

UNIVERSITE TOULOUSE I CAPITOLE

ECOLE EUROPEENNE DE DROIT

Unité 3 : Dissertation sur un sujet de droit comparé

La protection des programmes d'ordinateurs : comparaison des systèmes européens et étasuniens.

Par

Emmanuelle Gaillard

Septembre 2018

Plan

Introduction

Partie 1 : Les modes traditionnels de protection des programmes d'ordinateur dans les législations étasuniennes et européennes.

- Chapitre 1 : Une opposition des législations en matière de brevet.

Section 1 : Les divergences de bases législatives quant aux inventions brevetables.

Section 2 : Opposition des développements et des exceptions jurisprudentielles.

- Chapitre 2 : Une convergence apparente des législations en matière de droit d'auteur.

Section 1 : Une inclusion sans encombre des programmes informatiques dans les deux systèmes.

Section 2 : Des exceptions affectant différemment la protection selon le système.

Partie 2 : Les modes alternatifs de protection des programmes d'ordinateur dans les législations étasuniennes et européennes.

- Chapitre 1 : L'utilisation différenciée des modes alternatifs de protection des programmes d'ordinateur.

Section 1 : Des régimes juridiques venant variablement en aide selon le territoire de protection.

Section 2 : Des outils permettant de répondre aux difficultés : l'avantage étasunien ?

- Chapitre 2 : L'utilisation unifiée des concepts alternatifs de protection des programmes d'ordinateur.

Section 1 : Le logiciel libre : une réelle absence de protection ?

Section 2 : La protection sui generis : une alternative oubliée ?

Conclusion

Annexes

Table des sigles

Bibliographie

Table des matières

Introduction

Le juge Petersen avait remarqué que *‘ce qui vaut le coup d’être copié vaut, à première vue, le coup d’être protégé’*.¹ Cette citation est remarquablement vraie de nos jours à propos de l’exploitation des programmes d’ordinateur. Même si les programmes d’ordinateur sont maintenant utilisés depuis plus d’un demi-siècle, la question de leur protection n’a vraiment eu de l’importance juridiquement parlant que récemment, depuis l’avènement des ordinateurs personnels.

L’objet de cette étude nécessite une précision de langage. « Programme » et « logiciel » sont souvent utilisés de façon interchangeable. Le « programme d’ordinateur » (ou programme informatique) n’est pas défini par le Code de la Propriété intellectuelle ou la Directive 91/250/CE du 14 Mai 1991. Il ne bénéficie pas non plus d’une définition harmonisée par des traités internationaux. Le Traité de l’OMPI sur le droit d’auteur et l’accord sur les ADPIC² se contentent simplement de le classer parmi les œuvres littéraires.³ Les Etats-Unis définissent les programmes de la manière suivante : *« un ensemble d’indications ou d’instructions destiné à être utilisé directement ou indirectement dans un ordinateur pour permettre d’obtenir un certain résultat »*.⁴ L’Union Européenne s’est alignée sur les Etats-Unis en se dotant d’une directive,⁵ mais n’a pas proposé de définition des programmes. S’agissant des logiciels, un arrêté du 22 Décembre 1981 sur l’enrichissement de la langue française⁶ adopte la définition suivante : *« l’ensemble des programmes, procédés et règles ainsi que la documentation qui leur est éventuellement associée, relatifs au fonctionnement d’un matériel de traitement de l’information. »* Un logiciel est donc une notion plus large qu’un programme, ce dernier n’étant qu’un élément d’un logiciel. Le logiciel est l’ensemble des composants immatériels utilisés par un ordinateur ou une machine : système d’exploitation,⁷ applicatifs ou utilitaires,⁸

¹ “What is worth copying is prima facie worth protecting” dans Chancery Division, 1916, University of London Press Ltd v University Tutorial Press Ltd, 1916, 2 Ch 601, 610, par Petersen J

² Aspects des droits de propriété intellectuelle qui touchent au commerce, Annexe 1 C de l’Accord de Marrakech instituant l’Organisation mondiale du commerce, signé à Marrakech, au Maroc, le 15 avril 1994.

³ Traité de l’OMPI sur le droit d’auteur (WCT) du 20 décembre 1996, Article 4 ; ADPIC article 10

⁴ Copyright Amendment Act du 12 Décembre 1980, section 101 : A “computer program” is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.

⁵ Directive 91/250/CEE du Conseil, du 14 mai 1991, concernant la protection juridique des programmes d’ordinateur, OJ L 122, 17.5.1991, p. 42–46 (abrogée)

⁶ Arrêté du 22 décembre 1981 relatif à l’enrichissement du vocabulaire de l’informatique, JO 17-01-1982 p. NC 624-626

⁷ Comme Windows, Linux, Android, MAC OS...

⁸ Traitement de texte, tableur, traducteur, jeux vidéo

logiciel embarqué ou « firmware ».⁹ Nous utiliserons les deux termes de façon interchangeable.

Au tout début du développement de cette technologie, les logiciels informatiques étaient utilisés par les grandes entreprises, principalement aux Etats-Unis, pour des raisons industrielles et non pour l'usage privé des consommateurs. La protection des logiciels était contenue au sein des conditions contractuelles des entreprises, complétée par des actions pour abus de confiance.¹⁰ Mais avec l'arrivée des microprocesseurs et ordinateurs personnels, l'utilisation des logiciels s'est largement développée, intégrant la plupart des objets qui nous entourent : les téléphones, les voitures, les télévisions, les appareils électroménagers, etc... C'est une industrie riche et créatrice d'emploi qui ne cesse d'accroître son poids économique mais qui devient particulièrement exposée à un risque démesuré de piratage. Il est dès lors devenu impossible de dépendre uniquement de conditions contractuelles ou d'actions pour abus de confiance. Comme développer un logiciel est long et coûteux, les acteurs de ce domaine ont cherché des moyens de protection. En revanche, même si cette recherche de protection est très importante commercialement, trouver une solution n'est pas simple.

Les Etats-Unis, figurant comme point central de l'innovation en matière de logiciel, ont majoritairement mené les tendances de protection. Le marché des logiciels est concentré aux Etats-Unis. En effet, 74 des 100 premiers éditeurs de logiciels se trouvent aux Etats-Unis tandis que 14 se trouvent en Europe.¹¹ Microsoft, IBM, HP et Oracle, fleurons de l'économie américaine, sont en tête. Face au géant de l'industrie des logiciels, il faut voir comment l'Union Européenne se positionne. Le domaine du numérique, intégrant les logiciels, est une des thématiques prioritaires de l'Union Européenne. Afin d'assurer la compétitivité de son industrie du numérique, l'Union Européenne adopte une stratégie pour un marché unique numérique¹² et la Commission propose d'investir 9,2 milliards d'euros afin de soutenir le domaine numérique.¹³ Constant les efforts que l'Union Européenne déploie afin d'assurer sa compétitivité, il conviendra d'explorer parallèlement les systèmes étasuniens et européens dans leur manière de protéger les logiciels puisqu'encadrer cet outil participe à la croissance pérenne des industries.

⁹ Ensemble des instructions et des structures de données intégrées dans un matériel informatique pour assurer son fonctionnement

¹⁰ Rowland D, Kohli U et Charlesworth A, *Information Technology Law*, 4ème édition, Routledge, Royaume-Uni, 2012, p 355

¹¹ Selon le classement fourni par PricewaterhouseCoopers sur le site suivant :

https://www.lemonde.fr/technologies/article/2011/01/06/les-americains-dominent-toujours-autant-le-marche-mondial-du-logiciel_1461988_651865.html

¹² Voir le site web de la Commission: https://ec.europa.eu/commission/priorities/digital-single-market_fr#policy-areas

¹³: Voir le Communiqué de Presse de la Commission : http://europa.eu/rapid/press-release_IP-18-4043_fr.htm?locale=FR

Ces deux modèles ont été parallèlement à la recherche de la protection la plus idéale. Au début des années 1970, les instances internationales et académiciens ont exploré ces problèmes et ont examiné la meilleure protection pour les logiciels par le système des brevets, par le droit d'auteur ou par un système sui generis. Il pourrait être rapidement conclu que le droit d'auteur, protégeant la forme d'une œuvre automatiquement sous condition d'originalité, est une solution adaptée car un logiciel peut être exprimé sous forme écrite.¹⁴ Obtenir une protection par le droit d'auteur permet à l'auteur de contrôler l'usage et la diffusion de son travail, et d'être protégé contre toute copie, même si ces droits sont contrebalancés par un droit d'étudier et de copier une certaine quantité du travail pour des raisons précises et sous certaines conditions. C'est la solution qui fut privilégiée aux Etats-Unis, suivi par l'Union Européenne. De plus, les logiciels peuvent être protégés dans certains cas par le brevet dans ces deux systèmes. Les brevets protègent la fonctionnalité du logiciel en plus de sa forme, contrairement aux droits d'auteurs qui protègent seulement la forme (le code écrit). En revanche, un brevet n'est pas automatiquement attribué lors de la création du travail : l'auteur doit faire une demande qui sera ensuite examinée. L'avantage est qu'un brevet confère un monopole pour une période limitée moyennant la divulgation de son invention, de même que le droit d'auteur octroie un droit exclusif à l'auteur sur la forme de son travail tout en laissant libre court aux idées. Le brevet peut être très bénéfique en ce sens qu'il permet de récupérer les dépenses engagées dans le développement du logiciel, encourageant l'innovation. Le droit auteur favorise la créativité.

Mais faut-il à tout prix chercher à octroyer une protection quasi-monopolistique à l'auteur ? La raison première de ce monopole est de récompenser l'auteur pour son innovation en lui permettant de rentabiliser ses investissements, ce qui encourage par ricochet les autres potentiels inventeurs à investir. Investir nécessite un environnement sécurisé. Il en découle que les inventeurs bien protégés par la propriété intellectuelle créeront plus et contribueront à l'innovation perpétuelle. En revanche, il y a une multitude d'arguments s'opposant à cette thèse. En effet, les petites et moyennes entreprises ont des difficultés à déposer des brevets et les entreprises américaines sont d'ores et déjà très dominantes sur ce marché. Le droit d'auteur, bien que peu coûteux, est trop insuffisant pour les protéger de façon optimale : souvent seule la forme du programme est protégé, et pas les fonctionnalités. Si les PME veulent s'assurer une protection, elles doivent alors décider de réduire leur budget dédié à la recherche. Plus drastique encore, les coûts des actions en justice sont si dissuasifs qu'elles

¹⁴ Cf annexe page 53

décident de ne pas se lancer dans la R&D puisque le projet ne pourra pas être porté à bout.¹⁵ La conséquence est que les PME ne contribueraient plus à l'innovation. Cependant, affermir la protection risquerait de « figer » le marché car cela renforcerait de même les grands éditeurs, comme Microsoft ou Oracle. Un autre argument contre un renforcement serait qu'octroyer une trop forte protection aux auteurs leur permettrait de refuser l'interopérabilité avec d'autres produits logiciels. Ils pourraient dès lors facilement obtenir un monopole en développant leurs propres produits compatibles et fermer le marché. Enfin, il peut être suggéré que protéger de façon optimale les logiciels reviendrait à les rendre encore plus secret. Or, ce que nous a montré l'affaire Volkswagen¹⁶ est que l'on peut facilement cacher des logiciels illicites. Ainsi, il serait préférable que les logiciels soient rendus publics et contrôlés.¹⁷ Par conséquent, on peut penser aux alternatives de protection permettant de remédier à toutes ces critiques : des alternatives pouvant octroyer plus de protection à l'auteur quand il en a besoin, comme la concurrence déloyale, le droit des marques, des dessins et modèles, mais aussi des alternatives permettant une protection tout en assurant une ouverture. On peut également se demander si l'absence de protection ne serait pas bénéfique puisqu'elle permettrait de laisser libre court à l'innovation.

En conséquence, il serait intéressant d'étudier la manière dont les grands principes de protections existantes (brevets et droits d'auteur) s'appliquent aux logiciels aux Etats-Unis et dans l'Union Européenne, et sont complétés par des alternatives soit spécifiques à ces deux systèmes, soit universelles.

Notre démarche comporte deux grands stades d'analyse de la protection des programmes d'ordinateur. La première partie permet de cerner l'application des principes de protections existantes aux logiciels, droit des brevets et droit d'auteur, sous les angles étasunien et européen. Nous y verrons parfois des légères similitudes tout comme des profondes divergences. Enfin, la deuxième partie examine les alternatives de protection retrouvables dans les deux systèmes. De même, nous évoquerons la disparité des offres entre les Etats-Unis et l'Union Européenne, avant d'examiner des alternatives plus générales.

¹⁵ Avis du Comité économique et social sur la "Proposition de directive du Parlement européen et du Conseil concernant la brevetabilité des inventions mises en œuvre par ordinateur" (COM(2002) 92 final — 2002/0047 (COD)), Journal officiel n° C 061 du 14/03/2003 p. 0154 - 0163, § 1.5, 3.8.

¹⁶ Scandale industriel et sanitaire lié à l'utilisation par le groupe Volkswagen, de 2009 à 2015, de différentes techniques visant à réduire frauduleusement les émissions polluantes (de NOx et de CO2) de certains de ses moteurs diesel et essence lors des tests d'homologation. Un logiciel avait été créé pour plafonner les émissions de gaz afin de passer les tests antipollution américains : https://www.lesechos.fr/13/01/2017/lesechos.fr/0211685386169_la-veritable-histoire-du-scandale-volkswagen.htm

¹⁷ Valat, Grimaud, Propositions pour un encadrement du régime juridique des logiciels, 01/02/2016, n°123, page(s) 34-36

Développement

PARTIE 1 : Les modes traditionnels de protection des programmes d'ordinateur dans les législations étasuniennes et européennes.

L'Union Européenne et les Etats-Unis se basent sur les mêmes notions afin de protéger les programmes d'ordinateur : le brevet (Chapitre 1) et le droit d'auteur (Chapitre 2). En revanche, il semble apparaître que ces notions n'aboutissent pas au même degré de protection de part et d'autre de l'atlantique, voire conduisent à un résultat opposé.

Chapitre 1 : Une opposition des législations en matière de brevet.

Il est important d'étudier globalement le brevet puisque notre monde économique, industriel et technologique tend à s'internationaliser. Cette comparaison nous permet de constater que, de façon opposée sur le fond, les deux systèmes posent une base législative (Section 1) qui est massivement complétée par la jurisprudence (Section 2).

Section 1 : Les divergences de bases législatives quant aux inventions brevetables.

La tendance est que les programmes d'ordinateurs se voient reconnaître la brevetabilité. Cette tendance ne semblait pas être présente dans la base du système européen des brevets (§ 1), tandis qu'elle fut très vite inscrite dans le système des Etats-Unis (§ 2).

Paragraphe 1 : La brevetabilité du logiciel en Europe : le refus de principe.

L'exclusion des programmes informatiques de la protection par le brevet est expressément inscrite dans la loi (A), malgré les tentatives vaines de faire évoluer ce droit (B).

A) Une non-brevetabilité presque catégorique.

Bien que l'article 52(1) de la Convention de Munich sur la délivrance des brevets européens (CBE) du 5 Octobre 1973¹⁸ autorise les brevets à être attribués « *pour toute invention dans tous les domaines technologiques* », l'article 52(2) exclu les programmes de la brevetabilité « en tant que tels », répondant au motif que les logiciels n'appartiennent pas à un domaine de

¹⁸ Le système des brevets européens constitue un ordre juridique propre constitué de 38 Etats membres, à savoir l'ensemble des Etats membres de l'Union Européenne et 11 autres Etats.

la technique. Les raisons pour cette exclusion étaient principalement politiques¹⁹ et économiques.²⁰ Le système européen dispose donc d'une exclusion précise concernant les programmes informatiques.

Il y avait un argument avançant que cette exclusion n'était pas nécessaire et que seuls la non inventivité et non-évidence devraient rester des obstacles à la brevetabilité. L'article 27 de l'ADPIC par exemple, bien qu'excluant les tissus humains et animaliers, semble n'envisager aucune autre exclusion et requiert simplement « *qu'un brevet pourra être obtenu pour toute invention, de produit ou de procédé, dans tous les domaines technologiques, à condition qu'elle soit nouvelle, qu'elle implique une activité inventive et qu'elle soit susceptible d'application industrielle.* » L'exclusion catégorique inscrite en l'article 52(2) peut alors soulever des questions, laissant se demander pourquoi les développements si utiles tel que des programmes avec application industrielle directe seraient exclus per se tandis que tous les autres domaines de technologie appliquée sont inclus.

En revanche, cette exclusion doit être interprétée restrictivement dans la mesure où l'article 52(3) établit que : « *Le paragraphe 2 n'exclut la brevetabilité des éléments qu'il énumère que dans la mesure où la demande de brevet européen ou le brevet européen ne concerne que l'un de ces éléments, considéré en tant que tel.* »

B) Une tentative d'évolution avortée.

Le problème est que le domaine des programmes d'ordinateur a évolué très rapidement, et il n'était sans doute pas possible pour les législateurs de l'époque de prédire un tel engouement pour ce domaine. Plusieurs tentatives ont été faites afin d'adopter une approche plus libérale.

Une proposition de directive publiée le 20 Février 2002 concernant la brevetabilité des inventions mises en œuvre par ordinateur²¹ prévoyait une brevetabilité possible des inventions « *dont l'exécution implique l'utilisation d'un ordinateur, d'un réseau informatique ou d'autres appareils programmables et présentant une ou plusieurs caractéristiques [...] qui sont réalisées totalement ou en partie par un ou plusieurs programmes d'ordinateur.* »²²

¹⁹ L'OEB voulait suivre les Etats-Unis, qui avaient par exemple refusé la protection des programmes informatiques par le brevet avec son arrêt de la Cour Suprême des Etats-Unis du 20 Novembre 1972, *Gottschalk v Benson*, 409 US 63.

²⁰ L'Europe avait un grand retard sur les Etats-Unis au niveau de la technologie, alors ne pas permettre de breveter un programme était une solution pour laisser libre court à l'innovation, voir *Jurisclasseur* 2009, Fasc. 4220 n°73 : exclusion de brevetabilité - Règles relatives au logiciel, Christian Le Stanc

²¹ Proposition de directive du Parlement européen et du Conseil concernant la brevetabilité des inventions mises en œuvre par ordinateur, *Journal officiel* n° 151 E du 25/06/2002 p. 0129 - 0131

²² Article 2(a) de la proposition de directive

Cette directive aurait demandé aux Etats membres d'adopter une approche plus souple afin d'attribuer une protection aux inventions relatives aux logiciels, à condition tout de même qu'elle respecte les critères généraux de brevetabilité.²³ La condition essentielle à surmonter aurait été la contribution technique apportée par le logiciel, permettant ainsi une vraie analyse.²⁴ Mais ce projet controversé fut l'origine de vifs débats, et fut finalement rejeté par le parlement en 2005.

Même si le débat a été réactivé à la suite de l'adoption du règlement n°1257/2012²⁵ sur le brevet unitaire pour les pays de l'Union faisant partie de la coopération renforcée concernant ce domaine, il est fort de constater qu'aucune avancée n'a été réalisée. Ainsi, l'interdiction de principe demeure.

Paragraphe 2 : La brevetabilité du logiciel aux Etats-Unis : une ouverture de principe.

Contrairement à l'Europe, les Etats-Unis ne disposent pas d'une exclusion visant expressément les programmes (A), d'autant plus que les tribunaux ont contribué à l'expansion de leur protection déjà ouverte (B).

A) L'ouverture législative des inventions brevetables.

La décision européenne d'exclure les programmes informatiques de la protection par le brevet fut très critiquée car elle ne reflète pas la réalité industrielle. De plus, elle diverge de l'approche des Etats Unis qui était plus flexible dans l'attribution de brevets de logiciels.

En effet, le Patent Act,²⁶ en son article 35 U.S.C. § 101, intitulé « Inventions patentable », dispose seulement que quiconque invente ou découvre « *tout procédé, machine, procédé de fabrication ou composition de matière, nouveau et utile, ou tout perfectionnement nouveau et utile de ces inventions* » peut obtenir un brevet. Dans l'arrêt *Diamond v Chakrabarty*, la Cour Suprême a relevé qu'en choisissant des termes aussi larges que « fabrication » et « composition de matière », associés à un large « tout », le Congrès a clairement considéré que le droit des brevets doit avoir un champ large.²⁷ Elle reprend également les propos émis

²³ Article 4(1) de la proposition de directive.

²⁴ Article 3 et article 4(2) de la proposition de directive

²⁵ Règlement (UE) n ° 1257/2012 du Parlement européen et du Conseil du 17 décembre 2012 mettant en œuvre la coopération renforcée dans le domaine de la création d'une protection unitaire conférée par un brevet, Journal Officiel n° L 361, p. 1–8

²⁶ U.S. Patent Act, 35 U.S.C. §§ 1 et seq., § 101 : "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title."

²⁷ Cour Suprême des États-Unis du 16 Juin 1980, *Diamond c. Chakrabarty*, 447 U.S. 308, 309 : "In choosing such expansive terms as "manufacture" and "composition of matter," modified by the comprehensive "any" Congress plainly contemplated that the patent laws would be given wide scope."

lors des travaux préparatoires du Patent Act 1952 selon laquelle le champ de la brevetabilité inclut « tout ce qui a été fait par l'homme ». ²⁸ Les cours fédérales ont pris le soin d'y rajouter des restrictions, comme les idées abstraites ou les lois de la nature qui ne peuvent pas être brevetées. ²⁹ Seule la mise en œuvre *pratique et utile* d'une idée abstraite, d'une loi de la nature ou d'un phénomène naturel peut être brevetée.

C'est dans ce contexte qu'il faut apprécier l'évolution de la jurisprudence américaine face aux inventions logicielles, d'autant que les conditions de nouveauté et utilité posées par le Patent Act ne sont pas difficiles à surmonter. On peut donc noter l'absence d'une exclusion légale similaire à celle en Europe.

B) La précision jurisprudentielle quant à la brevetabilité des logiciels.

La base législative étant ouverte, il revenait à la jurisprudence de préciser comment les programmes informatiques devaient s'y insérer.

D'abord exclus par l'arrêt *Gottschalk v Benson*, ³⁰ la protection des programmes informatiques fut affirmée par l'arrêt *Diamond v Diehr*, ³¹ disposant que tant qu'un système dans son ensemble était l'objet de la demande, la brevetabilité du programme de ce système pouvait se baser sur celle du fonctionnement général du système dès lors que le programme était mis en œuvre dans une application industrielle spécifique. Le raisonnement fut assoupli par l'arrêt *Re Alappat* ³² qui considère l'ordinateur comme une machine et le programme comme un « perfectionnement nouveau et utile. » Dans d'autres arrêts comme *AT & T Corp v Excel Communications*, ³³ la Cour a disposé que le test était basé sur l'identification d'un résultat concret, utile et tangible du programme.

Toutes ces décisions ont contribué à la libéralisation des critères de brevetabilité des programmes d'ordinateurs aux Etats Unis. De ce fait, l'USPTO est favorable à la brevetabilité des logiciels, y compris lorsqu'ils ne produisent pas d'effet technique. Cela est lié au fait que l'approche américaine, qui diffère de l'approche européenne, repose sur des notions comme la transformation d'un élément physique ou l'utilité d'une machine. ³⁴ Cela a pu mener l'USPTO à breveter des méthodes commerciales sous forme de programme tout comme le « 1-click

²⁸ Ibid

²⁹ Cour Suprême des Etats-Unis du 10 Juillet 1853, *Le Roy v. Tatham*, 55 U.S. 156, 175

³⁰ Cour Suprême des Etats-Unis du 20 Novembre 1972, *Gottschalk v Benson*, 409 US 63, 64

³¹ Cour Suprême des Etats-Unis du 3 Mars 1981, *Diamond v Diehr*, 450 US 175.

³² Cour d'appel des États-Unis pour le circuit fédéral du 29 Juillet 1994, *Re Alappat*, 33 F 3d 1526.

³³ Cour d'appel des États-Unis pour le circuit fédéral du 14 Avril 1999, *AT & T Corp v Excel Communications*, 172 F 3d 1352.

³⁴ « the machine-or-transformation test »

purchase » d'Amazon.com (qui a expiré le 12 Septembre 2017),³⁵ ou encore des brevets pour les interfaces graphiques utilisateurs (GUI)³⁶ sous des *utility patents*,³⁷ pour l'apparence du logiciel.³⁸

Ainsi, les tribunaux aux Etats-Unis ont largement participé à la précision législative concernant les logiciels, afin de prendre une direction que l'Union Européenne rejetait.

Section 2 : Opposition des développements et des exceptions jurisprudentielles.

Il n'y a presque aucun doute que de nos jours, les inventions mises en œuvre par un ordinateur peuvent être brevetées. Les demandes auprès de l'Office européen des brevets (OEB) atteignent les 100 par mois,³⁹ et près de 30000 brevets avaient été attribués par l'Office en 2003.⁴⁰ Il n'y a pas de statistiques fiables depuis 2003, mais l'on peut imaginer que le nombre n'a cessé d'augmenter. Il faudra donc examiner cette évolution européenne (§ 1), qui doit être contrastée avec le renfermement étasunien (§ 2).

Paragraphe 1 : Une volonté grandissante de protection en Europe.

Malgré l'exclusion de base à la protection par le brevet, l'OEB a conscience que les programmes sont dignes de protection et va interpréter l'article 52(2) en ce sens (A). Cela conduit l'Office à aborder une jurisprudence extensive (B).

A) Le sentiment de nécessité reflété dans l'interprétation de la loi.

L'interdiction de principe est largement évitée par les praticiens et l'OEB, ce dernier acceptant le dépôt de brevet incluant un logiciel (donc pas « en tant que tel ») depuis longtemps. L'astuce est d'éviter les termes « logiciel » et « information » lors du dépôt de demande de brevet : il faut rédiger le brevet afin que le logiciel soit présenté comme une étape dans un procédé matériel ou physique au sein d'une machine. D'ailleurs, c'était

³⁵ Brevet n°US5960411, accessible via le lien suivant sur le site de l'EPO:
https://worldwide.espacenet.com/publicationDetails/biblio?FT=D&date=19990928&DB=&locale=en_EP&CC=US&NR=5960411A&KC=A&ND=3#

³⁶ Ensemble de règles qui régissent la présentation visuelle ainsi que le comportement des interfaces graphiques, notamment l'usage des couleurs, la typographie, la présentation, la signification des logos et des icônes, la présentation des fenêtres, l'emplacement, la forme et le comportement des widgets, et les formes du curseur.

³⁷ Brevets aux Etats-Unis.

³⁸ Voir l'*utility patent* US9223477B2 de Google, qui protège un menu en haut de page :
<https://patents.google.com/patent/US9223477B2/en>

³⁹ Commission Européenne, Base de donnée des Communiqués de Presse, 20 Février 2002, Proposal for a directive on the patentability of computer implemented inventions – frequently asked questions, <http://europa.eu/rapid/press-release_MEMO-02-32_en.htm?locale=FR>

⁴⁰ <http://eupat.ffii.org/>

spécifiquement pour répondre à cette problématique de contournement par la jurisprudence que le projet de directive du 20 Février 2002 avait été mis en route.⁴¹

En ce sens, dans l'arrêt *Computer program product c. IBM*, la Chambre, mentionnant les développements aux Etats Unis et Japon concernant la protection des programmes, a pu les remarquer ceci : « *Il n'en demeure pas moins que, comme l'a fait observer le requérant, ces développements peuvent être considérés comme une indication utile des tendances les plus récentes. Pour la Chambre, ils peuvent favoriser la poursuite de l'harmonisation, hautement souhaitable, du droit des brevets à l'échelle mondiale.* »⁴² Ce commentaire nous montre que l'OEB n'est pas insensible aux motivations étasuniennes.

L'Office avait pu déjà montrer une volonté d'octroyer une protection par le brevet aux logiciels en 1984 : « *une invention qui serait brevetable au regard des critères classiques de brevetabilité ne doit pas être exclue de la protection du simple fait que des moyens techniques modernes tel un programme d'ordinateur sont employés pour sa réalisation.* »⁴³ Mais l'OEB prend garde à distinguer le logiciel « brut » (en tant que tel) du logiciel intégré à un ensemble (invention mise en œuvre par ordinateur), car seul le second est brevetable.

Par conséquent, l'interprétation du « en tant que tel » va être importante concernant les inventions mises en œuvre par ordinateur.

B) La jurisprudence européenne au service des brevets.

Une première vraie évolution de la jurisprudence de l'OEB consiste à dire qu'une invention mise en œuvre par ordinateur est brevetable dès que le « caractère technique » est constaté, c'est-à-dire que l'invention a un effet technique ou la solution constaté à un effet technique.⁴⁴

Plus tard, l'OEB exposera le critère de « *l'effet technique supplémentaire* » en 1999,⁴⁵ impliquant que les inventions mises en œuvre par ordinateur qui présenteraient des caractéristiques techniques doivent aller « *au-delà des interactions physiques normales existant entre le programme (logiciel) et l'ordinateur (matériel) sur lequel il fonctionne* », et également apporter de façon autonome des particularités techniques propres excluant le simple traitement d'information. La Chambre reconnaitra que cette décision de 1999 est

⁴¹ David Forest, *Droit des logiciels*, 2017, Lextenso édition, p 21

⁴² Office Européen des Brevets, affaire T 1173/97, Chambre des recours du 1er Juillet 1998, *Computer program product c. IBM*, ECLI:EP:BA:1998:T117397.19980701, para 2.6

⁴³ Office Européen des Brevets, affaire T 208/84, Chambre de Recours Technique du 15 juillet 1986, *Vicom Systems Inc., JO OEB 1987/014*.

⁴⁴ *Ibid* [16] ou 79 et 80-1

⁴⁵ Office Européen des Brevets, affaire T 1173/97, Chambre des recours du 1er Juillet 1998, *Computer program product c. IBM*, ECLI:EP:BA:1998:T117397.19980701

légèrement différente des précédentes mais n'y voit pas d'inconsistance, et n'est pas allée au-delà de la signification disposé dans la CBE.⁴⁶ Enfin, en 2004, l'OEB semble être contenté de l'existence de moyens techniques.⁴⁷ On exclut donc les programmes lorsqu'ils sont abstraits et n'apportent rien techniquement.⁴⁸ Ainsi, tout ce qui présente une technicité spécifique allant au-delà est, au contraire, potentiellement brevetable bien que se présentant sous forme logicielle. L'article 52 a donc été modifié, et le caractère technique de l'invention semble faire disparaître la restriction.⁴⁹

Dès lors, l'OEB explique « *qu'une revendication portant sur un procédé technique réalisé sous la commande d'un programme [que celui-ci soit mis en œuvre au moyen de matériel ou d'un logiciel] peut être admissible car c'est pour l'application du programme qui détermine la succession des étapes du procédé que la protection est en fait recherchée* ». ⁵⁰ Cette consécration du critère de caractère technique fut réitérée dans plusieurs affaires ultérieures, en plus des critères de nouveauté et inventivité des aspects fonctionnels.⁵¹ Ainsi, le programme doit être utilisé dans un contexte technique spécifique capable d'application industrielle afin d'être protégé, même s'il ne peut pas être protégé seuls.⁵² Cela a conduit l'OEB à rajouter dans ses Directives d'examen qu'une invention comprenant un programme peut être brevetable si elle produit un effet technique et que cette demande ne doit pas être automatiquement exclue si elle contient un programme d'ordinateur.⁵³ Il en va de même pour les GUI, qui peuvent être brevetées si elles apportent une solution technique à un problème technique,⁵⁴ même si cela reste complexe à démontrer.

⁴⁶ Ibid para 10.2

⁴⁷ Office Européen des Brevets, affaire T 0258/03, Chambre des recours du 21 Avril 2004, Méthode d'enchères/HITACHI, 21 Avril 2004, ECLI:EP:BA:2004:T025803.20040421, para 4.7

⁴⁸ Office Européen des Brevets, affaire T 1173/97, Chambre des recours du 1er Juillet 1998, Computer program product c. IBM, ECLI:EP:BA:1998:T117397.19980701, para 11.5

⁴⁹ A Laakkonen and R Whaite, 'The EPO leads the way, but where to?', 2001, European Intellectual Property Review, 244, au sujet de l'affaire T931/95, Pension Benefit Systems Partnership de l'Office Européen des Brevets, Chambre des recours du 8 Septembre 2000, ECLI:EP:BA:2000:T093195.20000908

⁵⁰ Office Européen des Brevets, affaire T 208/84, Chambre de Recours Technique du 15 juillet 1986, Vicom Systems Inc., JO OEB 1987/014.

⁵¹ Dans Office Européen des Brevets, affaire T 0258/03, Chambre des recours du 21 Avril 2004, Méthode d'enchères/HITACHI, 21 Avril 2004, ECLI:EP:BA:2004:T025803.20040421, para 3.5 ; Office Européen des Brevets, affaire T 0424/03, Chambre des recours du 23 Février 2006, Microsoft (Clipboard format), ECLI:EP:BA:2006:T042403.20060223, para 5.1 ; Office Européen des Brevets, affaire T1188/04, Chambre des recours du 5 Mars 2008, Sharp/Graphical user interface, ECLI:EP:BA:2008:T118804.20080305, para 14.

⁵² Il est notable que c'est ce que prévoyait la proposition de directive de 2002 : ainsi, l'Office fait dans un sens renaître cette proposition avortée.

⁵³ Office Européen des Brevets, Directives relatives à l'examen pratiqué, Partie G-II (3.6) Programmes d'Ordinateurs ; Accessible via le lien : https://www.epo.org/law-practice/legal-texts/html/guidelines/f/g_ii_3_6.htm

⁵⁴ Ibid Directive G-II 3.7.1 ; voir Office Européen des Brevets, affaire T 1741/08, Chambre des recours du 2 Août 2012, GUI layout/SAP, ECLI:EP:BA:2012:T174108.20120802

L'OEB frôle jurisprudence américaine, tout en précisant la notion d'effet technique afin de ne continuer à accepter que des demandes de brevet présentant un véritable caractère technique. Il libéralise son système mais ne suit pas les Etats-Unis, n'acceptant pas le simple critère « d'utilité » trop laxiste. Certains auteurs considèrent même qu'il est possible de revendiquer un brevet pour un produit programme, c'est-à-dire un programme en tant que tel.⁵⁵

Paragraphe 2 : Un système s'essouffant de l'autre côté de l'Atlantique.

A contresens de la libéralisation du système européen des brevets, les Etats-Unis semblent refermer leur protection (A), l'impact se faisant ressentir auprès des détenteurs de brevets (B).

A) La fin d'un système trop accueillant.

Les Etats-Unis ont pu récemment infléchir leur position suite aux fameuses décisions *Bilski v Kappos*,⁵⁶ et *Alice corp v CLS Bank International*.⁵⁷ Ces deux arrêts exposent de nouvelles conditions de brevetabilité pour les méthodes d'affaires mises en œuvre sur un logiciel via un test de brevetabilité plus restrictif :⁵⁸

- 1) (a) La demande porte-t-elle sur l'une des catégories disposées à l'article 35 USC 101 (soit une machine, produit manufacturé, composition de matière, procédé) ?
(b) La demande porte-t-elle sur une idée abstraite ? Produit-elle un effet tangible ?
- 2) Si les effets sont tangibles, il faut ensuite se demander si l'invention comprend des caractéristiques additionnelles qui la rendent brevetable. C'est-à-dire un effet technique supplémentaire qui va au-delà du fonctionnement normal d'un ordinateur (cet effet pourrait résider alors dans une amélioration du fonctionnement de l'ordinateur).

De là, la simple mise en œuvre par un ordinateur générique ne transforme plus une idée abstraite en invention brevetable, et il en va de même si l'idée abstraite est appliquée par un ordinateur précis. Similairement à ce qu'a fait l'OEB, et en réponse à cette jurisprudence, le Manual of Patent Examining Procedure (MPEP) a été renouvelé afin de préciser les pratiques de l'Office concernant l'exclusion des idées abstraites. Il confirme que les demandes incluant par exemple un algorithme doivent être plus qu'une idée abstraite, notamment en démontrant

⁵⁵ Reto M. Hilty et Christophe Geiger, *Breveter le logiciel? Une analyse juridique et socio-économique*, 2005, *Propriétés Intellectuelles*, n° 16, p296, 301

⁵⁶ Cour Suprême des Etats Unis du 28 Juin 2010, *Bilski v Kappos*, 561 U.S., relative à une méthode de protection contre les risques de fluctuations de prix sur le marché de l'énergie

⁵⁷ Cour Suprême des Etats-Unis du 19 Juin 2014, *Alice corp v CLS Bank International*, 573 U.S.

⁵⁸ Ce test reprend le test de l'arrêt de la Cour suprême des États-Unis du 20 mars 2012, *Mayo c/ Prometheus*, 566 U.S.

une amélioration du fonctionnement d'un ordinateur ou dans un autre domaine technique, tout en donnant de nombreux exemples d'inventions logicielles exclues ou valides.⁵⁹

Cette récente jurisprudence montre un rapprochement des systèmes européen et étasunien en ce qu'ils arborent maintenant tous deux le critère de l'effet technique : l'un l'utilisant pour restreindre raisonnablement la brevetabilité (Etats-Unis) et l'autre pour libéraliser son système (Union-Européenne).

B) Un nouveau jour impactant les droits acquis.

Il faut tout de même noter que la nouvelle approche n'empêche pas de breveter un procédé mis en œuvre par un logiciel, mais elle en offre la possibilité d'une manière plus raisonnable qu'avec l'ancien critère de l'utilité. Par exemple, une invention portant sur un système de gestion de la mémoire d'un ordinateur a été jugé, bien qu'en l'application des arrêts *Alice* et *Mayo*, comme ne portant pas sur une idée abstraite et dès lors constituait une invention brevetable.⁶⁰ Il a pu être souligné que cette convergence vers le système européen aura un effet vertueux sur la guerre des brevets et le phénomène des patents trolls.⁶¹

On peut se douter que les grandes entreprises pourraient en prendre avantage et demander l'invalidation de brevets concurrents pour la raison officielle qu'ils ne sont plus en conformité avec la loi, mais surtout car ils sont nuisibles à leur activité. En conséquence, cette décision semble avoir entraîné un nombre important de décisions invalidant des brevets de logiciels ou de méthodes mises en œuvre par des logiciels, comme l'a relevé Steven Callahan.⁶² En effet, Callahan note qu'en 2015, depuis la décision *Alice*, 57 brevets ont été invalidés sur 78 décisions pour des affaires similaires. Dans *Digitech Image Technologies, LLC v. Electronic for Imaging*,⁶³ le brevet relatif à des informations générées sur une image et une méthode pour gérer ces informations fut invalidé. Comme autre exemple, un brevet concernant un système de sécurisation de transactions électroniques en ligne fut également invalidé dans *buySAFE, Inc., v. Google, Inc.*⁶⁴ On peut également penser qu'il sera plus difficile de breveter une GUI.

⁵⁹ USPTO, Manual of Patent Examining Procedure (MPEP), Chapter 2106.04(a) Abstract Ideas, Ninth Edition, Revision 08.2017, Last Revised January 2018 - accessible via le lien suivant :

https://www.uspto.gov/web/offices/pac/mpep/s2106.html#ch2100_d29a1b_13a9e_2dc

⁶⁰ Cour d'appel pour le Circuit Fédéral du 15 Août 2017, Visual Memory LLC v NVIDIA Corp, 2016-2254, relative au brevet US n°5.953.740

⁶¹ Laure Marino, « États-Unis : Alice au pays des brevets de logiciels », 6 Novembre 2014, Gazette du Palais - n°310 - page 20

⁶² Steven Callahan, "Alice : The Death of Software-Related Patents ?" (1er Mai 2015), accessible via le lien :

<http://www.ndtexpblog/?p=3550>

⁶³ Cour d'appel des États-Unis pour le circuit fédéral du 11 Juillet 2014, Digitech Image Technologies, LLC v. Electronics For Imaging, Inc. ,758 F. 3d 1344

⁶⁴ Cour d'appel des États-Unis pour le circuit fédéral du 3 Septembre 2014, buySAFE, Inc., v. Google, Inc, 765 F. 3d. 1350

Enfin, la particularité est que les deux notions clefs de cette nouvelle jurisprudence, soit « l'idée abstraite » et « l'effet technique supplémentaire », n'ont pas été définies par la Cour.⁶⁵ Cela rajoute dès lors une difficulté aux demandeurs de brevets pour un procédé mis en œuvre par un logiciel car ils restent dans l'incertitude de savoir si leur invention rentrera dans une de ces deux notions, le fruit de leurs investissements pouvant potentiellement être réduits à zéro.⁶⁶ Toutefois, ce test pourrait être surmonté en démontrant une amélioration du fonctionnement de l'ordinateur ou d'un autre domaine technique.⁶⁷ Les statistiques indiquent que les demandes de brevets de logiciels a même augmenté depuis 2015, montrant que les inventeurs ne sont pas effrayés par l'arrêt *Alice*.⁶⁸

Il semble alors que les Etats-Unis s'orientent vers un système où la brevetabilité des logiciels est pondérée, répondant à des critères plus restreints. Ce choix de critères rapproche sensiblement les Etats-Unis de l'Europe.

Pour conclure sur la brevetabilité des logiciels aux Etats-Unis et en Europe, il semblerait que ces deux systèmes, bien que très différents à la base, tendent vers une convergence. Les développeurs européens pouvant se permettre de breveter leur invention bénéficient donc d'une compétitivité renforcée. Ce qui ne peuvent se le permettre se tournent vers le droit d'auteur. Il convient à présent d'explorer ce régime, toujours en comparant ce qui est pratiqué de part et d'autre de l'Atlantique.

Chapitre 2 : Une convergence apparente des législations en matière de droit d'auteur.

Le droit d'auteur fut la protection privilégié par les institutions, et surtout par les Etats-Unis, quand se posait la question du genre de protection à attribuer aux logiciels. Il a fallu alors adapter les notions de ce régime aux particularités du programme (Sections 1) tout en assurant

⁶⁵ Gene Quinn, « Software Patent Eligibility at the Federal Circuit 2017 », IPWatchdog, 17 Décembre 2017 -accessible via le lien suivant : <http://www.ipwatchdog.com/2017/12/17/software-patent-eligibility-federal-circuit-2017/id=91169/>

⁶⁶ Malgré un Quick Reference Guide émis par l'USPTO afin de faciliter la recherche : <https://www.uspto.gov/sites/default/files/documents/ieg-qrs.pdf>

⁶⁷ Voir par exemple : Cour d'appel des États-Unis pour le circuit fédéral du 12 Mai 2016, Enfish LLC v. Microsoft Corp., 822 F.3d 1327) ; Cour d'appel des États-Unis pour le circuit fédéral du 8 Mars 2017, Thales Visionix Inc. v. U.S., 850 F.3d 1343

⁶⁸ Mark Summerfield, New USPTO Data Set Reveals Recent Trends in Software and Business Method Patent Grants and Filings, Patentology, 17 Avril 2017, accessible via le lien suivant : <https://blog.patentology.com.au/2017/04/new-uspto-data-set-reveals-recent.html>

l'application des exceptions du droit d'auteur (Section 2), à des degrés différents selon le système.

Section 1 : Une inclusion sans encombre des programmes informatiques dans les deux systèmes.

Tout comme pour les brevets, le régime du droit d'auteur préexistait aux logiciels. Il a fallu adapter ce droit afin d'offrir aux programmeurs une protection sans besoin de brevet. Il conviendra d'examiner cette adaptation au sein des systèmes européen (§ 1) et étasunien (§ 2).

Paragraphe 1 : L'application des critères connus du droit d'auteur européen.

Il s'agit ici de constater la manière dont les programmes informatiques ont été considérés comme œuvre littéraire (§ 1), et dont ils se sont vu appliquer le critère de l'originalité (§ 2).

A) Le programme d'ordinateur comme œuvre littéraire.

La question s'étant d'abord posée au niveau des Etats membres, l'UE ne pouvait pas rester inactive dans le développement juridique de la protection des programmes informatiques. L'UE a adopté la Directive 91/250/CEE⁶⁹ comme réponse au débat, mais surtout pour le fait que sans harmonisation, le fonctionnement du Marché Unique est compromis. Ce fut ensuite au tour de l'accord ADPIC du 15 Décembre 1993, en son article 10, d'octroyer une protection aux programmes d'ordinateur en tant qu'œuvres littéraires en vertu de la Convention de Berne.

Différents Etats Membres avaient adopté des attitudes divergentes pour protéger les programmes informatiques. Ainsi, même si certaines juridictions avaient déjà basé leur protection sur le droit d'auteur, cela n'était pas le cas pour l'Allemagne qui considérait que les programmes informatiques étaient des produits scientifiques et technologiques, et pas des œuvres littéraires.⁷⁰ De ce fait, l'UE adopta cette directive en 1991 qui entra en vigueur le 1^{er} Janvier 1993. Un rapport sur son application établi par la Commission Européenne montre que sa mise en œuvre par les Etats Membres est satisfaisante et ses effets sont bénéfiques, ce qui amena la CE à conclure que la directive n'avait pas besoin de révision.⁷¹ En effet, aucune

⁶⁹ Anciennement Directive 91/250/CEE du Conseil, du 14 mai 1991, concernant la protection juridique des programmes d'ordinateur, OJ L 122, 17.5.1991, p. 42-46 (abrogée)

⁷⁰ A Wiebe, « European copyright protection of software from a German perspective », 1993, Computer Law & Practice, Vol 9, p 79

⁷¹ Rapport de la Commission au Conseil, au Parlement européen et au Comité économique et social sur la mise en œuvre et les effets de la directive 91/250/CEE concernant la protection juridique des programmes d'ordinateur, COM/2000/0199 final, p 21

modification n'a été faite pour cette directive dans la Directive 2001/29/CE,⁷² et la Directive 91/250/CEE fut codifiée mais non modifiée en 2009.⁷³

D'après son article 1(1) : « *les États membres protègent les programmes d'ordinateur par le droit d'auteur en tant qu'œuvres littéraires au sens de la convention de Berne pour la protection des œuvres littéraires et artistiques. Le terme « programme d'ordinateur », aux fins de la présente directive, comprend le matériel de conception préparatoire.* » Les programmes informatiques sont donc ajoutés à la liste des œuvres littéraires : le code écrit est protégé mais pas sa fonctionnalité. La protection par le droit d'auteur attribue ainsi au développeur des prérogatives exorbitantes, principalement disposées à l'article 4 de la directive : l'auteur peut contrôler la reproduction de l'œuvre, peut la modifier ou l'adapter, la mettre sur le marché et peut en tirer profit. L'auteur bénéficie également de droits moraux.⁷⁴

La particularité de l'Union Européenne, contrairement aux Etats Unis, est qu'elle n'est pas un Etat Fédéral. Ce qui est alors remarquable est qu'elle utilise le principe de l'interprétation autonome et uniforme en droit d'auteur, comme par exemple dès les arrêts *Sena*⁷⁵ et surtout *Infopaq*⁷⁶ afin de pouvoir construire un véritable droit d'auteur européen.⁷⁷

B) L'incontournable critère de l'originalité appliqué aux programmes informatiques.

Le droit d'auteur a comme condition l'originalité de l'œuvre. La définition de ce critère n'est pas sans difficulté. De surcroît, dans un système comme celui de l'UE, ce qui peut poser problème est la disparité d'interprétation de cette condition d'originalité entre le système de droit Civiliste et la Common Law. Le système civiliste voit cette condition comme l'approche innovatrice et créative spécifique à l'auteur, se rapprochant de sa personnalité, alors que le

⁷² Directive 2001/29/CE du Parlement européen et du Conseil du 22 mai 2001 sur l'harmonisation de certains aspects du droit d'auteur et des droits voisins dans la société de l'information, OJ L 167, 22.6.2001, p. 10–19

⁷³ Directive 2009/24/CE du Parlement européen et du Conseil du 23 avril 2009 concernant la protection juridique des programmes d'ordinateur (version codifiée) OJ L 111, p. 16–22

⁷⁴ Certes, limités, comme par exemple en France où l'auteur bénéficie des dispositions des articles L. 121-1 et L. 121-4 du Code de la Propriété Intellectuelle, mais sous réserve de l'application de l'article L. 121-7 : il ne peut pas invoquer le droit de repentir ou de retrait, ne peut pas s'opposer à une modification ou adaptation lorsqu'elle n'est préjudiciable ni à son honneur ni à sa réputation.

⁷⁵ Arrêt de la Cour de Justice de l'Union Européenne du 6 février 2003, aff. C-245/00, *Sena*, Rec., p. I-1251.

⁷⁶ Arrêt de la Cour de Justice de l'Union Européenne, quatrième chambre du 16 juillet 2009, affaire C-5/08, *Infopaq International A/S contre Danske Dagblades Forening*, Rec., p. I-6569.

⁷⁷ Céline CASTETS-RENARD, L'interprétation du droit d'auteur européen sur les logiciels par la Cour de justice, 1er novembre 2012, *Revue Lamy Droit de l'immatériel*, no. 87.

système de la Common Law considère le critère de l'originalité comme l'effort de l'auteur (*skill, labour and judgement*).⁷⁸

Toutefois, la Directive 2009/24/EC dispose cette condition en son article 1(3) : « *Un programme d'ordinateur est protégé s'il est original, en ce sens qu'il est la création intellectuelle propre à son auteur. Aucun autre critère ne s'applique pour déterminer s'il peut bénéficier d'une protection.* » Bien que complexe, le programme informatique doit satisfaire cette condition. La Directive 2009/24/EC cherche à harmoniser cette pluralité d'interprétations, mais un conflit persiste puisque la phrase « *création intellectuelle propre à son auteur* » peut malheureusement être interprété des deux manières. Certains commentateurs ont donc appelé à plus de renvois préjudiciels devant la CJUE concernant les programmes informatiques.⁷⁹

En pratique, les tribunaux regardent principalement si le programme est une œuvre propre à son auteur, dans le sens où elle n'est pas copiée, et ne regardent pas vraiment s'il y a une marque personnelle de l'auteur.⁸⁰ L'une des raisons pour cette analyse réduite est que le code source est principalement tenu secret et les tribunaux ne peuvent que supposer l'originalité en appréciant l'aspect extérieur du programme.⁸¹ De plus, un juge de formation purement juridique peut éprouver des difficultés à établir l'originalité dans un domaine si technique. On va alors se dire que c'est par la liberté de choix qu'est traduite l'originalité : si l'auteur avait le choix, alors on assumera que le travail vient de lui et porte son empreinte.⁸² Le principal problème est la standardisation des expressions, c'est-à-dire quand une fonctionnalité ne peut être exprimée que d'une manière plus ou moins précise.

S'agissant des interfaces graphiques, le *look and feel*⁸³ est à mi-chemin des fonctionnalités de l'interface. L'interface graphique utilisateur permet la communication entre le programme

⁷⁸ Voir par exemple l'utilisation du critère de l'originalité, à savoir l'utilisation des *skill, labour and judgement* de l'auteur, dans l'arrêt de la Haute Cour de justice d'Angleterre et du Pays de Galles, Chancery Division du 30 Juillet 2004, *Navitaire, Inc v Easyjet Airline Company and Bulletproof Technologies, Inc*, England and Wales High Court Report (Ch), 1725, para 127.

⁷⁹ J Deene, « Originality in software law: Belgian doctrine and jurisprudence remain divided », *Journal of Intellectual Property Law & Practice*, Vol 2, 692, 698.

⁸⁰ Ashwin van Rooijen, *The Software Interface Between Copyright and Competition Law : A legal analysis of interoperability in computer programs*, 2010, Kluwer Law International, p 57

⁸¹ Les programmes informatiques sont souvent disponible sous la forme de code objet, incompréhensible pour l'humain et nécessitant d'être décompilé pour extraire le code source : c'est la *closed code practice* ; James Gibson, « Once and Future Copyright », (2005) 81 *Notre Dame Law Review*, 167, 175

⁸² Par exemple en France : Tribunal de Commerce de Bobigny du 21 Janvier 1995, *Computer Associates International, Inc v Faster and Altai*, 166 RIDA 324 ; Voir aussi Proposal for a Council Directive on the legal protection of computer programs (Commission Européenne du 17 Mars 1989), COM (88) 816 final, para 2.7

⁸³ Ensemble de règles qui régissent la présentation visuelle ainsi que le comportement des interfaces graphiques, notamment l'usage des couleurs, la typographie, la présentation, la signification des logos et des icônes, la présentation des fenêtres, l'emplacement, la forme et le comportement des widgets, et les formes du curseur.

informatique et l'utilisateur (via des icônes, symboles, cases de saisie de données...).⁸⁴ A ce sujet, l'Union Européenne est plus claire que les Etats-Unis. La CJUE a pu juger que l'interface graphique utilisateur n'est pas une forme d'expression, puisqu'elle ne permettrait pas aux tiers de reproduire le programme, et donc ne relève pas de la directive 91/250/CE.⁸⁵ En revanche, l'interface graphique utilisateur peut bénéficier du régime commun du droit d'auteur de la Directive 2001/29/CE à condition qu'elle en respecte les conditions : il faut caractériser l'originalité puis exclure les interfaces uniquement dictées par la fonction technique.⁸⁶

Paragraphe 2 : L'application des critères connus du droit d'auteur étasunien.

Le droit d'auteur aux Etats-Unis requiert des conditions additionnelles. Bien que similairement à l'UE, il ne semble pas avoir de difficulté à intégrer les logiciels au régime du droit d'auteur (A), il faut que le programme réponde aux conditions de fixation écrite et d'enregistrement, en plus de l'originalité (B).

A) L'adaptation au droit d'auteur : Une œuvre littéraire écrite.

La section 102(a) du Copyright Act définit les œuvres protégeables comme suit : « *La protection du copyright s'étend, conformément aux dispositions du présent titre, aux œuvres de l'esprit originales fixées sous une forme tangible d'expression...* ».⁸⁷

On retrouve alors la même caractéristique que pour les brevets aux Etats-Unis : la définition de base consiste en une liste ouverte des œuvres protégeables et un critère (ici le critère de l'originalité) applicable à l'ensemble des œuvres. Les œuvres littéraires sont comprises dans les œuvres protégées. Elles sont définies comme « *des œuvres autres qu'audiovisuelles qui sont exprimées sous forme de mots, de nombres ou d'autres symboles ou signes verbaux ou numériques, indépendamment de la nature des objets matériels, tels que livres, périodiques, manuscrits, phonogrammes, films, bandes, disques ou cartes, qui leur servent de support.* »⁸⁸

⁸⁴ A ne pas confondre avec les interfaces, qui assurent l'interconnexion et interaction entre des logiciels et des matériels : Directive 91/250/CE, considérant 11.

⁸⁵ Cour de Justice de l'Union Européenne, troisième chambre du 22 Décembre 2010, affaire C-393/09, *Bezpečnostní softwarová asociace*, Journal officiel n° C11 ; Cour de Justice de l'Union Européenne, Grande Chambre du 2 mai 2012, *SAS Intitute Inc. c/ World Programming Ltd.*, affaire C-406/10, RLDI 2012/82, n° 2741.

⁸⁶ Cour de Justice de l'Union Européenne, troisième chambre du 22 Décembre 2010, affaire C-393/09, *Bezpečnostní softwarová asociace*, Journal officiel n° C11 , point 42

⁸⁷ "Copyright protection subsists, in accordance with this title, in original works of authorship fixed in any tangible medium of expression [...]"

⁸⁸ Section 101, Copyright Act 1976, 17 U.S.C. §§ 101-810, Pub.L. 94-553 : "Literary works" are works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the

Les enregistrements concernant des programmes d'ordinateur commencèrent à être acceptés par le Copyright Office dès 1964. Le Copyright Act 1976 ne les mentionnait pas au début, les programmes d'ordinateurs seront inclus explicitement par le Copyright Amendment Act du 12 Décembre 1980. La section 101 les définit comme « *un ensemble d'indications ou d'instructions destiné à être utilisé directement ou indirectement dans un ordinateur pour permettre d'obtenir un certain résultat* », contrairement à l'UE qui ne fait que préciser l'étendue de ce qui est compris par un programme d'ordinateur.⁸⁹

Les programmes d'ordinateurs sont considérés comme une sous-catégorie des œuvres littéraires, comme indiqué à la section 101, et sont protégés comme telles sous réserve de règles spécifiques. L'incorporation des programmes au régime étasunien du droit d'auteur ne semble par conséquent sensiblement pas différente de l'incorporation de ceux-ci dans le régime européen.

Tout comme en Europe, la protection par le droit d'auteur octroi à l'auteur des droits exclusifs de faire ou autoriser à faire.⁹⁰ L'auteur a le droit de créer des copies de son travail, créer un travail dérivé de l'œuvre originale, disséminer des copies de son travail par la vente, le transfert de propriété, la location, le prêt, et de publier son travail.

B) Les critères de fixation écrite, d'originalité et d'enregistrement appliqués aux programmes informatiques.

Les Etats-Unis ont choisi de donner comme condition que le travail bénéficiant du droit d'auteur soit 'un écrit' et refuse d'accorder protection à un travail de nature utilitaire.

Cela laisse penser que si le *code source*, tel que le programmeur l'écrit,⁹¹ pourrait bénéficier de cette protection, il n'en serait pas le cas pour le *code objet*, traduction du code source dans

material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied.

⁸⁹ Directive 2009/24/CE, considérant 7 : « *Aux fins de la présente directive, les termes «programme d'ordinateur» visent les programmes sous quelque forme que ce soit, y compris ceux qui sont incorporés au matériel. Ces termes comprennent également les travaux préparatoires de conception aboutissant au développement d'un programme, à condition qu'ils soient de nature à permettre la réalisation d'un programme d'ordinateur à un stade ultérieur.* »

⁹⁰ Article 17 U.S.C. § 106 : "Subject to sections 107 through 122, the owner of copyright under this title has the exclusive rights to do and to authorize any of the following: (1) to reproduce the copyrighted work in copies or phonorecords; (2) to prepare derivative works based upon the copyrighted work; (3) to distribute copies or phonorecords of the copyrighted work to the public by sale or other transfer of ownership, or by rental, lease, or lending; (4) to perform the copyrighted work publicly; [...] (5) to display the copyrighted work publicly; [...]"

⁹¹ Le code source (généralement appelé « code ») est le programme tel qu'écrit par le programmeur et peut être écrit en différents langages informatiques, comme le C, le C++ ou le Java. Ce sont des langages littéraires à part entière, avec leurs propres règles de syntaxe et grammaire, et nécessitent au lecteur de connaître le langage.

un langage compris par la machine afin de lui donner des ordres, ne pouvant pas être interprété comme un « écrit ».⁹² Mais les tribunaux ont confirmé que les programmes informatiques bénéficient de la protection pour leur code source et code objet, à condition qu'ils remplissent la condition d'originalité.⁹³ La fonction du programme, et le fait qu'il soit utilisé par l'homme ou une machine, n'a pas d'influence sur la protection.⁹⁴ La protection existe alors de façon indépendante par rapport à la nature du programme, malgré ce critère de fixation écrite. Enfin, les tribunaux ont également conclu que la manière dont le code était fixé à l'écrit n'était pas importante.⁹⁵

Afin d'obtenir la protection, le programme doit être original. C'est une condition universelle du droit d'auteur. Mais la Cour Suprême des Etats-Unis a pu dire que cette condition n'était pas stricte.⁹⁶ L'œuvre doit provenir de l'auteur, dans le sens où elle ne doit pas être copiée, et doit avoir un minimum de créativité.⁹⁷ Ainsi, le programme ne peut être automatiquement protégé par le droit d'auteur puisqu'il doit répondre à la condition d'originalité, car seule l'expression originale de l'auteur bénéficie de ce régime.

Une autre condition est que, dans le cas où le droit d'auteur est enfreint par une personne tierce et que l'auteur souhaite engager une action, l'œuvre doit avoir été au préalable déposée auprès de l'US Copyright Office.⁹⁸ En effet, la protection par le droit d'auteur est automatique, mais il faut tout de même déposer son code afin de prévenir tout litige.⁹⁹ Il y a également d'autres avantages à déposer son programme. Tout d'abord, cela permet de garder une trace de ses droits, tout comme un constat d'huissier ou une enveloppe Soleau en France.¹⁰⁰ Cela crée aussi une présomption que tout ce qui est revendiqué par l'auteur dans

⁹² Le code objet est la version traduite ou compilée du code source afin que l'ordinateur puisse comprendre la tâche à accomplir, ne pouvant traiter que des pulsations électroniques. Par exemple, la traduction finale est de transposer le code source en des séries de pulsation électroniques représenté par des séries de chiffres 1 et 0 symbolisant la présence ou absence de pulsation. Cette conversion du code source au code objet est généralement réalisée par un programme appelé le « compilateur » ou « interpréteur » selon le type de langage utilisé.

⁹³ Cour d'Appel des Etats-Unis pour le troisième circuit, 24 Juin 1981, Williams Elecs., Inc. v. Artic Int'l, Inc., 685 F.2d 870 ; Cour d'appel des États-Unis pour le troisième circuit, 30 Août 1983, Apple Computer Inc v Franklin Computer Corp., 714 F.2d 1240, 219 USPQ 113 ; Cour d'appel des États-Unis pour le neuvième circuit, 8 Février 1984, Apple Computer Inc. v Formula Intern. Inc., 725 F.2d 521, 525.

⁹⁴ Ibid

⁹⁵ Ibid

⁹⁶ Cour Suprême des Etats Unis du 27 Mars 1991, Feist Publications, Inc., v. Rural Telephone Service Co., 499 U.S. 340, 358 (1991).

⁹⁷ Ibid 345

⁹⁸ 17 U.S.C. § 411(a) : Except for an action brought for a violation of the rights of the author under section 106A(a), and subject to the provisions of subsection (b),[1] no civil action for infringement of the copyright in any United States work shall be instituted until preregistration or registration of the copyright claim has been made in accordance with this title.

⁹⁹ Les détails sur cette procédure sont contenus dans la US Copyright Office, Circular 61, Copyright Registration of Computer Programs

¹⁰⁰ En revanche, en France, il ne s'agit pas d'une condition préalable pour faire valoir ses droits, mais plutôt une assurance additionnelle permettant de sécuriser dès le départ son droit d'auteur.

son dépôt est vrai, renversant la preuve sur le défendeur en cas de litige. Enfin, cet enregistrement permet d'obtenir des dommages et intérêts substantiels en cas de litige, et peut servir à dissuader les potentiels contrefacteurs. De plus, l'enregistrement ne coûte que \$45 et ne consiste qu'en un formulaire simple.¹⁰¹

Section 2 : Des exceptions affectant différemment la protection selon le système.

Tout régime dispose d'exceptions, et le droit d'auteur dispose d'exceptions presque universelles : celle de la non protection des idées et celle du droit d'étudier l'œuvre. Le but est de ne pas permettre à l'auteur de jouir d'un monopole exclusif sur son œuvre. Il convient d'examiner comme ces exceptions sont appliquées aux logiciels dans l'UE (§ 1) et aux Etats-Unis (§ 2).

Paragraphe 1 : Des limites affectant sensiblement la protection européenne.

Il s'agit ici de relever comment les exceptions d'absence de protection des idées (A) et du droit d'analyse et interopérabilité (B) affecte la protection des programmes informatiques dans l'UE.

A) L'éternelle problématique de l'idée et de l'expression en droit de l'Union Européenne.

Interdire de s'inspirer du travail des autres serait dommageable pour l'innovation et le progrès dans n'importe-quelle discipline. Or, c'est bien l'incitation à innover que le droit de la propriété intellectuelle recherche. Il faut alors distinguer ce qui est acceptable de ce qui ne l'est pas dans l'utilisation des idées des autres. Mais une règle du droit d'auteur consiste à dire que seule l'expression est protégée, et pas l'idée. De ce fait, plusieurs auteurs peuvent avoir la même idée exprimée sous une forme différente.

Cette situation est très problématique concernant les programmes informatiques puisque souvent, donner une fonction précise à un programme revient à restreindre la façon dont le programme va être écrit, ce qui peut amener deux programmeurs à écrire indépendamment un code similaire pour une même utilité/idée. Malgré cela, il est disposé à l'article 1(2) de la Directive 2009/24/CE que cette protection s'arrête à l'expression de l'œuvre et ne s'étend pas

¹⁰¹ <https://www.authorsalliance.org/2018/02/14/copyright-registration-part-1why-register-your-copyright/> ; Oliver Herzfeld, Five Reasons To Copyright Register Your Software Now, Forbes Magazine, 26/10/2015, accessible via le lien suivant : <https://www.forbes.com/sites/oliverherzfeld/2015/10/26/five-reasons-to-copyright-your-software-now/#3e0027d34dd6>

à l'idée.¹⁰² Il fut conclu que l'article 1(2) n'est pas raisonnablement clair, menant les juges à poser une question préjudicielle à la CJUE concernant l'interprétation de cet article 1(2), soulevée par l'arrêt SAS.¹⁰³ La question est souvent liée à la situation où un programme concurrent reproduit les fonctionnalités d'un premier programme sans avoir forcément accès à son code. La CJUE reconnut que les fonctionnalités, le langage de programmation et les formats des fichiers de données n'étaient pas des expressions protégeables par le droit d'auteur. Le code source et objet le sont. Mais elle suggéra que les langages de programmation et fichiers de données pouvaient l'être sous la Directive 2001/29/CE (EUCD)¹⁰⁴ si ils constituent des créations intellectuelles propres à l'auteur, comme il a été conclu pour les GUI¹⁰⁵ On peut donc en déduire que seules les fonctionnalités ne peuvent bénéficier de protection par le droit d'auteur, figurant comme un élément non littéraire assimilé à l'idée du logiciel.¹⁰⁶

Ces difficultés à distinguer l'idée de l'expression a soulevé beaucoup de critiques. Cornish et Llewelyn ont pu décrire la situation comme « une distinction avec une frontière mal définie. »¹⁰⁷ Malgré tout, la distinction est internationalement reconnue dans l'ADPIC à l'article 9(2) et dans la Directive 2009/24/CE à l'article 1(2), même si aucun n'explique comment appliquer cette distinction. Cependant, cette distinction est cruciale puisqu'elle permet de définir si un travail remplit les conditions du droit d'auteur, ou encore si une atteinte a été faite sur une partie protégée de l'œuvre.

La conséquence de cette distinction appliquée aux logiciels est que le droit d'auteur européen ne protège pas la fonctionnalité du programme (l'idée), même si celle-ci est la plus importante, mais protège les efforts du développeur à mettre cette idée en place par le biais de son code source et matériaux annexes. De ce fait, comme il y a plusieurs façons d'exprimer

¹⁰² « La protection prévue par la présente directive s'applique à toute forme d'expression d'un programme d'ordinateur. Les idées et principes qui sont à la base de quelque élément que ce soit d'un programme d'ordinateur, y compris ceux qui sont à la base de ses interfaces, ne sont pas protégés par le droit d'auteur en vertu de la présente directive. »

¹⁰³ Cour de Justice de l'Union Européenne, Grande Chambre du 2 mai 2012, SAS Intitute Inc. c/ World Programming Ltd., affaire C-406/10, RLDI 2012/82, n° 2741. Dans cet arrêt, SAS avait développé des applications pour des analyses de statistiques en langage SAS. Mais World Programming créa des programmes alternatifs avec une fonctionnalité similaire sans utiliser le même code. SAS invoqua alors que WP avait indirectement enfreint ses droits d'auteurs.

¹⁰⁴ Directive 2001/29/CE du Parlement européen et du Conseil du 22 mai 2001 sur l'harmonisation de certains aspects du droit d'auteur et des droits voisins dans la société de l'information, Journal officiel n° L 167 du 22/06/2001 p. 0010 - 0019

¹⁰⁵ Article 2(a) de la Directive 2001/29/CE, selon Cour de Justice de l'Union Européenne, quatrième chambre du 16 juillet 2009, affaire C-5/08, Infopaq International A/S v Danske Dagblades Forening, 16 Juillet 2009, Recueil de jurisprudence 2009 I-06569, points 37-39, 47-48, 51 ; La Cour de cassation a également retenu dans un arrêt en date du 17 octobre 2012 que le critère de protection dégagé par la CJUE dans l'arrêt Infopaq s'applique aux logiciels.

¹⁰⁶ Ibid para 40

¹⁰⁷ Cornish W R et Llewelyn D, *Intellectual property : patents, copyright, trademarks and allied rights*, page 455, 6ème édition, London: Sweet & Maxwell, Royaume-Uni, 2007

une fonctionnalité, elles ne sont pas bloquées par le premier développeur. C'est une limitation cruciale de la protection afin de préserver la création.

B) Des exceptions pesantes : droit d'analyse et interopérabilité.

La Directive octroie des droits habituels aux tiers liés au régime du droit d'auteurs, ce qui fait que l'utilisation ne portant pas atteinte aux droits peut être possible avec autorisation de l'auteur et dans les limites des exceptions d'utilisations.

La Directive 2009/24/CE introduit des droits d'utilisation raisonnable avec ou sans autorisation, contenus dans l'article 5, incluant la correction, la copie de sauvegarde, et pouvoir « *observer, étudier ou tester le fonctionnement de ce programme afin de déterminer les idées et les principes* » du programme. Dans ce cas, aucune atteinte au droit d'auteur n'est caractérisée dans la mesure où ils « *ne sauraient porter atteinte à l'exploitation normale du logiciel ou causer un préjudice injustifié aux intérêts légitimes de l'auteur* ». ¹⁰⁸ Ces exceptions ne peuvent pas être exclues par un contrat. ¹⁰⁹

Ensuite, afin de permettre l'interopérabilité ¹¹⁰ entre les programmes informatiques, il fallait autoriser la décompilation ou l'ingénierie inverse pour obtenir le code source. ¹¹¹ Beaucoup d'entreprises ont un intérêt à créer un programme compatible avec le plus de logiciels et machines possible, et ont donc un grand attrait pour la décompilation mais ont peur d'enfreindre des droits. D'autres entreprises produisant des systèmes complets ont moins intérêt à voir la décompilation autorisée. Délimiter les circonstances où la décompilation serait légitime était l'une des raisons pour l'adoption de la Directive 2009/24/EC. La section 6(1) dispose que la décompilation est permise quand elle est indispensable afin d'obtenir l'interopérabilité de programmes indépendamment créés, et l'article 6(2) prévoit que les informations obtenues ne peuvent être utilisées pour un autre but que l'interopérabilité, et ne peuvent être utilisées pour créer « *un programme d'ordinateur dont l'expression est fondamentalement similaire ou pour tout autre acte portant atteinte au droit d'auteur.* » Cette Directive cherche donc un équilibre entre les différents intérêts et besoins, autorisant la recherche dans le but de rendre compatible deux produits, mais pas pour autoriser une

¹⁰⁸ Cour de Justice de l'Union Européenne, Grande Chambre du 2 mai 2012, SAS Intitute Inc. c/ World Programming Ltd., affaire C-406/10, RLDI 2012/82, n° 2741

¹⁰⁹ Article 8 de la Directive 2009/24/CE.

¹¹⁰ Il s'agit de « la capacité d'échanger des informations et d'utiliser mutuellement les informations échangées (Directive 2009/24/CE, considérant 10), c'est-à-dire rendre des programmes compatibles en incluant les mêmes caractéristiques, afin de permettre leur interaction de façon cohérente

¹¹¹ La décompilation consiste à rétablir le code source à partir du code objet, il s'agit d'une forme d'ingénierie inverse propre au domaine des programmes informatiques.

compétition directe avec le produit original.¹¹² Cela a été affirmée dans l'arrêt SAS.¹¹³ Comme la décompilation serait autorisée afin de créer un programme compatible avec l'original mais sans être substantiellement similaire à l'original, des doutes subsistent à savoir si créer un programme concurrent pas substantiellement similaire peut être autorisé.

Par conséquent, le droit d'auteur européen ne semble pas clair concernant la distinction entre idée et expression, présentant tout au plus une liste de ce qui est considéré comme des idées ou des expressions. La Directive semble un peu plus tranchée au niveau des exceptions, où il est possible de constater une limitation aux droits mais aussi une protection assurée contre la reproduction suite à une décompilation.

Paragraphe 2 : Des limites à l'étendue de la protection par le droit d'auteur étasunien.

Les droits exclusifs octroyés par le régime du droit d'auteur ont des limites disposées par l'US Copyright Act et des décisions judiciaires. Il s'agit également de la non protection des idées (A) et de la *fair use defence*, soit l'exception d'utilisation (B).

A) L'éternelle distinction de l'idée et l'expression en droit d'auteur étasunien.

Cette distinction est apparue dans l'arrêt *Baker v Selden*,¹¹⁴ puis repris dans l'US Copyright Act of 1976 à la section 102(b) de manière plus détaillée : les idées, procédés, systèmes et méthodes d'opération ne sont pas protégées.¹¹⁵

Les tribunaux étasuniens sont dans la même difficulté qu'en UE comme évoqué par le juge Hand J dans l'arrêt *Nichols*,¹¹⁶ qui conclut que personne n'a pu trouver de frontière et personne ne pourra jamais en trouver.¹¹⁷ Dans cet arrêt, il fut établi que si un élément non-littéraire ou structurel de l'œuvre était concret, il pouvait bénéficier de la protection du droit d'auteur, comme par exemple les algorithmes dans les programmes.¹¹⁸

Cette complexe distinction a donné naissance à la méthode dite de « fusion entre l'idée et l'expression » (*merger of idea and expression*) qui exclut de la protection une expression si elle est indissociable de l'idée exprimée (en ce qu'elle est nécessairement dictée par l'idée).

¹¹² Krockner, ER, « The Computer Directive and balance of rights », 1997, European Intellectual Property Review, 247.

¹¹³ Note en bas de page 82, para 60 et seq.

¹¹⁴ Cour Suprême des Etats-Unis du 19 Janvier 1879, Baker v Selden, 101 US 99.

¹¹⁵ "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work."

¹¹⁶ Cour d'appel des États-Unis pour le deuxième circuit du 10 Novembre 1930, Nichols v. Universal Pictures Corp., 45 F.2d 119.

¹¹⁷ Cour d'appel des États-Unis pour le deuxième circuit du 10 Novembre 1930, Nichols v. Universal Pictures Corp., 45 F.2d 119.

¹¹⁸ Méthode dite de l'abstraction : les éléments abstraits ne sont pas protégés et relèvent des idées.

Cela permet de ne pas octroyer un monopole sur une idée via la protection de son unique possible expression. Cette doctrine est particulièrement invoquée dans le domaine des logiciels. Mais ce débat se confond souvent avec celui de l'originalité. Il semblerait alors que les éléments non-littéraires, comme la structure, les GUI,¹¹⁹ ou les fonctionnalités ne seraient pas protégés.

Après une série de décisions allant de l'attribution d'une presque totale protection de l'architecture du logiciel (*Whelan*¹²⁰) à une limitation grâce à un test plus précis (*Altai*¹²¹), une approche plus libérale fut adoptée par l'arrêt *Lotus*.¹²² En revanche, la décision *Oracle*¹²³ de 2014 rejette explicitement l'approche de l'arrêt *Lotus*. C'est en démontrant qu'il y a différentes possibilités d'expression des principes à la base de ses interfaces qu'Oracle a pu écarter l'application de la doctrine du *merger*. La cour conclut dans cet arrêt que la structure, les séquences et l'organisation (éléments non-littéraires) de ces interfaces de programmation (API) étaient protégés, tout comme le code (élément littéraire), car ces éléments étaient originaux puisqu'il n'y avait pas de choix d'expression limité ou dicté par la fonction.¹²⁴ Cet arrêt rejette les précédents arrêts en admettant que la structure, les séquences et l'organisation du programme peuvent comprendre l'expression de l'idée et ne sont pas seulement des idées, même s'ils dirigent le programme et qu'ils sont une méthode d'opération.¹²⁵ Même si cet arrêt concerne les API, il est applicable à d'autres éléments d'un logiciel comme les GUI. La conséquence de cette décision est qu'elle ouvre le spectre de protection octroyée par le droit

¹¹⁹ Partie du logiciel n'incluant pas le code ou sa structure, mais plutôt la partie relative à son apparence et manière d'opérer.

¹²⁰ Cour d'Appel des Etats-Unis pour le troisième circuit, 4 Août 1986, *Whelan Associates, Inc. v. Jaslow Dental Laboratory Inc.*, 797 F.2d 1222, 230 USPQ 481 : s'il y a des façons limitées pour exprimer la fonction alors cela serait interprété comme faisant partie de l'idée. Dans cet arrêt, le but du programme était l'organisation des archives d'un établissement dentaire. La structure et le GUI du programme n'étaient pas nécessaires au but, pouvant obtenir une protection.

¹²¹ Cour d'Appel des Etats-Unis pour le deuxième circuit du 22 Juin 1992, *Computer Associates v. Altai*, 982 F.2d 693 : formulation du test « abstraction-filtration-comparaison (AFC) » pour déterminer l'étendue de la protection des éléments non-littéraires d'un programme. Ce nouveau test impose de décomposer le travail, séparant les éléments n'étant pas protégés par le copyright et comparer ce qui reste avec le programme portant atteinte en les hiérarchisant, filtrant les éléments commandés par l'efficacité, les facteurs externes et appartenant au domaine public. Ce qui reste est comparé avec le logiciel du défendeur.

¹²² Cour d'appel des États-Unis pour le premier circuit, 5 Mars 1995, *Lotus Development Corp. V. Borland International, Inc.*, 49 F.3d 807, 821, rejetant le test AFC de l'arrêt *Whelan*. Cet arrêt n'attribue pas de protection pour des menus déroulants (GUI), car il s'agissait d'une fonctionnalité/méthode d'opération qui était devenu un standard, se classant parmi les idées, procédures, procédés, systèmes, méthodes d'opération de la section 102(b).

¹²³ Cour d'appel des États-Unis pour le circuit fédéral, 9 Mai 2014, *Oracle America v. Google Inc.*, 750 F.3d 1339 : Elle concerne les interfaces de programmation (API) du langage « Java », développé par Oracle, qui sont des éléments logiciels organisés sous forme hiérarchique et structurée, destinés à être intégrés dans des programmes opérant sous l'environnement Java et correspondant à certaines fonctionnalités. En utilisant ces API, cela évite de réécrire les parties de programme correspondant à ces fonctionnalités.

¹²⁴ Arrêt Oracle, para 1358, 1359-1360, 1363-1364

¹²⁵ Ibid para 1366

d'auteur. Le système des Etats-Unis semble s'étendre au gré des décisions au profit d'une propriété presque exclusive pour l'auteur sur les éléments littéraires et non-littéraires.

Les programmes informatiques aux Etats-Unis sont donc dans une situation assez instable, où pouvant bénéficier du droit des brevets et du droit d'auteur, et pouvant trouver une plus grande protection au sein du régime de ce dernier. Certains ont pu dire que la décision dans l'arrêt Oracle ne suivait pas la lettre de la Section 102(b), mieux exprimée par l'arrêt Lotus.¹²⁶ Déjà dans l'arrêt *Altai*, les juges appelaient à un « CONTU II » afin de pouvoir clarifier la situation et de mieux voir comment intégrer les programmes informatiques au sein du régime du droit d'auteur. Il semblerait que les dernières avancées permettraient au créateur d'avoir un monopole grâce au droit d'auteur sur certaines parties non-littéraires du logiciel.

Ainsi, la distinction de l'idée et expression semble être ouverte à plus de discussions aux Etats-Unis, ayant donné lieu à une doctrine. Elle semble pour l'instant servir les programmeurs, contrairement au semblant de liste établit par la CJUE qui refuse catégoriquement la protection des fonctionnalités.

B) Une exception pesante : la *fair use* doctrine.

La Cour Suprême des Etats-Unis a pu constater que l'ingénierie inverse est « *une partie essentielle de l'innovation* », qui « *peut mener à des avancées importantes pour la technologie* », et que « *la réalité concurrentielle de l'ingénierie inverse peut agir comme stimulant l'innovation* ». ¹²⁷ De ce fait, la décompilation et ingénierie inverse peuvent être excusées grâce à la *fair use doctrine* contenue à la section 107, si elles sont conduites à des fins éducatives ou de recherche. ¹²⁸

Concernant la décompilation et l'utilisation légitime, les Etats Unis ont reconnu dans l'arrêt *Atari*¹²⁹ que l'ingénierie inverse pour discerner les idées non protégeables d'un programme était une utilisation équitable (*fair use*). Il en est de même pour l'arrêt *Sega*, suggérant que la

¹²⁶ Noam Shemtov, *Beyond The Code : Protection of Non-Textual Features of Software*, 2017 Oxford University Press, p 133

¹²⁷ Cour Suprême des Etats-Unis du 21 Février 1989, *Bonito Boats, Inc v Thinder Craft Boats, Inc*, 141 US 489.

¹²⁸ 17 U.S.C. § 107 : Notwithstanding the provisions of sections 106 and 106A, the fair use of a copyrighted work, including such use by reproduction in copies or phonorecords or by any other means specified by that section, for purposes such as criticism, comment, news reporting, teaching (including multiple copies for classroom use), scholarship, or research, is not an infringement of copyright. In determining whether the use made of a work in any particular case is a fair use the factors to be considered shall include the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes ; the nature of the copyrighted work; the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and the effect of the use upon the potential market for or value of the copyrighted work.

¹²⁹ Cour d'Appel des Etats-Unis pour le circuit fédéral du 10 Septembre 1992, *Atari Games Corp. v. Nintendo of America Inc.*, 975 F.2d 832, 24 USPQ2d 1015.

décompilation pouvait être permise dans certaines circonstances, notamment lorsque le produit final ne contient pas d'éléments protégés ou quand le copiage est nécessaire pour accéder à la fonctionnalité du programme.¹³⁰ Dans l'arrêt *Sony*,¹³¹ les juges ont pu trouver que comme l'utilisation du code était transformatrice, le défendeur avait réalisé un apport créatif. Ces décisions sont portées par le principe sous-jacent du droit d'auteur et de la *fair use doctrine* qui est de stimuler la création artistique par l'accès aux idées et éléments fonctionnels du programme (non-protégés). Ces arrêts ont également conclu que les programmes d'ordinateur bénéficiaient d'une protection moins forte que les autres œuvres littéraires. Cela peut apparaître décevant pour les développeurs propriétaires, mais un récent rapport établit que ces restrictions sont nécessaires pour les consommateurs, techniciens, réparateurs et autres intéressés.¹³²

De plus, la nouvelle section 1201 de l'USCA 1976 interdit de contourner les systèmes de protection de droit d'auteur.¹³³ La section 1201(f)¹³⁴ semble suivre la jurisprudence concernant l'ingénierie inverse, mais il est toujours possible d'interdire la décompilation et ingénierie inverse par des provisions contractuelles, comme souligné dans l'arrêt *Davidson*,¹³⁵ rendant facilement contournable l'exception de la *fair use doctrine*, contrairement à l'exception contenue dans le droit européen.

Dès lors, l'exception du *fair use* permet aux tiers de réellement s'approprier un programme tout en faisant attention à ne pas reprendre des éléments protégés et en se distinguant suffisamment. Cela rejoint ce qui se fait dans l'Union Européenne. En revanche, cette possibilité de restreindre le jeu de cette exception par des clauses contractuelles est interdite en droit de l'UE, éloignant les deux systèmes.

¹³⁰ Cour d'Appel des Etats-Unis pour le neuvième circuit du 20 Octobre 1992, *Sega Enterprises Ltd. v. Accolade Inc.*, 977 F.2d 1510, 24 USPQ2d 1561 ; dans cet arrêt, le défendeur avait décompilé le logiciel de Sega afin de trouver le code d'activation permettant de débloquent le dispositif permettant d'empêcher des cartouches de jeux vidéo non autorisées d'être utilisées sur la console de Sega.

¹³¹ Cour d'Appel des Etats Unis pour le neuvième circuit du 20 Février 2000, *Sony Computer Entertainment, Inc v Connectix Corp*, 203 F.3d 596 ; dans cet arrêt, le défendeur, Connectix, cherchait à obtenir accès au code de Sony PlayStation Games afin de pouvoir imiter ses fonctions.

¹³² United State Copyright Office, *Software-Enabled Consumer Products : a report of the register of copyrights*, December 2016, p 39 ; accessible via le lien suivant : <https://www.copyright.gov/policy/software/software-full-report.pdf>

¹³³ "No person shall circumvent a technological measure that effectively controls access to a work protected under this title."

¹³⁴ "Notwithstanding the provisions of subsection (a)(1)(A), a person who has lawfully obtained the right to use a copy of a computer program may circumvent a technological measure that effectively controls access to a particular portion of that program for the sole purpose of identifying and analyzing those elements of the program that are necessary to achieve interoperability of an independently created computer program with other programs, and that have not previously been readily available to the person engaging in the circumvention, to the extent any such acts of identification and analysis do not constitute infringement under this title."

¹³⁵ Cour d'Appel des Etats Unis pour le huitième circuit du 1er Septembre 2005, *Davidson & Associates v. Jung*, 422 F.3d 630

Pour conclure concernant les droits d'auteur, il peut être commenté que la protection assurée n'est pas également suffisante pour les développeurs dans les deux systèmes. L'un fait disparaître du spectre de protection les fonctionnalités, principal atout des logiciels, et l'autre les inclus en suivant une construction jurisprudentielle complexe. De plus, le droit d'auteur étasunien et européen permet aux tiers de facilement reproduire les mêmes fonctionnalités, tout en produisant un code original puisqu'il y a une multitude de manières de coder. Là encore, les Etats-Unis autorisent une protection plus forte grâce au mécanisme de droit des contrats. En revanche, ces régimes sont encore instables, et il est possible que d'autres décisions jurisprudentielles viennent s'y greffer afin de les préciser.

Le droit des brevets et du droit d'auteur ont donc été étendus afin d'accueillir les programmes informatiques. Cela était nécessaire, les instances internationales n'ayant pas trouvé de consensus sur un régime propre. Cependant, les pans de la propriété intellectuelle ne sont pas sans inconvénients : parfois trop faibles, ils ne permettent pas une protection optimale au propriétaire ; parfois trop complexes, le logiciel échappe à leur protection ; parfois trop forts, ils permettent de créer un monopole dommageable pour l'innovation et la création. Ces critiques s'appliquent bien sûr aux deux régimes étasunien et européen. De ce fait, des alternatives ont pu être utilisées ou même créées au fil des années afin de pallier ces écueils propres aux protections de principe.

PARTIE 2 : Les modes alternatifs de protection des programmes d'ordinateur dans les législations étasuniennes et européennes.

Les modèles décrits dans la première partie ne restent pas sans critiques, offrant parfois une protection exclusive pour les auteurs, mais les laissant aussi sans moyens de faire valoir leurs droits à d'autres moments. Il est alors opportun d'étudier les options additionnelles de protection aux Etats-Unis et Union Européenne (Chapitre 1), avant de se s'interroger sur le dilemme entre absence de protection et protection *sui generis* de façon générale (Chapitre 2).

Chapitre 1 : L'utilisation différenciée des modes alternatifs de protection des programmes d'ordinateur.

Tout en disposant d'une protection de base offerte par le brevet ou le droit d'auteur, une protection supplémentaire peut être trouvée sous certains régimes juridiques (Section 1), mais aussi des outils juridiques (Section 2) qui viennent compléter le régime principal. L'utilisation de ces compléments est plus ou moins appuyée suivant le système juridique.

Section 1 : Des régimes juridiques venant variablement en aide selon le territoire de protection.

Dans le cas où le régime de base de protection n'est plus suffisant pour les auteurs de logiciels, il est possible de se tourner vers d'autres droits de propriété intellectuelle (§1), en plus du régime des secrets d'affaires (§2).

Paragraphe 1 : Un autre pan de la propriété industrielle européenne et américaine au secours des développeurs.

Il conviendra d'examiner l'application au logiciel du droit des marques et des dessins et modèles (A), pour ensuite observer l'application du principe de concurrence déloyale (B).

A) Droit des marques et des dessins et modèles appliqués aux logiciels.

Nous avons vu que le code, les matériaux préparatoires, et plus rarement les fonctionnalités et méthodes sont protégés par le droit d'auteur. Mais qu'en est-il de la GUI, de loin l'élément le plus important aux yeux du consommateur ?¹³⁶ Nous avons constaté précédemment que le droit d'auteur pouvait servir à protéger les GUI, mais n'est pas le régime le plus efficace. Le droit des marques (1) et le droit des dessins et modèles (2) viennent remédier à ce doute.

¹³⁶ Par exemple, l'utilisation de nos smartphone est entièrement faite via l'écran, où les couleurs, la mise en page, la typographie (le « look ») et les boutons, menus interactifs (le « feel »), sont les éléments déterminants de la popularité commerciale de celui-ci.

1) Le droit des marques.

Les Etats-Unis permettent la protection des GUI par le *trade-dress*¹³⁷ si le demandeur prouve en cas de litige que le *trade-dress* est distinctif, non-fonctionnel, que sa GUI est suffisamment bien connue des consommateurs et qu'il y a un risque de confusion avec le *trade-dress* du défendeur, comme dans l'affaire *Tetris*.¹³⁸ Il y a deux catégories de *trade-dress* : le packaging du produit et le design du produit lui-même ; les GUI seraient le design du logiciel en lui-même. Si la GUI n'a pas encore acquis assez de notoriété, le droit d'auteur et des dessins et modèles peuvent offrir une protection.¹³⁹ Cependant, le demandeur peut éprouver des difficultés à démontrer que sa GUI n'est pas dictée par la fonction car certaines configurations sont très simples. Mais certaines GUI ont prouvé que ce n'était pas tout le temps vrai.¹⁴⁰

Pour ce qui est de l'Union Européenne, la GUI est assimilée à une marque tridimensionnelle, bénéficiant du régime du droit des marques. Les mêmes conditions doivent être démontrées lors d'un litige. Mais contrairement aux Etats-Unis, il n'y a pas de distinction entre packaging et design du produit, et une GUI peut être intrinsèquement distinctive si elle s'éloigne de ce qui est la norme dans son secteur.¹⁴¹ Ensuite, depuis la Directive 2015/24/36,¹⁴² le caractère fonctionnel ne s'examine plus seulement par rapport à la forme tridimensionnelle du produit mais par rapport à toute autre caractéristique, laissant supposer qu'il sera difficile pour des GUI d'obtenir une protection contrairement aux Etats-Unis, d'autant que la pluralité de GUI possibles pour une fonction n'est pas indicative de non-fonctionnalité.¹⁴³

En somme, les conditions ne semblent pas grandement différer, mais il apparaît plus facile d'obtenir une protection aux Etats-Unis en droit des marques. Cette alternative permet de protéger l'aspect visuel du programme, élément contribuant énormément au succès commercial.

¹³⁷ Le *trade-dress* est l'image du produit et son apparence globale, sa taille, couleur, texture, graphiques, particularité de technique de vente : Cour Suprême des Etats-Unis du 26 Juin 1992, *Two Pesos, Inc, Taco Cabana, Inc.* 505 US 763, 764.

¹³⁸ Cour de district des États-Unis du 30 Mai 2012, *Tetris Holding, LLC v Xio Interactive, Inc* 863 F.Supp.2d 394 (D.N.J. 2012)

¹³⁹ Ibid 414

¹⁴⁰ Voir les enregistrements n° 3495193 pour un système d'interface consommateur et 77981717 pour une GUI animée.

¹⁴¹ Cour de Justice de l'Union Européenne en sa première chambre du 22 juin 2006, affaire C-24/05 P, *Storck v OHIM*, Recueil du jurisprudence 2006 I-05677, para 24 et seq.

¹⁴² Directive (UE) 2015/2436 du Parlement Européen et du Conseil du 16 décembre 2015 rapprochant les législations des États membres sur les marques (Texte présentant de l'intérêt pour l'EEE), JO L 336, p. 1–26, article 4(1)(e)(ii)

¹⁴³ Cour de Justice de l'Union Européenne du 18 Juin 2002, affaire 299/99, *Koninklijke Philips Electronics NV contre Remington Consumer Products Ltd.*, 2002 I-05475

2) Le droit des dessins et modèles.

Enfin, il est possible de faire bénéficier aux GUI du droit des modèles pour leurs aspects esthétiques, via le *design patent*¹⁴⁴ pour les Etats-Unis et les *dessins et modèles* pour l'UE, afin de protéger l'apparence du logiciel.

Aux US, les *design patents* sont largement octroyés pour les GUI sous la Section 171 du US Patent Act.¹⁴⁵ Les conditions sont que le *design* doit être nouveau, non-évident et ornemental (ce dernier critère n'étant pas insurmontable pour les GUI, étant plus étroit que pour les *trade-dress*).¹⁴⁶ En revanche, les GUI ont dû être incluses dans les produits pouvant bénéficier de cette protection, contrairement à l'Union Européenne qui les protège généralement malgré l'exclusion des programmes d'ordinateur sous l'article 3(b) du Règlement (CE) 6/2002.¹⁴⁷ Les conditions ne sont pas si différentes : le modèle doit être nouveau, individuel, et ne doit pas être dicté par sa fonction technique.¹⁴⁸ Le dernier critère pourrait poser problème, d'autant que trois tests sont utilisés par les juridictions européennes, mais il semblerait que l'approche tend vers celle des Etats-Unis.¹⁴⁹

Finalement, des divergences ne sont pas flagrantes dans ce domaine. Tout bien considéré, ce qui est intéressant grâce à ces alternatives est qu'un programmeur peut combiner plusieurs modes de protection. Le meilleur exemple est le cas de Google : Google possède un *utility patent* (brevet)¹⁵⁰, un *design patent* (dessin et modèle)¹⁵¹, et une marque¹⁵² pour son moteur de recherche et affichage de résultats.

B) La concurrence déloyale appliquée aux logiciels.

La concurrence déloyale sert à combattre les pratiques malhonnêtes et frauduleuses des affaires de façon à imiter ou usurper ses concurrents, causant un préjudice. Elle s'applique totalement au domaine des logiciels, qui ne manque pas de ce genre de situation.

¹⁴⁴ Voir l'exemple du design patent USD599372 où Google protège ça page d'accueil de moteur de recherché tout en écrivant "Google" en pointillés afin de pouvoir agir si un tiers reproduit la même structure avec un autre nom.

¹⁴⁵ 35 U.S.C. §171(a) : Whoever invents any new, original and ornamental design for an article of manufacture may obtain a patent therefor, subject to the conditions and requirements of this title.

¹⁴⁶ R Stigler, 'Ooey GUI : the messy protection of graphical user interfaces' (2015) 12(3) Northwestern Journal of Technology and Intellectual Property, 240

¹⁴⁷ Noam Shemtov p 211 ; Règlement (CE) n° 6/2002 du Conseil du 12 décembre 2001 sur les dessins ou modèles communautaires, JO L 3, p. 1-24

¹⁴⁸ Article 4 et 8 du Règlement n° 6/2002

¹⁴⁹ Noam Shemtov p 213

¹⁵⁰ US Patent No. 7,146,358

¹⁵¹ US Patent. No. D533,561

¹⁵² US Trademark No 2,806,075

La concurrence déloyale est une branche du droit de la propriété intellectuelle aux US, précisément en droit des marques. Les Etats-Unis contrôlent les violations concernant les logiciels en s'appuyant sur ce principe avec le Lanham Act¹⁵³ en sa section 43(a)(1)(A) pour le risque de confusion¹⁵⁴ et section 43(c) pour le risque de dilution.¹⁵⁵ Par exemple, dans l'arrêt *Baby Be Mine*,¹⁵⁶ la Cour fédérale a jugé qu'il y avait un risque de confusion entre le *look and feel* du défendeur comparé au site web Baby Be Mine, et que plusieurs similarités entre ces sites avaient été démontrées par des captures d'écran. Il est notable que, aux Etats-Unis, le principe de concurrence déloyale ne peut s'invoquer qu'en droit des marques, supposant que le développeur dispose d'un trade-dress, donc d'un droit privatif.

En droit de l'UE, la concurrence déloyale est un principe qui est laissé aux Etats-Membres, en raison des trop grandes disparités.¹⁵⁷ En effet, deux modèles coexistent : les systèmes de la *Common Law* et civiliste. Les pays suivant le système de la *Common Law* n'ont pas un droit de la concurrence déloyale au sens large : le principe du *passing off* protège des *misrepresentation* trompant le consommateur.¹⁵⁸ En revanche, ce concept est plus large dans les systèmes civilistes en ce qu'ils permettent d'empêcher le détournement de réputation, la distorsion, la tromperie et pratiques déloyales, lorsque le titulaire n'a pas de droit privatif ou souhaite une protection supplémentaire. En France, sont ainsi appréhendés les actes de concurrence déloyale dans le cas où un tiers profite du succès commercial du logiciel du demandeur, à défaut de prouver son droit de propriété intellectuelle, quand les deux logiciels sont similaires, pouvant mener à la confusion.¹⁵⁹ De plus, une autre notion, le parasitisme, est utilisée en cas d'usurpation de notoriété, de signes de ralliement de la clientèle ou

¹⁵³ The Lanham (Trademark) Act, Pub.L. 79-489, 60 Stat. 427, enacted July 5, 1946, codified at 15 U.S.C. § 1051 et seq. (15 U.S.C. ch. 22).

¹⁵⁴ Soit 15 U.S. Code § 1125 : Any person who, on or in connection with any goods or services, or any container for goods, uses in commerce any word, term, name, symbol, or device, or any combination thereof, or any false designation of origin, false or misleading description of fact, or false or misleading representation of fact, which is likely to cause confusion, or to cause mistake, or to deceive as to the affiliation, connection, or association of such person with another person, or as to the origin, sponsorship, or approval of his or her goods, services, or commercial activities by another person.

¹⁵⁵ "Subject to the principles of equity, the owner of a famous mark that is distinctive, inherently or through acquired distinctiveness, shall be entitled to an injunction against another person who, at any time after the owner's mark has become famous, commences use of a mark or trade name in commerce that is likely to cause dilution by blurring or dilution by tarnishment of the famous mark, regardless of the presence or absence of actual or likely confusion, of competition, or of actual economic injury."

¹⁵⁶ Cour du District des Etats Unis, District de la Californie du Nord du 1^{er} Avril 2014, *Ingrid & Isabel, LLC v Baby Be Mine, LLC* F. Supp.3d, 2014 WL 4954656 (N.D. Cal. 2014)

¹⁵⁷ Considèrent 40 de la Directive 2015/2436

¹⁵⁸ Voir par exemple l'arrêt de la Chambre des Lords du Royaume-Uni, *Reckitt & Colman Products Ltd v Borden, Inc. And Others*, 1990, 1 All ER 873.

¹⁵⁹ Cour de cassation, Première Chambre Civile, Arrêt du 14 novembre 2013, N° de pourvoi: 12-20687 (protection par la concurrence déloyale lorsque le demandeur n'avait pas réussi à prouver qu'il détenait un droit d'auteur, son logiciel ne satisfaisant pas le critère d'originalité)

d'investissements sans nécessairement qu'il y ait tromperie ou confusion.¹⁶⁰ Toutefois, comme les pays membres de l'UE proviennent de systèmes juridiques différents, le droit de la concurrence déloyale n'est pas harmonisé comme aux Etats-Unis, faisant qu'une action en justice sera envisageable dans certains pays mais impossible dans d'autres pour une même situation.¹⁶¹

Il est tout de même notable qu'en France, la protection est plus large qu'aux Etats-Unis puisqu'elle ne se limite pas au droit des marques puisqu'elle couvre les situations où le développeur n'a pas de droit privatif. Elle offre une réelle protection supplémentaire aux programmeurs contrairement aux principes limités des Etats-Unis.

Paragraphe 2 : Protection des logiciels par le régime des secrets d'affaires.

La plupart des droits de la propriété intellectuelle naissent dans le secret. Même après la période de conception, le secret d'affaires est utile en tant que protection supplémentaire pour les caractéristiques impossible à découvrir, comme les commentaires sur le code, mais aussi les caractéristiques pouvant l'être, comme les algorithmes et structures des données.¹⁶² Il convient d'observer le régime européen (A) puis étasunien (B).

A) Le droit européen des secrets d'affaires appliqué aux logiciels.

L'Union Européenne a récemment adopté la Directive(UE) 2016/943.¹⁶³ La protection actuelle varie grandement entre les Etats-Membres : protéger son logiciel par le secret d'affaires à travers l'UE est coûteux et compliqué. L'objectif de cette Directive est d'apporter une harmonisation minimum, laissant la flexibilité aux Etats-Membres de choisir les moyens.

L'article 2 pose la définition des secrets d'affaires en donnant trois conditions.¹⁶⁴ D'après le considérant 14, les informations technologiques sont couvertes par la directive : ainsi, tout processus de conception, architecture interne, code source d'un logiciel pourrait donc être compris. Or, un problème surviendrait dès l'examen de la première condition : l'information

¹⁶⁰ Fondé sur le nouvel article 1240 du Code Civil, par exemple : Cour d'appel d'Aix en Provence, 1ère chambre C, Arrêt du 10 octobre 2013 n° 12/22423 ;

¹⁶¹ Noam Shemtov 207

¹⁶² Noam Shemtov 220

¹⁶³ Directive (UE) 2016/943 du Parlement européen et du Conseil du 8 juin 2016 sur la protection des savoir-faire et des informations commerciales non divulgués (secrets d'affaires) contre l'obtention, l'utilisation et la divulgation illicites (Texte présentant de l'intérêt pour l'EEE), OJ L 157, 15.6.2016, p. 1–18, qui devra être implémentée avant la fin de cette année 2018.

¹⁶⁴ Les informations (i) sont secrètes en ce sens que, dans leur globalité ou dans la configuration et l'assemblage exacts de leurs éléments, elles ne sont pas généralement connues des personnes appartenant aux milieux qui s'occupent normalement du genre d'informations en question, ou ne leur sont pas aisément accessibles, (ii) ont une valeur commerciale parce qu'elles sont secrètes, (iii) ont fait l'objet, de la part de la personne qui en a le contrôle de façon licite, de dispositions raisonnables, compte tenu des circonstances, destinées à les garder secrètes.

ne doit pas être généralement connue des personnes appartenant au milieu s'occupant de ce genre d'information. Qu'en est-il de l'ingénierie inverse ? Comme ce procédé est complexe, coûteux, et hasardeux, il semblerait que l'information en lien avec un logiciel ne soit pas généralement connue.¹⁶⁵ L'article 3 prévoit que l'ingénierie inverse d'un objet à disposition du public ou acquis licitement, non interdite par un contrat, est un moyen licite d'acquisition d'information. Or, dans certains cas, l'ingénierie inverse peut devenir une activité illicite d'après l'article 4 de la Directive si l'information est obtenue illégalement, soit sans le consentement du détenteur du secret d'affaires et sans autorisation d'accès.

En conséquence, l'ingénierie inverse d'un logiciel semble être autorisée par la Directive, sauf quand le logiciel n'est pas mis à disposition du public. D'autant que, comme vu plus haut, l'ingénierie inverse ne peut être interdite dans les contrats.¹⁶⁶ Cela réduit considérablement les bases juridiques pour prononcer l'illicéité de l'acquisition du secret du logiciel. Cependant, il serait nécessaire de prévoir de telles restrictions pour certaines situations comme pour la mise à disposition d'un logiciel-tests à quelques consommateurs pendant la période de conception. Ainsi, il serait souhaitable que la Cour éclaircisse la relation entre ces deux directives, d'autant que le début de l'article 8 de la Directive 2009/24/CE dispose clairement que : « Les dispositions de la présente directive n'affectent pas les autres dispositions légales concernant notamment [...] le secret des affaires [...] ». Certains appellent à un système flexible prenant en compte les circonstances de l'espèce, par exemple en s'aidant en plus de la notion « d'usages honnêtes en matière commerciale » de l'article 4(2)(b).¹⁶⁷

B) Le droit étasunien des secrets d'affaires appliqué aux logiciels.

Pendant longtemps, la protection des secrets d'affaires se faisait au niveau des Etats fédérés. Mais la plupart avaient adopté une version plus ou moins identique à l'Uniform Trade Secrets Act (UTSA) de 1979.¹⁶⁸ Depuis le Defend Trade Secrets Act (DTSA) de 2016,¹⁶⁹ une protection supplémentaire des secrets d'affaires est rajoutée au niveau fédéral. Le DTSA ne préempte pas l'UTSA : les deux régimes fonctionnent en parallèle.¹⁷⁰ La différence est que le DTSA peut être invoqué lorsque le litige concerne un produit utilisé dans un contrat

¹⁶⁵ Noam Shemtov 227.

¹⁶⁶ Article 8 de la Directive 2009/24/CE.

¹⁶⁷ Noam Shemtov 232.

¹⁶⁸ National Conference of Commissioners on Uniform State Law, Uniform Trade Secrets Act (UTSA) of 1979, amended in 1985 : http://www.uniformlaws.org/shared/docs/trade%20secrets/utsa_final_85.pdf

¹⁶⁹ Defend Trade Secrets Act of 2016, Public Law 114-153 114th Congress (11 May 2016), accessible via le lien suivant : <https://www.congress.gov/114/plaws/publ153/PLAW-114publ153.pdf>

¹⁷⁰ Ibid section 2(f): Nothing in the amendments made by this section shall be construed to modify the rule of construction under section 1838 of title 18, United States Code, or to preempt any other provision of law.

interétatique ou à l'étranger.¹⁷¹ Pour prétendre à une protection, l'auteur doit avoir pris les mesures nécessaires afin que l'information soit secrète, et cette information doit avoir une valeur économique, actuelle ou potentielle, impliquant qu'elle ne doit pas être généralement connue ou facilement accessible.¹⁷² Ainsi, la définition ne s'éloigne pas vraiment de celle de l'Union Européenne, et le problème lié à l'ingénierie inverse qui rend le programme « facilement » accessible n'en est pas non plus un.¹⁷³

La divulgation du secret par des moyens illégitimes peut donner lieu à des poursuites pour fausse déclaration.¹⁷⁴ En revanche, il est clairement établi dans l'UTSA que la découverte par l'ingénierie inverse d'un produit obtenu légitimement est un moyen licite d'acquisition du secret.¹⁷⁵ L'idée que l'ingénierie inverse contribue aux avancées technologies constituant un moyen légitime de découverte du secret est confirmée par la Cour Suprême des Etats-Unis.¹⁷⁶

En revanche, et contrairement à l'UE, l'ingénierie inverse peut être interdite par des clauses contractuelles incorporées dans un contrat de licence, rendant la découverte des informations secrètes par le titulaire de licence illicite.¹⁷⁷ Comment concilier ces deux approches ? Les réponses varient selon les Etats fédérés. Dans l'arrêt *Telerate Systems v Caro*,¹⁷⁸ la District Court établissait que la provision contractuelle interdisant l'ingénierie inverse était suffisante pour rendre la découverte du secret illégitime, les circonstances de chaque cas devant être examinées.¹⁷⁹ Dès lors, une clause restrictive pourrait permettre de rendre l'ingénierie inverse illégitime d'après l'UTSA. En revanche, il a pu être décidé le contraire, avec l'idée que l'autorisation de procéder à l'ingénierie inverse est d'ordre public.¹⁸⁰ Egalement, il fut

¹⁷¹ Ibid section 2(b): An owner of a trade secret that is misappropriated may bring a civil action under this subsection if the trade secret is related to a product or service used in, or intended for use in, interstate or foreign commerce.

¹⁷² USTA, section 1(4) : "Trade secret" means information, including a formula, pattern, compilation, program, device, method, technique, or process, that derives independent economic value, actual or potential, from not being generally known to, and not being readily ascertainable by proper means by, other persons who can obtain economic value from its disclosure or use, and is the subject of efforts that are reasonable under the circumstances to maintain its secrecy."

¹⁷³ Voir l'arrêt de la Cour d'appel pour le circuit fédéral, 10eme circuit, *Gates Rubber Co. V Bando Chem Indus Ltd*, 9 F.3d 823, 848 (10th Cir. 1993)

¹⁷⁴ Section 1(2)(b) de l'UTSA : "Misappropriation" means: acquisition of a trade secret of another by a person who knows or has reason to know that the trade secret was acquired by improper means; or disclosure or use of a trade secret of another without express or implied consent by a person who used improper means to acquire knowledge of the trade secret; [...]

¹⁷⁵ Commentaires de la Section 1 de l'UTSA : Proper means include ... Discovery by "reverse engineering", that is, by starting with the known product and working backward to find the method by which it was developed. The acquisition of the known product must, of course, also be by a fair and honest means, such as purchase of the item on the open market for reverse engineering to be lawful.

¹⁷⁶ Cour Suprême des Etats-Unis du 13 Mai 1973, *Kawane Oil Co v Bicon Corp*, 416 US 470, 476 (1974) ; Cour Suprême des Etats-Unis, *Bonito Boats Inc v Thunder Craft Boats Inc*, 489 US 141, 160

¹⁷⁷ Section 1(1) de l'UTSA 1985

¹⁷⁸ Cour du District des Etats-Unis, District du Sud de New York du 10 Juin 1988, *Telerate Systems, Inc. v. Caro*, 689 F. Supp. 221 (S.D.N.Y. 1988)

¹⁷⁹ Ibid 233

¹⁸⁰ Cour du District du Centre de la Californie du 13 Février 2012, *Aqua Connect, Inc. v. Code Rebel, LLC.*, Case No CV 11-5764-RSWL

considéré que le régime du droit d'auteur fédéral préemptait le droit des secrets d'affaires étatique, autorisant l'ingénierie inverse.¹⁸¹ Or, il est avancé que dorénavant, les développeurs ne verraient pas leur demande préemptée par le droit d'auteur s'ils se basent sur le nouveau DTSA, qui est lui aussi de niveau fédéral.¹⁸²

En conclusion, il semble que l'UE ne s'oriente pas vers une propriété exclusive des informations protégées par le secret d'affaires,¹⁸³ tel que le font les Etats-Unis grâce au mécanisme des clauses contractuelles. Cette alternative serait ainsi plus forte aux Etats-Unis.

Section 2 : Des outils permettant de répondre aux difficultés : l'avantage étasunien ?

Il s'agit ici d'étudier des constructions juridiques ou techniques palliant l'insuffisance juridique quant à l'exploitation des logiciels, permettant d'ouvrir son logiciel au public de façon profitable et contrôlée (§1), ou de restreindre son accès (§2).

Paragraphe 1 : Les nouveaux outils juridiques et améliorations techniques assurant plus de sécurité.

Il conviendra d'examiner les *patent pledges* encadrant publiquement l'utilisation des brevets (§1), et les mesures techniques de protection (MTP) protégeant techniquement le logiciel (§2)

A) Le Patent Pledge.

Il s'agit d'un engagement public de la part d'un détenteur de brevet à ne pas poursuivre en justice les développeurs pour violation de son brevet, à condition qu'ils satisfassent certaines conditions.

Souvent, il doit s'agir d'un brevet listé parmi ceux bénéficiant du pledge ("Pledged Patents"), et les bénéficiaires du pledge ne doivent pas invoquer ou utiliser le pledged patent (ou le fruit de son utilisation) contre l'entreprise ayant émis le Pledge, ne pas contester un Pledged Patent, et ne pas créer de contrefaçons. Enfin, l'entreprise se donne généralement le droit de terminer son pledge et d'engager une action dans la mesure où elle juge nécessaire pour se protéger. Le principe de la bonne foi est aussi fréquemment utilisé dans ce genre de déclaration.¹⁸⁴

¹⁸¹ Cour du District des Etats Unis, District de la Californie du Nord du 28 Février 2014, *Jobscience v CV Partners* (2014) WL 852477 (ND Cal 2014)

¹⁸² Noam Shemtov 244

¹⁸³ Directive (UE) 2016/943, considérant 16 :

¹⁸⁴ Voir les exemples du Patent Pledge de Google : <https://www.google.com/patents/opnpledge/pledge/> et de Tesla : https://www.tesla.com/fr_FR/about/legal#patent-pledge

Ainsi, le détenteur du brevet conserve ses droits tout en s'engageant publiquement à ne pas les exécuter d'une certaine manière. Le *patent pledge* n'est pas un contrat ou une licence, mais une déclaration souvent faite sur le site web du détenteur de brevet. Ce mécanisme est possible en droit américain grâce aux doctrines des *implied license*,¹⁸⁵ de *l'equitable estoppel*,¹⁸⁶ et des *laches*,¹⁸⁷ en ce qu'elles permettent au développeur de s'appuyer sur les représentations faites. Cet outil est largement utilisé pour les logiciels libres, qui peuvent comporter des parties de code appartenant à un brevet.¹⁸⁸ On peut la voir aussi dans le domaine industriel utilisant des logiciels, comme par exemple dans l'industrie automobile.¹⁸⁹ Mais cette pratique initialement privée n'est pas régulée par un organisme, ce qui crée un grand débat académique sur la création d'une entité ou mécanisme de régulation.¹⁹⁰ Ce phénomène est pour l'instant centralisé aux Etats-Unis.¹⁹¹

Le *patent pledge* va plus loin que les mécanismes de non-assertion et licence FRAND, cette dernière étant seulement applicables quand il est question d'un brevet essentiel figurant comme standard dans l'industrie.¹⁹² En droit de l'UE, il est plus courant de recourir à ces licences FRAND. On voit particulièrement des litiges en droit de la concurrence concernant ces licences dans le cas où un développeur contrevient aux termes de la licence d'un brevet essentiel, reconnu par un organisme d'élaboration de normes, mais que les termes de celle-ci ne sont pas justes, raisonnables et non-discriminants. Le développeur peut alors invoquer en défense l'article 102 TFUE.¹⁹³ Mais les licences FRAND ne sont pas des déclarations publiques et n'ont d'effet qu'entre le détenteur d'un brevet essentiel et le titulaire de la

¹⁸⁵ Cour d'Appel des Etats-Unis pour le circuit fédéral du 3 Janvier 1997, Wang Labs v. Mitsubishi, 103 F.3d 1571, 1582 (Fed. Cir.1997), ou une implied licence fut accordée pour la création d'un produit contrefaisant dans le but de développer des standards de production, car le contrefacteur s'était appuyé sur une déclaration publique et la conduite du détenteur du brevet

¹⁸⁶ Cour d'Appel des Etats-Unis pour le circuit fédéral du 6 Septembre 2001, Ecolab, Inc. v. Envirochem, Inc., 264 F.3d 1358, 1371 (Fed. Cir. 2001) quand (i) le détenteur du brevet, par une attitude trompeuse fait comprendre qu'il n'exécutera pas ses droits, (ii) qu'une autre personne se fie à cette conduite, et (iii) sera subira un préjudice si le détenteur exécute ses droits.

¹⁸⁷ Cour d'Appel des Etats-Unis pour le circuit fédéral, Mainland Industries, Inc. v. Standdal's Patents Ltd., 799 F.2d 746,748 (Fed. Cir. 1986) : le contrefacteur doit prouver que (i) l'action en contrefaçon a été engagée après un délai déraisonnable, et (ii) que cela lui cause préjudice.

¹⁸⁸ Voir exemple du patent pledge de Google

¹⁸⁹ Voir la déclaration d'Elon Musk : Elon Musk, All Our Patent Are Belong To You, 12 juin 2014

https://www.tesla.com/fr_FR/blog/all-our-patent-are-belong-you?redirect=no ; Exemple du patent pledge de Telsa

¹⁹⁰ European Commission, Invitation to tender – JRC/SVQ/2016/J.3/0003/OC, Interplay of Standard Developing Organizations and Intellectual Property Right Systems in the ICT Industry, (3 November 2016).

¹⁹¹ Jorge L. Contreras, The Evolving Patent Pledge Landscape, Avril 2018, CIGI Papers No. 166, p 12 ;

https://www.cigionline.org/sites/default/files/documents/Paper%20no.166%20Cover_0.pdf

¹⁹² Engagement pris par le détenteur de brevet d'octroyer des licences de leur brevet de standard à des conditions « fair, reasonable, and non-discriminatory » ; Jorge Contreras, "Patent Pledges" (2015) 47:3 Arizona State Law Journal, 543, 547

¹⁹³ Lorsqu'il y a un abus de position dominante : 120(a) du TFUE (redevances très élevées) ; Arrêt du Tribunal de première instance (grande chambre) du 17 septembre 2007, T-201/04, Microsoft Corp. v. Commission, p.II-3620 (refus de contracter) ;

licence. De ce fait, l'ampleur du phénomène est limitée dans l'Union Européenne qui ne semble pas s'être encore penchée sur la question de ces déclarations publiques appelées *patent pledges*.¹⁹⁴

L'intérêt premier de ces *pledges* est d'encourager l'innovation en ouvrant les barrières créées par les brevets, tout en assurant au détenteur du brevet une certaine sécurité.¹⁹⁵ Cela permet également de plus facilement développer des standards de technologie au bénéfice du détenteur du brevet : plus de produits seront compatibles avec le sien, qui deviendra un incontournable standard, lui faisant gagner des parts de marché.¹⁹⁶

B) Mesures Techniques de Protection (MTP) et leur encadrement juridique.

Au-delà des outils juridiques, les propriétaires de logiciels peuvent recourir à des MTP, c'est-à-dire des solutions techniques afin de permettre aux usagers de ne pas accéder ou reproduire leur logiciel. La technique de base est d'instaurer des contrôles d'accès et mécanismes de contrôles de reproduction, d'autant que des lois interdisent le contournement de ces MTP. Il est suggéré que le langage de ces lois est si large qu'il pourrait très bien englober les mécanismes d'authentification, d'acceptation des termes et conditions obligatoire, signature du code, obfuscation de code,¹⁹⁷ chiffrement.¹⁹⁸

En effet, au sein de l'UE, la Directive 2001/29/EC, protégeant les GUI, dispose de dispositions « anti-contournement ». Mais les GUI semblent peu facilement protégeables, au-delà de restreindre l'option copier-coller.¹⁹⁹ Concernant l'architecture du logiciel, c'est la Directive 2009/24/EC qui s'applique afin de protéger les MTP, mais de façon décevante : l'acte de contournement des MTP n'est pas sanctionné, c'est seulement la mise en circulation de moyens de contourner les MTP qui l'est.²⁰⁰

¹⁹⁴ Peut-être il en sera question dans la demande de consultation suivante, dont les résultats devraient être publiés en 2018, le professeur américain Jorge L Contreras devant y contribuer : European Commission, Invitation to tender – JRC/SVQ/2016/J.3/0003/OC Interplay of Standard Developing Organizations and Intellectual Property Right Systems in the ICT Industry, (3 November 2016).

¹⁹⁵ Jorge Contreras, "Patent Pledges" (2015) 47(3) Arizona State Law Journal 543, 573 ; University of Utah College of Law Research Paper No. 93. Lien vers SSRN: <https://ssrn.com/abstract=2525947>

¹⁹⁶ Jorge L. Contreras, The Evolving Patent Pledge Landscape, Avril 2018, CIGI Papers No. 166, p 5 ; https://www.cigionline.org/sites/default/files/documents/Paper%20no.166%20Cover_0.pdf

¹⁹⁷ Il s'agit de changer l'apparence du code pour le rendre difficilement décompilable, comme par exemple Skater : http://www.smrtx.com/RS/obfuscator_net.htm

¹⁹⁸ Noam Shemtov p 4

¹⁹⁹ Considérant 47 et Article 6(1) de la Directive 2001/29/EC.

²⁰⁰ Article 7(1)(c) de la Directive 2009/24/EC - d'autant que cet article est sans préjudice aux articles 5 et 6, permettant la décompilation et ingénierie inverse.

La situation est différente aux Etats-Unis. Le Digital Millenium Copyright Act (DMCA)²⁰¹ interdit de la même manière le contournement des MTP²⁰² ; en revanche, des exceptions viennent affaiblir cette interdiction : la section 1201(c)(1) dispose que l'interdiction n'affecte pas la *fair use doctrine*²⁰³ et la section 1201(f) autorise le contournement par ingénierie inverse afin d'obtenir l'interopérabilité.²⁰⁴ Si le but du contournement est permissible par le droit d'auteur, alors il n'y a pas de violation du DMCA. Cependant, les tribunaux n'ont pas une vision uniforme concernant ce raisonnement. Dans deux arrêts *Universal City Studios*, les tribunaux ont adopté une vision opposée : l'une considérant qu'il y avait violation des provisions anti-contournement indépendamment d'une violation du droit d'auteur²⁰⁵ et l'autre rejetant la violation en s'appuyant sur le *fair use*.²⁰⁶ Puis d'autres décisions disposaient que le contournement sans violation de droit d'auteur ne pouvait pas entraîner de poursuite sous la section 1201.²⁰⁷ Enfin, un dernier jugement dispose qu'il ne faut pas faire le lien entre contournement et violation des droits d'auteurs.²⁰⁸ Certains commentateurs appellent à une décision de la Cour Suprême afin de trancher.²⁰⁹ La conclusion est que les contournements de MTP ne sont pas totalement blanchis : ce fut le cas par exemple dans l'arrêt *Blizzard v BnetD*,²¹⁰ même lorsque l'ingénierie inverse a pour objectif de rendre deux logiciels interopérables.

Ainsi, avec cette alternative complémentaire, le propriétaire du logiciel peut contrôler l'accès à son logiciel et l'utilisation qui en est faite de façon sécurisée. En revanche, la législation protégeant ces MTP contre leur contournement n'est pas sans faille : soit elle est insuffisante en Union Européenne, soit instable aux Etats-Unis.

²⁰¹ Digital Millenium Copyright Act 1998, 17 U.S.C., Pub. L. 105-304, amendant le § 1201 de 17 U.S. Code

²⁰² Ibid section 1201 : No person shall circumvent a technological measure that effectively controls access to a work protected under this title.

²⁰³ "Nothing in this section shall affect rights, remedies, limitations, or defenses to copyright infringement, including fair use, under this title."

²⁰⁴ Voir note en bas de page n° 129

²⁰⁵ Cour d'Appel des Etats-Unis pour le deuxième circuit du 28 Novembre 2001, *Universal City studios, Inc v Corley*, 273 F.3d 429 (2nd Cir. 2001)

²⁰⁶ Cour du District des Etats-Unis, District du Sud de New York du 17 Août 2000, *Universal City Studios v Reminder*, 111 F.Supp.2d 294 (S.D.N.Y 2000)

²⁰⁷ Cour d'Appel des Etats-Unis pour le circuit federal du 31 Août 2004, *Chamberlain Group, Inc v Skyline Techs, Inc* 381 F.3d 1178, 1201 (Fed. Cir. 2004)

²⁰⁸ Cour d'Appel des Etats-Unis pour le neuvième circuit du 14 Décembre 2010, *MDY Industries v Blizzard Entertainment*, 629 F.3d 928 (9th Cir. 2010)

²⁰⁹ Noam Shemtov p 8

²¹⁰ Cour d'Appel des Etats Unis pour le huitième circuit du 1er Septembre 2005, *Davidson & Associates v. Jung*, 422 F.3d 630, 649 (8th Cir. 2005)

Paragraphe 2 : L'utilisation extensive de mécanismes connus assurant un accès limité au programme.

Il s'agit ici de voir dans quel sens le mode de distribution (A) et le droit des contrats de licence (B) des logiciels peuvent être utilisés afin de pallier l'ouverture autorisée par le droit d'auteur, ne protégeant pas les idées et posant l'exception du *fair use*.

A) Le contrat de service : la barrière technique du mode de distribution.

Le but du droit d'auteur est d'encourager la création et diffusion des expressions en laissant les idées libres de toute protection. Mais il faut noter que le plus grand obstacle à la réalisation de cette fonction pour les logiciels est que les procédés et concepts sont dissimulés par l'aspect technique de celui-ci.

La raison est que la publication des logiciels est faite sous la forme de code objet après compilation : le logiciel n'apparaît que sous la forme de langage compilé. Il n'est possible de découvrir le logiciel qu'en procédant à l'ingénierie inverse. La protection est ici accidentelle puisque l'intérêt premier est de rendre le code source lisible par la machine afin qu'elle comprenne les instructions.²¹¹ C'est la situation unique dans laquelle se trouve le programme au sein du régime du droit d'auteur : les idées non protégées ne peuvent être découvertes sans entraîner une violation du droit d'auteur dans certains cas.²¹²

Ce phénomène est encore plus flagrant avec les logiciels SaaS (Software as a Service - logiciel en tant que service). Cela consiste en la fourniture de logiciel, hébergé par un tiers à travers le cloud, pour mettre à disposition des services par le biais d'internet aux clients sous la forme d'un contrat de service de logiciel.²¹³ Il ne s'agit alors pas d'une licence.²¹⁴ De ce fait, le logiciel n'opère pas sur l'ordinateur du client, mais sur un serveur externe. Le client doit rentrer des données d'accès (identifiant, mot de passe...), et des données de sorties sont renvoyés au client sans qu'il puisse voir le procédé sous-jacent. De plus, la localisation des serveurs n'est pas une information connue du client. Ainsi, il n'a ni accès au code source, ni au code objet.²¹⁵ Les exemples les plus connus sont Microsoft Office 365 ou Dropbox.

En conséquence, puisque le logiciel est délivré sous forme de code incompréhensible, voire sans accès à aucun code par son mode d'utilisation, il est difficile pour les autres

²¹¹ <https://www.app.asso.fr/centre-information/base-de-connaissances/code-logiciels/la-protection-des-elements-composant-un-logiciel/la-protection-par-le-droit-dauteur-specifique-des-logiciels>

²¹² Noam Shemtov, p 28

²¹³ <https://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service>

²¹⁴ Noam Shemtov p 3

²¹⁵ *Ibid*

développeurs de l'étudier.²¹⁶ Mais cela n'empêchera pas un concurrent de développer un logiciel avec les mêmes fonctions en ayant simplement étudié le visuel du logiciel. La méthode de la « *clean room* » ou « *black box testing* » permet d'éviter le délit de contrefaçon, permettant de développer et d'utiliser un logiciel similaire pour lequel un contrat de licence est conclu. Il s'agit de l'étude d'un logiciel par une équipe dans le but d'extraire les principales fonctionnalités afin de les faire développer de manière autonome par une deuxième équipe étrangère à l'étude. Il est possible d'apporter la preuve que les similarités sont fortuites dans le cas où le titulaire du logiciel bénéficiant de la licence agit en contrefaçon.

B) La licence : la restriction par des clauses contractuelles.

La distribution de logiciel par le moyen de licence sous multiples formes est courante dans l'industrie.²¹⁷ Le propriétaire du logiciel peut y insérer des clauses pour élargir sa protection, notamment en interdisant l'ingénierie inverse et la reproduction du logiciel au-delà des éléments protégés par le droit d'auteur et incluant les éléments non-protégés. En effet, le langage utilisé dans ces clauses est bien souvent large.²¹⁸

L'UE a opté pour un rejet catégorique de ce type de clause. L'article 8 de la Directive 2009/24/CE dispose que « *toute disposition contractuelle contraire à l'article 6 ou aux exceptions prévues à l'article 5(2) et (3), est nulle et non avenue* ». Il semblerait que cette provision déteigne sur les clauses interdisant de reproduire des éléments non-protégeables pour des questions d'ordre public.²¹⁹ En revanche, aux Etats-Unis, on ne trouve pas ce type de disposition. Le mécanisme des *anti-reverse engineering licensing provisions* est largement utilisé afin de restreindre l'ingénierie inverse. En effet, comme nous l'avons vu précédemment, l'ingénierie inverse et la décompilation peuvent être autorisés par la *fair use doctrine* en droit d'auteur.²²⁰ Or, contrairement à l'UE, il est possible de lever cette exception

²¹⁶ De toute manière, même après décompilation, le développeur ne peut jamais retrouver le vrai code source puisqu'il n'aura pas accès aux instructions, commentaires, notes incluses par le programmeur original.

²¹⁷ Licence de site, licence de réseau, licence par place, licence par usagers, etc.

²¹⁸ Voir par exemple du Intel Software License Agreement : "This Agreement is subject to these conditions: 1. You may not copy, modify, rent, sell, distribute, externally display, externally perform or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software. 2. You may not reverse engineer, decompile, or disassemble the Software..." Accessible via le lien suivant : <https://software.intel.com/en-us/protected-download/385047/494732>

²¹⁹ Noam Shemtov p 48 : en effet, en recourant à l'ingénierie inverse, les parties non-protégées sont recopiées - il va de soit que cette situation est englobée par l'article 8.

²²⁰ Opt cit, voir les arrêts Sega, Atari, Sony

par des clauses contractuelles, plus précisément par le biais de provisions restrictives dans les contrats de licence, comme il fut réitéré par l'arrêt Davidson.²²¹

Cependant, des outils de régulation permettraient de contrôler cette tendance aux Etats-Unis. Il existe les concepts de *Federal* et *Constitutional preemption*,²²² invoqués quand une loi équivalente est contraire aux principes de la loi ou objectif du régime fédéral droit d'auteur. Mais il est majoritairement conclu que le Congrès ne cherche pas à restreindre la liberté contractuelle des parties, autorisant ces clauses.²²³ Il est également toujours possible d'invoquer le droit de la concurrence avec l'abus de position dominante²²⁴ et *anti-competitive agreements*,²²⁵ mais il est rare que les conditions soient réunies.²²⁶ Enfin, la clause peut être contrée par la jeune doctrine du *copyright misuse* qui sert à éviter de créer un droit exclusif étendu aux éléments non-protégés²²⁷ dans les litiges concernant la violation des droits d'auteurs. Néanmoins, la doctrine du *copyright misuse* n'est pour l'instant pas employée dans les litiges contractuels : elle empêche seulement l'auteur d'invoquer ses *droits d'auteurs*, mais ne rend pas la clause contractuelle nulle s'il invoque la violation contractuelle.²²⁸ Ainsi, il pourrait être possible pour les auteurs de logiciels de restreindre la possibilité de rendre compatible leur programme, afin de créer un monopole sur le code source et les fonctionnalités non protégées grâce au droit des contrats. Ce phénomène est critiqué par l'IEEE USA, dans un Board of Directors Position Statement concernant l'ingénierie inverse, qui invoque l'importance de pouvoir étudier les éléments non-protégés pour rendre interopérable des logiciels afin de développer l'industrie et condamne la faculté de limiter le *fair use privilege* par le droit des contrats.²²⁹

²²¹ Cour d'Appel des Etats-Unis pour le huitième circuit du, 1 Septembre 2005, *Davidson & Associates v. Jung*, 422 F.3d 630

²²² Section 301 du US Copyright Statute pour le federal pre-emption et US Constitution, Article VI, cl 2 pour le Constitutional pre-emption

²²³ Pour la *federal* : *Katz Dochtermann & Epstein, Inc v Home Box Office* (1999) 50 USPQ 2d 1957, 1959 ; et pour le *constitutional pre-emption* : *Davidson & Associates DBA Blizzard Entertainment, Inc; Vivendi Universal Inc v Jung et al* (2005)422 F3d 630 (8th cir)

²²⁴ Article 102 TFUE, Section 2 du Sherman Antitrust Act of 1890, 26 Stat. 209, 15 U.S.C. §§ 1–7. : Every person who shall monopolize, or attempt to monopolize, or combine or conspire with any other person or persons, to monopolize any part of the trade or commerce among the several States, or with foreign nations, shall be deemed guilty of a felony...

²²⁵ Section 1 du Sherman Act : Every contract, combination in the form of trust or otherwise, or conspiracy, in restraint of trade or commerce among the several States, or with foreign nations, is declared to be illegal.

²²⁶ R Pitofsky "Antitrust and intellectual property : unresolved issues at the heart of the new economy" 16 Berkeley Technology Law Journal 535 (2001) :

<https://scholarship.law.georgetown.edu/cgi/viewcontent.cgi?article=1312&context=facpub>

²²⁷ Cour d'Appel des Etats-Unis pour le quatrième circuit du 16 Août 1990, *Lasercomb Am, Inc v Reynolds* 911 F2d 970 (4th Cir 1990)

²²⁸ Jonas P. Herrell, « The Copyright Misuse Doctrine's Role in Open and Closed Technology Platforms" (2011), Berkeley Technology Law Journal, Volume 26, Number 1, p 441

²²⁹ IEEE-USA Position Statement, Reverse Engineering, Adopté par le IEEE-USA Board of Directors, 18 Juin 2016 ; Accessible via le lien suivant : <http://globalpolicy.ieee.org/wp-content/uploads/2016/08/16015.pdf>

Ainsi, les Etats-Unis semblent donner plus d'importance à la volonté des parties du contrat puisque les clauses contractuelles dans les licences ne sont pas facilement invalidées, tandis que l'UE reste tournée vers l'intérêt public avec la Directive 2009/24/EC : l'ingénierie inverse est bénéfique d'après le législateur.

Pour conclure sur ce chapitre, nous pouvons constater que de multiples alternatives sont possibles aux Etats-Unis et en Union Européenne : droit des marques, du design, des secrets d'affaire, de la concurrence, droit des contrats, outils techniques... Cependant, cela complique la protection des logiciels et la rend des fois trop forte. De ce fait, il faut se demander si des concepts plus larges ne permettraient pas aux auteurs de plus facilement sécuriser leur travail, et permettraient également à l'innovation de perdurer.

Chapitre 2 : L'utilisation unifiée des concepts alternatifs de protection des programmes d'ordinateur.

Si le libre accès au programme est bénéfique pour l'innovation, il faut songer à une alternative où le logiciel n'est pas protégé, abandonnant la quête de protection parfaite. Il conviendra alors d'examiner les deux extrêmes : une liberté rimant peut-être avec absence de protection (Section 1), face aux restes du débat extrême d'une protection *sui generis* (Section 2).

Section 1 : Le logiciel libre : une réelle absence de protection ?

Il s'agit d'étudier cette alternative qui est de mettre son logiciel en libre accès, libre exploitation pour tous. Nous verrons que cette liberté est tout de même encadrée (§ 1), et possède ses grands avantages et surmontables inconvénients (§ 2).

Paragraphe 1 : La liberté contrôlée par des licences.

Il a pu être relevé que les restrictions quant à la décompilation récompensaient démesurément les propriétaires de logiciels, l'industrie pouvant mieux bénéficier d'un relâchement de protection.²³⁰ Les logiciels libres sont ainsi très bien accueillis dans ce domaine. Cependant, cette liberté originellement attendue (A) n'empêche pas un encadrement (B).

A) Une liberté désirée.

²³⁰ Frederico Morando, "Software reverse engineering and open source software : do we need more FUD to be satisfied" (2008), 3rd Annual Conference of the EPIP Association Bern, Switzerland October 3rd-4th, 2008. - accessible via le lien suivant à la page 682 : <https://static1.squarespace.com/static/584e7eca6b8f5bf2d3f027ba/t/586b21b4d1758eb4882b0d80/1483416117157/EPIP-2008-papers.pdf>

C'est en réaction à ce blocage d'accès au code que les Free and Open Source Softwares (FOSS) sont nés. Historiquement, le logiciel libre fut introduit par Richard Stallman en 1983 lorsqu'il voulut écrire un système d'exploitation complet et entièrement libre compatible avec Unix appelé « GNU » (« Gnu's Not Unix »). Ensuite, en 1991, Linus Thorvalds crée le fameux système d'exploitation non-propritaire « Linux », suivi par de nombreux logiciels « open-source ». D'autres types de licences se sont développés.²³¹

Le mot « open source » vient de l'Open Source Initiative (OSI), lancée par Eric Raymond en 1998, toujours une alternative aux logiciels propriétaires, qui respecte l'idée de partage et propose une méthode correspondant mieux aux intérêts des ingénieurs. Selon la définition de l'OSI, des critères doivent être remplis afin de recevoir la qualification d'*open source* : libre redistribution du logiciel, accès au code source, préservation de l'intégrité du code, autorisation de le modifier ou de créer des dérivés, absence de discrimination entre les personnes ou des champs d'application, neutralité technologique...²³²

Ce modèle est très bénéfique pour l'industrie du logiciel, si bien que les géants de l'internet se tournent vers le logiciel libre : de Twitter à Google qui souhaite même récompenser les développeurs ayant contribué à des améliorations sur les logiciels libres les plus utilisés. En effet, il y a beaucoup d'avantages pour les entreprises à adopter ce modèle. De plus, Il est très facile de créer un projet, sur la plateforme en ligne de développement de logiciel GitHub par exemple, et de le rendre public sous ce régime en l'indiquant dans le formulaire.²³³

Ainsi, les logiciels *open source* permettent l'accès libre au code source, permettant aux autres développeurs de l'améliorer et le modifier. Il est aux antipodes de l'exclusivité d'exploitation réservée à l'auteur ne se dotant pas de ce type de licence.

B) Une liberté encadrée.

Cependant, les logiciels libres ne le sont pas tout à fait : ils utilisent les principes du droit d'auteur et n'appartiennent donc pas au domaine public. Les modalités de son utilisation sont donc encadrées et formalisées dans un contrat de licence, comme par exemple la « Licence Publique Générale dite GNU (GNU GPL) ». ²³⁴ Il s'agit d'un mode particulier d'exploitation : par le biais d'une licence, l'auteur autorise la copie, diffusion, modification de l'œuvre logicielle. Mais en aucun cas l'auteur du logiciel n'a renoncé à ses droits d'auteur ou voit sa

²³¹ MIT : <https://choosealicense.com/licenses/mit/> ; Apache 2.0 : <https://choosealicense.com/licenses/apache-2.0/>

²³² OSI, « The Open Source Definition » via le lien <https://opensource.org/osd> ; voir aussi la définition de la Free Software Fondation <http://www.gnu.org/philosophy/free.sw.fr.html>

²³³ Voir le lien de création de projet : <https://help.github.com/articles/creating-a-new-repository/>

²³⁴ Voir l'exemple de licence GNU GPLv3 : <https://choosealicense.com/licenses/gpl-3.0/>

liberté restreinte par les utilisateurs.²³⁵ Les logiciels libres sont alors bien soumis au droit : si les utilisateurs ne respectent pas les quatre libertés, ils enfreignent les clauses de la licence et perdent leur protection par le droit d'auteur sur leurs développements annexes.²³⁶

On peut recenser plusieurs catégories de licences. Il y a tout d'abord la *licence-type libre ou « restrictive »*, où l'utilisation de source étend son effet aux développements associés et interdit l'utilisation du code source dans une application propriétaire.²³⁷ Comme la licence est étendue aux améliorations réalisées, on dit qu'elle a un effet « contaminant ». Puis il y a la *licence-type semi-libre ou « diffusive »*, qui permet de redistribuer le logiciel avec modifications ou non, sous la licence initiale avec de nouveaux éléments qui eux peuvent être soumis à d'autres licences : cela donne des logiciels hybrides, mêlant licences libres et propriété.²³⁸ Enfin, il y a la *licence-type « permissive »*, qui impose seulement le respect de la paternité : on peut réutiliser ou inclure le code sous licence permissive d'un autre logiciel propriétaire.²³⁹ Bien souvent, celui qui distribue ou modifie le code source original soumis à ces licences devra reproduire les mêmes termes de la licence et les renoncations (*disclaimers*). Les licences open sources peuvent obtenir une certification.²⁴⁰

Ainsi, comme souligné par Pierre de Oliveira, la licence libre n'est pas une licence d'utilisation mais de diffusion de l'œuvre logicielle : c'est un contrat d'auteur sur l'œuvre en elle-même et emporte transfert des fruits d'exploitation.²⁴¹ De plus, la liberté d'accès ne rime pas avec gratuité et il est possible que certains logiciels libres impliquent le paiement d'une redevance dans le cadre de la licence. Toutefois, la grande majorité des logiciels libres sont gratuits.

Paragraphe 2 : Une liberté avantageuse mais pas sans risque.

L'industrie des logiciels évoluant à une vitesse importante, les besoins en idées sont grands. Le logiciel libre a de nombreux avantages aux yeux de cette industrie (A), mais comme toute liberté, ne reste pas sans problèmes (B).

²³⁵ G Maronne et A Tesselonikos, Logiciels Libres, quels enjeux pour l'entreprise ?, 2014, Expertises des Systèmes d'information, n°389, p 102, 103

²³⁶ FSF, Système d'exploitation GNU, « Qu'est-ce que le logiciel libre ? », accessible via le lien suivant : <https://www.gnu.org/philosophy/free-sw.fr.html>

²³⁷ Quelques exemples : GPL ("GNU General Public License"), CeCILL ("Cea Cnrs Inria Logiciel Libre")

²³⁸ Quelques exemples : MPL ("Mozilla Public License"), LGPL ("Licence publique générale limitée GNU ») ou CECILL-C.

²³⁹ BSD ("Berkely Software Distribution"), ASF ("Apache Software License") ou CeCILL-B

²⁴⁰ Les détenteur de licences peuvent, s'ils respectent les conditions de l'open source, apposer « OSI Certified » sur leur licence : Communiqué de Presse de la OSI accessible via le lien suivant : <https://opensource.org/pressreleases/certified-open-source.php>

²⁴¹ Pierre de Oliveira, Droit du Logiciel : Genèse d'une théorie matérialiste, 2012, FYP éditions, p 402, 403

A) Les innombrables atouts du logiciel libre.

La naissance du logiciel libre est basée sur des fondements économiques efficaces, même si il peut aller à rebours de la traditionnelle protection de l'innovateur : ici, l'important est la diffusion de la connaissance.

Le modèle de l'open source concurrence grandement le modèle « propriétaire » pour les programmes.²⁴² Il semble être une bonne alternative, surtout en ce qu'il correspond mieux au domaine technologique des logiciels, puisque par principe, un logiciel est souvent une amélioration d'un logiciel qui lui précède. Les développeurs s'inspirent de ce qu'il y a sur le marché afin de développer leur propre logiciel, et de créer des améliorations. On peut en déduire que le modèle *open source* permet d'avoir des logiciels plus performants grâce à la possibilité continue d'innovation.

La durée de vie d'un logiciel est courte, encore plus que celle d'un brevet ou d'un droit d'auteur, et cela en raison des évolutions rapides. De plus, le brevet et droit d'auteur sont parfois utilisés par les grandes entreprises afin de se « réserver » ou de « verrouiller » des domaines de la technologie et de la recherche.²⁴³ De surcroît, il est très courant de trouver des « trivial patent », soit des brevets de logiciel pour des systèmes très simples.²⁴⁴ Ces droits acquièrent donc une fonction stratégique en dépit de l'innovation. Au contraire, les logiciels libres permettent même aux plus petits d'accéder aux outils et de devenir plus compétitifs. Il permet de ne plus fonctionner par monopole puisque chaque développeur contribue au projet, peut bénéficier des apports collectifs sans en supporter les coûts et accroît le « stock de connaissance ».²⁴⁵

Ce système n'a pas entraîné de déboires. Les logiciels dont la réputation est bien établie, comme Linux, OpenOffice ou Apache, logiciels ayant contribué au succès de ce modèle. Par ailleurs, les partisans de la limitation de protection des logiciels, à l'époque de la consultation liée à la proposition de directive de 2002, étaient principalement en faveur des logiciels libres.²⁴⁶

B) Les problématiques risques d'insécurité juridique.

²⁴² Y Dietrich, Logiciels Open Source : une réalité juridique au sein des entreprises, RLDI, 2005, n°4, p 28 ; Greg Goth, "Open Source Business Models : Ready for Prime Time", 2005, IEEE Software, 98-100

²⁴³ F Lévêque et Y Menière, Economie de la Propriété Intellectuelle, Paris, La Découverte, 2003, p 62

²⁴⁴ Richard Stallman, The Anatomy of a Trivial Patent (2000, GNU Operating System) accessible via le lien suivant : <https://www.gnu.org/philosophy/trivial-patent.en.html>

²⁴⁵ François Pellegrini et Sébastien Cavenet, Droit des Logiciels : Logiciels privatifs et logiciels libres, 2013, PUF, P 469, 470

²⁴⁶ Recherche et propriété intellectuelle - Brevetabilité des logiciels, Examen dans le cadre de l'article 88-4 de la Constitution, Texte déposé au Sénat le 20/03/2002, Texte E 1965 - COM (2002) 92 final (Procédure écrite du 12 novembre 2002), via le lien <https://www.senat.fr/ue/pac/E1965.html>

Mais des insécurité juridiques peuvent naître par le biais de ces licences. Cela peut survenir lorsqu'un développeur souhaite incorporer du code sous licence libre dans un autre logiciel.

Dans le cas d'une licence type libre restrictive, si un élément de code source soumis à ce type de licence libre est intégré dans un autre logiciel (propriétaire), l'ensemble de ce logiciel propriétaire sera soumis à la licence type libre restrictive. Si c'est incompatible avec les autres titres tenus sur le logiciel, alors le développeur se verra contraint de ne pas intégrer le code soumis à la licence. Mais s'il choisit de l'intégrer tout de même, cela crée une insécurité juridique.²⁴⁷ Ainsi, dans le cadre d'une licence restrictive, l'entreprise disposant du logiciel final pourrait être contrainte à distribuer l'ensemble du logiciel sous licence GNU-GPL, et donc de rendre l'intégralité de son code accessible au public.²⁴⁸ Il peut également être considéré que, de la part de l'entreprise cliente du développeur, il y ait une utilisation contrefaisante si le formalisme de la licence libre n'est pas suivi.

Le problème est que les personnes concernées, les développeurs, n'ont pas nécessairement conscience de ce type de complexité.²⁴⁹ En conséquence, une sensibilisation à la dimension juridique de l'Open Source serait primordiale, d'autant que les licences prévoient des clauses exonératoires de responsabilité pour l'auteur initial et les auteurs secondaires du programme sous licence, représentant une absence de garantie pour l'entreprise utilisatrice. Mais dans le cas où l'entreprise aurait recours à un prestataire tiers, la responsabilité peut se négocier.

Dans ce cas, l'entreprise éditrice du logiciel a un devoir d'information concernant le code soumis à la licence libre incorporé dans le logiciel, et concernant les difficultés qui pourraient en naître.²⁵⁰ Il faut donc accueillir ce développement majeur du secteur informatique tout en maîtrisant les risques liés à la nouveauté et technicité du domaine. La Commission Européenne a même élaboré une licence, l'EURL v1.2,²⁵¹ certifiée par l'OSI²⁵² et faisant partie de la liste officielle de la FSF.²⁵³ Elle est disponible en 23 langues officielles de l'UE et

²⁴⁷ Logiciels Libres, Quels enjeux pour l'entreprise ?, 2014, Expertises des Systèmes d'information, n°389, p 102, 103

²⁴⁸ Logiciels Libres, Quels enjeux pour l'entreprise ?, 2014, Expertises des Systèmes d'information, n°389, p 102, 104

²⁴⁹ Des sociétés proposent de réduire les risques en analysant la provenance des lignes de code, comme par exemple la société Black Duck.

²⁵⁰ C'est le cas en France : CA Paris, 16 Septembre 2009, Jurisdata : 04/24298, relatif à la résolution d'un contrat ne respectant pas les exigences de la licence GNU GPL

²⁵¹ Dernière version du texte de la licence : <https://joinup.ec.europa.eu/collection/eupl/eupl-text-11-12>

²⁵² Patrice-Emmanuel Schmitz, EUPL analysis published in IFOSSLR, 11/12/2013 ; accessible via le lien suivant :

<https://joinup.ec.europa.eu/news/eupl-analysis-published-if>

²⁵³ <https://joinup.ec.europa.eu/news/better-late-never>

assure la compatibilité avec le droit d'auteur européen tout en restant presque similaire aux autres licences de logiciel libre.²⁵⁴

En conclusion, les logiciels libres ne signifient pas absence de protection pour l'auteur. La liberté n'est que dans l'exploitation. Cette liberté est encadrée, et comporte des risques à maîtriser, mais les bénéfices pour l'industrie du logiciel sont trop grands pour les considérer comme des obstacles. Il s'agit d'une alternative très satisfaisante.

Section 2 : La protection *sui generis* : une alternative oubliée ?

Dans les années 1980, la question d'une protection propre aux programmes d'ordinateurs avait engendré une vive discussion parmi les académiciens et instances internationales, perdue depuis les années 2000 (§ 1). En revanche, la question semble être toujours pertinente (§ 2).

Paragraphe 1 : Un grand débat éteint depuis les années 2000...

Il ne s'agit pas d'une réflexion nouvelle. En effet, les instances internationales avaient tout d'abord privilégié la protection *sui generis*.

L'OMPI en 1977 avait publié des dispositions types pour que les pays puissent compléter leur législation, incluant un régime *sui generis* qui se rapprochait du droit d'auteur. Les dispositions types furent reprises dans un projet de traité en 1983.²⁵⁵ Mais en cherchant quelle protection accorder aux programmes, il y a eu une plus forte préférence pour un régime de propriété intellectuelle qui existait déjà.²⁵⁶ Il s'agit d'une solution facile, en ce qu'il est moins complexe d'utiliser une solution préexistante et de l'adapter pour l'industrie afin d'obtenir un consensus international. Les Etats-Unis choisirent le droit d'auteur en 1980, entraînant les autres pays dans ce même choix.

La protection *sui generis* était toujours régulièrement envisagée par la doctrine (principalement américaine) dans les années 1980-1990,²⁵⁷ et des contours avaient été établis

²⁵⁴ Voir l'article d'un des contributeurs à cette licence Patrice-Emmanuel Schimtz, The European Union public Licence (EURL), International Free and OpenSource Software Review, Volume 5, Issue 2, Décembre 2013, p121, 122, accessible via le lien suivant : <https://www.ifosslr.org/public/ifosslr-v5i2.pdf>

²⁵⁵ Reto M. Hilty et Christophe Geiger, Breveter le logiciel? Une analyse juridique et socio-économique, 2005, Propriétés Intellectuelles, n° 16, p296, 298

²⁵⁶ Commission Européenne, Green Paper on Copyright and the Challenge of the New Technology - Copyright Issues Requiring Immediate Action, 1988, p 175 ; Peter S. Menell, "An Epitaph for Traditional Copyright Protection of Network Features of Computer Software"(1998) 43 Antitrust Bulletin, 651 ; United States' National Commission on New Technological Uses of Copyrighted Works (CONTU), Final Report of the National Commission on New Technological Uses of Copyrighted Works, 1978 ;

²⁵⁷ Voir par exemples : John C Phillips, Sui Generis Intellectual Property Protection for Computer Software, George Washington Law Review, Vol. 60, Issue 4 (April 1992), pp. 997-1041 ; Gerald Dworkin, Copyright, Patents and/or Sui Generis: What Regime Best Suits Computer Programs, International Intellectual Property Law & Policy, 1996, Vol. 1, pp. 165-182 ; Howard K. Szabo, International Protection of Computer Software: The Need for Sui Generis Legislation, Loyola of

dans le fameux article de Samuelson, Davis, Mitchell, Kapor et Reichman.²⁵⁸ L'article établissait une vision contre l'extension des protections existantes.²⁵⁹ Parce que les logiciels sont importants par leur comportement et non pour leur texte, qu'ils sont énormément complexes et qu'ils peuvent être facilement reproduits à l'identique sans enfreindre le droit d'auteur, l'absence de protection peut anéantir l'incitation à l'innovation.²⁶⁰ Ces auteurs proposaient une solution concentrée sur ce qui rend le logiciel attractif (soit son comportement et le savoir-faire), prenant en compte l'économie de marché, et en se basant seulement sur les protections existantes.²⁶¹ Plusieurs régimes sont proposés,²⁶² mais en général, il s'agirait d'assurer au propriétaire une période de blocage au début pour lui laisser de l'avance sur l'innovation, et d'autoriser par la suite l'appropriation contrôlée par des tiers, pour des raisons d'interopérabilité par exemple.²⁶³

Bien sûr, ces développements ont été formulés avant l'évolution des régimes de protections existants, ces évolutions permettant dans certains cas de mieux correspondre au logiciel. Mais ce manifeste reste un exemple incontournable du débat de protection *sui generis*. Cependant, avec l'avancement des protections existantes, le débat était clos, la proposition de directive de 2002 évoquant même le fait qu'une protection *sui generis* n'était pas nécessaire.²⁶⁴

Paragraphe 2 : ... Ressurgissant fréquemment dans les récentes analyses.

Même s'il semblerait que le débat soit officiellement clos, les académiciens ne restent pas fermés à cette idée. Souvent, on peut retrouver dans certains commentaires des arguments pour un régime *sui generis*, ou bien une adaptation plus poussée du régime actuel.

En effet, certains reconnaissent toujours que le type de protection *sui generis* permettrait de prendre mieux en considération la nature fonctionnelle du programme et de correspondre à ses besoins.²⁶⁵ Surtout, il a pu être dit que la protection par le droit d'auteur et des brevets était

Los Angeles International and Comparative Law Journal, Vol. 8, Issue 3 (1986), pp. 511-550 ; Jane C. Ginsburg, Four Reasons and a Paradox: The Manifest Superiority of Copyright over *Sui Generis* Protection of Computer Software, Columbia Law Review, Vol. 94, Issue 8 (December 1994), pp. 2559-2572.

²⁵⁸ P Samuelson, R Davis, Mitchell, D Kapor et J-H Reichman, A manifesto concerning the legal protection of computer programs, Columbia Law Review, 1994, volume 94, p 2308

²⁵⁹ Ibid 2311

²⁶⁰ Ibid 2339

²⁶¹ Ibid 2365

²⁶² Ibid 2413

²⁶³ Ibid 2371, 2408

²⁶⁴ Considérant 14

²⁶⁵ Lucie M.C.R. Guibault, "Les programmes d'ordinateur et le droit d'innovation technologique", 1997, 9 Cahiers de Propriété Intellectuelle, 171-202 ; Silke von Lewinski, International Copyright Law and Policy, Oxford University Press, 2008, p 234 ; Pamela Samuelson et al., « A Manifesto Concerning the Legal Protection of Computer Programs », 1994, Columbia Law Review, 2308 ; World Intellectual Property Organisation, « Model Provisions on the Protection of Computer Software », 1978, 14 Copyright 6

étendue mais que les conditions de ces régimes n'étaient pas pour autant suivies, se demandant s'il s'agissait d'une évolution naturelle ou d'une instrumentalisation pour arriver à ses fins.²⁶⁶

Il est fréquemment proposé de seulement aménager la protection par le brevet.²⁶⁷ Une protection plus courte pourrait être envisageable. Il faudrait également rompre avec le système de brevet unique pour tout type d'inventions et autoriser un brevet spécial de logiciel, prévoyant qu'un développeur ayant amélioré le logiciel breveté obtienne facilement une licence de dépendance.²⁶⁸ Mais l'inconvénient actuel est la longueur des procédures judiciaires dans le cas où le titulaire du brevet de base n'accorderait pas de licence de perfectionnement. Peut-être une rémunération équitable dès que le logiciel breveté est perfectionné et commercialisé sous une invention nouvelle serait préférable.²⁶⁹ Également, il a pu être évoqué de combiner le régime des brevets et des droits d'auteurs.²⁷⁰ Récemment aux Etats-Unis, et de façon plus spécifique, il a été proposé de rajouter un chapitre au US Copyright Act afin d'inclure une protection *sui generis* des GUI, permettant de mélanger le droit des design, du brevet, du *trade dress* et le droit d'auteur.²⁷¹ Mais toutes ces propositions ne font qu'étirer encore plus le système actuel. Cela peut être une solution, mais il faudrait également réfléchir à une protection propre pour ne pas dénaturer ces régimes de base.

D'autres sont plus ambitieux dans leurs propositions. Le Professeur le Tourneau a pu rappeler que les idées et concepts, qu'il différencie, ont une valeur commerciale importante et ne pas les protéger conduit à l'inertie et la rétention de l'innovation.²⁷² Pour qu'une idée soit protégeable en cas de litige, il propose qu'elle réponde à certains critères.²⁷³ Il relève que ce qui est à protéger pour les logiciels est *l'investissement* et non la création. La protection serait

²⁶⁶ Reto M. Hilty et Christophe Geiger, *Breveter le logiciel? Une analyse juridique et socio-économique*, 2005, *Propriétés Intellectuelles*, n° 16, p296, 297

²⁶⁷ Reto M. Hilty et Christophe Geiger, *Breveter le logiciel? Une analyse juridique et socio-économique*, 2005, *Propriétés Intellectuelles*, n° 16, p296, 309

²⁶⁸ Comme prévu en France sous l'article L 613-15 du Code de la Propriété Intellectuelle, ou l'article 31 de l'accord ADPIC.

²⁶⁹ Reto M. Hilty et Christophe Geiger, *Breveter le logiciel? Une analyse juridique et socio-économique*, 2005, *Propriétés Intellectuelles*, n° 16, p296, 310 ; Steven B. Toeniskoetter, *Protection of Software Intellectual Property in Europe: An Alternative Sui Generis Approach Intellectual Property Law Bulletin*, Vol. 10, Issue 1 (Fall 2005), pp. 65, 73

²⁷⁰ Joseph Francis III Agnelli, *Computer Programs under the United States Intellectual Property System: SUI Generis Legislation is Needed*, *Southern New England Roundtable Symposium Law Journal*, Vol. 3, Issue 1 (2008), pp. 109, 133

²⁷¹ Rachel Stigler, *Ooey GUI: The Messy Protection of Graphical User Interfaces*, *Northwestern Journal of Technology and Intellectual Property*, 2014, Vol.12 pp.215, 247

²⁷² Philippe le Tourneau, *L'illustre Gaudissart était visionnaire ! De la nécessité de protéger les idées apportant un avantage concurrentiel, soit indirectement par le parasitisme, soit de préférence par un droit sui generis spécifique à créer*, *Communication Commerce électronique* n° 10, Octobre 2017, étude 16

²⁷³ Elle ne doit pas être artistique, doit être divulguée, doit être distinctive, être le fruit de l'imagination et résultat d'un travail créatif et arbitraire, ne doit pas avoir été intégré dans le domaine public, doit avoir une valeur économique et procure un avantage concurrentiel, ne pas être nécessaire et habituelle dans le domaine en question, et qu'elle soit utilisée sans autorisation dans un but commercial pour être litigieuse.

temporaire. Ces développements rappellent ce qui était proposé par Samuelson, Davis, Mitchell, Kapor et Reichman. L'auteur explique de plus qu'une jurisprudence s'inspirant du droit des bases de données, octroyant une protection *sui generis* de 15 ans interdisant l'extraction et/ou la réutilisation par les tiers à l'auteur ayant fait preuve d'un investissement substantiel, pourrait suffire.²⁷⁴ D'autres auteurs proposent un système mêlant protection actuelle et protection *sui generis* supplémentaire.²⁷⁵ En revanche, il est difficile d'établir si la protection par le logiciel est opportune car les avis divergent. Les grandes entreprises dans le domaine du logiciel sont fortement en faveur, tandis que les petites et moyennes entreprises hésitent, et les développeurs sont opposés.²⁷⁶

Finalement, il est possible de douter sur la capacité des juridictions à se mettre d'accord sur une protection *sui generis*. Ce projet ayant été abandonné dans les années 1980 et tous les efforts s'étant orientés vers l'application des moyens connus, l'idée d'une protection *sui generis* semble perdue. Du côté des Etats-Unis, il n'était pas constaté de mouvement significatif vers l'établissement d'un régime *sui generis*.²⁷⁷ Il semble de même pour l'Union Européenne. Mais des dispositions prenant en compte la valeur et spécificité du logiciel seraient les bienvenues. Des meilleurs exemples viendraient de Singapour et d'Australie. Le Singapore Copyright Act of 1987²⁷⁸ (révisé en 2006) dispose à la section 35 que la décompilation de programmes pour des raisons commerciales et industrielles est exclue de la doctrine de l'usage légitime. La High Court of Australia trouva également que l'ingénierie inverse sur des programmes ne pouvait pas bénéficier de la doctrine de l'utilisation légitime, mais signala que cette problématique devait être adressée par voie législative.²⁷⁹

Conclusion

Pour conclure, nous pouvons constater que les acteurs de ce secteur doivent faire face à une contrefaçon multiforme et grandissante, que ce soit aux Etats-Unis ou en Europe. Des

²⁷⁴ Directive 96/9/CE du Parlement européen et du Conseil, du 11 mars 1996, concernant la protection juridique des bases de données, Journal Officiel n° L 77, 27.3.1996, p. 20–28, article 7.

²⁷⁵ Reto M. Hilty et Christophe Geiger, *Breveter le logiciel? Une analyse juridique et socio-économique*, 2005, *Propriétés Intellectuelles*, n° 16, p 296

²⁷⁶ Reto M. Hilty et Christophe Geiger, *Breveter le logiciel? Une analyse juridique et socio-économique*, 2005, *Propriétés Intellectuelles*, n° 16, p296, 304

²⁷⁷ Sheldon W. Halpern, Kenneth L. Port, Sean B. Seymore, *Fundamentals of United States Intellectual Property Law: Copyright, Patent, and Trademark*, 2015, 5th Edition, p 16

²⁷⁸ Singapore Copyright Act of 1987, revised edition 2006

²⁷⁹ Haute Court d'Australie, 30 Septembre 1999, *Data Access v Powerflex Services*, HCA 49; 202 CLR 1; 73 ALJR 1435; 166 ALR 228; 45 IPR 353

sanctions existent et ont été renforcées, mais le contentieux est très complexe et les juges ne sont pas vraiment formés pour les traiter. Cette contrefaçon multiforme est ainsi abordée par une protection multiforme : droit d'auteur, droit des brevets, concurrence déloyale, droit des marques et dessins et modèles, droit des contrats, mesures techniques, droit des secrets d'affaires, patent pledge... On peut en déduire que le droit des logiciels devient peut être un droit pour ceux qui savent naviguer à travers tous ces régimes. Globalement, il semblerait que les Etats-Unis autorisent plus souvent une protection forte à l'auteur du logiciel, tandis que l'Union Européenne assure le respect des fonctions de la propriété intellectuelle. Il a pu être conclu que les Etats-Unis offraient même parfois une propriété presque exclusive proche du monopole sur des fonctionnalités. Les logiciels libres viennent pallier cet obstacle à l'innovation en ouvrant l'accès aux logiciels tout en respectant les droits d'auteur. Cette alternative s'avère très bénéfique pour l'industrie. Néanmoins, ce domaine soulève de nombreuses questions juridiques et débats au sein des régimes étasunien et européen. Cela pourrait mener à terme à un besoin de réforme car ceci démontre que le système en place est trop complexe. Face aux insuffisances et excès de système, la réflexion sur une protection *sui generis* n'est pas totalement éteinte de part et d'autre de l'atlantique. Le besoin d'harmonisation et de transparence est en effet grand. L'idée d'une protection *sui generis* est très attrayante, d'autant que l'Union Européenne et les Etats-Unis ont déjà eu recours à ce genre de solution lorsque le droit d'auteur et des brevets ne permettaient pas une protection efficace.²⁸⁰ Cependant, le système de protections existant est trop ancré, il semble difficile d'obtenir un nouveau consensus sur une protection unique. Dès lors, il faudrait penser à aménager totalement l'un des systèmes comme proposé plus haut, particulièrement le droit des brevets, afin de mélanger une protection courte pour l'inventeur et les bénéfices du logiciel libre après l'écoulement d'une certaine période. Nous pouvons attendre avec plus ou moins d'impatience en Europe les développements à venir grâce à la nouvelle Juridiction Unifiée du Brevet qui fournira des procédures juridictionnelles plus simples, plus rapides et plus efficaces et harmonisera le droit matériel des brevets.²⁸¹

²⁸⁰ Avec la Directive 96/9/CE pour l'Union Européenne concernant les bases de données, et le US Semiconductor Chip Protection Act of 1984 (SCPA) 17 USC §§ 901-914 pour les Etats-Unis concernant la topographie de puces semi-conductrices.

²⁸¹ <https://www.unified-patent-court.org/>

Annexes

Exemple de code objet :

```
00000100: 965e 847f ea19 441f f8c3 f657 7e99 0d00  ^.....D...H...
00000110: 0ea0 05b5 d9fa 876d 6015 005d eb01 50bb  ..e...mi...].P.
00000120: fd07 cd60 2f00 8ab2 be75 0e7d 711e ba7c  ...'/....q...|
00000130: 5e52 c4e2 2c67 2bab dcdc 5c4b 039f 0b29  ^R...g+...AK.K)
00000140: 2fe8 effa 9f0e 7f43 5f7c cf52 bedd afa5  /.....C...R...
00000150: 6178 f393 3092 7431 435e 376e 067a a644  ax...B.t1C^7nfz.D
00000160: c4c8 ce02 70f9 0ce6 9f87 fb1f 07fe 751e  ...p.....u.
00000170: 1611 fc24 be00 2f94 4544 cba6 4c20 4c96  ...$....ED...L.
00000180: b55b c813 8895 0942 8640 f89f 9afb 0fc3  .{.....B.@.....
00000190: fna4 d9b9 9e89 dafb 11d0 9630 02a5 211a  .....X...l.
00000200: 407e 1e00 202a 1120 907b 642b d0ef 7dbb  @...(*...{d+...}
00000210: c647 03f9 cd8b d199 9b9d fbcf 82fe 7d57  .G.....W
00000220: h84c fec8 1024 7f8e 6347 4432 b812 51ca  .L...$....cGD2...Q.
00000230: ec9a fc5a 0234 2000 4540 03ea 401b e803  ...Z.A...f@...@...
00000240: 13c0 04b6 c011 b800 0fe0 0302 4320 8804  .....A(...
00000250: 7160 11e0 0214 9001 4420 1714 80b5 a018  q'1.....D.....
00000260: 9482 ad60 27a8 0e75 a011 3483 3670 1874  ...'...u...4...6p.t
00000270: 81c3 e034 3007 2e81 cb50 04dc 0352 300e  .c.4B...R...
00000280: 9e00 29f0 0ecc 4010 8485 cb10 1552 8774  ..)...@.....R...T
00000290: 2043 c81c b285 5800 1ba4 0305 4311 501c  .C...X...C.P...
00000300: 9408 2543 4248 0215 40eb a852 a81c aa80  ..NCBH...@...R...
00000310: ea81 60e0 5be0 2074 10ba 000d 43b7 a051  ..f...{t...C...Q
00000320: 0812 fa15 7a07 2330 09a6 c15a b011 6c05  h...l...#0...Z...l.
00000330: b360 4f30 080e 8417 c1c9 0332 381f 2a82  .'.0B.....2B...
00000340: b7c0 9570 037c 10ee 044f c397 e011 580a  ...p...[...O...X.
00000350: 3f81 a711 0010 131a a28b 3011 16c2 4642  ?.....l...0...F0
00000360: 9378 2409 1121 ab90 12a4 0260 40da 901e  .x$....l...3@...
00000370: a41f b90a 4091 a7c8 5b14 0645 4531 504c  ...H...[...EE1PL
00000380: 940b ca1f 1585 e2a2 06a1 56a1 36a3 aa51  .....V...6...Q
00000390: 0750 9da8 1ed4 55d4 206a 0af5 134d 460b  .P...U...{...MFk
00000400: a2cd d1ce e000 742c 3a19 9d8b 2e40 57a0  .....t...FW.
00000410: 0bd0 1d08 b3e8 11fa 38fa 1500 83a1 638c  .....B.....K.
```

Exemple de code source :

```
<!--[[if IE ]></endif-->
<!DOCTYPE html>
<!--[[if IEMobile 7]]><html class="iem7" lang="en" dir="ltr"></endif-->
<!--[[if lte IE 6]]><html class="lt-ie9 lt-ie8 lt-ie7" lang="en" dir="ltr"></endif-->
<!--[[if (IE 7)&(!IEMobile)]]><html class="lt-ie9 lt-ie8" lang="en" dir="ltr"></endif-->
<!--[[if IE 8]]><html class="lt-ie9" lang="en" dir="ltr"></endif-->
<!--[[if (gte IE 8) !gt IEMobile 7]]><!--><html lang="en" dir="ltr" prefix="og: http://ogp.me/naf"><!--</endif-->
</html>
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta charset="utf-8" />
<meta name="dc:creator" content="Technology Center 3900" />
<link rel="shortlink" href="/node/158423" />
<meta name="dc:type" content="Text" />
<meta name="dc:identifier" content="/patents-getting-started/patent-basics/types-patent-applications/design-patent-application-guide" />
<meta scheme="IS08601" name="dc:date.reviewed" content="2017-11-16" />
<meta scheme="IS08601" name="dc:date.created" content="20090701-04" />
<link rel="canonical" href="/patents-getting-started/patent-basics/types-patent-applications/design-patent-application-guide" />
<meta name="dc:format" content="text/html" />
<meta name="twitter:card" content="summary_large_image" />
<meta name="dc:subject" content="Patents, Sectional Applications, Nonprovisional (35 U.S.C. 111(a)), Design" />
<meta name="dc:title" content="Design Patent Application Guide" />
<meta name="description" content="Definition of a Design Types of Designs and Modified Forms" />
<meta name="uspto_hnh" nid="158423" title="Design Patent Application Guide" section="TC2900" vid="330433" en_id="82817977ec4bhl013fa7c10d0d7113e" />
<meta name="dc:description" content="Definition of a Design Types of Designs and Modified Forms" />
<meta scheme="DCTERMS:RFC1766" name="dc:language" content="en-US" />
<meta property="og:type" content="article" />
<meta property="og:description" content="Definition of a Design Types of Designs and Modified Forms" />
<meta property="og:title" content="Design Patent Application Guide" />
<meta name="twitter:title" content="Design Patent Application Guide" />
<title>Design Patent Application Guide | USPTO</title>
```

Base de l'exemple (Interface graphique utilisateur) :

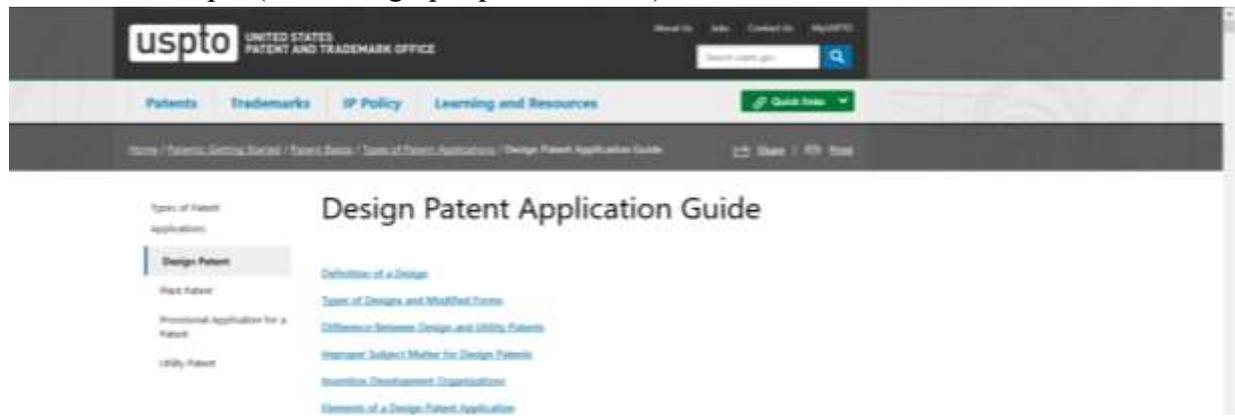


Table des sigles

ADPIC : Accord sur les Aspects des Droits de Propriété Intellectuelle qui Touchent au Commerce

CJUE : Cour de Justice de l'Union Européenne

CBE : Convention de Munich sur la délivrance des brevets européens

GUI : Interface Graphique Utilisateur (Graphical User Interface)

IEEE : Institute of Electrical and Electronics Engineers USA

OEB : Office Européen des Brevets

OMPI : Organisation Mondiale de la Propriété Intellectuelle

MTP : Mesure Technique de Protection

USCA 1976 : US Copyright Act 1976

USPTO : United States Patent and Trademark Office

Bibliographie

▪ Ouvrages :

Cornish et Llewelyn, *Intellectual property : patents, copyright, trademarks and allied rights*, page 455, 6ème édition, London: Sweet & Maxwell, Royaume-Uni, 2007

De Oliveira, *Droit du logiciel : Genèse d'une théorie matérialiste*, FYP éditions, 2010

Forest, *Droit des logiciels*, Lextenso édition, 2017

Ginsburg et Treppoz, *International Copyright Law: U.S. and E.U. Perspectives: Text and Cases*, Edward Elgar Publishing, 2015

Kamina, *Droit anglo-américain des propriétés intellectuelles : Copyright, dessins et modèles, brevets, obtentions végétales, marques, indications d'origine, concurrence déloyale, secrets d'affaires*, 1re édition, L.G.D.J, 2017

Lévêque et Menière, *Economie de la Propriété Intellectuelle*, Paris, La Découverte, 2003, p 62

Pellegrini et Cavenet, *Droit des Logiciels : Logiciels privatifs et logiciels libres*, PUF, 2013

Rowland, Kohl et Charlesworth, *Information Technology Law*, 4ème édition, Routledge, Royaume-Uni, 2012

Sheldon, Halpern, Kenneth, Port, Sean et Seymore, *Fundamentals of United States Intellectual Property Law: Copyright, Patent, and Trademark*, Kluwer Law International, 5th Edition, 2015

Shemtov, *Beyond The Code: Protection of Non-Textual Features of Software*, Oxford University Press, 2017

Van Rooijen, *The Software Interface Between Copyright and Competition Law: A legal analysis of interoperability in computer programs*, Kluwer Law International, 2010

La contribution de la jurisprudence à la construction de la propriété intellectuelle en Europe, sous la direction de Christophe Geiger, LexisNexis, Collection du C.E.I.P.I., 2013

▪ Revue juridique :

Arizona State Law Journal, 2015, 47:3, Contreras, Patent Pledges, 543

Berkeley Technology Law Journal, 2011, Volume 26, Herrell, The Copyright Misuse Doctrine's Role in Open and Closed Technology Platforms, Number 1, p 441

Berkeley Technology Law Journal, 2001, 16, Pitofsky "Antitrust and intellectual property: unresolved issues at the heart of the new economy", 535

Columbia Law Review, 1994, volume 94, Samuelson, Davis, Mitchell, Kapor et Reichman, A manifesto concerning the legal protection of computer programs, p 2308

European Intellectual Property Review, 1997, Krockner, The Computer Directive and balance of rights, 247

European Intellectual Property Review, 2001, Laakkonen and R Whaite, 'The EPO leads the way, but where to?', 244

Intellectual Property Law Bulletin, 2005, Vol. 10, Issue 1, Toeniskoetter, Protection of Software Intellectual Property in Europe: An Alternative Sui Generis Approach, p. 65

International Free and OpenSource Software Review, 2013, Volume 5, Issue 2, Schimtz, The European Union public Licence (EUPL), p121

Journal of Intellectual Property Law & Practice, 2007, Vol 2, Deene, Originality in software law: Belgian doctrine and jurisprudence remain divided, 692

CIGI Papers No. 166, 2018, Contreras, The Evolving Patent Pledge Landscape, p 12

IEEE Software Magazine, 2005, Goth, "Open Source Business Models : Ready for Prime Time", 98-100

Notre Dame Law Review, 2005, 81, Gibson, « Once and Future Copyright », 167

Northwestern Journal of Technology and Intellectual Property, 2015, 12(3), Stigler, 'Ooey GUI : the messy protection of graphical user interfaces', 240

Southern New England Roundtable Symposium Law Journal, 2008, Vol. 3, Issue 1, Agnelli, Computer Programs under the United States Intellectual Property System: SUI Generis Legislation is Needed, p. 109

3rd Annual Conference of the EPIP Association Bern, Switzerland October 3rd-4th, 2008, Morando, "Software reverse engineering and open source software : do we need more FUD to be satisfied"

Communication Commerce électronique, 2017, n° 10, étude 16, Le Tourneau, L'illustre Gaudissart était visionnaire ! De la nécessité de protéger les idées apportant un avantage concurrentiel, soit indirectement par le parasitisme, soit de préférence par un droit sui generis spécifique à créer.

Expertises des Systèmes d'information, 2014, n°389, Maronne et Tessalonikos, Logiciels Libres, quels enjeux pour l'entreprise ?, p 102

Gazette du Palais, 2014, n°310, Marino, États-Unis : Alice au pays des brevets de logiciels, page 20

Propriétés Intellectuelles, 2005, n° 16, Hilty et Geiger, Breveter le logiciel? Une analyse juridique et socio-économique, p 296

Revue Lamy Droit de l'immatériel, 2005, n°4, Dietrich, Logiciels Open Source : une réalité juridique au sein des entreprises, p 28

Revue Lamy Droit de l'immatériel, 2012, Castets-Renard, L'interprétation du droit d'auteur européen sur les logiciels par la Cour de justice, no. 87

Revue Lamy droit de l'immatériel, 2016, n°123, Valat, Grimaud, Propositions pour un encadrement du régime juridique des logiciels, 01/02/2016, p 34-36

Computer Law & Practice, 1993, Vol 9, Wiebe, « European copyright protection of software from a German perspective », p 79

▪ **Textes de lois :**

Textes internationaux :

Aspects des droits de propriété intellectuelle qui touchent au commerce, Annexe 1 C de l'Accord de Marrakech instituant l'Organisation mondiale du commerce, signé à Marrakech, au Maroc, le 15 avril 1994.

Convention de Berne pour la protection des œuvres littéraires et artistiques du 9 Septembre 1886, Article 7

Convention de Munich sur la délivrance des brevets européens (CBE) du 5 Octobre 1973, Article 52(1), (2)

Traité de l'OMPI sur le droit d'auteur (WCT) du 20 décembre 1996

Traité sur le Fonctionnement de l'Union Européenne, Journal officiel n° C 326

Union Européenne :

Directive 91/250/CEE du Conseil, du 14 mai 1991, concernant la protection juridique des programmes d'ordinateur, OJ L 122, 17.5.1991, p. 42–46 (abrogée)

Directive 96/9/CE du Parlement européen et du Conseil, du 11 mars 1996, concernant la protection juridique des bases de données, Journal Officiel n° L 77, 27.3.1996, p. 20–28

Directive 2001/29/CE du Parlement européen et du Conseil du 22 mai 2001 sur l'harmonisation de certains aspects du droit d'auteur et des droits voisins dans la société de l'information, Journal Officiel n° L 167, 22.6.2001, p. 10–19

Directive 2009/24/CE du Parlement européen et du Conseil du 23 avril 2009 concernant la protection juridique des programmes d'ordinateur (version codifiée) [2009] OJ L 111, 5.5., p. 16–22

Directive (UE) 2015/2436 du Parlement Européen et du Conseil du 16 décembre 2015 rapprochant les législations des États membres sur les marques (Texte présentant de l'intérêt pour l'EEE), JO L 336, p. 1–26

Directive (UE) 2016/943 du Parlement européen et du Conseil du 8 juin 2016 sur la protection des savoir-faire et des informations commerciales non divulgués (secrets d'affaires) contre l'obtention, l'utilisation et la divulgation illicites (Texte présentant de l'intérêt pour l'EEE), OJ L 157, 15.6.2016, p. 1–18

Règlement (CE) n° 6/2002 du Conseil du 12 décembre 2001 sur les dessins ou modèles communautaires, JO L 3, p. 1–24

Règlement (UE) n° 1257/2012 du Parlement européen et du Conseil du 17 décembre 2012 mettant en œuvre la coopération renforcée dans le domaine de la création d'une protection unitaire conférée par un brevet, Journal Officiel n° L 361, p. 1–8

Proposition de directive du Parlement européen et du Conseil concernant la brevetabilité des inventions mises en œuvre par ordinateur, Journal officiel n° 151 E du 25/06/2002 p. 0129 - 0131

Etats-Unis :

Copyright Act of 1976, 17 U.S.C. §§ 101-810, Pub.L. 94–553

Defend Trade Secrets Act of 2016, Public Law 114-153 114th Congress (11 May 2016),

Digital Millenium Copyright Act 1998, 17 U.S.C., Pub. L. 105-304

National Conference of Commissioners on Uniform State Law, Uniform Trade Secrets Act (UTSA) of 1979, amended in 1985

Lanham (Trademark) Act, Pub.L. 79–489, 60 Stat. 427, enacted July 5, 1946, codified at 15 U.S.C. § 1051 et seq. (15 U.S.C. ch. 22)

Semiconductor Chip Protection Act of 1984 (SCPA) 17 USC §§ 901-914

Sherman Antitrust Act of 1890, 26 Stat. 209, 15 U.S.C. §§ 1–7.

Singapore Copyright Act of 1987, revised edition 2006

Divers :

Code Civil, article 1240

Code de la Propriété Intellectuelle, articles L. 121-1, L. 121-4, L. 121-7.

▪ Recueil de jurisprudence :

Jurisprudences nationales :

Tribunal de Commerce de Bobigny du 21 Janvier 1995, Computer Associates International, Inc v Faster and Altai, 166 RIDA 324

Tribunal de première instance (grande chambre) du 17 septembre 2007, T-201/04, Microsoft Corp. v. Commission, p.II-3620

Cour d'appel d'Aix en Provence, 1ère chambre C, Arrêt du 10 octobre 2013 n° 12/22423

Cour de cassation, Première Chambre Civile, Arrêt du 14 novembre 2013, N° de pourvoi: 12-20687

Haute Cour de justice d'Angleterre et du Pays de Galles, Chancery Division du 30 Juillet 2004, Navitaire, Inc v Easyjet Airline Company and Bulletproof Technologies, Inc, England and Wales High Court Report (Ch), 1725.

Haute Cour de justice d'Angleterre et du Pays de Galles, Chancery Division, 1916, University of London Press Ltd v University Tutorial Press Ltd, 2 Ch 601

Chambre des Lords du Royaume-Uni, Reckitt & Colman Products Ltd v Borden, Inc. And Others, 1990, 1 All ER 873.

Haute Court d'Australie, 30 Septembre 1999, Data Access v Powerflex Services, HCA 49; 202 CLR 1; 73 ALJR 1435; 166 ALR 228; 45 IPR 353

Jurisprudences européennes :

Cour de Justice de l'Union Européenne, troisième chambre du 22 Décembre 2010, affaire C-393/09, Bezpečnostní softwarová asociace, Journal officiel n° C11

Cour de Justice de l'Union Européenne, quatrième chambre du 16 juillet 2009, affaire C-5/08, Infopaq International A/S contre Danske Dagblades Forening, Rec., p. I-6569.

Cour de Justice de l'Union Européenne du 18 Juin 2002, affaire 299/99, Koninklijke Philips Electronics NV contre Remington Consumer Products Ltd., 2002 I-05475

Cour de Justice de l'Union Européenne, Grande Chambre du 2 mai 2012, SAS Intitute Inc. c/ World Programming Ltd., affaire C-406/10, RLDI 2012/82, n° 2741

Cour de Justice de l'Union Européenne du 6 février 2003, aff. C-245/00, Sena , Rec., p. I-1251.

Cour de Justice de l'Union Européenne en sa première chambre du 22 juin 2006, affaire C-24/05 P, Storck v OHIM, Recueil du jurisprudence 2006 I-05677

Office Européen des Brevets, affaire T 1173/97, Chambre des recours du 1er Juillet 1998, Computer program product c. IBM, ECLI:EP:BA:1998:T117397.19980701

Office Européen des Brevets, affaire T 1741/08, Chambre des recours du 2 Août 2012, GUI layout/SAP, ECLI:EP:BA:2012:T174108.20120802

Office Européen des Brevets, affaire T 0258/03, Chambre des recours du 21 Avril 2004, Méthode d'enchères/HITACHI, 21 Avril 2004, ECLI:EP:BA:2004:T025803.20040421

Office Européen des Brevets, affaire T 0424/03, Chambre des recours du 23 Février 2006, Microsoft (Clipboard format), ECLI:EP:BA:2006:T042403.20060223

Office Européen des Brevets, affaire T931/95, Chambre des recours du 8 Septembre 2000, Pension Benefit Systems Partnership, ECLI:EP:BA:2000:T093195.20000908

Office Européen des Brevets, affaire T1188/04, Chambre des recours du 5 Mars 2008, Sharp/Graphical user interface, ECLI:EP:BA:2008:T118804.20080305

Office Européen des Brevets, affaire T 208/84, Chambre de Recours Technique du 15 juillet 1986, Vicom Systems Inc., JO OEB 1987/014

Jurisprudences des Etats-Unis :

Cour du District des Etats Unis, District de la Californie du Nord du 1er Avril 2014, Ingrid & Isabel, LLC v Baby Be Mine, LLC F. Supp.3d, 2014 WL 4954656 (N.D. Cal. 2014)

Cour du District des Etats Unis, District de la Californie du Nord du 28 Février 2014, Jobsience v CV Partners (2014) WL 852477 (ND Cal 2014)

Cour du District du Centre de la Californie du 13 Février 2012, Aqua Connect, Inc. v. Code Rebel, LLC., Case No CV 11-5764-RSWL

Cour du District des Etats-Unis, District du Sud de New York du 10 Juin 1988, Telerate Systems, Inc. v. Caro, 689 F. Supp. 221 (S.D.N.Y. 1988)

Cour du District des Etats-Unis, District du Sud de New York du 17 Août 2000, Universal City Studios v Reminder, 111 F.Supp.2d 294 (S.D.N.Y 2000)

Cour de district des États-Unis, District de New York, du 30 Mai 2012, Tetris Holding, LLC v Xio Interactive, Inc 863 F.Supp.2d 394 (D.N.J. 2012)

Cour d'appel des États-Unis pour le premier circuit, 5 Mars 1995, Lotus Development Corp. V. Borland International, Inc., 49 F.3d 807, 821

Cour d'Appel des Etats-Unis pour le deuxième circuit du 22 Juin 1992, Computer Associates v Altai, 982 F 2d 693

Cour d'appel des États-Unis pour le deuxième circuit du 10 Novembre 1930, Nichols v. Universal Pictures Corp., 45 F.2d 119.

Cour d'Appel des Etats-Unis pour le deuxième circuit du 28 Novembre 2001, Universal City studios, Inc v Corley, 273 F.3d 429 (2nd Cir. 2001)

Cour d'appel des États-Unis pour le troisième circuit, 30 Août 1983, Apple Computer Inc v Franklin Computer Corp., 714 F.2d 1240, 219 USPQ 113

Cour d'Appel des Etats-Unis pour le troisième circuit, 24 Juin 1981, Williams Elecs., Inc. v. Artic Int'l, Inc., 685 F.2d 870

Cour d'Appel des Etats-Unis pour le troisième circuit, 4 Août 1986, Whelan Associates, Inc. v. Jaslow Dental Laboratory Inc, 797 F.2d 1222, 230 USPQ 481

Cour d'Appel des Etats-Unis pour le quatrième circuit du 16 Août 1990, Lasercomb Am, Inc v Reynolds 911 F2d 970 (4th Cir 1990)

Cour d'Appel des Etats Unis pour le huitième circuit du 1er Septembre 2005, Davidson & Associates v. Jung, 422 F.3d 630

Cour d'appel des États-Unis pour le neuvième circuit, 8 Février 1984, Apple Computer Inc. v Formula Intern. Inc., 725 F.2d 521, 525

Cour d'Appel des Etats-Unis pour le neuvième circuit du 14 Décembre 2010, MDY Industries v Blizzard Entertainment, 629 F.3d 928 (9th Cir. 2010)

Cour d'Appel des Etats-Unis pour le neuvième circuit du 20 Octobre 1992, Sega Enterprises Ltd. v. Accolade Inc., 977 F.2d 1510, 24 USPQ2d 1561

Cour d'Appel des Etats Unis pour le neuvième circuit du 20 Février 2000, Sony Computer Entertainment, Inc v Connectix Corp, 203 F.3d 596

Cour d'appel pour le circuit federal, dixième circuit, Gates Rubber Co. V Bando Chem Indus Ltd, 9 F.3d 823, 848 (10th Cir. 1993)

Cour d'Appel des Etats-Unis pour le circuit fédéral du 10 Septembre 1992, Atari Games Corp. v. Nintendo of America Inc., 975 F.2d 832, 24 USPQ2d 1015

Cour d'appel des États-Unis pour le circuit fédéral du 14 Avril 1999, AT & T Corp v Excel Communications, 172 F 3d 1352

Cour d'appel des États-Unis pour le circuit fédéral du 3 Septembre 2014, buySAFE, Inc., v. Google, Inc, 765 F. 3d. 1350

Cour d'Appel des Etats-Unis pour le circuit federal du 31 Août 2004, Chamberlain Group, Inc v Skyline Techs, Inc 381 F.3d 1178, 1201 (Fed. Cir. 2004)

Cour d'appel des États-Unis pour le circuit fédéral du 11 Juillet 2014, Digitech Image Technologies, LLC v. Electronics For Imaging, Inc. ,758 F. 3d 1344

Cour d'Appel des Etats-Unis pour le circuit fédéral du 6 Septembre 2001, Ecolab, Inc. v. Envirochem, Inc., 264 F.3d 1358, 1371 (Fed. Cir. 2001)

Cour d'appel des États-Unis pour le circuit fédéral du 12 Mai 2016, Enfish LLC v. Microsoft Corp., 822 F.3d 1327)

Cour d'Appel des Etats-Unis pour le circuit fédéral, Mainland Industries, Inc. v. Standdal's Patents Ltd., 799 F.2d 746,748 (Fed. Cir. 1986)

Cour d'appel des États-Unis pour le circuit federal, 9 Mai 2014, Oracle America v. Google Inc., 750 F.3d 1339

Cour d'appel des États-Unis pour le circuit fédéral du 29 Juillet 1994, Re Alappat, 33 F 3d 1526

Cour d'appel des États-Unis pour le circuit fédéral du 8 Mars 2017, Thales Visionix Inc. v. U.S., 850 F.3d 1343

Cour d'appel pour le Circuit Fédéral du 15 Août 2017, Visual Memory LLC v NVIDIA Corp, 2016-2254

Cour d'Appel des Etats-Unis pour le circuit fédéral du 3 janvier 1997, Wang Labs v. Mitsubishi, 103 F.3d 1571, 1582 (Fed. Cir.1997),

Cour Suprême des Etats-Unis du 19 Juin 2014, Alice corp v CLS Bank International, 573 U.S.

Cour Suprême des États-Unis du 19 Janvier 1879, Baker v Selden, 101 US 99

Cour Suprême des Etats Unis du 28 Juin 2010, Bilski v Kappos, 561 U.S

Cour Suprême des Etats-Unis du 21 Février 1989, Bonito Boats, Inc v Thinder Craft Boats, Inc, 141 US 489

Cour Suprême des États-Unis du 16 Juin 1980, Diamond c. Chakrabarty, 447 U.S. 308

Cour Suprême des Etats-Unis du 3 Mars 1981, Diamond v Diehr, 450 US 175.

Cour Suprême des Etats Unis du 27 Mars 1991, Feist Publications, Inc., v. Rural Telephone Service Co., 499 U.S. 340, 358 (1991).

Cour Suprême des Etats-Unis du 20 Novembre 1972, Gottschalk v Benson, 409 US 63

Cour Suprême des États-Unis du 13 Mai 1973, Kawanee Oil Co v Bicron Corp, 416 US 470, 476 (1974)

Cour Suprême des Etats-Unis du 10 Juillet 1853, Le Roy v. Tatham, 55 U.S. 156

Cour suprême des États-Unis du 20 mars 2012, Mayo c/ Prometheus, 566 U.S.

Cour Suprême des Etats-Unis du 26 Juin 1992, Two Pesos, Inc, Taco Cabana, Inc. 505 US 763, 764.

▪ **Documents officiels :**

Arrêté du 22 décembre 1981 relatif à l'enrichissement du vocabulaire de l'informatique, JO 17-01-1982 p. NC 624-626

Avis du Comité économique et social sur la "Proposition de directive du Parlement européen et du Conseil concernant la brevetabilité des inventions mises en œuvre par ordinateur" (COM(2002) 92 final — 2002/0047 (COD)), Journal officiel n° C 061 du 14/03/2003 p. 0154 - 0163,

European Commission, Invitation to tender – JRC/SVQ/2016/J.3/0003/OC Interplay of Standard Developing Organizations and Intellectual Property Right Systems in the ICT Industry, (3 November 2016)

Office Européen des Brevets, Directives relatives à l'examen pratiqué, Partie G – Brevetabilité, Chapitre II, (3)

Rapport de la Commission au Conseil, au Parlement européen et au Comité économique et social sur la mise en œuvre et les effets de la directive 91/250/CEE concernant la protection juridique des programmes d'ordinateur, COM/2000/0199 final, p 21

Recherche et propriété intellectuelle - Brevetabilité des logiciels, Examen dans le cadre de l'article 88-4 de la Constitution, Texte déposé au Sénat le 20/03/2002, Texte E 1965 - COM (2002) 92 final (Procédure écrite du 12 novembre 2002),

US Copyright Office, Circular 61, Copyright Registration of Computer Programs

United State Patent and Trademark Office, Manual of Patent Examining Procedure (MPEP), Chapter 2106.04(a) Abstract Ideas, Ninth Edition, Revision 08.2017, Last Revised January 2018

United State Copyright Office, Software-Enabled Consumer Products : a report of the register of copyrights, December 2016.

▪ **Sources de l'internet :**

Agence pour la protection des programmes : <https://www.app.asso.fr/centre-information/base-de-connaissances/code-logiciels/la-protection-des-elements-composant-un-logiciel/la-protection-par-le-droit-dauteur-specifique-des-logiciels>

Authors Alliance : <https://www.authorsalliance.org/2018/02/14/copyright-registration-part-1-why-register-your-copyright/>

Patentology : <https://blog.patentology.com.au/2017/04/new-uspto-data-set-reveals-recent.html>

Choose an open source license : <https://choosealicense.com/licenses/mit/>

Commission Européenne, Base de données des communiqués de presse : http://europa.eu/rapid/press-release_MEMO-02-32_en.htm?locale=FR

Système d'exploitation GNU ; <http://www.gnu.org/philosophy/free.sw.fr.html>

GitHub : <https://help.github.com/articles/creating-a-new-repository/>

Google Open Patent Non-Assertion Pledge: <https://www.google.com/patents/opnpledge/pledge/>

Google Patents : <https://patents.google.com/patent/US9223477B2/en>

IP Watchdog: <http://www.ipwatchdog.com/2017/12/17/software-patent-eligibility-federal-circuit-2017/id=91169/>

Forbes : <https://www.forbes.com/sites/oliverherzfeld/2015/10/26/five-reasons-to-copyright-your-software-now/#3e0027d34dd6>

Joinup : <https://joinup.ec.europa.eu/collection/eupl/eupl-text-11-12>

Northern District of Texas Blog : <http://www.ndtexasblog.com/?p=3550>

Open Source Initiative : <https://opensource.org/osd>

TechTarget : <https://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service>

Tesla : https://www.tesla.com/fr_FR/blog/all-our-patent-are-belong-you?redirect=no

USPTO : <https://www.uspto.gov/sites/default/files/documents/ieg-qrs.pdf>

EPO : https://worldwide.espacenet.com/publicationDetails/biblio?FT=D&date=19990928&DB=&locale=en_EP&CC=US&NR=5960411A&KC=A&ND=3#

Table des matières

Introduction.....	1
Partie 1 : Les modes traditionnels de protection des programmes d'ordinateur dans les législations étasuniennes et européennes.....	5
Chapitre 1 : Une opposition des législations en matière de brevet.....	5

Section 1 : Les divergences de bases législatives quant aux inventions brevetables.....	5
Paragraphe 1 : La brevetabilité du logiciel en Europe : le refus de principe.....	5
A) Une non-brevetabilité presque catégorique.....	5
B) Une tentative d'évolution avortée.....	6
Paragraphe 2 : La brevetabilité du logiciel aux Etats-Unis : une ouverture de principe.....	7
A) L'ouverture législative des inventions brevetables.....	7
B) La précision jurisprudentielle quant à la brevetabilité des logiciels.....	8
Section 2 : Opposition des développements et des exceptions jurisprudentielles.....	9
Paragraphe 1 : Une volonté grandissante de protection en Europe.....	9
A) Le sentiment de nécessité reflété dans l'interprétation de la loi.....	10
B) La jurisprudence européenne au service des brevets.....	10
Paragraphe 2 : Un système s'essouffant de l'autre côté de l'Atlantique.....	12
A) La fin d'un système trop accueillant.....	12
B) Un nouveau jour impactant les droits acquis.....	13
Chapitre 2 : Une convergence apparente des législations en matière de droit d'auteur.....	14
Section 1 : Une inclusion sans encombre des programmes informatiques dans les deux systèmes.....	15
Paragraphe 1 : L'application des critères connus du droit d'auteur européen.....	15
A) Le programme d'ordinateur comme œuvre littéraire.....	15
B) L'incontournable critère de l'originalité appliqué aux programmes informatiques.	16
Paragraphe 2 : L'application des critères connus du droit d'auteur étasunien.....	18
A) L'adaptation au droit d'auteur : Une œuvre littéraire écrite.....	18
B) Les critères de fixation écrite, d'originalité et d'enregistrement appliqués aux programmes informatiques.....	19
Section 2 : Des exceptions affectant différemment la protection selon le système.....	21
Paragraphe 1 : Des limites pouvant affectant sensible la protection européenne.....	21
A) L'éternelle problématique de l'idée et de l'expression en droit de l'UE.....	21
B) Des exceptions pesantes : droit d'analyse et interopérabilité.	23
Paragraphe 2 : Des limites à l'étendue de la protection par le droit d'auteur étasunien.....	24
A) L'éternelle distinction de l'idée et l'expression en droit d'auteur étasunien.....	24
B) Une exception pesante : la fair use doctrine.....	26
Partie 2 : Les modes alternatifs de protection des programmes d'ordinateur dans les législations étasuniennes et européennes.....	29
Chapitre 1 : L'utilisation différenciée des modes alternatifs de protection des programmes d'ordinateur.....	29
Section 1 : Des régimes juridiques venant variablement en aide selon le territoire de protection.....	29
Paragraphe 1 : Un autre pan de la propriété industrielle européenne et américaine au secours des développeurs.....	29
A) Droit des marques et des dessins et modèles appliqués aux logiciels.....	29
1) Le droit des marques.....	30
2) Le droit des dessins et modèles.....	31
B) La concurrence déloyale appliquée aux logiciels.....	31
Paragraphe 2 : Protection des logiciels par le régime des secrets d'affaires.....	33
A) Le droit européen des secrets d'affaires appliqué aux logiciels.....	33
B) Le droit étasunien des secrets d'affaires appliqué aux logiciels.....	34
Section 2 : Des outils permettant de répondre aux difficultés : l'avantage étasunien ?.....	36
Paragraphe 1 : Les nouveaux outils juridiques et améliorations techniques assurant plus de sécurité.....	36
A) Le Patent Pledge.....	36
B) Mesures Techniques de Protection et leur encadrement juridique.	38
Paragraphe 2 : L'utilisation extensive de mécanismes connus assurant un accès limité au programme.....	40

A) Le contrat de service : la barrière technique du mode de distribution.....	40
B) La licence : la restriction par des clauses contractuelles.....	41
Chapitre 2 : L'utilisation unifiée des concepts alternatifs de protection des programmes d'ordinateur.....	43
Section 1 : Le logiciel libre : une réelle absence de protection ?	43
Paragraphe 1 : La liberté contrôlée par des licences.....	43
A) Une liberté désirée.....	43
B) Une liberté encadrée.....	44
Paragraphe 2 : Une liberté avantageuse mais pas sans risque.....	45
A) Les innombrables atouts du logiciel libre.	46
B) Les problématiques risques d'insécurité juridique.....	47
Section 2 : La protection sui generis : une alternative oubliée ?.....	48
Paragraphe 1 : Un grand débat éteint depuis les années 2000.....	48
Paragraphe 2 : ...Ressurgissant fréquemment dans les récentes analyses.....	49
Conclusion.....	51
Annexes.....	53
Table des sigles.....	54
Bibliographie.....	54
Table des matières.....	60