



Pilier CODES SOURCES

Guichet de la donnée UP



Extrait de la formation LICENCES LOGICIELLES

Il faut une licence sur mon code !

URL <https://scienceouverte.univ-poitiers.fr/piliercodeslogiciel/licences-logicielles/>



Bruno MERCIER

11/03/2026



2

Il faut une licence sur mon code !

Pour pouvoir **partager son code source**, il faut **poser une licence** sur son code !

- Pourquoi ?

Parce que **sans licence**, votre code source est vu comme un code **sous licence propriétaire** !

Et **diffuser** un code **sans licence** permet aux utilisateurs finaux

- de **lire votre code** (le seul droit que vous venez de concéder en partageant votre code !)

mais... ils n'ont

- **Ni le droit** de le **compiler**, **l'interpréter** ou **l'exécuter** !
- **Ni le droit** de le **sauvegarder**, le **modifier**, en **extraire des parties** ou le **redistribuer** !



Plan

- **Droit d'auteur** appliqué au logiciel
- Les grandes différences entre les **catégories de licences logicielles**
- Les **compatibilités entre les licences logicielles**
- Comment **choisir une licence logicielle** ? (divers exemples)
- Où poser l'ensemble des fichiers de licence / la liste des auteurs et des contributeurs ?
- Les **licences sur les données et la documentation**



Bruno MERCIER

11/03/2026





Génèse du droit du logiciel

- **Premiers ordinateurs compatibles** avec les grands systèmes IBM, à la fin des années 1960
 - IBM décida donc de facturer séparément logiciel et matériel (avant : annexe du matériel)
 - IBM PC en 1981 (Personal Computer) + interpréteur BASIC Microsoft sous licence
- Deux choix pour caractériser juridiquement le logiciel (Code de la **Propriété intellectuelle**)
 - **Droit des brevets**
 - Inclus dans le « droit de la propriété industrielle » => Réservé aux **inventions techniques nouvelles** : protection d'une application **industrialisable** pendant 20 ans
 - Si **Logiciel embarqué dans une invention technique** => Suivre la Formation « **Propriété Intellectuelle** » avec le **SPVR** (Droit des brevets, droit des marques, droit d'auteur, ...)
 - **Droit d'auteur**
 - Inclus dans le « droit de la propriété littéraire et artistique » => Réservé aux **œuvres**
 - **Choisi pour le Logiciel** (Code source + Code binaire et Documentation associés)
 - 1980 aux États-Unis (modification du Copyright Act)
 - 1985 en France (loi du 3 juillet 1985 => abrogée et remplacée en 1992)
 - Existence de la Convention de Berne (1886)
 - Protection internationale automatique et immédiate

Licence propriétaire



Droit d'auteur (en France)

- **Droits moraux d'un logiciel**
 - **Un seul droit d'auteur est conservé** pour un logiciel
 - **Droit à la paternité** (mention de l'auteur)
 - **Les 3 autres droits disparaissent**
 - **Droit de divulgation** (moment et conditions de livraison)
 - **Droit de repentir** (retrait du logiciel => Vol 501 d'Ariane 5 en 1996)
 - **Droit au respect de l'œuvre** (opposition aux modifications => Louvois de 1996 à 2021)
- **Droits patrimoniaux d'un logiciel**
 - C'est le **droit d'exploitation** d'un logiciel
 - **La représentation** (communication)
 - **La reproduction** (sur support physique)
 - **Le choix de la licence** (par défaut, propriétaire)



La dévolution des droits patrimoniaux pour les auteurs de logiciels

- [L'Article 2 de la Loi n° 94-361 du 10 mai 1994](#) modifie [l'Article L113-9 du code de la propriété intellectuelle](#) sur le *Chapitre III : Titulaires du droit d'auteur* en stipulant :
 - *"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"* ;
 - Ces dispositions *"sont également **applicables aux agents de l'Etat**, des collectivités publiques et des établissements publics à caractère administratif"*.

Transfert automatique des « droits patrimoniaux » aux Employeur(s)

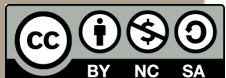


Droit d'auteur (en France)

- **Droits moraux d'un logiciel**
 - **Un seul droit d'auteur est conservé** pour un logiciel
 - **Droit à la paternité** (mention de l'auteur)
 - **Les 3 autres droits disparaissent**
 - **Droit de divulgation** (moment et conditions de livraison)
 - **Droit de repentir** (retrait du logiciel => Vol 501 d'Ariane 5 en 1996)
 - **Droit au respect de l'œuvre** (opposition aux modifications => Louvois de 1996 à 2021)
- **Droits patrimoniaux d'un logiciel**
 - C'est le **droit d'exploitation** d'un logiciel
 - **La représentation** (communication)
 - **La reproduction** (sur support physique)
 - **Le choix de la licence** (par défaut, propriétaire)

Auteur(s)

Employeur(s)



Exception pour les stagiaires, auteurs de logiciels

- Un **stagiaire** reste donc **titulaire de ses droits patrimoniaux** sur le logiciel qu'il a développé dans le cadre de son stage (si stage effectué avant le 17/12/2021)
- La **clause de cession de la propriété intellectuelle** dans la convention de stage n'est pas applicable sur un logiciel qui n'existe pas dès le début du stage. Seule la signature d'un **contrat de cession de droits de l'invention et des droits d'auteur** à l'entreprise à la fin de son stage permet de céder les droits sur le logiciel nouvellement créé.

ATTENTION

- Conserver la liste de TOUS les auteurs (et leur employeur éventuel)
- Conserver TOUS les documents de cession des droits de PI
- Accord de TOUS pour poser/changer la licence



La dévolution des droits patrimoniaux pour les auteurs de logiciels

- [L'Article 2 de l'Ordonnance n° 2021-1658 du 15 décembre 2021](#) ajoute l'[Article L113-9-1 au code de la propriété intellectuelle](#) sur le *Chapitre III : Titulaires du droit d'auteur* en stipulant :
 - *"Sauf stipulations contraires, lorsque des personnes qui ne relèvent pas de l'article L. 113-9 et qui sont **accueillies dans le cadre d'une convention** par une personne morale de droit privé ou **de droit public** réalisant de la recherche **créent des logiciels** dans l'exercice de leurs missions ou d'après les instructions de la structure d'accueil, leurs **droits patrimoniaux sur ces logiciels et leur documentation sont dévolus à cette structure d'accueil**, seule habilitée à les exercer, **si elles se trouvent à l'égard de cette structure dans une situation où elles perçoivent une contrepartie** et où elles sont placées sous l'autorité d'un responsable de ladite structure".*

Transfert automatique des « droits patrimoniaux » aux Employeur(s)



Exception pour les stagiaires sans gratification ?

- La contrepartie n'est pas forcément financière !
 - Lorsque le **stage** est **obligatoire**, la contrepartie peut correspondre à l'obtention du diplôme / la validation du stage suite à l'évaluation d'un rapport et/ou une soutenance orale (rappel : pas d'obligation de financement si inférieur à 308h)
- Un **stagiaire** reste donc **titulaire de ses droits patrimoniaux** sur le logiciel qu'il a développé dans le cadre de son stage si ce stage est optionnel, inférieur à 308h, sans gratification et à l'initiative personnelle du stagiaire pour améliorer ses compétences !

ATTENTION

- Conserver la liste de TOUS les auteurs (et leur employeur éventuel)
- Conserver TOUS les documents de cession des droits de PI
- Accord de TOUS pour poser/changer la licence

Exception pour les prestataires de service ?

- Le **prestataire de service** (entreprise extérieure) reste donc **titulaire de ses droits patrimoniaux**
- Il appartient à celui qui commande un logiciel de veiller à la validité de la cession de droits à son profit. La cession des droits de propriété intellectuelle doit être rédigée conformément à l'[article L.131-3 du code de la propriété intellectuelle](#) et être annexée au contrat de développement
 - **Faire appel au SPVR** (Service Partenariat et Valorisation de la Recherche) pour rédiger correctement cette **clause de cession des droits de propriété intellectuelle** avant de passer une commande de marché public

ATTENTION

- Conserver la liste de TOUS les auteurs (et leur employeur éventuel)
- Conserver TOUS les documents de cession des droits de PI
- Accord de TOUS pour poser/changer la licence

L'intéressement pour les auteurs de logiciels

- Le [décret n° 96-858 du 2 octobre 1996](#) relatif à *l'intéressement de certains fonctionnaires et agents de l'Etat et de ses établissements publics ayant participé directement à la création d'un logiciel* stipule :
 - Dans l'article 1, *Les **fonctionnaires ou agents publics de l'Etat** [...] qui ont directement participé [...] à la **création d'un logiciel** [...] bénéficient d'une **prime d'intéressement** aux produits tirés, par la personne publique, de ces créations.*
- **Tout chercheur** (fonctionnaire, en CDI ou CDD) **peut bénéficier de l'intéressement** au même titre que son employeur => les **doctorants** sont concernés

ATTENTION

- Conserver le pourcentage d'effort de recherche de TOUS les auteurs
- L'effort de recherche n'est pas que du développement logiciel (encadrement scientifique, gouvernance du projet, ...)

L'intéressement pour les auteurs de logiciels

- Le [Décret n° 2023-772 du 11 août 2023](#) stipule :
 - *Les personnes physiques mentionnées à l'article L. 113-9-1 du code de la propriété intellectuelle, **accueillies dans le cadre d'une convention** par une personne morale de **droit public** mentionnée à l'article 1er, qui ont directement participé, lors de l'exécution de leurs missions ou d'après les instructions de la structure d'accueil, à la **création d'un logiciel, bénéficiant d'une prime d'intéressement** ;*
 - *sa contribution est déterminée conformément à l'article 4 du décret du 2 octobre 1996 relatif à l'intéressement de certains fonctionnaires et agents de l'Etat et de ses établissements publics ayant participé directement à la création d'un logiciel*
- **Tout non-salarié** (convention d'accueil) **peut bénéficier de l'intéressement** au même titre que sa structure publique d'accueil => les **stagiaires** sont concernés

ATTENTION

- Conserver le pourcentage d'effort de recherche de TOUS les auteurs
- L'effort de recherche n'est pas que du développement logiciel (encadrement scientifique, gouvernance du projet, ...)

L'intéressement pour les auteurs de logiciels

- Le [décret n° 2023-1321 du 27 décembre 2023](#) portant partie réglementaire du code de la recherche
 - a abrogé le [décret n° 96-858 du 2 octobre 1996](#) relatif à l'intéressement de certains fonctionnaires et agents de l'Etat et de ses établissements publics ayant participé directement à la création d'un logiciel
 - et codifié L'INTÉRESSEMENT DES CHERCHEURS dans le [Chapitre II \(Articles R532-1 à D532-13\) du code de la recherche](#).

ATTENTION

- Le montant total des bénéficiaires ne peut excéder 50%
- La liste des bénéficiaires et leurs attributions
 - proposés par le directeur (composante, laboratoire ou service)
 - fixés par le Conseil d'Administration de l'établissement

L'intéressement depuis le 1^{er} janvier 2024

- Le [Chapitre II \(Articles R532-1 à D532-13\) du code de la recherche](#) relatif à *L'INTÉRESSEMENT DES CHERCHEURS* stipule :
 - Dans la section 2, *la prime d'intéressement des agents publics liée à la création d'un logiciel* (Articles D532-2 à D532-6) ;
 - Dans la section 3, *l'intéressement versé aux auteurs de logiciels accueillis par une personne morale de droit public réalisant de la recherche* (Articles D532-7 à D532-9) ;
 - Dans la section 4, *l'intéressement lié aux services rendus lors de la participation à des opérations de recherche scientifique ou de prestations de services* (Articles D532-10 à D532-13) ;

Tout chercheur, stagiaire ou prestataire peut prétendre à l'intéressement



Définition d'un « Logiciel »

- Selon [l'arrêté du 29 novembre 1973 sur l'enrichissement du vocabulaire de l'informatique \(p.485 du JO\)](#) :
 - « *Logiciel (n.m.) : Ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de l'information (en anglais : software)* » ([arrêté du 22/12/1981](#) : « de l'information » => « des données »)
 - « *Matériel (n.m.) : Ensemble des éléments physiques employés pour le traitement de l'information (en anglais : hardware)* » (1981 : « de l'information » => « des données »)
- Depuis 1992, selon [l'article L. 112-2 du Code de la propriété intellectuelle \(CPI\)](#) :
 - « *Les logiciels, y compris le matériel de conception préparatoire [...] sont considérés [...] comme œuvres de l'esprit [...]* » donc protégés par le droit d'auteur.

Le droit d'auteur protège les œuvres de l'esprit

Mais l'œuvre doit être « mise en forme » et « originale » : le logiciel !
Car les idées, les fonctionnalités, les algorithmes ne sont pas protégeables.

Utilité d'une « Licence »

- La licence complète le cadre juridique, c'est un **contrat** qui protège les auteurs, les collaborateurs au développement et les utilisateurs finaux du logiciel.
 - Elle octroie des **droits / libertés** que l'on **peut** user/prendre et impose des **obligations** que l'on **doit** respecter => ne surtout pas la rédiger seul (le SPVR est là pour cela => **juristes**)
 - Si **aucune licence** posée => « licence privative / propriétaire » => « **Tous droits réservés** »
- Selon l'[article L. 355-2 du CPI](#) :
 - Toute personne utilisant, copiant, modifiant ou diffusant le logiciel **sans autorisation** explicite des détenteurs des droits patrimoniaux est coupable de **contrefaçon** et passible de **trois d'ans d'emprisonnement et 300 000€ d'amende** en France.

Mais il faut pouvoir prouver la paternité et surtout l'antériorité du code !

Déposer l'intégralité des éléments qui composent un logiciel (codes sources et documentations) chez un tiers de confiance :

- Sur une forge logicielle souveraine (celle de l'employeur)
- À l'APP (Agence de Protection des Programmes) => horodatage certifié

Logiciel libre par la FSF (Free Software Foundation)

- Fondation créée en 1985 par Richard Stallman pour **promouvoir le « logiciel libre »**
- Un logiciel est « **libre** » s'il garantit **4 libertés fondamentales** :
 - **Liberté d'exécuter** le programme, pour tous les usages (liberté 0) ;
 - **Liberté d'étudier** le fonctionnement du programme et **de l'adapter** à ses besoins (liberté 1) ;
 - Liberté **de redistribuer** des copies, donc d'aider son voisin (liberté 2) ;
 - Liberté **d'améliorer** le programme et **d'en faire bénéficier** la communauté (liberté 3).
- La **garantie** de ces 4 libertés nécessite un **accès au code source** => le **code source** doit être **ouvert** !

Naissance des premières licences libres :
Licences GNU avec notion de « Copyleft »

Le « Copyleft » : Gauche d'auteur ou copie laissée
Autorisation donnée par l'auteur d'un travail soumis au droit d'auteur
d'utiliser, d'étudier, de modifier et de diffuser son œuvre,
dans la mesure où cette même autorisation reste préservée



Logiciel « libre » ne veut pas dire « non commercial » !

Code ouvert par l'OSI (Open Source Initiative)

- Le terme « **open source** » est apparu en 1998 afin de lever l'**ambiguïté** sur le terme « **free** » qui désigne à la fois « **libre** » et « **gratuit** » (libre accès) et ainsi rassurer les entreprises (**commerce**)
- Un code est « **ouvert** » s'il respecte **10 principes** (aspects technique et pratique) :
 - **Redistribution libre** (la licence n'empêche aucune partie de vendre ou céder le logiciel) ;
 - **Code source** (le code source doit être fourni) ;
 - **Travaux dérivés** (les modifications et dérivations doivent être redistribuées à l'identique) ;
 - **Intégrité du code source de l'auteur** (la redistribution des modifications sous un nom et numéro de version différents peut être exigée) ;
 - **Pas de discrimination** contre les personnes, les groupes ou les domaines de compétence
 - **Distribution de la licence** (les droits s'appliquent lors de la redistribution) ;
 - **La licence ne doit pas être spécifique à un produit** (les droits s'appliquent lors d'une extraction d'un morceau du programme) ;
 - **La licence ne doit pas restreindre d'autres logiciels** (pas de restriction sur d'autres logiciels distribués avec le programme) ;
 - **La licence doit être neutre sur le plan technologique.**

L'expression « F/OSS » ou « FLOSS » (Free Libre Open Source Software) tente de faire la synthèse de ces différents mouvements
=> Un programme distribué sous une « licence libre » est un FLOSS !

Licences à Copyleft fort (diffusives)



- La licence initiale s'impose sur tout
- L'**obligation de réciprocité** évite de fermer un code libre
 - Tout ajout doit avoir une **licence à Copyleft fort**
 - **Pas d'utilisation commerciale** par un contributeur en fermant le code
- Exemples
 - [GPL](#) / GNU GPL (General Public License) => **La plus utilisée !**
 - [EUPL](#) (European Union Public License) => traduite en 23 langues !
 - [CeCILL](#) (Cea Cnrs Inria Logiciel Libre) => Droit français !



GPL-3.0-or-later

EUPL-1.2

La GPL a un Copyleft strict => les modifs doivent être publiées sous GPL !

Droits des Auteurs (pour chaque version)

- Peuvent changer la licence lors de la publication d'une nouvelle version
- Peuvent rendre propriétaire la nouvelle version (utilisation commerciale)

Licences à Copyleft faible (persistantes)

- La licence initiale reste
- Les ajouts peuvent avoir une autre licence (mais à **Copyleft faible ou fort**)
- Exemples
 - [LGPL](#) / GNU LGPL (Lesser **G**eneral **P**ublic License) => **La plus utilisée !**
 - [CeCILL-C](#) (Cea Cnrs Inria Logiciel Libre) => Droit français !



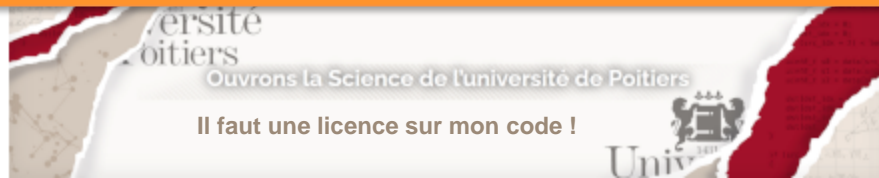
LG**P**L-3.0-or-later

Utilisée pour diffuser des Bibliothèques (Libraries)

- Peuvent être incluses dans un logiciel propriétaire (sans être modifiées)
- Doivent être republiées avec le logiciel propriétaire si modifiées

=> sous licence a Copyleft faible (a minima)

Exemples de bibliothèques sous LGPL :
Bibliothèques de composants graphiques comme GTK / SDL / Qt



Licences sans Copyleft (permissives / évanescentes)

- La licence initiale ne s'impose pas
- Les produits dérivés
 - peuvent avoir **n'importe quelle licence** (libre ou propriétaire)
 - mais doivent **conserver** a minima les informations de
 - **droit d'auteur** (liste des auteurs),
 - **copyright** (liste des établissements) et
 - **licence** de la version **initiale**
- Exemples
 - [MIT](#) (Massachusetts Institute of Technology) => **La plus utilisée !**
 - **Clause de fin de brevet** (non soumis au brevet incluant le code initial)
 - **Clause de non responsabilité**
 - [Apache](#) (Apache Software Foundation) => très utilisée !
 - **Modifications** du code doivent être **clairement indiquées**
 - **Clause de fin de brevet** (non soumis au brevet incluant le code initial)
 - [BSD 3-Clause](#) (Berkeley Software Distribution) => Proche du domaine public
 - **Publicité** soumise à **autorisation préalable** pour le réemploi des noms
 - **Clause de non responsabilité**
 - [CeCILL-B](#) (Cea Cnrs Inria Logiciel Libre) => Droit français !



BSD

MIT

Apache-2.0

BSD-3-Clause



Licence propriétaire (privative)

- **Ne pas mettre de licence !**

La législation internationale le permet
=> mais la distribution sera impossible !

- **Rédiger un contrat de licence**

- Qui explique ce qui est **autorisé** et ce qui est **interdit**
 - Généralement, pas d'accès au code source, publication uniquement des binaires
 - Clause d'interdiction de décompilation
- Qui détaille les **cas particuliers d'utilisation** du logiciel
 - Revente possible / interdite ?
- Qui spécifie les **conditions de commercialisation** du logiciel
 - Achat d'une licence perpétuelle ? / Abonnement ? (tarifs => votés au CA de l'établmt)
 - Achat d'une licence par utilisateur ? / par poste informatique ?
 - Système de jetons (avec / sans accès réseau) ?
 - Mises à jour (durée) ?
 - Evolutions (liste, coût) ?
 - Etc.

Utiliser un modèle existant
est plus rassurant

ATTENTION

- Ne jamais se lancer dans la rédaction de ce type de contrat tout seul !
- Faire appel au SPVR (Service du Partenariat et de la Valorisation de la Recherche)

Qui doit choisir la licence ?

- **Légalement**
 - Le **titulaire des droits patrimoniaux** => les **employeurs** des différents auteurs
Il y a des conventions entre les divers employeurs comme le CNRS, l'INSERM, l'INRA et l'UP
- **En pratique**
 - **Les principaux acteurs** => **auteurs**
 - Le **développeur** principal du projet (**doctorant**)
 - Le **responsable scientifique** du projet (**encadrant**)
 - Peut être soumis pour validation au Directeur d'Unité de recherche
- La **licence** doit être en **adéquation** avec le **financeur du projet**
 - Thèse **MESR / ANR / Région** => **Licence libre**
 - Thèse **Cifre**
 - **Contrat de collaboration** signé avec l'entreprise
 - Si besoin, le SPVR peut faire établir un **avenant au contrat** avec l'entreprise

La législation française impose de diffuser son code et ses données sous « licence libre » dès qu'au moins 50% du projet est financé par des fonds publics

Quelle licence choisir ?

- **Si vous modifiez un logiciel existant**
 - **Conserver la licence existante** si elle vous convient (et qu'elle existe !)
 - Sinon
 - Faire un état des lieux des **bibliothèques utilisées** dans le logiciel
 - Faire une **liste exhaustive des licences** de ces bibliothèques
 - Regarder les **compatibilités entre ces licences**
 - Puis **choisir une licence qui respecte les contraintes** de toutes les bibliothèques
- **Si vous développez votre propre code**
 - Qui n'utilise aucune bibliothèque (code indépendant, script)
 - **Choisir une licence qui corresponde à votre éthique, vos besoins et vos obligations !**
 - **La soumettre** à l'ensemble des auteurs **pour approbation (par écrit !)**
 - Qui utilise une ou plusieurs bibliothèques
 - **Choisir une licence qui respecte les contraintes** de toutes les bibliothèques

Problème de compatibilité avec une bibliothèque ?

- **Demande de nouvelle licence à l'auteur principal**
- **Changement de bibliothèque pour obtenir une bibliothèque compatible**

Quelles sont les licences compatibles ?

- Si vous agglutinez du code provenant de plusieurs programmes
 - Si tous ces programmes ont une **licence sans copyleft** (MIT, Apache-2.0, BSD-3-clause)
 - Vous pouvez licencier votre code sous **n'importe quelle licence** (libre ou privative)
=> en respectant les clauses de maintien des auteurs, copyright et notices de licence
 - Si un de ces programmes a une **licence à copyleft faible** (LGPL), les autres sans copyleft
 - Vous devez licencier votre code sous **licence à copyleft faible ou fort**
 - Si un de ces programmes a une **licence à copyleft fort** (EUPL-1.2)
 - Vous devez licencier votre code sous **licence à copyleft fort**
 - Si un de ces programmes a une **licence GPL-3.0-or-later** (copyleft fort / strict)
 - Vous devez licencier votre code sous **licence GPL-3.0-or-later**

Lorsque vous agglutinez des codes de provenances diverses :

- Toujours relicencier sous la licence à copyleft la plus restrictive
- Toujours conserver une trace des auteurs / copyright / notices des licences initiales

On a un problème avec le code issu des LLM (ChatGPT, etc) !

Quelles sont les licences compatibles ?

- **Si vous écrivez du code faisant appel à plusieurs bibliothèques**
 - Si toutes ces bibliothèques ont une **licence sans copyleft** (MIT, Apache-2.0, BSD-3-clause)
 - Vous pouvez licencier votre code sous **n'importe quelle licence** (libre ou privative)
=> en liant statiquement ou dynamiquement les bibliothèques à votre programme
 - Si une de ces bibliothèques a une **licence à copyleft faible** (LGPL), les autres sans copyleft
 - Vous pouvez licencier votre code sous **n'importe quelle licence** (libre ou privative)
=> en conservant la bibliothèque dans sa licence d'origine et en republiant ses modifs
 - Si une de ces bibliothèques a une **licence à copyleft fort** (EUPL-1.2)
 - Vous devez licencier votre code sous **licence à copyleft fort**
 - Si une de ces bibliothèques a une **licence GPL-3.0-or-later** (copyleft fort / strict)
 - Vous devez licencier votre code sous **licence GPL-3.0-or-later**

Lorsque vous faites appel à des diverses bibliothèques :

- Une bibliothèque à copyleft fort contaminera votre code (licence virale)
- Si pas de bibliothèque à copyleft fort, vous pouvez apposer n'importe quelle licence sur votre code

Comment ôter cette licence virale ?


- Si vous écrivez du code faisant appel à une bibliothèque sous licence à copyleft fort
 - Soit vous licenciez votre code sous cette même **licence à copyleft fort** (le plus simple !)
 - Vous liez statiquement ou dynamiquement la bibliothèque à votre programme
 - Vous publiez les modifications faites sur cette bibliothèque (sinon, un lien vers la bib)
 - Soit vous licenciez votre code sous une **licence moins restrictive mais**
 - **Vous ne pouvez pas lier la bibliothèque à votre programme !**
 - Vous devez déposer la bibliothèque dans un **dossier « THIRD_PARTY »** avec sa licence
 - La bibliothèque doit être **compilée séparément** à votre programme et produire un **outil externe (.exe)** et non une bibliothèque (.dll / .lib ou .so / .a interdit !)
 - Votre programme doit donc faire directement **appel à l'outil externe** généré par cette bibliothèque.
 - Vous devez indiquer que cet outil externe est une **dépendance obligatoire** pour le fonctionnement de votre programme
 - Si vous souhaitez **installer automatiquement** cet outil externe depuis l'installation de votre programme, vous devez
 - Afficher un **message clair** sur cet **outil externe tiers obligatoire**
 - **Faire valider** explicitement les **conditions d'utilisation de la licence** de cet outil
 - Et fournir a minima un **lien vers le code source** de cet outil

Exemple :
FFmpeg



Bruno MERCIER

11/03/2026



Ouvrons La Science de l'université de Poitiers
Il faut une licence sur mon code !

29

COMPATIBILITÉ

Comme

- **Si vous écrivez**

- **Soit vous**

- **V**

- **V**

- **Soit vous**

- **V**

- **V**

- **L**

- **e**

- **V**

- **b**

- **V**

- **f**

- **S**

- **V**

```
# License

Most files in FFmpeg are under the GNU Lesser General Public License version 2.1 or later (LGPL v2.1+). Read the file
`COPYING.LGPLv2.1` for details. Some other files have MIT/X11/BSD-style licenses. In combination the LGPL v2.1+ applies to
FFmpeg.
```

```
Some optional parts of FFmpeg are licensed under the GNU General Public License version 2 or later (GPL v2+). See the file
`COPYING.GPLv2` for details. None of these parts are used by default, you have to explicitly pass `--enable-gpl` to configure to
activate them. In this case, FFmpeg's license changes to GPL v2+.
```

```
Specifically, the GPL parts of FFmpeg are:
```

```
- optional x86 optimization in the files
  - `libavcodec/x86/flac_dsp_gpl.asm`
  - `libavcodec/x86/idct_mmx.c`
  - `libavfilter/x86/vf_removegrain.asm`
```

```
- [...]
- the following filters in libavfilter:
```

```
- `signature_lookup.c`
- `vf_blackframe.c`
- [...]
- `vf_vaguedenoiser.c`
- `vsrc_mptestsrc.c`
```

```
Should you, for whatever reason, prefer to use version 3 of the (L)GPL, then the configure parameter `--enable-version3` will
activate this licensing option for you. Read the file `COPYING.LGPLv3` or, if you have enabled GPL parts, `COPYING.GPLv3` to
learn the exact legal terms that apply in this case.
```

```
There are a handful of files under other licensing terms, namely:
```

```
* The files `libavcodec/jfdctfst.c`, `libavcodec/jfdctint_template.c` and `libavcodec/jrevdct.c` are taken from libjpeg, see the
top of the files for licensing details. Specifically note that you must credit the IJG in the documentation accompanying your
program if you only distribute executables.
```

```
You must also indicate any changes including additions and deletions to those three files in the documentation.
* `tests/reference.pnm` is under the expat license.
```

```
## External libraries
```

```
FFmpeg can be combined with a number of external libraries, which sometimes affect the licensing of binaries resulting from the
combination.
```

```
### Compatible libraries
```

```
The following libraries are under GPL version 2:
```

```
- avisynth
```

Exemple :
FFmpeg



Pour aller plus loin sur les licences logicielles

- Qu'est ce qu'une **licence ouverte**
 - <https://www.app.asso.fr/open-source/comprendre-l-open-source.html>
- Qu'est ce qu'une **licence propriétaire**
 - <https://cpl.thalesgroup.com/fr/software-monetization/proprietary-software-license>
- Fiche DORANum : **5 licences logicielles libres** (GPL, BSD, APACHE, MIT et CeCILL) - droits, obligations et **compatibilités des licences entre-elles**
 - https://doranum.fr/wp-content/uploads/fiche_synthetique_logiciels.pdf
- **Licences de logiciel libre compatibles avec la GPL**
 - <https://www.gnu.org/licenses/license-list.html#GPLCompatibleLicenses>
- **Licences de logiciel libre incompatibles avec la GPL**
 - <https://www.gnu.org/licenses/license-list.html#GPLIncompatibleLicenses>
- **Philosophie du logiciel libre**
 - <https://www.gnu.org/philosophy/free-sw.html#four-freedoms>

Outil en ligne pour choisir une licence en fonction de quelques critères simples

<https://choosealicense.com/>





COMPAT

Pour

- Qu'est
- Qu'est
- Fiche
- Licenc
- Licenc
- Philos

GNU GPLv3

Permissions of this strong copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.

Permissions	Conditions	Limitations
<ul style="list-style-type: none"> Commercial use Distribution Modification Patent use Private use 	<ul style="list-style-type: none"> Disclose source License and copyright notice Same license State changes 	<ul style="list-style-type: none"> Liability Warranty

[View full GNU General Public License v3.0 »](#)

GNU LGPLv3

Permissions of this copyleft license are conditioned on making available complete source code of licensed works and modifications under the same license or the GNU GPLv3. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. However, a larger work using the licensed work through interfaces provided by the licensed work may be distributed under different terms and without source code for the larger work.

Permissions	Conditions	Limitations
<ul style="list-style-type: none"> Commercial use Distribution Modification Patent use Private use 	<ul style="list-style-type: none"> Disclose source License and copyright notice Same license (library) State changes 	<ul style="list-style-type: none"> Liability Warranty

[View full GNU Lesser General Public License v3.0 »](#)

Mozilla Public License 2.0

Permissions of this weak copyleft license are conditioned on making available source code of licensed files and modifications of those files under the same license (or in certain cases, one of the GNU licenses). Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. However, a larger work using the licensed work may be distributed under different terms and without source code for files added in the larger work.

Permissions	Conditions	Limitations
<ul style="list-style-type: none"> Commercial use Distribution Modification Patent use Private use 	<ul style="list-style-type: none"> Disclose source License and copyright notice Same license (file) 	<ul style="list-style-type: none"> Liability Trademark use Warranty

[View full Mozilla Public License 2.0 »](#)

le L) - droits,

bles

Out



Pour aller plus loin sur les licences logicielles

- Qu'est ce qu'une **licence ouverte**
 - <https://www.app.asso.fr/open-source/comprendre-l-open-source.html>
- Qu'est ce qu'une **licence propriétaire**
 - <https://cpl.thalesgroup.com/fr/software-monetization/proprietary-software-license>
- Fiche DORANum : **5 licences logicielles libres** (GPL, BSD, APACHE, MIT et CeCILL) - droits, obligations et **compatibilités des licences entre-elles**
 - https://doranum.fr/wp-content/uploads/fiche_synthetique_logiciels.pdf
- **Licences de logiciel libre compatibles avec la GPL**
 - <https://www.gnu.org/licenses/license-list.html#GPLCompatibleLicenses>
- **Licences de logiciel libre incompatibles avec la GPL**
 - <https://www.gnu.org/licenses/license-list.html#GPLIncompatibleLicenses>
- **Philosophie du logiciel libre**
 - <https://www.gnu.org/philosophy/free-sw.html#four-freedoms>

Outil en ligne pour choisir une licence en fonction de quelques critères simples

<https://choosealicense.com/>

Autre outil en ligne : celui de l'ANCT – Agence Nationale de la Cohésion des Territoires

<https://licence-libre.incubateur.anct.gouv.fr/licence>

← Précédent

Étape 1 sur 2

Décrivez votre situation

Étape suivante : Choisissez votre licence

Mon projet ...



A vocation à créer une application autonome

Elle n'est pas créée pour servir de base à d'autres développements ou projets

Ex : Entourage Réseaux Solidaire

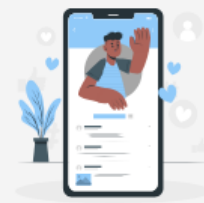


A vocation à être réutilisé par d'autres projets pour créer des logiciels/services

Ex : le design système de l'Etat



A vocation à créer une extension, une "surcouche" d'un projet existant



Permet de fournir un service au travers d'un site ou d'une application web

Pour aller plus loin sur les licences logicielles

- Qu'est ce qu'une **licence ouverte**
 - <https://www.app.asso.fr/open-source/comprendre-l-open-source.html>
- Qu'est ce qu'une **licence propriétaire**
 - <https://cpl.thalesgroup.com/fr/software-monetization/proprietary-software-license>
- Fiche DORANum : **5 licences logicielles libres** (GPL, BSD, APACHE, MIT et CeCILL) - droits, obligations et **compatibilités des licences entre-elles**
 - https://doranum.fr/wp-content/uploads/fiche_synthetique_logiciels.pdf
- **Licences de logiciel libre compatibles avec la GPL**
 - <https://www.gnu.org/licenses/license-list.html#GPLCompatibleLicenses>
- **Licences de logiciel libre incompatibles avec la GPL**
 - <https://www.gnu.org/licenses/license-list.html#GPLIncompatibleLicenses>
- **Philosophie du logiciel libre**
 - <https://www.gnu.org/philosophy/free-sw.html#four-freedoms>

Outil en ligne pour choisir une licence en fonction de quelques critères simples

<https://choosealicense.com/>

Autre outil en ligne : celui de l'ANCT – Agence Nationale de la Cohésion des Territoires

<https://licence-libre.incubateur.anct.gouv.fr/licence>

« En tant qu'organisme public, dois-je publier ce code source ? »

Guide juridique interactif sur la publication des codes sources de l'administration.

Le code source contient-il des secrets faisant obstacle à la communication du logiciel ?

La présence de secrets dans un code source peut empêcher la publication du logiciel associé.

Il peut s'agir d'un secret absolu : certains logiciels, dans l'intérêt public, ne peuvent être rendus publics par vous, notamment pour des questions sécuritaires et de sûreté nationale.

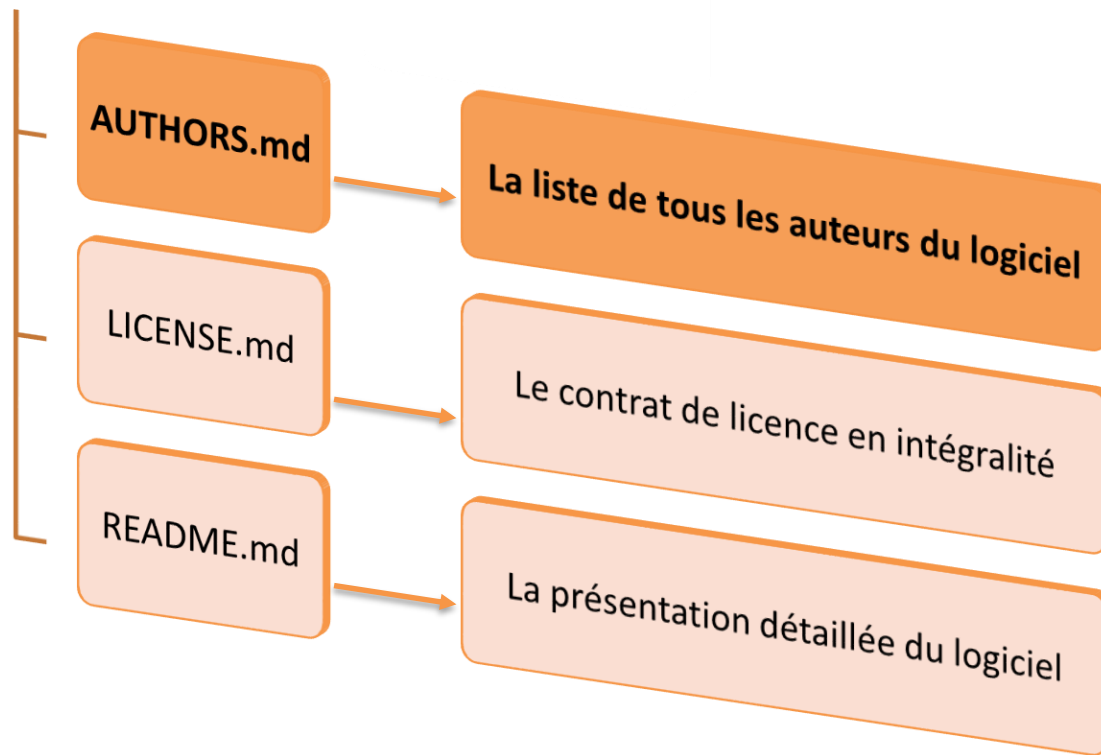
Le secret peut aussi être relatif, comme des informations relatives à la vie privée d'une personne, des informations comportant une appréciation ou un jugement de valeur sur une personne physique ou des informations commerciales et industrielles risquant de porter préjudice à un acteur économique. Dans ce cas, le code source du logiciel n'est communicable qu'aux personnes pour qui ces informations n'ont pas à rester secrètes.

Non

Oui

Poser 3 fichiers à la racine de votre projet

RootDir



Comment lister les auteurs ?

- Identifier toutes les **personnes créditées dans le code source**
 - Regarder les **entêtes de tous vos fichiers** de code source (scripts, classes, etc)
 - Et récupérer le nom des **personnes ayant le plus codé...** mais pas que !
- Les personnes qui sont notées dans le fichier AUTHORS.md ont les **droits de créateurs** sur le code source / le logiciel
 - Les **auteurs** d'un logiciel sont les personnes qui ont **créé** le logiciel **initialement** ou qui l'ont **fortement fait évoluer**. Cela concerne :
 - **la gestion du projet** (le porteur qui gère les finances, les achats de matériel et les ressources humaines, et/ou l'encadrant principal qui gère la planification et la coordination des différentes tâches),
 - **la conception** (les chercheurs qui apportent leur contribution scientifique),
 - **le développement** (les doctorants et les stagiaires),
 - **et la création du logiciel dans son ensemble** (équipe technique en support au déploiement et à la maintenance).

Les auteurs

Sont considérés comme les propriétaires intellectuels du logiciel

Ne pas oublier les contributeurs !

- Identifier toutes les **personnes créditées dans le code source**
 - Regarder les **entêtes de tous vos fichiers** de code source (scripts, classes, etc)
 - Et récupérer le nom de **toutes les personnes** qui y apparaissent !
- Les personnes qui sont notées dans le fichier AUTHORS.md ont les **droits de créateurs** sur le code source / le logiciel **sauf les contributeurs** !
 - Les **contributeurs** d'un logiciel sont les personnes qui apportent des modifications, des améliorations ou des ajouts au logiciel **après sa création initiale**. Cela concerne :
 - Le développement de **fonctionnalités supplémentaires**,
 - La **correction de bugs**,
 - L'optimisation de **performances**,
 - L'apport de toute **autre contribution** au projet (script d'install, modèle de **site web**, etc).

Les contributeurs

Ne peuvent pas prétendre à la prime d'intéressement
Mais conservent leur droit d'auteur (paternité)

Contenu du fichier **AUTHORS.txt**

- **Auteurs**
 - **Prénom NOM (email institutionnel), fonction (Service/Composante/Unité, Employeur)**
[Taux global de participation au projet en %age]
 - Autant de lignes que de personnes identifiées comme **auteurs**
- **Contributeurs**
 - **Prénom NOM (email institutionnel), fonction (Service/Composante/Unité, Employeur)**
 - Autant de lignes que de personnes identifiées comme **contributeurs**

Les auteurs

Le taux global de participation au projet n'est pas forcément mentionné
(mentionné plus tard, lorsque le logiciel sera suffisamment abouti)

La somme globale des taux de participation ne doit pas excéder 100% !

Exemple de fichier **AUTHORS.md**

- **Auteurs**
 - Noël RICHARD (noel.richard@univ-poitiers.fr), MCF (XLIM, Université de Poitiers) [30%]
 - Michael NAUGE (michael.nauge@univ-poitiers.fr), PostDoc (XLIM, Univ de Poitiers) [20%]
 - Bruno MERCIER (bruno.mercier@univ-poitiers.fr), IGR (XLIM, Université de Poitiers) [50%]
- **Contributeurs**
 - Alexandre BONY (alexandre.bony@univ-poitiers.fr), PostDoc (XLIM, Université de Poitiers)
 - Florence BOURET (florence.bouret@chanel-corp.com), Esthéticienne (Département Evaluation Efficacité et Prospective, CHANEL)

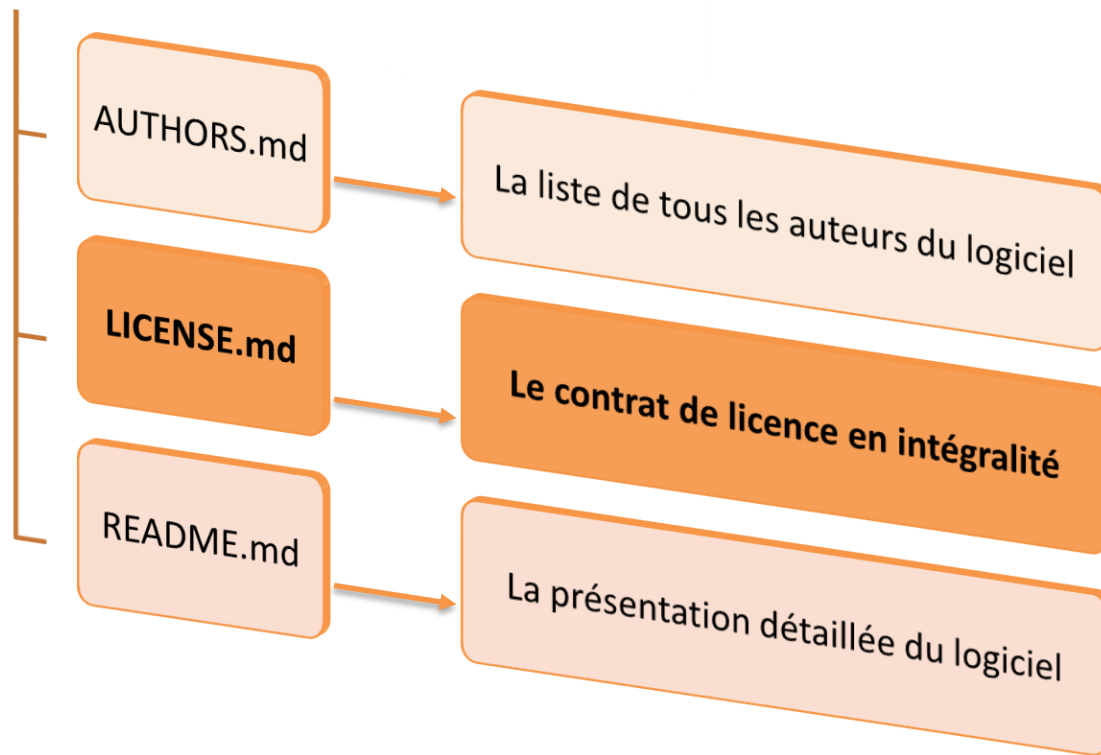
Les contributeurs
Peuvent être déportés dans un fichier séparé que l'on appellera
CONTRIBUTORS.md

Les droits patrimoniaux (d'exploitation) sont détenus exclusivement par
Université de Poitiers (100%)



Poser 3 fichiers à la racine de votre projet

RootDir



Rappel sur le choix d'une licence

- Bien que la **licence** devrait être **choisie par les tutelles**, en pratique :
 - Le déposant choisit la licence
 - **Le déposant est responsable de la compatibilité entre licences** (entre le code déposé et ses dépendances)
- Par conséquent : **ne pas la choisir seul**, se faire accompagner par une personne en appui aux dépôts de logiciels (sauf vous à l'issue de la formation : enfin j'espère !)
 - au sein de son laboratoire (Réfèrent « Science Ouverte »)
 - de sa tutelle :
 - **Licence propriétaire** : [SPVR](#) (Service du Partenariat et de la Valorisation de la Recherche)
 - **Licence libre** : Pilier « [Codes sources et logiciel](#) » de la Science Ouverte à l'UP

Ne pas tenter d'écrire sa propre licence !

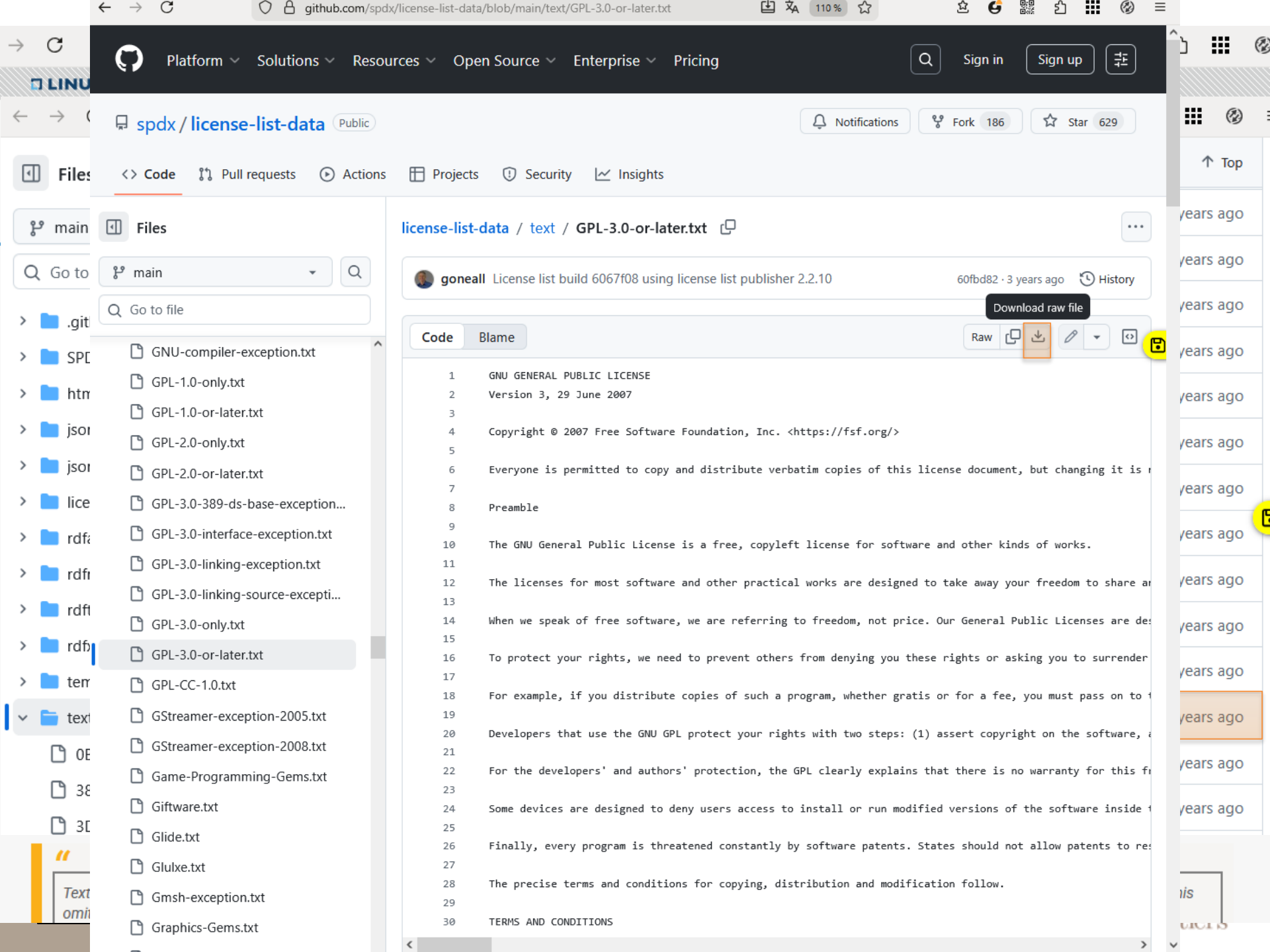
**Il y a suffisamment de « licences libres »
Pour en trouver une qui corresponde à vos besoins**



Contenu du fichier **LICENSE.txt**

- Choisir de préférence une licence **approuvée** et **par la FSF et par l'OSI**
 - **Cette approbation FSF / OSI** est visible sur le site **spdx.org**
 - <https://spdx.org/licenses/>
 - Cela permet également de visualiser le contenu des licences au format HTML
 - Et de récupérer l'**identifiant unique de chaque licence**
- Copier le **texte intégral du fichier de la licence choisie**
 - Ce texte au **format TXT** peut être recopié directement depuis cette page **GitHub**
 - <https://github.com/spdx/license-list-data/tree/main/text>
 - Ne pas mettre le lien vers le texte de la licence choisie mais **copier le texte !**

Est-ce que ce fichier LICENSE.md est suffisant ?
Pour faire simple : OUI !



Exemple de fichier **LICENSE.txt**

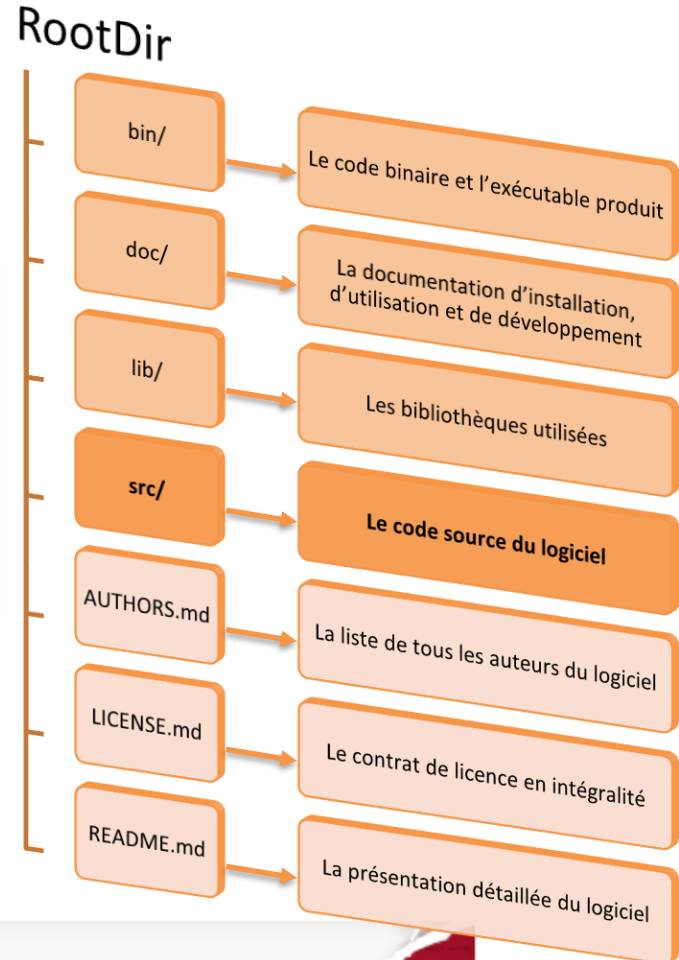
- On veut poser une **licence GPL** !
- Vérifier que la **licence GPL** est **approuvée** et **par la FSF** et **par l'OSI**
 - Sur le site <https://spdx.org/licenses/>
 - Cela permet également de visualiser le contenu des licences au format HTML
=> ne pas recopier ce format HTML
=> mieux vaut récupérer le format TXT (texte brut) ou le format MD (markdown)
- Noter l'**identifiant de la licence** (la dernière version approuvée FSF & OSI)
 - **GPL-3.0-or-later**
- Télécharger le **fichier de licence** (le fichier nommé « **GPL-3.0-or-later.txt** »)
 - Depuis la page <https://github.com/spdx/license-list-data/tree/main/text>
 - Le renommer en « **LICENSE.txt** »

Est-ce que ce fichier LICENSE.md est suffisant ?
Pour être complet : **NON !**

Où doit-on encore enregistrer cette licence ?

- A la racine du projet
 - Ici, dans le **fichier « LICENSE.md »**
 - Le **texte complet de la licence**

- Et dans l'entête de chaque fichier source
 - Ici, dans le **dossier « src »**
 - Sous forme de commentaires dans le code
 - L'**identifiant de la licence** au format SPDX
 - Les **auteurs** (nom + email)
 - Le **copyright** (année + les employeurs)



Où doit-on encore enregistrer cette licence ?

- Exemple d'entête d'un fichier « C++ »
 - Avec des commentaires « doxygen »

```
/**
 * \file      RawOpen.h
 * \author    Alexandre BONY (alexandre.bony@univ-poitiers.fr)
 * \author    Bruno MERCIER (bruno.mercier@univ-poitiers.fr)
 * \version   1.1
 * \date      2016-06-27
 * \copyright ©2014 Laboratoire XLIM, Université de Poitiers, CNRS, Université de Limoges
 *            Logiciel "Akhenatongle" sous licence GPL-3.0-or-later
 * \brief     Définition d'une Class CRawOpen.
 * \details   Cette classe permet de gérer l'ouverture des fichiers images au format NEF
 *            (RAW 12 et 14 bits de Nikon).
 *            Les méthodes sont implémentées dans le fichier "RawOpen.cpp".
 */
```

Champ « \copyright » sur 2 lignes :

- Année de création du fichier + nom du laboratoire + liste des tutelles
- (Nom du logiciel) + Identifiant de la licence

Et si j'ai des dizaines de fichiers à modifier ?

- Il existe un **outil** pour faire cela **automatiquement**
 - **REUSE SOFTWARE** : <https://reuse.software/fr/>
 - **Tutoriel** : <https://reuse.software/tutorial/>
 - **Installation** locale via python : « `python3 -m pip install reuse` »
 - installé dans « `~/local/bin/reuse` »
 - **Exemple concret**
 - **Pour ajouter un entête à tous mes fichiers « *.h » et « .cpp » du dossier « src »**

```
reuse annotate --contributor="Alexandre BONY <alexandre.bony@univ-poitiers.fr>" --  
contributor="Bruno MERCIER <bruno.mercier@univ-poitiers.fr>" --year="2014" --copyright="(c)  
Laboratoire XLIM, Université de Poitiers, CNRS, Université de Limoges" --license="GPL-3.0-  
or-later" src/*.h src/*.cpp
```

- **Voici l'entête de mes fichiers « *.h » et « .cpp » du dossier « src »**

```
/*  
 * SPDX-FileCopyrightText: 2014 (c) Laboratoire XLIM, Université de Poitiers, CNRS,  
Université de Limoges  
 * SPDX-FileContributor: Alexandre BONY <alexandre.bony@univ-poitiers.fr  
 * SPDX-FileContributor: Bruno MERCIER <bruno.mercier@univ-poitiers.fr>  
 *  
 * SPDX-License-Identifier: GPL-3.0-or-later  
 */
```

Et si j'ai des dizaines de fichiers à modifier ?

- Second exemple
 - Pour ajouter un entête à tous mes fichiers « *.py » du dossier « src »

```
reuse annotate --contributor="Alexandre BONY <alexandre.bony@univ-poitiers.fr>" --  
contributor="Bruno MERCIER <bruno.mercier@univ-poitiers.fr>" --year="2014" --copyright="(c)  
Laboratoire XLIM, Université de Poitiers, CNRS, Université de Limoges" --license="GPL-3.0-  
or-later" src/*.py
```

- Voici l'entête de mes fichiers « *.py » du dossier « src »

```
# SPDX-FileCopyrightText: 2014 (c) Laboratoire XLIM, Université de Poitiers, CNRS,  
Université de Limoges  
# SPDX-FileContributor: Alexandre BONY <alexandre.bony@univ-poitiers.fr>  
# SPDX-FileContributor: Bruno MERCIER <bruno.mercier@univ-poitiers.fr>  
#  
# SPDX-License-Identifier: GPL-3.0-or-later
```

Reuse

Adapte sa syntaxe de commentaires en fonction de l'extension du fichier

Et si j'ai des dizaines de fichiers à modifier ?

- Pour vérifier que tout mon dossier « src » est sous licence (analyse statique)

```
cd src  
reuse lint
```

- Voici le résultat affiché

RÉSUMÉ

```
* Mauvaises licences : 0  
* Licences obsolètes : 0  
* Licences sans extension de fichier : 0  
* Licences manquantes : GPL-3.0-or-later  
* Licences inutilisées : 0  
* Licences utilisées : GPL-3.0-or-later  
* Erreurs de lecture : 0  
* Files with copyright information: 2 / 2  
* Files with license information: 2 / 2
```

Malheureusement, votre projet n'est pas conforme à la version 3.2 de la spécification REUSE
:- (

Et si j'ai des dizaines de fichiers à modifier ?

- Pour télécharger la licence manquante

```
cd src  
reuse download --all
```

- Voici le résultat affiché

```
Téléchargement de ./LICENSES/GPL-3.0-or-later.txt terminé.
```

Reuse

Permet de récupérer automatiquement les textes des licences posées dans les entêtes de votre code source

- Voici une autre façon de poser le fichier de licence

```
cp LICENSES/GPL-3.0-or-later.txt ../LICENSE.txt
```

Et si j'ai des dizaines de fichiers à modifier ?

- Nouvelle vérification que tout mon dossier « src » est sous licence (analyse statique)

```
cd src  
reuse lint
```

- Voici le résultat affiché

RÉSUMÉ

```
* Mauvaises licences : 0  
* Licences obsolètes : 0  
* Licences sans extension de fichier : 0  
* Licences manquantes : 0  
* Licences inutilisées : 0  
* Licences utilisées : GPL-3.0-or-later  
* Erreurs de lecture : 0  
* Files with copyright information: 2 / 2  
* Files with license information: 2 / 2
```

**Vous pouvez donc détruire
le dossier « src/LICENSES »**

Félicitations ! Votre projet est conforme à la version 3.2 de la spécification REUSE :-)

Tout mon code est-il enfin sous licence ?

- Si vous avez utilisé « reuse lint » sur un **logiciel complet**
 - Il est probable qu'il vous réponde **NON !**

- **Pourquoi ?**
 - Car il faut mettre une ou plusieurs **autres licences sur ce qui n'est pas VOTRE code** mais **embarqué dans votre logiciel / projet**
 - sur les **bibliothèques de votre logiciel**
(si fournies et liées statiquement ou dynamiquement à votre exécutable)
 - Sous-dossier « **lib** » pour les bibliothèques (libraries)

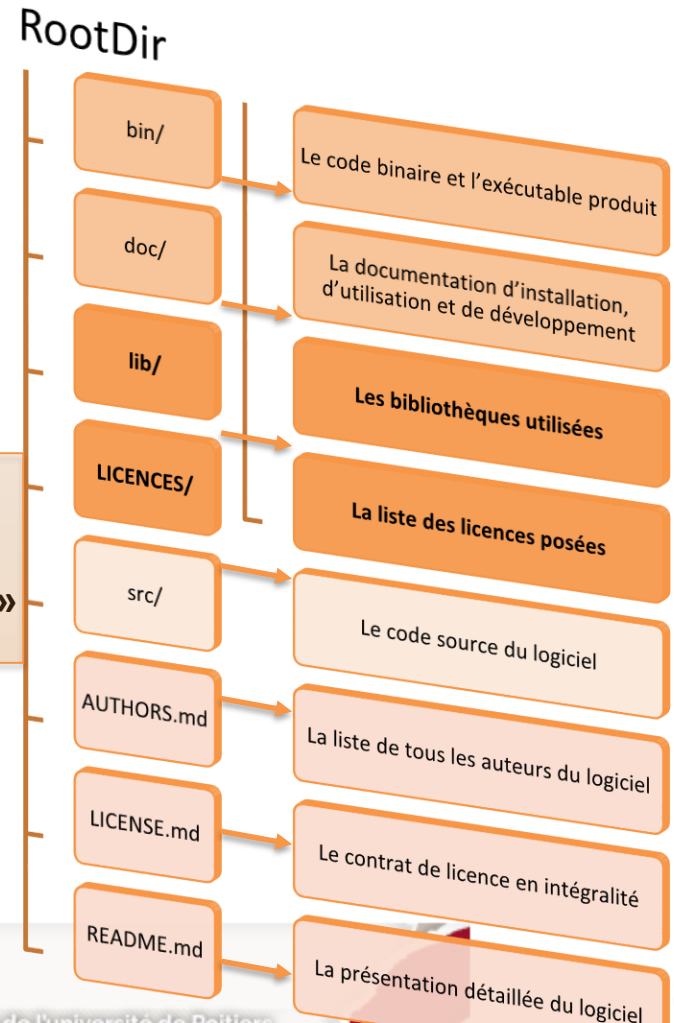
- **Comment faire ?**
 - Créer un **sous-dossier « LICENSES »** dans « RootDir » (comme avait fait l'outil « reuse » !)
 - Inclure le fichier de **licence de chaque bibliothèque** dans ce sous-dossier

Séparer les éléments différents dans des dossiers différents vous aidera à ne pas poser n'importe quelle licence avec l'outil « reuse »

Où doit-on enregistrer les licences ?

- Celle de **votre code source**
 - **A la racine** du projet
 - Dans le **fichier « LICENSE.md »**
 - Le **texte complet de la licence**
 - **Et dans l'entête** de chaque fichier source
 - Dans le **dossier « src »**
 - L'**identifiant de la licence**

- Et celles de **chaque bibliothèque**
 - Regarder dans le **dossier « lib »**
 - **Et poser les licences** dans le **dossier « LICENSES »**



Licences de mes bibliothèques

- Si vous avez inclus le **code source de certaines bibliothèques**
 - Vous pouvez **réutiliser** « reuse download --all » dans chacune des bibliothèques
 - Sinon, **recupérer le fichier de licence** sur le site de la bibliothèque

- **Exemple de ce que vous devez avoir**

```
lib/qt6/  
lib/opencv3/  
LICENSES/LICENSE_lib_qt6.txt  
LICENSES/LICENSE_lib_opencv3.txt  
LICENSE.txt
```

Difficile d'insérer l'identifiant de la licence
car certaines bibliothèques
ont leur propre modèle de licence

Conserver le fichier de licence de chaque bibliothèque dans le dossier
« LICENSES »

Et le nommer avec « lib » + le nom de la bibliothèque
(ou du sous-dossier contenant la bibliothèque)
+ le fichier « LICENSE.txt » de votre propre code source

Tout mon code est-il enfin sous licence ?

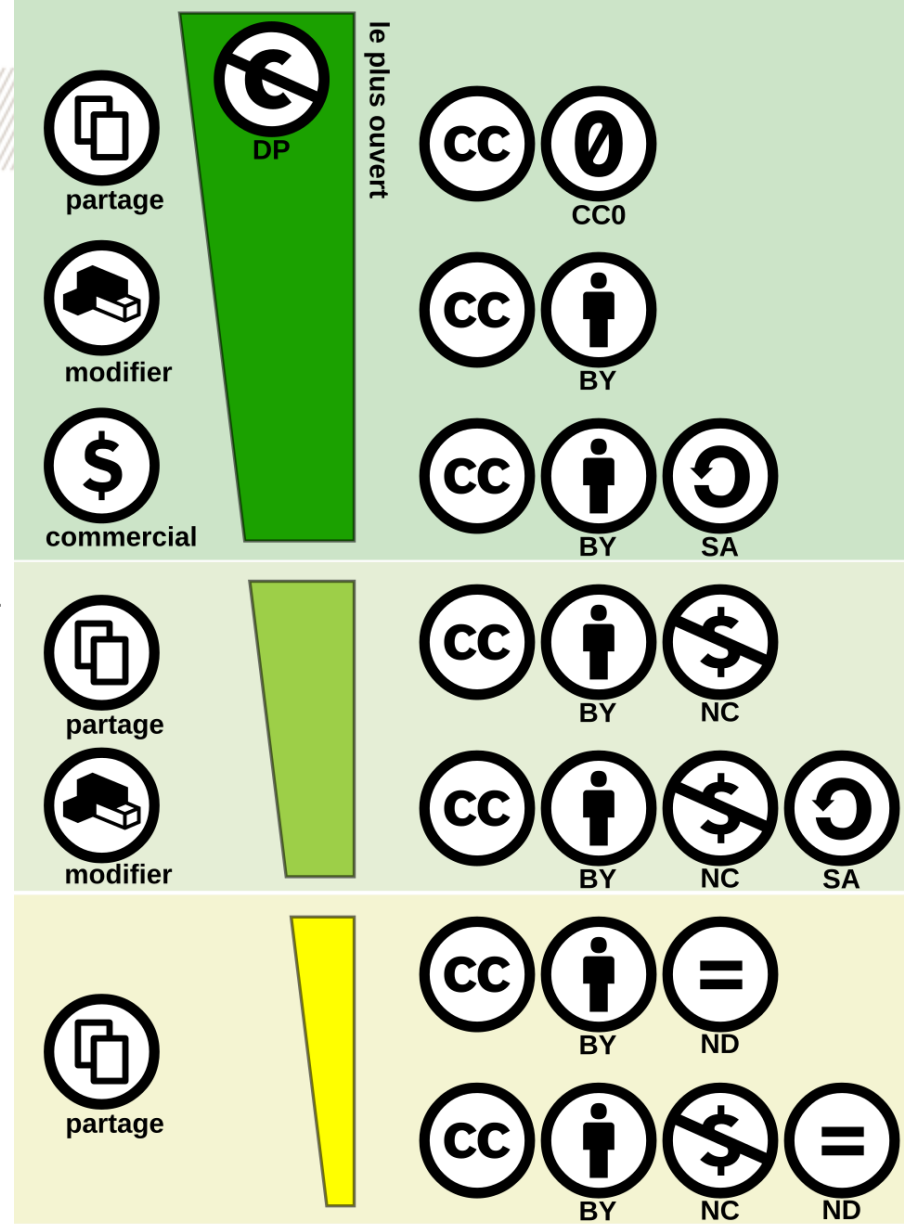
- Si vous avez utilisé « reuse lint » sur un **logiciel complet**
 - Il est probable qu'il vous réponde **NON !**
- **Pourquoi ?**
 - Car il faut mettre une ou plusieurs **autres licences sur ce qui n'est pas *DU* code** mais **embarqué dans votre logiciel / projet** (tout ce qui est livré dans « **RootDir** »)
 - sur les **images de votre logiciel** (boutons de votre interface graphique)
 - Sous-dossier « **res** » pour les ressources (peut-être un sous-dossier de « **src** »)
 - sur toute votre **documentation**
 - Sous-dossier « **doc** »
 - et sur vos **jeux de données** fournis à titre d'exemples dans vos tutoriels
 - Sous-dossier « **data** » (peut-être un sous-dossier de « **doc** »)

Séparer les éléments différents dans des dossiers différents vous aidera à ne pas poser n'importe quelle licence avec l'outil « reuse »

Ne pas mettre une licence « logicielle »
sur ces éléments !


Les licences CC Creative Commons

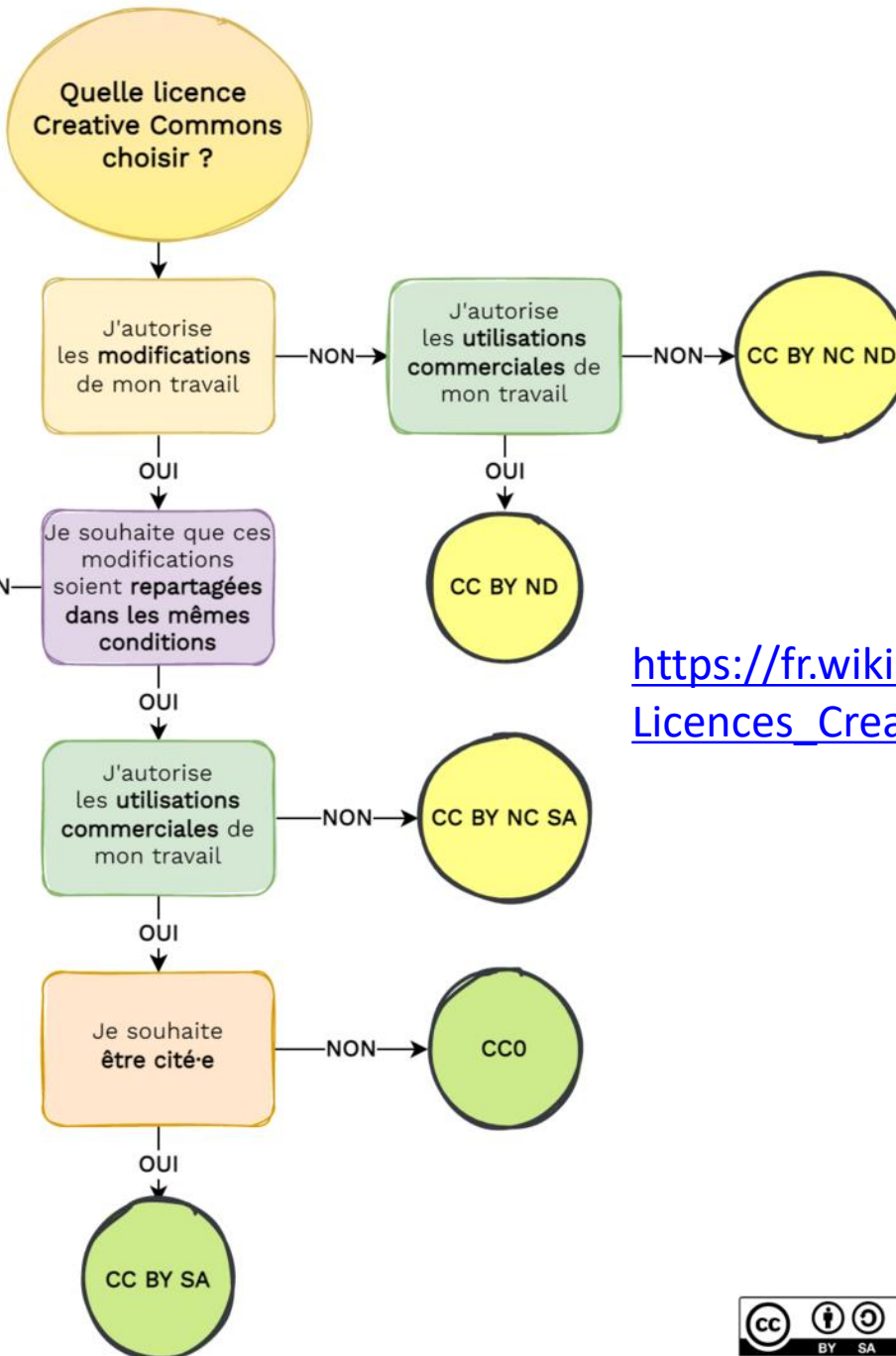
- Licences de **libre diffusion** d'une œuvre
 - les plus fréquemment utilisées !
 - juridiquement valables dans le monde entier depuis la **version 4.0**
- Les différents **sigles**
 - **CC0** => ~ **Domaine public** ~
 - Non compatible avec le droit français
 - **BY** => **Obligation** de citer l'auteur
 - Permet la traçabilité !
 - **SA** => **Obligation** de conserver la **même licence** après dérivation c'est appliquer un « **Copyleft strict** »
 - **NC** => **Interdiction** de commercer
 - **ND** => **Interdiction** de dériver / modifier



Compatibilités entre les différentes licences CC sur DORANum

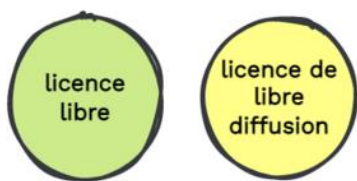
https://doranum.fr/wp-content/uploads/fiche_synthetique_Creative_Commons.pdf


 Condition préalable pour l'utilisation d'une licence Creative Commons : être d'accord que d'autres personnes copient et distribuent mon travail sans m'en demander la permission



https://fr.wikipedia.org/wiki/Licences_Creative_Commons

CC = (licence) Creative Commons
 BY = attribution
 NC = pas d'utilisation commerciale
 ND = pas de modification
 SA = partage dans les mêmes conditions



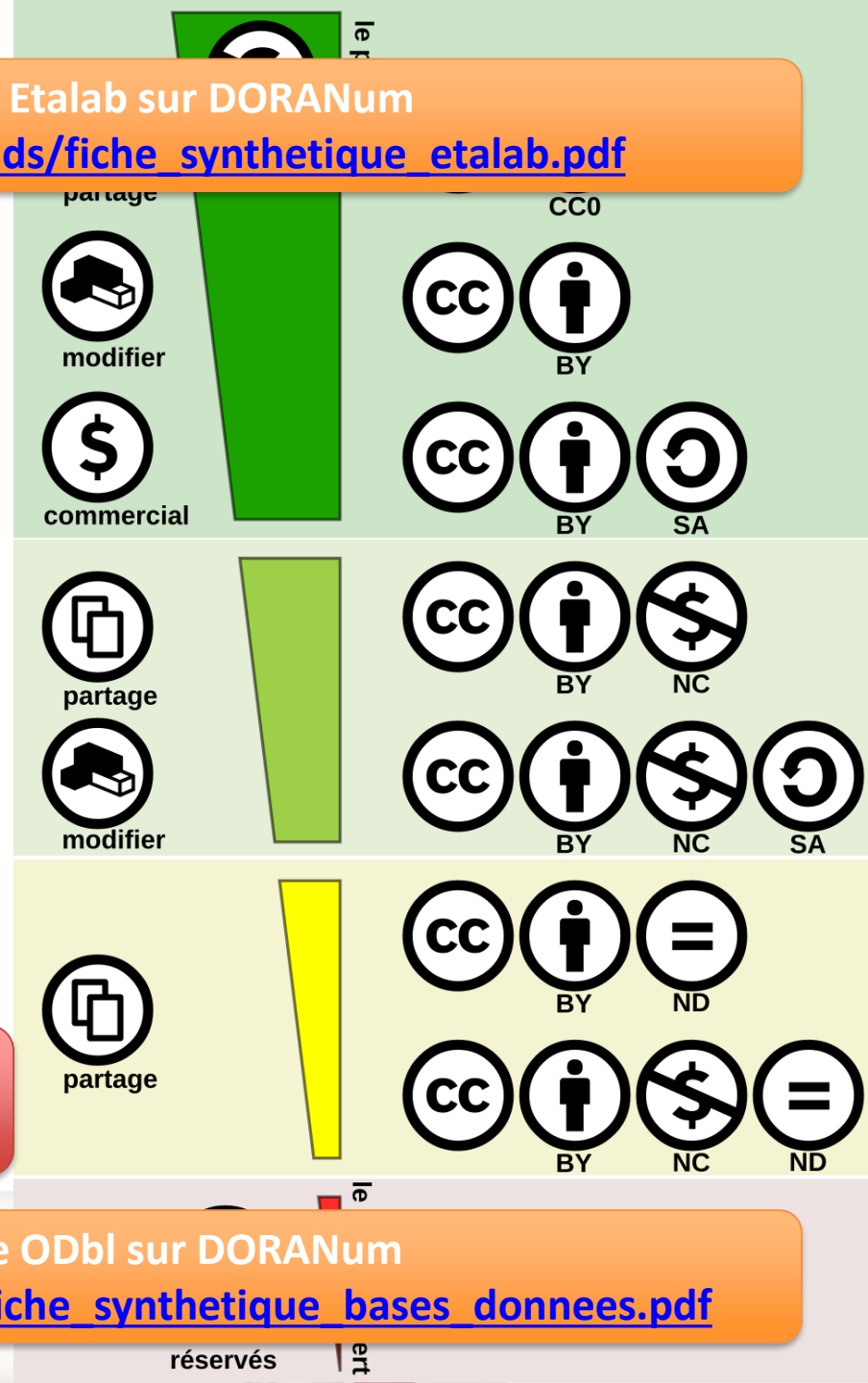
Compatibilités de la licence Etalab sur DORANum

https://doranum.fr/wp-content/uploads/fiche_synthetique_etalab.pdf

Les autres licences libres

- La licence **Etalab**
 - Établie par le gouvernement français
 - Sa version 2 :
 - « **etalab-2.0** »
 - Compatible et équivalente à « **CC-BY-4.0** »
 - Utilisée dans les administrations
 - Moins connue internationalement
- La licence « **ODbl-1.0** »
 - Reconnue par le gouvernement français au même titre que « **etalab-2.0** »
 - Diffusion de **Bases de données** avec obligation de **partage à l'identique**

Une base de données peut avoir une licence
Et ses données avoir des licences différentes !



Compatibilités de la licence ODbL sur DORANum

https://doranum.fr/wp-content/uploads/fiche_synthetique_bases_donnees.pdf

Base de données sous licence « ODb1-1.0 »

- Seule la base de données est sous licence « ODb1-1.0 »
- Les données peuvent être sous une autre licence

- Par dossier
- Par extension de fichiers
- Par fichier

Les données factuelles
(Techniques, Mesures machine, etc)
ne peuvent pas être protégées
par le droit d'auteur (pas d'originalité)

- Si pas d'autres fichiers de licence que celui de la base de données

- Les données factuelles sont sous « ODb1-1.0 »
- Les données d'œuvres créatives sont dans une situation ambiguë

Seules les données provenant d'œuvres
créatives peuvent être sous licence
(Photos, Textes, Images, Sons, Code source)

Ajouter une licence « CC-xx »
pour les œuvres créatives

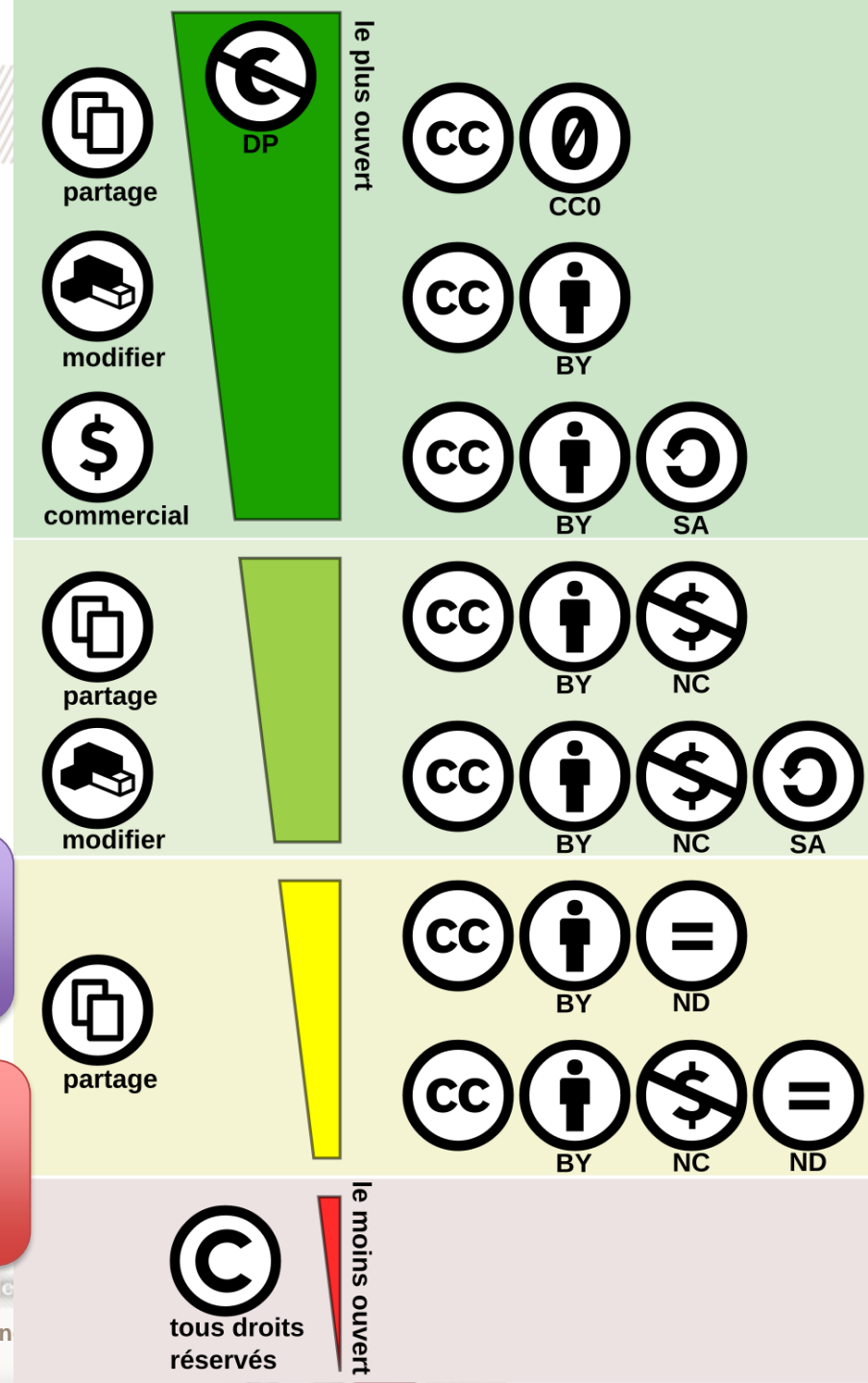


Quelle licence choisir ?

- « **CC-BY-4.0** »
 - Facile de compiler des jeux de données
 - Conservation de la traçabilité
- « **CC-BY-NC-SA-4.0** »
 - Pas de commerce
 - On peut modifier (corriger une coquille)
 - On peut compléter (jeux de données)
 - Avec des data « CC-BY-4.0 »
 - Et licencier le tout « CC-BY-NC-SA-4.0 »

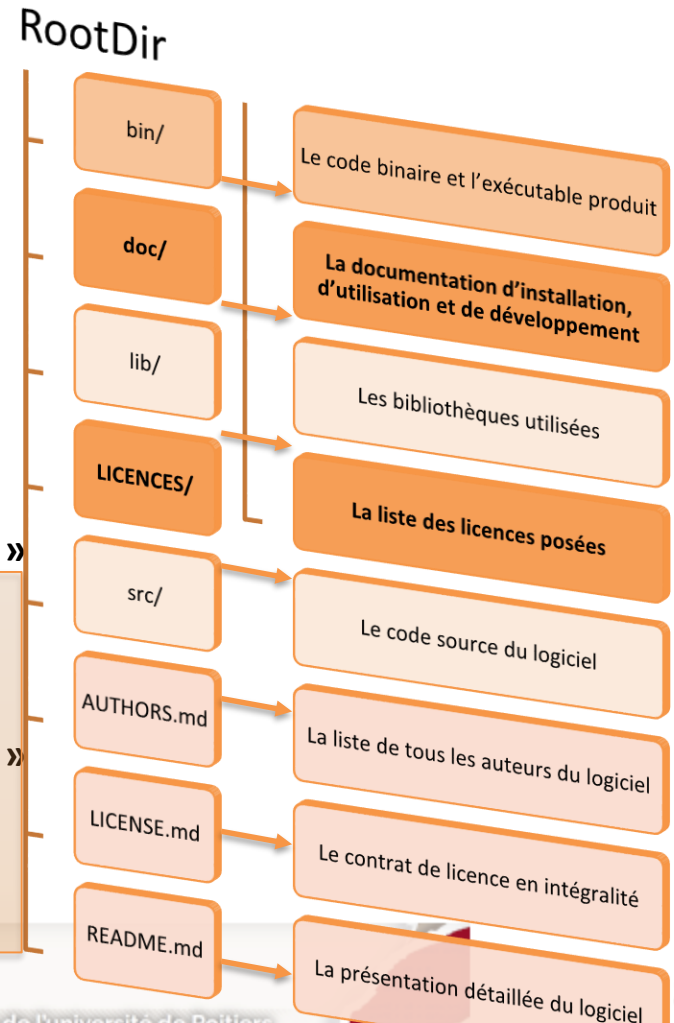
Le « SA » évite que quelqu'un ajoute « ND »,
Car dans ce cas :
Plus de modification, plus de complétion

Ne pas utiliser « CC-BY-SA-4.0 »,
Car dans ce cas :
Pas de complétion avec data « CC-BY-NC-SA-4.0 »



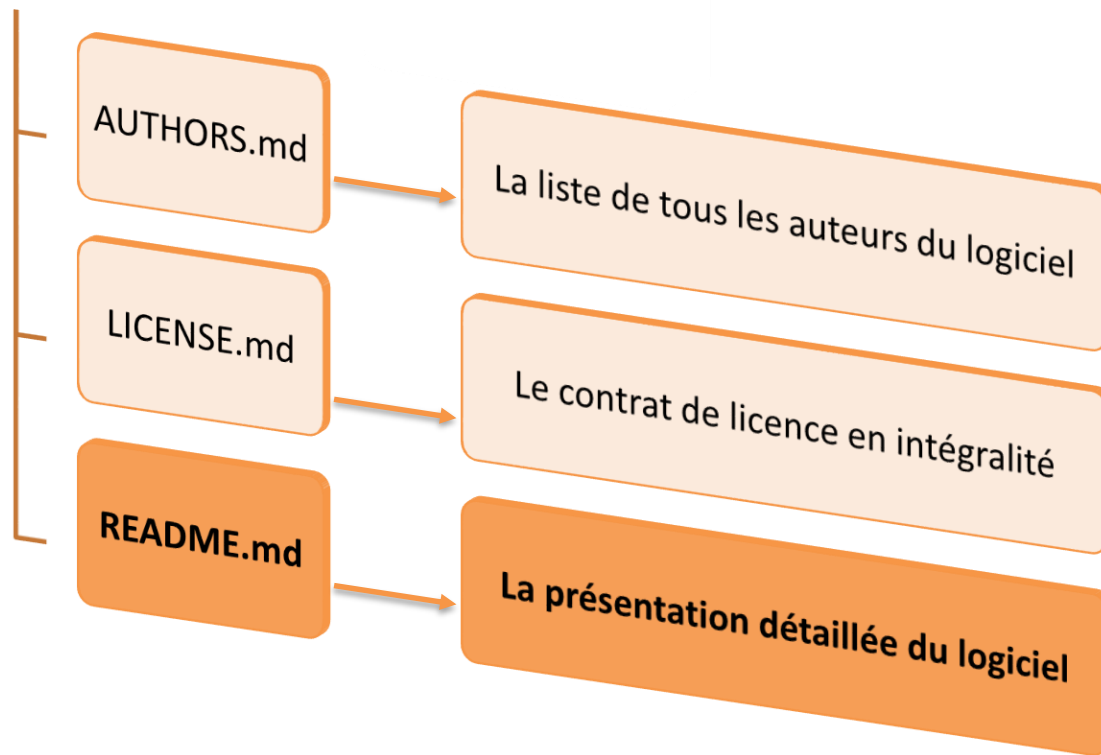
Où doit-on enregistrer les licences ?

- Celle de **votre code source**
 - **A la racine** du projet
 - Dans le **fichier « LICENSE.md »**
 - Le **texte complet de la licence**
 - **Et dans l'entête** de chaque fichier source
 - Dans le **dossier « src »**
 - L'**identifiant de la licence**
 - Et celles de **chaque bibliothèque**
 - Regarder dans le **dossier « lib »**
 - **Et poser les licences dans le dossier « LICENSES »**
 - Et celles de la **documentation**
 - Regarder dans le **dossier « doc »**
 - **Estampiller** certains **documents (pdf, txt)**
 - **Et poser les licences dans le dossier « LICENSES »**
 - Fichier **« LICENSE_doc.txt »**
 - Fichier **« LICENSE_doc_data.txt »**
 - Fichier **« LICENSE_src_res.txt »**



Poser 3 fichiers à la racine de votre projet

RootDir



Fichier de présentation de votre logiciel

- **Premier fichier créé lorsqu'on ouvre un dépôt sur une forge logicielle**
 - **Beaucoup de choses** peuvent figurer dans ce fichier mais celui-ci doit **rester clair et concis**
- **Liste minimale**
 - **Nom et description du logiciel**
 - Exigences du système
 - **Instructions d'installation**
 - Guide de démarrage rapide
 - **Fonctionnalités principales**
 - Exemples d'utilisation
 - **Licence du logiciel**
 - **Auteurs et crédits**
 - **État de développement**
 - **Informations de contact / support**
 - FAQ (Foire Aux Questions)



Exemple de fichier « README.md »

- **Nom et description du logiciel**

- **Akhenatongle** est un logiciel développé sous windows pour Chanel permettant d'analyser automatiquement la tenue de vernis à ongles dans leur centre d'expertise à partir d'une série de photographies importées au format **NEF (RAW)** de chez Nikon). Il produit des résultats exportables au format **CSV**

- **Instructions d'installation**

- Exécutable auto-installable sous **Windows 7 64bits** ou ultérieur
- Ne pas refuser la demande d'installation éventuelle des **redistribuables Visual C++**
- Ne pas refuser la demande de désinstallation automatique d'une précédente version d'**Akhenatongle**

Renvoyez vers des pages dédiées
lorsque vous avez beaucoup de choses à écrire (+ de 20 lignes /section).
Si beaucoup de cas particuliers pour l'installation, renvoyez vers
`INSTALL.md`

Exemple de fichier « README.md »

- **Fonctionnalités principales**

- **Chargement** d'une campagne d'acquisition composée de **photographies au format NEF** (chaque photographie doit contenir 4 doigts ou 1 pouce) rangées dans des dossiers (1 dossier par paneliste)
- **Extraction automatique des contours vernis et affichage** sous forme d'un **masque** venant couvrir **par transparence** l'ongle verni sur chaque photographie
- **Sauvegarde des statistiques moyennes** (par paneliste et par verni) sur la **couverture** et l'**uniformité chromatique** sous forme d'un **fichier CSV**



Exemple de fichier « README.md »

- **Licence du logiciel**
 - **Licence propriétaire** : voir le fichier **LICENSE.md**
 - Le logiciel **Akhenatongle** utilise les **bibliothèques dynamiques** suivantes
 - **Qt 5.7.0 64bits** sous licence **LGPL-3.0-only** : voir le fichier **LICENSE_lib_qt.txt**
 - **OpenCV 3.1.0 64bits** sous licence **BSD-3-Clause** : voir le fichier **LICENSE_lib_opencv.txt**
 - **dcraw 9.27 en open source sans restriction** (suppression des fonctions restreintes qui étaient sous licence GPLv2 dans le code source) : voir le fichier **LICENSE_lib_dcraw.txt**
 - **easyexif** sous licence **BSD-2-Clause** : voir le fichier **LICENSE_lib_easyexif.txt**

On rappelle le type de licence pour chaque bibliothèque
Identifiant-SPDX

Ne pas hésiter à mettre les liens vers les fichiers de licence,
Le format « markdown » le permet facilement, par exemple :
`[LICENSE_lib_qt.txt](./LICENSES/LICENSE_lib_qt.txt)`

Exemple de fichier « README.md »

- **Auteurs et crédits**

- Auteurs principaux : [Noël Richard](#) et [Bruno Mercier](#)
- Liste exhaustive des auteurs : voir le fichier **AUTHORS.md**
- Liste exhaustive des contributeurs : voir le fichier **CONTRIBUTORS.md**

- **État de développement**

- Dernière version stable (**5.0.1**) compilée avec **MSVC2015** sous **Windows 7 64bits**
- Pas de nouvelle version prévue

- **Informations de contact / support**

- **Pour toute demande** (support, signalement de bug), veuillez contacter : [Bruno Mercier](#)



Bruno MERCIER

11/03/2026



Description détaillée d'un fichier « README.md »

- **Nom et description du logiciel** : Expliquez brièvement ce que fait le logiciel et quel problème il résout
 - *Nom complet et acronyme*
 - *Objectif du logiciel*
 - *Fonctionnalités principales*
 - *Cas d'utilisation généraux*
- **Exigences du système** : Indiquez les exigences minimales du système ainsi que les systèmes non compatibles / non testés
 - *Version du système d'exploitation requise*
 - *Espace de stockage nécessaire*
 - *Mémoire RAM minimale*
 - *Autres exigences matérielles ou logicielles*



Description détaillée d'un fichier « README.md »

- **Instructions d'installation** : Fournissez des étapes claires et concises pour installer le logiciel sur différentes plateformes ou systèmes d'exploitation
 - *Étapes détaillées pour installer le logiciel*
 - *Configuration requise*
 - *Dépendances ou bibliothèques externes nécessaires*
 - *Commandes ou scripts spécifiques pour l'installation*
- **Guide de démarrage rapide** : Fournissez un guide étape par étape pour permettre aux utilisateurs de commencer rapidement à utiliser le logiciel
 - *Étapes clés pour démarrer rapidement avec le logiciel*
 - *Exemples simples d'utilisation*
 - *Principales commandes ou fonctionnalités à connaître*
 - *Lien vers la documentation complète destinée à l'utilisateur final*



Description détaillée d'un fichier « README.md »

- **Fonctionnalités principales** : Décrivez les principales fonctionnalités et capacités du logiciel. Cela aidera les utilisateurs à comprendre ce qu'ils peuvent attendre du logiciel
 - *Liste détaillée des fonctionnalités offertes par le logiciel*
 - *Explications sur chaque fonctionnalité*
 - *Utilisations pratiques de chaque fonctionnalité*
 - *Lien vers la documentation technique / d'API qui spécifie les entrées/sorties de votre logiciel, les formats d'échange de données*
- **Exemples d'utilisation** : Fournissez des exemples concrets d'utilisation afin que les utilisateurs puissent comprendre comment appliquer le logiciel à leurs propres besoins
 - *Scénarios pratiques pour mettre en évidence l'utilisation du logiciel*
 - *Instructions étape par étape pour chaque exemple*
 - *Résultats attendus pour chaque exemple*



Description détaillée d'un fichier « README.md »

- **Licence du logiciel** : Mentionnez clairement la licence du logiciel, qu'il s'agisse d'une licence open source ou propriétaire. Cela permettra aux utilisateurs de connaître les conditions d'utilisation du logiciel
 - *Mention de la licence du logiciel*
 - *Licences des dépendances ou bibliothèques internes (inclues dans le logiciel)*
 - *Conditions d'utilisation*
 - *Restrictions éventuelles*
- **Auteurs et crédits** : Mentionnez les auteurs principaux du logiciel et éventuellement les contributeurs. C'est une bonne pratique pour donner crédit aux personnes qui ont contribué au développement du logiciel
 - *Noms des auteurs principaux*
 - *Contributions spécifiques de chaque auteur*
 - *Mention des contributeurs ayant participé au projet*
 - *Copyright / Crédits (Année de création + Droits patrimoniaux)*



Bruno MERCIER

11/03/2026



Description détaillée d'un fichier « README.md »

- **État de développement** : Si le logiciel est en développement actif, mentionnez-le pour que les utilisateurs sachent à quoi s'attendre en termes de mises à jour et de nouvelles fonctionnalités
 - *Indication claire si le logiciel est en développement actif ou s'il s'agit d'une version stable*
 - *Informations sur les mises à jour ou les nouvelles fonctionnalités prévues (feuille de route)*
- **Informations de contact / support** : Fournissez des informations de contact pour obtenir de l'aide, signaler des problèmes liés au logiciel ou proposer de collaborer sur un logiciel ouvert
 - *Adresse e-mail ou autre moyen de contact pour le support technique*
 - *Liens vers des forums de discussion ou des pages de support en ligne*
 - *Instructions pour signaler les bugs ou demander de l'aide*
 - *Instructions pour pouvoir collaborer au développement du logiciel*
 - *Lien vers la documentation complète destinée au développeur*



Description détaillée d'un fichier « README.md »

- **FAQ (Foire Aux Questions)** : Répondez aux questions fréquemment posées sur le logiciel, en particulier celles liées à l'installation, la configuration et l'utilisation
 - *Questions fréquemment posées sur l'installation, l'utilisation ou la résolution de problèmes*
 - *Réponses claires et concises à chaque question*

Outil de génération de fichier README au format « markdown »

<https://www.makeareadme.com/>



Markdown Input (editable)

Foobar

Foobar is a Python library for dealing with word pluralization.

Installation

Use the package manager [pip](https://pip.pypa.io/en/stable/) to install foobar.

```
```bash
pip install foobar
```
```

Usage

```
```python
import foobar

returns 'words'
foobar.pluralize('word')

returns 'geese'
foobar.pluralize('goose')

returns 'phenomenon'
foobar.singularize('phenomena')
```
```

Contributing

Pull requests are welcome. For major changes, please open an issue first to discuss what you would like to change.

Please make sure to update tests as appropriate.

License

[MIT](https://choosealicense.com/licenses/mit/)

Rendered

Foobar

Foobar is a Python library for dealing with word pluralization.

Installation

Use the package manager [pip](#) to install foobar.

```
pip install foobar
```

Usage

```
import foobar

# returns 'words'
foobar.pluralize('word')

# returns 'geese'
foobar.pluralize('goose')

# returns 'phenomenon'
foobar.singularize('phenomena')
```

Contributing

Pull requests are welcome. For major changes, please open an issue first to discuss what you would like to change.

Please make sure to update tests as appropriate.

License

[MIT](#)



Bruno MERCIER

11/03/2026

Ouvrons la Science de l'université de Poitiers

Il faut une licence sur mon code !



76

Université
de Poitiers

Valorisation LOGICIELS DE RECHERCHE Le saviez-vous ? – DOCUMENTATION

URL https://scienceouverte.univ-poitiers.fr/piliercodeslogiciel/#Le_saviez-vous



Bruno MERCIER

11/03/2026



77

Le saviez-vous ? **DOCUMENTATION**

Nous avons rédigé de la **documentation** sur :

- [Je veux poser une licence sur mon code : comment faire ?](#)
- [L'aspect juridique des licences logicielles](#)
- [Les grandes catégories et compatibilités des licences entre-elles](#)
- [Je veux utiliser un moyen de calcul ou de stockage mutualisé : comment faire ?](#)

Une **question sans réponse** dans cette documentation ?

- Contactez le service « [guichet de la donnée](#) »

Il est possible de **déposer le code source** de son logiciel dans **HAL & [Software Heritage](#)**.

- Comment déposer le code source d'un logiciel dans HAL
<https://doc.archives-ouvertes.fr/deposer/deposer-le-code-source/>
- Un tutoriel en français et en anglais : <https://hal.science/hal-01872189>
- Cela permet son référencement dans le **catalogue des logiciels libres de la recherche académique** : <https://logiciels.catalogue-esr.fr/list>

Pour **aller plus loin**, découvrez :

- Une présentation de Teresa Gomez-Diaz sur les [logiciels de la recherche et leurs licences](#)
- Une présentation de François Pellegrini sur le [droit des logiciels](#)



Bruno MERCIER

11/03/2026



Avec cette documentation, le **chercheur** est capable de :

- Connaître les **grandes catégories de licences logicielles**
- Comprendre les **compatibilités des licences entre elles**
- Savoir **poser la licence appropriée** sur son code
- Savoir sur **quelle forge institutionnelle** déposer son code
- Savoir référencer son logiciel sur **HAL & Software Heritage**

Une question sans réponse dans cette documentation ?

- Contactez le service « [guichet de la donnée](#) »

Le service « [guichet de la donnée](#) » est capable de :

- **Conseiller le chercheur** sur le type de licence le plus approprié (à ses besoins et ses dépendances)
- **Consulter le SPVR** lorsqu'il y a un doute (accord de consortium, etc)
- **Aider au dépôt** du code source / Vérifier l'adéquation de la licence posée
- **Aider au référencement** dans HAL & Software Heritage / Vérifier les métadonnées posées

Valorisation LOGICIELS DE RECHERCHE À lire sans modération – FORMATION

URLS <https://scienceouverte.univ-poitiers.fr/piliercodeslogiciel/#A lire sans modération>
<https://scienceouverte.univ-poitiers.fr/piliercodeslogiciel/#Formation>



Bruno MERCIER

11/03/2026



80

Nous proposons des **contenus en auto-formation** :

- [Passeport pour la Science Ouverte – Codes & Logiciels](#)
- [Les quiz DoRANum](#)
- [Le MOOC FUN – Recherche reproductible : principes méthodologiques pour une science transparente](#)
- [Le MOOC FUN – Reproducible Research II: Practices and tools for managing computations and data](#)
- [Open Source Guides \[FR\]](#)

Cette année 2025-2026, nous proposons une **formation pour les doctorants** :

- **Licences logicielles – Il faut une licence sur mon code !** ([voir le détail de la formation](#))
jeudi 5 février 2026 à la BU du Futuroscope (H7), salle Londres (9h – 12h)

Pour vous inscrire, vous devez passer par la plateforme [ADUM](#).

=> Cette formation est indispensable pour diffuser sereinement ses codes et logiciels de recherche.

Si cette formation intéresse les doctorants, nous projetons d'ouvrir une formation complémentaire pour 2026-2027 :

- **Initiation à GitLab – Je veux versionner mon code !**



Bruno MERCIER

11/03/2026



Valorisation LOGICIELS DE RECHERCHE

DMP OPIDoR – Plan de gestion de logiciels

URL https://scienceouverte.univ-poitiers.fr/piliercodeslogiciel/#Plan_de_gestion_de_logiciels



Bruno MERCIER

11/03/2026



82

Les **plans de gestion de logiciels** ont été introduits en 2017 dans [DMP OPIDoR](#) par la publication du modèle [PRESOFT](#) :

- L'utilisation de ce modèle n'est pas compatible avec l'utilisation d'un modèle de Plan de Gestion de Données.
- Il faut gérer plusieurs plans de gestion pour adresser la problématique complète de certains projets de recherche (un pour les données, un pour les logiciels)
=> soit 2 PGD pour un même projet, ce que peu de personnes font !

Depuis septembre 2025, il est possible d'utiliser le **modèle structuré [Science Europe](#)** pour décrire à la fois ses **jeux de données** et ses **codes sources / logiciels** dans un **même plan de gestion** grâce à l'introduction de PGD « machine-actionable » :

- Il suffit d'ajouter **plusieurs « produits de recherche »** dans un **même plan de gestion** toujours avec l'outil [DMP OPIDoR](#).
- L'utilisation d'un produit de recherche de type « **Logiciel** » ouvrira des champs dédiés à la description d'un logiciel et non d'un jeu de données.

=> Pour apprendre à utiliser DMP OPIDoR, vous pouvez suivre la [formation sur les PGD](#).

En cours...

FORGE GITLAB DE L'UNIVERSITÉ

**Indispensable pour proposer la formation « Initiation à GITLAB »
et permettre la diffusion des codes par tous !**

MERCI DE VOTRE ATTENTION !

Des questions ?

Bruno.Mercier@univ-poitiers.fr



Bruno MERCIER

11/03/2026

