Em002-3 Guide Licences OSS

Recommandations concernant l'informatique de l'administration fédérale[1]

Classification[2]	Non classifié
Effet juridique[3]	Recommandation
Domaine de planification[4]	Informatique de l'administration fédérale
Version	1.0
Version précédente	Nouveau document
État	Ébauche
Date de validation (cette version)	24.02.2025
Validé par, base légale	Validé par le délégué à la transformation numérique et à la gouvernance de l'informatique (délégué TNI), conformément à l'art. 17, al. 1, de l'ordonnance du 25 novembre 2020 sur la transformation numérique et l'informatique (OTNI ; RS *172.010.58*)
Langues	allemand (original), français, italien, anglais (traduction)
Licence	CC0 1.0 Universal By marking the work with a CC0 public domain dedication, the creator is giving up their copyright and allowing reusers to distribute, remix, adapt, and build upon the material in any medium or format, even for commercial purposes.

[1] Gestion des recommandations concernant l'informatique de l'administration fédérale, conformément à [P035], *chap. 4.6*

[2] Pour les échelons de classifications INTERNE et CONFIDENTIEL, voir l'ordonnance du 8 novembre 2023 sur la sécurité de l'information (OSI, *RS 128.1*)

[3] Voir note 1

[4] Domaines de planification selon la *Stratégie informatique de la Confédération 2020-2023 du 3 avril 2020 (SB000)*

Table des matières

1 Introduction 3

2 Droit d'auteur et licence : bases 3

3 Licences OSS 5

3.1 Bases 5
3.2 Licences *open source* 5
3.3 Copyleft 5

3.3.1 Considérations générales 5

3.3.2 Licences open source à copyleft fort 6

3.3.3 Licences *open source* à copyleft faible 6

3.3.4 Licences *open source* permissives 7

3.4 Compatibilité des licences *open source* 7

4 Licence d'utilisation 9

5 Licence pour des projets propres / création et contribution 10

6 Cas d'utilisation et licences appropriées 11

7 Aspects juridiques spéciaux 14

7.1 Aspects internationaux 14
7.2 Exclusion de garantie et de responsabilité 14
7.3 Copyleft et licence entre services de l'administration fédérale 15
7.4 Licences de brevets 15
7.5 Double licence 15
7.6 Modification ultérieure des licences 16
7.7 Mention des contributeurs (collaborateurs et tiers) 16
7.8 Accord de licence de contributeur (CLA) 17
7.9 Problématique (L)GPL-3.0 et IoT 7.4.4 18
7.10 Statut juridique de la documentation 18

8 Informations complémentaires sur les questions juridiques 19

Annexe 21

A. Changements par rapport à la version précédente 21
B. Références 21
C. Abréviations 21
D. Exemples d'OSS mis à disposition et licence utilisée 22

1 Introduction

L'art. 9 de la loi fédérale sur l'utilisation de moyens informatiques pour l'exécution des tâches des autorités (LMETA)[5] dispose que les autorités fédérales publient le code source des logiciels qu'elles développent ou font développer pour l'exécution de leurs tâches.

Le document vise à fournir :

- une introduction aux droits d'auteur en lien avec les licences OSS,
- une présentation des licences OSS que l'administration fédérale peut utiliser sans problème. Les licences qui ne sont pas mentionnées dans ce document et les logiciels soumis à ces licences ne doivent pas être utilisées sans examen complémentaire des dispositions contractuelles par le service juridique compétent,
- une aide au choix de la licence pour le développement ou pour l'achat d'un projet dans le cadre de l'art. 9 LMETA.

2 Droit d'auteur et licence : bases

Le droit d'auteur établit les droits moraux de l'auteur et les droits économiques (droits d'utilisation). Les droits moraux de l'auteur sont notamment le droit à la reconnaissance de la qualité d'auteur et le droit de décider si et quand son œuvre sera divulguée. Les droits d'utilisation sont notamment le droit de confectionner des copies (exemplaires) de l'œuvre et de les mettre en circulation.

La loi du 9 octobre 1992 sur le droit d'auteur (LDA)[6] établit une série d'autres droits (voir art. 8 ss). Ces droits naissent directement chez l'auteur au moment de la création de l'œuvre.

L'art. 2 LDA définit les conditions auxquelles le droit d'auteur est accordé.

Une œuvre est protégée par la loi uniquement si elle remplit les conditions suivantes :

- L'œuvre doit être une création, et pas seulement avoir été trouvée. Des données scientifiques issues de la recherche et d'expériences ne constituent pas une œuvre.
- L'œuvre doit être une création de l'esprit, c'est-à-dire d'origine humaine.
- L'œuvre doit pouvoir être vue ou entendue. L'idée sous-jacente à une œuvre n'est pas protégée, mais uniquement sa forme extérieure, soit le texte imprimé sur papier ou le morceau de musique exécuté.
- Enfin, l'œuvre doit avoir un caractère un tant soit peu individuel. Une œuvre que tout le monde concevrait de la même manière n'est pas protégée.

Les œuvres qui peuvent être protégées sont de natures variées, il peut s'agir de textes littéraires ou scientifiques, d'œuvres musicales, photographiques, cinématographiques et d'autres œuvres visuelles et audiovisuelles.

L'art. 2, al. 3, LDA dispose que les logiciels peuvent également être considérés comme des œuvres s'ils remplissent les conditions requises.

Dès leur naissance, les droits d'auteurs appartiennent uniquement à l'auteur. Lui seul peut interdire que quelqu'un d'autre accomplisse l'acte en question. Il n'est toutefois pas obligé d'exercer lui-même ses droits :

[5] RS 172.019

[6] RS 231.1

- Il peut les transférer à des tiers (p. ex. en les vendant). Le nouveau titulaire peut alors faire valoir les droits comme s'il était l'auteur.
- L'auteur peut également céder ses droits à des tiers sous licence. Pour simplifier, une licence est un contrat par lequel le titulaire des droits autorise des tiers à utiliser une œuvre. L'auteur reste titulaire des droits, mais renonce à les faire valoir auprès du preneur de licence.

L'employeur dispose d'une licence exclusive pour l'utilisation d'un logiciel créé par un travailleur dans l'exercice de son activité (art. 17 LDA). Le travailleur conserve ainsi essentiellement le droit moral sur le logiciel (ou même, en fonction de la doctrine, uniquement une partie centrale de ce droit). Cette situation juridique s'applique au droit du personnel de la Confédération.

3 Licences OSS

3.1 Bases

Les principales caractéristiques d'un OSS sont les suivantes (d'après une définition de l'OSI[7]) :

- le logiciel peut être distribué gratuitement et sans restriction ;
- le logiciel inclut le code source ;
- la licence doit permettre les modifications du logiciel et leur distribution ;
- la licence ne doit pas discriminer une personne ou un groupe de personnes et ne doit empêcher personne d'utiliser le logiciel dans un domaine d'activité spécifique (en particulier l'usage commercial), et
- la distribution du logiciel avec d'autres logiciels (p. ex. logiciel propriétaire) ne doit pas être restreinte.

Le plus souvent, un OSS est utilisé en raison des coûts qui peuvent être économisés en ayant recours à un pool déjà conséquent de logiciels librement utilisables. Les entreprises qui utilisent fréquemment des OSS peuvent ainsi souvent économiser une part considérable de leur budget informatique. De plus, un code librement disponible peut facilement être adapté en fonction des besoins. Choisir les OSS évite en outre l'enfermement propriétaire[8].

3.2 Licences *open source*

Les licences *open source* sont avant tout des contrats de licence pour les programmes informatiques.

Mais contrairement aux contrats de licence normaux, les contrats de licence OSS sont conclus sans difficulté, lorsque seule une forme d'utilisation décrite dans la licence est exercée (p. ex. copier, distribuer ou modifier le logiciel).

3.3 Copyleft

3.3.1 Considérations générales

L'OSI indique qu'une licence *open source* peut contenir une disposition dite copyleft. Les modifications apportées à un logiciel sous licence doivent être proposées aux mêmes conditions que celles auxquelles est soumis le logiciel initial. Si quelqu'un modifie un logiciel et le distribue sous forme de code objet en langage machine, il doit y joindre le code source.

Dans l'esprit de la Free Software Foundation, l'effet copyleft protège la liberté du développeur et garantit qu'une fois libre, un logiciel restera toujours libre.

La licence copyleft induit une sorte d'échange entre une communauté d'une part et les entreprises utilisatrices de l'autre. Dès lors que les utilisateurs sont tenus de soumettre leurs propres développements à la même licence que celle du logiciel du pool dont ils se sont servis, ils contribuent à leur tour au développement du pool. Les deux parties en profitent.

---

[7] https://opensource.org/

[8] https://fr.wikipedia.org/wiki/Enfermement_propriétaire

La disposition copyleft crée une sorte de contagion : si un logiciel soumis à une licence copyleft est intégré dans un logiciel propriétaire, ce dernier doit alors être publié sous une licence *open source* compatible avec le logiciel propriétaire initial.

Par conséquent, il faut faire attention à n'intégrer un code source sous licence copyleft que dans un logiciel qui peut et doit être publié sous une licence *open source*.

En l'absence de disposition copyleft (licence permissive), la licence pour les versions révisées du code source peut être choisie librement. Il est notamment possible de replacer les modifications sous une licence de logiciel propriétaire et d'intégrer le code source dans un tel logiciel.

Les licences *open source* peuvent donc être classées grossièrement dans **trois catégories** :

1. les licences à **copyleft fort** ;
2. les licences à **copyleft faible** ;
3. les licences **permissives** ou **sans copyleft**.

3.3.2 Licences open source à copyleft fort

Avec ce type de licences, les versions dérivées du logiciel initial doivent être transmises à des tiers en respectant les conditions de la licence initiale et le code source doit être mis à disposition.

Sur le marché, les deux licences suivantes sont considérées comme les principales licences *open source* à copyleft fort :

• GNU General Public License (GPL) ; une grande partie des OSS y sont aujourd'hui soumis
• GNU Affero General Public License (AGPL)

La principale différence entre la GPL et l'AGPL concerne le type d'utilisation en fonction du copyleft : dans un logiciel GPL, les modifications du code source doivent être proposées ouvertement uniquement si la nouvelle version du logiciel est proposée à des tiers en tant que programme exécutable (p. ex. application mobile). Si le logiciel est mis à disposition uniquement sur Internet (exécution en nuage sur un serveur propre), par exemple en tant que logiciel sur demande (SaaS) ou interface de programmation (API), le copyleft ne s'applique pas.

Mais si le logiciel est soumis à l'AGPL, les modifications du code source doivent aussi être mises à disposition quand la fonctionnalité du logiciel est proposée sur Internet ou au moyen d'une API. L'AGPL est donc encore plus stricte que la GPL.

Dans le cas de licences à copyleft fort, le copyleft ne concerne pas seulement le module logiciel concerné (bibliothèque), mais l'ensemble du programme logiciel dans lequel un module logiciel sous licence copyleft peut être intégré. C'est le fameux effet de contagion.

3.3.3 Licences *open source* à copyleft faible

Les licences à copyleft faible exigent également que les modifications apportées à leur code source soient mises à disposition des tiers sous la licence *open source* initiale.

Mais contrairement aux licences à copyleft fort, elles n'ont pas pour effet de « contaminer » d'autres parties délimitables du logiciel (d'autres bibliothèques voire le programme principal) avec leur licence. Il est ainsi possible d'intégrer des OSS à copyleft faible dans des logiciels propriétaires ou dans des logiciels qui utilisent d'autres licences *open source* sans avoir à les mettre à disposition sous la licence initiale.

Les principales licences *open source* à copyleft faible sont :

• GNU Lesser General Public License (LGPL)
• Mozilla Public License 2.0 (MPL)
• Common Development and Distribution License (CDDL)
• Eclipse Public License (EPL)
• Microsoft Reciprocal License (Ms-RL)
• European Union Public License (EUPL)[9]

3.3.4 Licences *open source* permissives

Les licences sans effet copyleft permettent à leurs utilisateurs de redistribuer un logiciel dérivé sous une autre licence et donc de ne pas les obliger à divulguer le nouveau code source et les modifications de l'OSS à des tiers. Il est ainsi possible d'intégrer des logiciels soumis à des licences *open source* permissives dans des logiciels propriétaires afin de les développer.

Exemples de licences permissives :

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

3.4 Compatibilité des licences *open source*

Un programme informatique est généralement constitué de nombreux composants logiciels et de modules qui peuvent être reliés entre eux de différentes manières. Lors du développement d'un logiciel, des composants à code source ouvert existants sont souvent intégrés dans les applications et solutions internes. Dès que ces éléments sont accessibles de l'extérieur au moyen d'un site web, d'une API, de la distribution de logiciels ou sous une autre forme, il faut veiller à la compatibilité des licences *open source*, car des dispositions copyleft peuvent s'appliquer.

Certaines œuvres dérivées sont soumises à une licence spécifique et ne peuvent être distribuées qu'aux conditions prévues par cette licence. Dans ce cas, il n'est pas possible d'utiliser d'autres licences *open source* qui prévoient des obligations différentes ou contradictoires pour les œuvres dérivées.

Certaines licences *open source* contiennent des clauses dites d'ouverture, qui autorise l'utilisation du code qui leur est soumis également dans des projets soumis à d'autres licences. Par exemple, la version 2.1 de la LGPL autorise également l'utilisation du code sous la GPL. La GPLv3 contient entre autres une clause de compatibilité avec l'AGPL et l'Apache License 2.0. L'EUPL est conçu de manière très ouverte et contient en annexe une longue liste de licences compatibles, notamment des licences à copyleft faible pour la LGPL, ce qui affaiblit indirectement le caractère fort de sa protection copyleft.

La figure suivante représente les dépendances entre les principales licences *open source*[10].

[9] L'EUPL en soi a un copyleft fort, mais elle contient une clause d'ouverture qui permet de passer à des licences à copyleft faible, de sorte que la protection est limitée.

[10] Goldstein 2018

La représentation se fonde sur la visualisation de David A. Wheeler, qui a analysé la compatibilité des différentes licences selon leur numéro de version[11] et a été complétée par des données provenant du site https://www.gnu.org/licenses/license-list.fr.html.

Le graphique doit être lu comme suit : le code soumis à une licence qui apparaît plus tôt dans la chaîne peut être intégré dans un logiciel publié sous une licence qui apparaît plus tard dans la chaîne. Par exemple, un code modifié soumis à la licence modified BSD peut également être soumis à une variante LGPL ou GPL, mais la réciproque n'est pas vraie. Domaine public signifie que l'œuvre est un bien public.

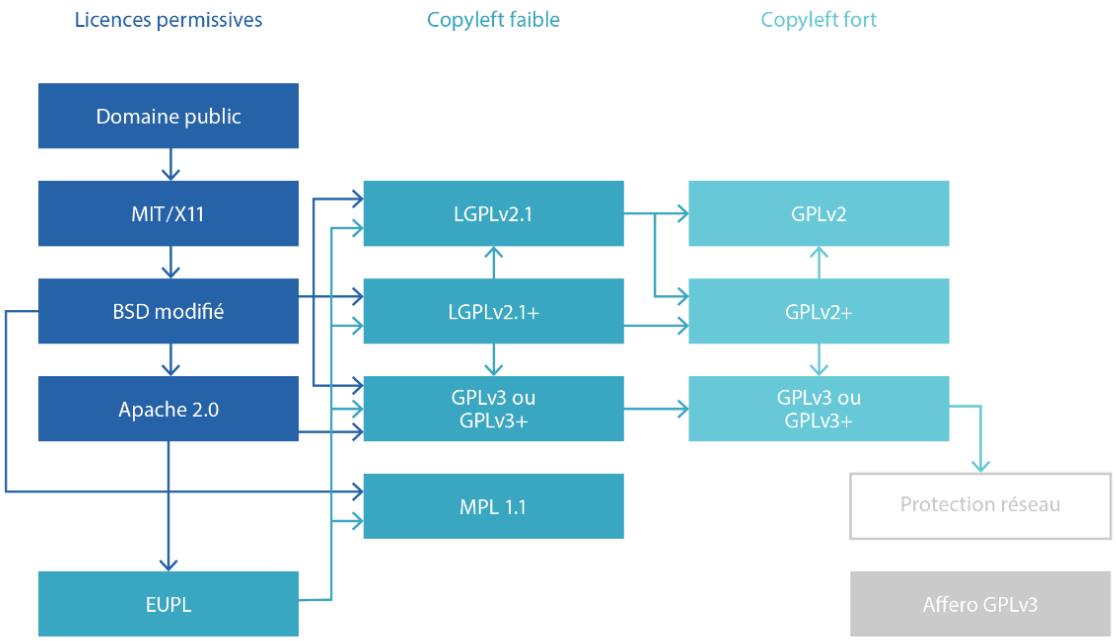

Fig 1 compatibilité des licences *open source* (d'après David A. Wheeler, 2007 et https://www.gnu.org/licenses/license-list.fr.html)

[11] Wheeler 2007

Internal

4 Licence d'utilisation

Le chapitre est consacré à la situation où il est prévu d'utiliser un logiciel soumis à une licence *open source* sans le modifier ou en utilisant les modifications uniquement à l'interne.

Les autorités fédérales peuvent en principe utiliser les licences ci-après sans problème. Il faut toutefois respecter certaines conditions, telles que le copyleft.

Il peut arriver que la licence à laquelle un module logiciel est soumis détermine le choix des licences envisageables pour l'ensemble du produit (voir chap. 3.4 et 0).

Certaines licences ne sont pas compatibles entre elles (voir fig 1).

Licence	**Copyleft**	**Particularités**
MIT	non	-
Apache 2.0	non	Utilisation de la marque du fournisseur de licence exclue, sauf dans la description de l'œuvre.
GPL v3	oui	Le copyleft s'applique aussi lorsque la partie soumise à la GPL est intégrée dans un programme entier. Gestion des droits numériques interdite. Règles de compatibilité particulières.
LGPLv3	faible	Dans certaines circonstances, il est possible de combiner du code fermé (propriétaire) avec le code LGPL, ce qui est interdit avec la GLP. Exemple : bibliothèques standard comme glibc. La LGPL permet de placer sous GPL le programme dont la bibliothèque fait partie.
Affero GPLv3	oui	L'utilisation en tant que SaaS déclenche le copyleft. Sinon, analogue à la GPLv3
BSD-3	non	Il est interdit d'utiliser le nom des auteurs pour promouvoir les œuvres dérivées (clause de publicité)
European Union Publice License (EUPL)	faible	Licence tenant compte du droit européen. Mise à disposition sous certaines licences compatibles explicitement autorisées.

Tableau 1 licences non problématiques pour les autorités fédérales (d'après [Sc2024])

Pour l'utilisation simple, d'autres licences sont envisageables. Il peut alors être nécessaire de consulter le service juridique de l'office concerné.

5 Licence pour des projets propres / création et contribution

Le chapitre concerne les cas suivants :

- Développement / contribution
  - Il s'agit des cas où les autorités fédérales développent un logiciel existant, ou
  - développent un nouveau logiciel en se fondant sur des bibliothèques logicielles OSS existantes, et
  - mettent le logiciel à disposition de tiers. Les organisations de l'administration fédérale décentralisée sont considérées comme des tiers, dans la mesure où elles ont leur propre personnalité juridique. Si les développements ne sont utilisés qu'à l'interne et ne sont pas mis à disposition de tiers, voir chap. 4.
- Projets propres / Création / Publication
  Cette partie concerne les cas où les autorités fédérales développent de nouveaux logiciels. Ces cas sont plutôt rares, car la plupart des développements se fondent sur des bibliothèques OSS existantes.
  Lorsqu'un logiciel entièrement nouveau est développé, il faut choisir un type de licence qui permet de créer une base solide et durable pour des développements ultérieurs. Pour cela, il est important que la licence en question soit bien acceptée par la communauté de développeurs concernée.

La compatibilité est déterminée conformément au chap. 7.4 du document *Em002-2 Instructions pour la publication de logiciels ouverts*. La compatibilité des licences est définie à la fig 1

Il faut tenir compte des points suivants :

- Cas d'utilisation
- Contrôle souhaité (*Em002-2.1 Liste de contrôle Examen préalable)*
- Possibilité de créer facilement une communauté *(Em002-4 Guide Communauté OSS*)
- Simplicité d'utilisation
- Sécurité juridique
- Distribution

La préférence est donnée aux licences décrites au chap. 4.

Les autorités fédérales utiliseront en priorité les **licences copyleft**, en particulier lorsque le but est de poursuivre un développement ouvert durable et que l'effet copyleft doit s'appliquer à l'héritage. Cela garantit que les logiciels payés par les contribuables et toutes leurs versions dérivées restent également ouverts.
Licences recommandées par ordre décroissant :

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

Les licences à utiliser lorsque les autorités fédérales ne souhaitent ou n'exigent **pas de copyleft** (par ordre décroissant) :

- **Apache License 2.0**
- BSD v3
- MIT

En cas de collaboration à des projets existants, les licences ci-après peuvent être utilisées sans problème :

- Mozilla Public License
- Microsoft Public License

S'il est prévu d'utiliser une autre licence, il est recommandé de justifier le choix lors de la mise à disposition dans le document *Em002-2.3 Liste de contrôle Mise à disposition et publication*.

L'art. 9, al. 4 LMETA dispose qu'il faut dans tous les cas utiliser des licences reconnues au niveau international.

D'autres outils pour vérifier les compatibilités de licence sont par exemple :

- Black Duck (www.blackducksoftware.com),
- FOSSA (www.fossa.com),
- Open Source License Comparison Grid (https://www.cmu.edu/cttec/forms/opensourcelicensegridv1.pdf)
- FOSSology (https://www.fossology.org).

6 Cas d'utilisation et licences appropriées

Cas d'utilisation	**Licence(s)**	**Justification**
Utiliser et contribuer à un projet existant	**Licence prévue dans le projet**	Nécessaire pour des raisons juridiques
La licence est imposée par l'utilisation de parties existantes (licences à copyleft)	**Utiliser une licence compatible avec toutes les parties**	Nécessaire pour des raisons juridiques
Écosystème existant avec licence privilégiée (p. ex. module d'extension dans un logiciel mis à disposition sous une licence MIT. L'utilisation de la licence MIT est alors la plus efficace, car elle est attendue par tous les autres utilisateurs et elle permet au logiciel de s'intégrer à l'écosystème.)	**Utiliser une licence de la liste, sinon utiliser si possible la même licence**	Il est important que la licence soit bien acceptée par la communauté.
L'objectif est de distribuer l'OSS le plus largement possible, notamment lors d'une implémentation de référence (p. ex. si une utilisation dans une solution commerciale est visée)[12].	BSD v3 MIT Apache License 2.0 LGPL v3	Ne pas appliquer de copyleft dans ce cas.
Les modifications du code doivent revenir aux autorités fédérales	**AGPL v.3** GPL v.3 EUPL	Même utilisée dans le nuage informatique, l'AGPL impose une mise à disposition par les tiers ; ce qui n'est pas le cas de la GPL et de l'EUPL.

[12] Pour rendre une loi plus facile à appliquer, une implémentation de référence peut être commandée par les autorités fédérales. Voir https://ech.ch/fr/ech/ech-0238/1.0

Les solutions SaaS ne seront pas exclues de ce retour des modifications apportées au code.	**AGPL v.3**	Même utilisée dans le nuage informatique, l'AGPL impose une mise à disposition par les tiers.
Une collaboration simple avec la communauté est importante pour moi.	**GNU GPL v3** Apache License 2.0	Préférer la GPL, afin que le logiciel reste libre. La licence Apache contient des règles de gouvernance pour la communauté, des règles de contribution (interdiction de changer de licence ; contribution en soi non soumise à des règles précises, le CLA est prioritaire), une licence de brevet, des règles concernant la concession des licences de brevet. Elle précise surtout que la concession de brevet sous Apache est annulée si un bénéficiaire attaque un autre contributeur pour violation de brevet. Les règles que l'Apache Foundation se fixe pour les contributions sont essentielles. Il est donc judicieux d'opter pour Apache si l'on souhaite avant tout établir une gouvernance de projet selon ses règles.
Petit module universel (bibliothèque) ou petite partie de logiciel dont la durabilité et l'entretien par une communauté ne sont pas importants.	**Apache License 2.0** **BSD v3** **MIT** LGPL v3	La LGPL n'établit un copyleft que pour le module proprement dit, mais pas pour l'ensemble du projet auquel celui-ci est intégré. Les autres licences sont permissives.
Licence permissive ou contrôle, peu importe pourvu que ce soit simple	**Apache License 2.0** **BSD v3** **MIT**	Permissif au maximum
Les autorités fédérales veulent savoir ce que devient le code et recevoir beaucoup d'amélioration en retour.	**GNU GPLv3**	copyleft
Il s'agit d'éviter que le logiciel ne devienne propriétaire et ne se positionne comme un produit concurrent sur le marché.	**GNU GPLv3**	copyleft
Il s'agit d'éviter que les autorités fédérales ne deviennent dépendantes de certains fournisseurs (enfermement propriétaire).	**GNU GPLv3**	Le copyleft empêche notamment qu'un fournisseur ne finisse par acquérir une position exclusive en utilisant un logiciel de la confédération dans un produit propriétaire, créant ainsi une nouvelle situation de dépendance pour la Confédération.

Tableau 2 licence à utiliser en fonction du cas d'utilisation pour la mise à disposition d'un nouveau logiciel conformément à l'art. 9 LMETA (gras=à utiliser de préférence)

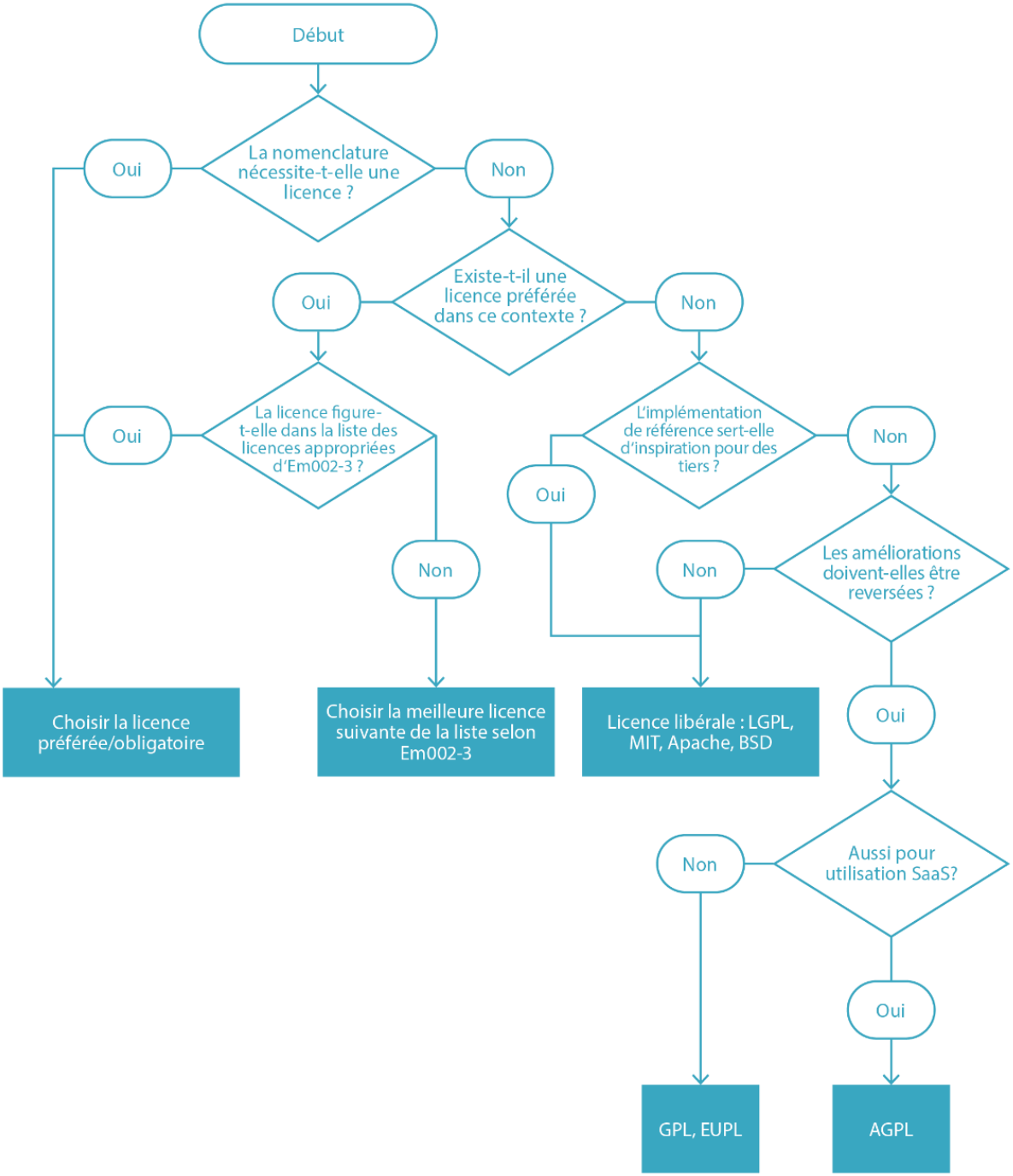

Fig. 2 arbre décisionnel pour le choix de la licence dans l'administration fédérale

Un dernier point peut influencer la décision : il est toujours plus facile de passer à une licence moins restrictive après coup. Dans le cas contraire, un *fork*[13] est presque inévitable.

[13] Voir la définition dans le document *Em002-1 Guide pratique Administration fédérale et logiciels ouverts*

7 Aspects juridiques spéciaux

7.1 Aspects internationaux

Dans la pratique, les questions juridiques liées aux OSS sont principalement de nature contractuelle.

Les questions concernant les droits de licence sont réglées en premier lieu par le droit choisi par les parties (Le chap. 8 de la Mozilla Public License MPL concède à la partie défenderesse le choix du droit applicable).

Le droit applicable dans l'État du donneur de licence s'applique en général lorsque ce droit n'est pas défini autrement, ce qui est le cas pour toutes les autres licences présentées. Par conséquent, le droit du pays où est établi le donneur de licence s'applique lorsque les autorités fédérales soumettent un logiciel à une licence d'un donneur étranger. La situation peut devenir compliquée quand plusieurs contributeurs sont impliqués. À l'inverse, le droit suisse s'applique en cas de litige entre un preneur de licence étranger et les autorités fédérales agissant en tant donneur de licence.

Les aspects concernant le droit contractuel et les droits d'auteur peuvent aussi être compliqués en fonction des cas. Le droit d'auteur traite notamment les questions concernant la cessibilité partielle ou totale du droit d'auteur et les limites dans lesquelles celle-ci est possible. Le droit du pays de protection est déterminant dans ce contexte. Il n'est pas possible d'en choisir un autre[14].

7.2 Exclusion de garantie et de responsabilité

Les licences *open source* excluent en règle générale toute garantie et toute responsabilité, dans les limites autorisées par la loi. Dans la pratique, de telles questions se posent rarement et les risques sont donc faibles.

On peut partir du principe que les clauses de responsabilité prévues dans toutes les licences présentées s'appliquent dès lors que la licence est octroyée correctement. Ainsi, la responsabilité n'est engagée qu'en cas de négligence grave ou de faute intentionnelle, et éventuellement en cas de dommages corporels.

Les CG peuvent être réservées, comme c'est le cas notamment en Allemagne ; toutefois, le droit suisse s'applique dès lors que la responsabilité des autorités fédérales en tant que donneur de licence est engagée (voir chap. 7.1) et celui-ci autorise l'exclusion de la garantie et de la responsabilité dans les relations entre entreprises.

Cela signifie en revanche aussi qu'il est souvent difficile de recourir contre les auteurs en cas d'erreur de programmation dans le code utilisé pour un OSS.

Lorsqu'on utilise des OSS, la procédure habituelle consiste donc en général à prendre les mesures de sécurité informatique usuelles pour se protéger contre les erreurs et à assurer la correction des erreurs en concluant des contrats de maintenance. On ne s'adresse pas aux contributeurs.

[14] Voir Jaeger/Metzger, p. 449 ss

7.3 Copyleft et licence entre services de l'administration fédérale

Le copyleft s'applique lors de la transmission entre différentes personnes morales (p. ex. à l'intérieur d'un groupe ou entre des services de l'administration fédérale décentralisée dotés d'une personnalité juridique propre).

Le copyleft ne s'applique pas en cas de transmission entre les services d'une seule et même personne morale (p. ex. entre des services de l'administration fédérale centrale).

7.4 Licences de brevets

Le droit européen fixe un cadre strict pour les brevets logiciels. Un brevet ne peut être octroyé qu'à condition que le logiciel apporte une contribution technique, c'est-à-dire qu'il résout un problème technique extérieur à l'ordinateur sur lequel il fonctionne. On peut penser à une commande moteur sur une voiture ou à la commande d'un robot. En Europe, un logiciel qui ne fonctionne que sur un ordinateur sans réellement produire d'effet extérieur ne peut pas être breveté. La plupart des logiciels ne relèvent donc pas du droit des brevets.

Ailleurs dans le monde, notamment aux États-Unis, les brevets sont plus facilement accordés pour des logiciels. Cependant, même aux États-Unis, une idée abstraite ou un modèle commercial ne devient pas une invention brevetable par sa simple mise en œuvre dans un logiciel.

Pour certaines licences, les éventuels brevets du donneur de licence ne sont pas inclus dans la licence (p. ex. GPL2, licences BSD et licence MIT).

Pour les projets qui pourraient également être utilisés en dehors de l'Europe ou qui produisent un effet extérieur, comme évoqué plus haut, il est recommandé de clarifier chaque cas du point de vue du droit des brevets.

7.5 Double licence

Un auteur peut proposer son logiciel sous plusieurs licences simultanément. On parle alors de double licence.

Dans la pratique, la double licence est intéressante parce qu'elle permet à l'auteur de proposer son logiciel en tant qu'OSS sous une licence à copyleft fort (évitant ainsi qu'il puisse être intégré dans un logiciel propriétaire), tout en proposant aux preneurs de licence intéressés d'acquérir une licence *payante*, qui les autorise à intégrer le logiciel dans leur logiciel propriétaire.

Les preneurs de licence sont prêts à payer une licence pour éviter d'avoir à mettre à disposition leur logiciel propriétaire en tant qu'OSS. Voir l'annexe du document *Em002-1 Guide pratique Open Source dans l'administration fédérale*.

La double licence n'est pas très intéressante pour les autorités fédérales. Mais tant qu'elles détiennent les droits d'auteur (ce qui est le cas notamment en vertu des CG de l'administration fédérale), elles peuvent décider elles-mêmes si elles souhaitent une double licence.

Si la maintenance et le développement sont confiés à des tiers, la situation peut être différente. Si les droits reviennent exceptionnellement à un tiers et que les autorités fédérales n'ont réservé que le droit d'accorder une licence *open source* avec copyleft, le tiers pourrait être tenté de soumettre le logiciel à une double licence. Pour l'en empêcher, il faut lui interdire contractuellement de le faire.

7.6 Modification ultérieure des licences

La règle veut que les licences OSS ne soient pas modifiables et qu'elles soient irrévocables. Tout preneur de licence qui a conclu un contrat de licence *avant* une modification peut continuer à utiliser le code, à le modifier et à le diffuser auprès de tiers conformément à l'ancienne licence.

Dans la pratique, la modification des licences conduit donc souvent à la création d'une nouvelle version du logiciel soumis à l'ancienne licence, qui est développée par les anciens preneurs de licence sous leur propre responsabilité (*fork*).

Il n'est possible d'introduire une nouvelle licence qu'à condition que tous les auteurs du logiciel soient d'accord. Un tel changement de licence intervient dans la pratique notamment quand on souhaite passer d'une licence permissive à une licence copyleft.

7.7 Mention des contributeurs (collaborateurs et tiers)

L'employeur dispose d'une licence exclusive pour l'utilisation d'un logiciel créé par un travailleur dans l'exercice de son activité (art. 17 LDA). Le travailleur conserve ainsi essentiellement le droit moral sur le logiciel (ou même, en fonction de la doctrine, uniquement une partie de ce droit). Il en va de même pour les tiers qui développent des logiciels pour les autorités fédérales pour autant que les CG de la Confédération[15] s'appliquent et prévoient le transfert des droits correspondants.

Toutefois, la question se pose de savoir s'il faut que les contributeurs (collaborateurs, tiers) aient le droit ou l'obligation d'indiquer dans le dépôt OSS qu'ils sont les auteurs des contributions.

L'un des arguments qui poussent les autorités fédérales à utiliser des OSS est de proposer des emplois attrayants : les OSS donnent à leurs contributeurs l'occasion de démontrer publiquement leurs connaissances et de se faire ainsi une réputation. Leurs capacités sont vérifiables et attestées par des pairs ; ceux qui ont acquis une certaine influence dans un projet OSS peuvent ensuite les mettre en avant lors de la recherche d'emploi.

Un autre argument est le droit moral, qui ne peut être entièrement retiré à l'auteur d'origine ni par contrat ni par la loi. Mais la partie qui revient à l'auteur est très petite dans le cas des logiciels et l'exigence de mentionner le nom du développeur irait sans doute trop loin.

De plus, un échange entre spécialistes dans la communauté présuppose que l'on puisse s'adresser à chacun des développeurs individuellement.

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

Mais du point de vue de la protection des données, il faut renoncer à une obligation, même si les échanges dans une communauté ne requièrent pas qu'on donne son vrai nom et que la règle veut plutôt qu'on utilise des pseudonymes. Les contributeurs qui ne veulent pas donner leur nom devraient donc avoir la possibilité d'utiliser un pseudonyme.

Dans tous les cas, il faut cependant faire en sorte que les autorités fédérales se présentent comme titulaires des droits.

[15] https://www.bkb.admin.ch/bkb/fr/home/themen/agb.html

7.8 Accord de licence de contributeur (CLA)

Lorsque plusieurs auteurs travaillent sur le code d'un projet OSS, ils en détiennent les droits en communs. La licence pour le code est octroyée à des tiers en même temps que la licence OSS, sachant qu'un contrat de licence distinct est conclu entre chaque utilisateur et chaque auteur. Une telle constellation peut brouiller la situation juridique, notamment dans les relations internationales (voir chap. 7.1). En outre, une entreprise qui décide de soumettre son logiciel à une licence OSS et d'accepter les contributions de tiers souhaite souvent garder un certain contrôle sur le code. Il peut, ainsi être intéressant de se réserver la possibilité de soumettre le projet à une nouvelle licence, ou d'ajouter une double licence à un code soumis à une licence à copyleft, ce qui demande davantage de droits que ceux que l'on pourrait obtenir soi-même de ses contributeurs en utilisant une simple licence OSS (voir chap. 7.5).

Un accord de licence de contributeur (CLA ; parfois appelé accord de transfert de droit d'auteur) vise à résoudre ce genre de problème. Le CLA propose deux instruments principaux : la cession des droits d'auteur des contributeurs au développeur principal ou à l'organisation responsable (dans le cas d'un accord de transfert de droit d'auteur), ou l'octroi par les contributeurs au développeur principal d'une licence très large et en général irrévocable. Les CLA sont omniprésents dans le monde OSS. La plupart des grands projets OSS sont régis par les règles d'un CLA.

Les CLA sont parfois controversés au sein de la communauté OSS, car ils peuvent ouvrir des brèches dans le copyleft. Des CLA ont donc été créées pour limiter les possibilités qu'a le développeur principal d'octroyer une nouvelle licence pour le code d'un projet.

Lorsque les autorités fédérales développent elles-mêmes un nouveau logiciel, il est préférable qu'elles concluent un tel CLA avec tous les contributeurs afin de préserver sa flexibilité et de pouvoir poursuivre facilement les éventuelles violations des droits. Les besoins en matière de CLA doivent être clarifiés pour chaque cas, en s'inspirant de modèles, par exemple d'Apache[16].

Lorsque les autorités fédérales contribuent à des logiciels tiers, les services juridiques compétents doivent pour chaque cas vérifier s'il est possible d'accepter le CLA existant.

Il faut alors faire attention aux points suivants :

- L'autorité fédérale est autorisée à exercer les droits d'utilisation sur le logiciel créé par ses employés conformément à l'art. 17 LDA. Si le logiciel est développé par des tiers, elle doit s'assurer les droits (p. ex. en faisant appliquer les CG de la Confédération).
- Pour chaque projet, il convient de déterminer si les interlocuteurs du développeur principal (et les personnes mentionnées dans le dépôt de logiciels) seront les collaborateurs de la Confédération (ou développeurs tiers) concernés ou les autorités fédérales elles-mêmes (voir à ce sujet le chap. 7.7).
- Si certains développeurs doivent répondre au développeur principal, ceux-ci doivent s'annoncer en tant que représentants de la Confédération en tant que titulaires des droits.

Lorsqu'un CLA est appliqué, les documents *Em002-2.1 Liste de contrôle Examen préalable* et *Em002-4.1 Liste de contrôle Communauté OSS* doivent le mentionner.

[16] https://www.apache.org/licenses/contributor-agreements.html

7.9 Problématique (L)GPL-3.0 et IoT 7.4.4

Le document [BITKOM2024] évoque la problématique que soulève l'utilisation de (L)GPL-3.0 en lien avec les objets connectés : ces objets doivent autoriser l'installation de nouvelles versions de logiciels ou de logiciels propres. Mais cela n'est pas toujours souhaitable pour des raisons de sécurité.

7.10 Statut juridique de la documentation

L'art. 5 LDA dispose que les décisions, procès-verbaux et rapports qui émanent des autorités ou des administrations publiques ne sont pas protégés par le droit d'auteur.

Cela va plus loin que ce que demande la CC-0[17], parce que le titulaire des droits ne peut pas « dédier l'œuvre au domaine public en renonçant dans le monde entier à ses droits sur l'œuvre ». L'œuvre relève d'office du domaine public. Le message du 19 juin 1989 concernant la LDA indique, concernant l'art. 5, ce qui suit : « La présente disposition n'empêche pas certaines catégories d'œuvres résultant d'une activité officielle de bénéficier comme par le passé de la protection du droit d'auteur.

Par exemple, les documents émanant de commissions de l'administration et de groupes de travail, les rapports d'experts et les revues publiées par des offices fédéraux ne tombent pas sous le coup de cette réglementation. Il n'existe pas d'intérêt prépondérant plaidant en faveur de leur libre diffusion, car ils n'influencent pas le statut juridique du citoyen. »

On peut donc affirmer qu'un grand nombre de documents et de documentations en lien avec les OSS que publie la Confédération ne sont pas protégés par le droit d'auteur. Le critère déterminant est de savoir si le document influence le statut juridique du citoyen et est donc exclu ou non de la protection par le droit d'auteur. Si ce n'est pas le cas, il faut utiliser une licence appropriée (CC-0, CC-BY, CC-BY-SA ou éventuellement LGPL) pour les documentations.

Malgré tout, la situation ne serait toujours pas claire pour de nombreux utilisateurs, notamment en cas d'utilisation des documentations au niveau international.

C'est pourquoi nous pensons qu'il est judicieux de mentionner la mise à disposition, comme nous le faisons pour ce document.

[17] https://www.creativecommons.ch/

8 Informations complémentaires sur les questions juridiques

Les documents présentés ci-après offrent des informations complémentaires sur les OSS et les caractéristiques concrètes des licences ou permettent d'approfondir les questions juridiques (voir aussi les documents de référence à la fin de ce document).

- Concernant l'application de l'art. 9 LMETA : *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 marchés publics et acquisitions TIC 2023 à Berne.
- Concernant les questions juridiques en lien avec le code source ouvert : *Leitfaden Open-Source-Software 2.0* de l'association allemande BITKOM. Ce document traite toutefois de la situation juridique en Allemagne.
- Concernant les subtilités juridiques du copyleft en lien avec le droit d'auteur suisse et les questions de compatibilité des licences *open source* : *Softwareschutz : Urheberrecht, Patentrecht, Open Source* de Wolfgang Straub. La partie consacrée aux OSS et la traduction allemande de certaines licences *open source* tenant compte de la terminologie juridique suisse sont librement accessibles sur le site *www.it-recht.ch.*
- Concernant de nombreuses questions juridiques en lien avec les OSS, les réponses se trouvent dans l'ouvrage complet *Open Source Software – Rechtliche Rahmenbedingungen der Freien Software* de Till Jaeger et Axel Metzger. Ce document traite toutefois de la situation juridique en Allemagne.
- *Open Source Compliance und Litigation*, CB 2019, p. 285 ss, 285, de Kropp Jonathan/Bauer Alexander.
- *Questions de responsabilité civile et contractuelle soulevées par la distribution de „logiciels libres" (open source)*, SZW 2022 p. 69 ss, de Reymond Michel José.

Il existe également plusieurs portails en ligne qui donnent des informations détaillées sur les caractéristiques de certaines licences *open source* et répondent à de nombreuses questions sur le sujet.

- https://choosealicense.com, une plateforme de GitHub, propose la licence *open source* appropriée en fonction des objectifs d'un projet.
- https://opensource.guide/fr/legal/ de GitHub est un guide en ligne qui répond à des questions juridiques concrètes.
- https://www.gnu.org/licenses/license-list.fr.html propose des commentaires succincts sur différentes licences avec des indications relatives à la compatibilité, en particulier avec la GPL.
- https://opensource.org/faq de l'OSI compile de nombreuses questions et réponses sur les aspects juridiques des licences *open source*.
- https://copyleft.org/guide propose un guide détaillé qui explique le copyleft.
- https://tldrlegal.com compile les principales licences *open source* en fonction de ce qu'elles permettent de faire (can), de ce qu'elles interdisent (cannot) et de ce qu'elles prescrivent (must).
- https://opensource.com/tags/licensing publie en continu des articles sur les questions actuelles en matière de licence.

- https://www.ifross.org/faq-haeufig-gestellte-fragen de l'institut privé *Institut für Rechtsfragen der Freien und Open Source Software* à Berlin (ifrOSS) publie de nombreuses réponses aux non moins nombreuses questions juridiques.

La conformité dans le domaine des licences *open source* est aujourd'hui principalement assurée à l'aide d'outils logiciels (voir aussi Fröhlich-Bleuler et Kuhn, Williamson et Sandler). Il existe à cet effet différents outils *open source* de la Linux Foundation regroupés sous la notion de « fossology » et publiés avec leur documentation sur le site https://www.fossology.org. D'autres fournisseurs commerciaux comme Black Duck[18] ou FOSSA[19] proposent également des solutions propriétaires qui permettent de vérifier la compatibilité des licences *open source* utilisées.

[18] https://www.blackducksoftware.com

[19] https://fossa.com

Annexe

A. Changements par rapport à la version précédente

Nouveau document

B. Références

Voir le document *Em002 Guide stratégique Administration fédérale et logiciels ouverts*

C. Abréviations

Voir les documents *Em002 Guide stratégique Administration fédérale et logiciels ouverts* et *Em002-6 FAQ OSS et art. 9 LMETA.*

D. Exemples d'OSS mis à disposition et licence utilisée

D.1 trustbroker.swiss

URL	https://github.com/trustbroker-swiss/trustbroker.swiss
Autorité fédérale	ChF, OFIT
Description	Trust Broker Swiss provides federation services between relying parties (applications, service providers, other IAM Systems or policy enforcement points) and identity Providers (IdP, aslo called claims providers) using trusted attribute stores to enrich authenticated users. It enables identity/claims providers and relying parties to exchange information via a third party hiding the IdP specifics providing a unified or at least additionally verified identity.
Année	2024
Licence	AGPL
Justification	Les modifications de code doivent dans tous les cas rester ouvertes et être restituées.

D.2 Geocat.ch

URL	https://github.com/geonetwork/core-geonetwork
Autorité fédérale	Swisstopo
Description	GeoNetwork is a catalog application to manage spatially referenced resources. It provides powerful metadata editing and search functions as well as an interactive web map viewer. It is currently used in numerous Spatial Data Infrastructure initiatives across the world.
Année	2012
Licence	GPL 2.0
Justification	-

D.3 GWEN / ampycloud / c4dl-multi / dvas /…

URL	https://meteoswiss.github.io/ampycloud/
Autorité fédérale	MeteoSwiss
Description	Python package to determine the sky coverage fraction and base height of cloud layers using ceilometer data
Année	2023
Licence	Diverses licences, BSD
Justification	-

4 EMSG (Gestion de la maintenance dans les zones bâties)

URL	https://github.com/astra-emsg/ASTRA.EMSG
Autorité fédérale	MeteoSwiss
Description	EMSG is a C# GIS Web Application developed by the Swiss government to handle the asset management of urban road systems.
Année	2017
Licence	BSD
Justification	-

D.5 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	Licence la plus permissive possible pour que tout le monde puisse utiliser le code.

D.6 Site web GovCert

URL	https://github.com/govcert-ch/website
Autorité fédérale	OFCS
Description	Source code of the website for the Computer Emergency Response Team (GovCERT) of the Swiss government
Année	2023
Licence	MIT
Justification	-

D.7 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 au code source de nos applications. Vous pouvez ainsi créer vos propres forks et annoncer des erreurs ou proposer des fonctionnalités élargies sur l'interface « Issues ».
Année	2023

Licence	AGPL, Apache, MIT
Justification	Les données ouvertes et les OSS vont dans le même sens. L'AGPL permet une ouverture maximale.

D.8 Apache FOP modifié pour la création d'archives PDF

URL	https://xmlgraphics.apache.org/fop/
Autorité fédérale	IPI
Description	L'IPI voulait une version d'Apache FOP capable de générer des PDF archivables. Pour ce faire, une extension a été confiée à l'un des développeurs principaux. Les modifications ont été directement intégrées dans le projet.
Année	2007
Licence	Apache
Justification	Des extensions ont été apportées à un projet *open source* existant.

D.9 Saisie électronique juridiquement sûre avec eKomm

URL	https://launchpad.net/ekomm
Autorité fédérale	IPI
Description	L'IPI voulait un logiciel pour la saisie électronique.
Année	2009
Licence	Apache
Justification	L'IPI voulait confier la mise à disposition au fournisseur (Glue), et ce de manière permissive (intégration dans une application propre).