

Cours Logiciels Libres

Licences libres (suite) et processus de contribution

Théo Zimmermann

Université de Paris & Inria

Vendredi 4 février 2022

Domaine public

Définition : une œuvre est dans le domaine public lorsqu'elle **n'est plus protégée** par des droits patrimoniaux.

“Mettre une œuvre dans le domaine public” : ce n'est **pas juridiquement possible** dans tous les pays. Par conséquent, des licences existent qui accordent des **droits équivalents** au cas où ce ne serait pas permis (Unlicense, CC-0).

Choisir une licence libre

Règle 1 : on n'oublie pas de mettre une licence sur son code !

Sans licence :

- Vos **utilisateurs** n'ont aucun droit.
- Vous n'avez aucun droit sur les **contributions externes** !

Choisir une licence libre

Règle 2 : on n'invente pas sa propre licence !

Conséquence de la prolifération de licences :

- Risques juridiques (licences moins comprises, moins testées).
- **Incompatibilité** entre licences (difficile de combiner des logiciels libres entre eux).

Il vaut mieux choisir une licence **populaire** :

- Soit une licence populaire en général (MIT, BSD2, BSD3, Apache 2.0, MPL 2.0, licences GNU).
- Soit une licence populaire dans l'**écosystème** auquel on contribue (mais en général elle fait partie de la liste précédente).

Choisir une licence libre

Règle 3 : on choisit une licence qu'on comprend !

- Certaines licences ont des conditions **complexes**. Par exemple la licence GNU LGPL donne des droits supplémentaires, mais à des conditions techniques qui ne sont pas autant adaptées à tous les langages de programmation.
- Il est **difficile de changer de licence** plus tard, une fois qu'on a reçu de nombreuses contributions externes.

Choisir une licence libre

- Domaine public ?
- Permissive ?
- Protection contre les brevets ?
- Copyleft faible ?
- Copyleft fort ?
- Le logiciel est-il destiné à tourner sur un serveur ?

Et pour la documentation ?

- Le plus commun est que la documentation soit sous la **même licence que le code**.
- Mais des licences spécifiques aux contenus autres que logiciels existent (par exemple, les licences **Creative Commons**).

Creative Commons

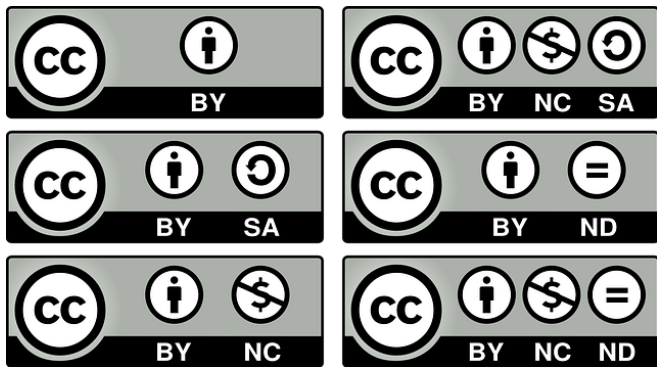


Figure 1: Les différentes licences Creative Commons

Attention ! Seules CC-BY et CC-BY-SA sont des licences libres.

Appliquer une licence libre

Les recommandations varient d'une licence à l'autre :

- En général, on inclut le **texte de la licence** dans un fichier LICENSE à la racine du projet.
- Certaines licences recommandent en plus l'ajout d'un **commentaire en haut de chaque fichier** de code source.

Application de licences aux contributions

Certains projets demandent de “signer” des documents :

- DCO (Developer Certificate of Origin),
- CLA (Contributor License Agreement),
- CTA (Copyright Transfer Agreement).

Quelles peuvent être les raisons de ces demandes ?

DCO (Developer Certificate of Origin)

- Créé pour le projet Linux, également utilisé pour d'autres projets.
- Le contributeur s'engage sur l'origine de la contribution et sur son droit à la soumettre à la licence du projet.
- Très léger : nécessite une ligne par commit (Signed-off-by: Your Name <your@email.com>), option `-s` de git.

CLA (Contributor License Agreement)

Même utilité qu'un DCO, plus des droits supplémentaires (par exemple pour l'entreprise qui maintient le projet) :

- Par exemple, le droit d'inclure la contribution dans un code propriétaire.
- Ou bien le droit de changer la licence.

CTA (Copyright Transfer Agreement)

- Cession des droits patrimoniaux.
- Utilisé par la Free Software Foundation pour pouvoir lancer des procès au nom de tous les contributeurs.

Application de licences aux contributions

Extrait des Terms of Services de GitHub :

Whenever you add Content to a repository containing notice of a license, you license that Content under the same terms, and you agree that you have the right to license that Content under those terms. If you have a separate agreement to license that Content under different terms, such as a contributor license agreement, that agreement will supersede.

Isn't this just how it works already? Yep. This is widely accepted as the norm in the open-source community; it's commonly referred to by the shorthand "inbound=outbound". We're just making it explicit.

Contribuer à un projet open source

Qu'est-ce que ça veut dire ?

Différents types de contributions

- Aider la communauté :
 - répondre à des questions sur les forums,
 - partager des extensions.
- Documentation.
- Rapports de bugs.
- Fonctionnalités.
- Maintenance.
- Diffusion.

Différents types de contributions

- Aider la communauté.
- Documentation :
 - écrire des tutoriels,
 - rapporter des problèmes dans la documentation officielle,
 - proposer des améliorations,
 - traduire la documentation (ou le logiciel).
- Rapports de bugs.
- Fonctionnalités.
- Maintenance.
- Diffusion.

Différents types de contributions

- Aider la communauté.
- Documentation.
- Rapports de bugs :
 - tester les nouvelles versions (notamment beta),
 - rapporter des problèmes en ouvrant des tickets,
 - aider à reproduire et à comprendre les problèmes,
 - écrire des correctifs.
- Fonctionnalités.
- Maintenance.
- Diffusion.

Différents types de contributions

- Aider la communauté.
- Documentation.
- Rapports de bugs.
- Fonctionnalités :
 - proposer de nouvelles fonctionnalités,
 - participer aux discussions,
 - implémenter des changements.
- Maintenance.
- Diffusion.

Différents types de contributions

- Aider la communauté.
- Documentation.
- Rapports de bugs.
- Fonctionnalités.
- Maintenance :
 - aider à la revue de code,
 - aider à l'intégration des contributions,
 - aider à la sortie de nouvelles versions.
- Diffusion.

Différents types de contributions

- Aider la communauté.
- Documentation.
- Rapports de bugs.
- Fonctionnalités.
- Maintenance.
- Diffusion :
 - faire connaître le logiciel,
 - produire des paquets pour différentes distributions.