

## Part III

### Cadre légal: licences

Roberto Di Cosmo

Cadre légal  
Free Software/Open Source  
Analyse  
Autres licences dans d'autres domaines

Logiciel libre, une introduction

Le Droit d'Auteur  
20 ans d'évolution du droit  
DADVSI  
La notion de licence de logiciel  
Reverse Engineering

- 1 Cadre légal
  - Le Droit d'Auteur
  - 20 ans d'évolution du droit
  - DADVSI
  - La notion de licence de logiciel
  - Reverse Engineering
- 2 Free Software/Open Source
  - The Free Software Foundation
  - The Open Source Definition
- 3 Analyse
  - Libre versus Propriétaire
- 4 Autres licences dans d'autres domaines
  - Creative Commons

## “Propriété Intellectuelle”

Il y a quatre instruments majeurs de “protection” de la “propriété intellectuelle” :

**le secret industriel** : on ne dévoile pas les secrets de fabrication

**la marque** : créer une image publique reconnue du produit

**le droit d'auteur** : vendre des “licences d'utilisation” du logiciel, qui ne peut être copié tel quel

**le brevet** : obtenir un monopole garanti par l'Etat sur des “inventions” ou “techniques” qui ne s'identifient pas à un produit

L'ordre de cette présentation n'est pas dû au hasard.

Une lecture intéressante:

<http://www-db.stanford.edu/~ullman/pub/focs00.html> le discours de J. Ullman “Ordinary skills in the art” pour son Knuth Award de 2000.

## L'expérience de 30 ans d'Informatique

**secret industriel** : usage massif dans l'électronique et les jeux

**droit des marques** : très rentable, compatible avec l'Open Source<sup>24</sup>

**droit d'auteur** : usage massif pour le logiciel

**brevet** : très utilisé pour les médicaments, les voitures, les avions, usage presque uniquement détourné dans le monde du logiciel

Developpement de l'informatique = pas de brevets logiciels.

*“If people had understood how patents would be granted when most of today's ideas were invented and had taken out patents, the industry would be at a complete standstill today...A future start-up with no patents of its own will be forced to pay whatever price the giants choose to impose.”*

## Un peu d'histoire et de textes de référence

- 1662 : Licensing Act
- 1709 : Statute of Anne (auteur et durée limitée)
- 1791/93 : Droit d'auteur (et des brevets) introduit en France
- 1886 : Convention de Berne (reconnaissance internationale, sans dépôt préalable, exclusivité traduction) <sup>25</sup>
- 1961 Convention de Rome<sup>26</sup>

---

<sup>25</sup> “compl. Paris (1896), rev Berlin (1908), compl. Berne (1914), rev Rome (1928), Brussels (1948), Stockholm (1967), Paris (1971), amend. 1979 (Berne Union)”

<sup>26</sup> pour la protection des artistes interprètes ou exécutants, des producteurs de phonogrammes et des organismes de radiodiffusion

## Quelques organisations

**WIPO/OMPI** : association privée sans financement, affiliée aux UN

**WTO/OMC** : Organisation Mondiale du Commerce

**INPI** : institut français pour la Propriété Intellectuelle

**EPO** : Office européen des brevets

**USPTO** : US Patent Office

## France: code de la propriété intellectuelle

Accessible sur <http://www.legifrance.gouv.fr>.

Il établit un droit de propriété exclusif; Art. L111-1:

*L'auteur d'une oeuvre de l'esprit jouit sur cette oeuvre, du seul fait de sa création, d'un droit de propriété incorporelle exclusif et opposable à tous.*

## France: code de la propriété intellectuelle

deux facettes

- droit moral (L121-1... L121-9):

*L'auteur jouit du droit au respect de son nom, de sa qualité et de son oeuvre.*

*Ce droit est attaché à sa personne.*

*Il est perpétuel, inaliénable et imprescriptible. (Art. L121-1)*

- droit patrimonial (L122-1... L122-12):

*Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite. (Art. L122-4)*

## Le code de la propriété intellectuelle et les logiciels

Les logiciels relèvent de ce droit

- Art. L112-2:

*Sont considérés notamment comme oeuvres de l'esprit au sens du présent code :*

...

*13 Les logiciels, y compris le matériel de conception préparatoire ;*

...

## Le code de la propriété intellectuelle et les logiciels

Avec des particularités: *Art 113-9*

*Sauf dispositions statutaires ou stipulations contraires, les droits patrimoniaux sur les logiciels et leur documentation créés par un ou plusieurs employés dans l'exercice de leurs fonctions ou d'après les instructions de leur employeur sont dévolus à l'employeur qui est seul habilité à les exercer.*

## Le cas des stagiaires

Pourtant, le logiciel que l'un de vous (Master II à Paris 7) développe entièrement dans le cadre d'un stage *lui appartient!*

La raison est que si:

- Le stage s'inscrit dans le cursus pour l'obtention d'un diplôme ou d'un titre.
- Le stagiaire reste sous la responsabilité juridique de l'établissement d'enseignement bien que hors de celui-ci.
- Il y a une convention de stage.
- Il n'y a pas de rémunération.

Alors le stagiaire n'est pas considéré comme un salarié.

Voir: circulaire ministérielle du 30/10/1959, et celles n° 22 du 26/03/1970 et n° 86.065 du 13/02/1986.

## Le code de la propriété intellectuelle et les logiciels

et aussi

- Art. L121-7

*Sauf stipulation contraire plus favorable à l'auteur d'un logiciel, celui-ci ne peut :*

- 1 S'opposer à la modification du logiciel par le cessionnaire des droits mentionnés au 2° de l'article L. 122-6, lorsqu'elle n'est préjudiciable ni à son honneur ni à sa réputation ;*
- 2 Exercer son droit de repentir ou de retrait.*

## Le code de la propriété intellectuelle: plusieurs auteurs

- oeuvre de collaboration: plusieurs auteurs identifiés
- oeuvre collective: une seule entité édite, publie, diffuse une oeuvre où les contributions individuelles ne sont plus clairement identifiables

Durée des droits (France):

- individuelle: 70 ans après la mort
- collaboration: 70 ans après la mort du dernier auteur
- collective: 70 ans après la publication

## Plus longtemps, plus strict

- 1993 Directive EU sur l'armonisation du droit d'auteur, durée = 70 ans, rétroactive (de l'Allemagne)
- 1994 Accords TRIPS (Trade-Related Aspects of Intellectual Property Rights)
- 1996 Traité de l'OMPI sur le droit d'auteur (WCT):

27

- 1998 "Mickey Mouse" Copyright Term Extension Act (70, ou 75 à 95)

<sup>27</sup>Le traité fait obligation aux parties contractantes de prévoir des sanctions juridiques contre la neutralisation des mesures techniques (cryptage, par exemple) qui sont mises en oeuvre par les auteurs dans le cadre de l'exercice de leurs droits et contre la suppression ou la modification d'informations, comme certains éléments permettant d'identifier l'oeuvre ou son auteur, qui sont nécessaires pour la gestion (concession de licences, perception et répartition des redevances) des droits des auteurs (information sur le régime des droits)

## Les implémentations du traité de l'OMPI

1998 Digital Millennium Copyright Act

2001 Chiarigliano vs Felten,

2001 Skylarov

(<http://www.freesklyarov.org/>)

2001 deCSS

2004 LCEN: [http://ameli.senat.fr/publication\\_pl/2003-2004/144.html](http://ameli.senat.fr/publication_pl/2003-2004/144.html), ou Legifrance

2006 DADVSI: (le Août), comme transposition de EUCD  
(voir <http://eucd.info>)

## Problèmes posés par DADVSI

Deux registres d'inquiétude :

- sur la liberté des logiciels eux-mêmes;
- sur la place des plateformes libres dans l'usage;

## Liberté des logiciels

### *Prévention du téléchargement illicite*

*Art. L. 336-1. - Lorsqu'un logiciel est principalement utilisé pour la mise à disposition illicite d'oeuvres [...], le président du tribunal de grande instance, statuant en référé, peut ordonner sous astreinte toutes mesures nécessaires à la protection de ce droit et conformes à l'état de l'art.*

*Les mesures ainsi ordonnées ne peuvent avoir pour effet de dénaturer les caractéristiques essentielles ou la destination initiale du logiciel. (sic!)*

N.B.: le compromis présidant à la directive 2001/29 excluait le caractère obligatoire pour les TPM.

## Problème

- l'auteur d'un Logiciel (libre ou pas) n'est pas maître de ses "usages"
- définition de la "mesure"
  - fermer le code? impossible pour le LL
  - adopter des techniques couvertes par brevets? les licences du LL, ainsi que les modèle économique des LL ne s'y accomodent pas, même en RAND

Risque d'exclusion pure et simple des Logiciels Libres, au cas par cas, jugement par jugement.

## Viabilité des plateformes libres

L'accès légal aux oeuvres protégées (ou l'usage légal de celles-ci) n'est pas assuré sur une plateforme libre:

- comment lire un DVD zoné sur un ordinateur GNU/Linux?

Risque de marginalisation du libre:

les DRM/TPM peuvent être un outil de restauration d'un contrôle propriétaire monopolistique.

## En droit français. . .

Ce qu'on appelle généralement *licence logicielle* est plus précisément un *contrat de mise à disposition de logiciels*.

**contrat** : accord privé entre des parties

**mise à disposition** : il ne s'agit pas d'une "vente" ou "cession de droits", mais une simple "mise à disposition" dans des conditions précisées par le contrat lui-même. Ces conditions sont limitées seulement par l'accord des parties et par le droit national applicable.

## Comment on valide un contrat?

Les droits nationaux diffèrent:

- signature manuscrite
- signature électronique
- ...

Pour les logiciels, on trouve souvent

- shrink-wrap: on accepte en ouvrant le cellophane
- click-through: on accepte en cliquant un bouton

N.B.: on peut le recuser, mais *rien* n'autorise l'utilisation du logiciel dans ce cas!

## Exemple: le cas de la detaxe Windows



Il n'est souvent pas possible de revendre la licence pre-installée!

## En pratique

Les limites qu'on trouve *dans les contrats propriétaires* portent sur

- l'interdiction de la copie
- des limitations d'usage (nombre de processeurs, machine, utilisateurs simultanés, etc.)
- l'interdiction de la décompilation (dans la limite du droit en vigueur, quand même)

## Directive EU de 1991

**1991** Directive EU sur les logiciels (durée = 50 ans)

### *Article 6 Decompilation*

*1. The authorization of the rightholder shall not be required where reproduction of the code and translation of its form within the meaning of Article 4 (a) and (b) are indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs, provided that the following conditions are met:*

*(a) these acts are performed by the licensee or by another person having a right to use a copy of a program, or on their behalf by a person authorized to do so;*

*(b) the information necessary to achieve interoperability has not previously been readily available to the persons referred to in subparagraph (a); and (c) these acts are confined to the parts of the original program which are necessary to achieve*

## Reverse Engineering dans la directive de 1991

2. *The provisions of paragraph 1 shall not permit the information obtained through its application:*
- (a) *to be used for goals other than to achieve the interoperability of the independently created computer program;*
  - (b) *to be given to others, except when necessary for the interoperability of the independently created computer program; or*
  - (c) *to be used for the development, production or marketing of a computer program substantially similar in its expression, or for any other act which infringes copyright.*
3. *In accordance with the provisions of the Berne Convention for the protection of Literary and Artistic Works, the provisions of this Article may not be interpreted in such a way as to allow its application to be used in a manner which unreasonably*

## Clean Room/Dirty Room reverse engineering

Pour produire un logiciel B concurrent du logiciel A, ce qui est interdit *par le droit d'auteur* est:

- la copie (de parties significatives) du code source de A
- la décompilation (de parties significatives) du code objet de A  
sauf si indispensable pour l'intéropérabilité

Par contre vous pouvez:

- observer le fonctionnement de A
  - réécrire *indépendemment* B avec les fonctionnalités de A
- A priori, clean room/dirty room sont équivalents à ce niveau,

M A I S . . .

## Un exemple concret

Le cas Softimage vs. Synx Relief [http://www.legalis.net/jurisprudence-decision.php3?id\\_article=1039](http://www.legalis.net/jurisprudence-decision.php3?id_article=1039) de 2003 montre comme on peut être en faute même si on respecte formellement le droit d'auteur.


## Les licences "libres": Open Source vs. Free Software

Le paysage est assez complexe:

libre :  Richard Stallman

copylefted : GPL/LGPL, etc.

non copylefted : BSD/MIT, etc.

open source :  Bruce Perenes/Eric Raymond  
Open Source Initiative (OSI): "California public benefit corporation", fondée en 1998.

## Free Software Definition (<http://www.gnu.org/philosophy/>)

L'expression "Logiciel libre" fait référence à la liberté pour les utilisateurs d'exécuter, de copier, de distribuer, d'étudier, de modifier et d'améliorer le logiciel. Plus précisément, elle fait référence à quatre types de liberté pour l'utilisateur du logiciel :

- La liberté d'exécuter le programme, pour tous les usages (liberté 0).
- La liberté d'étudier le fonctionnement du programme, et de l'adapter à vos besoins (liberté 1). Pour ceci l'accès au code source est une condition requise.
- La liberté de redistribuer des copies, donc d'aider votre voisin, (liberté 2).
- La liberté d'améliorer le programme et de publier vos améliorations, pour en faire profiter toute la communauté (liberté 3). Pour ceci l'accès au code source est une condition requise.

## Quelques remarques

- fort accent sur la *liberté* de l'utilisateur et du développeur
- pas d'antinomie avec le "commerce"

*"Logiciel libre" ne signifie pas "non commercial".  
Un logiciel libre doit être disponible pour un usage commercial, pour le développement commercial et la distribution commerciale.*

- les quatre libertés ne sont pas une "test suite" pour les licences (c'est voulu)

## Pas de test suite pour la "liberté"

<http://www.gnu.org/philosophy/free-sw.fr.html>

*Enfin, notez que les critères tels que ceux développés dans cette définition du logiciel libre demandent une réflexion sérieuse quant à leur interprétation. Pour décider si une licence de logiciel particulière est définie comme libre, nous la jugeons sur ces critères pour déterminer si elle convient à leur esprit tout comme à leur formulation précise. Si une licence inclut des restrictions inacceptables, nous la rejetons même si nous n'avons pas anticipé le problème dans ces critères.*

## 1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources.

The license shall not require a royalty or other fee for such sale.

*Rationale: By constraining the license to require free redistribution, we eliminate the temptation to throw away many long-term gains in order to make a few short-term sales dollars. If we didn't do this, there would be lots of pressure for cooperators to defect.*

## 2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form.

Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge.

The source code must be the preferred form in which a programmer would modify the program.

Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

*Rationale: We require access to un-obfuscated source code because you can't evolve programs without modifying them. Since our purpose is to make evolution easy, we require that modification be made easy.*

## 3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

*Rationale: The mere ability to read source isn't enough to support independent peer review and rapid evolutionary selection. For rapid evolution to happen, people need to be able to experiment with and redistribute modifications.*

## 4. Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

*Rationale: Encouraging lots of improvement is a good thing, but users have a right to know who is responsible for the software they are using. Authors and maintainers have reciprocal right to know what they're being asked to support and protect their reputations.*

*Accordingly, an open-source license must guarantee that source be readily available, but may require that it be distributed as pristine base sources plus patches. In this way, "unofficial" changes can be made available but readily distinguished from the base source.*

Roberto Di Cosmo

Logiciel libre, une introduction

## 5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

*Rationale: In order to get the maximum benefit from the process, the maximum diversity of persons and groups should be equally eligible to contribute to open sources. Therefore we forbid any open-source license from locking anybody out of the process.*

*Some countries, including the United States, have export restrictions for certain types of software. An OSD-conformant license may warn licensees of applicable restrictions and remind them that they are obliged to obey the law; however, it may not incorporate such restrictions itself.*

## 6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

*Rationale: The major intention of this clause is to prohibit license traps that prevent open source from being used commercially. We want commercial users to join our community, not feel excluded from it.*

## 7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

*Rationale: This clause is intended to forbid closing up software by indirect means such as requiring a non-disclosure agreement.*

## 8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

*Rationale: This clause forecloses yet another class of license traps.*

## 9. License Must Not Restrict Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

*Rationale: Distributors of open-source software have the right to make their own choices about their own software.*

*Yes, the GPL is conformant with this requirement. Software linked with GPLed libraries only inherits the GPL if it forms a single work, not any software with which they are merely distributed.*

## 10. License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface.

*Rationale: This provision is aimed specifically at licenses which require an explicit gesture of assent in order to establish a contract between licensor and licensee. Provisions mandating so-called "click-wrap" may conflict with important methods of software distribution such as FTP download, CD-ROM anthologies, and web mirroring; such provisions may also hinder code re-use. Conformant licenses must allow for the possibility that (a) redistribution of the software will take place over non-Web channels that do not support click-wrapping of the download, and that (b) the covered code (or re-used portions of covered code) may run in a non-GUI environment that cannot support popup dialogues.*

**libre** : quelques dizaines de licences, généralement très permissives, pour des centaines de milliers de logiciels. . . ; on vous laisse modifier, distribuer, vendre et utiliser le logiciel, sous des conditions pas toujours équivalentes, d'où le problème de compatibilité des licences, et l'importance d'éviter la prolifération des licences;

**propriétaire** : une licence par produit, souvent très restrictive, on ne peut pas faire grand chose avec ce logiciel à part l'exécuter, et encore. . .

## Possibilités

**propriétaire** : presque tout est interdit

**libre** : presque tout est permis

Le logiciel propriétaire n'est pas *plus simple* que le logiciel libre, il est *plus restrictif*.

## Quelques caractéristiques des licences libres

Ayant une "oeuvre" (logiciel, bout de code) libre A,

- on peut *toujours* réaliser une "oeuvre dérivée" B (on change le code de A, ou on incorpore dans B du code de A), si la licence de B le permet;
- mais le statut de l'"oeuvre dérivée" change selon la licence de A

## Impact sur les oeuvres dérivées

- A sous licence X ou BSD (version récente): presque aucune contrainte sur B
- A sous licence GPL: B doit être redistribué sous licence GPL (copyleft)
- A sous licence LGPL: selon les cas, B peut être redistribué sous une licence non LGPL, mais avec des contraintes techniques

## Obligations de disclosure

En aucun des cas susmentionnés on a l'obligation de diffuser les modifications *in house*, mais

- la QPL originale (Trolltech) comporte une obligation de disclosure *tout court*
- la Affero GPL comporte une obligation de disclosure envers les usagers du logiciels

# Problématique légale

Bien de confusion sur les licences libres vient de la technicité du logiciel: lequel des actes suivants produit une "oeuvre dérivée" de A?

**incorporation de sources** : le source de B contient du code source de A

**liaison statique** : B est lié (via compilation statique) avec le code de la librairie A

**liaison dynamique** : B est lié (via compilation dynamique) avec le code de la librairie A

Une lecture des licences sans expérience de programmation peut donner des résultats inattendus.

N.B.: tout cela n'est pas spécifique au Logiciel Libre.

# Matrice de compatibilité

## Matrice de compatibilité

[Licences compatibles ?](#)

	CeCill	GPL	LGPL	ZPL	ApacheSL	MPL	Artistic	MIT	BSD	QPL
<b>QPL</b>	P	NC	L(LGPL):QPL L(QPL):LGPL, sinon NC	P	P	L(QPL): P/ NC	QPL/P	QPL	3: QPL/ 4: QPL	Piege!
<b>BSD</b>	(3: CeCill / 4: NC), L(I): Original	3: GPL / 4: NC	3: I / 4: L(LGPL): BSD, sinon NC	3: ZPL ou P / 4: BSD	3: ASL / 4: Hybr	3: I ou P / 4: BSD ou P	(3: Artistic / 4: P ou Hybr), L(I) = Original	3: I / 4: BSD	BSD4B ou P	
<b>MIT</b>	CeCill/P	GPL	L(LGPL):I, sinon LGPL	ZPL ou P	ASL	MPL ou P	Artistic	X		
<b>Artistic</b>	NC/CeCill	GPL	L(LGPL):I, sinon LGPL	Artistic ou P	ASL	MPL ou P	X			
<b>MPL</b>	MPL/CeCill	NC	L(LGPL):MPL, sinon LGPL(MPL v1.1) ou NC (MPL v1)	P	ASL 1.0/1.1: NC, ASL 2.0: P	X				
<b>ApacheSL</b>	NC/P	NC	L(LGPL): ASL, sinon NC	ZPL + ASL 2.0 = ASL 2.0, ZPL + ASL 1.0/1.1 = P	X					
<b>ZPL</b>	CeCill/P	GPL	L(LGPL): ZPL, sinon LGPL	X						
<b>LGPL</b>	L(LGPL): CeCill, sinon GPL	GPL	X							
<b>GPL</b>	GPL/P	X								

### légende

- n/a : case non traitée
- le nom d'une licence : licence sous laquelle le programme résultant pourra être mis
- NC : les licences ne sont pas compatibles
- I : le résultat peut être mis indifféremment sous l'une ou l'autre des licences
- L(X) : on considère une librairie sous la licence X
- ? : deux groupes ne sont pas d'accord sur le résultat et en débattent
- Hybr : Licence Hybride reprenant les clauses de chaque licence
- P : Le programme est composé de morceaux de codes avec des licences différentes .

## Un cadre général pour les oeuvres (pas le logiciel)

En 2001, apparaît Creative Commons

(<http://creativecommons.org/about/history>).

Finalité: structurer les licences pour qui souhaite donner un certain degré de liberté à ses oeuvres.

La version originale (1.0) de la famille des licences est essentiellement obtenue par combinaison des restrictions suivantes:

**attribution** : obligation de mentionner l'auteur <http://creativecommons.org/licenses/by/1.0/>

[//creativecommons.org/licenses/by/1.0/](http://creativecommons.org/licenses/by/1.0/)

**non-commercial** : interdiction d'usage commercial <http://creativecommons.org/licenses/nc/1.0/>

[//creativecommons.org/licenses/nc/1.0/](http://creativecommons.org/licenses/nc/1.0/)

**no-derivs** : interdiction de faire des oeuvres dérivés <http://creativecommons.org/licenses/nd/1.0/>





















[//creativecommons.org/licenses/nd/1.0/](http://creativecommons.org/licenses/nd/1.0/)

**share-alike** : obligation de redistribuer sous la même

licence <http://creativecommons.org/licenses/by-sa/1.0/>

## Le mouvement Creative Commons

Pourtant, dans la version 1.0, il y a seulement 11 cas:

Attribution				
Attribution-NoDerivs				
Attribution-NoDerivs-NonCommercial				
Attribution-NonCommercial				
Attribution-NonCommercial-ShareAlike				
Attribution-ShareAlike				
NoDerivs				
NoDerivs-NonCommercial				
NonCommercial				
NonCommercial-ShareAlike				
ShareAlike				

Dans la version Française, seulement les 6 premières : pourquoi?