Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Chancellerie fédérale ChF
Transformation numérique et gouvernance de l'informatique TNI

14.09.2026

Em002-3 Guide de licences OSS

Version 2.0-1339ccb

Table des matières

1. Résumé à l’intention de la direction 3
2. Introduction 5
3. Droit d’auteur et licences : principes 5
4. Licences OSS 6
4.1. Principes 6
4.2. Licences open source 6
4.3. Copyleft versus permissif 7
4.3.1. Généralités 7
4.3.2. Licences open source à copyleft fort 7
4.3.3. Licences open source à copyleft faible 8
4.3.4. Licences open source permissives 8
4.4. Compatibilité des licences open source 9
5. Licences pour la simple utilisation 11
6. Licence pour ses propres projets ou la creation et pour les contributions 12
7. Cas d’application et licences appropriées 14
8. Thèmes juridiques particuliers 18
8.1. Aspects internationaux 18
8.2. Exclusion de responsabilité et de garantie 18
8.3. Copyleft et licences entre entités de l’administration fédérale 19
8.4. Licence de brevets 19
8.5. Double licence 19
8.6. Modification ultérieure des licences 20
8.7. Mention des contributeurs (employés et tiers) 20
8.8. Droit d’auteur, licences pour l’IA générative servant à créer du code 20
8.9. Contributor Licence Agreements (CLA) et Developer Certificate of Origin (DCO) 21
8.10. Problèmes avec la (L)GPL-3.0 et l’IoT 23
8.11. Statut juridique de la documentation 23
9. Informations complémentaires sur les questions juridiques 24

10. Annexe 25
10.1. Modifications par rapport à la version précédente 25
10.2. Références 25
10.3. Abréviations 26
10.4. Exemples de publications et de leur licence 26
10.5. D.1 Loom 26
10.6. D.2 trustbroker.swiss 26
10.7. D.3 Geocat.ch 27
10.8. D.4 GWEN / ampycloud / c4dl-multi / dvas /… 27
10.9. D.5 Application EMSG (gestion du patrimoine en milieu urbain) 27
10.10. D.6 Application certificat Covid 28
10.11. D.7 Site web GovCert 28
10.12. D.8 Diverses bibliothèques sur les données ouvertes (p. ex. linked data) 28
10.13. D.9 Apache FOP adapté pour des PDF archivables 29
10.14. D.10 Dépôt électronique conforme au droit avec eKomm 29

Clause de non-responsabilité : Le présent document est un projet en cours d'élaboration et fait partie des outils destinés à soutenir l'administration fédérale dans la publication de code open source. Pour de plus amples informations, voir le README principal.

---

1. Résumé à l'intention de la direction

Il est recommandé de définir stratégiquement au sein de l'administration fédérale, lors du lancement d'un nouveau projet, l'une des deux licences suivantes :

- **Licence AGPL** : effet copyleft souhaité. Le code modifié doit dans tous les cas revenir à la collectivité et peut ensuite être repris par l'administration fédérale. Il est recommandé d'utiliser la version la plus récente de la licence. C'est ainsi que le principe « public money – public code » est le mieux respecté. Les droits de tiers, en particulier en cas de développements ultérieurs, peuvent affaiblir cela jusqu'à la LGPL.
- **Licence MIT** : des tiers doivent pouvoir développer des applications propriétaires sur la base du code de l'administration fédérale. Si une clause de non-endossement est expressément exigée, la BSD-3-Clause est à privilégier.

L'aide à la décision suivante peut être utilisée pour choisir la licence :

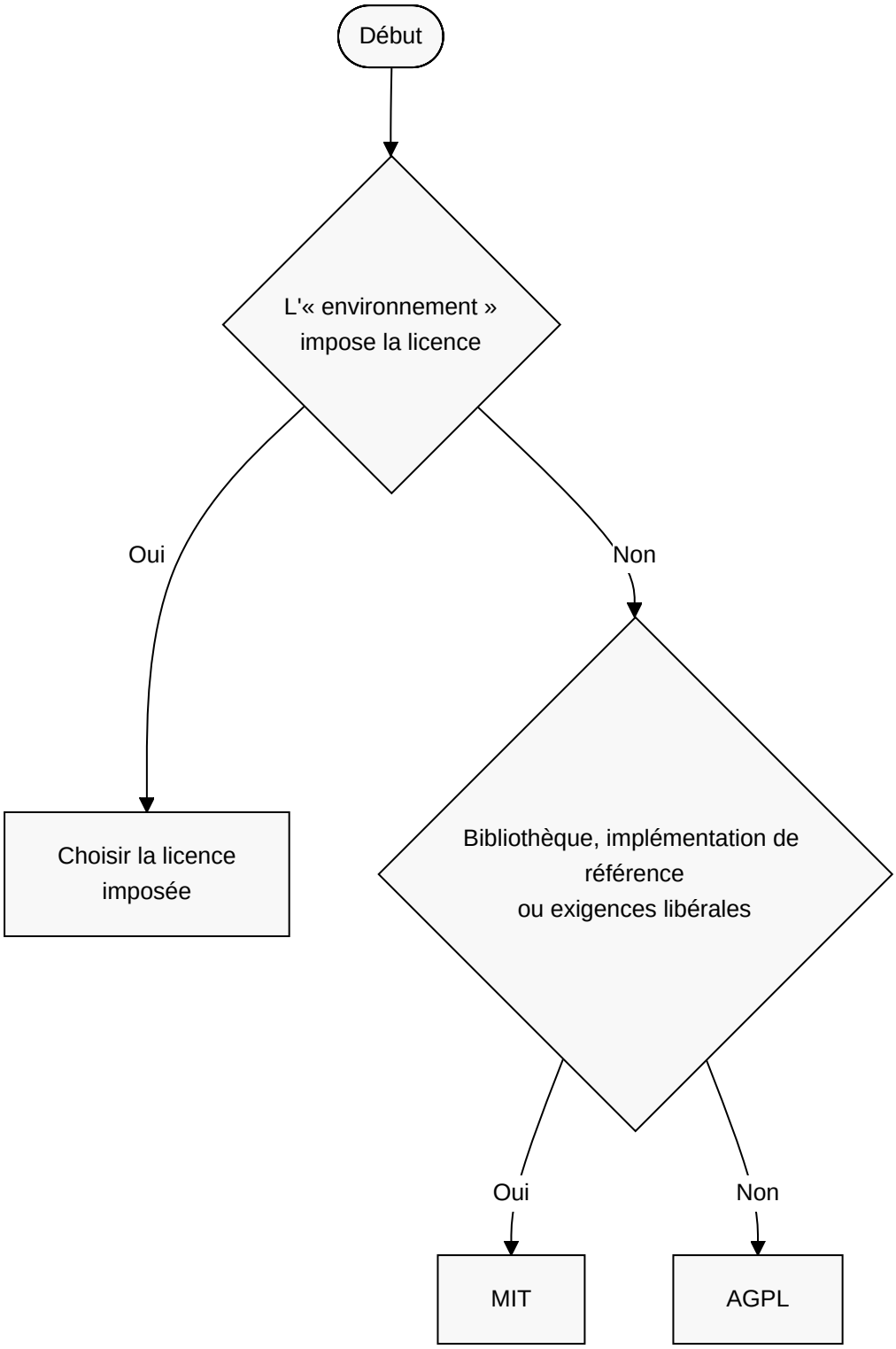


Figure 1 : Aide à la décision simplifiée pour le choix de la licence

Dans une certaine mesure, ces deux licences constituent les deux extrêmes du spectre des licences OSS. Il peut être tout à fait judicieux de choisir une licence mieux adaptée à un projet concret. Le présent guide entend apporter une aide concrète à cet égard.

Si l’analyse des bibliothèques logicielles révèle qu’une autre licence doit être utilisée, les développeurs attirent l’attention de la direction de projet sur ce point. L’analyse est exposée dans Em002-2 Instructions pour la publication de logiciels open source.

Lors de la participation à des projets existants, toutes les licences compatibles OSI sont en principe acceptables.

2. Introduction

L'article 9 de la loi fédérale du 17 mars 2023[1] sur l'utilisation des moyens électroniques pour l'exécution des tâches des autorités (LMETA) prévoit que les autorités fédérales doivent publier et placer sous licence open source les logiciels qu'elles développent ou font développer.

Le présent document poursuit les objectifs suivants :

- Introduction aux thèmes du droit d'auteur sur les logiciels et des licences OSS.
- Une présentation des licences OSS dont l'utilisation ne pose pas de problème au sein de l'administration fédérale. Les licences non mentionnées dans le présent document et les logiciels placés sous de telles licences ne peuvent pas être utilisés sans un examen supplémentaire des dispositions contractuelles par le service juridique compétent.
- Une aide au choix d'une licence pour le développement ou l'acquisition d'un projet au sens de l'art. 9 LMETA.

3. Droit d'auteur et licences : principes

Le droit d'auteur fonde tant des droits moraux que des droits patrimoniaux (droits d'exploitation). Les droits moraux comprennent par exemple le droit à la reconnaissance de la qualité d'auteur ou le droit de décider si et quand l'œuvre est divulguée. Les droits d'exploitation comprennent par exemple le droit de confectionner ou de diffuser des exemplaires de l'œuvre.

La loi fédérale du 9 octobre 1992[2] sur le droit d'auteur et les droits voisins (LDA) prévoit une série d'autres droits moraux et d'exploitation (voir art. 8 ss LDA). Ces droits reviennent directement à l'auteur au moment même de la création de l'œuvre.

L'art. 2 LDA fixe les conditions auxquelles la protection du droit d'auteur est accordée. Une œuvre protégée au sens de la loi n'existe que si ces conditions sont remplies.

Les conditions sont les suivantes :

- L'œuvre doit avoir été créée ; sa simple découverte ne suffit pas. Les données de recherche, par exemple, c'est-à-dire les résultats d'expériences, ne peuvent pas être qualifiées d'œuvres.
- La création doit être « intellectuelle », c'est-à-dire émaner d'un être humain.
- L'œuvre doit avoir été réalisée de manière perceptible. La simple idée sous-jacente à une œuvre n'est pas protégée, seule l'est sa forme extérieure, p. ex. l'impression d'un texte sur papier ou l'exécution d'un morceau de musique.
- Individualité : enfin, l'œuvre doit présenter un certain degré minimal d'individualité. Une œuvre que tout le monde concevrait de la même manière n'est pas protégée.

Les œuvres les plus diverses peuvent être protégées, telles que des textes littéraires ou scientifiques, des œuvres musicales, photographiques, cinématographiques et d'autres œuvres visuelles ou audiovisuelles.

1. RS 172.019
2. RS 231.1

Selon l'art. 2, al. 3, LDA, les programmes d'ordinateur sont également des œuvres si les conditions précitées sont remplies.

Les droits d'auteur reviennent, dès leur naissance, au seul auteur. Lui seul peut interdire à autrui d'accomplir les actes correspondants. L'auteur peut toutefois exercer ses droits autrement que par lui-même :

- Il peut transférer ses droits à des tiers (p. ex. en les vendant). Le nouveau titulaire des droits peut alors les faire valoir de la même manière que s'il était l'auteur.
- L'auteur peut également accorder une licence sur ses droits à des tiers. En termes simples, une licence est un contrat par lequel le titulaire des droits permet à un tiers d'utiliser l'œuvre. L'auteur conserve les droits, mais renonce à les faire valoir à l'égard du preneur de licence.

Lorsqu'un programme d'ordinateur est créé dans le cadre d'un rapport de travail, l'employeur obtient une licence exclusive pour son utilisation (art. 17 LDA). Il ne reste donc pour l'essentiel aux employés que les droits moraux sur le logiciel (ou, selon l'opinion juridique, seulement leur noyau dur). Il s'agit de la même situation juridique que sous la loi sur le personnel de la Confédération.

4. Licences OSS

4.1. Principes

Les OSS se caractérisent pour l'essentiel par les éléments suivants (d'après une définition de l'Open Source Initiative _[Pe1999]_[3]) :

- la redistribution illimitée et gratuite du logiciel est autorisée ;
- le logiciel est disponible sous forme de code source ;
- les modifications du logiciel et leur redistribution sous la même licence sont en principe autorisées ;
- aucune personne ni aucun groupe de personnes ne peut être exclu de l'utilisation du logiciel, et aucun domaine d'application ne peut être exclu (en particulier pas l'utilisation commerciale) ; et
- la diffusion du logiciel avec d'autres logiciels (tels que des logiciels closed source) ne doit pas être restreinte.

Les motifs les plus fréquents du recours aux OSS sont d'abord les économies réalisées grâce à l'utilisation du réservoir déjà considérable de logiciels librement utilisables. Les entreprises qui recourent fréquemment aux OSS parviennent souvent à économiser une part importante de leur budget informatique.

Le code librement disponible peut en outre être facilement adapté à ses propres besoins. De plus, le choix de l'OSS évite le vendor lock-in.[4]

4.2. Licences open source

Les licences open source sont avant tout des contrats de licence ordinaires pour des programmes d'ordinateur.

Contrairement aux contrats de licence ordinaires, les contrats de licence OSS se forment toutefois sans autre dès qu'une seule des formes libres d'utilisation décrites dans la licence est exercée (par exemple par la copie, la redistribution ou la modification du logiciel).

3. https://opensource.org/
4. https://fr.wikipedia.org/wiki/Enfermement_propriétaire

4.3. Copyleft versus permissif

4.3.1. Généralités

Une licence open source au sens de l'OSI peut contenir une clause de « copyleft ». Les modifications apportées à un logiciel placé sous une telle licence doivent être proposées à nouveau aux mêmes conditions que celles applicables au logiciel d'origine. Quiconque apporte de telles modifications doit, lors de la diffusion du logiciel open source sous forme de code objet lisible par machine, proposer également le code source aux destinataires.

Formulé selon la Free Software Foundation, l'effet copyleft protège la liberté des développeurs et garantit ainsi qu'un logiciel une fois libre reste toujours libre.

Avec les licences copyleft, une sorte d'échange s'opère entre la communauté et les entreprises utilisatrices. Les utilisateurs, tenus en contrepartie des avantages tirés du réservoir logiciel existant de replacer leurs propres développements sous la licence concernée, contribuent ainsi au développement de ce réservoir. Au final, les deux parties y gagnent.

La clause de copyleft a un certain « effet de contagion » : si un logiciel sous licence copyleft est intégré à un logiciel jusqu'alors propriétaire, le logiciel initialement propriétaire doit être divulgué sous une licence open source compatible.

Lors de l'utilisation de code source sous licence copyleft, il convient dès lors de veiller à ne l'intégrer que là où le logiciel qui en résulte peut et doit être publié sous une licence open source.

Si une licence open source ne contient pas de clause de copyleft (licence open source permissive), la licence des versions révisées du code peut être choisie librement. Il est en particulier également possible de replacer des modifications sous une licence logicielle propriétaire et d'intégrer le code source correspondant dans un logiciel propriétaire.

Pour l'essentiel, les licences open source peuvent être réparties dans **les catégories suivantes** :

1. licences open source à **copyleft fort** ;
2. licences open source à **copyleft faible** ; et
3. licences open source **permissives** ou **sans copyleft**.

4.3.2. Licences open source à copyleft fort

Lors de l'utilisation d'une licence open source à copyleft fort, les nouvelles versions dérivées du logiciel d'origine doivent, si elles sont transmises à des tiers, être placées sous les conditions de la licence d'origine et mises à la disposition de ces tiers sous forme de code source.

Les deux licences suivantes sont considérées sur le marché comme les licences open source déterminantes à copyleft fort :

- GNU General Public Licence (GPL) ; une grande partie des OSS est aujourd'hui placée sous cette licence
- GNU Affero General Public Licence (AGPL)

La principale différence entre la GPL et l'AGPL porte sur le type d'utilisation déclenchant le copyleft. Avec un logiciel sous GPL, le code source modifié ne doit être proposé que si la nouvelle version du logiciel est offerte à des tiers sous forme de programme exécutable (p. ex. sous forme d'application mobile). Si le logiciel est uniquement mis à disposition via Internet (depuis son propre cloud, sur ses propres serveurs), par exemple sous forme de « Software as a Service » (SaaS) ou d'interface de programmation (API), cela ne déclenche pas le copyleft.

Si le logiciel est placé sous AGPL, le code source modifié doit également être fourni lorsque les fonctionnalités du logiciel sont proposées via un site web ou une interface de programmation. L'AGPL est ainsi encore plus stricte que la GPL.

Avec les licences à copyleft fort, le copyleft ne concerne pas seulement le module logiciel (bibliothèque) en question, mais l'ensemble du programme dans lequel un module sous licence copyleft peut être intégré. L'« effet de contagion » évoqué prend ainsi effet.

4.3.3. Licences open source à copyleft faible

Les licences à copyleft faible exigent également que les modifications de leur code source soient publiées aux tiers destinataires sous la licence open source d'origine.

Contrairement aux licences à copyleft fort, elles n'ont toutefois pas pour effet d'« infecter » d'autres composants logiciels distinguables (d'autres bibliothèques ou le programme principal lui-même) avec leur licence. Cela permet l'intégration de logiciels open source à copyleft faible dans des logiciels propriétaires ou dans des logiciels sous d'autres licences OSS, sans devoir les publier sous la licence d'origine.

Voici les licences open source à copyleft faible les plus fréquemment utilisées :

- GNU Lesser General Public Licence (LGPL)
- Mozilla Public Licence 2.0 (MPL)
- Common Development and Distribution Licence (CDDL)
- Eclipse Public Licence (EPL)[5]
- Microsoft Reciprocal Licence (Ms-RL)
- European Union Public Licence (EUPL)[6]

4.3.4. Licences open source permissives

Les licences sans effet copyleft se caractérisent par le fait qu'elles n'imposent aucune exigence au preneur de licence quant à la licence de son logiciel dérivé, de sorte que ni son nouveau code source ni les modifications du logiciel open source ne doivent être divulgués aux tiers destinataires. Cela permet le développement de produits logiciels propriétaires par l'intégration de logiciels sous licences open source permissives.

Exemples de licences permissives :

- MIT Licence
- Apache Licence 2.0
- Berkeley Software Distribution (BSD) Licence (BSD 2-Clause et BSD 3-Clause)
- Microsoft Public Licence (Ms-PL)

5. Outre l'Eclipse Public Licence (EPL-1.0), il existe aussi l'Eclipse Distribution Licence (EDL-1.0). + Alors que l'EDL correspond à la BSD 3-Clause, l'EPL est une licence à copyleft faible.
6. L'EUPL elle-même comporte un copyleft fort, mais en raison de la clause d'ouverture, un passage à des licences à copyleft faible est possible, de sorte que la protection est limitée.

4.4. Compatibilité des licences open source

Les programmes se composent généralement d'une multitude de composants et de modules logiciels pouvant être reliés entre eux de différentes manières. Dans le développement logiciel, des composants open source existants sont souvent intégrés dans des applications et solutions internes propriétaires. Dès que celles-ci deviennent accessibles de l'extérieur via un site web, des API, une distribution logicielle ou d'une autre manière, il convient de veiller à la compatibilité des licences open source concernées, car le copyleft peut alors prendre effet.

Si des œuvres dérivées placées sous une licence déterminée ne peuvent être diffusées que sous les mêmes conditions de licence, il en résulte que d'autres licences open source assorties d'obligations différentes ou contradictoires pour les œuvres dérivées ne peuvent pas être utilisées.

Certaines licences open source contiennent des clauses dites d'ouverture, qui permettent l'utilisation du code placé sous ces licences dans des projets soumis à d'autres licences. Ainsi, la LGPL version 2.1 permet aussi l'utilisation du code sous GPL. La GPLv3 contient notamment une clause de compatibilité pour l'AGPL et l'Apache License 2.0. L'EUPL est conçue de manière très ouverte et contient en annexe une longue liste de licences compatibles, en particulier celles à copyleft faible telles que la LGPL. Il en résulte que la protection copyleft forte effectivement prévue dans la licence est indirectement affaiblie.

Le schéma suivant représente les dépendances des principales licences open source (Goldstein 2018). La représentation se fonde sur la visualisation de David A. Wheeler, qui a analysé les compatibilités des différentes licences et de leurs numéros de version (Wheeler 2007) et a été développée avec des informations tirées de https://www.gnu.org/licenses/license-list.fr.html.[7]

Le schéma doit être lu comme suit : du code sous une licence figurant plus tôt dans la chaîne peut être intégré dans un logiciel publié sous une licence figurant plus tard dans la chaîne. Ainsi, du code sous BSD modifiée peut être placé sous une variante de licence LGPL ou GPL, mais non l'inverse. Public domain signifie que l'œuvre n'est soumise à aucune restriction de droit d'auteur.

7. Une autre représentation, étendue, est utilisée par GEANT dans son confluence : « Software Licence Selection and Management in GÉANT - GEANT Software Development Support - GÉANT federated confluence », https://wiki.geant.org/pages/viewpage.action?pageId=1032978710. On y trouve quelques licences supplémentaires.

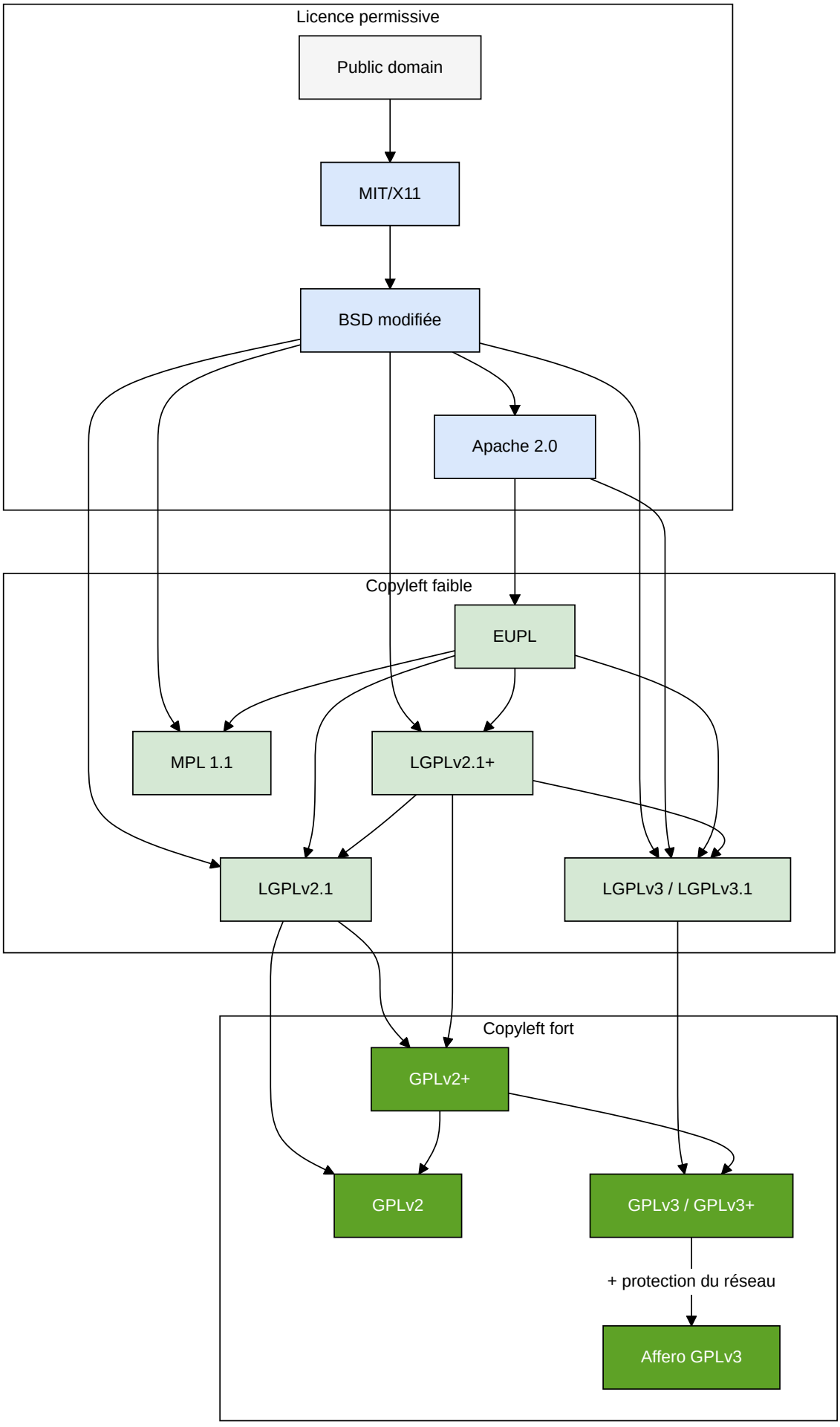

Licence permissive
Public domain
MIT/X11
BSD modifiée
Apache 2.0
Copyleft faible
EUPL
MPL 1.1
LGPLv2.1+
LGPLv2.1
LGPLv3 / LGPLv3.1
Copyleft fort
GPLv2+
GPLv2
GPLv3 / GPLv3+
+ protection du réseau
Affero GPLv3

Figure 2 : Compatibilité des licences open source (d'après le schéma de David A. Wheeler, 2007 et https://www.gnu.org/licenses/license-list.fr.html)

Important : les compatibilités de licences étant toujours pertinentes, les équipes de développement et les prestataires devraient vérifier au préalable les piles de développement et leurs bibliothèques quant aux éventuels problèmes de compatibilité. Autrement dit, avant de valider l'utilisation d'une bibliothèque, il convient de vérifier, outre la sécurité et l'homogénéité de la pile de développement, si la licence de la bibliothèque est compatible avec les licences déterminantes privilégiées par l'administration fédérale.

5. Licences pour la simple utilisation

Ce chapitre traite de la situation dans laquelle un logiciel sous licence OSS doit être utilisé, mais ne doit pas être modifié, ou dans laquelle les modifications ne sont utilisées qu'en interne.

Les licences suivantes ne posent en principe pas de problème lorsqu'un logiciel est utilisé par des autorités fédérales. Les conditions correspondantes, notamment un éventuel copyleft, doivent toutefois être respectées.

Dans certaines circonstances, la licence d'un module logiciel préexistant détermine les licences possibles pour l'ensemble du produit (voir chapitre 4.4).

Certaines licences sont incompatibles entre elles (voir figure 2).

Licence	Copyleft	Particularités
MIT	Non	Licence privilégiée de l'administration fédérale lorsqu'une large diffusion est recherchée et que des tiers doivent pouvoir en développer à nouveau des logiciels propriétaires.
Apache 2.0	Non	Apache est une licence très répandue, connue pour les projets placés sous l'égide de l'Apache Foundation. L'utilisation des marques des donneurs de licence est exclue, sauf pour décrire l'œuvre.
GPL v3	Oui	Il s'agit de LA licence copyleft classique. Le copyleft s'applique aussi lorsque des parties sous GPL sont intégrées dans un programme entier. Interdiction de la gestion des droits numériques. Règles de compatibilité particulières.
LGPLv3	Faible	Licence copyleft réduite, développée spécialement pour les bibliothèques logicielles. Contrairement à la GPL, la LGPL permet, à certaines conditions, de combiner du code fermé (c'est-à-dire propriétaire) avec du code LGPL. Les bibliothèques standard telles que glibc en sont des exemples. La LGPL permet de placer sous GPL le programme dont la bibliothèque est diffusée comme partie intégrante.

Licence	Copyleft	Particularités
Affero GPLv3 (AGPL)	Oui	L'utilisation en SaaS déclenche également le copyleft. Pour le reste, analogue à la GPLv3. Cette licence comble la lacune de la GPL, qui faisait qu'aucune publication n'était requise en cas d'utilisation pour des offres SaaS. Avec cette licence, l'administration fédérale peut garantir que les améliorations du code profitent aussi à la collectivité en cas de SaaS, et elle soutient au mieux le principe **« public money – public code »**.
BSD-3	Non	Licence générale et libérale. Les noms des auteurs ne peuvent pas être utilisés pour promouvoir des œuvres dérivées (clause de publicité)
European Union Public Licence (EUPL)	Faible	Licence particulière utilisée pour l'UE. Particulièrement utile comme base pour des projets européens. Licence tenant compte du droit européen. La publication sous certaines licences compatibles est expressément autorisée.

Tableau 1 : Licences ne posant pas de problème pour les autorités fédérales (selon *[Sc2024]*)

D'autres licences peuvent être utilisées. Dans ces cas, il peut être nécessaire de consulter le service juridique de l'office concerné.

6. Licence pour ses propres projets ou la creation et pour les contributions

Ce chapitre concerne les cas suivants :

- Développement ultérieur / contribution
  - lorsque des autorités fédérales développent un logiciel existant ; ou
  - développent un nouveau logiciel, mais s'appuient sur des bibliothèques logicielles OSS existantes ; et
  - mettent le logiciel correspondant à la disposition de tiers. Les tiers comprennent également les organisations de l'administration fédérale décentralisée, pour autant qu'elles aient leur propre personnalité juridique. Si les développements ultérieurs ne sont utilisés qu'en interne et ne sont pas mis à la disposition de tiers, seul le chapitre Licences pour la simple utilisation s'applique.
- Projets propres / creation / publication : ce chapitre concerne les cas dans lesquels les autorités fédérales développent un nouveau logiciel. Cela devrait être rare, la plupart des développements s'appuyant sur des bibliothèques logicielles open source existantes.

  Pour les développements entièrement nouveaux, il convient de choisir un type de licence permettant une base large et durable pour les développements ultérieurs. Il importe à cet égard que la licence concernée soit largement acceptée dans la communauté de développeurs déterminante.

L’examen de compatibilité est effectué selon Em002-2 Instructions pour la publication d’OSS, chapitre 7.4. La compatibilité des licences est présentée à la figure 2.

Les dimensions suivantes devraient être prises en compte :

- Cas d’application
- Examen souhaité Em002-2.1 Liste de contrôle Évaluation préalable OSS
- Facilité de constitution d’une communauté Em002-4 Guide des communautés OSS
- Facilité d’application
- Sécurité juridique
- Diffusion

Les licences selon le chapitre Licences pour la simple utilisation sont à privilégier.

Les **licences copyleft** suivantes devraient être utilisées en priorité par les autorités fédérales, en particulier si un développement ouvert et durable est recherché et si l’effet copyleft d’héritage doit s’appliquer. Cela garantit que les logiciels payés par la collectivité et tous leurs dérivés restent ouverts. C’est ainsi que le principe « **public money – public code** » est le mieux respecté, et il s’agit de **la recommandation privilégiée pour l’administration fédérale.**

Licences recommandées par ordre décroissant :

- **AGPL v3**
- GPL v3
- LGPL 3.0
- European Union Public Licence (EUPL)

Les licences suivantes devraient être utilisées par les autorités fédérales lorsque **aucun copyleft n’est souhaité** ou requis (par ordre décroissant) :

- **MIT**
- Apache Licence 2.0
- BSD v3

Lors de la participation à des projets existants, les licences suivantes peuvent en outre être utilisées sans problème :

- Mozilla Public Licence
- Microsoft Public Licence

Si une autre licence doit être utilisée, il est recommandé de le justifier brièvement dans la liste de contrôle Em002-2.3 Validation et publication OSS.

Selon l’art. 9, al. 4, LMETA, des licences établies à l’échelle internationale devraient être utilisées dans tous les cas.

Autres outils de vérification des compatibilités de licences :

- ORT Toolkit (https://oss-review-toolkit.org/)
- Black Duck (www.blackducksoftware.com),
- FOSSA (www.fossa.com),
- l'Open Source Licence Comparison Grid (https://www.cmu.edu/cttec/forms/opensourcelicensegridv1.pdf)
- FOSSology (https://www.fossology.org).
- Processus OpenCode : https://opencode.de/en/knowledge/general-conditions/standardised-open-source-licenses

7. Cas d'application et licences appropriées

Le tableau et le schéma suivants montrent comment des licences appropriées peuvent être choisies pour un logiciel open source en fonction de différents cas d'application.

Cas d'application	Licence(s)	Justification
Utiliser un projet existant et y contribuer	**Licence selon les prescriptions du projet**	Nécessaire pour des raisons juridiques.
La licence est imposée par l'utilisation de parties existantes (licences avec copyleft)	**Utiliser une licence compatible avec ces parties.**	Nécessaire pour des raisons juridiques. Remarque : un certain nombre de bibliothèques sont disponibles sous plusieurs licences. Une seule des licences doit correspondre. L'un des buts de la licence multiple est de garantir que les bibliothèques puissent être utilisées tant avec que sans copyleft.
Écosystème existant avec licence privilégiée (p. ex. il s'agit d'un plug-in pour un logiciel publié sous licence MIT. L'utilisation d'une licence MIT serait alors la plus judicieuse, car tous les autres utilisateurs s'y attendent et le logiciel s'intègre dans l'écosystème)	**Utiliser la licence si elle figure sur la liste, sinon utiliser autant que possible la même licence**	L'acceptation dans la communauté est centrale.

Cas d'application	Licence(s)	Justification
L'objectif est la diffusion la plus large possible, p. ex. pour une implémentation de référence[8] (p. ex. pour un logiciel destiné à être utilisé aussi dans des solutions commerciales).	**MIT** BSD v3 Apache Licence 2.0 LGPL v3	Dans ce cas, aucun copyleft ne devrait être utilisé.
Les modifications du code doivent revenir aux autorités fédérales	**AGPL v.3** GNU GPL v.3 EUPL	L'AGPL impose aussi la publication par des tiers en cas d'utilisation dans le cloud, ce que ne font ni la GPL ni l'EUPL.
Les solutions SaaS ne doivent pas être exemptées du retour des modifications de code	**AGPL v.3**	L'AGPL impose aussi la publication par des tiers en cas d'utilisation dans le cloud.
Une collaboration aisée avec la communauté est importante.	**AGPL v.3** GNU GPL v3 Apache Licence 2.0	La GPL est privilégiée pour maintenir le logiciel libre. Apache contient des règles de gouvernance communautaire, des règles supplémentaires pour les contributions (on ne peut pas indiquer une licence autre qu'Apache ; la contribution en tant que telle n'est pas réglée plus en détail, de sorte qu'un CLA existant prime), une licence de brevet et des règles de licence de brevet. En particulier, quiconque fait valoir des brevets contre un autre contributeur perd sa licence de brevet au titre de la licence Apache. Les règles fixées par l'Apache Foundation pour les contributions sont déterminantes. Apache est donc judicieux si la gouvernance du projet doit être construite principalement selon Apache.
Petit composant (bibliothèque) utilisable universellement ou petit logiciel pour lequel la durabilité ou la maintenance par une communauté n'est pas importante	**MIT** Apache Licence 2.0 BSD v3 LGPL v3	La LGPL ne prévoit un copyleft que pour le composant lui-même, mais pas pour l'ensemble du projet dans lequel il est inséré. Les autres licences sont permissives.

8. Par exemple, afin de faciliter la mise en œuvre d'une loi, une implémentation de référence peut être commandée par les autorités fédérales. Voir p. ex. https://ech.ch/de/ech/ech-0238/1.0

Cas d’application	Licence(s)	Justification
Cela doit être simple et permissif ; le contrôle n’est pas important	**MIT** Apache Licence 2.0 BSD v3	Le plus permissif possible
Il importe aux autorités fédérales de savoir ce qu’il advient du code et elles souhaitent recevoir le plus d’améliorations possible.	**AGPL v.3** GNU GPLv3	Copyleft
Il faut empêcher que le logiciel devienne propriétaire et soit positionné sur le marché comme produit concurrent	**AGPL v.3** GNU GPLv3	Copyleft
Il faut empêcher que les autorités fédérales deviennent dépendantes de fournisseurs isolés (« vendor lock-in »)	**AGPL v.3** GNU GPLv3	Le copyleft empêche qu’un fournisseur acquière avec le temps une position exclusive en utilisant le logiciel de la Confédération dans un produit propriétaire, ce qui pourrait à nouveau créer pour la Confédération une dépendance envers ce fournisseur.

Tableau 2 : Licence à utiliser selon le cas d’application pour la publication d’un nouveau logiciel conformément à l’art. 9 LMETA **(gras = recommandé lorsque rien ne s’y oppose)**

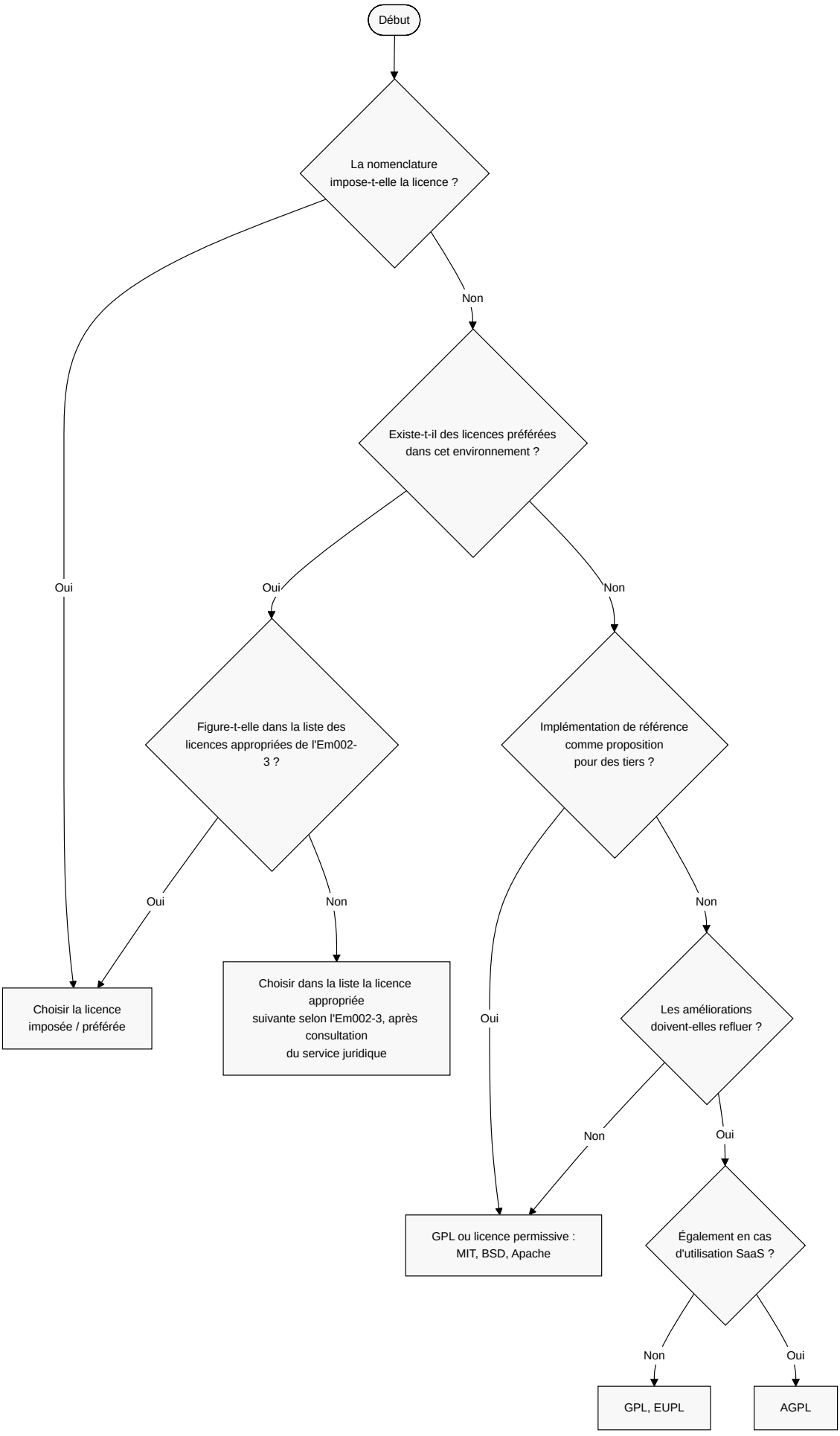

Début
La nomenclature impose-t-elle la licence ?
Oui
Non
Existe-t-il des licences préférées dans cet environnement ?
Oui
Non
Figure-t-elle dans la liste des licences appropriées de l'Em002-3 ?
Implémentation de référence comme proposition pour des tiers ?
Oui
Non
Oui
Non
Choisir la licence imposée / préférée
Choisir dans la liste la licence appropriée suivante selon l'Em002-3, après consultation du service juridique
Les améliorations doivent-elles refluer ?
Oui
Non
Oui
Non
GPL ou licence permissive : MIT, BSD, Apache
Également en cas d'utilisation SaaS ?
Non
Oui
GPL, EUPL
AGPL

Figure 3 : Arbre de décision étendu pour le choix de la licence dans l'administration fédérale (si ni AGPL ni MIT)

La nomenclature de la figure 3 est la software bill of materials (SBOM) de toutes les bibliothèques logicielles et sous-programmes préexistants intégrés dans la solution globale. Chaque bibliothèque a sa propre licence. Selon la compatibilité de celle-ci avec les autres (voir chapitre 4.4), le choix possible de licences pour la publication est restreint.

Un dernier point peut aussi influencer la décision : **le passage d'une licence plus restrictive à une licence moins restrictive est plus facile après coup**. Dans le cas inverse, un fork[9] est presque inévitable.

8. Thèmes juridiques particuliers

8.1. Aspects internationaux

Les questions juridiques essentielles des OSS dans la pratique relèvent principalement du droit des contrats de licence.

S'agissant des questions de droit des licences, le droit choisi par les parties s'applique en premier lieu (la Mozilla Public License MPL contient par exemple, au chiffre 8, une élection de droit en faveur du droit du défendeur).

En l'absence d'élection de droit, ce qui est le cas de toutes les autres licences présentées ici, le droit de l'État du donneur de licence s'applique en règle générale. Ainsi, lorsque les autorités fédérales acquièrent une licence de logiciel auprès de donneurs de licence étrangers, le droit du pays concerné s'applique. Si plusieurs contributeurs sont impliqués, cela peut conduire à des situations complexes. Inversement, le droit suisse s'applique en cas de litige entre des preneurs de licence étrangers et les autorités fédérales en tant que donneuses de licence.

La question de savoir quels aspects relèvent du contrat et lesquels du droit d'auteur ne peut pas recevoir de réponse générale et peut s'avérer très complexe dans les cas d'espèce. La transmissibilité du droit d'auteur ou de parties de celui-ci, ou encore l'étendue des droits pouvant être concédés, relèvent par exemple du droit d'auteur. Le droit du pays de protection concerné doit ici être appliqué. Une élection de droit divergente n'est pas possible (sur l'ensemble, voir Jaeger/Metzger, 449 ss).

8.2. Exclusion de responsabilité et de garantie

Les licences open source excluent régulièrement toute garantie et toute responsabilité dans la mesure permise par le droit. Dans la pratique, les questions de responsabilité ou de garantie se posent rarement ; les risques y afférents sont donc faibles.

On peut partir du principe qu'en cas de licence correcte, les clauses d'exclusion de responsabilité présentes dans toutes les licences examinées ici déploient leurs effets, de sorte que seule subsiste la responsabilité pour négligence grave et dol, le cas échéant pour dommages corporels.

Des règles relatives aux conditions générales demeurent réservées, comme en Allemagne par exemple ; toutefois, dans la mesure où les autorités fédérales sont sujets de responsabilité en tant que donneuses de licence, le droit suisse s'applique (voir 8.1 ci-dessus), lequel admet les exclusions de responsabilité et de garantie précitées dans les relations B2B.

À l'inverse, cela signifie cependant aussi qu'en cas d'erreurs de programme, il est généralement difficile d'agir contre les auteurs du code OSS intégré.

9. Pour la définition, voir Em002-1 Guide pratique pour les logiciels open source dans l'administration fédérale.

La démarche habituelle lors de l'utilisation d'OSS consiste donc régulièrement à se prémunir contre les erreurs par les mesures usuelles de sécurité informatique et à assurer la correction des erreurs au moyen de contrats de maintenance. Il n'existe pas de recours contre les contributeurs OSS.

8.3. Copyleft et licences entre entités de l'administration fédérale

Le copyleft s'applique lors de la transmission entre différentes entités juridiques (par exemple au sein d'un groupe ou entre services de l'administration fédérale décentralisée dotés de leur propre personnalité juridique).

Le copyleft ne s'applique pas lors de la transmission entre services d'une même entité juridique (par exemple au sein de l'administration fédérale centrale).

8.4. Licence de brevets

Les brevets logiciels ne sont possibles que dans des limites étroites selon la conception juridique européenne. L'élément déterminant est que le logiciel apporte une contribution technique, c'est-à-dire résolve un problème technique concret en dehors de l'ordinateur sur lequel il s'exécute. La commande d'un moteur dans une voiture ou la commande d'un robot en seraient des exemples. Un logiciel en tant que tel, qui ne s'exécute que sur un ordinateur mais n'a pas un tel effet externe, n'est en principe pas brevetable en Europe. La plupart des programmes logiciels ne relèvent donc pas du droit des brevets.

Dans d'autres parties du monde, en particulier aux États-Unis, les brevets logiciels sont délivrés un peu plus libéralement. Toutefois, même aux États-Unis, une idée abstraite ou un modèle d'affaires ne devient pas une invention brevetable du seul fait qu'il est mis en œuvre dans un logiciel.

Avec certaines licences, les éventuels brevets du donneur de licence ne sont pas concédés en même temps. Il s'agit de la GPL2, des licences BSD et de la licence MIT.

Pour les projets susceptibles d'être utilisés aussi hors d'Europe, ou qui présentent la forme d'effet externe précitée, il est recommandé de clarifier les aspects de droit des brevets au cas par cas.

8.5. Double licence

L'auteur peut placer un logiciel simultanément sous différentes licences. On parle alors de double licence.

Une forme intéressante de double licence consiste, pour le titulaire des droits, à placer le logiciel sous OSS avec un copyleft fort (ce qui empêche son intégration dans un logiciel propriétaire) et à proposer parallèlement aux preneurs de licence intéressés une licence *payante* leur permettant d'intégrer le logiciel dans leur logiciel propriétaire.

Comme les preneurs de licence intéressés peuvent ainsi éviter de publier leur logiciel propriétaire en OSS, ils sont disposés à payer pour la licence. Voir aussi l'annexe de Em002-1 Guide pratique pour les logiciels open source dans l'administration fédérale.

Pour les autorités fédérales elles-mêmes, la double licence ne devrait présenter d'intérêt que dans de rares cas. Tant que les droits d'auteur appartiennent aux autorités fédérales (ce qui est réglé par exemple par les conditions générales de l'administration fédérale), celles-ci peuvent décider seules d'une éventuelle double licence.

Lorsque la maintenance et le développement (ultérieur) sont externalisés à des tiers, il peut en aller autrement. Si les droits leur appartiennent exceptionnellement et que les autorités fédérales se sont seulement réservé le droit d'octroyer une licence OSS avec copyleft, le tiers pourrait être tenté de procéder à une double licence. Si cela doit être empêché, il convient de l'exclure contractuellement.

8.6. Modification ultérieure des licences

Il convient d'abord de relever que les licences OSS sont en principe immuables et irrévocables. Les preneurs de licence ayant conclu le contrat de licence *avant* une modification peuvent continuer à utiliser, modifier et transmettre le code à des tiers sur la base de l'ancienne licence.

Dans la pratique, un changement de licence conduit souvent à un fork d'une nouvelle version du logiciel sous l'ancienne licence, développé de manière autonome par les anciens preneurs de licence.

L'introduction d'une nouvelle licence pour un logiciel n'est possible que si tous les titulaires des droits d'auteur sur le logiciel y consentent. Des changements sont survenus dans la pratique, par exemple lors du passage d'une licence permissive à une licence copyleft.

8.7. Mention des contributeurs (employés et tiers)

Lorsqu'un programme d'ordinateur est créé dans le cadre d'un rapport de travail, l'employeur obtient une licence exclusive pour son utilisation (art. 17 LDA). Il ne reste donc pour l'essentiel aux employés que les droits moraux sur le logiciel (ou, selon l'opinion juridique, seulement leur noyau dur). Il en va de même pour les tiers qui développent des logiciels pour les autorités fédérales, pour autant que les CG[10] correspondantes de la Confédération s'appliquent, lesquelles transfèrent les droits qui en résultent aux autorités fédérales.

Il se pose néanmoins la question de savoir si les contributeurs (employés, tiers) doivent avoir le droit ou l'obligation de se faire connaître comme auteurs de contributions dans les dépôts open source.

Un argument en faveur du recours aux OSS par les autorités fédérales est l'attractivité de l'emploi : les OSS offrent aux contributeurs la possibilité de montrer publiquement leurs connaissances et de se forger ainsi une réputation. Leurs capacités deviennent vérifiables, sont attestées par des pairs, et celui qui a acquis de l'influence dans un projet OSS peut s'en servir comme levier dans sa recherche d'emploi.

Un autre argument pourrait être les droits moraux, dont le noyau ne peut être retiré à l'auteur d'origine ni par contrat ni par la loi. Ce noyau est toutefois restreint pour les programmes d'ordinateur ; exiger que le développeur d'un programme ait aussi un droit à être nommé irait probablement trop loin.

De plus, un échange entre spécialistes dans la communauté suppose qu'ils puissent être contactés individuellement par d'autres.

Selon le projet, il peut donc être judicieux de donner aux contributeurs la possibilité de se faire connaître comme contributeurs au sein de la communauté.

Du point de vue de la protection des données, toute contrainte en ce sens doit être évitée, étant précisé que l'échange au sein de la communauté ne suppose pas la divulgation des noms — les pseudonymes sont la règle plutôt que l'exception. Si des contributeurs ne souhaitent pas divulguer leur nom, il convient dès lors de leur donner la possibilité d'apparaître sous un pseudonyme.

Il est recommandé de mentionner les noms des développeurs dans les commits, si le projet ne s'y oppose pas.

Dans tous les cas, les autorités fédérales devraient agir en qualité de titulaires des droits.

8.8. Droit d'auteur, licences pour l'IA générative servant à créer du code

Nous ne visons ici que le code créé au moyen d'une IA générative. De plus amples informations sur ce sujet figurent dans Em002-6 FAQ sur l'OSS*, chapitre 2.2. Le code créé par un LLM n'est pas lui-

10. https://www.bkb.admin.ch/bkb/de/home/themen/agb.html

même protégé par le droit d'auteur. Les données d'entraînement utilisées par le LLM peuvent toutefois être protégées par le droit d'auteur. Si le code correspond au 1:1, la licence devrait également être reprise. Si le code a été créé par fragments puis modifié par les développeurs, le risque qu'il soit considéré comme un logiciel dérivé est faible. Il convient de noter que certains LLM exigent que le code soit désigné en conséquence ; les éventuels accords contractuels à ce sujet doivent être respectés.

8.9. Contributor Licence Agreements (CLA) et Developer Certificate of Origin (DCO)

Lorsque plusieurs auteurs travaillent sur le code d'un projet open source, ils obtiennent des droits communs sur le code qui en résulte.

La concession de licence du code à des tiers s'effectue via la licence OSS, un contrat de licence distinct étant conclu entre chaque utilisateur et chaque auteur (faisceau de licences). Cette constellation peut conduire à une situation juridique peu lisible, par exemple dans les relations internationales (voir 8.1 ci-dessus). Lorsque des entreprises décident de placer leur logiciel sous licence OSS et d'accepter des contributions de tiers, elles éprouvent parfois le besoin de conserver un certain contrôle sur le code.

Ainsi, il peut être intéressant de garder ouverte la possibilité de placer le projet sous une nouvelle licence, ou de vouloir soumettre du code sous copyleft à une double licence, ce qui requiert plus de droits que ceux obtenus des contributeurs par une simple licence OSS (voir 8.5 ci-dessus). Les Contributor Licence Agreements (parfois aussi Copyright Transfer Agreements) visent à résoudre de tels problèmes.

Un **Contributor Licence Agreement (CLA)** est un **accord juridiquement contraignant** qui définit les **conditions applicables aux contributions au projet**.

Les contributeurs sont tenus d'approuver cet accord avant de pouvoir apporter du contenu au projet. Pour l'essentiel, les contributeurs accordent au projet et à ses utilisateurs le droit d'**utiliser, de modifier et de redistribuer** les contributions apportées. Il est en outre confirmé que les contributions **ont été rédigées de manière autonome ou légitimées par des tiers**.

Le **Developer Certificate of Origin (DCO)** est un **système de vérification introduit** par la Linux Foundation en 2004, dont le but est proche du Contributor Licence Agreement (CLA) et qui constitue un instrument alternatif approprié. Les contributeurs confirment pour chaque contribution au projet que **(a) ils sont habilités à apporter la contribution, (b) la contribution est couverte par une licence compatible avec le projet et (c) ils consentent à la diffusion et à l'utilisation de la contribution en open source**.

L'instrument principal des CLA est soit la cession des droits d'auteur des contributeurs au développeur principal ou à l'organisation porteuse (dans le cas des Copyright Transfer Agreements), soit l'octroi par les contributeurs au développeur principal de la licence la plus étendue possible, généralement irrévocable (Apache a p. ex. défini un CLA). Les contributions à des projets fermés sont elles aussi souvent soumises à un CLA.

Les CLA sont controversés dans la communauté OSS[11], notamment parce qu'ils peuvent ouvrir des brèches dans le copyleft. C'est pourquoi il existe désormais aussi des CLA qui limitent le développeur principal quant à l'octroi d'une nouvelle licence pour le code du projet.

11. Par exemple : https://ben.balter.com/2018/01/02/why-you-probably-shouldnt-add-a-cla-to-your-open-source-project/

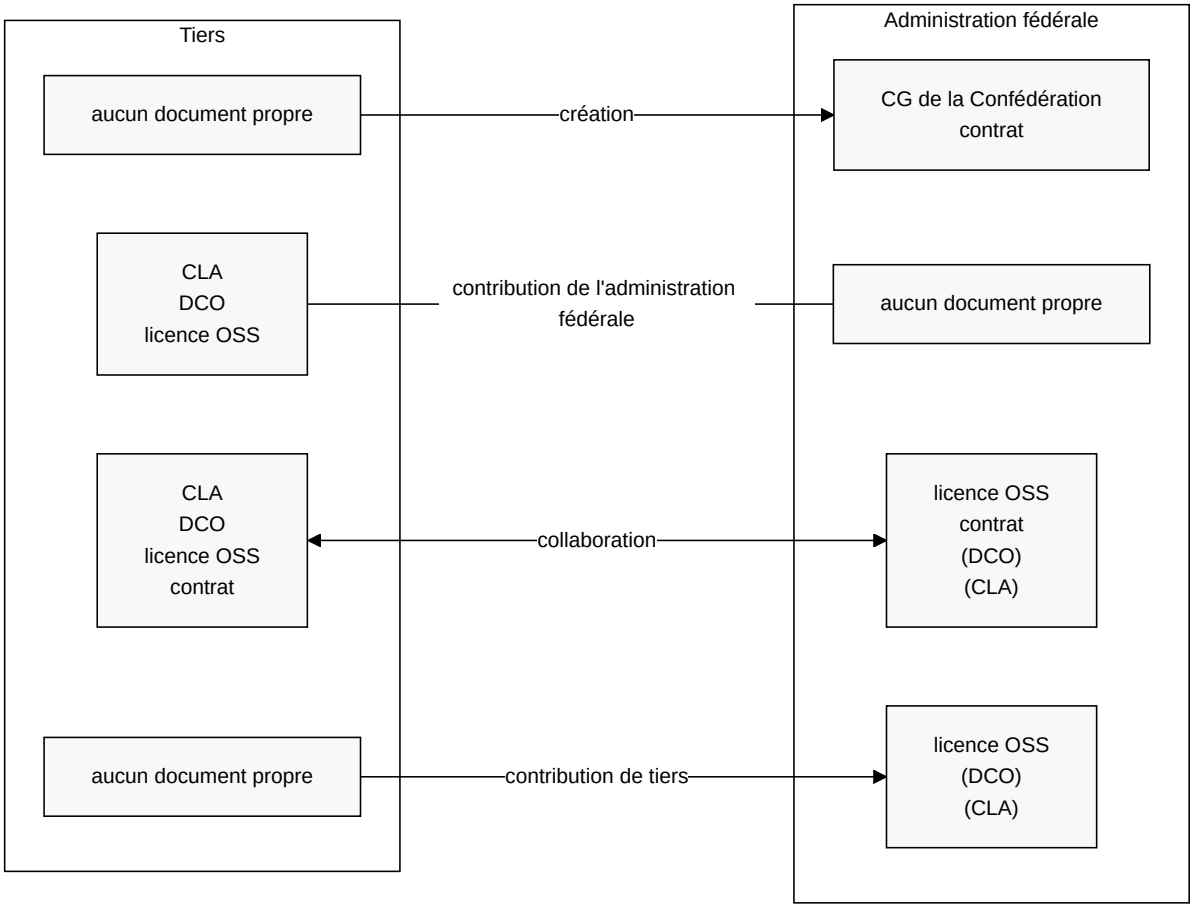


Figure 4 : Transfert du code source et des droits dans différentes constellations

Légende :

- Les cadres indiquent **où les droits sont définis** dans chaque constellation.
- **Création** et **contribution de tiers** : l'administration fédérale dispose du pouvoir de façonner la situation.
- **Collaboration** : les documents sont convenus entre les parties.
- **Contribution de l'administration fédérale** : l'administration fédérale ne dispose d'aucun pouvoir de façonner la situation, un examen est nécessaire.
- Les documents juridiques sont classés par ordre de priorité, de haut en bas ; les documents entre parenthèses sont plutôt à éviter.

Pour les **développements propres** (creation), il n'y a pas de problème : selon l'art. 17 LDA, l'autorité fédérale est titulaire des droits d'utilisation sur les logiciels créés par ses propres employés. Pour les **développeurs tiers**, les autorités fédérales devraient s'assurer les droits (par exemple en utilisant les CG correspondantes de la Confédération). Aucun CLA ni DCO n'est donc nécessaire ici.

Lorsque des autorités fédérales sont développeurs principaux et acceptent des contributions et des collaborations, il convient de renoncer à l'utilisation d'un CLA. Des Digital Certificates of Origin (expliqués avec un exemple dans Em002-4 Guide des communautés OSS, chapitre 3.3) peuvent être utilisés à la place. En principe, les contributions aux projets des autorités fédérales devraient s'appuyer sur la licence OSS afin de ne pas relever davantage la barrière à la contribution.

Si un besoin de CLA devait néanmoins être constaté dans un cas d'espèce après examen approprié, l'autorité fédérale devrait utiliser le CLA d'Apache.[12] Dans ce cas, cette décision et le CLA devraient être mentionnés dans les remarques finales tant de la liste de contrôle Em002-2.1 Évaluation préalable que de la liste de contrôle Em002-4.1 Communauté OSS.

Lorsque des autorités fédérales contribuent à un logiciel de tiers exigeant un CLA, le service juridique doit examiner au cas par cas si l'autorité fédérale peut accepter le CLA existant.[13] Les points suivants doivent être déterminés :

- Une autre autorité fédérale a-t-elle déjà signé le CLA ?
- Qui examine le CLA ?
- Qui le signe ? (Le signataire doit être habilité à signer pour la Confédération à l'égard de tiers)
- Où est-il classé ?
- Comment le CLA ou le DCO est-il documenté dans le projet ?
- Comment s'assure-t-on que le code publié le reste ? (Cela demeure de la responsabilité de l'autorité fédérale au titre de l'art. 9 LMETA, voir [KBB-MB].)

Critères possibles pour l'examen :

- Compatibilité avec l'art. 9 LMETA : le code reste-t-il ouvert ?
- L'autorité fédérale voit-elle des risques pour elle-même ?
- L'autorité fédérale voit-elle un bénéfice pour elle-même (p. ex. le code ne doit pas être maintenu ou développé seule) ou pour d'autres (au sein du fédéralisme suisse ou à l'échelle européenne, un impact plus large peut être obtenu sans charge supplémentaire) ?
- Les droits cédés sont-ils raisonnables ?
- Aucune obligation (inutile) n'est-elle imposée à l'autorité fédérale ?
- Le CLA est-il substantiellement équivalent au CLA d'Apache ?

8.10. Problèmes avec la (L)GPL-3.0 et l'IoT

[BITKOM2023] signale un problème lié à l'utilisation des licences (L)GPL-3.0 en lien avec les logiciels embarqués : de tels appareils devraient permettre l'installation de versions logicielles propres ou nouvelles. Cela peut être indésirable pour des raisons de sécurité.

8.11. Statut juridique de la documentation

Selon l'art. 5 LDA, « les décisions, procès-verbaux et rapports qui émanent des autorités et des administrations publiques » ne sont pas protégés par le droit d'auteur.

Cela va plus loin que CC-0, car le titulaire des droits n'a pas besoin de verser l'œuvre dans le domaine public en renonçant mondialement à tous les droits d'auteur et droits voisins. Elle est déjà dans le domaine public de par la loi. Le message relatif à l'art. 5 LDA précise : « La disposition admet

12. https://www.apache.org/licenses/contributor-agreements.html

13. Pour un destinataire donné, l'administration fédérale n'aurait à examiner la signature d'un CLA qu'une seule fois. Dans d'autres organisations, cela est dès lors traité comme une affaire de haut niveau, les CLA signés étant conservés dans le répertoire des contrats. Les CLA ne doivent pas contourner les obligations de l'autorité fédérale au titre de l'art. 9 LMETA.

toujours la protection par le droit d'auteur pour toute une série d'œuvres nées d'une activité officielle ou en lien avec celle-ci.

Les documents de commissions d'étude et de groupes de travail internes à l'administration, les expertises ou les revues des offices fédéraux, par exemple, ne tombent pas sous le coup de la norme. Il n'existe pas d'intérêt prépondérant à leur libre diffusion, car ils n'influencent pas la situation juridique du citoyen. »

On peut ainsi soutenir que, pour bon nombre des documents pertinents et de la documentation publiée par les autorités fédérales en lien avec les logiciels open source, aucune protection par le droit d'auteur n'existe. Le critère est de savoir si le document affecte ou non le statut juridique du citoyen et échappe ainsi à la protection du droit d'auteur. Si tel n'est pas le cas, une licence appropriée (CC-0[14], CC-BY, CC-BY-SA ou éventuellement LGPL) devrait être utilisée pour la documentation.

Néanmoins, en particulier en cas d'utilisation internationale de la documentation, la situation resterait peu claire pour de nombreux utilisateurs.

Nous estimons donc judicieux que, comme pour le présent document, une publication correspondante soit également indiquée par sécurité.

9. Informations complémentaires sur les questions juridiques

De plus amples informations sur les logiciels open source, sur les caractéristiques de licences spécifiques et sur des aspects juridiques approfondis figurent dans de nombreuses publications, présentées ci-après (voir aussi les références à la fin du document).

- Le texte *« Open Source Software im EMBAG, Analyse des neuen Art. 9 des Bundesgesetzes über den Einsatz elektronischer Mittel zur Erfüllung von Behördenaufgaben »* de Rika Koch et Simon Schlauri, dans les actes de la Conférence sur les acquisitions informatiques 2023 à Berne, offre des orientations supplémentaires sur la mise en œuvre de l'art. 9 LMETA.
- Le *Leitfaden Open-Source-Software 2.0* de la BITKOM allemande traite en détail des questions juridiques liées à l'open source *[BITKOM2023]*. Il se fonde toutefois sur la situation juridique en Allemagne.
- Wolfgang Straub examine, dans son ouvrage *Softwareschutz: Urheberrecht, Patentrecht, Open Source*, les détails juridiques du copyleft en relation avec le droit d'auteur suisse et étudie la compatibilité des licences open source [St2011]. Le chapitre consacré aux logiciels open source (ainsi que des traductions allemandes de diverses licences open source tenant compte de la terminologie juridique suisse) sont librement accessibles sur *www.it-recht.ch.*
- Till Jaeger et Axel Metzger répondent de manière approfondie à de nombreuses questions juridiques liées aux logiciels open source dans leur ouvrage complet *Open Source Software - Rechtliche Rahmenbedingungen der Freien Software [JaAx2016]*. Ils fondent toutefois leur analyse sur la situation juridique en Allemagne.
- Kropp Jonathan/Bauer Alexander, Open Source Compliance und Litigation, CB 2019 p. 285 ss, 285.
- Reymond Michel José, Questions de responsabilité civile et contractuelle soulevées par la distribution de « logiciels libres » (open source), SZW 2022 p. 69 ss.

Divers portails en ligne fournissent en outre des informations détaillées sur les caractéristiques et les questions relatives à des licences open source spécifiques.

14. https://www.creativecommons.ch/

- Sur la plateforme GitHub https://choosealicense.com, les objectifs souhaités pour un projet open source peuvent être sélectionnés et la licence open source appropriée est proposée.
- Sur https://opensource.guide/legal/, GitHub met à disposition un guide en ligne traitant de questions juridiques spécifiques.
- Sur https://www.gnu.org/licenses/license-list.fr.html figurent de brefs commentaires sur diverses licences avec des indications sur leur compatibilité, en particulier avec la GPL.
- Sur https://opensource.org/faq, l'Open Source Initiative traite de nombreuses questions et réponses sur les aspects juridiques des licences open source.
- Sur https://copyleft.org/guide figure un guide détaillé expliquant les détails du copyleft.
- Sur https://tldrlegal.com, les principales licences open source sont résumées selon ce que la licence concernée permet (« can »), ce qu'elle interdit (« cannot ») et ce qu'elle prescrit (« must »).
- Sur https://opensource.com/tags/licensing sont publiés en continu des articles sur des questions de licence actuelles.
- Sur https://www.ifross.org/faq-haeufig-gestellte-fragen, l'« Institut für Rechtsfragen der Freien und Open Source Software » privé de Berlin (ifrOSS) a publié de nombreuses réponses à des questions juridiques fréquentes.

Aujourd'hui, le respect des licences open source est mis en œuvre en premier lieu à l'aide d'outils logiciels (pour de plus amples informations sur la conformité open source, voir aussi Fröhlich-Bleuler *[Fr2012]* et Kuhn, Williamson et Sandler *[KuWiSa2008]*). Divers outils open source de la Linux Foundation sont ainsi documentés et publiés sous le terme « fossology » sur https://www.fossology.org. Des fournisseurs commerciaux tels que Black Duck[15] ou FOSSA[16] proposent également diverses solutions propriétaires permettant de vérifier la compatibilité des licences open source utilisées.

10. Annexe

10.1. Modifications par rapport à la version précédente

- Résumé à l'intention de la direction ajouté.
- Les licences AGPL et MIT sont désormais recommandées comme préférables.
- Nouveau au chapitre 8.8 : droit d'auteur pour le code généré par IA.
- Chapitre 8.9 CLA et CDO fortement étoffé.
- Diverses améliorations rédactionnelles et précisions.
- Davantage d'exemples donnés.

10.2. Références

Voir Em002 Lignes directrices stratégiques pour les logiciels open source dans l'administration fédérale.

15. https://www.blackducksoftware.com
16. https://fossa.com

10.3. Abréviations

Voir Em002 Lignes directrices stratégiques pour les logiciels open source dans l'administration fédérale et Em002-6 FAQ sur l'OSS.

10.4. Exemples de publications et de leur licence

À l'avenir, cette liste sera remplacée par une collecte de métadonnées via publiccode.yml.

10.5. D.1 Loom

URL
https://gitlab.com/swiss-armed-forces/cyber-command/cea/loom

Autorité fédérale
Cyber Command

Description
Loom est un moteur de recherche documentaire puissant et facile à utiliser. Il automatise l'indexation des sources de données configurées, effectue l'OCR, extrait le contenu et les métadonnées, permet le marquage et offre de puissantes fonctions de recherche et d'utilisation.

Année
2025

Licence
MIT

Justification
-

10.6. D.2 trustbroker.swiss

URL
https://github.com/trustbroker-swiss/trustbroker.swiss

Autorité fédérale
ChF, OFIT

Description
Trust Broker Swiss fournit des services de fédération entre relying parties (applications, fournisseurs de services, autres systèmes IAM ou policy enforcement points) et fournisseurs d'identité (IdP, également appelés claims providers), en recourant à des magasins d'attributs de confiance pour enrichir les utilisateurs authentifiés.
Il permet aux fournisseurs d'identité/de claims et aux relying parties d'échanger des informations via un tiers qui masque les spécificités de l'IdP et fournit une identité unifiée ou du moins vérifiée en sus.

Année
2024

Licence
AGPL

Justification
Les modifications du code doivent toujours rester ouvertes et revenir au projet.

Un examen très détaillé de la question de la licence a été réalisé pour le Trust Broker. Pour de très grands projets, il peut être mené à ce niveau de détail. Toutes les licences compatibles avec la software bill of materials ont été recensées dans un pool de licences.

10.7. D.3 Geocat.ch

URL
https://github.com/geonetwork/core-geonetwork

Autorité fédérale
Swisstopo

Description
GeoNetwork est une application de catalogue pour la gestion de ressources géoréférencées. Elle offre de puissantes fonctions d'édition et de recherche de métadonnées ainsi qu'un visualiseur de cartes web interactif. Elle est actuellement utilisée dans de nombreuses initiatives d'infrastructures de données spatiales à travers le monde.

Année
2012

Licence
GPL 2.0

Justification
-

10.8. D.4 GWEN / ampycloud / c4dl-multi / dvas /…

URL
https://meteoswiss.github.io/ampycloud/

Autorité fédérale
MétéoSuisse

Description
Paquet Python permettant de déterminer la proportion de couverture du ciel et la hauteur de base des couches nuageuses à partir de données de célomètre.

Année
2023

Licence
Diverses licences, FSS

Justification
-

10.9. D.5 Application EMSG (gestion du patrimoine en milieu urbain)

URL
https://github.com/astra-emsg/ASTRA.EMSG

Autorité fédérale
MétéoSuisse

Description
EMSG est une application web SIG en C# développée par l'administration fédérale suisse pour gérer la gestion du patrimoine des systèmes routiers urbains.

Année
2017

Licence
BSD

Justification
-

10.10. D.6 Application certificat Covid

URL
https://github.com/admin-ch/CovidCertificate-Documents

Autorité fédérale
OFIT

Description
Application suisse pour les certificats COVID

Année
2020

Licence
MIT

Justification
Aussi permissive que possible afin que chacun puisse utiliser le code.

10.11. D.7 Site web GovCert

URL
https://github.com/govcert-ch/website

Autorité fédérale
NCSC

Description
Code source du site web de l’équipe d’intervention en cas d’urgence informatique (GovCERT) de l’administration fédérale suisse

Année
2023

Licence
MIT

Justification
-

10.12. D.8 Diverses bibliothèques sur les données ouvertes (p. ex. linked data)

URL
https://github.com/SwissFederalArchives

Autorité fédérale
AFS

Description
Le dépôt des Archives fédérales suisses sur GitHub donne accès aux codes sources de nos applications. Il vous permet de créer vos propres forks et de signaler des erreurs ou de proposer des fonctionnalités étendues via l’interface « Issues ».

Année
2023

Licence
AGPL, Apache, MIT

Justification
Les données ouvertes et les logiciels open source vont dans la même direction. L’AGPL permet une ouverture maximale.

10.13. D.9 Apache FOP adapté pour des PDF archivables

URL
https://xmlgraphics.apache.org/fop/

Autorité fédérale
IPI

Description
L'IPI souhaitait une version d'Apache FOP capable de générer des PDF archivables. À cette fin, une extension a été commandée auprès de l'un des développeurs principaux (Jeremias Märki). Les adaptations ont été intégrées directement dans le projet.

Année
2007

Licence
Apache

Justification
Des extensions ont été apportées à un projet open source existant.

10.14. D.10 Dépôt électronique conforme au droit avec eKomm

URL
https://launchpad.net/ekomm

Autorité fédérale
IPI

Description
L'IPI souhaitait un logiciel pour le dépôt électronique.

Année
2009

Licence
Apache

Justification
L'IPI souhaitait que le fournisseur (Glue) procède à la publication et l'a autorisée en mode permissif (intégration dans ses propres applications).