



**HAL**  
open science

## Brevets et open source: combiner l'incombinable

Gabriel Bouvigne

► **To cite this version:**

Gabriel Bouvigne. Brevets et open source: combiner l'incombinable. domain\_shs.info.econ. 2019. mem\_03573548

**HAL Id: mem\_03573548**

**[https://memic.ccsd.cnrs.fr/mem\\_03573548v1](https://memic.ccsd.cnrs.fr/mem_03573548v1)**

Submitted on 14 Feb 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NoDerivatives 4.0 International License

**Master 2**  
**STRATEGIE DE LA PROPRIETE INTELLECTUELLE ET**  
**INNOVATION**  
**Promotion 2018-2019**

**MEMOIRE**  
**BREVETS ET OPEN SOURCE : COMBINER**  
**L'INCOMBINABLE**

Gabriel BOUVIGNE

*Les opinions exprimées dans ce document sont propres à l'auteur; et ne représentent pas nécessairement celles de ses clients et/ou employeurs.*

## Résumé

Depuis vingt ans, l'open source prend une place grandissante dans l'informatique. Les logiciels open source, initialement considérés comme marginaux, sont aujourd'hui nécessaires au fonctionnement de toutes les entreprises. Les acteurs du secteur informatique considèrent l'utilisation de l'open source comme un axe stratégique. En parallèle, les brevets ont également une haute importance dans les préoccupations et les stratégies des entreprises du secteur informatiques et des secteurs voisins.

Il semble cependant y avoir contradiction entre les idéaux et objectifs de l'open source (moyen de permettre aux tiers une libre utilisation du logiciel) et ceux des brevets (moyen de restreindre l'utilisation par les tiers de l'objet breveté). La combinaison de ces deux outils, en apparence opposition, constitue le thème de ce mémoire.

Dans une première partie, nous traitons des problématiques liées à la contrefaçon de brevet par le logiciel open source : nous abordons les points spécifiques liés au logiciel open source dans le cadre de la contrefaçon, puis étudions la façon dont le logiciel open source impacte les possibilités d'action dans le cas d'une contrefaçon.

Dans une seconde partie, nous étudions les façons de combiner brevets et open source, afin d'utiliser le brevet pour soutenir une stratégie open source.

Enfin, dans une troisième partie, nous observons la matérialisation des interactions entre ces deux domaines : nous quantifions les portefeuilles de brevets des entreprises du secteur open source, et nous observons l'impact des brevets sur les logiciels open source dans deux domaines, retenus comme exemple (l'encodage audionumérique au format MP3, et un sous-ensemble de l'*Internet of Things*).

Ce mémoire montre qu'il est tout à fait réalisable, et même avantageux, dans le cadre d'une stratégie open source, de lui adjoindre une stratégie de brevets. Il faut cependant garder à l'esprit que la prolifération des brevets dans les secteurs de l'informatique et des télécoms pose un certain nombre de problèmes, allant au-delà des logiciels open source. Ces problèmes pourraient impacter négativement le développement de l'économie, et il est souhaitable que des évolutions positives soient apportées.



# Sommaire

<b>Introduction .....</b>	<b>9</b>
I - L'expression et la fonction : la double protection du logiciel .....	10
II - Le brevet .....	11
A - Historique.....	11
B - Types de brevets .....	12
C - Le brevet, le logiciel et le brevet logiciel .....	13
III - Le logiciel open source .....	15
A - Historique.....	15
B - Différence entre « open source » et « logiciel libre » .....	17
C - Les motivations derrière l'open source.....	18
<b>Première partie : La contrefaçon de brevet par le logiciel open source.....</b>	<b>21</b>
I - Contrefaçon de brevet - rappels.....	21
A - Limites géographiques.....	21
B - La contrefaçon de brevet en droit français. ....	21
C - Contrefaçon littérale et fourniture de moyens.....	22
II - Points particuliers liés à l'open source .....	22
A - Détection aisée de contrefaçon .....	22
B - Traçabilité des contributions .....	23
C - Le cadre privé et l'exception de recherche .....	23
D - Différents cas de distribution .....	24
1. Les acteurs de la distribution .....	24
2. Les formes de distribution.....	25
III - Comment agir en cas de contrefaçon .....	27
A - Agir à l'encontre de l'entité utilisatrice .....	28
B - Agir à l'encontre du distributeur .....	28
C - Agir à l'encontre de l'éditeur .....	29
D - Agir à l'encontre des intermédiaires techniques ?.....	30
E - Les clauses de résiliation de licence .....	30
<b>Seconde partie : Combiner brevets et open source .....</b>	<b>33</b>
I - Le brevet pour soutenir une stratégie open source .....	34
A - Contrôler la distribution de la connaissance .....	34
1. Palier à la perte de savoir-faire.....	34
2. Favoriser la libre exploitation des connaissances dérivées .....	35

B - Faire partie de l'écosystème des brevets .....	36
1. Ajout à l'état de l'art .....	36
2. Crédibilité envers les partenaires .....	37
C - Conquête .....	37
1. Faire respecter les termes de la licence .....	37
2. Accroître ses parts de marché .....	38
D - Défense.....	38
1. Licences croisées .....	38
2. Portefeuille défensif .....	38
E - Autres stratégies « habituelles ».....	39
II - Points de vigilance .....	39
A - Compatibilité entre licences open source et ses propres brevets .....	40
1. Code couvert par ses propres brevets .....	40
2. Redistribution du logiciel .....	41
B - Compatibilité entre licences GPLv2/LGPLv2.1 et brevets licenciés.....	42
C - Contributor license agreement.....	45
D - Référentiel de code et divulgation .....	46
1. Emploi d'un dépôt privé .....	46
2. Le délai de grâce .....	48
III - Le partage de brevets .....	50
A - Patent pledges .....	51
B - Open Invention Network .....	52
C - Licences ouvertes de brevets .....	52
<b>Troisième partie : Observations de terrain .....</b>	<b>55</b>
I - Portefeuilles brevets des entreprises travaillant exclusivement dans l'open source .....	55
A - Données.....	55
B - Analyse .....	57
II - Exemples d'interactions entre open source et technologies brevetées.....	59
A - L'encodage MP3 .....	59
1. La technologie.....	59
2. Les brevets relatifs à la technologie .....	59
3. Les projets open source .....	61
4. Effet des brevets sur les projets open source .....	62
B - L'IoT (Internet of Things).....	63
1. Le domaine technique .....	63
2. Les brevets relatifs à la technologie .....	63
3. Les projets open source – <i>frameworks</i> de communication IoT .....	65
<b>Conclusion.....</b>	<b>67</b>
<b>Bibliographie .....</b>	<b>71</b>

<b>Annexe 1 : Définitions</b> .....	<b>75</b>
<b>Annexe 2 : Sections relatives aux brevets des licences open source les plus usuelles ..</b>	<b>79</b>
I - Apache license .....	80
II - BSD license.....	80
III - General Public License (GPL) v2.....	80
IV - General Public License (GPL) v3.....	81
V - Lesser General Public License (LGPL) v2.1.....	83
VI - Lesser General Public License (LGPL) v3.....	83
VII - MIT license .....	83
VIII - Mozilla public license (MPL).....	84
<b>Annexe 3 : Exemples de Contributor License Agreements</b> .....	<b>85</b>
I - Google Individual Contributor License Agreement (Android Open Source Platform).....	85
II - Microsoft Contribution License Agreement .....	87
<b>Annexe 4 : Mozilla Open Software Patent License Agreement .....</b>	<b>89</b>



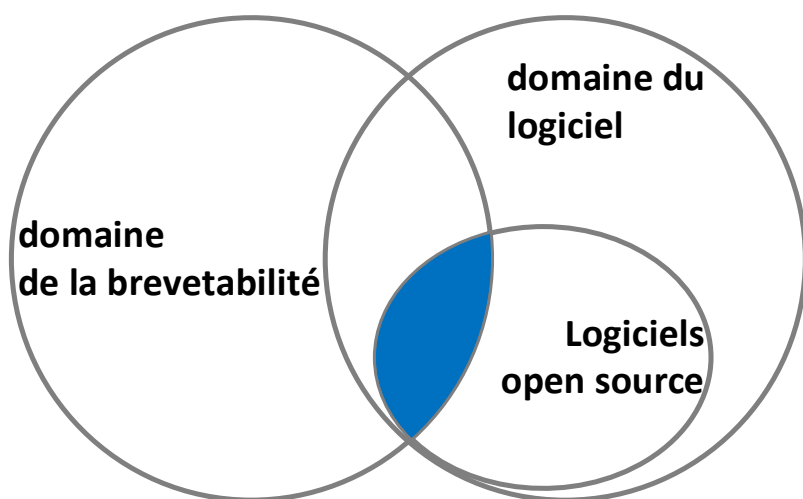


# INTRODUCTION

Les brevets d'invention donnent un certain nombre de droits sur des inventions de nature technique. Étant entendu que les logiciels sont de nature technique, il y a donc une intersection possible entre brevets<sup>1</sup> et logiciels lorsqu'un système, ou un élément de ce système, est réalisé en tout ou partie par le biais d'une implémentation sous forme logicielle.

Le principal droit conféré par le brevet est un droit d'exclusion envers les tiers, il s'agit donc d'un droit « négatif ». Dans le cas d'un logiciel open source<sup>2</sup>, les principaux droits accordés sont le droit de modification et le droit de redistribution, il s'agit d'un droit « positif ». Nous avons donc, à l'intersection du domaine du brevet et de celui du logiciel, la conjonction possible entre un droit positif et un droit négatif, pour les logiciels couverts par une licence open source.

*Figure 1 : En bleu, l'intersection faisant l'objet de la présente étude*



Avant de nous pencher plus particulièrement sur cette intersection, il conviendra d'effectuer de brefs rappels, d'une part sur le brevet et d'autre part sur l'open source, afin de permettre de délimiter les contours de la présente étude.

---

<sup>1</sup> Pour ne pas alourdir inutilement la rédaction, le terme « brevet d'invention » sera simplement nommé « brevet » dans le présent document.

<sup>2</sup> Nous précisons plus loin dans cette introduction quelle signification nous donnons ici au terme « open source », ce terme pouvant potentiellement faire l'objet d'interprétations différentes en fonction des contextes.

## I - L'expression et la fonction : la double protection du logiciel

Tout d'abord, il nous faut rappeler la particularité de la double protection possible du logiciel par les droits de propriété intellectuelle : le logiciel peut être protégé à la fois par le droit d'auteur, et par le brevet. Cette double protection suscite parfois débat, un même élément ne pouvant normalement pas faire l'objet de deux protections distinctes par des droits de propriété intellectuelle. Certains considèrent cette double protection comme une anomalie, qualifiant à ce titre le logiciel de « cumulard » (Macrez, 2011).

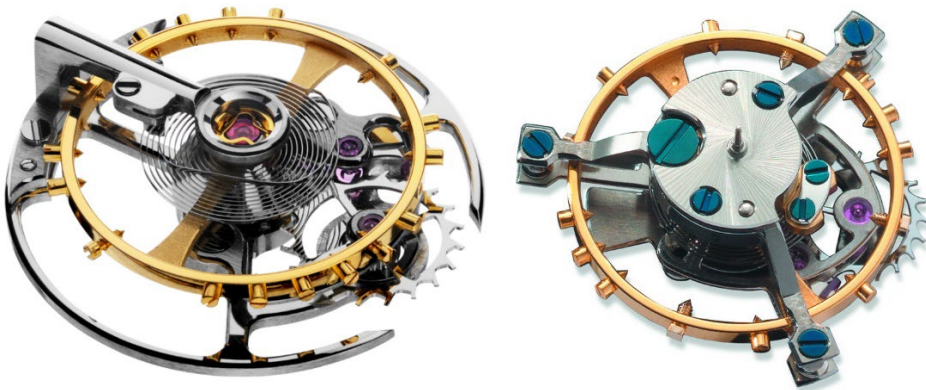
En réalité, les deux protections portent sur des éléments différents :

- Le brevet protège la fonction technique, indépendamment de sa forme
- Le droit d'auteur protège la forme, sans aucune considération pour la fonction réalisée.

Ce qui est parfois difficile à appréhender pour une personne n'ayant pas pratiqué le développement logiciel est que, même si la forme (les lignes de code écrites par le développeur) est effectivement contrainte par la fonction à réaliser, le développeur dispose très souvent d'une infinité de façons possibles pour réaliser cette fonction. L'expression de la réalisation portera donc bien souvent « l'empreinte » de son auteur, reflétant ses choix esthétiques personnels.

Nous pouvons, à titre illustratif, faire un parallèle avec le domaine de l'horlogerie. Les mécanismes d'horlogerie doivent réaliser certaines fonctions, et les procédés de réalisation de ces fonctions peuvent être brevetés. Cependant, l'horloger conserve, pour réaliser une même fonction, une infinie liberté dans ses choix esthétiques.

Figure 2 : Deux réalisations différentes d'une même fonction (le « tourbillon ») dans l'horlogerie



Sources : <https://www.hoursandminutes.co> (gauche), <https://www.revolution.watch> (droite)

Dans l'image ci-dessus, nous pouvons voir deux réalisations distinctes d'une même fonction (appelée « tourbillon »). Cette fonction est protégée par un brevet<sup>3</sup>. Il est indéniable que les deux réalisations de ce même procédé sont bien différentes, et que leur esthétique reflète les choix personnels des horlogers les ayant conçues. À ce titre, ces deux réalisations différentes d'un même brevet seront protégées par le droit d'auteur.

<sup>3</sup> Brevet obtenu en 1801 par Abraham-Louis Breguet

Il en est de même pour le logiciel, où la forme (l'expression) et la fonction peuvent être protégées par des droits de propriété intellectuelle différents.

De nombreuses études traitent des rapports entre logiciel open source et droit d'auteur<sup>4</sup>, et en conséquence nous faisons le choix de ne pas traiter cet aspect ici. Au contraire, le présent mémoire a pour objet principal l'articulation entre logiciel open source et brevet, ce sujet étant bien moins abordé dans la littérature.

## II - Le brevet

### A - Historique

Le brevet, que l'on peut définir de manière générale comme un monopole d'exploitation temporaire en échange d'une divulgation de l'invention, est un système assez ancien. Les premières formes, remontant à l'antiquité, semblent accorder les monopoles d'exploitation de façon discrétionnaire, en tant que récompenses.

Les historiens s'accordent sur le fait que c'est au début de la Renaissance que le brevet a pour la première fois été établi en tant que système, à Venise (Lapointe, 2000). Au XV<sup>ème</sup> siècle, Venise est une république en pleine expansion, haut lieu d'échanges commerciaux. C'est un lieu d'échanges cosmopolites, qui attire les inventeurs de toutes origines. Reconnaissant là un formidable moyen de développement culturel et économique, les vénitiens commencent à accorder des monopoles aux inventeurs, de manière discrétionnaire.

Cette pratique s'étant avérée bénéfique, elle est institutionalisée sous forme d'un statut promulgué en 1474 par le sénat vénitien. Le préambule de l'acte officiel déclare<sup>5</sup> :

*« Vivent parmi nous des hommes très talentueux, enclins à inventer et à découvrir des choses très ingénieuses qui, grâce aux vertus et à la splendeur de notre ville, affluent ici depuis différentes parties du monde. Par conséquent, s'il était prévu que les œuvres et les outils découverts par de telles personnes ne puissent être reproduits par d'autres, qui ne les priveraient pas de leurs mérites, d'autres hommes appliqueront alors leur ingéniosité par le biais de nouvelles découvertes qui contribueront à la construction d'outils d'une grande utilité et au bénéfice des utilisateurs et de l'état entier. »*

Il s'agit d'instituer un système vertueux, pour le bénéfice du peuple, qui permettra indirectement des retombées économiques pour l'Etat. Les éléments de notre brevet moderne sont présents dans le reste de la déclaration :

- Le procédé doit être nouveau<sup>6</sup>, inventif (« *ingegnosi* »), suffisamment perfectionné pour pouvoir être réalisé
- Le monopole est accordé pour une durée limitée (10 ans)
- L'action en contrefaçon est prévue

---

<sup>4</sup> Par exemple (de Laat, 2005), (Horne, 2001), (Kelty, 2004), (Kennedy, 2001), (Moglen, 1999)

<sup>5</sup> Traduction réalisée de manière assistée depuis le texte original en italien.

<sup>6</sup> La nouveauté était alors considérée uniquement sur le territoire de la république de Venise

Ce n'est qu'à la fin du XVIII<sup>ème</sup> siècle, avec l'arrivée de la première révolution industrielle, que le système de brevets s'est généralisé dans les différents pays. En peu de temps, les nations européennes et les États-Unis adoptent des lois instituant des systèmes de brevets, afin de favoriser l'essor de leur économie, et donc le bien de leurs citoyens. La volonté affichée est tant économique qu'humaniste, les deux étant considérés comme allant de pair. Ainsi, en 1787, les États-Unis incluent le principe du brevet dans leur constitution<sup>7</sup> :

*« The Congress shall have power [...] to promote the progress of science and useful arts, by securing for limited times to authors and inventors the exclusive right to their respective writings and discoveries »*

En France la loi du 7 janvier 1791 dispose :

*« Toute découverte ou nouvelle invention, dans tous les genres d'industrie, est la propriété de son auteur »*

Les systèmes de brevets de la première partie du XIX<sup>ème</sup> siècle sont cependant entachés d'un fort protectionnisme. Certains pays adoptent des lois interdisant l'export des inventions brevetées, d'autres mettent en place une déchéance du brevet national en cas de dépôt d'un brevet à l'étranger portant sur la même invention.

Petit à petit, les législations nationales évoluent, sur fond d'observations mutuelles des différentes législations, et de débats cherchant à déterminer si les brevets sont un frein ou un encouragement à l'innovation.

En 1883, des représentants de 140 pays se réunissent à Paris pour comparer les différents droits nationaux de propriété industrielle, et tenter d'établir des principes communs. La même année, 11 pays signent ce qui sera connu sous le nom de « convention de Paris », instituant entre autres un droit de priorité permettant d'étendre un dépôt national à d'autres pays. A ce jour, 177 états ont ratifié la version actuelle de cette convention.

## **B - Types de brevets**

Ce qui est appelé simplement « brevet » dans le contexte juridique français désigne en réalité un « brevet d'invention » se rapportant à une invention technique, dont les critères nécessaires sont la nouveauté, l'activité inventive, et la possibilité d'application industrielle.

Le certificat d'utilité, ne différant que dans sa forme (absence d'examen préalable et durée de protection plus courte) mais pas dans ses critères de validité, est considéré comme étant inclus sous le terme « brevets » dans la présente étude. Il est assez peu utilisé en France, mais certains pays ont un équivalent qui se trouve plus largement utilisé (en particulier la Chine, avec le modèle d'utilité)

Ce que nous entendons donc par « brevet » dans la présente étude est un titre de propriété industrielle, protégeant une invention technique, conférant à son détenteur le droit d'interdire l'usage de cette invention par des tiers, et ce pour une durée limitée.

D'autres type de titres de propriété industrielle accordés dans d'autres pays que la France sont souvent traduits de manière erronée par le même mot « brevet » (en particulier par les non-spécialistes), alors que leur

---

<sup>7</sup> Il faut noter ici que certains spécialistes, dont le juge Giles Rich, sont d'avis que l'intention initiale de concernait que le système de copyright et non pas le brevet, le mot « science » d'alors se référant au savoir. Le terme équivalent du mot « science » actuel était à l'époque « *natural philosophy* ». (Rich, 1994)

objet est différent, ce qui peut prêter à confusion dans le cadre du logiciel. Il s'agit en particulier des titres suivants :

- Le « design patent » américain. Ceci désigne un équivalent des dessins et modèles en droit français<sup>8</sup>.
- Le modèle d'utilité chinois, s'il se rapporte à la forme du produit. Il s'agit également d'un équivalent, du point de vue de son objet, aux dessins et modèles.

Ces titres, n'étant pas des brevets d'invention, ne sont pas pris en compte dans la présente étude, même s'ils peuvent être relatifs à certaines caractéristiques des logiciels (leur interface graphique).

## C - Le brevet, le logiciel et le brevet logiciel

*« Les brevets logiciels ne sont pas valables en Europe, donc les brevets n'y impactent pas les logiciels »*

Ceci est en substance, exprimé de manière volontairement simplifiée et caricaturale, l'avis qu'ont de nombreuses personnes. La réalité est bien plus complexe, et l'affirmation mentionnée ci-dessus est on ne peut plus fausse.

Il est exact que la Convention sur le brevet européen (CBE), dans son article 52, exclue du champ de la brevetabilité les demandes portant sur « les programmes d'ordinateur » « en tant que tels », et qu'une telle exception n'existe pas aux Etats-Unis.

### **Le brevet logiciel aux Etats-Unis**

Nous avons assisté jusqu'à récemment, aux Etats-Unis, à une certaine dérive sur la brevetabilité des inventions mises en œuvre par ordinateur. Un nombre croissant de brevets ont été accordés sur des méthodes abstraites, mises en œuvre par ordinateur<sup>9</sup>, et un certain nombre de pratiques commerciales douteuses se sont créées pour profiter de ce type de brevets<sup>10</sup>. En 2014, la Cour Suprême des Etats-Unis, par son arrêt *Alice v. CLS Bank* (SCOTUS, 2014), a mis un terme à ces pratiques en rappelant que les méthodes abstraites ne sont pas brevetables, la simple mise en œuvre par ordinateur ne suffisant pas à conférer l'effet technique requis pour la brevetabilité<sup>11</sup>. Cela ne signifie pas pour autant la fin des « brevets logiciels » aux Etats-Unis, mais simplement la fin des brevets où l'ajout de « mis en œuvre par ordinateur » semblait la seule différence par rapport à l'état de l'art (Tran, 2016). L'acceptabilité des demandes de brevet concernant les inventions mises en œuvre par ordinateur semble maintenant se rapprocher entre l'Europe et les Etats-Unis.

---

<sup>8</sup> La confusion est malheureusement grande auprès du grand public français, auquel la presse a expliqué pendant l'affaire *Apple v. Samsung* qu'Apple avait « breveté » un téléphone rectangulaire avec des coins arrondis.

<sup>9</sup> Exemple : Le brevet US 8'209'618 revendique un système de vote pour que plusieurs utilisateurs puissent choisir leur photo préférée au travers d'un réseau, sans qu'il n'y ait pour autant de précisions sur les méthodes techniques.

<sup>10</sup> Le plus souvent, une société se trouvait approchée par un détenteur de ce type de brevets, avec le choix de payer immédiatement quelques dizaines de milliers de dollars ou prendre le risque d'un procès.

<sup>11</sup> Le lecteur souhaitant approfondir les impacts de cet arrêt pourra se référer à une très ample littérature, par exemple Tran (2016)

## Les inventions mises en œuvre par ordinateur (IMO)

Bien que « les programmes d'ordinateur » « en tant que tels » soient exclus de la brevetabilité, cela ne signifie pas l'exclusion des IMO, admises à la brevetabilité par l'Office Européen des Brevets (OEB).

La première classe d'IMO est celle où l'ordinateur ne réalise qu'une partie des étapes de la revendication. Dans ce cas, il s'agit d'un système ou d'un procédé plus large que la partie réalisée par ordinateur, et ce type d'invention est admis à la brevetabilité. En effet, la nécessité de réaliser une partie du traitement par ordinateur n'est pas un critère d'exclusion.

La seconde classe d'IMO est celle où toutes les étapes sont réalisées par ordinateur. Dans ce cas, l'OEB considère que ces inventions sont brevetables si elles « produisent un effet technique ». Dans certains cas, l'effet technique est facile à percevoir. C'est par exemple le cas lorsque le traitement réalisé modifie un signal extérieur, ou active des actionneurs<sup>12</sup> ou des systèmes de commande. Dans d'autres cas, l'appréciation peut être plus délicate. Par exemple, l'OEB précise :

*[...] un programme d'ordinateur revendiqué en tant que tel n'est pas exclu de la brevetabilité s'il est capable de produire, lorsqu'il est mis en œuvre ou chargé sur un ordinateur, un effet technique supplémentaire allant au-delà des interactions physiques "normales" existant entre le programme (logiciel) et l'ordinateur (matériel) sur lequel il fonctionne.*

Il est donc tout à fait possible d'avoir des IMO qui soient admises par l'OEB, dans des cas assez divers de logiciels. On trouve par exemple :

- Une méthode pour réaliser des copier/coller (EP 1'577'769)
- Une méthode pour construire une requête de bases de données depuis une interface graphique (EP 0'575'358)
- Une méthode de manipulation d'une table dans un tableur (EP 1'643'387)
- Une méthode de gestion dynamique de la mémoire disponible (EP 0'940'755)
- Une méthode « implémentée par ordinateur » d'amélioration du contraste d'une image (EP 1'742'178)

*Note : On pourra également s'interroger sur ce qu'est un ordinateur. L'OEB considère par exemple le logiciel embarqué dans une imprimante comme une invention mise en œuvre par ordinateur. A l'heure où le logiciel embarqué dans une machine à laver ou une voiture peut être mis à jour, doit-on comprendre « programme d'ordinateur » comme tout logiciel, quel que soit la plate-forme d'exécution (Bouvier, 2018) ? Quid des microcontrôleurs, dont la frontière avec les microprocesseurs devient de plus en plus difficile à cerner, sont-ils eux aussi considérés comme des ordinateurs du point de vue des critères de brevetabilité ?*

*Il ne semble pas y avoir de frontières précises, et les limites évoluent par la jurisprudence au fur et à mesure des changements de la technique. Il semble que ces questionnements puissent continuer à alimenter de nombreux travaux de recherche.*

<sup>12</sup> Un actionneur est un élément agissant de manière physique, tel qu'un électroaimant, une pompe ou une électrovanne. Il s'oppose aux capteurs, qui reçoivent les informations physiques.

## **La mise en œuvre, par ordinateur, d'une invention**

Il conviendra de ne pas assimiler les IMO, considérées comme impliquant nécessairement un ordinateur, et le fait de mettre en œuvre par ordinateur une invention ne nécessitant pas un ordinateur.

L'amélioration permanente des capacités de traitement des ordinateurs permet en effet d'utiliser maintenant du logiciel pour réaliser des tâches qui étaient auparavant dévolues à du matériel. C'est par exemple le cas de nombreux traitements électroniques, parfois remplacés par du logiciel, afin de profiter des avantages que cela induit (réduction des coûts, possibilités de mises à jour...).

Les brevets associés à ces techniques ne sont généralement pas spécifiques aux logiciels. Implémenter par ordinateur l'enseignement d'un brevet n'évite aucunement la contrefaçon. De nombreux logiciels sont dans ce cas de figure, et pourront ainsi contrefaire des brevets qui ne sont ni des brevets logiciels, ni des brevets d'IMO.

### **La réalité du terrain**

Comme nous venons de l'exposer, les limites de la brevetabilité du logiciel sont complexes. Il peut donc arriver que certains brevets soient accordés alors que leur objet n'est pas brevetable. Les examinateurs des offices de brevets sont faillibles, et les limites en évolutions permanentes.

Il peut donc arriver que certains brevets dans le domaine du logiciel soient finalement annulés conséquemment à des assignations en contrefaçon<sup>13</sup>. Le défendeur aura tout de même à dépenser temps et argent dans la procédure, et ce risque doit être pris en compte.

## **III - Le logiciel open source**

### **A - Historique**

Le logiciel étant lié à l'informatique, il est forcément bien plus récent que le système de brevets. On peut considérer que c'est à partir des années 50 que le logiciel est apparu, avec la mise au point des premiers ordinateurs permettant d'exécuter un programme stocké en mémoire, remplaçant les programmations précédemment effectuées par le biais de câblages, d'interrupteurs ou par cartes perforées<sup>14</sup>.

L'avènement progressif des micro-ordinateurs, remplaçant les précédents gros systèmes (dits « *mainframes* »), combiné au développement des réseaux et systèmes d'échanges de données, a ensuite permis l'essor extrêmement rapide des logiciels (Jones, 2013).

Les tous premiers logiciels étaient bien souvent considérés comme de simples accessoires des ordinateurs, la valeur économique étant alors attribuée quasi exclusivement à la machine. Initialement, la situation juridique des logiciels était plutôt floue. Ce n'est par exemple qu'en 1974 que le logiciel a été reconnu aux Etats-Unis

---

<sup>13</sup> Par exemple *Free v. Orange*, où en 2015 le TGI de Paris a annulé certaines revendications du brevet détenu par Orange pour cause de non brevetabilité des programmes d'ordinateur en tant que tels.

<sup>14</sup> On considère que c'est en 1948 que le premier logiciel est apparu, programmé par Tom Kilburn sur le « Manchester Baby ».



comme pouvant être couvert par le copyright (Dunn, 1986) (Ballardini, 2012), à la condition qu'il soit doté d'originalité<sup>15</sup>.

Vers la fin des années 70, de nombreux logiciels étaient vendus (donc leur valeur économique était reconnue), et au début des années 80 ces logiciels furent de moins en moins accompagnés de leur code source<sup>16</sup>. Ils n'étaient alors plus distribués que sous leur forme compilée<sup>17</sup>, dédiée spécifiquement à un type d'ordinateur et de système d'exploitation.

Une personne en particulier s'est alors opposée à cette « privatisation » du logiciel : En 1983, Richard Stallman a annoncé le lancement du projet GNU<sup>18</sup>, dont le but était de créer un système d'exploitation entièrement « libre » pour remplacer les systèmes d'exploitation de type UNIX, dont chacun pourrait avoir accès au code source et ainsi le modifier<sup>19</sup>. Peu de temps après, il a mis en place la *Free Software Foundation* (FSF)<sup>20</sup>, dont le but est de promouvoir le projet GNU. Il s'agit ici clairement d'une action militante servant à atteindre un idéal que l'on peut qualifier de philosophique.

A la suite d'un différend relatif à l'utilisation du code source d'un logiciel, Richard Stallman a écrit en 1988/1989 la licence GPL (« *GNU Public License* »), une très astucieuse utilisation du copyright pour s'assurer de garder la possibilité de distribuer les logiciels soumis à cette licence. Pour l'aider à atteindre son objectif militant, Stallman a rendu cette licence virale, dans le sens où la combinaison d'un logiciel sous licence GPL avec un autre logiciel non GPL propage alors la licence GPL à cet autre logiciel. La création et la diffusion d'œuvres dérivées est ainsi explicitement autorisée, à la condition que l'œuvre dérivée soit couverte par la même licence. Ce mécanisme est parfois nommé copyleft, par opposition au copyright.

Au début des années 90, le système d'exploitation Linux<sup>21</sup> est apparu, sous licence GPL. La réussite technique de Linux a clairement accéléré le développement des logiciels alors dits « libres ». Petit à petit, le modèle de développement dit « collaboratif », c'est-à-dire dans lequel le développement d'un même logiciel est

---

<sup>15</sup> Dans notre système de droit d'auteur, cette condition d'originalité est également requise. Le logiciel n'est ainsi couvert par le droit d'auteur que s'il porte l'empreinte de son auteur. C'est le cas la plupart du temps, mais n'est pas systématique. Par exemple, la Cour d'appel de Paris, dans son arrêt 13/24577 du 24 Novembre 2015, a statué sur le fait qu'un logiciel ne faisant pas preuve d'originalité ne soit pas couvert par le droit d'auteur, quand bien même il dispose d'un auteur identifié et reconnu.

<sup>16</sup> Le code source est l'expression originale du logiciel qui est alors écrit et mis en forme par son auteur, dans un langage d'expression qualifié de « langage de programmation ». Bien que l'un des buts de cette expression par l'auteur soit technique (pour réaliser la fonction voulue), l'écriture sous forme de code source comporte des possibilités de variations infinies, plus ou moins élaborées, permettant de porter l'empreinte de l'auteur. La très grande majorité des professionnels du logiciel s'accordent même pour reconnaître que l'auteur peut atteindre différents niveaux d'élégance dans cette expression, de manière similaire à une œuvre littéraire (Coleman, 2013).

<sup>17</sup> La forme compilée est le résultat de la transformation (compilation) du code source (langage destiné à être lu par des humains) en langage machine (destiné à être lu et exécuté par l'ordinateur).

<sup>18</sup> Le terme « GNU » est un acronyme dit « récursif » dont les initiales signifient *GNU's Not UNIX*.

<sup>19</sup> Dans ce contexte, il reste tout de même le problème de compétence pour être à même de modifier ou d'adapter un logiciel. Avoir accès au code source donne les moyens de modifier le logiciel, mais il n'empêche que le logiciel peut parfois être extrêmement complexe.

<sup>20</sup> « *Free* » en anglais peut se traduire en français par deux termes distincts : libre et gratuit. Dans le présent contexte, il faut comprendre « *free* » comme « libre ».

<sup>21</sup> Initialement l'œuvre de Linus Torvald.

réalisé par plusieurs personnes situées à des endroits différents, avec très peu d'encadrement et de planification, a montré son efficacité et surtout son efficience.

A la fin des années 90, nous avons assisté à une professionnalisation d'une partie de ce mouvement. L'organisation *Open Source Initiative* (OSI) a été créée par Bruce Perens et Eric Raymond, pour promouvoir ce modèle de développement, non pas pour son aspect idéaliste (but), mais pour son aspect efficient (moyen) dans le contexte du développement de logiciels.

Petit à petit, les entreprises du logiciel ont adopté ou intégré, de manière plus ou moins rapide, le modèle dit « open source ». Aujourd'hui de très nombreuses entreprises du logiciel développent ou contribuent au développement de logiciels selon ce modèle et quasiment tout un chacun utilise des logiciels open source, le plus souvent sans le savoir.

*Note : Le lecteur intéressé par l'histoire et l'étude sociologique des logiciels libres et open source pourra se référer à de nombreux ouvrages, par exemple Raymond (2001) ou Kirk (2007).*

*Licences Creative Commons :*

*Les licences Creative Commons, dont les premières sont apparues en 2002, reprennent des principes similaires aux licences open source, permettant la libre réutilisation (avec citation de l'origine) et parfois la modification de l'œuvre.*

*Les licences Creative Commons ne sont pas destinées aux logiciels, mais aux œuvres littéraires et artistiques. Les logiciels open source ne sont donc normalement pas licenciés sous licences Creative Commons.*

*Il est cependant indéniable que la technique contractuelle utilisée par les licences des logiciels open source a été une très forte source d'inspiration pour la création des licences Creative Commons (Lessig, 2001)*

## **B - Différence entre « open source » et « logiciel libre »**

Comme nous l'avons vu, le logiciel libre poursuit un but idéaliste, alors que le logiciel de type « open source » a pour but la réalisation de ce même logiciel via un moyen considéré comme efficient, ce qui est une considération pragmatique.

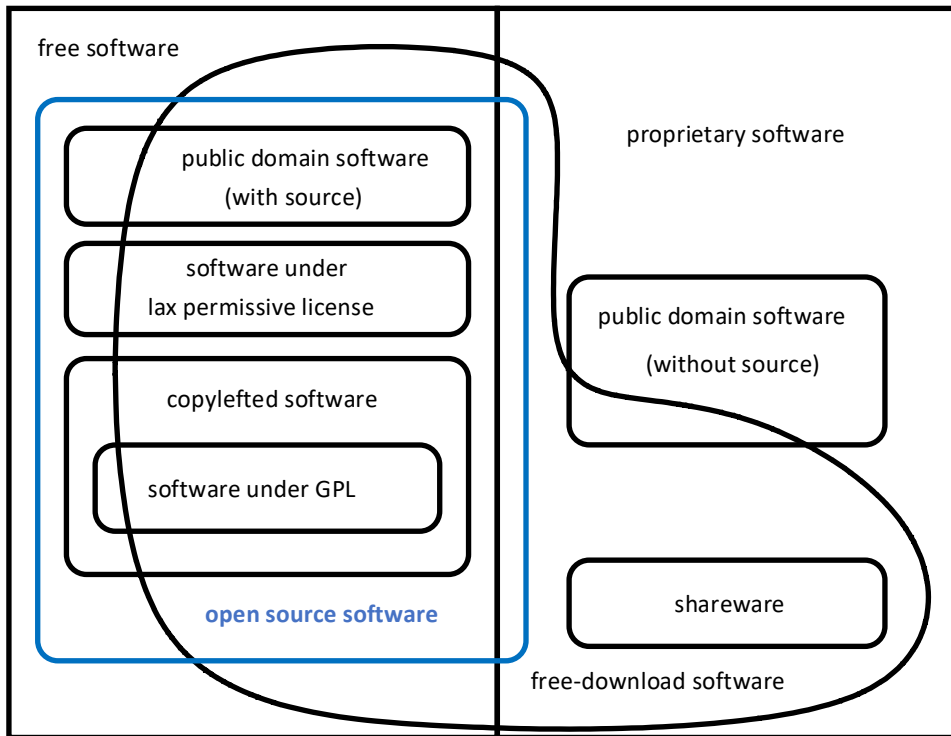
Ainsi, alors qu'il peut y avoir opposition entre les buts recherchés, les moyens employés (accès au code source, possibilité donnée aux tiers d'apporter des modifications) sont les mêmes. Cette commonalité de moyens peut engendrer une confusion pour les personnes extérieures au domaine, les conduisant parfois à faire l'amalgame entre les deux mouvances. Il existe même un acronyme, FOSS<sup>22</sup> (*Free and Open Source Software*), permettant de désigner l'union des deux mouvances.

---

<sup>22</sup> L'acronyme est même parfois FLOSS (*Free/Libre Open Source Software*) lorsqu'il y a volonté d'éviter l'ambiguïté sur le terme « *free* », en portant l'accent sur le but idéaliste.

Dans la présente étude, ce sont les problématiques particulières liées à la divulgation et à la diffusion du code source qui nous intéressent, qui sont des caractéristiques du logiciel open source. Le but sociétal pouvant être recherché dans le cas du logiciel libre n'est pas considéré comme un critère dans cette étude, sauf par les conséquences qu'il entraîne indirectement quant à l'interaction entre le logiciel et les brevets<sup>23</sup>.

Figure 3 : Place de l'open source dans l'écosystème du logiciel



Adapté depuis Chao-Kuei et al. / Wikipedia

Il est cependant indéniable qu'il existe un fort recoupement entre ces deux mouvances : de très nombreux logiciels développés selon un mode open source sont distribués selon des licences dites « libres » (que leurs auteurs poursuivent ou non un but idéaliste), et la proportion de logiciels propriétaires disponibles en open source est actuellement très faible.

## C - Les motivations derrière l'open source

Du côté des contributeurs<sup>24</sup> de projets open source, les motivations peuvent être diverses, par exemple :

- La poursuite d'un idéal moral. Comme indiqué dans les paragraphes précédents, c'est en particulier le cas pour les logiciels revendiqués comme libres.

<sup>23</sup> Par exemple, certaines licences logicielles peuvent être rédigées de façon à aider l'atteinte d'un but sociétal.

<sup>24</sup> Les contributeurs sont les personnes réalisant une partie d'un logiciel Open Source. Tous les participants ne sont pas forcément contributeurs de code, ils peuvent également réaliser d'autres activités comme du support technique, de la gestion de projets, ...

- Le souhait de participer à une réalisation technique, en poursuivant un but d'amélioration qualitative.
- Le souhait d'apprendre et de participer à la diffusion de la connaissance. Le code source du logiciel étant redistribué, il est une source de connaissances qui profite au contributeur, ce même contributeur participant également par sa propre contribution à l'extension de cette connaissance mise à la disposition de tous.
- La volonté de reconnaissance. Bien que le logiciel open source puisse être modifié par des tiers, le nom d'un contributeur y reste attaché. Il y a en général un très fort respect de cet aspect moral par la communauté open source<sup>25</sup>.

Les motivations des entreprises sont bien sûr différentes, et le développement open source y est vu soit comme étant le cœur du business model, soit comme étant un moyen de supporter efficacement les activités stratégiques de l'entreprise (Jullien et Zimmermann, 2013). On pourra citer les moyens d'utilisation de l'open source suivants :

- Réduction des coûts de développement : C'est en général le point qui vient tout d'abord à l'esprit. L'ouverture du code source et l'acceptation de contributions extérieures permet de réduire le coût de développement, notamment en ayant une plus grande vélocité de développement.
- La contribution aux standards : De nombreux organismes de normalisation, en particulier dans le domaine des télécommunications, incluent le développement d'un logiciel de référence, dont les sources sont ouvertes, pendant le processus d'élaboration du standard. Se positionner à ce moment est extrêmement intéressant pour les entreprises, qui s'assurent alors de disposer d'une base de code servant de référence pour le standard. Les autres entreprises doivent ensuite s'adapter à ce standard, il s'agit ainsi d'un levier stratégique très important.
- La diffusion d'une version gratuite et open source, en parallèle d'une version payante plus élaborée : La version open source sert à diffuser et faire connaître le logiciel, et l'entreprise bénéficie de contributions extérieures sur cette base. En parallèle, une version plus élaborée ou des modules optionnels sont vendus par l'entreprise, et les clients potentiels ont déjà pu évaluer la version « de base ».
- La vente de services : Le logiciel est pleinement open source, mais l'entreprise à l'origine du logiciel (ou y contribuant fortement) devient reconnue comme spécialiste de ce logiciel. Les consultants de cette entreprise sont alors considérés comme ayant une grande valeur par les autres entreprises ayant besoin de support technique ou d'adaptations de ce logiciel.
- La vente de licence : Dans ce cas, le logiciel est diffusé sous forme open source avec une licence ne permettant pas tous les cas d'usage commerciaux. Une entreprise peut évaluer le logiciel, mais ne peut pas l'exploiter dans son cas d'usage commercial, sauf à acquérir une autre licence (permettant l'exploitation commerciale) auprès des créateurs du logiciel. Dans cette approche dite « *dual licensing* », le logiciel est disponible simultanément sous plusieurs licences (l'une étant commerciale et l'autre non).

Il y a ainsi de nombreux moyens d'utiliser l'open source au cœur de la stratégie d'une entreprise du domaine technologique. Le brevet fait également partie des outils stratégiques pour de telles entreprises. Aussi,

---

<sup>25</sup> Ce point semble plus se rapprocher de l'axe principal de notre droit d'auteur qui est centré sur le droit moral, alors que le *copyright* de la *common law* semble plus centré sur des aspects économiques (Benhamou et Farchy, 2007).

nous allons rapidement nous retrouver face au problème de la combinaison de ces deux outils, qui semblent à priori peu compatibles entre eux, pour plusieurs raisons :

- L'open source se base sur un droit d'inclusion, alors que le brevet se base sur un droit d'exclusion.
- Le brevet nécessite une non-divulgation de l'invention avant son dépôt, alors que dans le modèle open source la divulgation du code source est effectuée le plus tôt possible.

De plus, il est indéniable que le brevet, permettant l'exclusion, est peu compatible avec l'idéal des partisans du logiciel libre, ceux-ci étant souvent farouchement opposés au système de brevets.

Nous nous pencherons dans cette étude sur les points liés à l'interaction de ces deux outils. Dans une première partie, nous traiterons des points particuliers liés à la contrefaçon d'un brevet par un logiciel open source. Dans la seconde partie, nous aborderons les moyens permettant de mettre en place une protection par brevet dans une stratégie de développement open source.

# PREMIERE PARTIE :

## LA CONTREFAÇON DE BREVET PAR LE LOGICIEL OPEN SOURCE

### I - Contrefaçon de brevet - rappels

#### A - Limites géographiques

Le brevet étant limité géographiquement (au pays où il est déposé), il en est de même pour la contrefaçon. Or, le logiciel open source est généralement mis à disposition sans considération de zone géographique de diffusion, depuis des serveurs répartis dans le monde.

Il n'y aura donc pas un seul acte de contrefaçon global, mais au contraire autant d'actes de contrefaçon que de zones géographiques où le brevet est déposé. Comme il est fréquent que la portée des brevets issus d'une même famille soit différente en fonction des pays où le brevet a été accepté, la potentielle contrefaçon devra être appréciée en fonction des revendications déposées dans chaque zone géographique.

#### B - La contrefaçon de brevet en droit français.

En droit français<sup>26</sup>, l'article L613-3 du Code la propriété intellectuelle définit les actes constitutifs de contrefaçon de brevet :

*Sont interdites, à défaut de consentement du propriétaire du brevet :*

*a) La fabrication, l'offre, la mise dans le commerce, l'utilisation, l'importation, l'exportation, le transbordement, ou la détention aux fins précitées du produit objet du brevet ;*

*b) L'utilisation d'un procédé objet du brevet ou, lorsque le tiers sait ou lorsque les circonstances rendent évident que l'utilisation du procédé est interdite sans le consentement du propriétaire du brevet, l'offre de son utilisation sur le territoire français ;*

*c) L'offre, la mise dans le commerce, l'utilisation, l'importation, l'exportation, le transbordement ou la détention aux fins précitées du produit obtenu directement par le procédé objet du brevet.*

Si la liste de ces interdictions est bien adaptée pour les éléments « matériels », ces règles deviennent beaucoup moins évidentes pour du logiciel, dématérialisé et potentiellement multiforme.

---

<sup>26</sup> Les actes constitutifs de contrefaçon sont généralement similaires dans les autres pays.

## C - Contrefaçon littérale et fourniture de moyens

La contrefaçon n'est pas forcément effectuée uniquement lorsque l'élément contrefaisant réalise l'intégralité de la revendication d'un brevet. Ce premier cas, lorsque tout l'enseignement de la revendication est mis en œuvre constitue bien sûr une contrefaçon, dénommée contrefaçon littérale.

Le second type de contrefaçon s'effectue par fourniture de moyens. Elle est définie par l'article L643-4(1) du Code la propriété intellectuelle :

*Est également interdite, à défaut de consentement du propriétaire du brevet, la livraison ou l'offre de livraison, sur le territoire français, à une personne autre que celles habilitées à exploiter l'invention brevetée, des moyens de mise en œuvre, sur ce territoire, de cette invention se rapportant à un élément essentiel de celle-ci, lorsque le tiers sait ou lorsque les circonstances rendent évident que ces moyens sont aptes et destinés à cette mise en œuvre.*

Un logiciel qui permettrait (« apte ») et aurait pour but (« destiné ») de contribuer à réaliser un procédé protégé par brevet serait contrefaisant, au titre de la fourniture de moyens, quand bien même il ne réaliserait pas l'intégralité de la revendication, d'autres parties étant réalisées par d'autres moyens que ce logiciel.

Etant ainsi établi qu'un logiciel peut tout à fait se trouver être contrefaisant d'un brevet (y compris en Europe), par différents moyens, le logiciel open source n'échappe pas à cette possibilité. Toutefois, il existe quelques particularités à prendre en compte dans le cadre du logiciel open source.

## II - Points particuliers liés à l'open source

### A - Détection aisée de contrefaçon

La principale particularité du logiciel open source qui vient à l'esprit est que, comme l'indique son qualificatif, son code source est ouvert à tous.

Cela simplifie nettement la détection de la contrefaçon : toutes les étapes de traitement étant lisibles, la comparaison avec les revendications du brevet sera directement réalisable. Dans le cas où plusieurs méthodes alternatives existent pour obtenir le même effet technique que dans la revendication du brevet, il n'y aura pas de doute, dans le cas du logiciel open source, quant à la méthode employée. Le détenteur du brevet aura alors la certitude que le logiciel incriminé enfonce ou non son brevet<sup>27</sup>.

Il en est de même pour la démonstration de la contrefaçon : que ce soit lors d'une action en contrefaçon ou lors de discussions préliminaires, il sera aisé pour le détenteur de démontrer la contrefaçon, celui-ci ayant alors la possibilité d'établir une correspondance entre les portions de code source et les étapes de la revendication contrefaite.

---

<sup>27</sup> Plus exactement, il s'agit d'une conviction. La certitude sera apportée par la décision de justice.

## **B - Traçabilité des contributions**

Les logiciels open source sont le plus souvent réalisés par plusieurs personnes, agissant pour leur propre compte ou pour le compte de leur employeur. Pour certains logiciels, il peut y avoir plusieurs dizaines de contributeurs<sup>28</sup>.

Afin de pouvoir gérer la traçabilité et l'organisation de ces contributions, la plupart des projets open source s'appuient sur un système de gestion de versions<sup>29</sup> qui permet d'obtenir et de retracer l'état du logiciel à n'importe quelle version. Ces systèmes de gestion de versions disposent quasiment toujours d'une interface publiquement consultable, qui permet d'identifier le contributeur de chaque ligne de code.

Il sera donc aisé, même dans le cas où le logiciel n'est pas proposé par une seule personne/société, d'identifier qui (ou quelle société) est à l'origine de l'insertion du code contrefacteur dans le logiciel, quand bien même le logiciel n'est pas explicitement mis sur le marché par un acteur donné.

## **C - Le cadre privé et l'exception de recherche**

Le Code de la propriété intellectuelle, dans son article L613-5, restreint les droits conférés aux brevets dans plusieurs situations<sup>30</sup>. Deux de celles-ci sont particulièrement importantes dans le cas du logiciel open source :

*Les droits conférés par le brevet ne s'étendent pas :*

*a) Aux actes accomplis dans un cadre privé et à des fins non commerciales ;*

*b) Aux actes accomplis à titre expérimental qui portent sur l'objet de l'invention brevetée ;*

Dans le cadre privé, il y a exemption à la contrefaçon. L'utilisateur (individuel) de logiciel open source ne peut donc être inquiété.

Les actes accomplis à titre expérimental correspondent à ce que l'on nomme parfois « exception académique », ou « exception de recherche ». Considérant que le processus de développement est un acte expérimental, le développeur de logiciel se trouve également exempt de contrefaçon, le droit des brevets ne peut empêcher le développement du logiciel.

La plupart des logiciels open source ont un développement réalisé de manière publique, à la vue et à l'accès de tous : il n'y a pas besoin d'étape supplémentaire pour que l'objet du développement soit accessible à tous. Ceci fait partie du fonctionnement intrinsèque de ce type de développement, où chacun doit pouvoir participer.

Il semble donc qu'il puisse y avoir, dans le cas du logiciel open source, un passage direct entre la création et l'utilisateur individuel, sans qu'il n'y ait contrefaçon selon le droit français.

Cela ne sera pas forcément le cas dans d'autres législations : cette exception de recherche découle de la volonté de permettre le travail sur les médicaments génériques avant l'expiration des brevets, pour qu'ils

---

<sup>28</sup> Sur le plan du droit d'auteur, il s'agit donc d'œuvres collectives.

<sup>29</sup> Les principaux systèmes de gestion de versions utilisés par les projets Open Source sont CVS, SVN et Git (CVS ne restant utilisé que pour les anciens projets)

<sup>30</sup> Les législations étrangères ont le plus souvent des restrictions similaires, mais non identiques



puissent être directement mis sur le marché à l'expiration des brevets<sup>31</sup>. De nombreux pays limitent ainsi cette exception de recherche aux médicaments ou aux travaux réalisés dans le but d'obtenir une autorisation de mise sur le marché pour les médicaments (Tridico, Jacobstein, Wall, 2014). Le législateur français n'ayant pas apporté de précisions au cadre de cette exemption a ainsi, sans le vouloir, favorisé le développement du logiciel open source, ainsi que l'une de ses formes de diffusions.

## D - Différents cas de distribution

Comme nous venons de le voir, le logiciel open source présente certains modes de distributions qui n'existent pas (ou généralement pas) dans le cadre du logiciel « classique ». Un logiciel est classiquement distribué sous forme compilée<sup>32</sup>, soit par un distributeur/revendeur, soit directement par l'éditeur<sup>33</sup>. Les logiciels open source peuvent être distribués/fournis/accessibles de manière similaire, mais peuvent également être obtenus selon des modes qui n'existent pas dans le cas des logiciels classiques.

### 1. Les acteurs de la distribution

Trois cas de figure sont à envisager pour la distribution du logiciel :

Le premier cas est celui correspondant à la distribution des logiciels classiques. Le logiciel open source est distribué ou mis à disposition, sous forme compilée, soit directement par son éditeur, soit par un distributeur<sup>34</sup>. S'il y a contrefaçon, il s'agit ici d'une contrefaçon par offre ou mise dans le commerce (en plus d'une potentielle contrefaçon par fabrication).

Le second cas est celui de la distribution d'un ensemble de plusieurs logiciels open source, comme dans le cas des « distributions<sup>35</sup> » Linux. De manière identique au cas précédent, il s'agirait d'une contrefaçon par offre ou mise dans le commerce.

Le troisième cas est celui de la mise à disposition par le biais d'une plate-forme faisant office d'intermédiaire technique (les forges logicielles). Ce cas est très courant dans le cas du logiciel open source, quelques plateformes très populaires fournissent toute l'infrastructure nécessaire au développement coopératif (principalement les dépôts<sup>36</sup> de sources) et à la distribution (y compris les miroirs<sup>37</sup>). La mise à disposition du logiciel contrefaisant par la plateforme relève de la contrefaçon (offre). Le fournisseur de cette plateforme pourrait argumenter de son rôle d'intermédiaire technique, n'étant pas alors dans l'obligation de vérifier les contenus placés sur cette plateforme (de manière similaire aux cas de contrefaçon d'œuvres littéraires et artistiques), et se contentant de supprimer l'accès aux contenus après avoir reçu une demande de retrait.

---

<sup>31</sup> À la suite de l'affaire Roche Products, Inc. v. Bolar Pharmaceutical Co., jugée en 1984 aux Etats-Unis

<sup>32</sup> Cf. définitions en annexe 1.

<sup>33</sup> Dans le cas de logiciels réalisés intégralement par une seule personne, l'auteur peut aussi tenir le rôle d'éditeur.

<sup>34</sup> Bien qu'il soit tout à fait possible de vendre un logiciel Open Source, il est rare que le distributeur soit un revendeur du logiciel. Ce qui est potentiellement vendu par le distributeur est le plus souvent une offre de service associée au logiciel, le logiciel étant fourni à titre gratuit.

<sup>35</sup> Cf. définitions en annexe 1.

<sup>36</sup> Cf. définitions en annexe 1.

<sup>37</sup> Cf. définitions en annexe 1.

## 2. Les formes de distribution

En plus de la distribution sous forme compilée, habituellement rencontrée pour les logiciels dont le code source n'est pas divulgué, le logiciel open source peut également être distribué sous différentes formes additionnelles.

### Code source

La première est la distribution sous forme de code source. Dans ce cas, c'est à l'utilisateur de compiler le code source afin d'obtenir une version exécutable correspondant au type de processeur et au système d'exploitation de l'ordinateur cible. Il s'agit du mode de distribution original des logiciels open source, pour différentes raisons :

- Cela évite à l'éditeur/auteur d'avoir à se soucier des nombreuses combinaisons d'architecture et de système d'exploitation, et de créer une version exécutable pour chacune de ces combinaisons.
- Ceci permet à l'utilisateur de s'assurer de ce qui est dans le logiciel, et ainsi de réduire les risques quant à la sécurité informatique : il peut en effet inspecter le code source, alors que l'inspection d'un binaire est une tâche qui, d'un point de vue pratique, est difficilement réalisable.
- Le code source prend bien moins de place que la version compilée (il est en général plusieurs milliers de fois plus petit), ce qui était particulièrement important lorsque les réseaux de communication n'avaient pas la vitesse actuelle.

Il convient de s'interroger sur le fait qu'un code source puisse ou non être une contrefaçon de brevet. En effet, le code source nécessite d'être transformé (compilé) pour pouvoir être exécuté par un ordinateur. Sans compilation, il n'est qu'une description textuelle d'algorithmes et de procédés, exprimés selon un certain formalisme (le langage de programmation). Ce formalisme est certes strict, mais ni plus ni moins qu'un formalisme mathématique. Il est admis que le code source relève de la propriété littéraire et artistique (vu qu'il peut être protégé par le droit d'auteur), ce qui peut être un argument fort pour considérer que la libre circulation du code source ne puisse être freinée par un brevet<sup>38</sup>.

On pourrait être tenté d'argumenter que le code source puisse constituer une contrefaçon par fourniture de moyen essentiel. Cependant, bien qu'il soit potentiellement destiné à réaliser une revendication d'un brevet, il n'est pas vraiment apte à le faire tant qu'il n'a pas été compilé.

La reconnaissance de la possibilité de contrefaçon par un code source pourrait entraîner la reconnaissance d'un potentiel de contrefaçon par d'autres descriptions textuelles. Il en est ainsi des algorithmes décrits dans les manuels techniques, les livres ou les thèses universitaires, et même des textes des standards techniques édités par les organismes de normalisation, qui contiennent très souvent une description algorithmique très détaillée, similaire à celle d'un code source<sup>39</sup>.

---

<sup>38</sup> La revendication du code source comme forme d'expression artistique est particulièrement important auprès de certains développeurs. Ce point est abordé du point de vue anthropologique par Coleman (2013, chapitre 5).

<sup>39</sup> Il serait même techniquement possible de créer un compilateur transformant le formalisme algorithmique de certains standards en un logiciel exécutable par un ordinateur.

*Note : On peut établir ici un parallèle avec le domaine de l'impression 3d, où la fourniture de fichiers de données ne constitue pas une contrefaçon, alors que l'impression réalisée à partir de ces fichiers peut contrefaire un modèle et enfreindre les droits de propriété intellectuelle associés (Le Goffic, 2015).*

A notre connaissance, argumenter qu'un code source n'est pas une contrefaçon mais plutôt une description écrite n'a pas encore fait l'objet d'une décision de justice.

Nous pouvons cependant constater que plusieurs projets open source ont choisi de ne fournir que du code source (et non pas également des versions compilées) sur la base de cet argument<sup>40</sup>, sans avoir été inquiétés sur ce point par des détenteurs de brevets<sup>41</sup> (qui sont par ailleurs parfaitement au fait de l'existence de ces projets). A l'inverse, les mêmes brevetés ont demandé des redevances aux projets open source fournissant des encodeurs sous forme de logiciels compilés<sup>42</sup>.

Bien que le fait qu'un logiciel soit disponible sous forme de code source (et donc nécessite d'être compilé, puis installé) constitue un obstacle important à son utilisation pour la grande majorité du public, il faut remarquer que cela n'est pas le cas pour les utilisateurs de systèmes d'exploitation de type Linux (ces systèmes étant quasiment toujours fournis et installés avec les outils nécessaires à la compilation). Ainsi, il faudra uniquement exécuter quelques commandes<sup>43</sup> bien connues des utilisateurs de ces systèmes pour obtenir un logiciel prêt à l'emploi. Pour ces habitués la barrière entre code source et logiciel utilisable est donc particulièrement faible.

*Note : L'auteur de ce mémoire a personnellement appliqué pendant de nombreuses années une telle stratégie de diffusion de logiciel sous forme de code source, principalement au travers de sa participation au projet d'encodeur MP3 open source nommé « LAME ».*

*Le projet, ne détenant aucun des brevets sur les procédés d'encodage, a toujours soigneusement évité de diffuser des versions compilées de l'encodeur, afin d'éviter la contrefaçon<sup>44</sup>.*

*Les éditeurs de logiciels souhaitant inclure cet encodeur MP3 dans leurs propres logiciels obtenaient des licences de brevets auprès des brevetés. (Les brevets pertinents sont maintenant expirés)*

## **Code interprété**

Une autre forme de distribution de logiciel open source correspond aux logiciels écrits dans un langage interprété, tels que le Javascript ou le Perl. Dans le cas des langages interprétés, le code source est directement interprété par logiciel spécifique dit « interpréteur », sans qu'il n'y ait besoin d'étape de compilation. Ces logiciels sont donc forcément distribués sous forme de code source, puisqu'il n'existe pas de forme binaire.

Le code source est donc directement prêt à être utilisé pour réaliser les fonctionnalités programmées. Si l'on considère le code source comme une description, le fait d'utiliser le logiciel est donc l'étape transformant

---

<sup>40</sup> Par exemple FFMPEG, x264 ou LAME, projets implémentant des encodeurs et décodeurs audio/vidéo.

<sup>41</sup> Ceci est uniquement une constatation de certaines pratiques, et ne permet pas de conclure sur la potentielle contrefaçon ou la non contrefaçon d'une telle pratique.

<sup>42</sup> Voir l'article "MP3 owners get stroppy with open source coders", The Register, 2001 : <https://bit.ly/2LGGWah>

<sup>43</sup> Ces commandes sont quasiment toujours les trois mêmes, quel que soit le logiciel :

```
./configure; make; make install
```

<sup>44</sup> Ceci est une constatation des pratiques rencontrées, et ne constitue pas opinion juridique.

cette description en réalisation. Dans l'hypothèse où le code source n'est pas considéré comme une contrefaçon potentielle, alors seul l'utilisateur final réaliserait le procédé contrefaisant.

Originellement, les langages interprétés étaient considérés comme lents, et non appropriés aux tâches nécessitant une forte puissance de calcul. L'augmentation de la puissance des ordinateurs, combinée avec l'amélioration de l'efficacité des interpréteurs, rendent maintenant ces langages aptes à être utilisés pour de nombreux traitements. De fait, des sites internet peuvent maintenant tout à fait, via leurs pages web, déclencher des traitements « lourds » réalisés dans le navigateur du visiteur, par le biais de code Javascript. Il est aujourd'hui par exemple possible de décompresser des flux audio ou vidéo totalement en Javascript<sup>45</sup>. Dans le cas du particulier, ce type de traitement ne peut être considéré comme une contrefaçon, car relevant de l'utilisation dans le cadre de la sphère privée, alors même que le décodage de ces formats est couvert par de nombreux brevets.

Dans l'hypothèse où l'on considérerait le code source comme n'étant pas une contrefaçon (mais une description), un tel traitement réalisé dans le navigateur de l'utilisateur individuel permettrait donc d'avoir un système évitant de tomber dans le cadre de la contrefaçon. Si, a contrario, l'on considère que le code source est une potentielle contrefaçon, le code javascript pourrait tout à fait être placé sur un serveur situé dans un pays où les brevets n'ont pas été déposés, et il n'y aurait pas non plus contrefaçon.

Le fait de pouvoir potentiellement ainsi faire usage du procédé breveté sans qu'il n'y ait pour autant contrefaçon est bien évidemment problématique pour les intérêts des brevetés, qui se trouvent alors privés de redevances potentielles, sans pour autant être fondés à tenter une action en justice.

### III - Comment agir en cas de contrefaçon

Les affaires liées à la contrefaçon par les logiciels open source peuvent parfois être médiatiques comme lorsqu'en 2007 Microsoft avait publiquement indiqué que le système d'exploitation Linux et les logiciels relatifs enfreindraient 235 de ses titres de propriété industrielle<sup>46</sup>. Il s'agissait alors probablement d'une annonce faisant partie d'une stratégie de communication, en accord avec le positionnement qu'avait à cette époque Microsoft face au mouvement open source.

Dans la plupart des cas, le breveté souhaitera ne pas se contenter d'effets d'annonce, et évaluer ses possibilités d'actions en cas de contrefaçon par un logiciel open source.

Tout d'abord, il convient de rappeler que dans le cadre d'une action en contrefaçon, la charge de la preuve incombe presque toujours au breveté<sup>47</sup>. Une contrefaçon potentielle par un logiciel open source facilite grandement la démonstration de la contrefaçon, le code source du logiciel open source étant public, il décrit sans

---

<sup>45</sup> La bibliothèque libde265.js permet le décodage en Javascript de flux vidéo codés au format HEVC, considéré comme très gourmand en ressources de calcul. Une démonstration est disponible à l'adresse <http://strukturag.github.io/libde265.js/>

<sup>46</sup> Le décompte était alors de 180 brevets (utility patents) et 60 dessins et modèles (design patents) (voir <https://cnn.it/30wWdxL>). La presse américaine a tendance à agréger ces deux catégories sous le terme « patents », en conséquence de quoi la presse française traduit souvent ceci en « brevets », qu'il s'agisse de brevets ou de dessins et modèles.

<sup>47</sup> Sauf potentiellement dans le cadre de brevets de procédés d'obtention de produits, ou le tribunal peut décider de renverser la charge de la preuve. Ce cas est peu probable dans le cadre d'un logiciel.

ambiguïté les traitements effectués par le logiciel. La difficulté peut cependant potentiellement résider dans la preuve de l'utilisation d'un logiciel donné, voire d'une version particulière de ce logiciel (si les traitements varient d'une version à l'autre).

En matière de brevet, le breveté n'a pas obligation d'agir en cas de suspicion de contrefaçon. Il faut cependant garder à l'esprit que pour obtenir une indemnité à la suite d'une action en contrefaçon, il faut qu'il y ait un préjudice. Si le breveté n'exploite pas l'invention, et qu'il est notoire qu'il ne demande pas de redevances, une éventuelle action en contrefaçon risque d'être peu productive. Le breveté sera donc avisé de n'agir que s'il peut démontrer un préjudice.

Nous allons maintenant voir à l'encontre de quelle entité l'on pourrait envisager une action en contrefaçon. Qui assigner en justice dépend des cas de contrefaçon, et de choix stratégiques.

## **A - Agir à l'encontre de l'entité utilisatrice**

Il est tout d'abord possible d'agir à l'encontre d'une entité utilisatrice du logiciel contrefaisant<sup>48</sup>. Au contraire des logiciels classiques qui sont souvent fournis avec des clauses de garantie, les logiciels open source sont le plus souvent fournis sans garantie. C'est par exemple le cas de la licence GPLv2, dont voici la section 11<sup>49</sup>:

*11. Because the program is licensed free of charge, there is no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide the program "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.*

Sauf à avoir souscrit à des garanties additionnelles, l'entité utilisatrice d'un logiciel open source contrefaisant ne pourra appeler son fournisseur en garantie. Elle disposera donc potentiellement de moins de moyens pour assurer sa défense. Il est cependant possible, voir probable, qu'elle obtienne une assistance juridique de la part d'entités telles que la *Free Software Foundation* (FSF) ou de l'*Electronic Freedom Foundation* (EFF), coutumières de la défense des logiciels open source.

## **B - Agir à l'encontre du distributeur**

La seconde possibilité est d'agir à l'encontre du distributeur, lorsqu'il y en a un. Le contexte le plus fréquent dans lequel on trouve un distributeur est celui d'une « distribution<sup>50</sup> » de système d'exploitation (de type Linux). L'éditeur de cette distribution se trouve être, de fait, distributeur des logiciels qui sont inclus dedans. Le distributeur est donc potentiellement coupable de contrefaçon par suite de l'offre ou de la mise à disposition du logiciel contrefaisant. La grande majorité des logiciels présents dans une distribution l'étant également dans de nombreuses autres distributions, un jugement (ou une menace suffisamment crédible) à

---

<sup>48</sup> S'il ne s'agit pas d'un usage privé, exempté de contrefaçon au regard de la loi française.

<sup>49</sup> Cette section entièrement écrite en majuscules dans la licence GPLv2 est ici reproduite avec les règles de capitalisation habituelles afin de ne pas nuire à la lisibilité.

<sup>50</sup> Cf. définitions en annexe 1.

l'encontre d'un éditeur de distribution Linux pourrait avoir un effet dissuasif sur l'ensemble de ces distributeurs. Nous allons alors rencontrer deux cas de figure potentiels :

- Le logiciel contrefaisant n'est pas nécessaire à la distribution : Dans ce cas, il est probable que l'ensemble des éditeurs de « distributions » n'incluent plus le logiciel contrefaisant, par mesure de prudence de leur part. L'effet dissuasif sera alors atteint. C'est par exemple ce qui s'est passé avec les brevets sur le format de compression audio MP3, où la plupart des distributions avaient cessé d'inclure les encodeurs et décodeurs pour ce format, et ne les ont réinclus qu'à l'expiration des derniers brevets connus<sup>51</sup>.
- Le logiciel contrefaisant est nécessaire à la « distribution » (et est donc vraisemblablement un composant clé de Linux) : Le plus probable est alors un front commun des éditeurs de distributions Linux, mettant leurs ressources en commun soit pour remplacer la partie contrefaisante, soit pour assurer leur défense. Il est très probable que la FSF et l'EFF apporteraient également leur soutien, et potentiellement même certaines sociétés réalisant une importante part de leur chiffre d'affaire grâce aux services et conseil à destination des plateformes Linux (on retiendra par exemple IBM).

Attaquer un éditeur de distribution sur un composant clé, bien que légalement possible, présente un risque majeur de voir se dessiner un front de défense (et de riposte) unissant de nombreux acteurs. Dans la position d'un industriel, qui doit produire et livrer des produits, le niveau de risque semble très élevé.

En 2007, Richard Hillesley, auteur et journaliste, écrivait dans un article intitulé « *The open source patent war* »<sup>52</sup>:

*« Suing the distributors would be the equivalent of shooting Archduke Ferdinand in 1914. All kinds of alliances might come into play, the battlefields could be far flung, and there would be no assurances that [the opponent<sup>53</sup>] would end the war a winner. Too many vendors have too many vested interests in Linux. »*

Un peu plus de dix ans plus tard, cette métaphore est toujours d'actualité, et nul ne peut prévoir les résultats d'une action à l'encontre des éditeurs de distributions Linux.

## **C - Agir à l'encontre de l'éditeur**

Comme dans le cas du logiciel classique, il est possible d'agir à l'encontre de l'éditeur du logiciel open source, si tant est qu'il y en ait un (ce qui est loin d'être systématique). En présence d'un éditeur, dans l'immense majorité des cas celui-ci fournira des versions compilées de son logiciel à destination des principaux systèmes d'exploitation, ce qui constitue potentiellement une contrefaçon. En fonction des cas, l'éditeur pourrait avoir intérêt à prendre une licence sur les brevets contrefaits.

---

<sup>51</sup> Voir par exemple <https://bit.ly/32GqNXF> pour le cas de la distribution Fedora.

<sup>52</sup> Voir <https://bit.ly/2MnJ1aA>

<sup>53</sup> L'opposant était alors Microsoft. La position de Microsoft a maintenant radicalement changé, mais la citation pourrait s'appliquer à n'importe quel adversaire.

En revanche, si le but du breveté est de faire cesser l'usage du procédé contrefaisant, une action à l'encontre du distributeur risque de n'avoir qu'un effet minime, celui-ci pouvant dans la plupart des cas se contenter de fournir le logiciel sous forme de code source sans que cela n'ait d'impact important sur son activité.

Alors que dans le cas du logiciel classique l'éditeur ne fournit son logiciel que sous forme compilée, dans le cas du logiciel open source la forme de distribution la plus importante pour l'éditeur est celle sous forme de code source, a priori non constitutive de contrefaçon.

## **D - Agir à l'encontre des intermédiaires techniques ?**

De nombreux projets open source bénéficient de services techniques de plateformes telles que GitHub, permettant entre autres l'accès au logiciel par tout un chacun. Ces plateformes sont des intermédiaires techniques jouant, pour l'accès au logiciel, le rôle d'hébergeur.

Des moyens existent pour demander le retrait par les hébergeurs de contenus contrefaisants dans le cadre de la propriété littéraire et artistique, comme les procédures de *DMCA takedown*<sup>54</sup> aux Etats-Unis ou les notifications prévues par la Loi pour la confiance dans l'économie numérique (LCEN) en France, mais ceci ne concerne que les atteintes au droit d'auteur. Les cas envisagés dans ces cadres sont principalement des contrefaçons intégrales d'œuvres, dans lesquels la contrefaçon apparaît comme manifeste.

Dans le cadre de contrefaçon de brevet, ces procédures ne sont pas applicables. Autant il est normalement aisé pour le breveté de montrer sa titularité des droits, autant il est difficile de démontrer la contrefaçon de brevet dans le cadre de logiciels.

Statuer sur la contrefaçon relevant du rôle de la justice, un intermédiaire technique, tel qu'un hébergeur, serait peu enclin à prendre en compte une demande de retrait ou de blocage de l'accès à un logiciel sur la base d'une simple allégation de contrefaçon de brevet. Pour un hébergeur, accéder à une telle demande le mettrait commercialement dans une position délicate en jetant un discrédit sur la confiance que l'on peut lui accorder dans le cadre des projets open source, et un tel comportement mettrait probablement en péril les activités de cet hébergeur. Les chances de succès d'une telle demande envers un hébergeur semblent donc assez faibles, au vu de la position dans laquelle il se trouverait placé en accédant à cette demande.

Intenter une action en justice à l'encontre d'un hébergeur semble également peu judicieux. Ce dernier se retrancherait probablement derrière son statut de simple intermédiaire technique, et il est peu probable qu'une telle action permette de récupérer les frais engagés pour la procédure.

Si en revanche un jugement en contrefaçon est obtenu à l'encontre d'une autre entité (comme par exemple l'éditeur du logiciel), alors une demande de retrait à l'hébergeur ne devrait pas poser de problème particulier.

## **E - Les clauses de résiliation de licence**

Avant d'intenter une action en contrefaçon de brevet liée à un logiciel open source, il conviendra d'être attentif à ne pas perdre une licence dont on pourrait soi-même avoir besoin. En effet, certaines licences open

---

<sup>54</sup> Procédure introduite par le Digital Millennium Copyright Act, permettant de demander à un intermédiaire technique le retrait ou le déréférencement d'une occurrence de contrefaçon de droit d'auteur.

source contiennent des clauses de résiliation de licence en cas d'action en contrefaçon à l'encontre d'un tiers (*patent retaliation clause*). A titre d'exemple, voici les clauses que l'on peut rencontrer dans certaines licences :

Clause de la licence Apache 2.0 :

*If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.*

Clause de la licence Mozilla Public License (MPL) 2.0 :

*5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.*

Clause de la licence GNU General Public License (GPL) v3 :

*[...] you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.*

Une action en contrefaçon sur la base d'un logiciel couvert par une telle licence fera donc perdre le bénéfice de la licence de ce logiciel au plaignant, qui se trouvera contraint de ne plus pouvoir redistribuer ledit logiciel. La plus grande prudence doit donc être de mise dans ces cas.





## SECONDE PARTIE : COMBINER BREVETS ET OPEN SOURCE

Après avoir traité dans la première partie de la problématique de la contrefaçon de brevet par le logiciel open source, nous allons maintenant traiter des possibilités de combiner la protection par brevets et le logiciel open source, au sein d'une stratégie conjointe.

Alors qu'historiquement le logiciel open source était plutôt ancré dans le milieu académique, et les brevets dans le milieu industriel, la vision d'une opposition manichéenne entre deux camps n'est plus pertinente aujourd'hui : de nos jours, les universités valorisent leur recherche grâce aux brevets, et plus aucune entreprise industrielle ne peut ignorer les logiciels open source, devenus des éléments nécessaires au fonctionnement et au développement de nombreux systèmes.

L'exemple le plus frappant de cette évolution est le positionnement de Microsoft : En 2001, Steve Ballmer, le CEO de Microsoft, comparait Linux à un Cancer :

*Linux is a cancer that attaches itself in an intellectual property sense to everything it touches.*

Quinze ans plus tard, les choses ont bien changé :

- En 2016, Microsoft obtenait un siège au comité de direction de la Linux Foundation
- En 2018, Microsoft rachetait GitHub, la plus importante forge de logiciels open source, pour 7,5 milliards de dollars.

Les organisations industrielles actuelles doivent donc organiser leur stratégie à la fois au niveau des brevets, et au niveau de l'open source, qu'elles se doivent de prendre en compte.

En parallèle des organisations industrielles « traditionnelles », nous avons les sociétés dont le développement de logiciels open source est le cœur de métier, qui du fait de leur croissance<sup>55</sup> ont à faire face à la menace grandissante de brevets de tiers, et qui peuvent souhaiter mettre en place une stratégie brevets pour assurer leur développement et leur protection.

Dans les deux cas, il est vrai que la combinaison entre une stratégie de développement open source et une stratégie de brevets semble peu évidente. Nous traiterons dans cette seconde partie des moyens possibles pour combiner les deux approches, en évitant les potentiels écueils.

---

<sup>55</sup> A titre d'exemples, en 2018 Mozilla Corporation (la société coordonnant le développement du navigateur Firefox) a plus de 1000 employés et un revenu annuel d'environ 500 millions de dollars et Red Hat (éditeur de distribution Linux, de solutions d'infrastructures cloud et fournisseur de services associés) a plus de 13 000 employés et un revenu annuel de 3,4 milliards de dollars.

## I - Le brevet pour soutenir une stratégie open source

Lorsque la stratégie de l'organisation comporte une forte composante open source, lui adjoindre une composante brevet peut s'avérer être un renfort de grande valeur.

Le brevet pourra notamment être utilisé pour atteindre les buts suivants :

### A - Contrôler la distribution de la connaissance

#### 1. Palier à la perte de savoir-faire

Le « savoir-faire » est défini<sup>56</sup> comme :

*« Savoir-faire » : un ensemble d'informations pratiques non brevetées, résultant de l'expérience et testées, qui est :*

- i. Secret, c'est-à-dire qu'il n'est pas généralement connu ou facilement accessible ;*
- ii. Substantiel, c'est-à-dire important et utile pour la production des produits contractuels, et*
- iii. Identifié, c'est-à-dire décrit d'une façon suffisamment complète pour permettre de vérifier qu'il remplit les conditions de secret et de substantialité ;*

Même s'il ne fait pas partie des titres de propriété intellectuelle, le savoir-faire est indéniablement un avantage concurrentiel pour une entreprise.

Cependant, le développement de logiciels open source détruit le savoir-faire, par le fait que le caractère secret n'est pas préservé. De fait, une fois le code source divulgué, tout un chacun a accès à l'information qui se trouvait auparavant secrète, y compris les concurrents, et est en mesure de la réutiliser.

Dans un tel cas de perte de savoir-faire, le seul moyen de préserver l'avantage compétitif est de s'assurer le contrôle des techniques sous-jacentes par le biais de brevets. Le savoir-faire participant à la valeur d'une entreprise, il s'agit de préserver cette valeur en transférant la valeur précédemment conférée par le savoir-faire en valeur conférée par le brevet. Il semble donc particulièrement intéressant (voir nécessaire), dans un souci de bonne gestion de la valeur d'une entreprise, de déposer des brevets avant une divulgation par le biais de logiciel open source.

L'entreprise est ensuite libre de déterminer la façon dont elle exerce ou non le monopole qui lui est conféré par le brevet, mais au moins elle évite, par le biais du brevet, la dévalorisation de son capital immatériel : elle transfère la valeur de sa connaissance, qui était portée par son savoir-faire, en valeur portée par le brevet.

---

<sup>56</sup> Définition venant du règlement CE 772/2004, le Code de la propriété intellectuelle ne définissant pas le savoir-faire

Note :

Sémantiquement, il y a une forte différence entre la notion de « savoir-faire » du point de vue juridique et du point de vue du langage courant. Dans le langage courant, seul le caractère substantiel est généralement retenu, le secret et la description n'étant bien souvent pas considérés.

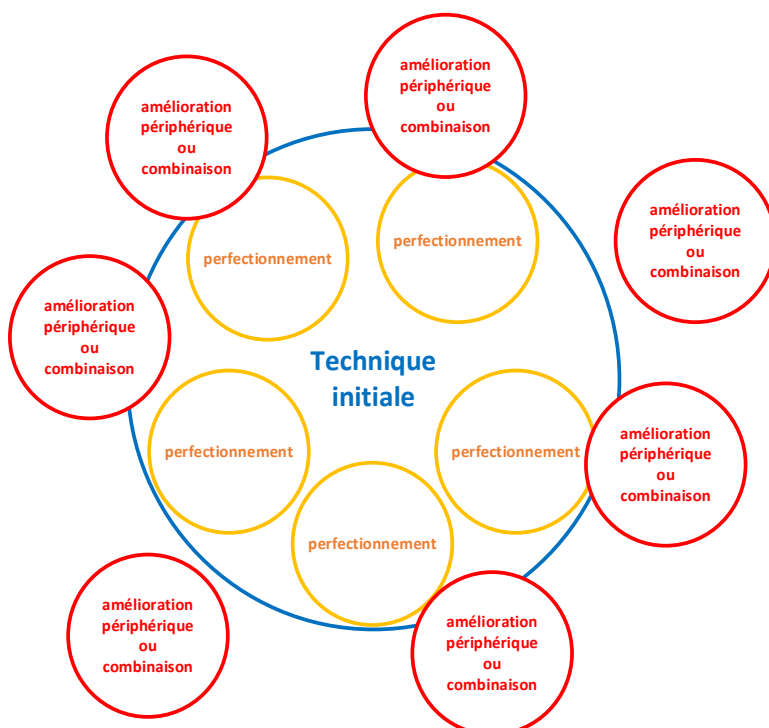
Le non-juriste, confondant souvent savoir-faire et compétence, pourra donc éprouver quelques difficultés à envisager la perte de savoir-faire induite par la divulgation publique de code source, alors même que cela fait économiquement pleinement sens.

## 2. Favoriser la libre exploitation des connaissances dérivées

Dans le cas des organisations dont le but est la mise à disposition de savoir à destination du public, comme les organismes publics de recherche (OPR), la détention de brevets combinée à un modèle de licence ouverte de brevet<sup>57</sup> avec des clauses de réciprocité<sup>58</sup> permet d'éviter la privatisation d'améliorations des savoirs divulgués (Pénin, 2010).

Par exemple, à l'issue d'une publication effectuée par un OPR, une entité pourrait déposer des brevets d'améliorations sur le procédé divulgué, et ainsi réaliser un encerclement de la technologie par ses propres brevets. Cette entité se trouverait donc seule à contrôler l'utilisation de la technologie, allant à l'encontre de l'intention de l'OPR, qui est que les travaux réalisés profitent au plus grand nombre.

Figure 4 : Encerclement technologique



<sup>57</sup> Licence gratuite ou à faible coût, non exclusive, disponible sans discrimination.

<sup>58</sup> Le licencié doit mettre à disposition ses améliorations selon les mêmes conditions.

*Encerclement technologique :*

*- La technique initiale est divulguée par une première entité. Cette technique, nouvelle, est décrite de façon large.*

*- Une ou plusieurs autres entités déposent des brevets sur des perfectionnements (plus précis), ou des combinaisons de la technique initiale avec d'autres techniques.*

*Le résultat est que si suffisamment de nouveaux brevets sont déposés, la technique initiale n'est alors plus libre d'exploitation, y compris par la première entité.*

Un système de licence de brevets ouverte et réciproque, telle que la licence MOSPL de Mozilla (voir section « Licences ouvertes de brevets ») permet d'éviter cet écueil, tout en permettant au plus grand nombre de profiter aussi des améliorations.

La diffusion de logiciel open source se trouve confrontée à ce même problème potentiel d'appropriation technologique par un tiers : il est possible pour une entité de réaliser un encerclement par brevets des connaissances divulguées par le logiciel open source, et ainsi d'en prendre le contrôle en empêchant sa libre exploitation.

Le dépôt de brevet, associé à une licence ouverte de brevets, est un moyen efficace de s'assurer de la libre exploitation des améliorations ultérieures.

Cette stratégie souffre cependant d'un problème de charge financière : pour qu'elle fonctionne, il est nécessaire de procéder au dépôt du brevet (ce qui représente un coût initial), mais également de maintenir le brevet en vigueur en payant les annuités (ce qui représente un coût grandissant avec le temps).

Ceci est un investissement pour le bien collectif, qui peut représenter un coût difficile à supporter alors que l'on ne peut en retirer de contrepartie financière directe. Il est cependant possible de réaliser des optimisations sur le coût, par exemple en arrêtant de payer les annuités une fois que l'invention initiale est considérée comme suffisamment encadrée par de nouvelles connaissances publiquement disponibles et exploitables, considérant qu'à ce moment l'invention initiale ne peut raisonnablement plus être privatisée par encerclement.

L'élaboration et l'optimisation d'une telle stratégie pourraient constituer le cœur de travaux dédiés, dépassant le cadre du présent mémoire.

## **B - Faire partie de l'écosystème des brevets**

### **1. Ajout à l'état de l'art**

Le logiciel open source, une fois publié, fait partie de l'état de l'art, et est opposable, en tant qu'art antérieur, aux demandes ultérieures de brevets.

Il est cependant très difficile de relier des revendications d'une demande de brevet à du code source déjà publié, du fait de la difficulté à effectuer des recherches dans du code source. Bien que l'humain puisse comprendre l'intention technique du code source en le lisant, la machine, tout comme les outils de recherche, ne comprennent pas l'intention. Bien sûr, la machine peut interpréter le code source une fois compilé, mais elle ne peut, en l'état actuel des choses, savoir quel est le but, ni quels sont les moyens mis en œuvre par le code source. Ceci rend le code source imperméable à la plupart des techniques de recherches documentaires.

En conséquence, l'examineur brevets ne trouvera qu'extrêmement rarement du code source constituant un art antérieur. Les rares cas où un examinateur produit du code source pour objecter de la nouveauté d'une demande sont des cas où le code source est accompagné d'une description textuelle de l'intention et des étapes de réalisation techniques, comme par exemple le code source de standards techniques, qui est dans ce cas accompagné du texte du standard. C'est alors le texte du standard, ou les textes des propositions techniques, qui permettent à l'examineur d'identifier le code source pertinent. Ainsi, les outils actuels ne permettent pas à l'examineur d'effectuer une recherche solide d'art antérieur dans les codes sources qui sont à disposition du public.

Déposer une demande de brevet est donc un moyen, pour le projet open source, de s'assurer que l'invention soit transcrite en une forme accessible aux outils de recherche, et que cette demande, en étant intégrée dans les bases de brevets, fasse partie du corpus technique facilement accessible aux examinateurs des offices de brevets.

En conséquence, cela réduit le risque de devoir se défendre ultérieurement face à un brevet revendiquant la même invention que celle déjà implémentée.

## 2. Crédibilité envers les partenaires

Dans certains cas, détenir des brevets permet d'améliorer sa position lors de discussions avec des partenaires. Le brevet peut être un gage de crédibilité augmentant la valeur perçue d'une solution par rapport à une autre.

Lors de discussions avec des partenaires possédant des brevets, avoir soi-même des brevets permet de se trouver à égalité lors des discussions.

C'est par exemple ce qui s'est passé lorsque Mozilla a cherché à convaincre des partenaires industriels de participer au développement de formats de compression audio et vidéo qui ne soient pas sujet à redevances<sup>59</sup> (« *royalty free* ») :

*We found that, without related patents of our own, it was extremely difficult to persuade companies (particularly large ones) to openly license their patents or adopt standards based on our developed technology. We ran into this problem repeatedly, especially in spaces that are more commonly patented. Obtaining a patent [...] gave us leverage in these discussions.*

*Denelle Dixon, Chief Operating Officer, Mozilla*

## C - Conquête

### 1. Faire respecter les termes de la licence

Le brevet pourra être utilisé à l'encontre d'une entité ne respectant pas les termes de la licence du logiciel open source, afin de la convaincre (ou de la forcer) de respecter les conditions de la licence.

Ce point présente un intérêt non négligeable, la menace d'une action en contrefaçon de brevet (potentiellement assortie d'une injonction pour faire cesser la contrefaçon) en sus d'une action pour violation de droits d'auteur pouvant être particulièrement dissuasive.

---

<sup>59</sup> Voir <https://mzl.la/2YsNQ9c>

## 2. Accroître ses parts de marché

Le fait qu'un logiciel soit distribué sous forme open source n'exclut pas la possibilité de stratégies de conquête de marché. L'une de ces stratégies est de faire jouer le brevet à l'encontre de solutions concurrentes (soit directement auprès des éditeurs, soit auprès des utilisateurs). Ceci peut éventuellement amener les éditeurs et/ou utilisateurs de solutions concurrentes à utiliser le logiciel open source plutôt que de payer une redevance sur le brevet.

Cette technique relevant d'une stratégie ouvertement offensive, il conviendra cependant d'anticiper les retombées sur l'image du breveté : par exemple, dans le cas d'un projet collaboratif, la connaissance d'une telle pratique pourrait jouer sur l'opinion des contributeurs externes, réduisant potentiellement leur souhait de participer au projet.

## D - Défense

### 1. Licences croisées

Comme dans le cas d'un logiciel qui ne soit pas open source, le brevet peut être utilisé dans des négociations pour obtenir des licences croisées.

Dans le cas du logiciel open source, la cible sera de préférence des entités ne contribuant pas déjà au logiciel (les contributions engendrant déjà accord de licence -de brevet- selon beaucoup de licences open source), mais rien n'empêche d'établir un accord de licences croisées avec une entité contribuant au logiciel. On pourra, par exemple, établir un accord plus général que le cadre d'un logiciel particulier, en incluant certains logiciels auxquels le breveté tiers ne contribue pas.

*Note : En plus des licences croisées entre deux entités, il est également possible, comme nous le détaillerons plus loin, de rejoindre des systèmes de pools défensifs spécifiques à l'open source.*

### 2. Portefeuille défensif

Un portefeuille de brevets peut servir de bouclier, en tant que portefeuille défensif (Rice, 2015). Une stratégie de développement basée sur l'open source n'empêche pas d'être la cible d'attaques potentielles sur la base de brevets. Avoir un portefeuille défensif pour parer à de telles attaques est une stratégie souvent recherchée par les grandes entreprises, qui peut tout aussi bien s'appliquer au domaine de l'open source.

Red Hat a ainsi créé un important portefeuille à usage défensif, comprenant 2600 familles de brevets en vigueur, et s'explique ainsi <sup>60</sup>:

*« Red Hat has consistently taken the position that software patents generally impede innovation in software development and that software patents are inconsistent with open source/free software [...] At the same time, we are forced to live in the world as it is, and that world currently permits software patents.*

*[...]*

*One defense against such misuse is to develop a corresponding portfolio of software patents for defensive purposes. Many software companies, both open source and proprietary, pursue this strategy.*

---

<sup>60</sup> Voir <https://red.ht/32VY7tX>

*In the interests of our company and in an attempt to protect and promote the open source community, Red Hat has elected to adopt this same stance. We do so reluctantly because of the perceived inconsistency with our stance against software patents; however, prudence dictates this position. »*

## **E - Autres stratégies « habituelles »**

En plus des points évoqués ci-dessus, la plupart des stratégies d'utilisation de brevets restent applicables dans le cas des logiciels open source. On pourra par exemple citer les stratégies d'encercllement ou de barrage technologique.

*Note : Le lecteur souhaitant plus d'informations sur ce type de stratégies relatives aux brevets pourra par exemple se référer à Rolnik (2002).*

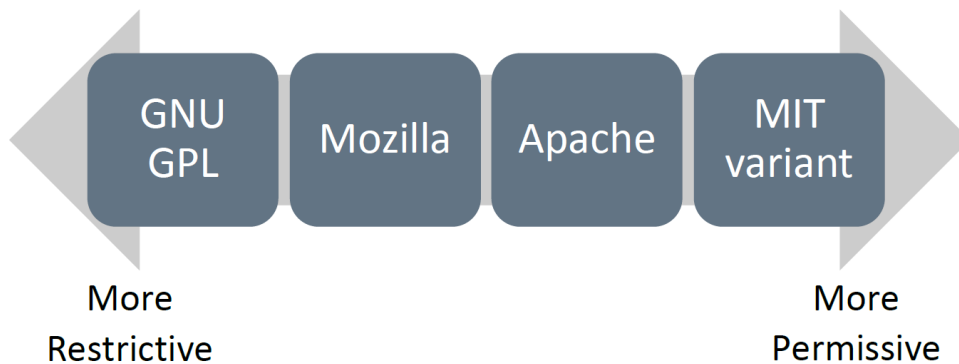
En revanche, les stratégies destinées à brouiller l'information (brevets leurres, figures et résumés de brevet intentionnellement abscons, ...) ne seront pas efficaces dans le cas de l'open source, le développement du logiciel étant réalisé de manière publique.

## **II - Points de vigilance**

Les licences open source les plus anciennes ont été écrites alors que le nombre de contentieux pour contrefaçon de brevet était bien moins important qu'actuellement. Initialement, les principaux droits permettant de restreindre la circulation des logiciels étaient le copyright (pour les pays relevant de la *common law*) et le droit d'auteur, c'est pourquoi les plus anciennes licences ne mentionnent pas explicitement les brevets d'invention. Basées sur la notion de copyright, les licences telles que la GPLv2 ne traitent même pas du cas de l'utilisation du logiciel, le droit d'utilisation n'étant pas restreint par le copyright ou le droit d'auteur. Pour autant ces licences ont tout de même des impacts, parfois très importants, sur les droits associés aux brevets.

La prise de conscience de l'importance des brevets, y compris pour les procédés implémentés par logiciel, a eu pour effet que les licences plus récentes traitent généralement explicitement des problématiques de brevets. De plus, il est indéniable que dans leurs buts, certaines licences aient un positionnement différent : à un extrême nous trouvons les licences de type GPL, qui font en sorte d'abolir toute forme de restriction à l'encontre des utilisateurs (et qui, paradoxalement, sont plus restrictives dans leurs termes), alors qu'à l'autre extrême nous trouvons des licences très permissives, telles que la licence MIT, qui sont bien plus adaptées aux stratégies d'entreprise.





Source : Hamilton Brook Smith Reynolds - <https://bit.ly/32RT2mm>

Que les brevets soient traités de manière implicite ou explicite, les licences plus ou moins permissives, voyons quels sont les points particuliers à prendre en compte au niveau des interactions entre les licences open source et les brevets.

## A - Compatibilité entre licences open source et ses propres brevets

### 1. Code couvert par ses propres brevets

Les licences open sources sont toutes construites pour permettre la redistribution ultérieure du logiciel avec les mêmes droits que la distribution initiale. Si un détenteur de brevet utilise, modifie ou diffuse le code du logiciel open source, il est réputé le faire en accord avec les termes de la licence.

*Note : La licence de brevet peut être induite ou non selon différents cas (utilisation, copie, distribution, modification), et implicite ou explicite, en fonction des différentes licences open source.*

En conséquence, le breveté modifiant/redistribuant le logiciel open source donne tacitement son accord pour une redistribution gratuite du logiciel couvert par son brevet, et ne pourra pas ultérieurement demander de licence portant sur ce logiciel.

Dans le cas où le brevet couvre une portion de code modifiée par le breveté, cet accord de licence implicite peut sembler assez naturel. Il faut cependant prendre garde à la portée de cette licence (de brevet), qui varie grandement en fonction des licences open source : en cas de changement apporté à un logiciel open source, alors que la licence Apache 2.0 stipule que ceci constitue un accord de licence (de brevets) uniquement pour les changements accordés, la licence GPLv3 entraîne un accord de licence (de brevets) pour l'ensemble du logiciel. Modifier un logiciel open source sous une licence telle que la GPLv3 revient donc à donner une licence pour ce logiciel pour tous les brevets que l'on détient.

Selon les termes des licences open source, les droits accordés sont transmissibles, dans les mêmes conditions, aux travaux dérivés, qui doivent alors bénéficier de la même licence. Cela inclut bien sûr les modifications apportées ultérieurement au logiciel, mais couvre aussi la réutilisation de code par d'autres logiciels. Dans ce cas, la potentielle licence gratuite de brevet sera considérée comme étendue aux autres logiciels réutilisant le code.

Cela ne signifie pas pour autant que l'acceptation d'une licence open source détruit la valeur des brevets détenus. Il est bien sûr toujours possible d'utiliser ces brevets hors du cadre du logiciel open source, tout comme dans le cadre de l'open source (comme nous l'avons vu dans les sections précédentes).

*Note : Certaines licences, en particulier les licences GPLv2 et LGPLv2.1 sont ambiguës quant au fait qu'elles incluent ou non une licence implicite de brevets (Haapanen, 2015).*

## 2. Redistribution du logiciel

Le logiciel open source et son code source étant protégés par le droit d'auteur et/ou le copyright, en effectuer une copie pour le mettre à disposition d'autrui nécessite également l'acceptation de la licence, sauf à enfreindre la propriété littéraire liée au logiciel.

De nombreuses entreprises, particulièrement dans l'informatique et les télécoms, mettent à la disposition du public les logiciels open source qu'elles utilisent dans leurs produits. Les licences open source obligeant souvent à donner à l'utilisateur le moyen d'obtenir le code source du logiciel, les entreprises ont deux principales possibilités :

- Pour chaque produit, dresser la liste des sites internet de chacun de ces logiciels open source utilisés, et mettre cette liste à disposition du public
- Pour chaque produit, réaliser une archive du code source de tous les logiciels open source utilisés, et mettre cette archive à disposition du public.

Bien souvent, par souci de simplicité, c'est la seconde option qui sera retenue. Ce qui est souvent négligé, c'est qu'en agissant ainsi, l'entreprise met le logiciel à disposition du public (à titre gratuit)

Le logiciel étant possiblement couvert par des brevets de l'entreprise, ceci correspond à une mise sur le marché du produit breveté, par le détenteur du brevet.

Les droits sur le brevet sont donc, dans le contexte de ce logiciel, épuisés, et l'entreprise ne pourra ultérieurement demander de redevance pour les copies du logiciel qu'elle a elle-même fournies.

Ce raisonnement peut être poussé encore plus loin dans les juridictions reconnaissant le principe d'*Estoppel*<sup>61</sup> (principe selon lequel une partie ne peut se contredire au détriment d'un tiers) (Houtcieff, 2011) : l'entreprise distribuant le logiciel open source indiquant par cet acte son acceptation des termes de la licence, accepte donc que le logiciel soit ensuite à nouveau redistribué selon les mêmes termes de licence, ce qui inclut la licence de brevet. En mettant le logiciel open source à disposition du public, l'entreprise a renoncé à toute redevance sur ses brevets couvrant ce logiciel, y compris les œuvres dérivées de ce logiciel.

Ce point peut donc potentiellement avoir des conséquences économiques ennuyeuses pour le breveté, et il convient d'y être particulièrement attentif.

---

<sup>61</sup> L'*Estoppel* est considéré comme un principe sous la *common law*. En droit français, ce principe est reconnu par la jurisprudence depuis 2005, mais ses contours et limites ne semblent pas encore complètement définis.

*Note : Certaines licences, telles que la GPLv2 ne placent pas de limite quant à ce qui constitue une œuvre dérivée. L'œuvre dérivée peut donc se trouver très différente de l'œuvre originale, et l'argument de l'Estoppel est alors bien moins solide. Il est en effet difficile de soutenir que le breveté ait eu l'intention d'accorder une licence de brevet pour cette œuvre fort éloignée de la version « acceptée » (Pugh et Majerus, 2006).*

### **Code source et épuisement du droit**

Nous venons d'expliquer que la mise à disposition du code source engendrait un épuisement du droit de ses propres brevets. Or, dans la première partie de la présente étude, nous avons argumenté du fait que la diffusion de code source ne constitue pas une contrefaçon, le code source étant une description et non une réalisation.

Le lecteur pourrait donc s'interroger sur l'interaction entre les deux arguments : si le code source ne constitue pas une contrefaçon, la distribution de code source peut-elle engendrer un épuisement du droit sur les brevets ?

L'affaire *Cascades v. Samsung*<sup>62</sup>, jugée en 2014 aux Etats-Unis, nous apporte une réponse positive. L'affaire était la suivante :

La société Cascades avait concédé un brevet en licence à Google, les revendications du brevet étant mises en œuvre dans un élément du système d'exploitation Android. Cascades a ensuite assigné Samsung en contrefaçon, sur la base des téléphones vendus par Samsung et fonctionnant sous Android.

La Cour a jugé que puisque le code source d'Android était fourni par Google, il y avait épuisement du droit sur le brevet, la fourniture du code source par Google constituant une vente autorisée<sup>63</sup>.

Cet exemple nous montre que la mise à disposition autorisée de code source constitue bien un fait potentiellement générateur d'épuisement du droit. Il n'y a pas nécessité que cette opération soit faite par le biais d'une version compilée.

## ***B - Compatibilité entre licences GPLv2/LGPLv2.1 et brevets licenciés***

Se pose parfois la question de savoir si certaines licences open source (en particulier les licences GPL et LGPL) sont compatibles avec des brevets que l'on ne détient pas, mais sur lesquels on bénéficie d'une licence.

Le questionnement vient d'une clause particulière des licences GPLv2 et LGPLv2.1, clause souvent appelée « *liberty or death* » :

*7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations*

<sup>62</sup> Voir jugement à l'adresse <https://bit.ly/2Y6CNTP>

<sup>63</sup> Le fait que le code source soit fourni gratuitement n'impacte pas le qualificatif de « vente autorisée », la vente étant caractérisée par le transfert et non par le prix.

*under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all.*

*For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.*

[...]

Cette clause a pour but d'empêcher l'ajout de contraintes non prévues dans la licence. Si une telle contrainte supplémentaire existe, il n'est alors plus possible de distribuer le logiciel. L'exemple cité laisse penser que si l'on bénéficie d'une licence payante sur un brevet, il ne serait alors plus possible de distribuer le logiciel, car l'on ne peut garantir une redistribution gratuite.

*Note : D'autres licences, telles que les licences BSD ou MIT n'ont pas ce problème potentiel, rien dans leur formulation ne laissant présager une éventuelle incompatibilité avec une licence de brevet obtenue de façon séparée.*

Un cas représentatif de cette problématique et celui des bibliothèques logicielles sous licence LGPLv2.1 qui fournissent des fonctionnalités d'encodage et de décodage de formats audio/vidéo, alors que ces formats sont couverts par des brevets donnés en licence par des pools de brevets<sup>64</sup>.

Pour mieux comprendre les possibilités relatives à ce cas, reprenons l'argumentaire fourni par Google au sujet de l'utilisation d'un décodeur vidéo sous licence LGPL dans le navigateur Chrome<sup>65</sup>, le décodeur étant couvert par des brevets tiers licenciés par Google :

---

<sup>64</sup> Un pool de brevets est une entité permettant d'obtenir, sur une technologie donnée, une licence commune de brevets accordée par plusieurs bailleurs.

<sup>65</sup> La discussion d'origine se trouve à l'adresse <https://bit.ly/2K1CpMn>

[...] il y a deux cas possibles :

*Cas1:*

- a. *L'entité A fournit à l'entité B une bibliothèque sous licence LGPL 2.1 avec une licence de brevet stipulant que seule l'entité B a le droit d'utiliser la bibliothèque*
- b. *L'entité B souhaite distribuer la bibliothèque à des tiers*

*Il s'agit du cas donné en exemple dans le texte de la licence LGPL 2.1 et constituerait probablement une violation [des termes de la licence]*

*Cas2 :*

- a. *L'entité A fournit à l'entité B une bibliothèque sous licence LGPL 2.1*
- b. *L'entité B obtient une licence de brevet de l'entité C*
- c. *L'entité B distribue la bibliothèque sous LGPL 2.1*

*Cette situation n'est \*pas\* interdite par la LGPL 2.1 (se référer à la LGPL 3.0 pour une licence qui traite de ce cas). Sous la LGPL 2.1, le fait que l'entité B puisse avoir une licence de brevet obtenue d'un tiers n'est pas pertinent tant que cela n'empêche pas l'entité B de donner au public les droits que la LGPL 2.1 requiert qu'elle lui donne (précisément, seulement le droit qu'elle a reçus de l'entité A).*

*Chris DiBona, Open Source Director, Google*

La position adoptée par Google est l'une des interprétations possibles de la licence LGPLv2.1, mais force est de constater qu'il y a matière à débat. Les licences GPLv2 LGPL ont clairement été rédigées en prenant en compte un aspect militantiste, dans le but d'atteindre un idéal de liberté. Conjuguer ces licences avec la réalité commerciale des brevets est un exercice qui n'est pas trivial.

Finalement, la question est peut-être de savoir si une partie peut ou non s'estimer lésée : si les brevetés sont satisfaits (par le paiement de la licence), et si les ayants-droits « littéraires » sont satisfaits (par la diffusion du logiciel), alors le risque juridique semble faible.

De manière empirique, nous pouvons constater qu'il existe de nombreux cas de bibliothèques sous licence LGPLv2.1 intégrées dans d'autres logiciels (open source ou non), couvertes par des brevets dont les licences doivent être contractées auprès de tiers. A notre connaissance, ce système n'a jusqu'à maintenant pas été contesté en justice, et semble être admis tant par les auteurs des logiciels (détenteurs des droits d'auteur et/ou copyright) que par les détenteurs de brevets.

*Note : Sur la base de l'argumentaire de Google relatif au cas évoqué ci-dessus, nous pouvons remarquer que si une entité crée/modifie/diffuse un logiciel open source dont la licence oblige à donner une licence de brevet, il suffirait qu'elle ne détienne pas elle-même les brevets pour ne pas se trouver contrainte (par la licence open source) de donner une licence de brevets.*

*On peut légitimement s'interroger si le fait de placer ses brevets dans une structure séparée ne permettrait donc pas de s'affranchir des contraintes de licence (brevets) obligatoire et gratuite imposées par les logiciels open source. Ceci serait bien évidemment contraire à l'esprit de la licence open source, mais pas forcément à sa lettre, illustrant les possibilités de faire dévier cette licence de l'idéal qu'elle cherche à atteindre.*

## C - Contributor license agreement

En plus de la licence open source choisie, certains projets comportent un contrat supplémentaire relatif aux contributions, appelé *Contributor Licence Agreement* (CLA). Il s'agit très généralement de gros projets, le plus souvent portés par des entreprises.

Le CLA est un contrat additionnel auquel les contributeurs doivent adhérer pour pouvoir soumettre du code au projet. Les licences open source étant le plus souvent standardisées (afin de faciliter la réutilisation de code entre projets), leurs clauses peuvent ne pas être suffisantes pour couvrir les besoins du projet. Le CLA permet d'ajouter des clauses et des clarifications qui soient propres au projet, en fonction des besoins précis du projet.

Les types de clauses que l'on trouve dans les CLA sont par exemple :

- Attestation du droit du contributeur à apporter du code source au projet, avec si besoin un accord de l'employeur<sup>66</sup>.
- Attestation que le contributeur soit réellement l'auteur du code contribué.
- Attribution du copyright à l'éditeur du logiciel (ce qui permet éventuellement à l'éditeur de relicencier le code sous une autre licence)
- Mention explicite de l'accord de licence gratuite de brevet
- Autorisation donnée au projet d'ester en justice au nom du contributeur
- Désignation de la juridiction compétente en cas de conflit

Les CLA peuvent ainsi être un bon moyen de sécuriser les droits du projet, sans pour autant remettre en cause l'utilisation de licence standardisée.

A la décharge des CLA, leurs détracteurs considèrent qu'un CLA réduit l'incitation à contribuer, en ajoutant un point de friction supplémentaire, du fait d'avoir à donner explicitement son accord sur des points contractuels. (On ne peut leur donner tort, vu que le but d'un CLA est justement de s'assurer de l'accord sur des termes contractuels).

---

<sup>66</sup> Certaines juridictions, telles que la France, attribuent le droit d'auteur des logiciels à l'employeur et non au salarié

## **D - Référentiel de code et divulgation**

Dans le mode de développement traditionnel de l'open source, le code source est placé le plus tôt possible dans le référentiel de code afin de permettre son examen et sa prise en compte immédiate par les autres développeurs.

Ceci pose, dans le cadre d'un dépôt de brevet, un problème de divulgation, détruisant la nouveauté requise pour accéder à la brevetabilité. Il convient donc d'adopter une stratégie particulière (et potentiellement peu habituelle), afin que le développement open source réalisé par une organisation ne lui bloque pas la possibilité de développer son portefeuille de brevets.

Cela nécessite une étroite coopération au sein de l'entreprise, entre les personnes s'occupant du développement et celles s'occupant des dépôts de brevets, afin de s'assurer que l'ordre est bien respecté (dépôt de demande de brevet d'abord, divulgation du code source ensuite).

Cependant, demander aux développeurs d'attendre avant de placer leur code dans le référentiel de versions est difficile. En effet, il est bien plus facile de mettre fréquemment à jour le référentiel pour de petits changements simples que de le mettre à jour moins souvent avec des modifications plus importantes et plus complexes.

Cette façon de procéder étant préjudiciable au développement, nous allons aborder quelques alternatives possibles.

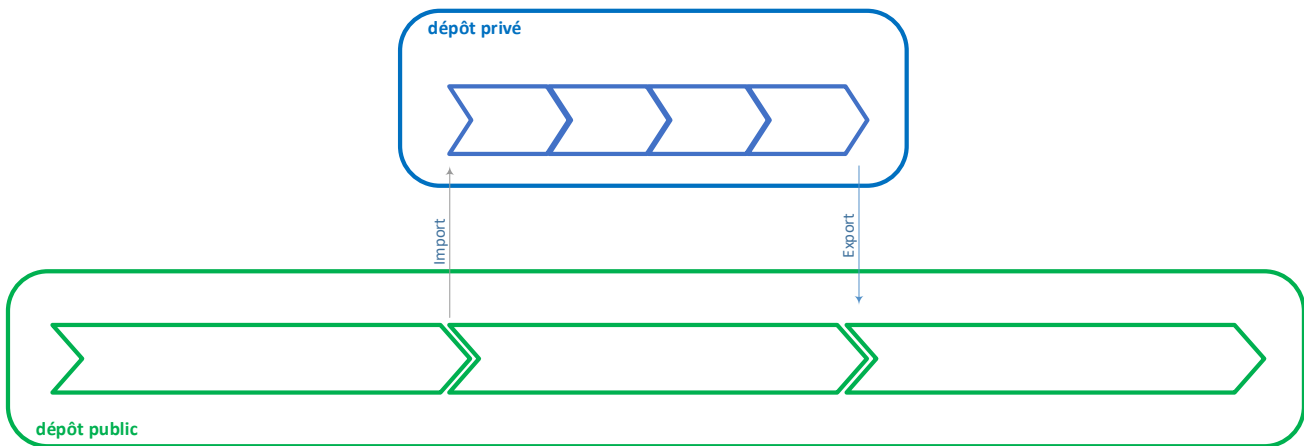
### **1. Emploi d'un dépôt privé**

La première solution serait de réaliser le développement en mode privé, et de ne faire que de temps à autres des reports sur le dépôt public.

Dans ce cas, les développeurs utilisent un dépôt interne à l'organisation (non accessible par le public) pour effectuer le travail de développement. Ceci permet de continuer à faire des mises à jour fréquentes du référentiel privé, facilitant le travail de développement.

Le référentiel public sera mis à jour moins fréquemment, uniquement pour des changements considérés comme stables (par exemple une nouvelle version du logiciel). Ainsi, ceci laissera plus de temps pour traiter les demandes de brevet, la divulgation au public ne se faisant que par le biais du référentiel public.

Figure 6 : Développement en utilisant un dépôt privé

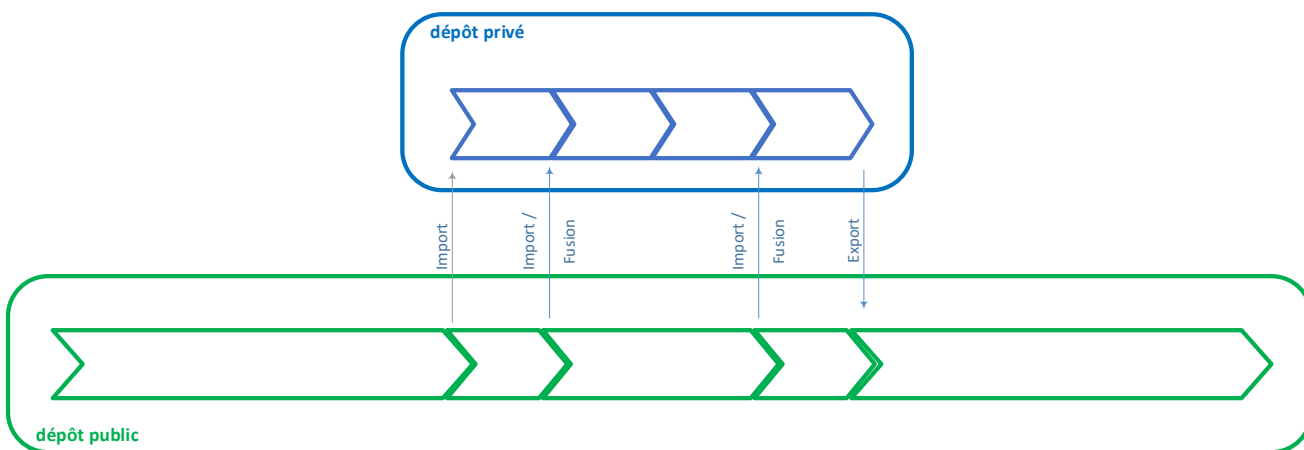


Ce schéma a cependant l'inconvénient de ne pas permettre d'avoir un développement réalisé en commun sur une même fonctionnalité entre plusieurs organisations, le référentiel utilisé pour le développement étant en réalité privé.

Les mises à jour depuis et vers le référentiel public, si ce dernier est aussi alimenté par d'autres, seront potentiellement plus difficiles que si le développement était directement réalisé sur le dépôt public, car ces mises à jour auront à prendre en compte des changements plus importants.

Cette difficulté peut cependant être amoindrie en important et fusionnant régulièrement les évolutions provenant du dépôt public :

Figure 7 : Développement en utilisant un dépôt privé et des imports réguliers



Ce mode de fonctionnement permet d'avoir des mises à jour plus faciles à réaliser, puisqu'elles contiennent chacune moins de changements.

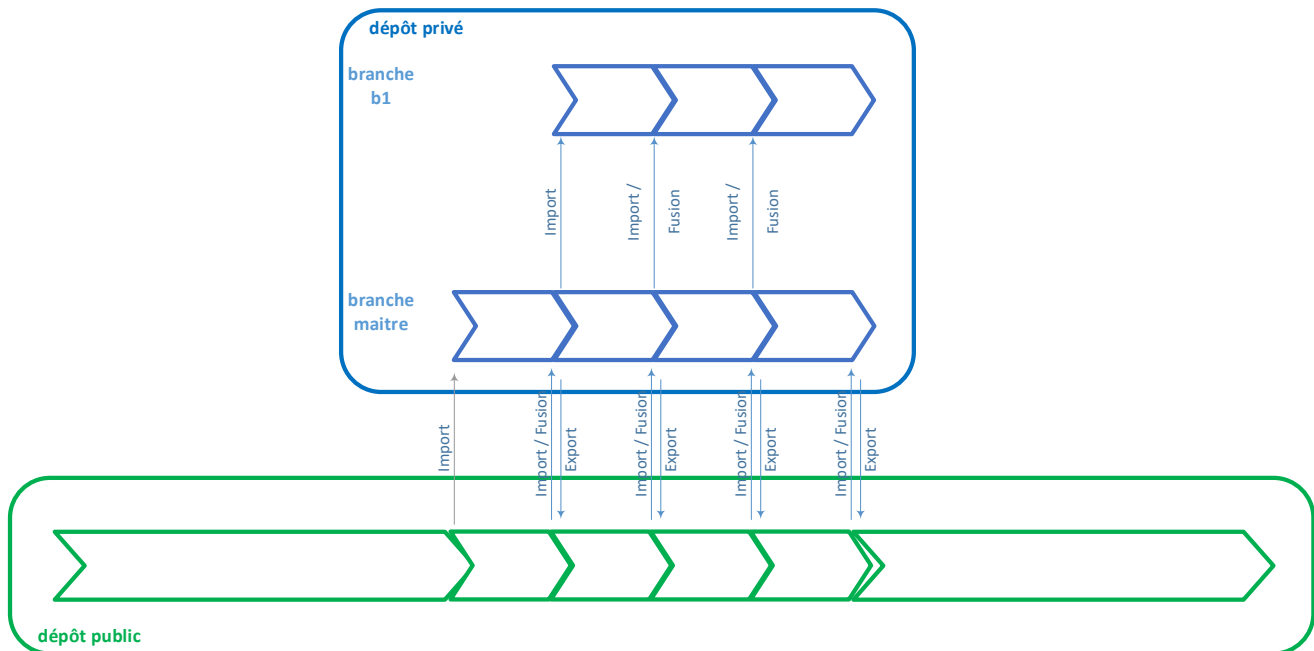
Dans le cas où l'organisation souhaite contribuer activement au projet open source, avec des mises à jour régulières de code vers le dépôt public, mais souhaite en même temps travailler sur le développement d'une



fonctionnalité potentiellement brevetable, on pourra alors employer une « branche » dédiée à cette fonctionnalité, au sein du dépôt privé.

Dans un système de gestion de versions, une branche est une succession de versions qui divergent de la ligne principale de développement (appelée alors maitre ou tronc). Des mises à jour sont possibles entre les différentes branches (ou avec le tronc), ce qui permet de développer une fonctionnalité sans impacter la ligne principale.

Figure 8 : Développement en utilisant un dépôt privé muni de plusieurs branches



En procédant ainsi, il est possible d'isoler le développement de la fonctionnalité à breveter, tout en bénéficiant du mode de développement coopératif du logiciel open source.

Historiquement ce mode de développement avec un dépôt public et un dépôt privé était complexe à mettre en place, les outils de gestion de versions habituellement utilisés<sup>67</sup> ne permettant pas la gestion de multiples dépôts. Les synchronisations entre dépôts devaient alors se faire manuellement.

Aujourd'hui, l'outil de gestion de versions le plus employé (Git) permet nativement la gestion et la synchronisation de dépôts multiples (c'est en partie pour cela qu'il a été créé). Le mode de développement décrit ici est donc pleinement compatible avec les outils maintenant employés.

## 2. Le délai de grâce

Si lors du développement du logiciel l'organisation n'a pas pris soin d'utiliser un dépôt de sources privé, il peut rester une possibilité de brevetabilité pour certains pays.

Quelques pays (en particulier les Etats-Unis, le Canada et le Japon) ont en effet institué une exception particulière au critère de nouveauté du brevet, sous la forme d'une période de temps antérieure au dépôt de

<sup>67</sup> Pour les logiciels Open Source, initialement l'outil CVS puis ensuite SVN

brevet, pendant laquelle une divulgation par l'inventeur ne peut être considérée comme opposable à la nouveauté. Cette période est appelée « délai de grâce ».

Cette particularité est généralement utilisée pour permettre la présentation de l'objet de l'invention à une conférence ou un salon professionnel avant le dépôt de brevet<sup>68</sup>.

Les principaux délais de grâce disponibles par le monde sont résumés dans le tableau suivant :

Pays	Délai	Point de départ	Actes émanant de l'inventeur donnant droit au délai de grâce	Formalités
<b>Brésil</b>	12 mois	Priorité la plus ancienne	Tout type de divulgation	aucune
<b>USA</b>	12 mois	Priorité la plus ancienne	Tout type de divulgation	aucune
<b>Mexique</b>	12 mois	Priorité la plus ancienne	Tout type de divulgation	déclaration obligatoire
<b>Argentine</b>	12 mois	Priorité la plus ancienne	Tout type de divulgation sauf demandes de brevet PCT	déclaration obligatoire
<b>Canada</b>	12 mois	Date de dépôt au Canada	Tout type de divulgation	aucune
<b>Corée du Sud</b>	12 mois	Date de dépôt en Corée du Sud	Tout type de divulgation (sauf publications de demandes de brevet Coréennes)	déclaration obligatoire
<b>Australie</b>	6 mois (spécifique) 12 mois (principale)	Date de dépôt en Australie	- spécifique : exposition internationale reconnue ou communication lue/ publiée par une société savante - principale : tout type de divulgation (sauf secondaire)	aucune sauf pour les divulgations à une exposition internationale reconnue (déclaration alors obligatoire)
<b>Japon</b>	6 mois	Date de dépôt au Japon	Tout type de divulgation (sauf publications de demandes de brevet Japonaises)	déclaration obligatoire

Crédit : Estelle Derval / Marks & Clerk - <https://bit.ly/2JKypRC>

Le fait que le délai de grâce soit peu répandu à travers le monde limite bien sûr grandement les possibilités d'extension internationale du brevet, mais certaines des zones accessibles peuvent présenter un fort intérêt commercial ou stratégique aux yeux des déposants.

Par exemple, une entreprise de nationalité américaine qui choisirait de déposer l'essentiel de ses brevets uniquement sur le territoire américain (même si elle vend à l'international<sup>69</sup>), peut bénéficier du délai de grâce, et ainsi effectuer des développements open source de manière usuelle sans que cela ne bloque sa stratégie de dépôts de brevets.

<sup>68</sup> Ou bien souvent, pour permettre tout de même d'effectuer le dépôt de brevet quand celui-ci n'a pas été effectué suffisamment tôt.

<sup>69</sup> Certaines entreprises font le raisonnement suivant : Leurs brevets servent principalement de levier défensif face à d'autres industriels. Or, le marché américain est suffisamment important pour que les autres industriels soient obligés de l'aborder. Les brevets américains seront alors suffisants pour être utilisés comme levier dans des négociations de licences croisées.

### III - Le partage de brevets

Nous avons vu précédemment que le modèle de développement open source engendre souvent un accord de licence de brevets de la part du contributeur, voire de la part de l'organisation qui redistribue le logiciel.

Ce mécanisme, cependant, ne permet pas de s'assurer dans tous les cas de l'obtention des licences de brevets nécessaires. Par exemple, les brevets détenus par des entités ne contribuant pas au logiciel ne se trouvent pas couverts. Même si l'intention d'un détenteur de brevet était de permettre son utilisation dans un certain logiciel, le mécanisme des licences open source ne permet pas de formaliser cette intention sans effectuer d'acte relevant du droit d'auteur ou du copyright (modification, redistribution).

Pourtant, certains détenteurs de brevets peuvent souhaiter permettre l'utilisation de leurs brevets dans le contexte des logiciels open source. Contreras (2013) identifie quatre classes de motivations qu'ont les brevetés pour partager de la sorte leurs brevets<sup>70</sup> :

- L'incitation - réduit les barrières liées aux brevets pour inciter les acteurs du marché à adopter une norme ou une plate-forme technologique particulière privilégiée par le garant.
- L'action collective - contribue à la réalisation d'un objectif collectif bénéfique pour le donateur.
- La contention volontaire - restreint la capacité du mandataire à faire valoir ses brevets, répondant souvent aux préoccupations des autorités gouvernementales.
- La philanthropie - fait avancer une cause sociale ou un autre bien public avec des avantages concomitants en matière de relations publiques pour le donateur.

Le panel des motivations possibles est assez large, et nous montre que souhaiter partager ses brevets ne relève pas forcément du pur altruisme : il est également possible d'en retirer des bénéfices économiques indirects.

Il est ainsi de plus en plus répandu que des entreprises souhaitent partager leurs brevets avec la « communauté » open source.

Un accord de licence (potentiellement à titre gratuit) pourrait être conclu entre le breveté et l'éditeur du logiciel s'il y en a un, mais lorsque le logiciel n'est pas soutenu par un éditeur, à qui accorder une telle licence ? De plus, si un détenteur de brevets souhaite accorder une licence pour tous les projets open source, il semble évident que contractualiser une licence avec chaque éditeur n'est pas une solution efficace.

D'autres mécanismes ont donc été créés pour permettre aux brevetés de laisser utiliser leurs brevets par les logiciels open source, tout en assurant les implémenteurs de cette permission (Belcher et Casey, 2016) (Volanis et Prentoulis, 2009).

---

<sup>70</sup> Contreras a établi cette liste pour le cas spécifique du *patent pledge*, mais les mêmes motivations sont également pertinentes pour les autres méthodes de partage « ouvert » de brevets.

## A - Patent pledges

Le premier mécanisme est celui du « *patent pledge* », qui pourrait se traduire en français par « serment de brevet » ou « promesse de brevet ».

Le *pledge* est un engagement public, de la part d'un détenteur de brevets, de laisser utiliser ses brevets dans un contexte donné. Dans le cas qui nous intéresse, le contexte est celui des logiciels open source, mais le mécanisme du *pledge* n'est pas limité à l'open source<sup>71</sup>.

La valeur d'une telle promesse peut sembler limitée du point de vue français (cet engagement n'étant pas un contrat mais une simple déclaration), mais est plus compréhensible dans les pays relevant de la *common law* (Maracke et Metzger, 2016), où le principe d'*estoppel* pourrait être appliqué à une telle déclaration, la rendant potentiellement applicable devant les tribunaux<sup>72</sup>.

Le *pledge* n'est cependant pas un abandon total de ses brevets. Il est le plus souvent assorti d'une liste précise de brevets, et comporte des clauses de défense, permettant au breveté de ne plus accorder le bénéfice de ses brevets à une entité si ses intérêts sont menacés par cette entité.

Une « menace » des intérêts peut bien sûr venir d'une procédure à l'encontre du breveté, mais aussi, en fonction des termes du *pledge*, par certaines menaces des intérêts économiques (par exemple un produit concurrent).

Un cas illustre cette limite : En 2010, un différend opposait les sociétés IBM et TurboHercules (éditeur d'un logiciel open source). Lors des échanges entre les deux sociétés, IBM a indiqué que TurboHercules pourrait contrefaire une liste de brevets appartenant à IBM. Or, il y avait dans cette liste des brevets faisant partie du *patent pledge* dont IBM avait annoncé faire bénéficier la communauté open source. IBM s'est justifié de la sorte :

*In 2005, when IBM announced open access to 500 patents that we own, we said the pledge is applicable to qualified open-source individuals or companies. We have serious questions about whether TurboHercules qualifies. TurboHercules is a member of organizations founded and funded by IBM competitors such as Microsoft to attack the mainframe. We have doubts about TurboHercules' motivations.*

Cet aspect n'a pas fait l'objet d'un jugement, mais montre tout de même les limites potentielles d'un *patent pledge*.

*Note : le lecteur souhaitant davantage de détails sur le système de patent pledge et ses évolutions récentes pourra se référer à Contreras (2018).*

<sup>71</sup> Voir par exemple le *patent pledge* de Tesla pour le domaine de l'automobile électrique : <https://bit.ly/2g7Gweb>

<sup>72</sup> Pour le moment le *patent pledge* semble être utilisé quasiment exclusivement par des entités américaines

## B - Open Invention Network

Le second moyen de partage de brevets dans le cadre de l'open source est l'adhésion à l'Open Invention Network (OIN)<sup>73</sup>.

L'OIN, créé en 2005, est un réseau de non-agression mutuel dédié au système d'exploitation Linux et à un certain nombre d'applications relatives à Linux<sup>74</sup>. Les membres de l'OIN s'engagent à ne pas utiliser leurs brevets à l'encontre d'une utilisation dans le cadre de systèmes Linux, et reçoivent en échange une licence couvrant :

- Les brevets détenus par ses membres pour une utilisation dans le cadre de Linux.
- Les brevets détenus en propre par l'OIN, sans limitation du cadre d'utilisation

Le réseau OIN rencontre un fort succès, comportant aujourd'hui plus de 3000 membres. Les membres présentent des profils divers tels que des acteurs de l'industrie « physique » (par exemple que Toyota), des sociétés du logiciel (Google, IBM, Microsoft ...), ou des projets open source.

L'arrivée, fin 2018, de Microsoft au sein de l'OIN marque la fin d'une époque (Jean, 2019). Il semble y avoir unanimité au sein de l'industrie du logiciel pour considérer Linux comme un élément stratégique dont il est devenu impossible de se passer, et aucun industriel ne souhaite plus attaquer le principe d'un système d'exploitation entièrement open source<sup>75</sup>.

*Note : Il existe d'autres systèmes de défense commune ou de mise en commun de brevets (comme par exemple RPX<sup>76</sup>, Unified Patents<sup>77</sup> ou LOT<sup>78</sup>), mais ceux-ci ne sont pas spécifiques à l'open source.*

## C - Licences ouvertes de brevets

Le dernier mécanisme permettant le partage de brevets ressemble à une combinaison entre les mécanismes de *patent pledge* et de licence open source : il s'agit des licences ouvertes de brevets.

Selon le principe de la licence ouverte de brevets, le breveté s'engage, selon les termes d'une licence spécifique, à ne pas utiliser ses brevets à l'encontre des autres bénéficiaires de cette même licence, sauf pour assurer sa propre défense.

---

<sup>73</sup> <https://www.openinventionnetwork.com>

<sup>74</sup> Les applications considérées sont assez diverses, et peuvent aussi être disponibles pour d'autres systèmes d'exploitation que Linux. Il y a par exemple Apache (serveur Web), Eclipse (environnement de développement), Firefox (navigateur), GIMP (retouche photo), et OpenOffice.org (suite bureautique).

<sup>75</sup> En revanche, une entité tirant la majorité de ses revenus d'activités de valorisation de brevets pourrait avoir une position différente.

<sup>76</sup> [www.rpxcorp.com](http://www.rpxcorp.com)

<sup>77</sup> <https://www.unifiedpatents.com>

<sup>78</sup> <https://lotnet.com>

A la différence du *patent pledge*, la licence est réciproque. L'utilisation d'un texte de licence commun à tous les membres permet une plus grande lisibilité dans les droits et obligations résultant de cette unique licence, au contraire des *patent pledges*, qui ont une rédaction spécifique à chaque entité

Les caractéristiques indiquées jusqu'ici sont celles du pool défensif de brevets. En 2015, Mozilla a étendu ce schéma par deux caractéristiques, afin de créer la Mozilla Open Software Patent Licence (MOSPL) :

- La licence permet explicitement l'utilisation des brevets par les logiciels open source
- La licence comporte une caractéristique virale.

La viralité de cette licence, inspirée des clauses de propagation des licences open source, tient à l'acceptation automatique de cette licence lorsque l'on réalise une action qui relèverait de la contrefaçon à défaut d'acceptation de la licence. En d'autres termes, l'utilisation d'un logiciel couvert par un brevet entraîne l'acceptation de la licence, à moins d'avoir un autre mécanisme de licence sur ce brevet.

En pratique la viralité ne semble pas encore avoir eu d'effet visible : Mozilla a initié le système avec ses propres brevets, mais ceux-ci sont peu nombreux (cinq) et très spécifiques. Ils sont actuellement tous relatifs à des technologies d'encodage audio/vidéo, qui sont déjà incluses dans des standards ouverts et libres de redevances (principalement Opus et AV1). En conséquence, l'utilisation de ces brevets dans le cadre de ces standards est déjà couverte par une licence, et le caractère viral de la MOSPL ne se déclenche pas.

Un détenteur de brevet peut aussi joindre cette licence en réalisant une déclaration publique, mais à notre connaissance cela ne s'est pas encore produit.

En conséquence, cette initiative ne semble pas encore avoir eu, à ce jour, d'impact conséquent sur le marché.



## TROISIEME PARTIE : OBSERVATIONS DE TERRAIN

Nous présentons dans cette troisième partie des observations relatives aux interactions entre brevets et open sources, issues des pratiques constatées.

Tout d'abord, nous nous pencherons sur les portefeuilles de brevets des sociétés centrées sur l'open source, puis nous traiterons de deux domaines technologiques utilisés ici comme exemple : l'encodage MP3, et un sous-ensemble de l'IoT<sup>79</sup>.

### I - Portefeuilles brevets des entreprises travaillant exclusivement dans l'open source

Cette section fournit la liste des sociétés du secteur open source réalisant plus de 100 millions de dollars de chiffre d'affaire en 2018.

Cette liste a été initialement créée<sup>80</sup> par Joseph Jacks, de la société d'investissements OSS Capital. Nous avons complété cette liste par les données relatives aux brevets, qui ont été obtenues par le biais du logiciel Orbit, de la société Questel.

#### A - Données

Société	Familles de brevets	Familles de brevets en vigueur	Employés	Revenu annuel estimé (millions)	Valorisation estimée (milliards)	Domaine technologique	OSS License
Acquia	0	0	900	\$200	\$3.0	Web Content Mgmt System	GPL v2
Alfresco	19	18	460	\$100	\$0.3	ECM <> BPM	LGPL v3
Automattic	1	0	830	\$300	\$3.0	Content Management System	GPL v2
Canonical	5	3	630	\$150	\$1.5	Operating System (Linux)	GPL

<sup>79</sup> Le terme IoT est expliqué plus loin dans la section y étant relative.

<sup>80</sup> Liste initiale disponible à l'adresse <http://oss.cash>



Chef	?	?	350	\$100	\$1.0	Configuration Management	Apache 2.0
Cloudera	32	31	3200	\$500	\$1.7	Big Data / Hadoop Ecosystem	Apache 2.0
Coinbase	8	8	1100	\$1,500	\$8.0	Cryptocurrency Wallet/Exchange/Platform	MIT/A2.0
Confluent	3	1	800	\$150	\$2.5	Big Data / Middleware / Streaming	Apache 2.0
Databricks	8	8	840	\$250	\$3.5	Big Data / Hadoop Ecosystem	Apache 2.0
Datastax	8	7	730	\$150	\$2.5	NoSQL Database	Apache 2.0
Docker	0	0	500	\$120	\$2.5	Developer / Ops / Infra Tools	Apache 2.0
Elastic	12	11	1600	\$400	\$5.0	Distributed Search Index + Tools	Apache 2.0
Fastly	46	44	550	\$200	\$2.0	CDN	BSD
ForgeRock	0	0	550	\$120	\$1.5	Identity and Access Management	CDDL
GitHub	9	9	1550	\$400	\$7.5	Distributed Source Code Version Control	GPL v2
HashiCorp	0	0	560	\$250	\$2.5	Developer / Ops / Infra Tools	MPLv2
Hortonworks	4	4	1500	\$400	\$1.5	Big Data / Hadoop Ecosystem	Apache 2.0
Instructure	1	0	1420	\$220	\$1.4	Learning Management	AGPL v3
JetBrains	0	0	750	\$300	\$3.0	Developer Tools	Apache 2.0
JFrog	0	0	460	\$150	\$1.5	Software Artifact Repository	AGPL v3
Kaltura	6	3	480	\$200	\$1.5	Video Editing Platform	AGPL v3
Liferay	2	2	870	\$250	\$2.5	Enterprise Portal	LGPL v2.1+
Magento Commerce	0	0	700	\$200	\$1.7	Web Content Mgmt System	OSL v3, AFL v3
Mapbox	17	17	450	\$100	\$1.5	Mapping Software	BSD-3
Mirantis	0	0	500	\$100	\$1.0	Infrastructure Software	Apache 2.0
MongoDB	15	15	1500	\$400	\$9.0	NoSQL Database	SSPL
Mozilla Corporation	5	5	1100	\$550	\$5.0	Web Browser	MPLv2

MuleSoft	4	4	1750	\$700	\$8.0	Middleware	CPAL
MySQL AB	8	4	800	\$1,000	\$1.1	Relational Database	GPL v2
Neo4j	2	2	300	\$100	\$1.0	Graph Database	GPLv3
Nicira	2	2	100	\$2,000	\$1.3	SDN / Network Virtualization	Apache 2.0
Odoo	0	0	750	\$100	\$1.0	Business Applications	LGPL v3
Pentaho	2	0	670	\$150	\$1.0	BI/ETL	Apache 2.0
Pivotal	110	110	2400	\$800	\$3.5	PaaS / Hadoop / Spring	Apache 2.0
Puppet Labs	0	0	560	\$250	\$2.5	Configuration Management	Apache 2.0
Red Hat	2714	2613	13100	\$3,500	\$34.5	OS, Middleware, Infrastructure Software	GPL v2
Smile	?	?	1600	\$100	\$1.0	Integrator	Many
Sourcefire	18	17	600	\$500	\$3.0	Network Intrusion Detection	GPL v2
SugarCRM	101	84	420	\$150	\$1.0	CRM	Apache 2.0
SUSE	25	22	1500	\$450	\$2.5	Operating System (Linux)	GPL v2
Talend	2	2	1300	\$200	\$2.0	SOA/ETL/AI/Middlewar e	Apache 2.0
WordPress Engine	0	0	700	\$150	\$1.0	Content Management System	GPL v2

## B - Analyse

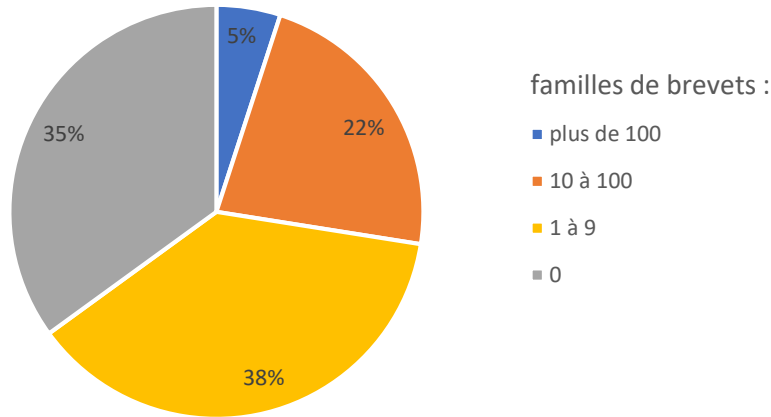
Afin d'analyser la présence de brevets dans l'écosystème open source, nous avons pris comme base la liste des sociétés réalisant plus de 100 millions de dollars de chiffre d'affaire annuel dans l'open source. Sur ces 42 sociétés, nous avons pu obtenir les portefeuilles de brevets de 40 d'entre elles.

Sur ces 40 sociétés, 29 ont déposé des brevets, ce qui représente 73% de ces 40 sociétés.

Seules 26 sociétés ont un portefeuille encore actif (ayant toujours au moins un brevet en vigueur), soit 65%. Ceci signifie que 8% des sociétés, bien qu'ayant déposé au moins un brevet, soit n'ont pas pu maintenir et/ou développer leur portefeuille, soit n'ont pas souhaité poursuivre dans cette voie.

Dans l'ensemble les portefeuilles sont relativement petits, 38% des sociétés ayant moins de 10 familles de brevets actifs. Seules 5% des sociétés ont plus de 100 familles actives :

Figure 9 : Tailles des portefeuilles de brevets des plus grosses sociétés open source



Les portefeuilles encore en vigueur ne montent qu'à une centaine de familles (110 familles pour la société Pivotal), à une exception près : Red Hat, qui fait en effet figure d'anomalie dans ce paysage, avec plus de 2600 familles de brevets en vigueur.

Le portefeuille de brevets de Red Hat participe possiblement à l'extraordinaire valorisation de cette société, mais au vu des données recueillies, nous n'observons cependant pas de corrélation statistique entre taille de portefeuille brevets et valorisation ou chiffre d'affaire<sup>81</sup>.

Nous pouvons en tout cas constater que les plus grosses sociétés du monde open source ne sont pas étrangères aux brevets, les deux tiers d'entre elles détenant des brevets.

*Note : Dans le domaine de l'open source, Red Hat est une société exceptionnelle<sup>82</sup>. Lorsque l'on compare ses métriques, elle devance le second du classement (pour chaque métrique)*

- de 24 fois sur le nombre familles de brevets en vigueur
- de 5 fois sur le nombre d'employés
- de 1,75 fois sur le chiffre d'affaire annuel
- de 4,6 fois sur la valorisation

<sup>81</sup> Le cas Red Hat étant isolé, on ne peut bien sûr pas s'appuyer sur une seule société pour dégager une tendance statistique.

<sup>82</sup> Fin 2018, IBM a annoncé son intention d'acquérir Red Hat, pour 34 milliards de dollars. Il s'agira de la plus importante acquisition jamais réalisé par IBM.

## II - Exemples d'interactions entre open source et technologies brevetées

### A - L'encodage MP3

#### 1. La technologie

Ce qui est couramment appelé format « MP3 » est un format de compression audio défini en 1992, et ratifié en 1993, par un organisme international de standardisation (*l'International Organisation for Standardization* – ISO). Le standard, établi par un groupe d'experts (universitaires et industriels), détermine ce qu'est le format MP3, et la façon de le décoder.

Le procédé d'encodage n'est pas standardisé, bien qu'un exemple soit fourni (à titre informatif) par le standard. En ne standardisant pas le procédé d'encodage, l'ISO a fait en sorte que les fournisseurs d'encodeurs puissent se distinguer les uns des autres (certains fournissant par exemple des encodeurs plus rapides, d'autres plus qualitatifs), et que les encodeurs puissent bénéficier des apports ultérieurs de la recherche dans ce domaine.

Nous pouvons faire l'analogie avec une langue, définie par son vocabulaire et sa grammaire : tout un chacun sachant comment interpréter grammaire et vocabulaire (le décodage) comprendra un texte (le flux encodé) par différents auteurs (les encodeurs). Proust et Balzac écrivaient tous deux en français, et pourtant leurs styles sont bien différents. De la même façon, il est possible d'avoir différents encodeurs MP3, qui bien qu'ils produisent des fichiers au même format, se démarquent par leurs différentes qualités intrinsèques.

#### 2. Les brevets relatifs à la technologie

Aucun des brevets relatifs au format MP3 n'est spécifique aux logiciels : il s'agit de brevets de procédés relatifs au traitement de signaux, et d'émetteurs/récepteurs/systèmes réalisant ces procédés. Ils s'appliquent donc aussi bien aux implémentations réalisées par des composants électroniques qu'aux implémentations logicielles.

Ces brevets peuvent être classés en deux catégories :

- Les brevets essentiels : un brevet est dit « essentiel » s'il couvre une technique rendue obligatoire par la norme. L'enseignement du brevet (c'est-à-dire ses revendications) est implémenté par tout système conforme à la norme, ces brevets ne peuvent pas être évités<sup>83</sup>.
- Les brevets non-essentiels : ces brevets couvrent des raffinements et améliorations possibles, qui ne sont pas nécessairement implémentés par un système conforme à la norme<sup>84</sup>.

Il va de soi que les brevets essentiels ont une plus grande valeur économique que les brevets non-essentiels, c'est pourquoi les détenteurs de brevets proposent le plus souvent uniquement des licences de « packages », leurs permettant de licencier à tous des brevets non-essentiels, qu'ils soient utilisés ou non dans une implémentation particulière.

---

<sup>83</sup> Exemple par analogie avec la langue française : toute phrase commence par une majuscule et finit par un point

<sup>84</sup> Exemple par analogie avec la langue française : l'usage de l'euphémisme (« Ce brevet, inventif, ... »)

Le domaine technologique étant fort restreint, il en est de même des brevets, qui ne représentent que quelques dizaines de familles de brevets. Les brevets identifiés comme étant relatifs au format MP3 sont les suivants :

### **Les brevets licenciés par Technicolor**

Il s'agit des brevets détenus par Technicolor<sup>85</sup> et par l'institut Fraunhofer<sup>86</sup>, Technicolor s'occupant de la gestion des licences conjointes pour les deux entités. Il s'agit du premier portefeuille sur lequel des licences ont été proposées, des licences ayant été réclamées pour les implémentations logicielles à partir de 1998. Ce n'est donc que cinq ans après la ratification du standard que les implémentateurs ont pu apprendre qu'il y avait un coût de licence.

Ces demandes de royalties ont eu en 1998 pour effet de stopper le développement et la distribution de la quasi-totalité des encodeurs mp3 alors disponibles gratuitement.

C'est en 2017 que ce programme de licensing a pris fin, avec l'expiration du dernier brevet essentiel de ce portefeuille<sup>87</sup>.

### **Les brevets licenciés par Sisvel**

Sisvel est un administrateur de pools de brevets (ensemble de titres possédés par différents brevetés, disponibles sous une licence groupée), dont l'un dédié au MP3. Sisvel agit en tant qu'entité de valorisation de brevets, n'étant ni un industriel ni un laboratoire de recherche, mais permettant à ces derniers d'obtenir un retour sur investissement.

C'est à partir de 2005 que Sisvel s'est réellement fait connaître au sujet des licences sur le MP3, à la suite d'actions de monétisation particulièrement efficaces d'un point de vue médiatique : ils ont tout d'abord intenté une action en contrefaçon à l'encontre de Technicolor, ce qui a permis aux industriels du secteur de découvrir qu'une licence obtenue auprès de Technicolor ne couvrait pas l'intégralité des brevets.

A partir de 2006, ils ont régulièrement déclenché des saisies douanières lors des salons professionnels tenus en Allemagne<sup>88</sup>. L'arrivée deux cents douaniers, armés, réalisant des saisies de centaines de produits sur des stands de dizaines d'exposants<sup>89</sup>, devant tous les journalistes du secteur, a démontré une excellente maîtrise de la médiatisation de la part de Sisvel<sup>90</sup>.

---

<sup>85</sup> Technicolor s'appelait à l'époque Thomson

<sup>86</sup> Institut de recherche Allemand

<sup>87</sup> Nous pouvons constater que les brevetés sont parvenus à avoir des brevets essentiels n'expirant que 25 ans après la finalisation de la norme. C'est principalement grâce à la législation américaine qui permet dans certains cas d'avoir une durée de validité du brevet s'étendant au-delà de 20 ans.

Les plus longues extensions proviennent des brevets américains déposés jusqu'en juin 1995 dont la fin de validité est soit de 20 ans après le dépôt initial, soit de 17 ans après l'obtention du brevet. Une « astuce » consistait alors à faire durer l'examen le plus longtemps possible pour prolonger la durée du brevet.

<sup>88</sup> <https://bit.ly/2TrQVks>

<sup>89</sup> Ces pratiques agressives ont valu le qualificatif de « *patent troll* » à Sisvel, mais l'on peut s'interroger sur le fait qu'il soit juste ou non de reprocher des pratiques visant à défendre l'intérêt des brevetés ayant contribué à l'élaboration d'une norme (Ann, 2009)

<sup>90</sup> <https://bit.ly/2KFmzr8>

## **Le portefeuille Alcatel-Lucent**

Entre 2003 et 2008, Alcatel-Lucent et Microsoft se sont opposés au travers d'une série d'actions en contrefaçon. L'un de ces procès a vu Microsoft être condamné, en 2007, à payer une indemnité de 1.5 milliard de dollars pour la contrefaçon de deux brevets relatifs au MP3, détenus par Alcatel-Lucent. Il s'agissait alors de la plus importante indemnité jamais obtenue à l'issue d'un procès en contrefaçon. Le jugement a finalement été annulé, et les deux parties sont parvenues en 2008 à un accord mettant fin à la plupart de leurs conflits.

A notre connaissance, Alcatel-Lucent n'a pas fait valoir ses droits dans d'autres affaires.

## **Le brevet Hybrid Audio**

A partir de 2010, la société Hybrid Audio a commencé à intenter des actions en contrefaçon à l'encontre de divers industriels, sur la base d'un brevet (US RE40'281), non spécifique au MP3, mais qui peut s'y appliquer. La société Hybrid Audio semble avoir été créée dans le seul but de valoriser ce brevet, acquis simultanément à la première action en contrefaçon. Hybrid Audio n'a pas d'autres actifs que ce brevet, et sa monétisation est la seule activité de cette société.

Ce brevet a expiré en 2019, 27 ans après sa date de première priorité<sup>91</sup>.

## **3. Les projets open source**

### **8Hz MP3 encoder**

Le projet 8Hz est le plus ancien encodeur MP3 open source. La version initiale est sortie en 1998, peu de temps avant que Technicolor/Fraunhofer ne commencent à demander des royalties. Le projet 8Hz, qui proposait à la fois du code source et des versions compilées, a rapidement été arrêté du fait de ces demandes de royalties.

### **Blade MP3 encoder**

Blade n'était pas initialement un projet open source. Initialement le fruit d'un seul auteur, la première version de cet encodeur a été diffusée en 1998, peu de temps avant les premières demandes de redevances. En conséquence de ces demandes, l'encodeur est devenu open source et la diffusion de version compilées a été stoppée, afin de ne plus fournir que du code source.

Ayant attiré peu de contributeurs, le projet a été abandonné en 2001.

### **LAME MP3 encoder**

LAME<sup>92</sup> est un projet communautaire<sup>93</sup> initié en 1998, toujours développé 20 ans après. C'est en partie en réaction à la demande de royalties de la part de Technicolor/Fraunhofer que le projet LAME a été initié, en prenant garde à ne pas distribuer de version compilée de l'encodeur.

Pleinement conscient de l'existence de brevets potentiellement implémentés par l'encodeur, les membres du projet ont toujours communiqué clairement sur ce point :

---

<sup>91</sup> Par une curieuse coïncidence, le brevet a expiré au jour de l'écriture de la présente section de ce mémoire, le 13 août 2019.

<sup>92</sup> <http://lame.sourceforge.net/>

<sup>93</sup> Développé et maintenu par des bénévoles.

- Les sociétés souhaitant inclure LAME sous forme compilée dans leurs produits (afin de permettre l'encodage au format MP3) sont invitées à consulter un juriste compétent pour évaluer leur situation au regard des brevets.
- Les utilisateurs (et intégrateurs) sont informés de l'existence de brevets couvrant potentiellement l'implémentation réalisée par LAME.

#### 4. Effet des brevets sur les projets open source

Les demandes de royalties de la part des brevetés ont permis de stopper la diffusion gratuite d'encodeurs sous forme compilées (que les encodeurs soient open source ou non), mais n'ont pas pour autant stoppé le développement d'encodeurs open source. Assez rapidement, les développeurs open source (qui dans le domaine de l'encodage MP3 agissent en tant que volontaires, et non pas en tant que salariés) se sont adaptés, afin de ne plus diffuser que du code source.

Il est manifeste que le fait qu'un encodeur soit uniquement disponible sous forme de code source est un frein important pour la diffusion auprès du grand public. Ceci ne pose cependant pas de problèmes pour l'intégration dans des logiciels tiers. En pratique, lorsque les brevets étaient toujours en vigueur, les sociétés souhaitant offrir une fonction d'encodage MP3 dans leurs produits avaient le choix entre :

- Acheter une implémentation commerciale, fournie avec une licence sur certains brevets. Par exemple, Thomson proposait une licence, par unité, de 2.5\$ pour son implémentation, et de 2.5\$ pour ses brevets (soit un total de 5\$).
- Utiliser un encodeur open source, et obtenir par ailleurs les licences de brevets. Dans la quasi-totalité des cas il s'agit de l'encodeur LAME, celui-ci ayant rapidement surpassé les quelques autres encodeurs, dont le développement a rapidement été arrêté.

Cette dernière solution s'avère plus économique, la société intégratrice n'ayant à payer que la licence de technologie (brevets) mais économisant la licence d'implémentation.

Les brevetés n'ont jamais manifesté d'hostilité à cette pratique. Certes, l'existence de LAME leur a causé un manque à gagner sur les licences d'implémentation, mais en abaissant le coût total d'accès, LAME a contribué à étendre le marché : avec 2.5\$ de plus nécessaires pour l'implémentation, certaines sociétés auraient choisi soit de ne pas fournir la fonctionnalité, soit de fournir une fonctionnalité similaire mais utilisant d'autres technologies d'encodage que le MP3. Il est possible d'émettre l'hypothèse que les bénéfices retirés de l'extension du marché aient été plus grands que le manque à gagner sur les ventes d'implémentations.

Les sociétés fournissant des logiciels gratuitement ont pour leur part bien souvent choisi de ne pas fournir la fonctionnalité. C'est par exemple le cas des distributions Linux, qui n'ont intégré LAME qu'à partir de 2017, à l'expiration des portefeuilles gérés par Technicolor et Sisvel.

L'incertitude quant aux redevances nécessaires, causée par les demandes tardives de certains détenteurs de brevets (le dernier connu ayant commencé à demander des redevances 18 ans après la finalisation du standard), n'a pas impacté les logiciels open source plus qu'elle n'a impacté les logiciels commerciaux : dans la plupart des cas, les actions en contrefaçon sont intentées au motif de la diffusion de produits intégrant une fonctionnalité d'encodage MP3, et non pas à l'encontre de fournisseurs d'encodeurs (qui sont ensuite intégrés dans d'autres logiciels). Le fait que l'encodeur utilisé soit open source ou non n'a donc pas d'impact quant aux actions en contrefaçon.

*Note : L'incertitude quant aux redevances nécessaires et les demandes parfois tardives de redevances sont représentatives de ce qui se pratique dans le domaine de l'encodage audio/vidéo, et la situation n'est pas très différente pour les standards actuels.*

*Par exemple pour le standard de compression vidéo HEVC, des redevances sont actuellement demandées par trois pools différents, ainsi que par des brevetés n'ayant joint aucun pool. Certains brevetés sont même membres de plusieurs pools, demandant des montants différents de redevances pour les mêmes brevets.*

Du point de vue de la diffusion de la connaissance, les brevets ont joué un rôle positif : ils font partie des sources d'informations utilisées par les membres du projet LAME<sup>94</sup>. Les brevets américains se sont avérés particulièrement utiles, comprenant souvent des informations plus précises dans leur description que les brevets européens. (Une explication possible est l'obligation de divulgation du « *best mode* » dans les brevets américains, encourageant à une divulgation plus détaillée.)

## **B - L'IoT (Internet of Things)**

### **1. Le domaine technique**

Le terme IoT (*Internet of Things*) a une portée assez vaste, qu'il convient de préciser.

Dans l'acronyme IoT, « *Internet* » fait référence aux réseaux comprenant des éléments potentiellement éloignés les uns des autres, sans qu'il n'y ait nécessairement un passage par internet. Le mot « internet » est ici utilisé de façon imagée, et pas nécessairement de façon littérale. « *Things* » fait référence aux objets, par opposition aux humains, indiquant que la finalité est la communication entre objets (par exemple des capteurs et de actionneurs), et non pas la communication à destination des humains (comme les e-mails ou les sites webs).

L'IoT utilise de nombreux protocoles de communications, qui diffèrent en fonction des contraintes et des usages, mais qui peuvent également être en concurrence sur des cas d'applications similaires ou identiques.

### **2. Les brevets relatifs à la technologie**

Le domaine de l'IoT est extrêmement vaste, et le marché potentiel est colossal : l'institut d'études économiques McKinsey estime que l'impact économique de l'IoT atteindrait 2 700 à 6 200 milliards de dollars en 2025 (McKinsey, 2013). Le nombre de brevets est en rapport avec l'enjeu : en 2018, le nombre de brevets relatifs à l'IoT était estimé à 178 000 (Relecura, 2018).

---

<sup>94</sup> Les autres sources d'information principales sont les articles scientifiques, les thèses universitaires, et les codes sources d'autres encodeurs audio numériques (encodeurs MP3 et encodeurs d'autres formats).



Les dix plus gros détenteurs de brevets du domaine sont les suivants :

Détenteur	Nombre de brevets (dépôts + accordés)
Samsung	10 807
Qualcomm	6 132
Intel	4 497
Nokia	3 149
Huawei	2 904
Ericsson	2 784
LG	2 666
Sony	2 414
ZTE	2 005
Philips	1 657

L'augmentation des investissements est manifeste depuis ces dix dernières années, comme en atteste l'augmentation annuelle du nombre de dépôts.

Figure 10 : Croissance du nombre de demandes brevets publiées dans l'IoT



Source : Relecura

Note : Les données ont été obtenues cours d'année 2018, cette dernière année n'est donc pas complète.

### 3. Les projets open source – *frameworks*<sup>95</sup> de communication IoT

Les domaines de l'IoT étant extrêmement vastes, nous nous concentrons dans cette section uniquement sur les projets open source de *frameworks* de communication IoT. Ces *frameworks* permettent la découverte et la gestion des objets connectés, ainsi que la communication entre objets (ces objets pouvant utiliser des protocoles de communication différents, qui ne sont pas nécessairement compatibles entre eux).

Les principaux projets open source de *frameworks* IoT sont portés par les industriels, qui y voient un moyen de porter la vente d'objets et de solutions de gestion associées, un *framework* interopérable et utilisé par plusieurs industriels étant un socle favorable au développement du marché. Le domaine est cependant en construction, et les alliances industrielles sont encore mouvantes, même si une stabilisation semble apparaître.

Pour la plupart des industriels du secteur, le grand nombre de brevets semble relever d'une stratégie de dissuasion. Afin de pouvoir développer une activité, la tendance principale est d'obtenir des accords de non-agression par le biais de *patent pledges*, comme nous pouvons le voir au travers des principaux *frameworks* open source de ces dernières années :

#### **OpenDOF**

Issu de 20 ans de développements, Panasonic a lancé en 2015 le projet OpenDOF. Ce *framework* est utilisé par Panasonic pour gérer ses propres déploiements de l'IoT dans le domaine industriel (panneaux solaires, éoliennes, climatiseurs, ...)

Ce projet open source est ouvert à tous, mais les contributeurs doivent accepter un *patent pledge* : une licence gratuite de brevet est donnée par le contributeur, couvrant sa propre contribution de code source.

Il semble que ce projet n'a pas su fédérer les autres industriels, seuls les employés de Panasonic y ayant pour le moment contribué.

#### **AllJoyn**

Ce *framework* a été initié par Qualcomm<sup>96</sup> en 2013, qui a créé pour l'occasion l'alliance AllSeen, dans laquelle il a été rejoint par plus de 100 membres.

Début 2015, l'alliance AllSeen a adopté une règle de *patent pledge* (similaire à celle du projet OpenDOF), afin de lever les incertitudes sur les brevets des contributeurs. Une licence gratuite de brevets est donnée pour tout produit ayant passé la certification AllSeen. Les brevets sont donc utilisés pour encourager l'interopérabilité des produits, et la liberté accordée est donc en quelque sorte « contrôlée ».

Fin 2016 l'alliance AllSeen a été intégrée dans le consortium OCF (voir ci-après). La dernière version de AllJoyn est également sortie fin 2016, et il est peu probable qu'une nouvelle version majeure soit diffusée dans le futur.

---

<sup>95</sup> Cf. définitions en annexe 1.

<sup>96</sup> Qualcomm est une entreprise fournissant des puces et solutions de communication, principalement dans les technologies sans fil.

## IoTivity

IoTivity est le *framework* open source promu par le consortium OCF (Open Connectivity Foundation). La première version est sortie fin 2015, et il est toujours développé. Ce *framework* implémente les spécifications OCF, approuvées fin 2018 en tant que standard par l'ISO (International Organization for Standardization).

IoTivity est indirectement protégé par un patent pledge : pour devenir membre de l'OCF, il faut s'engager dans un patent pledge qui protège les spécifications : toute implémentation du standard OCF ne peut faire l'objet d'action en contrefaçon de la part des membres de l'OCF (aujourd'hui plus de 300 sociétés). IoTivity bénéficie donc de cet accord, mais une autre implémentation de ce standard en profiterait aussi, qu'elle soit ou non open source.

*Note :*

*Tous les membres de l'industrie ne sont pas en accord avec ce principe de patent pledge. En particulier, Broadcom<sup>97</sup>, qui était membre fondateur du consortium OIC<sup>98</sup>, a quitté le consortium du fait de la condition de patent pledge, ne souhaitant pas mettre ses brevets gratuitement à disposition des autres membres.*

---

<sup>97</sup> Broadcom est un fournisseur de semiconducteurs.

<sup>98</sup> OIC : Open Interconnect Consortium. Le consortium OIC a été transformé pour établir l'OCF en 2016.

## CONCLUSION

Nous nous sommes interrogés dans la présente étude sur les possibilités de combiner une stratégie open source avec un portefeuille de brevets. Nous avons tout d'abord traité des particularités de l'open source pouvant impacter le traitement de cas de contrefaçon de brevet par des logiciels, puis avons étudié ce que les brevets peuvent apporter à un développement de logiciel open source.

Il en est ressorti que le développement ou la participation au développement de logiciels open source peut impacter et porter atteinte à la valeur d'un portefeuille de brevets. Il convient donc d'être prudent lors de la combinaison des deux, afin de s'assurer de faire des choix éclairés et d'agir en connaissance de cause. De manière inverse, les brevets de tiers peuvent bien sûr impacter la liberté d'implémentation de logiciels. Les logiciels open source ne font pas exception à cette règle, et peuvent être impactés par des brevets tiers, obligeant à des stratégies de contournement pour éviter la contrefaçon.

Pour éviter de se retrouver exclu du marché, ou de devoir assumer un coût de licence qui ne soit pas supportable par le modèle économique du développement open source, nous avons pu établir qu'il est devenu quasiment nécessaire, même pour développer des logiciels open source, d'avoir une stratégie de dépôt de brevet. Les tactiques doivent être réfléchies, préparées et planifiées. Quelques dépôts opportunistes ont peu de chances d'être suffisants pour bâtir un plan d'action basé sur des brevets.

Il semble, malheureusement, que les sociétés dont l'open source est le cœur de métier ne soient, pour l'essentiel, pas encore prêtes à utiliser les brevets pour bâtir un plan d'action. La prise de conscience d'une telle nécessité est en train de murir, les différents acteurs n'en étant pas aux mêmes stades. De manière générale, même chez les acteurs avancés dans leur stratégie, les portefeuilles de brevets semblent faibles comparés à ceux du reste de l'industrie du logiciel<sup>99</sup>.

La défense, sur le plan des brevets, des logiciel open source, semble plutôt actuellement assurée par les acteurs du logiciel qui se sont « convertis », par pragmatisme industriel, à l'open source au cours de ces dix ou quinze dernières années. Ils s'appuient sur l'open source en parallèle de leur activité de développement propriétaire. Ces entreprises importantes, telles que IBM, Google ou Microsoft, ont d'énormes portefeuilles de brevets, et semblent actuellement engagés dans la défense de l'open source à travers une stratégie qui ressemble à celle de la dissuasion nucléaire : il s'agit de s'armer massivement de brevets, qui sont destinés à une potentielle (mais non souhaitable) riposte.

Pourtant, le but sociétal ayant mené à la création du système de brevets est bien le même que celui ayant présidé à la création de la mouvance open source : il s'agit de promouvoir le progrès pour que le plus grand nombre puisse en bénéficier, en s'assurant que les nouvelles connaissances soient mises à la disposition de tous. Ainsi, même si les moyens employés sont bien différents (monopole d'exploitation temporaire pour le brevet,

---

<sup>99</sup> A notable exception près : Red Hat

partage systématique pour l'open source), les idéaux originaux sont très similaires. De même, le droit moral de reconnaissance est une valeur forte partagée entre brevet (droit moral de l'inventeur à figurer sur le titre, de manière inaliénable) et logiciel open source (où l'auteur est crédité de façon publique).

Nous pouvons également constater que brevet et open source sont complémentaires du point de vue de la diffusion des connaissances. Le logiciel open source est mis à jour si une amélioration est jugée utile, et contient ainsi en permanence le meilleur mode de réalisation. Le brevet, au contraire, est un document figé : Il ne décrit pas nécessairement le mode de réalisation jugé optimal au moment de son dépôt<sup>100</sup>. Quand bien même il le contiendrait, il ne peut décrire le travail d'améliorations itératives éventuellement apportées. A son crédit, le brevet a l'avantage de sa description. Longue, décrivant à la fois l'état de l'art et les apports de l'invention, cette description est incontestablement plus porteuse d'informations et de savoir que les quelques maigres commentaires présents dans la plupart des logiciels open source. Par leurs formes différentes, brevet et open source se supplémentent ainsi efficacement pour diffuser le savoir.

Avec de telles complémentarités et commonalités dans leurs buts, il est légitime de s'interroger sur l'opposition perçue entre brevet et open source : pourquoi un tel antagonisme, alors que les deux outils ont été créés pour atteindre le même idéal ?

Nous pouvons nous demander si les opposants au « brevet logiciel », particulièrement virulents en Europe, ne se trompent pas partiellement de combat, en se positionnant par rapport au seul point du brevet logiciel au lieu d'avoir une vue plus globale. Si par « brevet logiciel » nous devons comprendre les brevets dont le seul aspect novateur serait de rajouter la caractéristique « par ordinateur » à un procédé auparavant connu, il est effectivement difficile de leur donner tort. Il semble bien que le revirement des Etats-Unis à ce sujet, opéré depuis la décision « Alice »<sup>101</sup> en 2014, ait été grandement bénéfique, en réduisant significativement le nombre de procès en contrefaçon intentés sur la base de tels brevets (Sepulveda, 2019). Mais le problème semble plus large : la reconnaissance de non-validité de tels brevets ne change pas le fait que les logiciels soient couverts par une énorme quantité de brevets. Comme nous l'avons vu, les logiciels peuvent tout à fait être couverts par des brevets qui ne soient pas relatifs aux logiciels.

Le problème actuel entre brevets et logiciels est plus large que le cas particulier du brevet logiciel. Le gigantesque nombre de brevets du secteur semble actuellement être plus un frein qu'un encouragement à l'innovation. Cependant, il semble difficile de plaider pour l'exclusion pure et simple du logiciel du délit de contrefaçon (n'en déplaise aux opposants du « brevet logiciel ») : l'informatique est de nos jours trop pervasive pour envisager raisonnablement une telle exception. De plus en plus de fonctions sont maintenant réalisées par logiciel, la flexibilité apportée par le logiciel (comparé aux réalisations effectuées uniquement par matériel) ne pouvant qu'accentuer le phénomène. Il semble donc déraisonnable de défendre une option qui exclurait la majorité des réalisations des règles liées aux brevets, sans rendre en même temps caduque tout le système de brevets du domaine industriel.

---

<sup>100</sup> Les Etats-Unis ont une exigence de divulgation du mode de réalisation optimal, comportant des avantages évidents pour le partage de la connaissance (Selinger, 1994). Cependant, ne pas divulguer le mode de réalisation optimal n'est plus considéré comme un motif de nullité du brevet depuis la réforme de 2011 (portée par l'*America Invents Act – AIA*). Peu d'autres pays ont une telle exigence, à l'exception notable du Japon (Lu, 2011).

<sup>101</sup> Alice Corp. v. CLS Banks

Si nous observons le système d'un peu plus loin, nous pouvons constater que la prolifération excessive des brevets n'est pas cantonnée au logiciel, mais en fait à tout le secteur de l'informatique et des télécoms. Ce domaine technologique est un véritable « champ de mines » de brevet, où il est devenu impossible de réaliser ou concevoir un produit (ou même une portion d'un produit) sans enfreindre des centaines de brevets.

Afin de situer l'ordre de grandeur des brevets dans ces domaines techniques, prenons l'exemple du smartphone :

En 2014, le total des redevances demandées pour un smartphone était estimé à plus de 120\$ (Armstrong, Mueller et Syrett, 2014), soit plus que le coût de fabrication de l'appareil. Depuis, ce total a probablement augmenté du fait du nombre grandissant de technologies embarquées. L'Organisation Mondiale de la Propriété Intellectuelle (OMPI) cite des estimations de l'ordre de 250 000 brevets relatifs aux technologies relatives aux smartphones (WIPO, 2017). Les gros industriels du domaine se tiennent mutuellement en respect dans un fragile équilibre de licences croisées, chacun détenant un portefeuille de plusieurs dizaines de milliers de brevets. Ce contexte rend quasiment impossible l'arrivée de nouveaux acteurs, quel que soit leur niveau d'innovation. Seuls les plus gros acteurs peuvent encore participer à l'innovation dans le domaine, les portes du marché étant virtuellement fermées aux nouveaux entrants.

Non seulement l'objectif d'encouragement de l'innovation, à l'origine du brevet, est mis à mal, mais même le principe de libre concurrence de marché est mis en péril. Autant du point de vue humaniste que du point de vue économique, le brevet ne joue plus, dans le domaine de l'informatique et des télécoms, son rôle tel qu'initialement prévu. Nous sommes donc face à un problème systémique bien plus large que celui du « brevet logiciel ».

Pour autant, il serait périlleux de remettre complètement en question le système de brevets. Déjà, il y a 60 ans, la constatation était similaire :

*“If we did not have a patent system, it would be irresponsible, on the basis of our present knowledge of its economic consequences, to recommend instituting one. But since we have had a patent system for a long time, it would be irresponsible, on the basis of our present knowledge, to recommend abolishing it.”*

*(Machlup, 1958)*

Depuis, la situation, alors préoccupante, est devenue critique, empirant avec le développement du secteur des technologies de l'information et de la communication. Certes, il est indéniable que le système de brevets, dans sa forme actuelle, n'a pas empêché le formidable développement de ce secteur. Mais il est plus que légitime de mettre en doute la pérennité du système, et de se demander si nous ne risquons pas d'être confrontés au principe des rendements décroissants, l'exploitation industrielle des avancées devenant potentiellement plus freinée que favorisée par le système actuel. La même constatation est effectuée dans d'autres domaines : il y a 10 ans, Heller parlait de « tragédie des anticommons » au sujet des brevets de la santé (Heller, 2008), en détaillant les influences néfastes de la situation d'interblocage liée aux brevets trop nombreux.

Il semble juste que les inventeurs, et les investisseurs les soutenant, puissent avoir un retour sur investissement, voir même puissent obtenir de la part du marché une prime – raisonnable – pour leurs efforts, mais le système doit évoluer. Il doit non pas se réinventer, les principes généraux semblant raisonnables, mais s'adapter pour permettre de continuer à apporter des retombées positives au plus grand nombre.

Nous avons vu que dans certains secteurs, comme l'IoT, où des brevets sont déposés massivement, les industriels tendent à mettre un peu de place une sorte d'équilibre, à mi-chemin entre la coopération et la dissuasion à

coup de portefeuilles de brevets grossissant à un rythme effréné. Ce fragile équilibre présente des risques très importants : certains acteurs, non industriels, peuvent être tentés d'établir des stratégies de monétisation déraisonnables, allant à l'encontre du développement économique général. Est-il raisonnable que ce risque puisse durer plus de vingt ans, où la durée de validité des brevets est-elle trop importante dans le secteur du logiciel ?

Dans le système actuel, les brevets sont un atout important, mais font également peser un important risque économique sur certains secteurs, et il n'est pas déraisonnable de penser que des évolutions sont souhaitables, afin de rééquilibrer la part de risque actuelle. Ces évolutions pourraient venir de choix dictés par des visées humanistes, ou plus pragmatiquement des besoins du marché. A court terme, une réduction sensible du nombre de brevets de certains secteurs pourrait créer un « appel d'air » permettant d'aider l'exploitation économique de l'innovation (Le Bas et Pénin, 2015).

A partir du moment où la pression du système gêne plus qu'elle ne favorise, et semble tendre vers une future paralysie potentielle, gageons que les acteurs légaux, poussés par la nécessité de préserver l'économie de marché, sauront faire progresser les règles actuelles. Le revirement récent de l'affaire Alice est un exemple positif dans ce sens, et l'on ne peut qu'espérer que d'autres adaptations aient lieu à l'avenir, pour le bénéfice du plus grand nombre.

# BIBLIOGRAPHIE

- C. ANN, « PATENT TROLLS – MENACE OR MYTH? », IN PATENTS AND TECHNOLOGICAL PROGRESS IN A GLOBALIZED WORLD: LIBER AMICORUM JOSEPH STRAUS, BERLIN, HEIDELBERG: SPRINGER BERLIN HEIDELBERG, 2009, p. 355-364.
- A. ARMSTRONG, J. MUELLER, ET T. SYRETT, « THE SMARTPHONE ROYALTY STACK: SURVEYING ROYALTY DEMANDS FOR THE COMPONENTS WITHIN MODERN SMARTPHONES », SOCIAL SCIENCE RESEARCH NETWORK, ROCHESTER, NY, SSRN SCHOLARLY PAPER ID 2443848, MAI 2014.
- R. M. BALLARDINI, INTELLECTUAL PROPERTY PROTECTION FOR COMPUTER PROGRAMS: DEVELOPMENTS, CHALLENGES, AND PRESSURES FOR CHANGE. HANKEN SCHOOL OF ECONOMICS, 2012.
- M. BELCHER ET J. CASEY, « HACKING THE PATENT SYSTEM: A GUIDE TO ALTERNATIVE PATENT LICENSING FOR INNOVATORS », JUELSGAARD INTELLECTUAL PROPERTY & INNOVATION CLINIC AT STANFORD LAW SCHOOL, JANV. 2016.
- F. BENHAMOU ET J. FARCHY, DROIT D’AUTEUR ET COPYRIGHT. LA DECOUVERTE, 2007.
- T. BOUVIER, « BREVETABILITE DES PROGRAMMES D’ORDINATEUR : ET SI L’OEB S’ETAIT TROMPE... ? », INPUT IP, 29-JUIN-2018.
- E. G. COLEMAN, CODING FREEDOM: THE ETHICS AND AESTHETICS OF HACKING. PRINCETON UNIVERSITY PRESS, 2013.
- J. CONTRERAS, « PATENT PLEDGES », SSRN ELECTRONIC JOURNAL, p. 66, JANV. 2013.
- J. CONTRERAS, « THE EVOLVING PATENT PLEDGE LANDSCAPE », CENTER FOR INTERNATIONAL GOVERNANCE INNOVATION PAPER SERIES, n° 166, p. 24, AVR. 2018.
- P. B. DE LAAT, « COPYRIGHT OR COPYLEFT?: AN ANALYSIS OF PROPERTY REGIMES FOR SOFTWARE DEVELOPMENT », RESEARCH POLICY, VOL. 34, n° 10, p. 1511-1532, DÉC. 2005.
- S. A. DUNN, « DEFINING THE SCOPE OF COPYRIGHT PROTECTION FOR COMPUTER SOFTWARE », STANFORD LAW REVIEW, VOL. 38, n° 2, p. 497-534, 1986.
- B. EASTHAM, « PANASONIC AND THE OPENDOF PROJECT: OPEN-SOURCE VISION IN A LARGE COMPANY », XRDS: CROSSROADS, THE ACM MAGAZINE FOR STUDENTS, VOL. 22, n° 2, p. 58 - 61, DÉC. 2015.
- A. K. HAAPANEN, « FREE AND OPEN SOURCE SOFTWARE AND THE MYSTERY OF SOFTWARE PATENT LICENSES UNDER THE GPL », INTERNATIONAL FREE AND OPEN SOURCE SOFTWARE LAW REVIEW, VOL. 7, n° 1, p. 19-28, DÉC. 2015.
- M. HELLER, THE GRIDLOCK ECONOMY: HOW TOO MUCH OWNERSHIP WRECKS MARKETS, STOPS INNOVATION, AND COSTS LIVES, 1ST EDITION. NEW YORK: BASIC BOOKS, 2008.
- D. HOUTCIEFF, « LA CONSECRATION DE L’INTERDICTION DE SE CONTREDIRE », LA SEMAINE JURIDIQUE, n° 46, p. 2246-2248, 2011.
- N. T. HORNE, « OPEN SOURCE SOFTWARE LICENSING: USING COPYRIGHT LAW TO ENCOURAGE FREE USE », GA. ST. U. L. REV., VOL. 17, p. 863, 2001



- B. JEAN, « OPEN SOURCE ET BREVETABILITE DES INVENTIONS MISES EN ŒUVRE PAR ORDINATEUR. QUELLE COEXISTENCE ? QUELLE COMPATIBILITÉ ? », CEIPI COLLECTION, n° 67, p. 243, JUILL. 2019.
- C. JONES, THE TECHNICAL AND SOCIAL HISTORY OF SOFTWARE ENGINEERING. ADDISON-WESLEY, 2013.
- N. JULLIEN ET J.-B. ZIMMERMANN, « LE LOGICIEL LIBRE : UN RENOUVEAU DU MODELE INDUSTRIEL COOPERATIF DE L' INFORMATIQUE », IN HISTOIRE ET CULTURE DU LIBRE. DES LOGICIELS PARTAGES AUX LICENCES ECHANGEES, FRAMABOOK, 2013, p. 135-164.
- C. M. KELTY, « CULTURE' S OPEN SOURCES: SOFTWARE, COPYRIGHT, AND CULTURAL CRITIQUE », ANTHROPOLOGICAL QUARTERLY, VOL. 77, n° 3, p. 499-506, SEPT. 2004.
- D. M. KENNEDY, « A PRIMER ON OPEN SOURCE LICENSING LEGAL ISSUES: COPYRIGHT, COPYLEFT AND COPYFUTURE », ST. LOUIS U. PUB. L. REV., VOL. 20, p. 345, 2001.
- S. A. KIRK, HANDBOOK OF RESEARCH ON OPEN SOURCE SOFTWARE: TECHNOLOGICAL, ECONOMIC, AND SOCIAL PERSPECTIVES: TECHNOLOGICAL, ECONOMIC, AND SOCIAL PERSPECTIVES. IDEA GROUP INC (IGI), 2007.
- S. LAPOINTE, « L'HISTOIRE DES BREVETS », LES CAHIERS DE PROPRIETE INTELLECTUELLE, VOL. 12, n° 3, p. 19, 2000.
- C. LE BAS ET J. PENIN, « BREVET ET INNOVATION : COMMENT RESTAURER L'EFFICIENCE DYNAMIQUE DES BREVETS ? », REVUE D' ECONOMIE INDUSTRIELLE, n° 151, p. 127-160, SEPT. 2015.
- C. LE GOFFIC, « CONTREFAÇON DANS LE CADRE DE L'IMPRESSION 3D: RESPONSABILITES ET REMEDES », IN LA PROPRIETE INTELLECTUELLE & LA TRANSFORMATION NUMERIQUE DE L'ECONOMIE, INPI, 2015.
- L. LESSIG, THE FUTURE OF IDEAS: THE FATE OF THE COMMONS IN A CONNECTED WORLD, 1ST ED. NEW YORK, NY: RANDOM HOUSE, 2001.
- B. LU, « BEST MODE DISCLOSURE FOR PATENT APPLICATIONS: AN INTERNATIONAL AND COMPARATIVE PERSPECTIVE », SOCIAL SCIENCE RESEARCH NETWORK, ROCHESTER, NY, SSRN SCHOLARLY PAPER ID 1938859, JUIN 2011.
- F. MACHLUP, « AN ECONOMIC REVIEW OF THE PATENT SYSTEM », STUDY NO.15 OF THE UNITED STATES SENATE SUB-COMMITTEE ON PATENTS, TRADEMARKS AND COPYRIGHTS, JANV. 1958.
- F. MACREZ, « LOGICIEL : LE CUMULARD DE LA PROPRIETE INTELLECTUELLE », IN L'ARTICULATION DES DROITS DE PROPRIETE INTELLECTUELLE, DALLOZ, 2011, p. 47-63.
- C. MARACKE ET A. METZGER, « PLAYING NICE WITH PATENTS: DO VOLUNTARY NON-AGGRESSION PLEDGES PROVIDE A SOUND BASIS FOR INNOVATION? », NORTH CAROLINA JOURNAL OF LAW & TECHNOLOGY, VOL. 17, n° 3, p. 32, MARS 2016.
- MCKINSEY, DISRUPTIVE TECHNOLOGIES: ADVANCES THAT WILL TRANSFORM LIFE, BUSINESS, AND THE GLOBAL ECONOMY. MCKINSEY GLOBAL INSTITUTE, 2013.
- E. MOGLEN, « ANARCHISM TRIUMPHANT: FREE SOFTWARE AND THE DEATH OF COPYRIGHT », FIRST MONDAY, VOL. 4, n° 8, AOÛT 1999.
- J. PENIN, « QUELLE POLITIQUE DE LICENCE DE BREVET POUR LES ORGANISMES PUBLICS DE RECHERCHE ? EXCLUSIVITE VERSUS MODELES PLUS OUVERTS », MI, VOL. 14, n° 3, p. 47-58, 2010.
- A. PUGH ET L. MAJERUS, « POTENTIAL DEFENSES OF IMPLIED PATENT LICENSE UNDER THE GPL ». FENWICK & WEST LLP, 2006.
- E. S. RAYMOND, THE CATHEDRAL AND THE BAZAAR: MUSINGS ON LINUX AND OPEN SOURCE BY AN ACCIDENTAL REVOLUTIONARY. O'REILLY, 2001.
- RELECURA, « INTERNET OF THINGS PATENT LANDSCAPE REPORT», MAI 2018.

- J. RICE, « THE DEFENSIVE PATENT PLAYBOOK », BERKELEY TECHNOLOGY LAW JOURNAL, VOL. 30, N° 4, P. 725, NOV. 2015.
- G. S. RICH, « MY FAVORITE THINGS », IDEA THE JOURNAL OF THE FRANKLIN PIERCE CENTER FOR INTELLECTUAL PROPERTY, VOL. 34, N° 3, P. 1, 1994.
- H. ROLNIK, « LA GUERRE DES BREVETS: QUELLES STRATEGIES? », UNIVERSITE DE MARNE-LA-VALLEE, 2002.
- L. ROSEN, OPEN SOURCE LICENSING: SOFTWARE FREEDOM AND INTELLECTUAL PROPERTY LAW, 1 EDITION. PRENTICE HALL, 2004.
- J. R. SELINGER, « IN DEFENSE OF “BEST MODE” : PRESERVING THE BENEFIT OF THE BARGAIN FOR THE PUBLIC », CATHOLIC UNIVERSITY LAW REVIEW, VOL. 43, N° 4, P. 37, 1994.
- J. R. SEPULVEDA, « THE POST-ALICE JURISPRUDENCE PENDULUM AND ITS EFFECTS ON PATENT ELIGIBLE SUBJECT MATTER », TOURO L. REV., VOL. 35, P. 897, 2019.
- SUPREME COURT OF THE UNITED STATES, 13-298 ALICE CORP. V. CLS BANK INT’L (06/19/2014). 2014, P. 21.
- J. L. TRAN, « TWO YEARS AFTER ALICE V. CLS BANK », J. PAT. & TRADEMARK OFF. SOC’Y, VOL. 98, P. 354, 2016.
- A. TRIDICO, J. JACOBSTEIN, ET L. WALL, « FACILITATING GENERIC DRUG MANUFACTURING: BOLAR EXEMPTIONS WORLDWIDE », VOL. 121, N° 3, P. 17-20, 2014.
- N. VOLANIS ET N. PRENTOULIS, « (MIS-)USE OF PATENTS IN OPEN SOURCE SOFTWARE AND OPEN STANDARDS: AN EVALUATION OF TACTICS AND DEFENCES UNDER INTELLECTUAL PROPERTY AND COMPETITION LAW », P. 58, 2009.
- WIPO, WORLD INTELLECTUAL PROPERTY REPORT 2017: INTANGIBLE CAPITAL IN GLOBAL VALUE CHAINS. WORLD INTELLECTUAL PROPERTY ORGANIZATION, 2017.



# ANNEXE 1 : DEFINITIONS

## Code Source

*Le code source est un texte qui représente les instructions d'un programme telles qu'elles ont été écrites dans un langage de programmation sous une forme humainement lisible par un programmeur. Le code source se matérialise souvent sous la forme d'un ensemble de fichiers textes. (Source : Wikipedia)*

## Contributeur

*Un contributeur [...] est une personne qui collabore à l'écriture d'une œuvre commune, à la réalisation d'un projet collectif, notamment sur le réseau mondial. Un contributeur participe à un projet collectif, à partir du moment où ses propres intérêts convergent avec ceux du projet. Le concept de contributeur trouve tout son sens dans le logiciel libre. Le logiciel libre est une œuvre commune réalisée collaborativement par des personnes se trouvant autour du monde. (Source : Wikipedia)*

## Compilée (forme)

La forme compilée, ou le logiciel compilé, est le logiciel ayant subi l'opération de compilation (voir compilateur)

## Compilateur

*En informatique, un compilateur est un programme qui transforme un code source en un code objet. Généralement, le code source est écrit dans un langage de programmation (le langage source), il est de haut niveau d'abstraction, et facilement compréhensible par l'humain. Le code objet est généralement écrit en langage de plus bas niveau (appelé langage cible) [...] afin de créer un programme exécutable par une machine. (Source : Wikipedia)*

## Compilation

Voir compilateur

## Dépôt

*En informatique, un dépôt ou référentiel (de l'anglais repository) est un stockage centralisé et organisé de données. Ce peut être une ou plusieurs bases de données où les fichiers sont localisés en vue de leur distribution sur le réseau ou bien un endroit directement accessible aux utilisateurs. (source : Wikipedia)*

## Distribution (Linux)

Ce que l'on appelle une distribution Linux est un ensemble cohérent et intégré de logiciels open source permettant l'installation d'un système d'exploitation de type Linux, associé à un certain nombre de logiciels applicatifs. Exemples : Ubuntu, Debian,...

## Famille (brevets)

Une famille de brevets est un ensemble de brevets, déposés dans différents pays, qui revendiquent tous la priorité d'un même dépôt initial, dont ils sont issus.

## Forge logicielle

*En informatique, le mot forge est utilisé pour décrire les systèmes de gestion de développement collaboratif de logiciel. (Source : Wikipedia)*

## Framework (logiciel)

*En programmation informatique, un framework désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel.*

*[...]*

*Les frameworks sont donc conçus et utilisés pour modeler l'architecture des logiciels applicatifs, des applications web [...] et des composants logiciels. (Source : Wikipedia)*

## Miroir

*En informatique, un miroir est une copie exacte d'un ensemble de données. Sur le Web, un site miroir est une copie exacte d'un autre site web. Les sites miroirs sont d'usage surtout afin de fournir plusieurs copies de la même information, en particulier pour les téléchargements de grande envergure, répartissant ainsi la charge générée par un trafic élevé sur plusieurs serveurs et sites totalement séparés. (Source : Wikipedia)*

## Pool de brevets (*Patent pool*)

Un pool de brevets est une entité permettant d'obtenir, sur une technologie donnée, une licence commune de brevets accordée par plusieurs bailleurs. Ceci permet, en théorie, de n'avoir à contracter qu'avec un seul interlocuteur (le pool) au lieu d'avoir à contracter individuellement avec chacun des brevetés.

## Projet (open source)

Collectif de personnes travaillant de concert pour la création d'un logiciel open source. Ceci inclut les personnes effectuant le développement, mais aussi les autres travaux nécessaires (tests, documentation, support technique, gestion de projet, ...)

## **Propriétaire (logiciel)**

Un logiciel est dit propriétaire lorsque son développement est réalisé et contrôlé par une seule société. Généralement ces logiciels ne sont distribués auprès de tiers que sous forme compilée, mais il arrive exceptionnellement que le code source de certains logiciels propriétaires soit disponible. Dans ce cas, la licence associée ne permet pas la modification, ou la redistribution par des tiers.

## **Référentiel**

Voir dépôt

## **Versionnage**

*Une version d'un logiciel correspond à un état donné de l'évolution d'un produit logiciel utilisant le versionnage. [...] le versionnage (équivalent francophone de l'anglais versioning) est le mécanisme qui consiste à conserver la version d'une entité logicielle quelconque, de façon à pouvoir la retrouver facilement, même après l'apparition et la mise en place de versions plus récentes. (Source: Wikipedia)*



## ANNEXE 2 : SECTIONS RELATIVES AUX BREVETS DES LICENCES OPEN SOURCE LES PLUS USUELLES

Les sections relatives aux brevets des licences open source les plus usuelles sont listées dans la table ci-dessous :

Licence	URL	Sections relatives aux brevets
<b>Apache 2.0</b>	<a href="https://opensource.org/licenses/Apache-2.0">https://opensource.org/licenses/Apache-2.0</a>	Section 3
<b>BSD</b>	<a href="https://opensource.org/licenses/BSD-3-Clause">https://opensource.org/licenses/BSD-3-Clause</a>	Aucune
<b>General Public License (GPL) v2</b>	<a href="https://opensource.org/licenses/GPL-2.0">https://opensource.org/licenses/GPL-2.0</a>	Sections 7 et 8
<b>General Public License (GPL) v3</b>	<a href="https://opensource.org/licenses/GPL-3.0">https://opensource.org/licenses/GPL-3.0</a>	Sections 8 et 10
<b>Lesser General Public License (LGPL) v2.1</b>	<a href="https://opensource.org/licenses/LGPL-2.1">https://opensource.org/licenses/LGPL-2.1</a>	Sections 11 et 12
<b>Lesser General Public License (LGPL) v3</b>	<a href="https://opensource.org/licenses/LGPL-3.0">https://opensource.org/licenses/LGPL-3.0</a>	Référence à la licence GPL v3
<b>MIT</b>	<a href="https://opensource.org/licenses/MIT">https://opensource.org/licenses/MIT</a>	Aucune
<b>Mozilla Public License (MPL) 2.0</b>	<a href="https://opensource.org/licenses/MPL-2.0">https://opensource.org/licenses/MPL-2.0</a>	Sections 1.11, 2.1, 2.3 et 5.2

Seules ces sections spécifiques sont reprises ci-après dans cette annexe, les textes complets de ces licences pouvant être obtenus aux adresses mentionnées dans le tableau précédent. Le lecteur souhaitant une comparaison détaillée de ces différentes licences pourra se référer à Rosen (2004).



## I - Apache license

### 3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

## II - BSD license

Cette licence ne comprend aucune clause explicitement relative aux brevets.

## III - General Public License (GPL) v2

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

## IV - General Public License (GPL) v3

### 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

[...]

### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted

under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

## V - Lesser General Public License (LGPL) v2.1

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

## VI - Lesser General Public License (LGPL) v3

Cette licence, pour les sections relatives aux brevets, fait référence à la licence GPLv3.

## VII - MIT license

Cette licence ne comprend aucune clause explicitement relative aux brevets.

## VIII - Mozilla public license (MPL)

### 1.11. “Patent Claims” of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

[...]

### 2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

[...]

### 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party’s modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

[...]

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

## **ANNEXE 3 : EXEMPLES DE CONTRIBUTOR LICENSE AGREEMENTS**

### **I - Google Individual Contributor License Agreement (Android Open Source Platform)**

In order to clarify the intellectual property license granted with Contributions from any person or entity, Google LLC ("Google") must have a Contributor License Agreement ("CLA") on file that has been signed by each Contributor, indicating agreement to the license terms below. This license is for your protection as a Contributor as well as the protection of Google; it does not change your rights to use your own Contributions for any other purpose.

You accept and agree to the following terms and conditions for Your present and future Contributions submitted to Google. Except for the license granted herein to Google and recipients of software distributed by Google, You reserve all right, title, and interest in and to Your Contributions.

#### 1. Definitions.

"You" (or "Your") shall mean the copyright owner or legal entity authorized by the copyright owner that is making this Agreement with Google. For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"Contribution" shall mean any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by You to Google for inclusion in, or documentation of, any of the products owned or managed by Google (the "Work"). For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to Google or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, Google for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by You as "Not a Contribution."

2. Grant of Copyright License. Subject to the terms and conditions of this Agreement, You hereby grant to Google and to recipients of software distributed by Google a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Your Contributions and such derivative works.

3. Grant of Patent License. Subject to the terms and conditions of this Agreement, You hereby grant to Google and to recipients of software distributed by Google a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by You that are necessarily infringed by Your Contribution(s) alone or by combination of Your Contribution(s) with the Work to which such Contribution(s) was submitted. If any entity institutes patent litigation against You or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging that your Contribution, or the Work to which you have contributed, constitutes direct or contributory patent infringement, then any patent licenses granted to that entity under this Agreement for that Contribution or Work shall terminate as of the date such litigation is filed.

4. You represent that you are legally entitled to grant the above license. If your employer(s) has rights to intellectual property that you create that includes your Contributions, you represent that you have received permission to make Contributions on behalf of that employer, that your employer has waived such rights for your Contributions to Google, or that your employer has executed a separate Corporate CLA with Google.

5. You represent that each of Your Contributions is Your original creation (see section 7 for submissions on behalf of others). You represent that Your Contribution submissions include complete details of any third-party license or other restriction (including, but not limited to, related patents and trademarks) of which you are personally aware and which are associated with any part of Your Contributions.

6. You are not expected to provide support for Your Contributions, except to the extent You desire to provide support. You may provide support for free, for a fee, or not at all. Unless required by applicable law or agreed to in writing, You provide Your Contributions on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON- INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

7. Should You wish to submit work that is not Your original creation, You may submit it to Google separately from any Contribution, identifying the complete details of its source and of any license or other restriction (including, but not limited to, related patents, trademarks, and license agreements) of which you are personally aware, and conspicuously marking the work as "Submitted on behalf of a third-party: [named here]".

8. You agree to notify Google of any facts or circumstances of which you become aware that would make these representations inaccurate in any respect.

## II - Microsoft Contribution License Agreement

This Contribution License Agreement ( “ Agreement ” ) is agreed to by the party signing below ( “ You ” ), and conveys certain license rights to Microsoft Corporation and its affiliates ( “ Microsoft ” ) for Your contributions to Microsoft open source projects. This Agreement is effective as of the latest signature date below.

### 1. Definitions.

“ Code ” means the computer software code, whether in human-readable or machine-executable form, that is delivered by You to Microsoft under this Agreement.

“ Project ” means any of the projects owned or managed by Microsoft in which software is offered under a license approved by the Open Source Initiative(OSI)([www.opensource.org](http://www.opensource.org))anddocumentationoffered under an OSI or a Creative Commons license (<https://creativecommons.org/licenses>).

“ Submit ” is the act of uploading, submitting, transmitting, or distributing code or other content to any Project, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Project for the purpose of discussing and improving that Project, but excluding communication that is conspicuously marked or otherwise designated in writing by You as “ Not a Submission. ”

“ Submission ” means the Code and any other copyrightable material Submitted by You, including any associated comments and documentation.

2. Your Submission. You must agree to the terms of this Agreement before making a Submission to any Project. This Agreement covers any and all Submissions that You, now or in the future (except as described in Section 4below), Submit to any Project.

3.Originality of Work. You represent that each of Your Submissions is entirely Your original work. Should You wish to Submit materials that are not Your original work, You may Submit them separately to the Project if You (a)retain all copyright and license information that was in the materials as You received them, (b)in the description accompanying Your Submission, include the phrase “ Submission containing materials of a third party: ” followed by the names of the third party and any licenses or other restrictions of which You are aware, and (c)follow any other instructions in the Project’s written guidelines concerning Submissions.

4. Your Employer. References to “ employer ” in this Agreement include Your employer or anyone else for whom You are acting in making Your Submission, e.g. as a contractor, vendor, or agent. If Your Submission is made in the course of Your work for an employer or Your employer has intellectual property rights in Your Submission by contract or applicable law, You must secure permission from Your employer to make the Submission before signing this Agreement. In that case, the term “ You ” in this Agreement will refer to You and the employer collectively. If You change employers in the future and desire to Submit additional Submissions for the new employer, then You agree to sign a new Agreement and secure permission from the new employer before Submitting those Submissions.



## 5. Licenses.

a. Copyright License. You grant Microsoft, and those who receive the Submission directly or indirectly from Microsoft, a perpetual, worldwide, non-exclusive, royalty-free, irrevocable license in the Submission to reproduce, prepare derivative works of, publicly display, publicly perform, and distribute the Submission and such derivative works, and to sublicense any or all of the foregoing rights to third parties.

b. Patent License. You grant Microsoft, and those who receive the Submission directly or indirectly from Microsoft, a perpetual, worldwide, non-exclusive, royalty-free, irrevocable license under Your patent claims that are necessarily infringed by the Submission or the combination of the Submission with the Project to which it was Submitted to make, have made, use, offer to sell, sell and import or otherwise dispose of the Submission alone or with the Project.

c. Other Rights Reserved. Each party reserves all rights not expressly granted in this Agreement. No additional licenses or rights whatsoever (including, without limitation, any implied licenses) are granted by implication, exhaustion, estoppel or otherwise.

6. Representations and Warranties. You represent that You are legally entitled to grant the above licenses. You represent that each of Your Submissions is entirely Your original work(except as You may have disclosed under Section3).You represent that You have secured permission from Your employer to make the Submission in cases where Your Submission is made in the course of Your work for Your employer or Your employer has intellectual property rights in Your Submission by contract or applicable law. If You are signing this Agreement on behalf of Your employer, You represent and warrant that You have the necessary authority to bind the listed employer to the obligations contained in this Agreement. You are not expected to provide support for Your Submission, unless You choose to do so. **UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING, AND EXCEPT FOR THE WARRANTIES EXPRESSLY STATED IN SECTIONS 3, 4, AND 6, THE SUBMISSION PROVIDED UNDER THIS AGREEMENT IS PROVIDED WITHOUT WARRANTY OF ANY KIND, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY OF NONINFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.**

7. Notice to Microsoft. You agree to notify Microsoft in writing of any facts or circumstances of which You later become aware that would make Your representations in this Agreement inaccurate in any respect.

8. Information about Submissions. You agree that contributions to Projects and information about contributions may be maintained indefinitely and disclosed publicly, including Your name and other information that You submit with Your Submission.

9. Governing Law/Jurisdiction. This Agreement is governed by the laws of the State of Washington, and the parties consent to exclusive jurisdiction and venue in the federal courts sitting in King County, Washington, unless no federal subject matter jurisdiction exists, in which case the parties consent to exclusive jurisdiction and venue in the Superior Court of King County, Washington. The parties waive all defenses of lack of personal jurisdiction and forum non-conveniens.

10. Entire Agreement/Assignment. This Agreement is the entire agreement between the parties, and supersedes any and all prior agreements, understandings or communications, written or oral, between the parties relating to the subject matter hereof. This Agreement may be assigned by Microsoft.

# ANNEXE 4 : MOZILLA OPEN SOFTWARE PATENT LICENSE AGREEMENT

## Mozilla Open Software Patent License Agreement v1.1

This Open Software Patent License Agreement (“**Agreement**”) is made between you and the Licensor identified below. For purposes of this Agreement, you means you and your Affiliates, and Licensor means the Mozilla Corporation, Mozilla Foundation and any of their Affiliates.

### 1. Definitions

- a. “**Affiliate**” means any entity owned or controlled by you or Licensor, as applicable, either now or in the future.
  - b. “**Licensor Patents**” means all Patents owned, or licensable as described in this Agreement, at any time during the License Term, by Licensor.
  - c. “**Open Patent Licensing**” means royalty-free and non-discriminatory licensing or cross-licensing arrangements such as open source licenses or open standards licensing.
  - d. “**Open Source Software**” means Software that is made generally publicly available under an open source license, which means a license meeting the Open Source Definition as published by the Open Source Initiative. For clarity, Software that is made available under multiple licenses will be considered Open Source Software only to the extent it is used under an open source license.
  - e. A “**Participant**” means any party that is bound to any version of this Agreement, including you.
  - f. A “**Patent**” means a patent, and any continuation, divisional, continuation-in-part, or other patent that claims priority therefrom.
  - g. “**Software**” means instructions for programmable physical apparatus, but excludes all physical apparatus, such as computer processors, computer hardware, and peripheral devices. For clarity, “Software” excludes semiconductor mask works, and includes services consisting of making available the functionality of software.
  - h. “**Your Patents**” means all Patents owned, or licensable as described in this Agreement, at any time during the License Term, by you.
2. **Acceptance and Effectiveness.** You will be bound to this Agreement if you (a) make, use, sell, import or otherwise exploit any Software, or practice any method embodied in Software, in such a fashion that,

absent the licenses granted to you herein, would infringe any Licensor Patent; (b) take any action knowingly relying on the licenses granted to you herein, or (c) show in any other reasonable way your intention to be bound to this Agreement. The License Term will begin upon the first to occur of any of a, b, or c of this paragraph and continue unless and until your license is terminated under Section 4. In the event it is determined by a court of competent jurisdiction that this Agreement is not binding on you, then the licenses granted by Licensor and any other Participant to you hereunder will be void ab initio (i.e. will have never been granted to you).

### **3. Conditional License**

- a. Subject to Section 3(b), Licensor, on behalf of itself and each of its Affiliates, hereby grants to you and your Affiliates a royalty-free, fully-paid-up, worldwide, non-exclusive, non-transferable license under Licensor Patents to make, have made, use, sell, offer for sale, import, and otherwise exploit any Software and practice any method embodied in Software.
- b. In consideration for the rights granted hereunder in Section 3(a), you agree to grant, upon request and upon reasonable, non-discriminatory and royalty free terms and conditions, to Licensor and any Participant a royalty-free, fully-paid-up, worldwide, non-exclusive, non-transferable license under your Patents to make, have made, use, sell, offer for sale, import, and otherwise exploit any Open Source Software and practice any method embodied in Open Source Software. As used in this paragraph, “reasonable” means, without limitation, that such terms must not place any restrictions on the practice of your Patents that would prohibit use, modification and redistribution of the applicable software under the terms of Open Source Licenses. By way of example and not limitation, such terms must not restrict the field of use of the Software, or prohibit modification of the Software.
- c. Licensor or others may license Patents to you under other terms, such as Open Patent Licensing efforts, and any other such license granted to you will not limit your rights under this Agreement.
- d. During the License Term, you may transfer your rights, licenses and obligations under this Agreement in connection with a merger, acquisition, or sale of all or substantially all of your assets related to this Agreement, to a party that (1) expresses in writing its intention to be bound (sufficient to meet the conditions of Section 2), and (2) has not asserted Claims as described in Section 4 prior to the date of transfer.
- e. You acknowledge that this Agreement does not reflect a royalty that might otherwise have been negotiated at arms’ length for any Patent, and that this Agreement is not intended to affect any determination of whether infringement of any Patent may be adequately compensated by money damages.
- f. Each Participant is an intended third party beneficiary of this Agreement.

### **4. Termination**

- a. If you bring a Claim, the licenses granted to you by Licensor and all other Participants will immediately terminate as of the date you bring such Claim. For purposes of the foregoing, a “Claim” means (a) filing any lawsuit or other legal action (including any action to enjoin import of products) asserting infringement of your Patent by Software, (b) making any written claims of

infringement of any of your Patents by Software, including requests to cease and desist infringement, or (c) assisting any third party to bring any such claim. However, “Claims” will not include (x) claims of patent infringement brought by you as counter-claims or cross-claims in any third party claim of patent infringement against you; (y) invitations or offers by you to license Patents in connection with Open Patent Licensing; or (z) claims brought by you to enforce the terms of this Agreement or any related Open Patent Licensing (including intervention in a third party claim of patent infringement against another Participant).

- b. You may terminate the License Term by making a public statement of your intention to no longer be party to this Agreement, stating a date of termination of the License Term no earlier than the date of such statement, and stating in good faith that to your knowledge, you will not infringe any Licensor Patent as of the end of the License Term. In such case, the licenses granted to you by all Participants will terminate effective as of the end of the License Term, and your licenses under Section 3(b) will also terminate as of the end of the License Term.

## 5. General

- a. This is the entire agreement between Licensor and you on the subject matter of this Agreement. In the event that any term or condition contained in this Agreement shall be determined by any court of competent jurisdiction to be unenforceable by reason of its extending for too great a period of time or over too great a geographical area or by reason of its being too extensive in any other respect, it shall be interpreted to extend only over the maximum period of time for which it may be enforceable and/or over the maximum geographical area as to which it may be enforceable and/or to the maximum extent in all other respects as to which it may be enforceable, all as determined by such court in such action.
- b. The text of this Agreement is provided to you under the [CC0 1.0 Universal Waiver](#).
- c. Any law or regulation which provides that the language of a contract shall be construed against the drafter will not be used to construe the terms of this Agreement against any Participant.

If you would like to show your intention to be bound by this Agreement, you may wish to make the following statement: “ZXQ, Inc. hereby states that it intends to be bound by the Mozilla Open Software Patent License Agreement for Mozilla, version \_\_\_\_, as of [date].”

If you would like to use the text of this agreement for your own patent licensing, you may feel free to do so. However, to avoid confusion please remove all references to “Mozilla” from your version of the agreement. You may wish to say that your agreement is “based on the Mozilla Open Software Patent License Agreement v1.1,” for informational purposes, but you are not required to do so.