

L'Open Source : du choix stratégique des licences ? Mars 2021

Rédigé par Mathias Robert et Charlotte Montaud

Le saviez-vous ? En 2019, le marché français de l'Open Source était le plus important en Europe, devant l'Allemagne et le Royaume-Uni, en termes de volume par rapport aux autres grands pays européens, selon une étude réalisée à l'initiative du CNLL, de Syntec Numérique et de Pôle Systematic Paris-Region, Teknowlogy Group.

Cette étude prévoyait également que dans près de 80 % des entreprises, l'utilisation de l'open source allait augmenter dans les deux ans à venir.

Puis, la crise sanitaire est survenue en 2020, entraînant une accélération brutale et parfois forcée de la transformation digitale des entreprises.

De manière incontournable, la transformation numérique conduit progressivement les différentes entreprises à se tourner vers les logiciels dits libres ou open source, même lorsque ces acteurs éditent également des logiciels commerciaux ou propriétaires.

Par le passé, le monde du libre ou de l'open source s'opposait culturellement à celui du logiciel propriétaire : au copyright, s'opposait le « copyleft ». Cette opposition s'est peu à peu atténuée avec le temps. On citera par exemple le cas de la société Microsoft, pourtant farouchement opposée initialement au logiciel libre, qui est ensuite devenue l'un des plus grands contributeurs de la communauté open source. Force est de constater qu'aujourd'hui, la majorité des logiciels développés comporte des composants open source.

Ce regain d'attrait s'explique par le besoin d'interopérabilité, en particulier avec l'avènement de l'Internet des Objets (IoT), par la virtualisation des technologies et par la nécessité de mutualiser des investissements et d'en assurer la pérennité.

Selon l'étude précitée, pour 9 entreprises sur 10, le logiciel libre est incontournable ou préférable lorsqu'il est adopté pour l'intelligence artificielle, le data management et la blockchain.

Logiciels open source et logiciels libres

Les logiciels open source permettent d'accéder aux codes sources. Développés par des programmeurs qui travaillent en mode collaboratif, ils ont pour avantage d'être réactualisés et enrichis fréquemment par la communauté.

En conséquence, lors du développement d'un logiciel, les développeurs sont tentés d'utiliser du code open source existant, déjà publié, évalué et amélioré par d'autres.

Une telle pratique permet, au-delà de la mutualisation des compétences de la communauté, de maximiser le gain de temps et d'accélérer les innovations.

Le code existant peut être réutilisé à l'identique ou être modifié pour l'adapter aux besoins spécifiques.

Par ailleurs, le code existant peut être intégré à un code plus large ou être utilisé sous la forme d'une bibliothèque. On rappelle, qu'en informatique, une bibliothèque est un ensemble de fonctions mises

à disposition et regroupées par exemple par appartenance à un même domaine conceptuel (traitement mathématique, interfaces graphiques, traitement d'image, ...).

La reprise du code d'un logiciel libre ou open source, malgré la qualification de « libre » ou « open », mérite que l'on s'attarde sur l'acception qu'on donne à ces termes dans le domaine informatique.

Un logiciel est qualifié de libre lorsque que l'utilisateur est libre l'exécuter, l'étudier, de le modifier et de le distribuer à l'identique ou après modification. Un logiciel open source repose également sur ces principes mais doit répondre à des règles supplémentaires, dont des règles relatives aux licences.

La distribution d'un logiciel libre ou open source nécessite que son code source soit partagé, ce qui permet en particulier un audit du code afin de s'assurer que celui-ci ne contient pas de composant malveillant.

Ces principes de l'open source ont été transposés aux données (open data) et aux dispositifs (open hardware).

Les libertés citées précédemment et accordées lors de la réutilisation du code d'un logiciel libre ou open source sont des libertés conditionnelles. Il existe, en effet, des modalités d'exploitation encadrées par des licences, plus ou moins permissives. Ces licences définissent des droits, mais également des obligations.

Les différents types de licences Open Source

Plusieurs types de licences open source se sont imposées avec le temps, que l'on peut classer en trois catégories :

a) les licences dites à copyleft fort : les redistributions du logiciel (modifié ou non) et de tous les composants ajoutés ne peuvent se faire que sous la licence initiale. Dans le cadre de la réutilisation de code sous licence à copyleft fort (par exemple sous licence GNU GPL), tous les composants développés dans le cadre du nouveau logiciel et associés avec le code sous licence copyleft fort devront être distribués avec cette même licence. En d'autres termes, inclure des composants sous licence à copyleft fort n'est pas judicieux s'il est envisagé de distribuer le nouveau logiciel par l'intermédiaire d'une licence fermée, c'est-à-dire commerciale ou propriétaire.

b) les licences dites à copyleft faible : les redistributions du logiciel (modifié ou non) se font sous la licence initiale mais de nouveaux composants peuvent être ajoutés sous d'autres licences. Ces licences (par exemple la licence LGPL) sont notamment utilisées dans le cadre de la distribution de bibliothèques.

Ce type de licence autorise ainsi le développeur à distribuer le code qu'il a lui-même développé pour son logiciel sous tout type de licence (notamment une licence commerciale ou propriétaire), mais le code réutilisé restera sous sa licence open source d'origine avec les contraintes qui en découlent.

c) les licences dites sans copyleft : les redistributions du logiciel (modifié ou non) peuvent se faire sous n'importe quelle forme, y compris sous la forme d'un logiciel propriétaire. Ce type de licence (par exemple la licence MIT ou la licence BSD) autorise ainsi la distribution de l'intégralité du code nouvellement développé sous n'importe quel type de licence (notamment une licence commerciale ou propriétaire).

Ces licences sont construites sur des schémas différents et la première difficulté peut consister à identifier les composants logiciels libres et les différentes licences qui les régissent, lesquelles peuvent contenir des dispositions contradictoires ou incompatibles.

En cas de non-respect d'une licence open source, le risque de se voir assigner devant un tribunal est réel, l'autorité judiciaire pouvant alors sanctionner une telle absence de conformité aux conditions de la licence. Il existe d'ailleurs des décisions françaises venues sanctionner une telle imprudence.

On comprend dès lors qu'il est important de maîtriser les conséquences juridiques liées à la conception, au développement et à la distribution de tels logiciels. Cette situation exige une collaboration étroite entre des équipes techniques et juridiques afin de répondre aux besoins de sécurité, de traçabilité et de pérennité.

Une telle collaboration nécessite notamment la tenue, par l'équipe technique, d'un inventaire listant les composants logiciels utilisés ainsi que les licences associées, cet inventaire précisant par ailleurs les interactions entre ces composants afin d'identifier leurs liens de dépendance. L'équipe juridique s'appuiera ensuite sur l'inventaire afin d'établir la compatibilité entre les différentes licences des composants identifiés et évaluer la conformité du mode de distribution envisagé. Le cas échéant, des actions correctives pourront éventuellement être envisagées, telles que le remplacement de certains composants ou la redéfinition de la licence envisagée pour la distribution

Comparatif des licences open source : attention aux pièges, jdn

mars 2022

Entre Apache 2.0, BSD, GNU GPL, MIT, les licences libres intègrent un nombre plus ou moins important de droits et d'obligations : permissivité, copyleft fort ou faible, compatibilité inférieure ou supérieure...

Attention à la fausse liberté qui se cache derrière le concept d'open source. Les licences open source n'ont de libre que le nom et il convient de se pencher attentivement sur leur contenu pour ne pas commettre d'impair. Si les pionniers du libre étaient opposés, par principe, à toute forme d'action judiciaire, les esprits ont depuis évolué. Il y a quelques années, quatorze entreprises, dont Samsung, étaient poursuivies pour violation de la licence GNU-GPL par le Software Freedom Law Center. Plus récemment, Free a été assigné en justice pour le même motif.

D'autres risques pèsent sur les entreprises qui s'affranchiraient du contenu des licences. "Les membres des communautés open source sont très attachés aux règles du libre et n'hésitent pas dénoncer les entreprises qui les enfreignent", avance Géraldine Salord, avocat au Barreau de Paris et associé fondateur de Metalaw Avocats Associés. Au-delà de cette atteinte à la réputation, ils peuvent aussi rendre public le code produit estimant que celui-ci relève de l'open source. Pour rajouter à la complexité, il existe des milliers de licences open source et il s'en crée presque tous les jours, aux côtés des licences les plus connues des familles GNU et OSI.

Déterminer l'élément déclencheur

Pour Géraldine Salord, la première question à se poser avant de choisir une licence porte sur la finalité de la solution open source. "Les effets contraignants de la licence ne se déclenchent que lors de certaines actions. Cet élément déclencheur est généralement l'exploitation commerciale du code. Une entreprise pourra modifier le code pour un usage interne sans que cela porte à conséquence." Ainsi, un laboratoire qui a une finalité de recherche ne provoquera pas, pour la plupart des licences, d'élément déclencheur.

En revanche, une entreprise privée peut vouloir redistribuer commercialement tout ou partie du code. "Le simple fait d'intégrer de l'open source comme le ferait un constructeur automobile dans un système embarqué qui est ensuite vendu peut constituer un élément déclencheur", illustre Géraldine Salord. L'utilisation d'un site web ou d'un modèle d'IA à usage externe peut aussi constituer un élément d'activation. Il en va de même d'un transfert technologique entre un sous-traitant qui fait appel à une brique open source et un donneur d'ordre.

Une stratégie de licence "by design"

Au regard de ces contraintes, une entreprise doit définir, en amont d'un projet, le type de licences avec lesquelles elle accepte de travailler et donc d'en restreindre volontairement le nombre. A cet effet, le cabinet Metalaw Avocats Associés propose des tableaux de correspondances pour indiquer à une entreprise ce qu'elle peut faire ou non en combinant des licences.

L'absence de stratégie de licence pourrait aller jusqu'à remettre en cause un projet. "Un projet open source peut comporter une dizaine ou une vingtaine de briques open source", observe Géraldine Salord. "Les ingénieurs vont les utiliser sans se soucier de la comptabilité. A la fin, l'application ne pourra être exploitée ou bien il faudra la reprendre et remplacer certains composants."

Par ailleurs, une entreprise doit se poser la question de ce qui constitue véritablement un avantage concurrentiel à ses yeux. "Il est recommandé de cloisonner les différentes parties d'un logiciel pour les rendre hermétiques entre elles, conseille Géraldine Salord. "Cela évite la contamination et donnera accès à des briques stratégiques." Tout un travail d'architecture à entreprendre, là aussi, en amont.

Travail : Logiciels libres et droit

Vous devez produire un document constitué des éléments ci-dessous :

- Les caractéristiques des logiciels libres
- L'évolution économique récente du secteur des logiciels libres
- Les problèmes juridiques posés par la méconnaissance des règles en matière de licences libres et leurs conséquences récentes. Comment éviter ces problèmes ?

Application 1 – Licence MIT Le texte complet de la licence MIT est le suivant : MIT License Copyright (c) [year] [fullname]

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1. A` qui la licence accorde-t-elle des droits ?
2. Quelles sont les droits accordés par la licence ?
3. A` quelle condition puis-je utiliser un code sous licence MIT dans un logiciel distribué » sous forme d'exécutable binaire ? Y a-t-il une différence si ce logiciel est un logiciel propriétaire ou un logiciel libre ?
4. A` quoi sert la partie entièrement en lettres capitales ?

Application 2 – Comprendre le copyleft Le texte de la licence GNU GPL 3.0 commence ainsi :

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works. 1

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

Par ailleurs, la section 9 de la licence est ainsi écrite :

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

1. Comment le préambule de la GPL justifie-t-il l'imposition de restrictions sur la manière dont un logiciel libre sous cette licence peut être distribué ?

2. Quel est l'outil juridique utilisé pour donner des droits à l'utilisateur d'un logiciel libre ?
 3. Pour quels motifs pourrait être poursuivi une personne qui ne respecterait pas la licence ?
-

Application 3 – Distribution de logiciel sous licence GPL On considère un logiciel sous licence GNU GPL 3.0.

Dans chacun des cas suivants, doit-on considérer que le logiciel a été distribué ? Quelles sont les obligations imposées par la licence ?

1. J'ai téléchargé un binaire du logiciel. Je donne une copie de ce binaire à un ami.
2. Je modifie le logiciel et je l'installe sur tous les ordinateurs de mon entreprise.
3. J'installe le logiciel sur un serveur web, et je le rends accessible à mes clients via une interface web. J'ai développé moi-même le code JavaScript de l'interface web.
4. J'installe le logiciel sur un objet que je vends à mes clients. L'objet se comporte comme un serveur web et les clients ont accès au logiciel via une interface web. J'ai développé moi-même le code JavaScript de l'interface web.