », IEEE Computer, Mars 1997.

42 Ceci valant bien sûr pour les licences dites permissives, mais aussi pour toute licence *copyleft* puisque le *copyleft* ne perpétue que les termes *stricto sensu* de la licence.

43 La troisième version de la GNU GPL invite même dans un article dédié à user de cette faculté sur ses propres contributions (en distinguant d'une part les permissions additionnelles qui peuvent être ajoutées sans limitation et les termes supplétifs à tirer parmi une liste de sept types de clause).

L'utilisation tactique des Multilicences

On parle de multilicence lorsqu'une seule et même création est soumise à différentes licences : qu'elles soient toutes *open source* ou qu'au moins l'une d'elles ait cette qualité⁴⁴.

Multiplier les licences *open source* peut parfois offrir des situations véritablement avantageuses : assurer une compatibilité au bénéfice de plusieurs licences, bénéficier de la renommée d'une licence, optimiser la cession de droits, etc.

Le navigateur web Firefox, sous MPL/LGPL/GPL⁴⁵, est un très bon exemple de la souplesse qu'offre les licences. En effet, le logiciel est construit de façon modulaire⁴⁶ et gagne à conserver une licence qui intègre ce particularisme⁴⁷, mais, afin de rassurer les développeurs et les communautés sur leur faculté de disposer du code pour d'autres projets, les licences LGPL et GPL s'ajoutent à la MPL. De cette façon, le logiciel libre Firefox dispose d'une licence qui lui permet d'être compatible avec les licences les plus populaires tout en bénéficiant des développements sous la licence la plus adaptée aux projets de Mozilla – notamment une portée réduite qui amenuise l'aspect contraignant de son *copyleft*⁴⁸.

2. L'influence de la compatibilité dans le choix des licences

Si la démarche qui précède permet de s'orienter vers le choix optimal d'une licence, celle-ci peut se révéler impossible à mettre en œuvre si l'auteur est lui-même soumis à d'autres licences contenant des termes incompatibles lors de la distribution ou de l'utilisation du logiciel.

En effet, il n'y a pas *de facto* de licences dominantes : toutes ont la même force obligatoire et le licencié doit renoncer à exploiter une création lorsqu'il se retrouve en situation de ne pas pouvoir respecter l'ensemble de ses engagements. La seule raison pour laquelle il est possible de distribuer sous une seule licence un projet comprenant des briques sous d'autres licences⁴⁹ est que cette distribution sous une licence unique soit permise par l'ensemble des autres licences.

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

a) Gérer la compatibilité entre les licences

La situation la plus commode réside en la compatibilité expresse : soit la licence prévoit que le logiciel puisse indifféremment être redistribué selon ses propres termes ou selon ceux d'une licence déterminée⁵⁰, soit la compatibilité est limitée aux seules situations où l'ajout d'une brique sous une autre licence *open source* impose de redistribuer le tout sous cette unique autre licence⁵¹.

En l'absence d'une telle compatibilité, il convient de procéder à une analyse logique du contexte : l'enjeu consiste à gérer les flux de droits et d'obligations de la société pour savoir ce qu'elle peut faire ou ne pas faire. Et, lorsque l'on souhaite céder une contribution pour laquelle on est seulement licencié, les questions se résument à : s'assurer que l'on est bien en mesure de céder tous les droits que la licence confère et à veiller que l'on n'oblige pas moins l'utilisateur final que l'on est soi-même obligé⁵². La dernière version de la GNU GPL (v 3) s'appuie sur ce principe pour accueillir en son sein une modularité qui lui permet d'assurer la

44 Ce phénomène de double licence mixant licence libre et licence commercial sera examiné en deuxième partie.

45 Après avoir été quelque temps uniquement sous MPL, il a été estimé que cette seule licence n'était pas suffisante à motiver les contributions.

46 Chaque nouvelle brique apportant une nouvelle fonctionnalité autour d'un noyau. De nombreux projets *open source adoptent* cette structure qui permet notamment de répartir efficacement les tâches de développement.

47 En effet, la MPL permet l'adjonction de plug-in ou toute autre extension au logiciel tant que chaque fichier contenant du code sous MPL soit exclusivement sous MPL. Cette tolérance permet une plus large diffusion du logiciel.

48 La conséquence logique d'une telle multilicence est d'ailleurs que le *copyleft* final est celui le moins contraignant de l'ensemble des licences.

49 Par exemple, distribuer sous GNU GPL v3 une solution qui comporte des briques sous licence BSD et Apache.

50 C'est notamment ce que faisait la LGPL v2 au bénéfice de la GNU GPL.

51 Ce que l'on retrouve notamment dans l'European Public Licence au profit de cinq licences que sont l'OSL v. 2.1 et v. 3.0, la CPL v. 1.0, l'EPL v. 1.0, la CeCILL v. 2.0 et la GNU GPL v. 2.0

52 Cette compatibilité est aujourd'hui de mieux en mieux expliquée, comme le démontre un papier récent de la Software Freedom Law Center qui détaille la procédure permettant de conserver la licence BSD sur du code distribué sous Licence GNU GPL.

compatibilité à l'égard d'autres licences libres auparavant incompatibles⁵³ : en permettant, sur chaque nouvelle contribution, d'une part l'ajout de « permissions additionnelles » et d'autre part l'ajustement par l'utilisation de certains « termes supplétifs » limitativement autorisés. Le problème n'est néanmoins pas tout à fait résolu puisqu'il faudra ensuite confronter chaque licence aux différentes clauses, au cas par cas, afin d'être certain de leur compatibilité⁵⁴.

Cette réflexion peut se traduire mathématiquement grâce à la théorie des ensembles sous la forme qui suit⁵⁵ : la licence B est dite compatible (c'est-à-dire qu'elle peut être utilisée pour redistribuer le logiciel en respectant les termes de la licence originaire (A)) *si et seulement si* l'ensemble des droits de la licence B est inclus dans la licence A et que l'ensemble des obligations de la licence A est inclus dans la licence B. Ainsi, une licence est compatible avec elle-même, une licence *copyleft* n'est pas forcément compatible avec une licence permissive et une licence permissive ne l'est forcément pas à l'égard d'une licence *copyleft*.

Attention, la version des licences constitue dans certains cas un élément déterminant : même si la plupart permettent une compatibilité ascendante au bénéfice des nouvelles versions de la licence, certaines autorisent aussi aux auteurs de figer la licence à une version spécifique⁵⁶. Il y a alors un *copyleft* en deux temps : le premier caractérisant l'étendue de la licence et le second déterminant la version sur chaque contribution.

La situation la plus confortable consiste à rester dans la même « famille de licences » : il en existe plusieurs et celles-ci couvrent en théorie un large panel de choix tout en assurant une compatibilité parfaite entre elles⁵⁷.

b) Les solutions alternatives en cas d'incompatibilité

Certains conflits ne peuvent contractuellement se résoudre et si l'œuvre litigieuse s'avère trop complexe pour être réécrite, il faudra alors renoncer à la distribution ou à l'utilisation du tout⁵⁸.

Il reste néanmoins quelques autres solutions : contacter le ou les auteurs d'une brique litigieuse pour leur demander une dérogation, une exception ou un changement de licence ; acheter une licence commerciale à ces derniers ; se faire céder la titularité de leurs droits ; ou enfin contourner matériellement le problème posé par la licence.

Le premier expédient à l'avantage de la simplicité, mais son succès dépendra du nombre d'auteurs ayant participé à cette brique, de leur personnalité et de ce que l'entreprise est prête à investir dans le projet. Les deux autres mesures dépendront aussi totalement de la volonté des titulaires de droits. Le dernier procédé se révèle le plus précaire puisqu'il nécessite une connaissance parfaite des licences afin d'estimer ce qui peut matériellement être permis et ce qui ne le serait pas (certaines formes de distribution⁵⁹, certains modes de distribution⁶⁰ et certaines constructions du logiciel⁶¹ permettent par exemple d'alléger les contraintes).

Une fois adopté le choix définitif de la licence, on réalise que celui-ci ne constituait en réalité que la « première étape » : déterminant, mais non suffisant à construire sur le modèle de l'*open source*.

53 Comme les licences plus permissives comme les licences Apache, LaTeX, etc.

54 Avec le risque de divergence d'interprétation, comme c'était déjà le cas les licences GPL et Apache.

55 Sur les questions de compatibilité entre licences, voir Benjamin Jean, *Option Libre : « compatibilité entre contrats »*, mémoire de DEA sous la direction de Monsieur le Professeur Michel Vivant, Montpellier I / CNRS, 2006.

56 Si on prend l'exemple de la GNU GPL et du noyau *Linux* : toutes les contributions de Linus Torvald sont distribuées sous GNU GPL « version 2 seulement », ainsi, même si d'autres parties autorisent la distribution sous GNU GPL version 3, le noyau *Linux* ne pourra pas être distribué sous la troisième version de la GNU GPL tant que les briques sous GPL v2 n'auront pas été réécrites ou que leur auteur aura consenti à modifier la licence !

57 La plus connue est certainement la famille des licences GNU (GPL, LGPL, AGPL, etc.), mais bien d'autres peuvent être citées pour l'exemple : CeCILL (-A, -B, -C), OSL (OSL, AFL), etc.

58 Sauf bien sûr à trouver un remplacement pour la partie litigieuse.

59 Par exemple exécutable, tant que les sources des modifications sont disponibles par ailleurs.

60 On pense notamment aux logiciels dont seul le client est distribué alors que le serveur est hébergé chez l'éditeur.

61 Par exemple la construction modulaire pour l'utilisation de briques sous licence du type MPL, voire LGPL.

II. Les pratiques encadrant l'utilisation des licences *open source*

Ainsi, la licence *open source* utilisée n'est que l'une des facettes du système juridique organisé autour du logiciel. L'ensemble des autres droits de propriété intellectuelle (parfois les brevets⁶², mais surtout les marques et autres signes distinctifs), auquel s'ajoute une maîtrise fine des contrats, permettent de contrôler avec précision l'exploitation du logiciel.

A. DES INTÉRÊTS DIVERGENTS RELATIFS À L'ACTIVITÉ DE LA SOCIÉTÉ

Du logiciel dépend aussi le type de développement libre qui peut lui être associé : les outils et les besoins des éditeurs et des intégrateurs ne sont pas les mêmes, de la même façon qu'un logiciel (dont l'utilisation est) « grand public » ne devra pas être exploité de la même façon qu'un logiciel à destination des professionnels.

1. L'édition *open source*

Les schémas mis en place dans le monde de l'édition répondent à de multiples paysages, principalement dus au produit édité et aux utilisateurs concernés. Il est possible d'en dresser une liste non exhaustive qui détaille les principaux mécanismes, qui sont librement combinables.

Le plus souvent, les éditeurs assurent la protection de leur produit par l'usage du droit des marques⁶³ : le logiciel est librement diffusé sous licence *open source*, mais tout usage de la marque⁶⁴ est soumis à autorisation. Cette faculté qu'offre le cumul de droits exclusifs concurrents sur un même objet a ici l'énorme avantage de rendre quasi inexistant le coût d'entrée pour les utilisateurs, tout en assurant à l'éditeur principal un contrôle sur l'évolution, la distribution des copies et l'exploitation globale du logiciel. Les sociétés commerciales ont tendance à user le droit des marques de façon très agressive — allant jusqu'à limiter, au moins en partie, la jouissance concédée par la licence *open source* sur le logiciel⁶⁵ — tandis que les structures non commerciales en ont un usage plus modéré et équilibré qui a principalement pour objectif de défendre l'image associée au logiciel⁶⁶. Le risque inhérent à ce type d'utilisation du droit des marques est que l'éditeur, trop mercantile, se met à dos la communauté qui s'émancipe alors en enlevant toute allusion à la marque du logiciel : étant sous licence *open source*, celui-ci redevient alors pleinement exploitable⁶⁷.

Tous les éditeurs actuels ne font pas le choix exclusif de l'*open source* et nombreux sont ceux qui développent des offres commerciales. Certains proposent d'associer à la version libre des *add-ons*, *plug-ins*, etc. qui ne sont pas diffusés sous licence *open source*⁶⁸ et d'autres proposent complètement une licence commerciale présentée comme alternative à la licence *open source*⁶⁹ (cette deuxième solution nécessitant que

62 Nous ne parlons ici que d'une pratique outre-Atlantique, cette tendance est bien sûr tout à fait étrangère aux sociétés européennes. La société Microsoft est en effet particulièrement active dans le domaine et multiplie les accords de non-agression et de partenariat en faveur des sociétés éditrices de logiciel *open source*.

63 Article L. 711-2 et suivants du CPI.

64 L'article L. 713-2 disposant que « Sont interdits, sauf autorisation du propriétaire :

a) la reproduction, l'usage ou l'apposition d'une marque, même avec l'adjonction de mots tels que : « formule, façon, système, imitation, genre, méthode », ainsi que l'usage d'une marque reproduite, pour des produits ou services identiques à ceux désignés dans l'enregistrement ;
b) la suppression ou la modification d'une marque régulièrement apposée. »

65 Un très bon exemple est la société américaine Red Hat : elle n'autorise aucun usage de ces marques, sauf : pour un usage non commercial par des établissements éducatifs, des associations à but non lucratif, aux groupes d'utilisateurs et leurs personnels correspondants ; pour un usage personnel ou professionnel issu d'une version originale sous la condition de non-redistribution...

Mandriva propose, elle, une licence de marque pour la distribution d'une version non modifiée d'un produit Mandriva lorsque celui-ci est disponible gratuitement sur son site internet.

66 La Mozilla Foundation limite par exemple l'usage des marques Firefox et Thunderbird selon la version du logiciel (licence complète sur la version binaire non modifiée, licence large pour les versions officielles localisées, licence réduite pour la version communautaire et enfin aucune licence pour une version du logiciel trop modifiée.

67 Red Hat a ainsi vu naître CentOS, version « expurgée » de la première (opération qui est par ailleurs techniquement compliquée par Red Hat) et Firefox s'est vu opposer Icewease (avec un succès plus mitigé).

68 Cette pratique est surtout répandue en matière de distribution ou compilation de logiciel, c'est-à-dire des situations où des briques non libres peuvent être mêlées à d'autres qui le sont.

C'est notamment le cas de Red Hat avec ou Mandriva (Mandriva Linux Powerpack Edition, Mandrivalinux Move, etc.)

69 Encore faut-il que cette licence soit suffisamment contraignante pour intéresser au choix de la licence commerciale.

Ce sont les exemples de MySQL AB, Sleepycat Software Inc., TrollTech AS, etc. Ainsi, MySQL AB qui propose MySQL Entreprise sous licence GNU GPL v2 (avec exception en faveur d'autres logiciels libres) ou sous licence propriétaire pour ceux qui ne désire pas respecter la licence

la société détienne tous les droits sur le logiciel⁷⁰). Mixer licences *open source* et commerciales⁷¹ permet des montages intéressants : le principe réside alors à s'appuyer sur la non-exclusivité des cessions pour construire un équilibre par lequel les communautés continuent à trouver intérêt à utiliser et collaborer au développement du logiciel tandis que les utilisateurs professionnalisés préfèrent à terme opter pour une licence commerciale⁷².

Parfois, certains logiciels orientés « métiers » demandent de concevoir plus finement la collaboration entre les différents industriels, la licence *open source* étant insuffisante. S'ils sont concurrents, une licence *copyleft* sera une condition *sine qua non* afin d'assurer une relative pérennité et équité dans le développement du logiciel. On voit alors s'élaborer des contrats de consortium venant compléter tous les aspects non couverts par licence (comité de direction, procédé de développement, prise en compte des orientations, marketing et communication, etc.)⁷³. Ainsi, la licence *open source* assurera la confiance entre ces derniers vis-à-vis de ce qui est créé et permettra éventuellement à d'autres acteurs de bénéficier d'un coût d'entrée quasi inexistant — de façon à ce qu'ils puissent par la suite éventuellement rejoindre le cercle des partenaires ; par ailleurs, le consortium jouera le rôle d'un contrat d'indivision par lequel les partenaires s'organisent pour la gestion du projet.

Enfin, certaines sociétés usent du droit des contrats afin de limiter « accessoirement » l'utilisateur dans les droits que lui concède la licence *open source* : que ce soit lors de la vente de services aux utilisateurs qui souhaitent bénéficier du support et des mises à jour⁷⁴ ou dans les rapports entre distributeurs et revendeurs⁷⁵.

Dans cette diversité de comportement et de méthodes commerciales, il est intéressant de constater le rejet unanime des brevets logiciels : tant par les licences assez récentes⁷⁶ que par les industries attachées au logiciel libre. Celles-ci n'hésitent plus à opérer des regroupements des brevets à titre défensif : initiatives individuelles⁷⁷ ou collectives⁷⁸, elles permettent de mutualiser les efforts défensifs dans l'objectif d'immobiliser toute action tierce en contrefaçon de brevet (notamment du fait des patents trolls) : en s'engageant d'une part à ne pas les opposer aux communautés *open source* et, d'autre part, à s'en servir pour protéger ces communautés.

Les arguments juridiques ne sont probablement pas les plus déterminants pour justifier l'opposition aux brevets logiciels, mais l'une des difficultés bien réelles est l'impossibilité de contourner un brevet en cas de doute (comme on peut le faire aujourd'hui en réécrivant le passage litigieux). Ainsi, les particuliers qui ne disposent d'aucune information sur les brevets tiers seront définitivement écartés de la conception, ou au moins de l'exploitation, de code.

(notamment pour développer ou distribuer un projet propriétaire) ou veulent bénéficier de mises à jour, de service support et assurance.

70 Ce dont MySQL s'assure en n'intégrant dans le produit que les contributions qu'elle a pu réécrire ou racheter.

71 Le terme peut sembler ambigu : une licence *open source* autorise parfaitement un usage commercial du logiciel, mais elle n'en restera pas moins une cession non exclusive gracieuse ; à l'inverse, une licence dite commerciale sera conditionnée au paiement d'une souscription quelconque.

72 À la différence de la licence *open source* par ailleurs proposée (probablement *copyleft* ou tout du moins suffisamment contraignante pour que l'utilisateur ait un intérêt à opter pour la licence commerciale), la licence commerciale de l'éditeur offre aux clients des avantages comme la possibilité de ne pas diffuser le contenu de leur création dérivée (échappant ainsi à la réciprocité afin de continuer de disposer de leur monopole).

73 De type consortium : l'exemple du consortium Objectweb (aujourd'hui OW2) est le parfait exemple d'un regroupement de grandes sociétés autour de logiciels communs développés sous licence libre.

74 C'est notamment le cas de Red Hat avec les produits comme *JBoss Enterprise Middleware*, *Red Hat Directory Server*, *Red Hat Certificate System*, etc.

75 On retrouve ce type de limitations dans les contrats de MySQL, Ingres, Alfresco, etc. Le revendeur se retrouvant bien souvent contraint de ne pas utiliser ou conseiller la version communautaire.

76 Cet enjeu a parallèlement été intégré dans les licences, de différentes manières : en ajoutant dans la licence une cession non exclusive de brevet calquée sur celle portant sur le droit d'auteur (comme la licence Apache) ou en créant des « clauses de représailles » qui résilient toute cession de droits si le licencié vient opposer ses brevets sur le logiciel ou vis-à-vis d'une communauté (comme la *Common Public License*).

La GNU GPL, dans sa dernière version, encadre aussi les promesses qui seraient faites de ne pas agir à l'égard de certaines personnes afin de rompre l'inégalité ainsi générée entre les utilisateurs (en réaction contre l'accord Microsoft Novell de 2007).

77 Plusieurs milliers de brevets sont ainsi regroupés en garantie par des sociétés comme IBM, Nokia, Sun, etc. — notamment l'« *IBM Statement of Non-Assertion of Named Patents Against OSS* », le « Sun patent program » ou encore le « Novell Statement on Patents and open source Software ».

78 Notamment l'*open source Development Laboratory* (OSDL), « no software patents », l'*Electronic Frontier Foundation* (EFF — avec notamment l'initiative *Patent busting project*), la *Foundation for a Free Information Infrastructure* (FFII) ou l'*Open Invention Network* (OIN) auquel ont souscrit des sociétés comme Sony, IBM, NEC, Red Hat, Philips et Novell.

2. Le service et l'assistance autour de Logiciels Libres

De son côté, l'intégrateur — et plus généralement tous les non-éditeurs — recherche la simplicité dans un système qui lui permette de combiner les briques logicielles entre elles pour créer sans cesse de nouvelles combinaisons. Il n'y a pas ici d'usage du droit des marques à l'égard des produits, mais surtout une expertise approfondie des combinaisons de licences : lui permettant de s'assurer de la faisabilité du projet et de s'engager sur ce point auprès de son client et dans ses contrats.

En ce qui concerne les contributions, les pratiques sont diverses, mais on observe tout de même une spécificité due aux logiciels. Ainsi, l'intégrateur transférera très souvent un logiciel qu'il n'a pas lui-même développé, mais seulement adapté au besoin de son client : de ce fait, lorsqu'il fournit la solution objet du contrat, il est généralement lui-même limité par sa qualité de licencié à l'égard des auteurs des logiciels libres sur lesquels il intervient, et, *ne pouvant céder plus de droits qu'il n'en possède lui-même*, il transférera au client cesdits logiciels sous leur licence *open source* d'origine — ou toute autre licence compatible. Concernant ses propres développements, il semblerait finalement que le plus avantageux pour l'ensemble des parties soit que l'intégrateur cède l'ensemble de ses développements sous licence libre au client et qu'il conserve la titularité des droits : ce compromis reste fidèle au procédé de création *open source* dans lequel l'auteur reste titulaire de droits et cède sous licence *open source* de façon non exclusive, et permet tant au client de mutualiser les résultats de la prestation qu'à l'intégrateur de valoriser les travaux par tous moyens appropriés⁷⁹.

B. LA PLACE DU SERVICE JURIDIQUE DANS LES PRATIQUES LIÉES À L'UTILISATION DES LICENCES OPEN SOURCE

En se limitant à cette seule finalité de « pratiques liées à l'utilisation de licences *open source* », on s'aperçoit que les compétences requises sont assez larges : en propriété intellectuelle (appréhension globale et expertise plus poussée en matière de droit d'auteur et de signes distinctifs), de droit des contrats et de la concurrence en ce qui concerne la contractualisation et l'exploitation des produits.

En toute hypothèse, nous sommes ici loin de l'image du juriste qui siège du haut de sa tour d'ivoire : s'il lui est nécessaire de prendre de la hauteur pour agencer au mieux l'utilisation des différents droits exclusifs, il lui faut aussi être en contact direct avec l'équipe technique à l'origine des logiciels, les différents chefs de projets pour leur vision générale et les dirigeants pour décider des orientations à prendre.

1. Le travail au contact des équipes techniques

De façon pragmatique, lorsque l'on souhaite exploiter du contenu sous licence libre — en entreprise ou ailleurs —, il est nécessaire d'adapter la procédure de validation juridique et contractuelle : l'analyse se réalisant autant en amont des projets qu'en aval.

Dès en amont, il est nécessaire de déterminer la faisabilité juridique des propositions techniques et s'assurer que les engagements puissent être respectés (en gardant en tête le *leitmotiv*, qui est que l'« on ne peut donner plus de droits que l'on en possède » — c'est d'ailleurs cette seule idée qui ressort des contentieux français ayant mis en jeu la GNU GPL⁸⁰). Il sera par ailleurs fréquemment nécessaire d'intervenir en cours de négociation afin d'expliquer les choix de licences, ses implications et éventuellement adapter en fonction des éléments qui lui sont donnés.

Enfin, une dernière étude juridique est nécessaire en aval pour confirmer le choix final de la licence et procéder s'il y a lieu aux ajustements contractuels, en fonction du produit, des attentes et bien évidemment des licences en présence.

Ce travail ne peut être mené que conjointement entre l'équipe technique, seule apte à fournir les informations

⁷⁹ L'intégrateur se révèle être le mieux placé pour faire évoluer le produit et assurer la pérennité de son développement. Sa situation d'intermédiaire lui permet par ailleurs d'être idéalement placé pour favoriser les échanges indirects de contributions entre ses différents clients – l'évolution des solutions étant aussi dans l'intérêt de ceux-ci, tout le monde profite de cette collaboration.

⁸⁰ Il existe au moins deux jugements français en la matière : TI de Paris, le 28 mars 2007 et TGI de Chambéry, le 15 novembre 2007. Leur portée doit néanmoins être relativisée puisque dans les deux affaires le juge n'est pas saisi de la validité de la licence *open source* (GNU GPL).

détaillées sur le logiciel, et l'équipe juridique. Les données brutes nécessaires à l'étude juridique sont : le détail des briques logicielles utilisées, leur(s) licence(s) respective(s) (comprenant leur version, et toute autre information influant sur ces licences : en tête restrictive, interprétation, exception, etc.), et enfin les interactions existantes entre chacune de ces briques (en pratique, un schéma). Un dialogue s'établit par la suite pour affiner les derniers détails. Bien évidemment, de nombreuses navettes sont généralement indispensables à la conclusion du travail et il est très fréquent que le juriste ait à mettre les mains dans le code source pour vérifier l'existence et l'application des licences.

2. Le travail complémentaire lié à l'usage des licences open source

Tous les développements qui précèdent illustrent les enjeux attachés à une bonne expertise et utilisation de la propriété intellectuelle. Si certains comportements adoptés peuvent bousculer les manières habituelles, les finalités des différents droits de propriété intellectuelle n'en sont pas moins conservées puisque leur respect permet de garder la cohérence du système.

Les tâches à la charge du service juridique sont multiples : certaines étant classiques (maîtriser les licences, leur compatibilité et les techniques contractuelles afférentes ; maîtriser la propriété intellectuelle de manière générale) et d'autres propres au *business model* adopté. Par exemple, le choix d'une double licence (*open source* ou commerciale) influe de façon notable sur ce travail puisqu'il est alors nécessaire que la société reste l'unique titulaire de droits (afin de ne pas être elle-même soumise à la licence).

En outre, une sensibilisation des collaborateurs en matière de Propriété Intellectuelle et éventuellement de licences libres est d'autant plus importante que ceux-ci seront emmenés à brasser énormément de contenu soumis à un ou plusieurs droits exclusifs. De manière générale, il est nécessaire, au sein de la Société, d'adapter les pratiques internes (charte des contributions, charte de développement, etc.) comme externes (contrat d'assistance, de services, CGV, etc.) autour de cet objectif afin de pérenniser les engagements des salariés⁸¹ et de la société⁸². Juriste *open source*, un métier d'avenir !

81 Les créations salariales hors logiciels sont bien souvent plus facilement gérées grâce à une logique similaire aux contributions externes : étant le plus souvent disponibles sous licence libre, il suffit d'en respecter les termes.

82 D'autant que de nombreux risques inhérents à l'*open source* ne sont toujours pas assurables en Europe, à l'opposé des États-Unis qui sont encore en avance sur ces questions-là.