

Logiciels & Données : quelques éléments de Propriété Intellectuelle

Sylvain Karpf
Direction de la Valorisation de la
Recherche

Le droit de la Propriété Intellectuelle, c'est quoi ?

- Définit le ou les régimes juridiques applicables aux créations immatérielles (inventions, œuvres...)

Droit de la propriété littéraire et artistique

Droit d'auteur
(œuvres artistiques, livres ..., LOGICIEL)

Droit sui generis des bases de données

Droit de la propriété industrielle

Brevets

Marques

dessins et modèles

Alternative aux droits de PI: mise au secret

Dossier de savoir-faire

Liens avec les autres droits de propriété intellectuelle

- Un logiciel n'est pas protégeable en tant que tel par voie de brevet (en Europe)
- En revanche, une invention technique peut être implémentée dans un logiciel (dispositif, procédé...)
- Le nom d'un logiciel peut faire l'objet d'un dépôt de marque
- Les logiciels utilisent/intègrent souvent des données ou des bases de données, qui font elles-mêmes l'objet de dispositions légales spécifiques
 - Droit sui generis des bases de données
 - Droit d'auteur pour certaines données (images, photos, musiques, livres, ...)

Droits d'auteur, quels types de droits ?

Les droits moraux:

- appartiennent à l'auteur;
- ils sont perpétuels et inaliénables (on ne peut en priver l'auteur)
- En matière de logiciel, **ces droits se réduisent essentiellement au droit à la paternité**

Les droits patrimoniaux:

- Correspondent aux droits d'exploitation (modification, distribution...);
- Ont une durée de vie limitée dans la durée mais néanmoins longue (70 ans après première diffusion ou après la mort du dernier auteur, selon les cas)
- Pour un logiciel : Lorsque l'auteur est salarié et a réalisé le logiciel dans le cadre de ses fonctions, les droits patrimoniaux appartiennent à son employeur (loi du 10 mai 1994)
- **Depuis l'ordonnance du 15/12/2021 : lorsque l'auteur est un stagiaire gratifié, les droits patrimoniaux appartiennent à l'établissement qui accueille le stagiaire**

Logiciel : qu'est ce qui est protégeable au titre du droit d'auteur

- Le code source et le code objet
- Les spécifications détaillées et matériaux de conception
- L'architecture
- La documentation technique

En revanche, ne sont pas protégés à ce titre:

- Les algorithmes
- Les spécifications fonctionnelles (le cahier des charges)
- La documentation à destination des utilisateurs finaux (peut faire l'objet d'une protection en tant que telle, mais reste un objet distinct du logiciel)

Matérialisation dans le code source

Entête des fichiers source

Logiciel xxx version 1.0

Auteurs : yyy, zzz

© 2018-2020 Université de Lille & CNRS

Quel moyen de preuves pour sa titularité des droits ?

- Contrairement aux brevets, un logiciel est protégé par le droit d'auteur dès sa création, sans formalité particulière.
- Les preuves techniques (par exemple: copie du CVS ou Github à une certaine date) peuvent constituer des début de preuve mais ne sont pas considérés comme suffisants à eux seuls pour faire valoir ses droits
- Il existe également des preuves juridiques et certaines (dépôts probatoires), réalisées par des tiers de confiance (notaires, organisme réalisant des dépôts certifiés...), qui constituent à ce jour le meilleur moyen de faire valoir ses droits
 - APP (Agence de Protection des Programmes)
 - e-soleau (INPI)
 - ...
- Un tel dépôt permet de prendre date et facilite l'identification et la défense du logiciel (en cas de contrefaçon ou si l'Université est attaquée en contrefaçon)

Exploitation des droits patrimoniaux

- **Seuls les droits patrimoniaux peuvent faire l'objet d'une exploitation**
 - par le propriétaire du logiciel lui-même
 - par un tiers, auquel le propriétaire du logiciel concède une **licence d'exploitation**.
- Une licence d'exploitation est un contrat qui définit les modalités et les conditions dans lesquelles on autorise le licencié à exploiter le logiciel
- Dans la pratique, cela se traduit par l'existence d'une grande diversité de licences dans le monde du logiciel: **licences libres, licences propriétaires**

Différentes licences existantes

Licences libres

Licences à des
fins de test et
d'évaluation

Licence
d'exploitation à
des fins
industrielles et
commerciales

Avec accès
au code
source ou
pas

Licence
d'utilisation à des
fins de recherche

Incluant le
droit de
modifier/
adapter le
logiciel ou
pas

Gratuit ou
payant

Et beaucoup d'autres encore...

Licence Libre vs Licence non-libre à code ouvert

Pour être libre, une licence doit répondre à 4 critères, définis par la Free Software Foundation (FSF, fondée en 1985) :

- Liberté d'utiliser et reproduire le logiciel
- Liberté d'étudier librement le logiciel
- Liberté de modifier le logiciel
- Liberté de redistribuer le logiciel modifié ou non, a minima sous la licence d'origine

L'obligation de donner accès au code source est implicite (nécessaire pour étudier et modifier le logiciel)

Si une licence ne répond pas à un de ces critères, elle n'est pas libre au sens de la FSF

L'Open Source Initiative (OSI, fondée en 1998) a également défini une dizaine de critères, qui sont inclus de façon implicite dans les 4 libertés de la FSF.

Exemples de « fausses » licences libres à code ouvert

- Accès au code source, droit d'utiliser, modifier, **mais pas de droit de redistribuer**
- Accès au code source, droit d'utiliser, modifier, et droit de redistribuer **à des fins non-commerciales uniquement**
- Accès au source, droit de modifier, de redistribuer, **mais pas le droit d'utiliser à certaines fins (par exemple, à des fins militaires)**

Typologies Licences Libres

La typologie présentée ci-après est fondée sur le degrés de permissivité d'une licence libre en **termes de redistribution**, en fonction de la façon dont est réutilisé le code préexistant.

Type de licence	Philosophie du propriétaire	
Licence non permissive (copyleft) GNU GPL, GNU AGPL, CeCILL	« Si vous redistribuez mon code dans votre logiciel, c'est avec les mêmes règles du jeu pour les deux et en partageant avec la communauté »	« Si vous redistribuez mon code dans votre logiciel ou <u>si vous l'exploitez en mode SaaS</u> , c'est avec les mêmes règles du jeu pour les deux et en partageant avec la communauté »
Licence permissive en composition (weak copyleft) GNU LGPL, CeCILL-C, Mozilla Public License, Eclipse Public License	« Si vous redistribuez mon code (modifié ou pas) dans votre logiciel, il doit rester accessible et profiter à la communauté (« licensing back »); en revanche, vous choisissez vos règles du jeu pour le reste de votre logiciel »	
Licence permissive en composition et en dérivation (permissive) MIT, BSD, CeCILL-B, Apache	« Mon code, c'est cadeau, faites en ce que vous voulez! »	

Incidence de la réutilisation de code

- Réutilisation de composants logiciels externes
 - Open Source
 - Non Open Source
 - code propriétaire provenant d'un partenaire
 - code qui a l'air Open Source
- Pour chaque composant préexistant intégré dans un assemblage, il est important d'identifier et de qualifier la licence, pour vérifier son adéquation avec la **stratégie d'édition** (licence de redistribution/exploitation)
- Si problème de compatibilité de licence avec un composant
 - Demander au propriétaire de changer le licence
 - Trouver un composant équivalent avec une licence adaptée
 - Redévelopper le composant

Incidence du développement collaboratif

- Nécessite l'accord unanime de tous les copropriétaires sur le choix du mode d'exploitation du logiciel (y compris pour un changement de licence libre, par exemple)
- Copropriétaires nombreux = coûts de transaction et de coordination importants
- Copropriété parfois atypique à gérer (cas des logiciels avec des ayants droits à titre personnel, comme les stagiaires)

Accord entre les co-propriétaires indispensable pour choisir une licence.

En pratique

- Conventions de site, accord d'UMR, mandataire unique, etc ...
- Adapter les clauses PI/exploitation des contrats de recherche (prévoir par voie contractuelle de limiter les cas donnant lieu à copropriété)
- Mettre en place une politique d'acquisition des droits des ayants droits à titre personnel hébergés dans les équipes (~~par exemple, faire signer des contrats de cession de droits aux stagiaires~~)
- Faire signer **aux contributeurs externes** d'un projet open source des contrats permettant a minima d'intégrer les contributions dans le logiciel sous la licence de son choix
 - En pratique dans les projets Open Source : **Contributor Licence Agreements**
 - avec cession des droits
 - sans cession des droits

Et pour les données ?

Par exemple Licences type Creative Commons, inspirée des licences libres

CC BY : redistribution version initiale ou modifiée libre, obligation de citation

CC BY-SA : redistribution version initiale ou modifiée sous CC BY-SA uniquement, obligation de citation

CC BY-ND : modification/adaptations interdites de redistribution, obligation de citation

CC BY-NC : redistribution version initiale ou modifiée libre, obligation de citation, pas d'utilisation commerciale

CC BY-NC-SA : redistribution version initiale ou modifiée sous CC BY-NC-SA uniquement, pas d'utilisation commerciale, obligation de citation

CC BY-NC-ND : pas de modifications, pas d'utilisations commerciales, obligation de citation

Quelques Business Model utilisant de l'Open Source

Service

- Utilisation de logiciels Open Source, ou production et diffusion de logiciel Open Source (toutes licences possibles)
- Activité de service associée
 - distribution, packaging
 - Installation
 - Maintenance
 - intégration, développements spécifiques
 - etc ...

Dual Licensing

- Développement d'un logiciel
- Diffusion d'une version Open Source avec une licence non permissive type GPL
- Diffusion de la **même version** du logiciel sous licence propriétaire

Bien adapté aux bibliothèques

Exemples : MySQL, Qt

Open Core

- Diffusion du cœur du logiciel en licence libre (en général permissive)
 - diffusion d'une version plus complète sous licence propriétaire
 - diffusion de plug-in spécifiques à forte valeur ajoutée, sous licence propriétaire