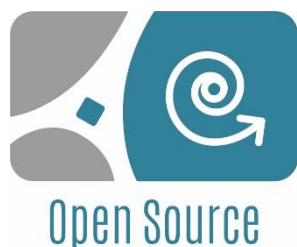


Synthèse de l'atelier « Réponses d'expert : Appréhender les enjeux juridiques de l'Open Source » du 8 juin 2022



Sommaire

1 Description de l'atelier	3
1.1 CONTEXTE	3
1.2 OBJECTIFS	3
1.3 DERoule ET FORMAT PARTICIPATIF	4
2 Synthèses des séances de travail en groupe.....	5
2.1 GROUPE « LICENCES OPEN SOURCE »	5
Une notion clef abordée : les types de licences (copyright, permissive, copyleft, droits moraux)	5
Questions fréquentes et éléments de réponse :	5
Liste des outils et ressources partagés :	8
Outils d'aide à la vérification de la compatibilité/conformité de licences	8
Ressources : Les bonnes pratiques du monde de l'Open Source et en matière de licences	8
2.2 GROUPE 2 « VALORISATION ET OPEN SOURCE »	8
Questions fréquentes & éléments de réponses	8
Liste des ressources partagées :	13
3 Pistes d'approfondissement.....	14
4 Annexes.....	15
4.1 PERSONNES AYANT CONTRIBUE	15
4.2 LISTE DES QUESTIONNEMENTS CONSIGNES SUR LES SUPPORTS DES ATELIERS	16
Tableau blanc Excalidraw du groupe « Licences »	16
Tableau blanc Excalidraw du groupe « Valorisation »	16

Version	Date	Auteur.e.s	Commentaires
1	15/07/2022	Benjamin Jean, Vincent Bachelet, Célya Gruson- Daniel, Maya Anderson-González, Romane Leban- Mathieu	Version livrée au Hub Open Source de Systematic Paris-Région
2	17/08/2022	Benjamin Jean, Vincent Bachelet, Célya Gruson- Daniel, Maya Anderson-González, Romane Leban- Mathieu	Version avec URL mis à jour

1 | Description de l'atelier

1.1 Contexte

Le pôle de compétitivité Systematic Paris-Region dispose d'un Hub Open Source constitué de membres issus de plusieurs secteurs.

Avec plus d'une centaine d'entreprises membres, à la fois des grands groupes, des PME et des centres de recherche, le sujet des instruments juridiques suscite un intérêt croissant directement corrélié à l'adoption généralisée de l'Open Source dans tous les secteurs. Cet intérêt est d'autant plus renforcé qu'il a été constaté un manque de compréhension et de maîtrise des spécificités juridiques et économiques du modèle Open Source.

Afin de répondre aux besoins des membres du Hub Open Source, le pôle organise depuis 2021 régulièrement des ateliers de sensibilisation et d'accompagnement sur des problématiques liées aux stratégies et aux modèles économiques Open Source. En 2022, de nouveaux ateliers sont programmés afin d'aborder plus spécifiquement les aspects juridiques des stratégies et modèles économiques Open Source.

En partenariat avec le cabinet inno³, le Hub Open Source a proposé un atelier au design participatif et sur mesure qui avait pour objectif de **partager une expertise transverse relative aux enjeux juridiques de l'Open Source** (licences, FOSS exceptions, Contributor License Agreement, DCO, etc.), **des éléments de contexte, des cas d'usage, mais aussi des outils et des ressources**.

Couvrant un périmètre relativement large, les thématiques abordées ont été définies en s'appuyant sur les besoins remontés par les participant.e.s tels que :

- L'état de l'art et analyse du cadre juridique afférent (notamment ses incidences dans le financement publics tels que les appels à projets ou les marchés publics)
- La présentation d'un panorama des licences Open Source avec un focus sur les dernières évolutions et les enjeux de compatibilité sous-jacents ;
- La compréhension des standards et bonnes pratiques en matières de conformité Open Source au sein des organisations (notamment la norme Open Chain, le format d'échange d'information Software Package Data Exchange (SPDX), le CLA Assistant et autres outils associés)
- Le rôle et le fonctionnement des principales fondations Open Source dans la structuration et le soutien des initiatives autour de l'Open Source (notamment Fondation Linux et Fondation Eclipse)

1.2 Objectifs

- Apporter une expertise pour aider à la connaissance et compréhension des éléments juridiques et des standardisations actuelles en matière d'Open Source ;
- Présenter des éléments de contexte et des ressources afin d'éclairer les enjeux associés aux principales problématiques non techniques rencontrées par les membres du Hub Open Source (récoltées par le biais d'un questionnaire diffusé en amont) ;

- Échanger avec les participants de l'atelier pour récolter et documenter leurs expériences spécifiques en matière d'Open Source ;
- Proposer des pistes de solutions pragmatiques ou théoriques aux questions spécifiques posées sur place, le cas échéant.

1.3 Déroulé et format participatif

L'atelier s'est déroulé en ligne sur la plateforme Big Blue Button et a duré un peu plus de 3 heures.

En première partie d'atelier, Benjamin Jean, juriste spécialisé en propriété intellectuelle et président du cabinet de conseil inno³, a effectué une présentation des principales préoccupations juridiques Open Source. Alimentée par les retours des participants sondés en amont de l'atelier, la présentation est revenue sur les fondamentaux des licences libres, leurs caractéristiques et le cadre de leur application, et a également abordé la notion de valorisation en lien avec l'Open Source, et la gestion de modèles Open Source au sein d'organisations.

Lors d'une seconde phase, les personnes présentes sont retrouvées en groupes de travail pour échanger de manière plus approfondie et pour amorcer une première base de documentation commune sur les problématiques liées aux thématiques « Licences Open Source » et « Valorisation et Open Source ».

2 | Synthèses des séances de travail en groupe

Les synthèses présentées à la suite attestent du travail collaboratif amorcé lors des séances de travail en groupe, puis retravaillé à l'aune des problématiques évoquées au cours de l'atelier.

Les questions posées, les éléments de réponses apportés ainsi que les ressources partagées constituent une base de documentation commune présentée sous forme de questions fréquentes (FAQ).

2.1 Groupe « Licences Open Source »

Une notion clef abordée : les types de licences (copyright, permissive, copyleft, droits moraux)

- Licence propriétaire : désigne toute licence pour laquelle le ou les titulaires de droits conserve un monopole sur leur création)
→ le droit d'auteur (ou copyright) protège par défaut l'auteur : tout ce qui n'est pas cédé expressément est « réservé » par son auteur.
- Licences libres (ou licences Open Source) : désigne toute licence conforme à la définition de l'[Open Source Definition](#) qui assure à l'utilisateur du logiciel les libertés d'utiliser, modifier et redistribuer le logiciel.
→ Il existe deux modèles qui cohabitent :
 - Les licences copyleft (à réciprocité) : qui imposent le repartage du logiciel selon les mêmes termes (copyleft faible ou étendue réduite : exemple de la LGPL ou de la MPL) voire qui imposent aussi le repartage sous la même licence de toute amélioration voire de tout logiciel basé (copyleft fort ou étendue forte : exemple de la GNU GPL)
 - Les licences permissives qui permettent la redistribution du code sous d'autres licences sous réserve du maintien de certaines obligations (par exemple, les licences Apache, BSD-3-Clause, MIT, etc.)

Ressources générales :

- Le [MOOC « Apprendre l'Open Source »](#) : composé de 31 vidéos diffusées sous licence libre (CC BY-SA 4.0), ce MOOC est coproduit par l'Open Source School et la Société Générale.
- [Le Guide Open Source](#) créés et impulsé par GitHub, avec l'aide de membres de la communauté externes.
- Le projet [Veni, Vidi, Libri](#) - à l'origine des guides [Comprendre les licences libres](#) et [Guide de l'auteur libre](#).
- L'ouvrage « [Option Libre, du bon usage des licences libres](#) » de Benjamin Jean.

Questions fréquentes et éléments de réponse :

Comment faire si un composant de mon logiciel change de licence au cours du projet ?

La licence est une offre de contracter que le titulaire de droit rattache à une version de son logiciel. Ainsi, tout changement de licence ne vous affecte pas a priori dès lors que vous respectez la licence sous laquelle cette copie du logiciel vous a été transmise. C'est néanmoins un élément de vigilance que vous

devez surveiller lorsque vous faites la mise à jour de vos dépendances car la montée de version peut entraîner un changement de licence.

Ressource :

- Ce cas de figure est traité dans l'[Open Source Guide : Le côté légal de l'Open Source](#).

Quelle compatibilité entre différentes licences au sein d'un même projet ?

La situation de compatibilité (ou d'incompatibilité) désigne une situation pour laquelle plusieurs licences Open Source imposent des obligations contradictoires pour un même logiciel. Afin de gérer cette question au sein d'un projet, il est nécessaire de réaliser un travail d'audit de code qui s'appuie sur une vision globale de l'architecture logicielle et une vision exhaustive de l'ensemble des composants Open Source tiers utilisés, qui figureront dans un inventaire global dénommé *Bill of Materials*.

Bon à savoir : La licence GPL est la plus restrictive mais la GPLv3 est néanmoins compatible avec la licence Apache 2.0.

Ressource :

- Pour plus d'informations, vous pouvez consulter la [page « Compatibilités de licences » du projet Veni, Vidi, Libri](#).

Quelles limites existent pour les licences permissives ?

Les licences permissives n'imposent pas le maintien de la licence sur le logiciel initial ni même l'extension sur ses évolutions, mais elles peuvent néanmoins contenir des obligations relativement engageantes en matière de citation dans la documentation ou encore dans toute communication relative au logiciel, ou encore en matière de licences de brevets (telle la clause *patent peace* de la licence Apache 2.0).

Ressource :

- Le site web du système d'exploitation GNU contient une page qui explique les [choix et utilisations des licences libres](#).

Y a-t-il des limites à l'obligation de réciprocité (copyleft) prévue par les licences libres ?

La clause copyleft peut avoir des effets plus ou moins étendus en fonction de sa formulation. Par exemple, la licence MPL 2.0 aura des effets limités aux seuls fichiers modifiés, la licence GPL-3.0 s'étendra à « toute application basée sur le logiciel » là où la licence LGPL-2.1 ne s'étendra pas aux applications qui utilisent seulement le logiciel « en tant que bibliothèque ».

Bon à savoir : les licences de type copyleft n'empêchent pas la commercialisation d'applications ou de services. Néanmoins, une telle commercialisation implique généralement la distribution gratuite de tout ou partie du code source associé, ce qui diminue grandement les voies de valorisation.

Comment savoir quelles sont les licences libres et Open Source à respecter dans mon logiciel ?

La liste des composants Open Source tiers est généralement un inventaire qui s'appuie à la fois sur une démarche automatisée et outillée (par exemple en s'appuyant sur les métadonnées collectées par les outils de gestion de dépendances) et sur une démarche de validation manuelle par l'équipe en charge du développement. Ce travail alimente un inventaire (software Bill of Materials) qui devra être mis à jour tout le long de la vie du projet.

Ressources :

- Des solutions Open Source existent pour analyser les composantes de logiciels comme [Fossology](#), [ScanCode](#) ou encore [OSS Review Toolkit \(ORT\)](#).

Comment faire respecter les licences libres ? A quel outillage juridique a-t-on recours pour faire respecter les licences libres ?

Juridiquement, les licences libres et Open Source sont des « offres de contracter » c'est-à-dire qu'un contrat est formé à chaque fois qu'une personne accepte (tacitement ou implicitement par le seul usage du composant) les conditions et obligations associées. Il est donc nécessaire, à l'échelle d'un projet (et idéalement de l'organisation) de mettre en place des outils qui permettent d'avoir une vision exhaustive des composants Open Source utilisés, des licences associées et des contextes d'utilisation. À partir de là, une politique Open Source (document spécifique interne) permettra d'identifier et de formaliser les processus devant être suivis afin d'assurer une complète conformité vis-à-vis des licences.

Ressources :

- [La page « Le côté légal de l'Open Source »](#) du guide Open Source impulsé par GitHub propose des réponses ciblées pour plusieurs cas de figure.
- [Le logiciel Hermine](#) permet d'outiller en interne les processus de validation des licences et le suivi de l'exécution des obligations sous-jacentes.
- Consultez la vidéo « [Les licences libre](#) » du MOOC [Apprendre l'Open Source](#).

Pourquoi et comment définir et mettre en œuvre une politique de gestion des usages de composants Open Source ?

Qu'elles soient formalisées ou non, il existe certainement déjà un certain nombre de bonnes pratiques suivies en interne par les équipes en charge du développement. La définition et la mise en application d'une politique Open Source vise à aborder l'ensemble de ces enjeux à l'échelle de l'organisation, afin d'impliquer l'ensemble des personnes susceptibles d'être concernées par l'usage de l'Open Source (développeurs et responsables produits, mais aussi les affaires juridiques, les achats, les personnes en charge de la qualité, de la sécurité, etc.). Elle se traduit ensuite par la définition d'un certain nombre de processus permettant d'assurer une validation de l'Open Source en fonction des licences, des contextes d'utilisation et des modalités d'usages. Une telle démarche permet de sécuriser l'usage de l'Open Source en se concentrant en priorité sur les situations à risque.

Ressource :

- Il existe des outils, comme [Hermine](#), pour faciliter la gestion des composants logiciels Open Source, les politiques de licences et leurs obligations.

Quelles sont les bonnes pratiques à adopter pour lancer puis gérer un projet Open Source ?

Lorsqu'on rejoint un projet déjà existant, il est d'usage de regarder la manière dont le programme est structuré, comment le projet est partagé comment la contamination est contrôlée.

Lorsqu'on lance un projet, il est recommandé de prendre en compte les risques juridiques en amont, ainsi que les obligations de la licence choisie.

Ressources :

- Un certain nombre de bonnes pratiques pour le développement de logiciels libres et la gestion de communautés associées ont été compilées et partagées dans la [Politique de contribution aux logiciels libres de l'État](#).
- Consultez la vidéo « [Organisation, gouvernance de projets Open Source](#) » du MOOC [Apprendre l'Open Source](#).

Liste des outils et ressources partagés :



Outils d'aide à la vérification de la compatibilité/conformité de licences

- **Outil 1** : [Hermine](#) : Outil de gestion des composants logiciels Open Source, des politiques de licences et de leurs obligations
- **Outil 2** : [Fossology](#) : boîte à outils pour scanner et trouver les licences utilisées dans des briques logicielles
- **Outil 3** : [BlackDuck](#) : Solution d'analyse de composants logiciels Open Source
- **Outil 4** : [OSS Review Toolkit \(ORT\)](#): outil d'aide à la vérification de conformité de licences



Ressources : Les bonnes pratiques du monde de l'Open Source et en matière de licences

- **Ressource 1** : [Politique de contribution aux logiciels libres de l'État](#)
- **Ressource 2** : [Choix et utilisations de licences libres - système d'exploitation GNU](#)
- **Ressource 3** : [Open Source Guide : Le côté légal de l'Open Source](#)
- **Ressource 4** : [MOOC « Apprendre l'Open Source »](#)

2.2 Groupe 2 « Valorisation et Open Source »

Questions fréquentes & éléments de réponses

Quelles sont les licences permettant d'allier l'Open Source avec une protection par le brevet ?

Le risque que peuvent présenter les brevets logiciels en matière de collaboration au sein de projets Open Source ont pendant longtemps été un frein à l'adoption du modèle. Aujourd'hui plusieurs mécanismes cohabitent et permettent de sécuriser relativement bien cet aspect : d'une part l'ajout quasi systématique

au sein des « nouvelles licences » (Apache 2.0, GPL 3.0, EPL 2.0 ou MPL 2.0) de clauses dites « patent peace », qui conditionnent le bénéfice des licences à l'absence d'action en contrefaçon de brevets contre un autre utilisateur ou contributeur du projet ; la systématisation des promesses de non-agression et la mise en place d'actions collectives telles que l'Open Invention Network ou le LOT (License on Transfer) Network.

Ressources :

- La question de l'Open Source et des brevets est traitée de manière approfondie dans cet article : [« Open Source et brevetabilité des inventions mises en œuvre par ordinateur. Quelle coexistence ? Quelle compatibilité ? »](#) de Benjamin Jean (2020).
- Le [Patent Pledge d'OVHcloud](#) est un exemple d'élément stratégique qui permet à cette entreprise d'articuler une démarche Open Source et une démarche de sécurisation en matière de brevets à l'international.
- L'[Open Invention Network](#) est un fonds de propriété intellectuelle créé pour réunir un ensemble de brevets en lien avec le noyau Linux et d'autres technologies connexes afin de les mettre gratuitement à disposition des développeurs de logiciels libres.
- Le [LOT Network](#) est une organisation à but non-lucratif créée pour lutter contre les « chasseurs de brevets » ou *patent trolls*.

Quelles sont les licences permettant d'allier l'Open Source et normalisation ?

L'Open Source est un modèle bien connu des organismes de normalisation, certains imposant même l'existence d'une implémentation Open Source de toute technologie normalisée. Néanmoins, certains organismes historiquement plus représentatifs d'acteurs des télécoms que du numérique tolèrent l'existence de brevets sur certaines normes à condition d'en permettre un accès sous des conditions FRAND (*Fair, Reasonable, and Non-Discriminatory*). Cela peut empêcher certains logiciels Open Source d'en bénéficier, le modèle décentralisé de l'Open Source n'étant pas adapté aux contractualisations nécessaires.

Ressources :

La question est traitée de manière approfondie dans les deux articles suivants :

- [« Open Source et brevetabilité des inventions mises en œuvre par ordinateur. Quelle coexistence ? Quelle compatibilité ? »](#) de Benjamin Jean (2019).
- [“Why Legal Issues are the Wrong Lens to understand the Open Source and FRAND issue”](#) de Simon Phipps (2019).

Est-il possible pour un projet Open Source de publier un certificat d'utilité afin de se protéger et d'éviter à la fois de se faire contraindre par les brevets des autres ?

À l'instar des brevets, les certificats d'utilité sont des titres de propriété industrielles qui peuvent être utiles pour assurer un premier niveau de protection à une invention. Dans un contexte numérique international, ces titres présentent néanmoins le défaut de ne pas être suffisamment harmonisés. Néanmoins, il n'est pas nécessaire de déposer de tels titres pour se protéger contre des brevets de tiers :

il suffit en effet de partager le maximum d'éléments publiquement afin d'alimenter un état de l'art qui constituera autant d'antériorités susceptibles de contester le brevet d'un tiers, voire son enregistrement.

Ressource :

- Pour plus d'informations sur le certificat d'utilité, consultez [la page dédiée sur site de l'INPI](#).

Comment peut-on réaliser des analyses de compatibilité de licences Open Source entre elles ?

La notion de compatibilité désigne une situation où plusieurs composants soumis à des licences peuvent être collectivement distribués au sein d'un tout soumis à une autre licence. Ainsi, l'analyse de la compatibilité entre licences repose sur une vision précise des éléments constitutifs d'un projet (inventoriés dans un *Software Bill of Materials*), une standardisation des éléments relatifs aux licences ainsi qu'une bonne compréhension des typologies d'obligations existantes, et enfin une connaissance précise de l'architecture logicielle dans laquelle interagissent l'ensemble de ces composants.

Dans le cas de figure des *Integrated Development Environment (IDE) Open Source (RStudio, Atom, Modelica, etc.)*, un code développé en langage R ou Python par exemple peut-il être distribué sous licence privée ou est-il automatiquement distribué sous licence PSFL?

Quid des conditions conduisant à la contamination par le copyleft par exemple dans le cas d'une utilisation d'un package copyleft R ou à l'inverse juste avec le langage R ?

Par principe, les logiciels développés dans un langage Open Source ne sont pas considérés comme étant directement dérivés du langage en lui-même. Ainsi, ils devraient pouvoir être distribués selon leur propres termes. Néanmoins, il arrive parfois que certaines bibliothèques Open Source soient combinées au code développé lors de sa compilation. En ce cas, il est par principe nécessaire de s'assurer du bon respect de la licence associée à ces bibliothèques, même si la majorité des langages Open Source utilisent à cette fin une licence permissive ou ajoutent des exceptions à leur licence pour éviter une application contraignante de la licence dans ce contexte.

Dans le cas de conditions conduisant à la contamination par le copyleft, le raisonnement est le même que sur les interfaces. On peut estimer que le langage est protégé, mais le langage est un objet de droit en lui-même indépendant des applications qui sont créées. Il faut une mention de la licence du langage, mais ensuite on peut mettre une licence différente sur une application.

Quid des licences appliquées sur les images docker (officielles/personnelles) qui utilisent plusieurs composants ? Par exemple, une image officielle Redis qui elle-même utilise la distribution Debian qui est soumise à une licence spécifique ?

Il faut distinguer la création d'un fichier Dockerfile (fichier texte qui contient les instructions nécessaires à la création d'une image de conteneur) et une image docker contenant l'ensemble des composants nécessaires au fonctionnement d'une application. Les images Docker vont être une agrégation d'un ensemble de composants soumis à des régimes très divers, et leur distribution impose de s'assurer du bon respect des licences sur l'ensemble des applications ainsi réunies, ce qui peut être très complexe.

Ressource :

- Le projet [Yocto](#) pour créer des images Linux sur mesure peut aider à voir comment systématiser les aspects de création d'objets complexes dans d'autres contextes similaires mais différents de celui des images Docker.

Comment associer Open Source et valorisation du projet, dans quelle mesure la licence va-t-elle me « protéger » vis à vis de la concurrence?

L'Open Source repose effectivement sur le principe d'un certain « lâcher-prise » de la part des éditeurs, condition *sine qua non* pour que les utilisateurs ou contributeurs tiers décident de contribuer au projet. Néanmoins, une telle diffusion en Open Source génère de la valeur, possiblement plus qu'une diffusion par un éditeur monopolistique, et l'enjeu est ainsi de pouvoir identifier d'une part la valeur créée et d'être en capacité d'en tirer les bénéfices d'autre part. Cette répartition de la valeur pourra ensuite être plus ou moins équilibrée en fonction des contributions effectives au sein du projet et de la volonté de faire vivre ou non un écosystème large de contributeurs tiers.

Dans le cadre d'une prestation de service autour d'un logiciel libre, comment trouver les entreprises utilisatrices ?

Exemple : Le logiciel Mercurial est déjà doté de principaux contributeurs, d'un *release manager* et d'un animateur de la communauté. Le modèle économique est de vendre du service expert autour du logiciel, mais il y a une difficulté à savoir qui sont les clients éventuels. De plus, il n'y a pas de structure qui s'occupe de l'édition du logiciel de manière formelle et administrative. Sachant que 70% du temps dépensé n'est pas facturé ou financé et se consacre majoritairement à la communauté et à la plateforme Heptapod, la question se pose de la gestion sur GitHub.

Il est important pour un projet Open Source d'avoir une vision de la communauté d'utilisateurs du projet. Contrairement aux contributeurs qui sont généralement connus (il est par exemple possible d'imposer la signature d'un *Developer Certificate of Origin* à tout contributeur du projet), les utilisateurs sont généralement beaucoup moins visibles. Certains outils et certaines bonnes pratiques existent néanmoins.

Ressources :

- Il existe des outils de visualisation comme ceux proposés par [Bitergia](#) afin d'évaluer l'état d'un projet Open Source.
- Consultez [la présentation d'une étude réalisée par le cabinet inno³ pour le CNES relative à l'évaluation de la valorisation par l'Open Source \(2019\)](#).

En tant qu'éditeur j'ai peur de me faire « dépasser » par des gros acteurs du numérique, que puis-je faire ?

Le risque est qu'un marché se développe où des acteurs qui sont les plus gros contributeurs et mainteneurs ont une prise sur la gouvernance du projet, comme c'est déjà le cas avec Google et Facebook pour l'utilisation de Mercurial. Afin d'éviter que cela se produise, il est nécessaire que les projets

anticipent autant que possible ces situations et encadrent les conditions liées à la participation, à la feuille de route (gouvernance), ainsi que l'utilisation de la marque du projet.

Ressources :

- L'atelier Systematic sur le sujet « [Passer d'un modèle économique de service à un modèle d'éditeur Open Source](#) » a abordé cette question dans le cas d'OSBL. La [vidéo](#) est consultable en ligne.
- [L'annuaire du CNLL](#) recense les entreprises du logiciel libre en France.

Faut-il signer un contrat avec chaque contributeur ou client ?

Non, la licence Open Source suffit à contractualiser les relations entre le projet et ses utilisateurs. Il est néanmoins courant que les projets sécurisent les relations avec les contributeurs assujettissant toute contribution *upstream* (au projet d'origine) à la signature de contrats de services additionnels, *Developer Certificate of Origin* (DCO) ou un *Contributor License Agreement* (CLA). Les DCO sont généralement communautairement très acceptés alors que les CLA sont souvent contestés car ils induisent un déséquilibre entre l'éditeur et les autres contributeurs.

Nous avons une communauté autour d'un logiciel GPL que nous souhaiterions encadrer. Que devons-nous savoir ? Quels sont les points d'attention en matière de politique de contribution, d'organisation de la gouvernance, de comitologie et autres documents contractuels ?

Il n'y a pas de « bons » ou de « mauvais » en termes de diffusion Open Source d'un projet, néanmoins le bénéfice que l'on tirera d'une démarche Open Source est directement lié à l'investissement qui sera fourni pour initier cette dynamique communautaire. En général, les contributions sont assez limitées au départ, donc il peut y avoir un coût assez élevé pour la structuration et la constitution de la communauté, mais ensuite le temps d'accompagnement peut être réparti et moins coûteux.

Exemple : Le projet [SciKitLearn](#) est un exemple de projet Open Source qui a demandé beaucoup de temps de gestion et de suivi des relations utilisateurs.

Une interface type API (ou protocole de communication) peut-elle être Open Source?

La question de la protection des APIs par un droit de propriété intellectuelle n'est pas définitivement tranchée : non prévue par la Loi, ces APIs vont pouvoir être considérées comme protégeables au cas par cas.

Exemple : Dans le cas du procès Oracle America, Inc. vs Google, Inc., le titulaire des droits d'auteur sur le langage de programmation Java a accusé la société Google d'avoir enfreint son droit de copyright. Le 5 avril 2021, la Cour Suprême des États-Unis décide que l'implémentation faite par Google de l'API Java consiste bel et bien en du « fair-use ». Référence (en anglais) : https://www.supremecourt.gov/opinions/20pdf/18-956_d18f.pdf

Néanmoins logiciels et API restent des objets de droit distincts susceptibles d'avoir des stratégies de valorisation différenciées nécessaires à articuler.

Ressource :

- Le [Interoperability Information Licence Agreement \(INTILA\)](#) propose un modèle de licence adaptable et configurable qui est proposé comme canevas pour concéder sous licences des Informations d'Interopérabilité (ce qui comprend APIs, formats et protocoles).



Liste des ressources partagées :

- **Ressource 1** : Le rapport de la Commission européenne sur le [INTILA Template \(Interoperability Information Licence Agreement\)](#)
- **Ressource 2** : L'article « [Open Source et brevetabilité des inventions mises en œuvre par ordinateur. Quelle coexistence ? Quelle compatibilité ?](#) » de Benjamin Jean.
- **Ressource 3** : [Open Source Business Lab #2 : Passer d'un modèle de service à un modèle éditeur \(octobre 2020\)](#) avec [vidéo en replay](#).
- **Ressource 4** : Le [Patent Pledge d'OVHcloud](#) est un exemple d'élément stratégique qui permet à cette entreprise d'articuler une démarche Open Source et une démarche de sécurisation en matière de brevets à l'international.
- **Ressource 5** : L'[Open Invention Network](#) est un fonds de propriété intellectuelle créé pour mettre gratuitement à disposition des développeurs de logiciels libres un ensemble de brevets en lien avec le noyau Linux.
- **Ressource 6** Le [LOT Network](#) est une organisation à but non-lucratif créée pour lutter contre les « chasseurs de brevets » ou *patent trolls*.
- **Ressource 7** : Le projet [Yocto](#) : ensemble flexible d'outils pour les développeurs de systèmes embarqués.
- **Ressource 8** : [Bitergia](#) aide les entreprises à analyser leurs projets de développement logiciel en leur fournissant des outils et des connaissances pour améliorer la prise de décision.
- **Ressource 9** : [La page dédiée au certificat d'utilité sur site de l'INPI](#).
- **Ressource 10** : [L'annuaire du CNLL](#) recense les entreprises du logiciel libre en France.
- **Ressource 11** : [la présentation d'une étude réalisée par le cabinet inno³ pour le CNES relative à l'évaluation de la valorisation par l'Open Source](#)
- **Ressource 12** : L'article "[Why Legal Issues are the Wrong Lens to understand the Open Source and FRAND issue](#)" de Simon Phipps (2019).

3 | Pistes d'approfondissement

Thématiques ayant émergé pouvant être approfondies ultérieurement lors de prochains ateliers.

- **Les modèles économiques de l'Open Source**
- **La problématique d'une matrice de compatibilité générale des licences libres et Open Source**
- **L'outillage juridique pour faire respecter les licences**
- **Les bonnes pratiques du monde de l'Open Source**
- **Les enjeux associés au choix de licence de briques logiciels**
- **L'articulation entre périmètres des différents modes de sécurisation pour une valorisation Open Source**
- **L'organisation de la gestion interne d'un projet Open Source en fonction des typologies d'intervenants (développeurs, chercheurs, personnes valorisation / transferts, juristes)**
- **La structuration d'une communauté Open Source**
- **La gouvernance d'un projet Open Source**

4 | Annexes

4.1 Personnes ayant contribué

Groupe « Licences Open Source »

- Florence BLAZY, Coordinatrice Hub Open Source, Systematic Paris-Région
- Vincent CHEVROTON, Group Legal IP, FAURECIA
- Jérôme HERLEDAN, Chef de projet Avant-vente & Busidev, ATOS
- Farid MANSOURI, Chef de Projet, CITEOS, VINCI ENERGIES
- Elisa NECTOUX, Responsable commerciale, Belledonne Communications - éditeur de Linphone

Groupe « Valorisation et Open Source »

- Pierre BAUDRACCO, Vice-président chargé Business du Hub Open Source / co-président CNLL/ Président BlueMind
- Franck BRET, Ingénieur logiciel, Octobus
- Thomas BRONNIART, Centre open source, Atos
- Philippe JARRIN, Directeur général, @UniText
- Thierry LEBLOND, Editeur cybersécurité, Scille / Parsec
- Aurore LEMASSON, Direction R&D partenariat propriété intellectuelle et développement logiciels, EDF
- Delphine PICAVET, Attachée services transfert innovation et partenariat, INRIA Saclay
- Véronique TANGUY, Direction de la valorisation, ONERA

Membres de l'équipe d'ino³

- Benjamin JEAN, Fondateur du cabinet et juriste spécialisé en propriété intellectuelle
- Vincent BACHELET, Juriste, consultant et doctorant CIFRE
- Célya GRUSON-DANIEL, Consultante, praticienne-chercheuse
- Maya ANDERSON-GONZALEZ, Chargée de missions
- Romane LEBAN-MATHIEU, Juriste

4.2 Liste des questionnements consignés sur les supports des ateliers

Tableau blanc Excalidraw du groupe « Licences »

1. Par où commencer pour sécuriser la problématique des licences Open Source? Quels sont les outils à disposition ? Quelles sont les bonnes pratiques ?
2. De quels moyens juridiques disposons-nous pour faire respecter la GPL, en France et à l'international ?
3. Quelles sont les limites aux licences dites permissives en terme de modification, réutilisation? Puis-je copier-coller du code du fichier original vers le mien ? Ou dois-je garder le fichier original et ses entêtes ? Puis-je simplement prendre tout un logiciel sous licence permissive, renommer le logiciel et le commercialiser tel quel ?
4. Plateforme de données territoriales en open source pour les collectivités publiques sous licence apache 2.0. Questions sur les licences des briques logicielles utilisées et la pérennité des licences.
5. Copyleft vs non copyleft vs copyright ? Compatibilité entre licences ? Faut-il lister les composants open source utilisés ? Droit paternité, Droit moral, Droit utilisation, Autres droits et libertés ?
6. Classification des licences OK/NOK en fonction des risques et des usages ? Comparaison MPL/LGPL ? GPL: les limites de la contamination ?
7. Recherche de bonnes pratiques et de méthodes pour sécuriser notre stratégie PI, en lien avec la partie brevet en vue d'un développement international (hors Europe). On est dans une phase de structuration, préventif.
8. Quelles sont les licences utilisées au sein de code source d'un logiciel (libraires, dépendances) ?

Tableau blanc Excalidraw du groupe « Valorisation »

1. OS et Brevets / OS et Normes : quelles sont les licences OS permettant d'allier l'OS avec une protection par le brevet ou la normalisation ?
2. Nous envisageons de déposer une demande de brevet pour notre technologie. En parallèle, nous envisageons de diffuser notre logiciel, basé sur nos technologies (dont celle faisant l'objet de la demande de brevet), en open-source. Comment monétiser l'usage d'une technologie en open-source (ou d'une licence de brevet) ? Est-ce possible ? Souhaitable ?
3. Publier un certificat d'utilité pour se protéger un peu et empêcher de se faire bloquer par les brevets des autres ?
4. Remédiation des codes / compatibilité des licences OS entre elles : quels outils sont à disposition pour nous aider à réaliser ces analyses ?
5. Cas des IDE OS (Logiciel R, Python, Modelica, etc ...) : par exemple, un code développé en langage Python peut-il être distribué sous licence privée ou est-il automatiquement distribué sous licence PSFL?
6. Une licence qui protège la marque et partage les revenus à la proportion des contributions ?

7. Je voudrais créer une dynamique Open Source, mais qui déclenche aussi des financements, des sources de revenu, avec une « protection » par la licence ...J'ai un objet unique et innovant dans @UniText, je voudrais me protéger et protéger un flux de revenu tout en bénéficiant des avantages de l'Open Source.
8. Une interface type API (ou protocole de communication) peut-elle être open source?
9. Dans le cadre de prestation de service autour d'un logiciel libre, comment trouver les entreprises utilisatrice ?
10. Monétisation de l'open source Mixte FSL / Open core Licence (type BSL)
11. Nous avons une communauté autour d'un logiciel GPL, nous souhaiterions l'encadrer ... quel est le B AB à savoir là-dessus, les points d'attention, etc... ==> politique de contribution, organisation de la gouvernance / comitologie, autres documents contractuels
12. Dans mon métier je suis amenée à devoir arbitrer un choix de licence open source ou propriétaire pour un usage soit de recherche, soit un choix de licence logiciel open source /propriétaire exploité dans le cadre d'un projet entrepreneurial qui va devenir une startup deeptech numérique. Je dois pouvoir aussi guider le chercheur et le porteur en le questionnant pour connaître les usages et les finalités du logiciel.
13. Les conditions conduisant à la contamination par le copyleft.