Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

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

14.09.2026

Em002-4 Guide des communautés OSS

Version 2.0-f4b576a

Table des matières

1. L’essentiel en bref 3
2. Introduction 3
3. Type, objectif et but d’une communauté 4
3.1. Cas particulier : nouvelle collaboration pour le développement, la maintenance et le support de logiciels open source 5
3.1.1. Dimension juridique 6
3.1.2. Dimension de droit des marchés publics 6
3.2. Cas particulier : adhésion à une collaboration existante 6
3.3. Cas particulier : contributions de code à des projets OSS de tiers 6
4. Concept de communauté 7
5. Décisions de principe lors de la constitution d’une communauté 9
5.1. Gestion du produit 11
5.1.1. Organisations au sein de la Confédération : 11
5.1.2. Commun (la Confédération dirige ou partenaires sur un pied d’égalité) 12
5.1.3. Organisation ouverte 12
5.1.4. Tiers 12
5.2. Intégration des fournisseurs 12
5.3. Répartition des coûts 13
5.4. Support 13
5.5. Développement 14
6. Contenus détaillés pour le concept de communauté 14
6.1. Données clés 15
6.2. Objectifs 15
6.3. Organisation et gouvernance 15
6.4. Feuille de route et processus de modification 18
6.5. Processus de développement 18
6.6. Partage des coûts et fournisseurs 19
6.7. Marketing 20
6.8. Traitement des contributions externes 21

6.9. Exploitation de l’application 22

7. Informations importantes pour les responsables de communauté 23

7.1. Cela prend du temps 23

7.2. Autres suggestions pour la constitution d’une communauté 23

7.3. Communication 23

7.4. Développement ouvert 23

7.5. Gestion des utilisateurs 23

7.6. Fusion de communautés 24

7.7. Traitement des contributions de tiers 24

7.8. Signalements relevant de la sécurité 24

7.9. Éviter les forks 24

7.10. Répartition des coûts 25

7.11. Prestations complémentaires selon l’art. 9, al. 5 et 6, LMETA 25

7.12. Souveraineté numérique 26

8. Annexe 26

8.1. Modifications par rapport à la version précédente 26

8.2. Références 26

8.3. Abréviations 27

8.4. Exemples de concepts de communauté existants 27

8.5. Exemples de collaborations 27

8.6. Collaboration : ZenDiS en Allemagne 27

8.7. Collaboration : Open Trip Planner (OTP) 28

8.8. Nouvelle collaboration initiée par la Confédération : Swiss Trust Broker 28

8.9. Nouvelle collaboration initiée par les cantons : inosca 28

8.10. SNOWPACK du WSL — contributions directes de développeurs 29

8.11. Inspiration pour des statuts d’association 29

8.12. Exemples de support et d’émoluments 29

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

1. L'essentiel en bref

Les points les plus importants du document sont énumérés ci-après.

- La constitution ou l'exploitation d'une communauté OSS **n'est pas exigée** par l'art. 9 LMETA.
- Une communauté est pertinente lorsqu'un groupe d'utilisateurs a du sens et que des **synergies** avec d'autres utilisateurs du logiciel doivent être créées.
- Une communauté ne se limite pas au logiciel lui-même, mais peut aussi englober les processus environnants et la standardisation des données.
- La constitution d'une communauté implique toujours un **effort initial**. Celui-ci se poursuit en règle générale. Dans de nombreux cas, il est judicieux que l'office pilote supporte également les coûts de coordination.
- Les communautés doivent remplir un **but**. Il faut des personnes intéressées et des participants.
- Une communauté peut émerger puis être formalisée.
- L'objectif de la communauté est d'accroître le **bénéfice pour tous**. Principe général : « on récupère au moins autant que ce que l'on apporte ».
- La communauté devrait être **structurée aussi simplement que possible**.
- Dans un premier temps, la communauté devrait être brièvement examinée à l'aide de la liste de contrôle Em002-4.1. Ce n'est que si elle est pertinente qu'elle devrait être développée.
- La formalisation s'effectue au moyen d'un **concept de communauté**. Il convient de reprendre autant que possible des sources existantes. Principe général : « ne pas réinventer la roue ».
- Si une collaboration est visée, cela doit être pris en compte dès le tout début du projet, car les partenaires potentiels devraient déjà être associés lors de l'analyse des exigences. Cela doit aussi être bien planifié dès le départ sur le plan juridique et sous l'angle du droit des marchés publics.
- Les possibilités d'adhérer à une collaboration existante doivent être examinées sur le plan juridique et sous l'angle du droit des marchés publics.
- Aucune communauté ne doit être constituée pour des contributions. Toutefois, si le projet l'exige, le **Contributor Licence Agreement (CLA)** correspondant doit être signé, ou les développeurs doivent être autorisés par le projet à soumettre le logiciel sous un Developer Certificate of Origin (DCO).

2. Introduction

Le présent guide s'adresse aux spécialistes TIC responsables d'une application qui souhaitent la proposer en open source ou contribuer à une telle application.

Cela doit être organisé soit formellement, soit informellement.

Le présent document fournit des informations importantes pour l'organisation et la constitution d'une communauté[1].] ainsi que pour le développement ouvert directement en tant que projet open source.

En deuxième élément, afin de favoriser la standardisation, un répertoire de concepts de communauté existants est tenu dans le cadre du présent document. L'idée est que de nouvelles communautés puissent être constituées avec un effort minimal sur la base de solutions éprouvées.

Le présent document ne contient pas de recommandations directes pour l'organisation de communautés autour de bibliothèques logicielles (parties de logiciels remplissant des fonctions partielles, p. ex. la génération de PDF). Normalement, aucun concept distinct n'est élaboré pour celles-ci ; elles utilisent plutôt le modèle simple défini dans le modèle de dépôt.

Le document Em002-01 Guide pratique pour les logiciels open source dans l'administration fédérale contient, au chapitre 4, les formes fondamentales de collaboration, au chapitre 8 les différents modèles de support et, en annexe, des informations sur le comportement de marché de l'open source.

3. Type, objectif et but d'une communauté

Une communauté peut poursuivre les buts suivants :

- Recensement d'exigences supplémentaires
- Diffusion de connaissances sur l'application
- Meilleurs retours des utilisateurs
- Promotion des traductions, de la documentation et des formations
- Diffusion de la standardisation
- Interaction globalement améliorée avec les utilisateurs
- Extension de la coopération à d'autres parties
- Promotion d'autres logiciels fondés sur le logiciel

1. La communauté d'un projet open source représente typiquement une pyramide. + La base de cette pyramide est constituée des utilisateurs du logiciel, en particulier ceux qui s'engagent et participent activement à la communauté, par exemple sous forme de signalements d'erreurs et de demandes de fonctionnalités ou par des contributions à des listes de diffusion. + Directement au-dessus, dans la pyramide, se trouvent les contributeurs. Il s'agit de parties de la communauté qui proposent leurs propres contributions de code. Ils n'ont typiquement pas d'accès en écriture au dépôt. Leurs contributions sont examinées par les mainteneurs du projet et intégrées au dépôt après avoir atteint la qualité propre au projet. + Au sommet de la pyramide se trouvent les mainteneurs ou committers d'un projet. Dans ce rôle, un développeur assume une responsabilité accrue. Cela se traduit par exemple par la possibilité d'accepter des contributions. Ce droit se manifeste souvent par un accès en écriture au dépôt. À ce niveau s'exerce le contrôle du logiciel, de sa qualité et de son étendue fonctionnelle. Dans les projets plus complexes, ce niveau peut être subdivisé. Ainsi, le noyau Linux compte notoirement des mainteneurs de sous-systèmes à plusieurs niveaux, et en définitive un seul développeur, Linus Torvalds, intègre les correctifs dans le dépôt du projet. Un autre exemple est celui des projets de l'Eclipse Foundation, qui disposent typiquement d'un project lead doté de droits supplémentaires dans le cadre du processus de développement de l'Eclipse Foundation, comme le déclenchement du processus de publication ou l'élection formelle de committers ou de project leads *[BITKOM2023*

Le type de communauté le mieux adapté à une application dépend fortement de l'environnement métier, des utilisateurs potentiels et de la stratégie de l'institution qui publie. Il existe de nombreuses manières de constituer des communautés. Idéalement, un projet approprié peut rejoindre une communauté existante. Une communauté peut aussi être constituée avec un ou plusieurs partenaires sur un pied d'égalité.

Il importe qu'une **structure de gouvernance** soit établie et que les points déterminants soient documentés dans un **concept de communauté** (voir aussi *[BITKOM2023]* chapitre 4.1 et *[IZqCab2023]*).

Il convient de noter que les communautés logicielles peuvent aussi être combinées avec celles portant sur les données, la standardisation et les thèmes métier. La communauté peut alors englober processus, standardisation, logiciels et données. Des cadres existants tels qu'eOperations, l'ANS et eCH peuvent également être utilisés lorsque cela est judicieux.

Un aspect important d'une communauté qui fonctionne est que les participants reçoivent plus qu'ils n'investissent.

Les constellations suivantes existent :

- **Communauté autour d'un OSS de la Confédération** : la gouvernance et le concept de communauté doivent être élaborés. Dans ce contexte, il faut aussi régler la manière dont des tiers peuvent apporter des contributions.
- **Nouvelle collaboration pour résoudre un problème** : outre la gouvernance et le concept de communauté, toute la structure juridique de la collaboration doit être mise en place.
- **Adhésion à une collaboration** : l'administration fédérale doit examiner si et comment elle peut le faire.
- **Contribution à un projet de tiers** : le point à vérifier est la politique du projet en matière de contributions (Contributor Licence Agreement, Developer Certificate of Origin, etc.).

Les questions de droit des marchés publics sont traitées dans Em002-7 Aspects stratégiques des achats et des logiciels open source.

3.1. Cas particulier : nouvelle collaboration pour le développement, la maintenance et le support de logiciels open source

Les coûts globaux d'un logiciel open source diminuent lorsqu'il est développé et entretenu en commun. Cela peut se faire sans accords de coopération formels, ou avec rien de plus qu'une planification commune.

Lorsqu'une collaboration plus formelle est recherchée, une analyse de marché devrait être effectuée afin de s'assurer que la collaboration est la forme de coopération la plus appropriée. Les collaborations sont plus avantageuses que le développement individuel par chaque partie (voir Em002-4.1).

Il convient de noter que pour les collaborations, il est judicieux d'engager la collaboration très tôt dans la méthodologie de projet HERMES[2] — concrètement durant les phases d'analyse de la situation et des exigences.

La collaboration est en principe admise en Suisse selon l'article 4 LMETA.

2. https://www.hermes.admin.ch/fr/gestion-de-projet/apercu-de-la-methode.html

Outre toutes les autres considérations, les collaborations comportent une dimension juridique et une dimension de droit des marchés publics. À l'heure actuelle, chaque cas est apprécié individuellement. Avec le temps, des modèles standardisés s'établiront.

3.1.1. Dimension juridique

Les adhésions à des associations sont envisageables, même si elles sont généralement appréciées au cas par cas : qui se trouve derrière l'association et quelles obligations sont contractées ? Une délégation des adhésions à eOperations ou à des organisations similaires serait possible.

Tant qu'aucun argent ne circule et qu'aucune obligation formelle ne doit être contractée, un Memorandum of Understanding au niveau de l'office fédéral ou de la Chancellerie fédérale est possible.

Une situation exigeant la conclusion d'un traité international devrait être évitée.

Il existe en outre deux cas particuliers[3] dans lesquels une collaboration serait déjà couverte :

- Les marchés adjugés en vertu d'un accord international concernant la réalisation conjointe d'un projet par des États signataires.
- Les marchés adjugés conformément aux procédures ou conditions particulières d'une organisation internationale.

3.1.2. Dimension de droit des marchés publics

Dans de nombreux cas, la collaboration se déroule entre organisations du secteur public. Dans de tels cas, l'acquisition ne pose pas de problème lorsque la configuration est « intra-étatique » (voir Em002-7 Aspects stratégiques des achats et des logiciels open source).

Des contrats avec des entreprises étrangères non domiciliées en Suisse sont possibles, pour autant que les exigences du droit des marchés publics soient respectées.

L'acquisition des ressources propres d'une organisation s'effectue dans le cadre d'un processus d'acquisition ordinaire, ou est déjà achevée.

3.2. Cas particulier : adhésion à une collaboration existante

Cela requiert à la fois une base légale et le respect du droit des marchés publics.

L'approche la plus simple consiste à participer selon le principe « chaque partie supporte ses propres coûts ». Cela peut aussi couvrir les frais généraux, pour autant que des ressources internes ou des ressources externes correctement mises au concours soient utilisées.

Une participation directe à des collaborations étrangères est plus difficile. Un simple rattachement via un Memorandum of Understanding est l'option la plus facile à réaliser.

3.3. Cas particulier : contributions de code à des projets OSS de tiers

Lorsque l'administration fédérale apporte des modifications au code d'un projet OSS existant, il est judicieux de contribuer ces modifications directement en amont au projet d'origine. Ainsi, la maintenance courante ne doit plus être assurée par l'administration fédérale.

3. https://www.eda.admin.ch/deza/fr/home/partnerschaften/auftraege-beitraege/auftraege/anforderungen/rechtlich.html

Selon l'article 9 LMETA, de telles contributions sont même souhaitées. Du point de vue du droit des marchés publics, cela ne pose pas de problème.

Il importe que ce soit le titulaire des droits qui apporte la contribution. Cela signifie que l'administration fédérale en assume la responsabilité pour le compte de ses collaborateurs et mandataires. Concrètement, tout Contributor Licence Agreement du projet doit être signé, ou un Developer Certificate of Origin doit être joint à la pull request.

Un **Developer Certificate of Origin**[4] **(DCO)** pour l'administration fédérale pourrait se lire ainsi :

> Copyright © 2004, 2006 The Linux Foundation and its contributors.
>
> Everyone is permitted to copy and distribute verbatim copies of this licence document, but changing it is not allowed.
>
> Developer's Certificate of Origin 1.1
>
> By making a contribution to this project, we certify that:
>
> (a) The rights to this contribution are owned by the Swiss Confederation and I am authorised to submit it under the open source licence indicated in the file; or
>
> (b) The contribution is based upon previous work that, to the best of my knowledge, is covered under an appropriate open source licence and I have the right under that licence to submit that work with modifications, whether created in whole or in part by me, under the same open source license (unless I am permitted to submit under a different licence), as indicated in the file; or
>
> (c) The contribution was provided directly to me by some other person who certified (a), (b) or (c) and I have not modified it.
>
> (d) I understand and agree that this project and the contribution are public and that a record of the contribution (including all personal information I submit with it, including my sign-off) is maintained indefinitely and may be redistributed consistent with this project or the open source licence(s) involved.
>
> *— Developer Certificate of Origin*
> *Version 1.1*

La structure d'un Contributor Licence Agreement est décrite dans Em002-3 Guide de licences OSS, chapitre 8.9.

Le transfert effectif du code s'effectue conformément aux directives du projet concerné.

4. Concept de communauté

Le cœur d'une communauté réside en fin de compte dans sa structure et sa gouvernance. Celles-ci sont consignées dans un concept de communauté.

4. https://developercertificate.org/

Le concept de communauté à élaborer par la direction de projet ou d'application (ou à confier à des tiers) devrait suivre la structure de chapitres ci-après. Selon la nature de la communauté concrète, tous les sous-chapitres ne sont pas nécessaires. Idéalement, de nombreux chapitres renverront à des documents déjà standardisés ou utiliseront les processus standard des dépôts correspondants. Le concept de communauté peut aussi s'appliquer à des projets existants dirigés par d'autres, en particulier lorsque la gouvernance effective n'est pas encore documentée par écrit.

La formalisation de la communauté rassemble tous les participants et simplifie l'intégration de nouveaux participants.

Projet de structure d'un concept de communauté :

1. Tableau de synthèse (nom, dépôt, adresse de contact, licence, niveau de support, existant/nouveau)
2. Objectifs
    a. de l'application
    b. de la communauté
3. Organisation
    a. Propriétaire du code
    b. Forme d'organisation / organes
    c. Droits de vote
    d. Direction de projet
    e. Questions d'acquisition (le cas échéant)
    f. Autres aspects de gouvernance
4. Feuille de route et processus de modification
5. Processus de développement
    a. Développement ouvert
    b. Droits de committer
    c. Processus de revue
    d. Répartition des tâches et mentions nominatives
    e. Sauvegardes
6. Financement de la communauté
7. Marketing
8. Traitement des contributions externes
    a. Traitement des erreurs signalées
    b. Traitement des demandes
    c. Traitement des pull requests

   d. Traitement des forks

   e. CLA, DCO et attribution des droits / de la propriété intellectuelle

9. Exploitation de l’application

En principe, chaque autorité fédérale est responsable de l’élaboration des concepts de communauté. Le secteur TNI peut publier des modèles et exemples existants et accepte des exemples correspondants en vue de leur publication. Il est également judicieux que les responsables de communauté échangent entre eux.

Une source possible pour certaines parties est également la documentation de l’Eclipse Foundation,[5] étant entendu qu’il convient ici de procéder à une adaptation appropriée dans l’esprit de « ne faire que le nécessaire ».

5. Décisions de principe lors de la constitution d’une communauté

Les principes de la communauté, ou la renonciation à celle-ci, sont déterminés à l’aide de Em002-4.1. Les questions de principe déterminantes figurent dans la boîte morphologique ci-dessous ; chaque dimension est présentée dans les sections qui suivent.

5. Voir https://www.eclipse.org/projects/handbook/#starting

Dimension	(a)	(b)	(c)	(d)	(e)
A — Gestion du produit	Confédération	Organisation commune	Organisation ouverte	Tiers	
B — Intégration des fournisseurs	Mandataire	Partenaire	Indépendant	Direction	
C — Répartition des coûts	Confédération	Chacun pour soi	Principe de causalité	Clé de répartition	Contribution
D — Support	Publication minimale	Support par la Confédération	Support par des tiers		
E — Développement	Interne	Tiers	Ouvert		

Boîte morphologique communauté OSS.

Il est recommandé, lors de la conception de la communauté, de **se concentrer avant tout sur le proche avenir** ; s'il n'existe par exemple pas encore de parties intéressées concrètes pour l'application, il n'est généralement guère judicieux de définir une communauté dotée d'une société simple, de sa propre association et de processus complexes pour l'élaboration de la feuille de route.[6] [7]

5.1. Gestion du produit

Selon le type d'application, son importance stratégique et sa position dans le cycle de vie, il existe différentes variantes pour représenter la gestion du produit. Pour les organisations au sein des autorités fédérales, cela concerne toujours les responsables d'application. La définition de la feuille de route fonctionnelle et technologique ainsi que les processus de publication sont particulièrement pertinents pour le concept de communauté.

Possibilités :

a. **Organisations au sein de la Confédération** : la Confédération veut garder le contrôle ; elle définit seule la feuille de route.

b. **Organisation commune** : avec d'autres utilisateurs ou fournisseurs.

c. **Organisation ouverte** : liens lâches, pas de feuille de route active.

d. **Tiers** : soit le produit existe déjà, soit le travail doit être externalisé.

5.1.1. Organisations au sein de la Confédération :

Si l'application revêt une grande importance stratégique ou si la Confédération dépend de sa capacité à mettre en œuvre rapidement ses modifications, il peut être judicieux de garder le contrôle.

L'autorité fédérale décide seule, par l'intermédiaire de la responsabilité des exigences, de ce qui est intégré dans le logiciel et à quel moment.

Pour les autres utilisateurs potentiels de l'application, cela signifie qu'ils sont pratiquement contraints d'utiliser leur propre version (fork) de l'application.

À défaut, ils n'ont guère la possibilité de mettre en œuvre des adaptations et extensions qui peuvent leur être importantes. Cela ne plaide pas contre une publication en open source : les outils de gestion de code[8] offrent un large soutien pour de tels scénarios, et les adaptations ainsi que les corrections d'erreurs peuvent être reprises de manière sélective dans les deux sens.

Un partage des coûts est typiquement difficile avec ce modèle.

---

6. Avec une association (ou une fondation), une personne morale distincte est créée pour s'occuper du logiciel et du produit. Cela ne se justifie qu'à partir d'un certain niveau d'importance, de complexité et de l'existence de plusieurs partenaires (sur un pied d'égalité). Une feuille de route sert à montrer au public élargi et aux utilisateurs potentiels supplémentaires du logiciel ce qui est prévu.
7. Il peut aussi être possible, en particulier pour des projets impliquant dès le départ plusieurs acteurs étatiques, de s'adresser à des fondations existantes et d'examiner si les projets et la gouvernance peuvent être conduits sous leur égide, à l'instar de https://finos.org ou https://osr.finos.org ou https://www.eclipse.org/collaborations/ ou https://iot.eclipse.org ou https://outreach.eclipse.foundation/open-regulatory-compliance.
8. p. ex. GitHub

5.1.2. Commun (la Confédération dirige ou partenaires sur un pied d’égalité)

Les décisions relatives à la feuille de route et aux priorités devraient être coordonnées et prises conjointement avec les autres membres de la communauté. La Confédération participe en tant que membre de la communauté, mais n’est pas l’organisation pilote.

Pour que la prise de décision commune fonctionne, des structures et des règles doivent être fixées au préalable. Cela définit aussi la manière dont les coûts de mise en œuvre des adaptations décidées en commun sont répartis entre les parties.

Avec ce modèle, une version unique de l’application est créée, puis utilisée par tous les membres. Les coûts totaux sont de ce fait nettement inférieurs par rapport aux autres variantes. La flexibilité de chaque utilisateur est toutefois moindre, et ils doivent coordonner au préalable leurs demandes d’adaptation avec tous les autres. La prise de décision est plus lente et plus laborieuse.

Ce modèle convient le mieux aux applications destinées à être développées conjointement avec d’autres organisations clairement définies (p. ex. les cantons).

5.1.3. Organisation ouverte

Dans ce modèle, les processus et structures ne sont définis que de manière lâche. La Confédération conserve formellement le contrôle, mais est ouverte aux contributions et adaptations.

Aucune feuille de route, ou seulement une feuille de route très sommaire, n’est définie ; le consensus est trouvé de manière informelle et dans des discussions directement sur la plateforme de publication. La communauté elle-même est plutôt dynamique ; des membres peuvent être très actifs un certain temps puis se retirer.

Ce modèle convient aussi aux applications plus petites déjà utilisées en production et « achevées ». C’est également la meilleure forme, en général, pour des applications ou composants à orientation technique qui s’adressent potentiellement à un public plus large ne pouvant pas être défini précisément à l’avance.

5.1.4. Tiers

Dans ce modèle, les processus et structures sont définis par un tiers et l’autorité fédérale est membre de la communauté sans en avoir la direction. Cela peut tenir au fait que le projet a par exemple été défini par un autre État ou canton. Ou il se peut que la Confédération ne soit pas intéressée à poursuivre elle-même le programme et transfère cette tâche.

5.2. Intégration des fournisseurs

Il existe en principe les possibilités suivantes pour intégrer les fournisseurs :

a. **Fournisseur comme mandataire** : il est en principe considéré comme remplaçable à moyen terme. Le fournisseur doit développer l’application de manière économique et de haute qualité sur la base des commandes de la gestion du produit (ou de la communauté), mais n’a au plus qu’une influence consultative sur la feuille de route et la priorisation.

b. **Partenaires à long terme avec codécision** : ils doivent pouvoir codéterminer activement le développement. Cela renforce leur rôle et ils sont potentiellement disposés à investir eux-mêmes dans l’application. Typiquement, dans ce modèle, les fournisseurs distribuent le logiciel et recherchent activement de nouveaux clients.

c. **Indépendant :** les fournisseurs définissent eux-mêmes s’ils veulent travailler sur le projet et de quelle manière. Cela peut se produire par exemple lorsque plusieurs fournisseurs veulent générer leur propre activité sur la base du logiciel. Des licences libérales peuvent potentiellement le

favoriser. Cela peut aussi se produire lorsque la stratégie de la Confédération diverge de celle du fournisseur. Dans ce scénario, il est probable que le fournisseur crée un fork.

d. **Direction** : si le fournisseur reprend la gestion du produit, il en a la direction. Le développement est alors déterminé par lui. En contrepartie, cela peut aussi signifier que l'administration fédérale n'a que très peu de responsabilités à assumer. Avec une publication minimale, il est possible que le fournisseur assume ce rôle de son propre chef.

Si le fournisseur assume un rôle fort, cela présente des avantages pour l'administration fédérale : par une commercialisation active de l'application, il y a une chance que la communauté croisse plus vite et plus fortement, réduisant ainsi plus rapidement les coûts par membre.

Si un modèle avec forte codécision des fournisseurs est choisi, il faut veiller à ce qu'il n'en résulte pas une dérogation aux règles du droit des marchés publics. Il est généralement recommandé d'impliquer au moins deux fournisseurs afin de ne pas créer une dépendance trop forte envers un seul. Il devrait aussi être possible pour d'autres fournisseurs de rejoindre la communauté.

5.3. Répartition des coûts

Les options suivantes sont disponibles pour répartir les coûts de maintenance, de développement ultérieur et de nouvelles fonctionnalités :

a. **Organisations au sein de la Confédération** : en particulier lors de la constitution d'écosystèmes ou si l'autorité fédérale conserve le contrôle total, elle doit aussi en supporter les coûts.

b. **Chacun pour soi** : chacun paie lui-même les modifications qu'il souhaite. Au besoin, il est décidé au cas par cas de financer certaines extensions en commun. Les coûts récurrents sont facturés selon une clé de répartition ou à la demande.

c. **Principe de causalité** : ceux qui ont effectivement besoin des prestations ou fonctionnalités les paient. Des tarifs horaires standardisés peuvent être appliqués.

d. **Clé de répartition** : les coûts sont répartis au sein de la communauté selon une clé définie. Seules les extensions spécifiques y dérogent. Celles-ci devraient être payées directement par les utilisateurs concernés. Le diviseur de coûts peut être, par exemple, la taille du canton concerné ou le nombre d'utilisateurs. Concernant la clé de répartition, voir aussi le chapitre « Répartition des coûts ».

e. **Contribution** : lorsque l'autorité fédérale n'apporte que du personnel à un projet existant.

Fondamentalement, la variante b (partage des coûts) n'a généralement de sens que s'il existe aussi une gestion commune du produit comme dans la variante b (commun). Seuls ceux qui peuvent avoir voix au chapitre seront disposés à supporter les coûts subséquents des décisions.

5.4. Support

Dans le cadre de la répartition des coûts et de la charge, il importe de clarifier qui fournit quel support. Selon l'art. 9 LMETA, une pure publication minimale est possible. Cette variante peut ne pas produire l'effet souhaité et ne constitue pas en soi une communauté dans laquelle l'autorité fédérale a de l'influence.

a. **Publication minimale** : aucun support (c'est-à-dire que la publication a généralement lieu sans support organisé ni participation de l'autorité fédérale).

b. **Support par la Confédération** : l'autorité fédérale (ou son prestataire) fournit un support défini.

c. **Support par des tiers** : l'autorité fédérale ou la communauté définit une organisation de support.

L’effet recherché, p. ex. sur l’écosystème suisse ou un financement initial d’une communauté, peut conduire la Confédération à fournir du support. L’étendue du support doit naturellement être définie. En principe, un émolument peut aussi être perçu. En raison des mécanismes de la Confédération, les prestataires ou fournisseurs sont alors plus à même de proposer un support commercial.

Les offices peuvent proposer un support payant conformément à la LMETA (le cas échéant aussi via des prestataires et des tiers). Pour ce faire, ils doivent publier les émoluments correspondants sur leur site web, la base légale nécessaire étant en place. Le raccordement au système de paiement peut être discuté avec le DFF.

Des exemples de support et d’émoluments figurent à l’annexe G.

5.5. Développement

La méthodologie fondamentale du développement peut aussi influencer le type de communauté.

a. **Interne** : l’utilisation de dépôts internes signifie que des tiers ne peuvent pas travailler directement. Plus la séparation est stricte, plus la Confédération doit fournir d’efforts d’intégration si elle veut reprendre des extensions de tiers. Le processus de publication est également plus complexe.

b. **Tiers** : soit un fournisseur travaille de son côté, soit le produit déjà lancé ou dirigé par un tiers est développé selon des règles définies.

c. **Ouvert** : le développement logiciel a lieu directement dans un dépôt public. La publication selon la LMETA est ainsi largement garantie. Les dépenses interviennent pendant le développement. Dans ce scénario, la collaboration avec des tiers peut aussi intervenir plus tôt (et plus facilement).

6. Contenus détaillés pour le concept de communauté

Ce chapitre donne des indications pour la rédaction du concept de communauté sur la base des décisions de principe prises (selon le chapitre 5).

Gestion du produit / Fournisseur comme / Développement — Proposition

ouvert / Partenaire / Confédération
Contenus ou suggestions, si les décisions de principe sont :
Gestion du produit c) Organisation ouverte
Intégration des fournisseurs : b) Fournisseur comme partenaire
Répartition des coûts : b) Chacun pour soi
Le texte ci-dessus peut être copié dans le modèle et adapté.

Commun / - / Chacun pour soi
Contenus ou suggestions, si les décisions de principe sont :
Gestion du produit b) Développement commun
Intégration des fournisseurs : a) ou b) (non pertinent)
Répartition des coûts : c) Principe de causalité ou d) Partage des coûts

Commun / - / Auteur

Le concept de communauté devrait être un document dynamique, pouvant être adapté en concertation avec la communauté. Le concept devrait refléter la situation actuelle et proposer des options de développement futures. Si d’autres membres rejoignent ultérieurement la communauté, l’organisation et les processus peuvent être précisés et des mécanismes plus complexes introduits. Les sous-chapitres suivants suivent directement la structure proposée du concept de communauté (voir chapitre 4).

Si la gestion du produit incombe à des tiers, ceux-ci définissent les concepts pour la communauté.

6.1. Données clés

Les données clés suivantes devraient être établies et publiées pour chaque communauté, afin que les personnes potentiellement intéressées puissent s'inscrire pour le logiciel et la communauté :

- Nom du logiciel
- Dépôt
- Propriétaire
- Unité d'organisation / office responsable
- Adresse de contact
- Licence utilisée
- Niveau de support
- Projet existant

Ces indications peuvent toutes être représentées avec publiccode.yml.[9]

6.2. Objectifs

L'objectif devrait contenir le but réel ou la « vision » de l'application. Cette vision devrait aussi figurer dans le fichier **README.md** du dépôt (voir liste de contrôle Em002-2.2 Analyse et préparation OSS).

Dans le même temps, il convient de décrire ce que vise la communauté :

- Qui doit adhérer ?
- Qui sont les parties potentiellement intéressées ?
- Faut-il rechercher activement de nouveaux membres ?
- Quels sont les principaux objectifs poursuivis par la publication en open source ?

6.3. Organisation et gouvernance

En principe, pour les nouveaux développements, l'objectif est que la Confédération soit titulaire des droits principaux sur le code source.

Le type de conduite de projet dépend de l'organisation qui avait la direction (SAFe, HERMES) au sein de la Confédération. Pour la forme d'organisation, il convient toujours de viser un minimum de frais généraux (existant plutôt que nouveau ; société simple plutôt qu'association).

Gestion du produit / Fournisseur comme / Développement — Proposition

Confédération / - / Interne
: Comme la Confédération conserve le contrôle direct, aucune structure organisationnelle supplémentaire n'est nécessaire.
L'entité qui publie (p. ex. un office fédéral) reste propriétaire et assume toutes les tâches de coordination. Dans ce cas, le chapitre « Organisation » peut être très bref.

Commun / - / Interne
: Pour ce scénario, l'organisation en **société simple** est généralement recommandée.
Afin de permettre le développement commun et la coordination de la feuille de route et des

9. https://yml.publiccode.tools/

exigences, il est recommandé de créer deux groupes, chacun comptant un participant par membre de la communauté :

- Organe de pilotage : définit la stratégie et la priorisation.
- Groupe d'experts : élabore les exigences et les contenus concrets au niveau technique. Selon l'ampleur, un groupe d'experts distinct peut aussi être institué par thème.

Le propriétaire du code est l'office qui l'a publié.

Si les structures deviennent plus complexes ou si plus de cinq utilisateurs ou membres supplémentaires sont impliqués, il peut valoir la peine d'examiner si la communauté ne serait pas mieux organisée sous forme d'association. Comparer les explications de la variante ci-dessous.

Commun / - / Tiers

L'organisation en **association** est recommandée.
Soit une association est fondée spécialement pour l'application, soit l'application est transférée à une association existante. Une association est recommandée car, à défaut, la répartition des coûts devient complexe et les tâches de coordination représentent vraisemblablement assez de travail pour employer un gérant à temps partiel pour l'association.

Les droits sur le code sont alors transférés à l'association, qui assume la gestion de la communauté et les commandes auprès des fournisseurs.

Commun / Mandataire / Interne

L'organisation en **association** est recommandée. Soit une association est fondée spécialement pour l'application, soit l'application est transférée à une association existante. Une association est recommandée car, à défaut, la répartition des coûts devient complexe et les tâches de coordination représentent vraisemblablement assez de travail pour employer un gérant à temps partiel pour l'association.

Les droits sur le code sont alors transférés à l'association, qui assume la gestion de la communauté et les commandes auprès des fournisseurs.
Ici, les fournisseurs ne sont pas membres de l'association, mais uniquement mandataires. Le gérant ne devrait pas être fourni par un fournisseur.

Commun / Mandataire / Tiers

L'organisation en **association** est recommandée. Soit une association est fondée spécialement pour l'application, soit l'application est transférée à une association existante. Une association est recommandée car, à défaut, la répartition des coûts devient complexe et les tâches de coordination représentent vraisemblablement assez de travail pour employer un gérant à temps partiel pour l'association.
Les droits sur le code sont alors transférés à l'association, qui assume la gestion de la communauté et les commandes auprès des fournisseurs.
Les fournisseurs sont membres de l'association ; le gérant peut aussi être fourni par un fournisseur.

ouvert / - / Interne

Il n'est **pas nécessaire de décrire en détail une organisation**, celle-ci étant délibérément conçue de manière ouverte. Il importe toutefois d'établir qui reçoit et traite les demandes externes (responsabilité). Il faut aussi déterminer si et dans quelle mesure des personnes externes peuvent être associées :

- Aucune intégration directe : les externes ne peuvent soumettre que des suggestions et des contributions (sous forme de pull requests).
- Intégration comme contributeurs : les externes qui apportent des contributions fréquentes et de qualité sont directement intégrés.

- Transfert à des externes possible : si, après la publication, des externes contribuent davantage au développement que l'organisation qui publie, l'application peut aussi leur être transférée.

L'organisation qui publie reste propriétaire. Cette variante correspond à l'organisation des projets open source « classiques » ; voir par exemple https://opensource.guide/leadership-and-governance/ pour plus d'informations.

ouvert / Mandataire / Interne

Il n'est **pas nécessaire de décrire en détail une organisation**, celle-ci étant délibérément conçue de manière ouverte. Il importe toutefois d'établir qui reçoit et traite les demandes externes (responsabilité). Il faut aussi déterminer si et dans quelle mesure des personnes externes peuvent être associées :

- Aucune intégration directe : les externes ne peuvent soumettre que des suggestions et des contributions (sous forme de pull requests).
- Intégration comme contributeurs : les externes qui apportent des contributions fréquentes et de qualité sont directement intégrés.
- Transfert à des externes possible : si, après la publication, des externes contribuent davantage au développement que l'organisation qui publie, l'application peut aussi leur être transférée.

L'organisation qui publie reste propriétaire.
Cette variante correspond à l'organisation des projets open source « classiques » ; voir par exemple https://opensource.guide/leadership-and-governance/ pour plus d'informations.
Le fournisseur ne devrait assumer que des tâches de coordination purement techniques (revue de code, évaluation).

Tiers / Partenaire / Interne

Il n'est **pas nécessaire de décrire en détail une organisation**, celle-ci étant délibérément conçue de manière ouverte. Il importe toutefois d'établir qui reçoit et traite les demandes externes (responsabilité). Il faut aussi déterminer si et dans quelle mesure des personnes externes peuvent être associées :

- Aucune intégration directe : les externes ne peuvent soumettre que des suggestions et des contributions (sous forme de pull requests).
- Intégration comme contributeurs : les externes qui apportent des contributions fréquentes et de qualité sont directement intégrés.
- Transfert à des externes possible : si, après la publication, des externes contribuent davantage au développement que l'organisation qui publie, l'application peut aussi leur être transférée.

L'organisation qui publie reste propriétaire. Cette variante correspond à l'organisation des projets open source « classiques » ; voir par exemple https://opensource.guide/leadership-and-governance/ pour plus d'informations.

S'agissant de la gouvernance, *[BITKOM2023]* chapitre 4.1, *[IZqCab2023]* ou https://github.com/todogroup/ospo-career-path/blob/main/OSPO-101/module7/README.md#governance-models peuvent également être consultés.

Des exemples de statuts d'association figurent à l'annexe E.

Remarque : la publication devant être garantie conformément à l'art. 9 LMETA, l'administration fédérale doit s'assurer que celle-ci ne puisse pas être soudainement retirée (voir les « directives Open Source dans les acquisitions — achat de logiciels selon l'art. 9 LMETA » *[BBL-WL]*, chiffre V.2).

6.4. Feuille de route et processus de modification

Gestion du produit / Fournisseur comme / Développement — Proposition

Confédération / Mandataire / -
L'autorité fédérale définit la feuille de route et décide quelles modifications sont mises en œuvre. Les processus standard de l'office XY sont utilisés à cette fin.

Confédération / Partenaire / -
L'autorité fédérale définit la feuille de route en collaboration avec l'entreprise XY. Elle décide quelles modifications sont mises en œuvre. Les processus standard de l'office XY sont utilisés à cette fin.

Commun / - / Interne
Le processus d'élaboration de la feuille de route et d'approbation des modifications doit être déterminé par les membres de la société ou de l'association. Selon la structure des membres (nombre, rapport de taille, etc.), d'autres processus peuvent être appropriés.
Les processus doivent toutefois comporter des mesures de résolution des conflits et de recherche de majorité.

Commun / - / Tiers
Le processus d'élaboration de la feuille de route et d'approbation des modifications doit être déterminé par les membres de la société ou de l'association. Selon la structure des membres (nombre, rapport de taille, etc.), d'autres processus peuvent être appropriés.

Les processus doivent toutefois comporter des mesures de résolution des conflits et de recherche de majorité.

Avec l'ajout qu'une clé de répartition des coûts doit également être établie. Il convient aussi de réfléchir à la manière de traiter le partage des coûts pour des adaptations non souhaitées par les membres concernés. En particulier dans les associations comptant des membres de tailles inégales, des situations peuvent survenir où un grand membre (p. ex. l'autorité fédérale) supporte une part importante des coûts d'une extension sans en tirer profit.

ouvert / - / -
Exemple, ne convient pas à toutes les situations
Il n'est pas nécessaire d'établir une feuille de route ; l'application répond actuellement aux besoins.

Les modifications sont décidées dans une discussion ouverte et un consensus est recherché. La décision finale appartient au propriétaire du code (Confédération suisse).

6.5. Processus de développement

Si le développement ouvert entre en jeu, il doit être défini ici. La répartition des tâches et la désignation des postes centraux dans le concept de communauté sont importantes. De même que la manière dont la sécurité du contenu du dépôt est assurée.

Gestion du produit / Fournisseur comme / Développement — Proposition

Confédération / - / -
Les droits de committer (accès en écriture au code) appartiennent aux personnes/entités/fournisseurs sélectionnés par la Confédération. À l'expiration du contrat, les droits correspondants sont révoqués.

Les prestataires et fournisseurs sont responsables de la conduite du processus technique de revue de code pour les contributions externes acceptées sur le fond par la Confédération. Ils répondent de la qualité technique. Les fournisseurs sont indemnisés pour ce travail par la Confédération.

Commun / Mandataire / Interne

En principe, toutes les entités mandatées par un membre de la communauté reçoivent un accès en écriture au code (droits de committer). À l'expiration du contrat, les droits correspondants sont révoqués.

Lorsque du code modifié touche des domaines centraux de l'application, toutes les modifications doivent être examinées par une entité spécialement mandatée à cet effet par la communauté. Cette entité répond de la qualité technique de l'application. Elle est aussi chargée d'examiner les modifications acceptées sur le fond par la communauté.

Commun / Partenaire / Tiers

Les droits de committer (accès en écriture au code) appartiennent aux membres de la communauté. Ces personnes s'organisent elles-mêmes et assument conjointement la responsabilité de la qualité du code.

Si un membre de la communauté mandate un fournisseur externe pour une adaptation ou une extension, celle-ci est examinée par un membre de la communauté. Il est indemnisé pour cela par le mandant.

Les modifications du cœur de l'application doivent être examinées par un second membre de la communauté.

Les fournisseurs membres de la communauté sont en outre chargés d'examiner les contributions externes et le nouveau code technique accepté sur le fond par l'association.

ouvert / - / -

Initialement, les droits de committer appartiennent aux développeurs ou à leurs employeurs. Les droits de committer sont accordés à tous les développeurs qui contribuent activement au projet pendant plusieurs mois et font preuve d'une compétence sociale appropriée.

Les développeurs conservent en principe leurs droits de committer, même s'ils changent d'employeur ou si la Confédération choisit un autre fournisseur. Les droits de committer ne s'éteignent que s'ils sont abandonnés volontairement ou si la majorité des autres committers décide de les révoquer.

La Confédération a le droit de désigner comme committers certains développeurs des entreprises qu'elle mandate. Elle n'a pas le droit de retirer sans motif les droits de committer à quelqu'un.

Les revues de code sont effectuées par au moins une personne agissant comme committer. Les committers s'organisent eux-mêmes.

La Confédération s'engage à indemniser le travail de revue dans la mesure où cela lui est financièrement possible.

6.6. Partage des coûts et fournisseurs

Gestion du produit / Fournisseur comme / Développement — Proposition

Confédération / Mandataire / Interne

Les fournisseurs sont sélectionnés périodiquement par appels d'offres OMC ou par une procédure de gré à gré (selon l'ampleur des prestations de maintenance).

Confédération / Partenaire / -

Important : vérifier au préalable si cela est admissible au regard du droit des marchés publics dans le cas concret.

Nous compléterons la documentation sur ce point. La Confédération choisit le fournisseur XXX comme partenaire stratégique et vise une coopération à long terme.

Commun / Mandataire / Interne

Chaque membre de la communauté mandate de manière autonome un ou plusieurs fournisseurs de logiciels pour mettre en œuvre les adaptations et extensions dont il a besoin. Pour les adaptations souhaitées par plusieurs membres, le partage des coûts est convenu au cas par cas. Le membre supportant la plus grande part des coûts est responsable de la sélection et du mandat au fournisseur. (Réglementations relatives à la maintenance logicielle)

Commun / Mandataire / Tiers

L'association sélectionne périodiquement les fournisseurs de logiciels de manière centralisée par appels d'offres OMC ou procédures de gré à gré (selon l'ampleur des prestations de maintenance). Les coûts sont attribués aux membres selon une clé : (à définir : par population, utilisateurs, etc.)

Commun / Partenaire / Interne

Chaque membre de la communauté mandate de manière autonome un ou plusieurs fournisseurs de logiciels pour mettre en œuvre les adaptations et extensions dont il a besoin. Pour les adaptations souhaitées par plusieurs membres, le partage des coûts est convenu au cas par cas. Le membre supportant la plus grande part des coûts est responsable de la sélection et du mandat au fournisseur.

(Réglementations relatives à la maintenance logicielle) Au besoin complété par une contribution des fournisseurs participants. Exemple : *Chaque entreprise de développement logiciel membre de la communauté s'engage à investir chaque année au moins XY jours de travail à ses propres frais dans le développement, la mise à jour technologique ou le marketing de l'application.*

Commun / Partenaire / Tiers

L'association sélectionne périodiquement les fournisseurs de logiciels de manière centralisée par appels d'offres OMC ou procédures de gré à gré (selon l'ampleur des prestations de maintenance). Les coûts sont attribués aux membres selon une clé : (à définir : par population, utilisateurs, etc.) Au besoin complété par l'ajout suivant : *Chaque entreprise de développement logiciel membre de la communauté s'engage à investir chaque année au moins XY jours de travail à ses propres frais dans le développement, la mise à jour technologique ou le marketing de l'application.*

Tiers / Mandataire / -

Les fournisseurs sont sélectionnés périodiquement par appels d'offres OMC ou par une procédure de gré à gré (selon l'ampleur des prestations de maintenance). Avec l'ajout suivant : *Les améliorations apportées par les fournisseurs potentiels et leurs activités dans la communauté sont prises en compte comme facteur dans l'évaluation.*

Les modifications souhaitées par des externes ne sont pas financées par la Confédération, mais doivent être commandées et payées par eux-mêmes.

Tiers / Partenaire / -

Les fournisseurs sont sélectionnés périodiquement par appels d'offres OMC ou par une procédure de gré à gré (selon l'ampleur des prestations de maintenance). Avec l'ajout suivant : *Les améliorations apportées par les fournisseurs potentiels et leurs activités dans la communauté sont prises en compte comme facteur dans l'évaluation. Les modifications souhaitées par des externes ne sont pas financées par la Confédération, mais doivent être commandées et payées par eux-mêmes.*

6.7. Marketing

Gestion du produit / Fournisseur comme / Développement — Proposition

Confédération / Mandataire / -
La Confédération ne commercialise pas activement l'application, mais la publie sur des plateformes de logiciels open source. La publication de l'application est annoncée dans les organes spécialisés. Si des parties intéressées se manifestent, nous examinons s'il convient de constituer une communauté commune ou de poursuivre avec leur version du logiciel.

Commun / Partenaire / -
En complément :
Les fournisseurs peuvent commercialiser l'application et rechercher d'autres parties intéressées. La Confédération peut être citée comme référence. Nous acceptons consciemment que des versions divergentes du logiciel puissent en résulter.

Commun / Mandataire / Interne
L'application est commercialisée par les membres ou par l'association.

Commun / Partenaire / Interne
L'application est commercialisée par les fournisseurs.

Commun / - / Tiers
L'application est commercialisée par l'association, la société simple et leurs membres.

ouvert / - / -
L'application peut être commercialisée par des fournisseurs, ou il peut être renoncé à un marketing explicite.

Alternative sans marketing explicite : la Confédération ne commercialise pas activement l'application, mais la publie sur des plateformes de logiciels open source. Dans les organes spécialisés, nous signalons que nous avons publié l'application et qu'une collaboration est bienvenue. Nous présentons le logiciel aux personnes intéressées et tentons de les intégrer dans nos processus d'élaboration de la feuille de route.

6.8. Traitement des contributions externes

Gestion du produit / Fournisseur comme / Développement — Proposition

Confédération / Mandataire / -
Les contributions et demandes externes doivent aussi être traitées même si la Confédération reste seule décideuse concernant le logiciel (sauf en cas de publication minimale, où un fork est inévitable). Nous proposons l'approche suivante :

Traitement des erreurs signalées
Nous tentons de reproduire les erreurs signalées par des personnes externes et demandons des précisions au besoin. Si la reproduction réussit, la Confédération commande la correction des erreurs. Si l'erreur ne peut pas être vérifiée ou si l'effort de correction est disproportionné, nous présentons nos excuses à la personne ayant signalé l'erreur et lui demandons si elle pourrait soumettre une pull request pour y remédier.

Traitement des demandes
Nous répondons à chaque demande entrante. Si une demande ne peut pas être traitée avec un effort raisonnable et qu'un investissement supplémentaire de temps semble inefficient, nous présentons nos excuses en invoquant des ressources limitées et proposons de mettre la personne en contact avec un fournisseur pour un conseil plus approfondi.

Traitement des pull requests
Nous examinons chaque pull request avec soin. Afin d'éviter un effort inutile du côté du contributeur, nous signalons immédiatement si quelque chose contredit notre feuille de route ou s'il est douteux que nous acceptions la contribution. La Confédération décide de manière autonome et selon des critères techniques quelles contributions elle accepte. Les corrections d'erreurs ayant passé le processus de revue sont toujours acceptées.

Traitement des forks
Nous soutenons les forks de notre application. Celui qui déploie l'application devrait créer son propre fork. Nous examinons les forks qui en résultent au moins une fois par an et reprenons les extensions bonnes et adaptées.

Commun / - / -
Externe désigne les contributions provenant de l'extérieur de la communauté définie (membres de l'association). Différences par rapport à ci-dessus :
Traitement des forks
Nous voulons éviter les forks (durables) de l'application. Les personnes intéressées devraient plutôt rejoindre directement la communauté. Au sein de la communauté, nous tentons de garantir que tous les membres utilisent la version principale.

Ouvert / - / -
La constitution d'une communauté qui fonctionne et d'une culture ouverte nous importe.

Traitement des erreurs signalées
Nous tentons de reproduire et de corriger les erreurs signalées. Nous présupposons la collaboration active des personnes qui signalent. Si possible, elles devraient créer directement une pull request avec une correction d'erreur.

Traitement des demandes
Nous traitons chaque demande dans un délai d'une semaine au maximum. Les demandes de contributeurs actifs sont traitées en priorité et de manière approfondie. Nous tentons d'associer d'autres membres au traitement des demandes.

Traitement des pull requests
Nous souhaitons recevoir autant de pull requests de haute qualité que possible. Nous associons tous les contributeurs actifs aux revues ainsi qu'à l'évaluation technique et métier des pull requests. Pour les contributions controversées, nous tentons de parvenir à une décision par un vote informel.

Traitement des forks
Nous tentons de rendre les forks inutiles par une gestion de communauté active et ouverte. Si des forks apparaissent malgré tout, nous les examinons activement et tentons de reprendre les extensions et améliorations qui y ont été développées. Dans de tels cas, nous sollicitons activement des pull requests.

De manière générale, il convient de relever que les contributions peuvent aller du soutien à d'autres utilisateurs jusqu'à la prise en charge de la responsabilité de parties entières d'un projet (voir *[BITKOM2023]* chapitre 4.2).

6.9. Exploitation de l'application

Gestion du produit / Fournisseur comme / Développement — Proposition

Commun / - / -
Une exploitation centralisée peut être proposée en option par l'association centrale, au sens du Software as a Service (SaaS). Il convient d'examiner si cela est souhaité.

- / Partenaire / -
Il convient de préciser si le fournisseur doit aussi exploiter le logiciel.

- / Direction / -
Il convient d'indiquer que le fournisseur est responsable de l'exploitation.

Les niveaux de support et les contacts de support devraient également être précisés dans ce chapitre.

7. Informations importantes pour les responsables de communauté

7.1. Cela prend du temps

La constitution de communautés est une démarche à planifier sur le long terme. La persévérance et la constance sont deux qualités importantes pour une communauté réussie.

7.2. Autres suggestions pour la constitution d'une communauté

D'autres suggestions pour une bonne gestion de communauté figurent sur https://opensource.guide/code-of-conduct/ et https://opensource.guide/best-practices/.

7.3. Communication

La publication de logiciels et de documentation ouvre un nouveau canal de communication pour l'administration fédérale. Si cela n'est pas fait avec soin et professionnalisme, l'image publique de la Confédération peut en pâtir.

Les communautés OSS de l'administration fédérale se trouvent dans un champ de tension en matière de communication : d'une part s'appliquent les **prescriptions générales de communication**, d'autre part la communauté doit permettre une **collaboration formelle et informelle simple** au-delà des frontières organisationnelles, de préférence sans contrôles qualité ni validations formels.

L'utilisation de dépôts publics et, le cas échéant, de suivis de problèmes et de wikis publics signifie que les propos de chaque collaborateur du projet sont visibles sans filtre par le public. Cela exige des collaborateurs du projet une attitude professionnelle et le respect de standards de qualité pour les publications dans de tels environnements.

Dans les projets OSS, cela est normalement réglé par le **Code of Conduct**. Le code de comportement de l'administration fédérale[10] est très général, mais le principe central est applicable ici : « Les collaborateurs accomplissent leurs tâches avec conscience, intégrité et loyauté. Dans leur vie privée aussi, ils veillent à ne pas porter atteinte à la réputation, à la crédibilité et à l'image de la Confédération. »

7.4. Développement ouvert

Si le développement logiciel est prévu dès le départ dans un dépôt ouvert, la publication se fait automatiquement. Les listes de contrôle déterminantes selon les instructions Em002-2 doivent être remplies au début, et les processus de développement du prestataire ou du fournisseur doivent le permettre. L'avantage est que la communauté peut être constituée dès le tout début. Plusieurs organisations peuvent en outre collaborer au développement.

Les dépôts fermés constituent une forme intermédiaire, mais sont utilisés pour d'autres partenaires dans une collaboration.

7.5. Gestion des utilisateurs

Le projet (ou l'organisation porteuse) détermine si les développeurs et autres collaborateurs du projet travaillent sous leur propre nom ou de manière anonyme. Si les personnes ne travaillent pas anonymement et disposent déjà de comptes sur les plateformes, il est plus simple d'utiliser ceux-ci (professionnels ou privés).

10. https://www.epa.admin.ch/dam/epa/fr/dokumente/aktuell/medienservice/120_verhaltenskodex_f.pdf.download.pdf/

La plateforme devrait en principe permettre l'intégration de tiers, faute de quoi aucun développeur extérieur à la Confédération ne pourra collaborer.

Lors du départ du projet, les droits correspondants sur le projet doivent être supprimés. La gestion du produit en est responsable.

7.6. Fusion de communautés

Dans la mesure du possible, le nombre de communautés devrait rester restreint. Cela signifie :

- S'il existe déjà une communauté où une structure similaire est prévue pour des problèmes similaires impliquant les mêmes personnes, une seule communauté devrait être utilisée.
- Plusieurs projets connexes s'adressant au même public cible peuvent être regroupés en une seule communauté.
- Il ne faut pas créer inutilement des structures de communauté fondamentalement différentes là où il en existe déjà.

7.7. Traitement des contributions de tiers

La contribution de logiciels à d'autres projets est traitée au chapitre 3.3. Les mêmes principes s'appliquent en sens inverse.

Les points suivants sont importants :

- Un Contributor Licence Agreement approprié doit être utilisé afin de garantir que le code source appartienne ensuite à la Confédération (voir aussi Em002-3 Guide de licences OSS, chapitre sur les Contributor Licence Agreements). Celui-ci doit être signé par le propriétaire de la contribution.
- Les confirmations doivent être classées.
- Les contributions doivent être examinées avant leur intégration dans la base de code.
- Les soumissions devraient recevoir une réponse dans un délai raisonnable (issues, requests, pull requests) et être traitées de la manière la plus constructive possible.
- Tous les canaux et arrangements de gouvernance devraient figurer dans le concept de communauté, sur le site web et dans le dépôt.

7.8. Signalements relevant de la sécurité

Outre les signalements d'erreurs usuels, il convient de prévoir — lorsque la nature du projet l'exige — que les vulnérabilités relevant de la sécurité puissent être signalées de manière confidentielle, comme c'est le cas par exemple d'un programme de bug bounty.

Les documents de trustbroker.swiss peuvent servir d'exemple :

- https://github.com/trustbroker-swiss/trustbroker.swiss/blob/main/Security-and-Vulnerability-Disclosure.md

7.9. Éviter les forks

Si les forks sont naturels dans le monde de l'open source, la réintégration du code source, des fonctionnalités et ainsi de suite issus de forks est loin d'être simple. Il peut valoir la peine d'approcher les personnes ou organisations envisageant un fork et de tenter de les maintenir engagées dans le projet principal. Trouver un équilibre entre des intérêts concurrents est souvent nécessaire dans de tels cas.

7.10. Répartition des coûts

Il est judicieux de convenir de clés de répartition générales entre les principaux partenaires. Celles-ci peuvent être adaptées tous les quelques années ou lorsque la situation change fondamentalement. Pour un travail efficace sur le logiciel, les grands partenaires devraient plutôt assumer des parts un peu plus importantes qu'ils ne le devraient effectivement (au besoin aussi la coordination générale). L'organisation doit travailler et non se disputer sur les finances. Si la collaboration porte sur plusieurs projets (par exemple les solutions sectorielles dans le domaine des chemins de fer à voie normale), une clé de répartition uniforme devrait être utilisée pour tous les projets. Du point de vue des autorités fédérales, c'est une amélioration si les coûts ne doivent pas être supportés seuls.

Critères possibles pour les clés de répartition :

- Estimation de la répartition des bénéfices (p. ex. entre une autorité fédérale et un fournisseur espérant d'autres clients)
- Nombre d'habitants/d'utilisateurs (p. ex. entre offices, communes, villes)
- Capacité financière (p. ex. cantons)
- Produit intérieur brut (p. ex. entre cantons)
- Qui veut davantage de contrôle

Pour éviter les discussions, la clé de répartition ne devrait pas être simplement liée au budget annuel. Une clause dans l'esprit du principe de causalité peut permettre des adaptations plus rapides pour certains partenaires s'ils sont disposés à payer un supplément.

La clé de répartition devrait aussi s'appliquer aux frais généraux, à la maintenance/au support et à un minimum de développement ultérieur. Ces parties devraient si possible être résolues sur le long terme.

Il convient de garder à l'esprit que, lorsque des financements de tiers interviennent, ces parties voudront typiquement avoir davantage voix au chapitre. La gestion de la communauté doit être outillée pour y faire face.

7.11. Prestations complémentaires selon l'art. 9, al. 5 et 6, LMETA

Les autorités fédérales peuvent (elles-mêmes, par l'intermédiaire de prestataires ou de fournisseurs) fournir des prestations complémentaires, en particulier pour l'intégration, la maintenance, la garantie de la sécurité de l'information et le support.

La limite est qu'elles doivent servir les tâches des autorités et pouvoir être fournies avec un effort proportionné.

Il en découle que l'autorité fédérale ne peut pas être l'entreprise de développement pour des tiers. Elle corrige des erreurs, peut traiter des aspects relevant de la sécurité, peut fournir un certain support et une aide à l'intégration (toutes activités contribuant à constituer une communauté). Les nouveaux développements et fonctionnalités supplémentaires pour des tiers devraient avant tout s'orienter sur le but de travail normal de l'autorité. Il se peut naturellement que la mise à disposition du logiciel fasse partie de la tâche ; dans ce cas, des développements peuvent être réalisés à tout moment. Il est également précisé que le développement logiciel n'est pas la tâche principale de l'autorité fédérale.

Cela signifie que si des fonctionnalités sont développées pour l'essentiel exclusivement pour des tiers, elles doivent être intégrées en partenariat ou par l'intermédiaire du fournisseur.

Pour éviter des problèmes de droit des marchés publics, il convient, lors de l'acquisition de logiciels et de prestations, de veiller à ce que des fonctionnalités puissent être fournies à des tiers (le cas échéant

uniquement à des tiers étatiques) par des fournisseurs.

La fixation des émoluments a déjà été abordée au chapitre « Support ». La LMETA constitue déjà la base légale de tels émoluments. Ceux-ci doivent être communiqués sur le site web de l'autorité fédérale concernée. Des exemples figurent à l'annexe G.

L'art. 9, al. 6, prévoit qu'il ne faut normalement pas concurrencer le secteur privé. Cela signifie que les émoluments ne devraient pas être fixés trop bas et devraient au moins couvrir les coûts. En cas de doute, il vaut mieux fixer des tarifs horaires trop élevés que trop bas.

De manière générale, il est recommandé de travailler avec un à trois tarifs horaires standard différents.

Si un département fédéral souhaite définir des exceptions à l'al. 6, cela ne doit pas concurrencer le secteur privé. Dans certaines circonstances, le plus simple peut être de fixer les émoluments et de les augmenter en cas d'opposition d'entreprises privées effectivement en mesure d'offrir la prestation.

7.12. Souveraineté numérique

Tant que la Confédération ne dispose pas de son propre dépôt, il est recommandé, pour assurer la souveraineté numérique, de réaliser régulièrement des copies des dépôts publics.

Il convient de noter que ce n'est pas seulement le code source qui est copié, mais aussi d'autres informations telles que le suivi des problèmes, le wiki, les configurations, etc.

Par ailleurs, la gestion des utilisateurs doit être réglée afin d'assurer le contrôle du code source publié. L'objectif général est de garantir un développement indépendant de la plateforme concernée et de permettre à la communauté de changer.[11]

8. Annexe

8.1. Modifications par rapport à la version précédente

- Chapitre 1 « L'essentiel en bref » ajouté
- But de la communauté ajouté au chapitre 3
- Chapitres 3.1 à 3.3 sur la collaboration ajoutés
- Nouveau chapitre 7.7 Traitement des contributions de tiers
- Précisé que les fournisseurs ne devraient pas publier entièrement de leur propre chef (conformément à *[BBL-WL]*, V.2)
- Publicode.yml mentionné
- Harmonisation avec le nouveau Em002-7
- Autres modifications et améliorations rédactionnelles

8.2. Références

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

11. Voir aussi https://www.bfh.ch/de/aktuell/news/2024/neue-studie-digitale-souveraenitaet/

8.3. Abréviations

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

8.4. Exemples de concepts de communauté existants

Dans l'esprit d'une collecte de bonnes pratiques figurent ci-après des concepts de communauté et documents comparables déjà élaborés. Il ne s'agit pas nécessairement de concepts d'autorités fédérales uniquement, mais aussi d'autres organisations.

- GERES Community (http://geres-community.ch)

Classification :

- Gestion du produit a) Développement commun
- Fournisseur plutôt a) Mandataire
- Répartition des coûts : b) Partage des coûts

Remarque : le registre communal GERES n'est pas open source, mais de nombreux aspects de l'organisation peuvent aussi être pertinents pour des applications open source. Documents : statuts (publics) et règlement intérieur (sur demande).

- iGov Portal

Ce chapitre contient actuellement des concepts du canton de Berne et sera complété dès que des exemples fédéraux seront disponibles.

8.5. Exemples de collaborations

Les exemples suivants illustrent comment des collaborations ont été structurées.

8.6. Collaboration : ZenDiS en Allemagne

ZenDiS[12] est une organisation dont le but est de réduire la dépendance dans l'administration publique (principalement en Allemagne) en promouvant les logiciels open source.

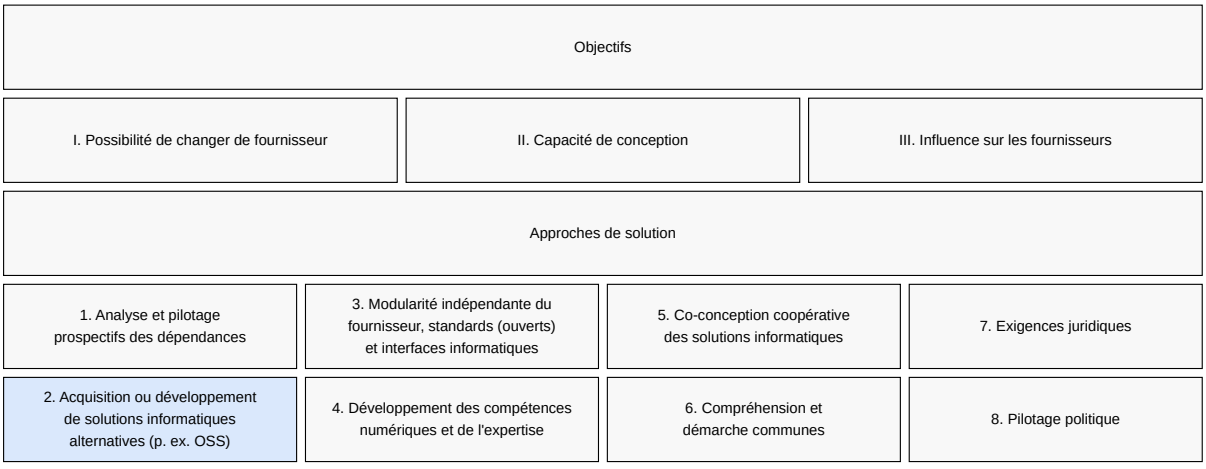


Figure 1 : Objectifs et approches pour renforcer la souveraineté numérique (ZenDiS)

12. Page d'accueil ZenDiS : Co-creating Digital Sovereignty

ZenDiS collabore à cette fin avec des partenaires stratégiques.

Une coopération entre l'administration fédérale et ZenDiS ne devrait donc guère être possible sans restriction, et une coordination ne serait pas contraignante. Cela peut changer avec le temps.

Concrètement, l'administration fédérale devrait mettre en place sa propre organisation pour la Suisse et se coordonner de manière informelle avec ZenDiS. Cela pourrait éventuellement se faire via eOperations, étant précisé qu'eOperations ne devient active que lorsqu'un besoin concret est annoncé et qu'un budget est mis à disposition. Une justification d'une telle organisation parallèle pourrait être trouvée dans le contexte de la cyberadministration ou de la LMETA.

Remarque : ZenDiS est soumise au droit des marchés publics de l'UE et de l'Allemagne. Il existe des différences importantes par rapport à la Suisse. ZenDiS ne peut donc pas être reproduite à l'identique en Suisse, et une coopération avec ZenDiS n'est pas évidente.

8.7. Collaboration : Open Trip Planner (OTP)

Open Trip Planner[13] est une application open source de planification de voyages. Elle est organisée comme un projet de la Software Freedom Conservancy à New York. Certains développeurs sont employés par des entreprises de transport et des fournisseurs. Fait notable, un acteur du secteur public — concrètement ENTUR[14] — a assumé l'un des rôles dirigeants. ENTUR développe avant tout pour ses propres besoins, mais a aussi une orientation coopérative, en particulier pour les pays nordiques. ENTUR est organisée comme une société détenue par le ministère norvégien des transports et des communications. En tant que plus grand acteur, ENTUR absorbe également une part importante des coûts de coordination. Chacun peut soumettre des demandes de fonctionnalités ; celles-ci sont discutées lors de réunions de coordination et développées si quelqu'un les finance. Le financement est assuré principalement par fonctionnalité, par la ou les parties intéressées. Pour l'essentiel, chaque partie supporte ses propres coûts. Les prestations de support devraient en outre faire l'objet d'un appel d'offres distinct en Suisse.

8.8. Nouvelle collaboration initiée par la Confédération : Swiss Trust Broker

Le Trust Broker[15] est un logiciel de la Confédération pour eIAM, disponible en open source. Le développement est assuré par la Confédération. Plusieurs cantons ont décidé de le déployer soit directement en SaaS, soit avec leurs propres instances. Comme il est disponible sous licence AGPL, les extensions réalisées profitent à leur tour à la Confédération. Compte tenu de son caractère critique pour la sécurité, le développement pour l'administration fédérale est assuré exclusivement par la Confédération. D'autres organisations peuvent soumettre des exigences, qui seront mises en œuvre si l'équipe de projet les juge appropriées.

8.9. Nouvelle collaboration initiée par les cantons : inosca

Via inosca,[16] plusieurs cantons ont constitué une communauté de développement axée sur les autorisations électroniques, née du projet eBau.[17] eBau comprend aussi Caluma,[18] un framework d'édition collaborative — un exemple de composant extrait d'une solution plus large et développé en un framework autonome et largement applicable. Celui-ci peut aussi être utilisé dans d'autres projets, même lorsque certaines parties sont soumises à des conditions d'exception, permettant ainsi au plus grand nombre d'autres entités de profiter du code.

13. OpenTripPlanner
14. https://entur.no/
15. GitHub - trustbroker-swiss/trustbroker.swiss: Documentation of the trustbroker.swiss service
16. https://inosca.ch/
17. https://github.com/inosca/ebau
18. https://github.com/projectcaluma

La communauté se réunit au moins une fois par mois (en option deux fois) pour échanger sur des sujets techniques et planifier la feuille de route commune. Tous les participants peuvent soumettre des sujets. Lorsqu'un sujet est jugé pertinent pour la communauté, les cantons intéressés se manifestent et s'organisent en groupe de travail. Les résultats sont restitués en continu à la communauté. inosca fonctionne selon un principe fondamental : qui conçoit, finance. Autrement dit, tous les cantons participant à un groupe de travail se partagent les coûts de la nouvelle fonctionnalité. La fonctionnalité est mise au concours et les coûts sont répartis au sein de la communauté selon une formule de partage des coûts prédéfinie. Dès qu'une fonctionnalité est achevée et déployée, elle devient immédiatement disponible gratuitement pour tous les autres cantons.

La communauté a en outre décidé de réserver un modeste montant annuel fixe pour couvrir les frais administratifs liés au fonctionnement de la communauté. Ce budget est attribué chaque année aux parties assumant un rôle organisationnel.

8.10. SNOWPACK du WSL — contributions directes de développeurs

SNOWPACK[19] est un projet open source développé par le WSL pour modéliser la formation du manteau neigeux. Il s'agit d'un modèle hautement spécialisé, et la communauté a en conséquence été intégrée dans la communauté spécialisée existante. Certains problèmes ont déjà été identifiés comme contributions potentielles. Le modèle de travail est ici que chaque partie supporte ses propres coûts et apporte des fonctionnalités. Les directives relatives au style de codage et au processus de publication sont également documentées dans ce contexte.

8.11. Inspiration pour des statuts d'association

Les statuts devraient être aussi simples que possible.

Les statuts suivants peuvent servir d'inspiration.

- eCH : https://www.ech.ch/sites/default/files/page/STAT_d_DEF_2014-04-10_ech-Statuten.pdf
- Stop Piracy : https://www.stop-piracy.ch/wp-content/uploads/2022/01/Statuten_d_10_09_2021.pdf

D'autres exemples de statuts seront ajoutés dès qu'ils seront disponibles.

8.12. Exemples de support et d'émoluments

- Émoluments pour les prestations statistiques : https://www.fedlex.admin.ch/eli/cc/2003/326/fr
- Émoluments du PFPDT : https://www.edoeb.admin.ch/edoeb/fr/home/datenschutz/grundlagen/dsfa.html
- Prestations gratuites de l'IPI pour des cours dans le domaine de la propriété intellectuelle (pas un logiciel, mais à titre d'exemple) : https://www.ige.ch/fr/prestations/ip-academy/informations-generales/prix

Des exemples seront ajoutés dès qu'ils seront disponibles.

19. https://snow-models.gitlab-pages.wsl.ch/snowpack-web/