

Em002-1 Guide pratique Administration fédérale et logiciels ouverts

Recommandations concernant l'informatique de l'administration fédérale¹

Annexe indépendante du document principal Em002.

Classification ²	Non classifié
Effet juridique ³	Recommandation
Domaine de planification ⁴	Informatique de l'administration fédérale
Version	2.0
Version précédente	1.1 du 25.02.2025
État	Validé
Date de validation (cette version)	9.12.2025 (Version française du 13.4.2026)
Validé par, base légale	Le délégué à la transformation numérique et à la gouvernance des TIC (D-DTI), conformément à l'article 40 de l'ordonnance du 1er mai 2025 relative aux services numériques et à la transformation numérique dans l'administration fédérale (ordonnance sur la numérisation, DigiV), SR 172.019.1
Langues	allemand (original), français, italien, anglais (traduction)

¹Gestion 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.	Utilisation du guide	3
2.	Définitions	5
3.	Potentiel et défis	8
3.1	Potentiel des OSS	8
3.2	Défis des OSS	10
4.	Utilisation et traitement d'OSS : les cas possibles.....	13
4.1	Maturité des OSS dans l'administration	13
4.2	Utilisation d'OSS non modifiés	15
4.3	Développement avec des composants OSS et mise à disposition du code source	15
4.4	Contribution à un projet ouvert.....	15
5.	Caractéristiques et choix des licences <i>open source</i>.....	15
6.	Acquisition d'OSS	16
7.	Trouver des solutions OSS.....	17
7.1	alternativeTo.....	17
7.2	Dépôts d'OSS	17
7.3	Open Hub.....	18
7.4	Dépôts de code particuliers	19
7.5	OSS Directory.....	19
8.	Modèles de support.....	20
8.1	Utilisation sans support professionnel	20
8.2	Utilisation avec support interne.....	20
8.3	Utilisation avec support externe	21
9.	Interlocuteur.....	21
	Annexes	22
A.	Changements par rapport à la version précédente	22
B.	Références.....	22
C.	Abréviations	22
D.	Modèle d'affaires OSS	23
D.1	Services et produits basés sur des OSS	23
D.2	Services pour les OSS	23
D.3	Souscriptions.....	23
D.4	Double licence.....	24

1. Utilisation du guide

L'art. 9 de la loi fédérale du 17 mars 2023 sur l'utilisation de moyens électroniques pour l'exécution des tâches des autorités (LMETA) oblige les autorités fédérales à publier le code source des logiciels qu'elles développent ou font développer pour l'exécution de leurs tâches.⁵ Le secteur TNI de la ChF a inscrit les principaux objectifs qui découlent de ce mandat légal dans le *Guide stratégique Administration fédérale et logiciels ouverts*.

Le guide pratique vise à concrétiser les prescriptions légales et les directives stratégiques. Il sert aux unités administratives de la Confédération pour l'utilisation, l'acquisition et la mise à disposition de logiciels à code source ouverts, aussi appelés logiciels ouverts (*open source software*, OSS). Il donne une vue d'ensemble de la thématique et renvoie à des informations complémentaires proposées dans d'autres outils d'aide.

Un lecteur qui dispose de connaissances préalables peut se contenter de lire des passages du guide ciblés en fonction de ses besoins et de consulter ensuite les autres outils d'aide. L'aperçu ci-dessous permet de trouver le bon point de départ. En cas de doute, il ne faut pas hésiter à lire le guide en entier.

Intention / Parties prenantes	Documents utiles
Introduction générale aux OSS	<ul style="list-style-type: none"> • Chap. 2, 3 et 7 • <i>Em002-5 Aide-mémoire OSS Appliquer l'art. 9 LMETA</i> • <i>Em002-3 Guide Licences OSS</i> • <i>Em002-6 FAQ-OSS</i>
Introduction générale LMETA pour les décideurs	<ul style="list-style-type: none"> • <i>Em002-5 Aide-mémoire OSS Appliquer l'art. 9 LMETA</i> • <i>Em002 Guide stratégique Administration fédérale et logiciels ouverts</i>
Acquisition de logiciel à utiliser seulement	<ul style="list-style-type: none"> • Chap. 4.1 et 4.2 • <i>Em002-7 Acquisition et OSS - Aspects stratégiques</i> • <i>Notice Acquisition de logiciels et art. 9 LMETA [CCMP-MB]</i> • <i>Directives sur les OSS dans les procédures d'acquisition [OFCL-WL]</i>
Nouveau développement informatique et mise à disposition	<ul style="list-style-type: none"> • Chap. 4.3 et 4.4 • <i>Em002-2 Instructions pour la publication de logiciels ouverts, y c. listes de contrôle</i> • <i>Em002-4 Guide Communauté OSS, y c. listes de contrôle</i>
Acquisition de logiciel développé (service ou œuvre)	<ul style="list-style-type: none"> • Chap. 4.1, 4.3 et 4.4 • <i>Em002-7 Acquisition et OSS - Aspects stratégiques</i> • <i>boîte à outils OFCL avec notamment les directives sur les OSS dans les procédures d'acquisition</i> • <i>Plateforme d'apprentissage et modèles de document pour l'administration publique du CCMP (www.perimap.admin.ch), avec notamment la notice Acquisition de logiciels et art. 9 LMETA [CCMP-MB]</i> • <i>Directives sur les OSS dans les procédures d'acquisition [OFCL-WL]</i> • <i>Em002-2 Instructions pour la publication de logiciels ouverts, y. c. listes de contrôle</i>

⁵ [RS 172.019](#)

	<ul style="list-style-type: none"> • <i>Em002-4 Guide Communauté OSS, y c. listes de contrôle</i>
Aspects juridiques de la mise à disposition	<ul style="list-style-type: none"> • <i>Em002-6 Foire aux questions sur la publication d'OSS (FAQ OSS)</i> • <i>Em002-3 Guide Licences OSS</i>
Bases de travail pour les responsables de projet	<ul style="list-style-type: none"> • <i>Documents HERMES (www.hermes.admin.ch)</i> • <i>Em002-5 Aide-mémoire OSS Appliquer l'art. 9 LMETA</i>
Contributions de logiciels à des tiers et collaborations	<ul style="list-style-type: none"> • <i>Em002-4 Guide Communauté OSS, y c. listes de contrôle</i> • <i>Em002-7 Acquisition et OSS - Aspects stratégiques</i> • <i>Em002-3 Guide Licences OSS</i>

Documents pertinents pour les OSS de l'administration fédérale.

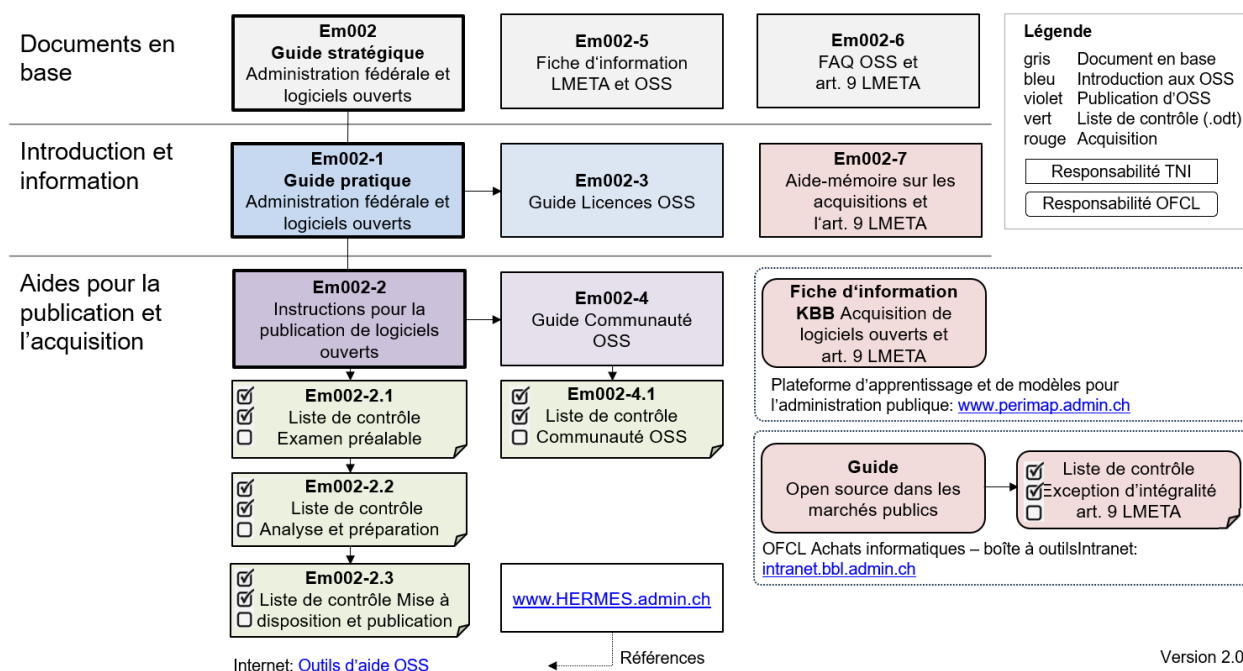


Fig. 1 aperçu des outils d'aide OSS et art. 9 LMETA

2. Définitions

Logiciel à code source ouvert	<p>Un logiciel est dit ouvert ou à code source ouvert (<i>open source software</i>, OSS) s'il est publié sous l'une des quelque 80 licences reconnues par l'Open Source Initiative [OSI2019]. Dans le détail, la définition du code source libre selon [Pe1999] prévoit dix critères que doivent remplir les licences <i>open source</i>. Ces dix critères peuvent être résumés en quatre libertés :</p>
	<ol style="list-style-type: none"> 1. la liberté d'exécuter le programme, pour tous les usages 2. la liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins 3. la liberté de redistribuer des copies du programme 4. la liberté d'améliorer le programme et de distribuer ces améliorations au public (à certaines conditions)
Logiciel libre	<p>Dans les années 1980, avant même l'apparition de la notion de code source ouvert, la Free Software Foundation (FSF) parlait déjà de logiciel libre. Sur le principe, le logiciel libre répond aux mêmes critères que le logiciel ouvert, mais il vise en plus à maintenir le logiciel en libre accès de manière irrévocable et à éviter son intégration dans des logiciels propriétaires.⁶</p>
Logiciel propriétaire	<p>Dans le cas d'un logiciel propriétaire, un fournisseur développe un logiciel et vend une licence d'utilisation à l'utilisateur. L'utilisateur ne connaît en général pas le code et n'a pas le droit de le modifier ni de le diffuser. Il doit se conformer aux prescriptions prévues dans la licence (p. ex. contrat de licence utilisateur final, CLUF) et doit payer des droits de licence : le nombre d'utilisateurs ou de processeurs peut par exemple être limité.⁷ En contrepartie des frais de maintenance, souvent annuels, le fournisseur s'engage à corriger les erreurs dans un délai raisonnable et à poursuivre le développement du logiciel.</p>
Licence	<p>Une licence est un document qui contient des directives contraignantes pour l'utilisation et la distribution d'un logiciel. La licence <i>open source</i> est le plus souvent conforme à l'OSI.⁸</p>
Communauté	<p>Dans le contexte des logiciels ouverts, la définition de la notion de communauté est large. Il peut s'agir d'un écosystème lâche ou d'une structure dotée d'une gouvernance qui fait évoluer le logiciel. Le document <i>Em002-4 Guide Communauté OSS</i> est consacré à ce sujet. Les membres d'une communauté peuvent s'occuper conjointement de la gestion du produit, le développer, le tester, le traduire, donner leur avis ou simplement utiliser le logiciel.</p>
Droits des tiers	<p>Les logiciels ouverts peuvent eux aussi faire l'objet de droits de propriété intellectuelle (auteur, marque, brevet) qui peuvent être revendiqués par des tiers, par exemple en cas d'utilisation de code source créé par des tiers.</p>
Exceptions prévues par la LMETA	<p>Les exceptions prévues à l'art. 9, al. 1, LMETA sont les droits de tiers et des raisons importantes en matière de sécurité. Ces deux éléments sont traités au chap. 3 du document <i>Em002-2 Instructions pour la publication de logiciels ouverts</i>.</p>

⁶ Voir https://fr.wikipedia.org/wiki/Logiciel_libre

⁷ Voir https://fr.wikipedia.org/wiki/Logiciel_propriétaire

⁸ Voir <https://opensource.org/>

Code source	En informatique, le code source est un texte qui présente les instructions composant un programme sous une forme lisible, telles qu'elles ont été écrites dans un langage de programmation. ⁹
Publication	Par publication, on entend essentiellement la publication du texte source sous une licence ouverte conforme à l'OSI. Mais pour des raisons pratiques, la publication porte souvent aussi sur la documentation et les instructions automatiques servant à la construction de l'application prévue dans le code source. Les tests automatiques et les documents de test sont aussi inclus. Pour qu'une communauté puisse se former, on utilise une plateforme qui reprend ensuite plusieurs des fonctions d'une plateforme de développement (build, test, support).
Souveraineté numérique	La définition utilisée est celle donnée dans le rapport du DFAE sur la souveraineté numérique [St2024] qui est en substance la suivante : la souveraineté numérique d'un État ou d'une organisation comprend obligatoirement la capacité à contrôler complètement les données enregistrées et traitées, et à décider en toute indépendance qui a le droit d'y accéder. Elle comprend aussi la capacité à développer de manière autonome des composantes technologiques et des systèmes, à les modifier, à les contrôler et à les compléter.
Dépôt de code source	Un dépôt de code source (<i>repository</i>) est un lieu de stockage qui utilise un outil de contrôle des versions pour les données et les métadonnées sur le code-base. Le dépôt permet à plusieurs contributeurs de travailler sur les mêmes fichiers et d'enregistrer différentes versions. La plupart des dépôts peuvent en outre contenir un suivi des problèmes et des demandes, une gestion des versions, des builds automatisés et de la documentation.
Fork	Le <i>Fork</i> ¹⁰ , aussi appelé reprise logicielle, d'un projet à code source ouvert est le processus par lequel un nouveau logiciel est créé à partir du code source d'un logiciel existant.
Accord de licence de contributeur (CLA)	Un accord de licence de contributeur (en anglais <i>contributor licence agreement</i>) ¹¹ est un document qui établit les droits, les obligations et les règles liés à la propriété intellectuelle dans un projet.
Gestion du code source (SCM)	La gestion du code source ¹² (en anglais <i>source code management</i> ou <i>version control</i>) est un outil qui sert à gérer efficacement les modifications et les numéros de version dans le cadre du développement de logiciel. Le logiciel libre Git ¹³ s'est établi comme outil de SCM distribuée.

⁹ Voir https://fr.wikipedia.org/wiki/Code_source

¹⁰ Voir [https://fr.wikipedia.org/wiki/Fork_\(d%C3%A9veloppement_logiciel\)](https://fr.wikipedia.org/wiki/Fork_(d%C3%A9veloppement_logiciel))

¹¹ Voir https://de.wikipedia.org/wiki/Contributor_License_Agreement (en allemand)

¹² Voir https://fr.wikipedia.org/wiki/Gestion_de_versions

¹³ Voir <https://fr.wikipedia.org/wiki/Git>

Développement de logiciel ouvert (OSSD)

Le développement de logiciel ouvert (*open source software-development, OSSD*)¹⁴ consiste à réaliser publiquement l'ensemble du processus de développement, en plus de la publication du code source. Tout est transparent, depuis les exigences (problèmes et demandes) jusqu'au code source. L'OSSD est soutenu par des outils de communication publics (liste de distribution, forums, etc.), un système de contrôle des versions (git), des listes de bogues et de fonctionnalités, une feuille de route et des outils de développeur.

Software Package Data Exchange (SPDX)

*Software Package Data Exchange (SPDX)*¹⁵ est un format de fichier standard pour la nomenclature logicielle (SBOM) qui vise à faciliter l'utilisation correcte des OSS.

Effet copyleft

Si la licence d'un logiciel contient une disposition copyleft, toute modification ou extension du code source doit être remise à disposition dans la licence de l'OSS en question (voir aussi le document *Em002-3 Guide Licences OSS*).

¹⁴ Voir https://en.wikipedia.org/wiki/Open-source_software_development

¹⁵ Voir https://en.wikipedia.org/wiki/Software_Package_Data_Exchange

3. Potentiel et défis

Dans l'informatique actuelle, l'OSS recèle un grand potentiel. Mais son utilisation représente aussi de grands défis. Ces deux aspects sont présentés ci-après. Les indications reposent notamment sur les résultats des études sur les OSS Suisse de 2018, 2021 et 2024 de l'Université de Berne, lors desquelles des personnes ont été interrogées sur les avantages et les inconvénients qu'elles voyaient en lien avec les OSS.

3.1 Potentiel des OSS

Les possibilités de réaliser le potentiel des OSS lors de leur utilisation et de leur mise à disposition sont présentées ci-après.

- | | |
|--|--|
| <p>1. Souveraineté numérique</p> | <p>Il s'agit de la capacité d'utiliser et de contrôler les prestations numériques. L'autodétermination s'étend à tout le cycle de vie d'un système numérique.</p> <p>L'OSS favorise la souveraineté numérique en garantissant la possibilité de changer de fournisseur ou de technologie, la capacité à concevoir des produits et l'influence en matière de communication des exigences¹⁶.</p> |
| <p>2. Pas de frais de licence</p> | <p>Les licences pour l'utilisation d'OSS ne sont pas payantes. Toutefois, il peut être intéressant de payer des souscriptions (abonnement pour le support) en cas d'utilisation de paquets OSS standard complexes.</p> |
| <p>3. Économies grâce à la coopération avec d'autres utilisateurs</p> | <p>Le principe qui régit les OSS est celui d'un développement unique pour une utilisation multiple. Dès lors, comme un logiciel soumis à une licence ouverte peut être utilisé et développé sans restrictions, les coûts pour en poursuivre le développement peuvent être partagés et il est possible de reprendre des développements opérés par d'autres services de l'administration. On peut également profiter de l'expérience des autres et de ce qu'ils ont développé.</p> |
| <p>4. Communauté et partage de connaissances</p> | <p>Les OSS facilitent la création de communautés et le partage de connaissances, notamment entre les différents niveaux de l'État. Il est possible de continuer à développer un logiciel en commun, de corriger des erreurs et de partager l'expérience individuelle. Le partage accru des connaissances entre les différents services administratifs permet de mieux comprendre ce sur quoi les autres travaillent, ce qui évite les doublons et permet de diffuser les meilleures solutions.</p> |

¹⁶ La définition utilisée est celle donnée dans le rapport [St2024] : la souveraineté numérique d'un État ou d'une organisation comprend obligatoirement la capacité à contrôler complètement les données enregistrées et traitées, et à décider en toute indépendance qui a le droit d'y accéder. Elle comprend aussi la capacité à développer de manière autonome des composantes technologiques et des systèmes, à les modifier, à les contrôler et à les compléter.

- | | |
|--|--|
| 5. Diminution de la dépendance par rapport à un fabricant | L'indépendance par rapport à un fabricant de logiciels (enfermement propriétaire) occupe une place prépondérante en informatique. L'utilisation d'OSS réduit cette dépendance dans la mesure où l'exploitation, la maintenance, le support, le développement et d'autres services peuvent faire l'objet d'appels d'offres publics. Il est possible d'acheter séparément des extensions ou un cycle de vie pour une solution OSS au lieu de devoir acquérir un système entier à cause d'un enfermement propriétaire. Il est dès lors possible de faire l'économie de gros efforts de migration. |
| 6. Standards ouverts et interopérabilité accrue | La compatibilité des OSS avec d'autres logiciels et systèmes informatiques (interopérabilité) est en général plus grande que celle des logiciels propriétaires. De plus, les solutions ouvertes utilisent presque exclusivement des formats de données ouverts, qui sont donc plus faciles à échanger avec d'autres systèmes. |
| 7. Structure de logiciel transparente | Le logiciel est disponible sous forme de code source, de sorte que la qualité peut par exemple être contrôlée dans le cadre de revues externes. Sur la base du code source, il est possible d'établir une documentation (p. ex. dans la perspective d'un nouvel appel d'offres pour poursuivre le développement ou remplacer une OSS à la fin de sa durée de vie). |
| 8. La transparence comme gage de sécurité et de confiance | Un OSS peut être plus sûr qu'un logiciel propriétaire grâce à la publicité de son code source. De plus, il est bien plus compliqué d'introduire des portes dérobées ou des failles dans un OSS, ce qui améliore la confiance dans le logiciel. |
| 9. Meilleure qualité et modularité du code source | La qualité d'un OSS peut être meilleure que celle d'un logiciel propriétaire, car un développeur a tendance à être plus motivé à produire un bon code lorsqu'il sait que celui-ci sera publié. De plus, les solutions ouvertes sont généralement très modulaires, de sorte que certains modules peuvent facilement être remplacés tout en continuant à utiliser les autres. ¹⁷ |
| 10. Facilité d'adaptation | L'accès au code source permet aux utilisateurs de poursuivre eux-mêmes le développement du logiciel ou de confier ce développement à des tiers. Ils peuvent ainsi rapidement adapter le logiciel à leurs besoins. |
| 11. Rapidité d'innovation et d'intégration | Une communauté développe un OSS rapidement et en permanence. Les nouvelles technologies et les nouveaux standards de données sont par exemple souvent publiés sous forme de bibliothèque logicielle à code source ouvert. Il est ainsi possible de réaliser rapidement des solutions informatiques innovantes. |

¹⁷ Par rapport aux logiciels propriétaires, un autre aspect de la sécurité réside dans le fait qu'il est plus facile d'établir la documentation relative aux bibliothèques utilisées (y c. dépendances). Inversement, cela permet d'identifier et d'éliminer plus rapidement les faiblesses (voir aussi SBOM).

12. Attractivité de l'employeur

L'utilisation de technologies à code source ouvert modernes et la collaboration informelle avec des communautés internationales renforcent la motivation des employés et améliorent ainsi l'attractivité de l'employeur, qui peut alors plus facilement recruter des spécialistes jeunes et qualifiés.

3.2 Défis des OSS

Les défis habituellement rencontrés lors de l'utilisation d'OSS sont présentés ci-après avec des ébauches de solution. Les défis généraux concernant tous les types de logiciel ne sont pas abordés (p. ex. nécessité d'examiner la cybersécurité et d'assurer le support pour les logiciels critiques).

1. Coûts de changement élevés et dépendances

Passer d'un logiciel propriétaire à une solution ouverte peut s'avérer très coûteux si l'application est étroitement intégrée dans le système informatique en place et qu'il existe des interfaces ou d'autres dépendances. Par conséquent, il ne vaut la peine de passer à un OSS qu'à la fin du cycle de vie d'un logiciel propriétaire.

2. Fonctionnalités manquantes ou absence de solution OSS appropriée

Lorsqu'il s'agit d'acheter un logiciel standard, il peut arriver de ne pas trouver d'OSS qui permette de remplacer la solution propriétaire utilisée jusque-là ou que les fonctionnalités de l'OSS de remplacement ne soient pas suffisantes. Dans ce cas, une approche consiste à collaborer avec d'autres utilisateurs pour développer les fonctionnalités manquantes et les ajouter à un projet OSS¹⁸.

3. Coûts d'intégration plus élevés

Les économies réalisées à court terme sur les licences pour un OSS sont souvent contrebalancées pour les coûts plus importants que génère l'intégration de l'OSS. Il faut donc avoir conscience que l'utilisation d'OSS n'est pas synonyme de gratuité, mais qu'elle génère des coûts internes et externes.

4. Responsabilités et support à définir

Les OSS sont souvent critiqués pour le manque de support et de maintenance fournis par des entreprises externes. Aujourd'hui, il existe cependant de nombreux prestataires qui proposent des services commerciaux pour les solutions OSS (support à long terme, maintenance continue, développement, responsabilité civile et garantie...). Le chap. 8 de l'annexe D présente les différents modèles d'affaires OSS et les différents modèles de support. Une vue d'ensemble actuelle des fournisseurs OSS et des services qu'ils proposent est disponible dans l'OSS Directory (<https://www.ossdirectory.com>) ou dans l'étude [OSS2024]¹⁹.

¹⁸ Ces nouvelles fonctionnalités relèvent alors d'un projet ou de la compétence des responsables d'application. Les modalités exactes sont réglées dans le projet. [Em002-2] et [Em002-4] aident à prendre des décisions sur ces modalités. En fonction de l'importance de la fonctionnalité, il n'est pas nécessaire de suivre l'intégralité des instructions. Il faut toujours respecter les aspects liés au droit des marchés publics en ce qui concerne les ressources.

¹⁹ Voir <https://www.oss-studie.ch/>

5. Un marché parfois restreint avec peu de fournisseurs

Comme certaines applications OSS, telles que les applications de bureau, ne demandent que peu ou pas de support ni de maintenance, le marché des fournisseurs est dans certains cas très restreint, voire inexistant. Un appel d'offres pour des services OSS peut dans ce cas contribuer à créer un marché susceptible de renforcer la concurrence entre les fournisseurs. Dans la perspective d'un appel d'offres pour de telles prestations, l'existence d'une large communauté de développement pour le logiciel en question et la qualité de la documentation de développement revêtent une importance pratique.

6. Visibilité réduite

Les communautés de projets OSS se concentrent principalement sur le développement de produits et peu sur leur commercialisation. À l'inverse, les fabricants de logiciels propriétaires investissent beaucoup dans le marketing et la vente de leurs produits. Ce manque de publicité pour les OSS donne souvent la fausse impression qu'il n'existe pas d'alternative aux produits propriétaires. Il existe pourtant des plateformes comme [alternativeTo²⁰](https://alternativeto.net/), qui offre une vue d'ensemble des solutions disponibles pour des tâches.

7. Acceptation moindre de la part des utilisateurs finals

Interface utilisateur inhabituelle, fonctionnalités manquantes, mauvaise ergonomie et absence de publicité, ces caractéristiques des OSS rendent certains produits difficiles à accepter pour les utilisateurs finals. D'où l'importance de miser sur la communication, la documentation et l'offre de formation. Les OSS peuvent aussi être mieux acceptés lorsqu'on sait qu'il est possible de les développer en fonction des besoins des utilisateurs.

8. Peu ou pas de savoir-faire interne

Les OSS évoluent rapidement et nécessitent des connaissances techniques approfondies. Il peut donc arriver que les collaborateurs internes ne disposent pas du savoir-faire nécessaire pour certains OSS. Un tel savoir-faire peut être mis en place grâce à la formation continue, à des offres d'autoapprentissage à partir de sources en ligne ou encore à la mise en place de wikis propre à l'entreprise.

9. Un futur difficile à anticiper

Pour une personne qui ne fait pas partie de la communauté d'un OSS, il est souvent difficile de savoir comment un produit est appelé à évoluer. C'est pourquoi il est important de pouvoir faire une estimation réaliste de l'évolution future d'une solution OSS. La plateforme Open Hub présentée au chap. 7.2 permet d'estimer l'activité et l'hétérogénéité de la communauté de développement et donc de mieux appréhender l'évolution future d'un projet OSS.

²⁰ <https://alternativeto.net/>

10. Incertitudes juridiques

La multitude de licences *open source* et le nombre encore faible de décisions de justice concernant les questions d'interprétation créent parfois des incertitudes juridiques concernant les OSS. Le guide pratique vise à donner une vue d'ensemble des principales licences *open source*, de leurs caractéristiques et de leurs compatibilités. Les sources permettant de trouver des réponses approfondies aux questions juridiques se trouvent dans les documents *Em002 Guide Licences OSS* et *Em002-6 FAQ OSS*.

11. Offre trop vaste d'OSS

Ces dernières années, l'offre d'OSS est devenue pléthorique. Les utilisateurs potentiels d'OSS se plaignent ainsi qu'il y a trop d'OSS sur le marché. Le chap. 7 Trouver des solutions OSS de remplacement présente deux plateformes qui permettent de comparer les offres, à savoir *alternativeTo* et *Open Hub*.

12. Efforts de coordination en cas de développement collaboratif

Lorsque l'administration fédérale prend part à des collaborations et les développe ou qu'elle doit entraîner une communauté, elle doit fournir des efforts supplémentaires de coordination. Il convient dans l'idéal de faire en sorte que cette charge supplémentaire soit compensée par une amélioration de la solution, une meilleure acceptation ou une participation de tiers.

4. Utilisation et traitement d'OSS : les cas possibles

4.1 Maturité des OSS dans l'administration

Lorsque l'administration fédérale utilise et développe des OSS, les obligations en lien avec les licences peuvent être différentes selon les cas de figure. Un modèle de maturité²¹ en vue du déploiement d'OSS dans une organisation peut se présenter comme suit :

1. Dénier – pas d'utilisation ou utilisation inconsciente d'OSS
2. Consommation – utilisation passive d'OSS
3. Participation – engagement dans les communautés open source (contribution)
4. Contribution v contributions pragmatiques aux projets d'OSS (contribution)
5. Leadership – Participation stratégique à l'OSS en vue d'augmenter la valeur de l'entreprise (collaboration)²²

Cas de figure	Effet
La simple utilisation d'un OSS existant (sans modification du code de programmation)	Il n'y a aucune obligation de diffuser le code source.
Le développement complet d'un logiciel à publier sous une licence <i>open source</i> .	Dans ce cas, le choix de la licence sous laquelle le logiciel sera publié est entièrement libre. Pour la Confédération, il faut toutefois se conformer au document <i>Em002-3 Guide Licences OSS</i> et tenir compte des documents <i>Em002-2 Instructions pour la publication d'OSS</i> et <i>Em002-4 Guide Communauté OSS</i> .
Le développement (interne ou externe) de composants à code source ouvert (avec ou sans effet copyleft ²³), lorsqu'il est prévu d'utiliser le logiciel uniquement au sein de la même organisation sans le distribuer.	Les composants OSS peuvent être combinés à volonté avec des composants tiers. Il n'y a pas d'obligation de publication du point de vue de la licence, tant que le logiciel est utilisé exclusivement au sein de la même organisation et n'est pas redistribué (exception AGPL). Comme la notion de « distribution au sein de la même organisation » n'est pas clairement définie sur le plan juridique, le scénario concernant les composants OSS soumis au copyleft n'entre en ligne de compte que dans des cas exceptionnels. Dans ce cas, il faut consulter le service juridique de votre organisation. Voir <i>Em002-3 Guide Licences OSS</i> et <i>Em002-6 FAQ OSS</i> Voir aussi <i>Em002-2 Instructions pour la publication de</i>

²¹ Modèle de maturité inspiré de <https://todogroup.org/resources/guides/a-guide-to-outbound-open-source-software/#maturity-levels>

²² La création n'est pas mentionnée, car elle découle directement de l'art. 9 LMETA.

²³ Voir *Em002-3 Guide Licences OSS* pour la notion de copyleft

	<p><i>logiciels ouverts et Em002-4 Guide Communauté OSS. La modification doit toujours être publiée en raison de l'obligation prévue à l'art. 9 LMETA. Le plus simple est de l'annoncer comme changement dans le projet OSS existant.</i></p>
<p>Le développement (interne ou externe) uniquement à partir de composants ouverts sans effet copyleft, lorsqu'il est prévu de distribuer le logiciel également à l'externe (p. ex. aux cantons).</p>	<p>Les composants OSS peuvent être combinés à volonté avec des composants tiers sans obligation de publication découlant d'une licence (voir aussi <i>Em002-3 Guide Licences OSS</i>). L'art. 9 LMETA exige en revanche la mise à disposition. Celle-ci se fait dans l'idéal dans le cadre du projet existant (pas de fork). Voir aussi <i>Em002-2 Instructions pour la publication de logiciels ouverts</i> et <i>Em002-4 Guide Communauté OSS</i>.</p>
<p>Le développement (interne ou externe) à partir de composants à code source ouvert avec effet copyleft, lorsqu'il est prévu de distribuer le logiciel également à l'externe (p. ex. aux cantons).</p>	<p>La publication est obligatoire en raison de la licence copyleft (voir le document <i>Em002-3 Guide Licences OSS</i> concernant l'effet copyleft). Voir aussi <i>Em002-2 Instructions pour la publication de logiciels ouverts</i> et <i>Em002-4 Guide Communauté OSS</i>.</p>
<p>La contribution à un projet OSS existant.</p>	<p>La LMETA impose la publication. On utilise alors la gouvernance et la licence prévues par le projet. Voir aussi <i>Em002-2 Instructions pour la publication de logiciels ouverts</i> et <i>Em002-4 Guide Communauté OSS</i>.</p>
<p>Le développement commun d'un projet nouveau ou existant avec une communauté et un partage des coûts.</p>	<p>La LMETA impose la publication. Le projet et sa gouvernance sont mis en place en se fondant sur les documents <i>Em002-2 Instructions pour la publication de logiciels ouverts</i>, <i>Em002-3 Guide Licences OSS</i> et <i>Em002-4 Guide Communauté</i>.</p>
<p>Toutes les options de développement sont réalisées par un fournisseur ou un mandataire.</p>	<p>La LMETA impose la publication. Voir les documents <i>Em002-2 Instructions pour la publication de logiciels ouverts</i>, <i>Em002-3 Guide Licences OSS</i> et <i>Em002-4 Guide Communauté</i>. La procédure d'acquisition s'attachera, si possible, à créer les conditions nécessaires à la publication et, le cas échéant, à communiquer au fournisseur ou à des tiers les exigences concernant la mise à disposition.</p>

4.2 Utilisation d'OSS non modifiés

Toutes les licences *open source* (voir chap. 2, Définitions) permettent une utilisation illimitée des OSS, quel que soit le nombre de postes de travail équipés, le nombre d'utilisateurs connectés, le nombre de serveurs et de processeurs, etc. La question de la licence utilisée n'est donc pas pertinente.

L'utilisation d'un OSS non modifié n'entraîne **aucune obligation de publication**, même pour les programmes placés sous une licence avec un effet copyleft strict (exception : licence AGPL, Affero General Public License).

Tant que l'OSS est déployé et exploité par les fournisseurs de prestations internes de la Confédération, cela peut se faire **sans appel d'offres public**, car la licence open source elle-même ne coûte en général rien au service d'achat et n'est donc pas en soi pertinente pour l'achat.

Lorsque la maintenance, le support et d'autres services doivent être confiés à un prestataire externe, le droit des marchés publics s'applique. Le chap. 9 présente différentes options pour le support. Il est en outre important de définir des critères d'aptitude et d'adjudication appropriés lors d'acquisition en lien avec un OSS. Voir le chap. 8 pour les bases.

4.3 Développement avec des composants OSS et mise à disposition du code source

Pour se conformer à l'obligation de publication prévue à l'art. 9 LMETA, il faut suivre la procédure décrite dans le document *Em002-2 Instructions pour la publication de logiciels ouverts*.

Le *Em002-3 Guide Licences OSS* et le *Em002-4 Guide Communauté OSS* doivent être consultés si nécessaire. L'objectif est de mettre le projet à disposition de manière contrôlée en s'appuyant sur les trois listes de contrôle [*Em002-2.1*], [*Em002-2.2*], [*Em002-2.3*].

4.4 Contribution à un projet ouvert

Des réflexions concernant une possible contribution de l'administration fédérale à des projets ouverts existants figurent au chap. 4.2 du guide [*BITKOM2024*]²⁴. Selon l'importance du projet pour l'administration fédérale, il faut évaluer les responsabilités que l'office concerné veut et peut endosser.

La collaboration à un projet ouvert et le développement direct dans le cadre d'un tel projet peuvent aussi se dérouler sur la base du document sur la mise à disposition. L'accent est alors mis sur le document *Em002-4 Guide Communauté OSS* et sa liste de contrôle [*Em002-4.1*].

5. Caractéristiques et choix des licences *open source*

Voir le document *Em002-3 Guide Licences OSS*.

²⁴ <https://www.bitkom.org/Bitkom/Publikationen/Bitkom-Leifaden-zu-Open-Source-Software-20.html>

6. Acquisition d'OSS

Les questions stratégiques en lien avec l'acquisition d'OSS sont traitées dans le document *Em002-7 Acquisition et OSS - Aspects stratégiques*.

Le CCMP a publié une nouvelle notice *Acquisition de logiciels et art. 9 LMETA* [[CCMP-MB](#)].

L'[intranet de l'OFCL](#) (uniquement accessible depuis le réseau de la Confédération) propose des aides supplémentaires, à savoir les [directives sur les OSS dans les procédures d'acquisition](#) [[OFCL-WL](#)] (en allemand) et la [liste de contrôle sur les exceptions à l'art. 9 LMETA](#) [[OFCL-CL](#)] (en allemand).

En ce qui concerne l'*enterprise readiness* (préparation à l'entreprise) de l'OSS, consulter le document [[Gu2024](#)] ou le guide [[BITKOM2024](#)]²⁵.

²⁵ <https://www.bitkom.org/Bitkom/Publikationen/Bitkom-Leifaden-zu-Open-Source-Software-20.html>

7. Trouver des solutions OSS

Différentes plateformes permettent aux organisations intéressées de trouver et d'analyser des OSS avant leur utilisation ou leur acquisition. Les données qui en résultent constituent alors une base décisionnelle.

Dans tous les cas, il faut disposer d'une compilation des exigences minimales et éventuellement d'une analyse de marché.

7.1 alternativeTo

Un outil pratique pour trouver des alternatives à un logiciel est le site *alternativeTo – Crowdsourced software recommendations*²⁶. Comme le nom du site l'indique, les solutions qu'il propose et leur évaluation sont générées par *crowdsourcing*, ou production participative, c'est-à-dire par les contributions de nombreux utilisateurs. Le site distingue quatre types de logiciels dont seul le premier peut être considéré comme un OSS au sens de l'art. 9 LMETA.

Free Open Source	Logiciel publié sous une licence <i>open source</i>
Free	Logiciel gratuit (freeware), mais dont le code source n'est pas librement accessible et qui ne peut être modifié
Freemium	Logiciel qui propose une version gratuite et une version premium, et dont la version gratuite permet d'utiliser toutes les principales fonctions. Les logiciels qui offrent gratuitement une période d'essai (p. ex. utilisation gratuite pendant un mois) ne font pas partie de cette catégorie, mais de la catégorie <i>Commercial</i> .
Commercial	Logiciel propriétaire avec licence payante

7.2 Dépôts d'OSS

Le service de développement *open source* le plus populaire au monde est actuellement GitHub. Il compte plus de 30 millions d'utilisateurs et plus de 100 millions de répertoires de code source²⁷. Aujourd'hui, pratiquement toutes les entreprises informatiques, en particulier les fabricants de logiciels propriétaires, sont présentes sur GitHub, de même que de nombreuses autres entreprises et organisations de toutes sortes. Toutes y publient des applications à code source ouvert, des outils, des bibliothèques de programmation, etc. Les autorités sont également de plus en plus nombreuses à publier leurs propres OSS sur *GitHub and Government* (<https://government.github.com>), parmi lesquelles des autorités suisses²⁸. Mais il existe aussi d'autres dépôts.

Il s'agit par exemple de

- Gitlab
- BitBucket
- Sourceforge
- Launchpad

Le service *Insights* de GitHub fournit de nombreuses statistiques pertinentes sur un projet d'OSS donné.

²⁶ <https://alternativeto.net/>

²⁷ <https://octoverse.github.com>

²⁸ <https://government.github.com/community/#switzerland>

Pulse	Vue d'ensemble des dernières activités concernant un projet OSS sur GitHub : résumé des données principales sur l'intensité du développement, l'hétérogénéité de la communauté, les annonces en cours de traitement, les améliorations (demande de tirage) ...
Contributeurs	Présentation des développeurs et de leurs activités (qui, quand, comment) : cet indicateur est important pour savoir si tout ne dépend que d'une personne ou s'il s'agit du travail d'une plus grande communauté.
Commits	Affichage des contributions au projet et indication du moment où elles ont été réalisées.
Fréquence de code	Visualisation des ajouts et des suppressions de contenu dans un code source
Graphe des dépendances	Dépendances d'un projet OSS par rapport à d'autres OSS (p. ex. pour identifier les mises à jour et les failles de sécurité)
Réseau	Représentation de l'historique des contributions d'un développeur à une branche de développement
Forks (duplication)	Liste de toutes les copies du projet OSS sur GitHub : indicateur de la popularité et de la distribution d'un OSS

7.3 Open Hub

Si l'on souhaite collecter des informations sur une solution à code source ouvert qui n'est pas nécessairement développée sur GitHub, Open Hub de l'entreprise Black Duck²⁹ est une bonne solution. Elle regroupe sous une forme compréhensible une multitude d'informations importantes sur près d'un demi-million de projets à code source ouvert.

Project Summary	Brève description du projet
In a Nutshell	Principaux faits concernant un projet, nombre de commits, de contributeurs, de lignes de code, indication du langage de programmation le plus utilisé, de la date du premier commits et de la dernière modification. Évaluation du codebase et de la taille de l'équipe de développeurs (p. ex. « Mozilla Firefox has a well-established, mature codebase maintained by a very large development team with stable Y-O-Y commits. »)
Quick Reference	Informations sur l'organisation, lien vers le projet, vers le code et vers des projets similaires
Licences	Indication de la ou des licence(s) auxquelles est soumis le projet et des conséquences qui en découlent
Project Security	Information sur la sécurité et la vulnérabilité du projet
Code	Graphique montrant l'évolution dans le temps du nombre de lignes de code en fonction du langage utilisé
Activity	Graphique montrant le nombre de commits par mois. Résumé des 30 derniers jours et des 12 derniers mois.
Communauté	Graphique montrant le nombre de contributeurs actifs par mois. Notation du projet sur une échelle de une à cinq étoiles

²⁹ <https://www.openhub.net/>

Open Hub offre en outre la possibilité de comparer plusieurs projets.³⁰ On peut ainsi rapidement déterminer le projet qui a la communauté la plus active, dont la durée de développement est la plus longue ou qui propose la licence appropriée.

7.4 Dépôts de code particuliers

À partir d'une certaine taille, les organisations ont leurs propres dépôts de code. C'est notamment le cas des autorités allemandes avec le dépôt `opencode.de`³¹. Ce sont justement les dépôts d'autres administrations publiques qui sont intéressants.

7.5 OSS Directory

Les organisations qui travaillent avec des OSS tiennent parfois des répertoires d'OSS intéressants et des entreprises qui soutiennent la démarche. En Suisse, il s'agit de l'OSS Directory de CH Open³².

³⁰ https://www.openhub.net/p/_compare

³¹ <https://opencode.de/de>

³² <https://www.ossdirectory.com/fr/home>

8. Modèles de support

Les OSS disponibles sur le marché peuvent être utilisés de trois manières :

1. Sans support professionnel
2. Avec support interne
3. Avec support externe

Les avantages et inconvénients des différentes possibilités sont expliqués ci-après. Le scénario approprié dans un cas donné doit être choisi en fonction de la situation et le choix dépend de trois facteurs principaux : l'importance stratégique de l'OSS, son intégration technique et les ressources en personnel disponibles.

Un support professionnel interne ou externe est obligatoire pour les logiciels critiques. Le cycle de vie prévu joue aussi un rôle dans la planification du support. Le support peut aussi être assuré par plusieurs prestataires.

8.1 Utilisation sans support professionnel

L'OSS est téléchargé gratuitement sur Internet, installé et utilisé en l'état.

Avantages	<ul style="list-style-type: none"> - Faibles coûts - Mise en œuvre rapide
Inconvénients	<ul style="list-style-type: none"> - Pas de support garanti - Pas de prétention en matière de responsabilité
Risque et Couverture	Risque élevé : il n'existe aucun contrat de support ou de garantie et le savoir-faire des développeurs est faible, voire inexistant, au sein de l'organisation.
Domaine d'utilisation type	OSS standard pouvant être actualisé indépendamment d'autres systèmes

8.2 Utilisation avec support interne

Une entreprise ou une institution publique développe un savoir-faire et des ressources sur certaines solutions ouvertes afin de les utiliser à long terme. C'est souvent le cas dans les secteurs critiques pour une entreprise.

Avantages	<ul style="list-style-type: none"> - Grande flexibilité grâce au savoir-faire interne - Pas de dépendance vis-à-vis des fournisseurs
Inconvénients	<ul style="list-style-type: none"> - Investissements élevés et temps important consacré à l'acquisition de savoir-faire - Coûts fixes internes plus élevés pour les collaborateurs
Risque et Couverture	Risque moyen : le support dépend du savoir-faire et de la disponibilité du service informatique interne.
Domaine d'utilisation type	Développements internes, OSS à usage stratégique pour lequel un grand savoir-faire est disponible

8.3 Utilisation avec support externe

Un fournisseur d'OSS externe professionnel accompagne le déploiement et la maintenance du logiciel. C'est souvent l'option choisie pour les secteurs critiques de l'entreprise où il est important de disposer immédiatement d'un savoir-faire approfondi. Cette option peut également inclure une partie ou la totalité de la mise à disposition du code source.

Avantages	<ul style="list-style-type: none"> - Accès direct au savoir-faire des développeurs d'OSS - Correction et développement sur une base contractuelle - Sélection de différents fournisseurs d'OSS - Engagement, couverture des risques de conformité
Inconvénients	<ul style="list-style-type: none"> - Coûts externes liés aux fournisseurs d'OSS - Savoir-faire dépendant du fournisseur d'OSS
Risque et Couverture	Risque faible : la garantie s'effectue conformément au contrat ou à l'accord de niveau de service.
Domaine d'utilisation type	OSS critique pour l'entreprise, pour lequel il n'existe pas ou peu de savoir-faire interne en matière de développement

9. Interlocuteur

Dans l'administration fédérale, il n'existe pas d'entité de type *open source program office*, qui ferait office d'interlocuteur central responsable pour les questions liées aux OSS. La mise en œuvre incombe aux offices.

Les demandes générales concernant les aides OSS du groupe de document *Em002* peuvent être adressées au secteur TNI de la ChF: opensource@bk.admin.ch.

Annexes

A. Changements par rapport à la version précédente

- Chap. 4.1: ajout de degré de maturité
- Chap. 6: modification et ajout des nouveaux documents de l'OFCL pour l'acquisition d'OSS
- Autres petites modifications et améliorations rédactionnelles

B. Références

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

C. Abréviations

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

Voir le glossaire dans le document *Em002-6 FAQ OSS*

D. Modèle d'affaires OSS

Un OSS n'est pas un modèle d'affaires en soi, étant donné que la vente d'un logiciel sous une licence *open source* n'est pas une option, contrairement à ce qui se fait pour les logiciels propriétaires. Mais les entreprises ont quand même plusieurs possibilités qui leur permettent d'appliquer des modèles d'affaires qui reposent sur un OSS. L'acquisition de services commerciaux est notamment incontournable lorsqu'il s'agit de confier le support professionnel pour un OSS à un tiers externe. Les quatre modèles commerciaux les plus courants avec les OSS sont présentés ci-après. Pour davantage de modèles et plus de détails, voir le guide [BITKOM2023].

D.1 Services et produits basés sur des OSS

Les entreprises peuvent proposer des services commerciaux basés sur des OSS, comme l'hébergement web ou des services en nuage, qui coûteraient bien plus cher en utilisant des logiciels propriétaires. Ainsi, la plupart des startups, des portails en ligne et des fournisseurs de commerce électronique mettent aujourd'hui en place des plateformes fondées sur les OSS. D'autres entreprises technologiques, comme les opérateurs de télécommunication, les fournisseurs de services de streaming ou même les fabricants de logiciels propriétaires, intègrent des OSS dans le matériel informatique, dans leurs logiciels et dans leurs services en ligne. Elles peuvent de cette manière innover en permanence, ce qui n'est pas toujours possible avec des logiciels propriétaires.

D.2 Services pour les OSS

Les fournisseurs d'OSS proposent des services spécifiques pour certains OSS. Ils disposent de développeurs expérimentés et peuvent donc proposer des services de support, de maintenance, d'exploitation, de développement, de conseil, de formation et autres. Ces services peuvent être fournis dans le cadre d'un mandat ou d'un contrat d'entreprise. Ils peuvent faire l'objet d'un appel d'offres public, car il n'y a pas de rapport de dépendance avec le fournisseur (enfermement propriétaire). Les appels d'offres doivent retenir des critères particuliers qui permettent de choisir effectivement le meilleur prestataire (voir chap. 0 et 8, chap. 3.3 [BITKOM2023] et [Gu2024]).

D.3 Souscriptions

Par souscription, on entend une sorte d'accord sur les prestations ou d'abonnement concernant la fourniture récurrente de services standardisés pour un OSS. Dans le cadre d'une souscription, une entreprise s'engage par exemple à fournir régulièrement des mises à jour de sécurité et une assistance par email ou par téléphone, à certifier la compatibilité de l'OSS avec d'autres logiciels et d'autre matériel informatique et à garantir la maintenance sur le long terme et la protection contre les revendications juridiques (droits d'auteur, brevets). En contrepartie, le client souscrit un abonnement payant pour chaque poste de travail ou processeur, comme il le ferait pour le droit de licence ou d'utilisation d'un logiciel propriétaire. En revanche, les souscriptions concernant des OSS ne constituent pas une condition préalable à l'utilisation du logiciel, elles représentent uniquement une valeur ajoutée au service d'un fournisseur d'OSS.

D.4 Double licence

Lorsqu'une entreprise détient la propriété intellectuelle d'un logiciel et que tous les composants à code source ouvert intégrés sous soumis à une licence permissive, il est possible d'utiliser une double licence ou une licence multiple. De cette manière, l'entreprise de développement peut à la fois publier son logiciel sous une licence *open source* copyleft et le vendre sous une licence propriétaire. La version commerciale est souvent qualifiée de version « entreprise » et comprend en général des fonctionnalités supplémentaires, comme l'intégration d'interfaces exclusives ou l'autorisation donnée à l'acheteur d'intégrer le logiciel dans son propre produit propriétaire. Selon l'importance des restrictions imposées dans la version à code source ouvert, la prudence est de mise lors de l'utilisation de logiciels à double licence, car l'achat de la version « entreprise » peut s'avérer indispensable et la dépendance vis-à-vis du fabricant est donc aussi élevée que dans le cas des logiciels propriétaires habituels.