Le logiciel libre est-il un Commun ?

La notion de commun semble recouvrir aujourd'hui un (trop) large éventail de significations, ce qui sans doute rend confus son usage. Cet article vous propose d'examiner à quelles conditions on peut considérer les logiciels libres comme des communs.

Nous vous proposons aujourd'hui la republication d'un article bien documenté qui a pu vous échapper au moment de sa publication en juin dernier et qui analyse les diverses dimensions de la notion de Communs lorsqu'on l'associe aux logiciels libres. Nous remercions Emmanuelle Helly pour la qualité de son travail : outre le nombre important de liens vers des ressources théoriques et des exemples concrets, son texte a le mérite de montrer que les nuances sont nombreuses et notamment que la notion de gouvernance communautaire est aussi indispensable que les 4 libertés que nous nous plaisons à réciter…

Qu'est-ce qu'un « Commun » ?

Par **Emmanuelle Helly**

Article publié initialement le 19/06/2017 sur [cette page du site de Makina Corpus]
Contributeurs : Merci à Enguerran Colson, Éric Bréhault, Bastien Guerry, Lionel Maurel et draft pour la relecture et les suggestions d'amélioration. Licence CC-By-SA-3.0

On parle de commun dans le cas d'un **système** qui se veut le plus ouvert possible avec au centre une ou plusieurs **ressources partagées**, gérées collectivement par une

communauté, celle-ci établit des **règles** et une **gouvernance** dans le but de préserver et pérenniser cette ressource.

Cette notion de *res communis* existe en réalité depuis les Romains, et a perduré en occident durant le Moyen Âge, avec par exemple la gestion commune des forêts, et dans le reste du monde avant sa colonisation par les Européens. Mais depuis la fin du 18e siècle dans nos sociétés occidentales, la révolution industrielle et avec elle la diffusion du mode de production capitaliste dans toutes les couches de la société ont imposé une dichotomie dans la notion de propriété : un bien appartient **soit à l'État, soit au privé**.

La théorie de la tragédie des communs par Garett Hardin en 1968, selon laquelle les humains sont incapables de gérer une ressource collectivement sans la détruire, contribue à mettre le concept des communs en sommeil pendant un moment dans nos sociétés occidentales.

En réponse Elinor Ostrom, prix Nobel d'économie en 2009, explique que les ressources qui sont mentionnées dans la «tragédie des communs» ne sont pas des biens communs, mais des biens non gérés. Grâce aux travaux qu'elle mène par la suite avec Vincent Ostrom, **les communs redeviennent une voie concrète pour gérer collectivement une ressource**, entre l'État et le privé. En France et ailleurs on met en avant ces pratiques de gestion de communs, les recherches universitaires en sciences humaines et sociales sur le sujet sont en augmentation, et même la ville de Gand en Belgique fait appel à un économiste spécialiste pour recenser les communs de son

territoire.

On peut distinguer en simplifiant trois types de ressources pouvant être gérées en commun :

- les **ressources naturelles** (une réserve de pêche ou une forêt) ;
- les **ressources matérielles** (un hackerspace ou un fablab) ;
- les **ressources immatérielles** (l'encyclopédie Wikipédia, ou encore OpenStreetMap).

Le logiciel libre, et plus largement les projets libres tels que Wikipédia, est une ressource immatérielle, mais peut-on dire que c'est un commun au sens défini plus haut ? Qu'en est-il de sa gestion, de la communauté qui le maintient et l'utilise, des règles que cette communauté a élaborées pour le développer et partager ?

Outre le fait que la ressource est partagée d'une manière ou d'une autre, les règles appliquées doivent être décidées par l'ensemble de la communauté gérant le logiciel, et l'un des objectifs de la communauté est la pérennité de cette ressource.

Une ressource partagée

Richard Stallman

Un logiciel libre est basé sur quatre libertés fondamentales,

qui sont :

- la liberté de **l'utiliser**,
- de **l'étudier**,
- de **le copier**,
- de **redistribuer des versions modifiées** de ce logiciel.

Ces règles constituent la base des licences libres, elles ont été établies par Richard Stallmann en 1983, et la Free Software Foundation est l'organisation qui promeut et défend ces libertés. D'autres licences sont basées sur ces 4 libertés comme les licences Creative Commons, ou encore l'Open Database Licence (ODbL), qui peuvent s'appliquer aux ressources immatérielles autres que les logiciels.

On peut évoquer le partage sous la même licence (partage à l'identique, ou **copyleft**) qui garantit que le logiciel modifié sera sous la même licence.

Un logiciel peut être placé sous une telle licence dès le début de sa conception, comme le CMS Drupal, ou bien être « libéré » par les membres de sa communauté, comme l'a été le logiciel de création 3D Blender.

Si un changement dans les règles de partage intervient, il peut être décidé par les membres de la communauté, développeurs, voire tous les contributeurs. Par exemple avant qu'OpenStreetMap ne décide d'utiliser la licence ODbL, la discussion a pris environ quatre ans, puis chaque contributeur était sollicité pour entériner cette décision en validant les nouvelles conditions d'utilisation et de contribution.

Mais ça n'est pas systématique, on peut évoquer pour certains projets libres la personnalisation de leur créateur de certains projets libres, « dictateur bienveillant » comme Guido Van Rossum, créateur et leader du projet python ou « monarque constitutionnel » comme Jimy Wales fondateur de Wikipédia. Même si dans les faits les contributeurs ont leur mot à dire il arrive souvent que des tensions au sein de la

communauté ou des luttes de pouvoir mettent en péril la pérennité du projet. Les questions de communauté et de gouvernance se posent donc assez rapidement.

La gouvernance du logiciel libre

Les quatre libertés décrivent la façon dont le logiciel est partagé, et garantissent que la ressource est disponible pour les utilisateurs. Mais si les choix inhérents à son développement tels que la *roadmap* (feuille de route), les conventions de codage, les choix technologiques ne sont le fait que d'une personne ou une société, ce logiciel peut-il être considéré comme un commun ? Qu'est-ce qui peut permettre à une communauté d'utilisatrices / contributeurs d'influer sur ces choix ?

Dans La Cathédrale et le Bazar, Eric S. Raymond décrit le modèle de gouvernance du *bazar*, tendant vers l'auto-gestion, en opposition avec celui de la *cathédrale*, plus hiérarchisée.

« Intérieur cathédrale d'Albi », photo Nicolas Lefebvre, CC-By 2.0, « Sunday Bazar » photo Zainub Razvi, CC-By-SA 2.0

À travers le développement de Fetchmail qu'il a géré pendant un moment, il met en avant quelques bonnes pratiques faisant d'un logiciel libre un bon logiciel.

Un certain nombre d'entre elles sont très orientées vers la communauté :

> *6. Traiter ses utilisateurs en tant que co-développeurs est le chemin le moins semé d'embûches vers une amélioration rapide du code et un débogage efficace. » ;*

ou encore

> *10. Si vous traitez vos bêta-testeurs comme ce que vous avez de plus cher au monde, ils réagiront en devenant effectivement ce que vous avez de plus cher au monde. » ;*

et

> *11. Il est presque aussi important de savoir reconnaître les bonnes idées de vos utilisateurs que d'avoir de bonnes idées vous-même. C'est même préférable, parfois. ».*

Du point de vue d'Eric Raymond, le succès rencontré par Fetchmail tient notamment au fait qu'il a su **prendre soin des membres de la communauté** autant que du code du logiciel lui-même.

D'autres projets libres sont remarquables du point de vue de l'attention apportée à la communauté, et de l'importance de mettre en place une gouvernance permettant d'attirer et d'impliquer les contributeurs.

Le projet Debian fondé en 1993 en est un des exemples les plus étudiés par des sociologues tels que Gabriela Coleman ou Nicolas Auray. Décédé en 2015, son créateur Ian Murdock a amené une **culture de la réciprocité** et un cadre permettant une **large contribution tout en conservant un haut niveau de qualité**, comme le rappelle Gabriela Coleman dans ce billet hommage.

A contrario, il existe des exemples de logiciels dont le code

source est bien ouvert, mais dans lesquels subsistent plusieurs freins à la contribution et à l'implication dans les décisions : difficile de dire s'ils sont vraiment gérés comme des communs.

Bien souvent, si un trop grand nombre de demandes provenant de contributeurs n'ont pas été satisfaites, si de nouvelles règles plus contraignantes sont imposées aux développeurs, un *fork* est créé, une partie des contributeurs rejoignent la communauté nouvellement créée autour du clone. Cela a été le cas pour LibreOffice peu après le rachat de Sun par Oracle, ou la création de Nextcloud par le fondateur de Owncloud en 2016, qui a mené à la fermeture de la filiale américaine de Owncloud.

Du logiciel libre au commun

Il est donc difficile de dire qu'un logiciel libre est systématiquement un commun à part entière. La licence régissant sa diffusion et sa réutilisation ne suffit pas, il faut également que **le mode de gouvernance implique l'ensemble de la communauté** contribuant à ce logiciel dans les décisions à son sujet.

Elinor Ostrom

Les 8 principes de gouvernance des communs relevés par Elinor Ostrom dans le cas de succès de gestion de biens non exclusifs, mais rivaux, sont mobilisables pour d'autres mode

d’auto-organisation, notamment la gouvernance des logiciels libres. Elinor Ostrom a d’ailleurs coécrit un livre avec Charlotte Hess sur les communs de la connaissance.

Les deux premiers principes sont assez liés :

> *1/ délimitation claire de l’objet de la communauté et de ses membres*
>
> *2/ cohérence entre les règles relatives à la ressource commune et la nature de cette ressource*

Nous l’avons vu la définition de « communauté » pose question : celle qui réunit les développeurs, les contributeurs ou la communauté élargie aux utilisateurs ? Existe-t’il une différence entre les contributeurs qui sont rémunérés et les bénévoles ? Les règles varient-elles selon la taille des logiciels ou projets libres ?

De même sur la « ressource » plusieurs questions peuvent se poser : qu’est-ce qui est possible de contribuer, et par qui ? Les règles de contribution peuvent être différentes pour le cœur d’un logiciel et pour ses modules complémentaires, c’est le cas bien souvent pour les CMS (Drupal, Plone et WordPress ne font pas exception).

La question prend son importance dans les processus de décision, décrits par le troisième principe :

> *3/ un système permettant aux individus de participer à la définition et à la modification des règles*

State of the Map 2013, Photo Chris Flemming – CC-By

Dans certains cas en effet, seule une portion des membres les plus actifs de la communauté seront consultés pour la mise à jour des règles, et pour d'autres tous les membres seront concernés : c'est un équilibre difficile à trouver.

On peut aussi mettre dans la balance la ou les entités parties prenantes dans le développement du logiciel libre : si c'est une seule entreprise derrière la fondation, elle peut du jour au lendemain faire des choix contraires aux souhaits des utilisateurs, comme dans le cas de Owncloud.

> *4/ des moyens de supervision du respect de ces règles par des membres de la communauté*
>
> *5/ un système gradué de sanction pour des appropriations de ressources qui violent les règles de la communauté*

On peut citer pour l'application de ces deux principes le fonctionnement de modération a posteriori des pages Wikipédia, des moyens de monitoring sont en place pour suivre les modifications, et certains membres ont le pouvoir de figer des pages en cas de guerre d'édition par exemple.

> *6/ un système peu coûteux de résolution des conflits*

Gnous en duel – Photo Yathin S Krishnappa – CC-By – from wikimedia commons

C'est peut-être l'un des principes les plus complexes à mettre en place pour des communautés dispersées dans le monde. Un forum ou une liste de discussion regroupant la communauté est très rapide d'accès, mais ne sont pas toujours pertinents pour la résolution de conflits. Parfois des moyens de communication plus directs sont nécessaires, il serait intéressant d'étudier les moyens mis en place par les communautés.

Par exemple, un « code de conduite » ne paraît pas suffisant pour la résolution de conflits, comme dans le cas de Sarah Sharp qui quitte la communauté des développeurs du noyau Linux en 2014. Elle a d'ailleurs écrit ensuite ce qui selon elle constitue une bonne communauté.

> *7/ la reconnaissance par les institutions extérieures de cette auto-organisation*

Les 4 libertés d'un logiciel libre sont reconnues dans le droit français, c'est un point important. On peut également relever les diverses formes juridiques prises par les structures qui gèrent le développement de projets libres. Les projets soutenus par la Free Software Foundation (FSF) ont la possibilité de s'appuyer sur l'assistance juridique de la Fondation.

Au-delà de la reconnaissance juridique, le choix d'un gouvernement d'utiliser ou non des logiciels libres peut également influer sur la pérennité de celui-ci. Le manque de soutien dont a souffert Ryxéo pour diffuser Abuledu, la solution libre pour les écoles, a certainement joué en sa défaveur, et plus récemment Edunathon, collectif dénonçant les accords hors marchés publics entre l'État et Microsoft a dû payer une amende pour avoir porté plainte contre l'État.

> *8/ Dans le cas de ressources communes étendues, une organisation à plusieurs niveaux, avec pour base les ressources communes au niveau local.*

Rares sont les projets libres qui sont exclusivement développés localement, ce dernier principe est donc important puisque les communautés sont dispersées géographiquement. Des grands projets tels que Debian ou Wikimedia se dotent d'instances plus locales au niveau des pays ou de zones plus restreintes, permettant aux membres de se rencontrer plus facilement.

Au niveau fonctionnel il peut également se créer des groupes spécialisés dans un domaine particulier comme la traduction ou la documentation, ou encore une délégation peut être mise en œuvre.

Conclusion

On se rend bien compte que les 4 libertés, bien que suffisantes pour assurer le partage et la réutilisation d'un logiciel libre, ne garantissent pas que la communauté des contributrices ou d'utilisateurs soit partie prenante de la gouvernance.

Les bonnes pratiques décrites par Eric S. Raymond sont une bonne approche pour assurer le succès d'un projet libre, de même que les principes de gestion de ressources communes

d'Elinor Ostrom. Il serait intéressant d'étudier plus précisément les communautés gérant des logiciels libres au prisme de ces deux approches.

> *« Le système des paquets n'a pas été conçu pour gérer les logiciels mais pour faciliter la collaboration » — Ian Murdock (1973-2015)*

Pour creuser la question des communs, n'hésitez pas à visiter le site les communs et le blog Les Communs d'Abord, ainsi que la communauthèque qui recense de nombreux livres et articles livres sur le sujet. En anglais vous trouverez beaucoup d'information sur le site de la P2P Foundation initié par l'économiste Michel Bauwens.

Des routes et des ponts (18) — À la croisée des chemins

Le 12 septembre dernier, Framalang commençait la traduction de l'ouvrage de Nadia Eghbal Des routes et des ponts. Aujourd'hui, nous vous proposons le dernier chapitre de ce livre.

Ce chapitre s'interroge sur la marche à suivre pour continuer les avancées technologiques et sociales des cultures open source *et libres. Cette conclusion rappelle qu'à l'heure de l'information, tout le monde est concerné par les technologies ouvertes, bien que nous n'en ayons souvent que peu conscience. Ainsi, afin de pouvoir continuer d'utiliser cette infrastructure qui a été mise à notre disposition, nous devons*

nous mobiliser pour en garantir la pérennité.

À la croisée des chemins

par **Nadia Eghbal**

Traduction Framalang : goofy, MO, Luc, Lumibd, Rozmador, serici, Bromind, lyn., Bam92

L'état actuel de notre infrastructure numérique est un des problèmes les moins bien compris de notre temps. Il est vital de le comprendre.

En s'investissant bénévolement dans notre structure sous-jacente, les développeurs ont facilité la construction de logiciels pour autrui. En la fournissant gratuitement plutôt qu'en la facturant, ils ont alimenté une révolution de l'information.

Les développeurs n'ont pas fait cela pour être altruistes. Ils l'ont fait car c'était la meilleure manière de résoudre leurs propres problèmes. L'histoire du logiciel *open source* est l'un des grands triomphes de nos jours pour le bien public.

Nous avons de la chance que les développeurs aient limité les coûts cachés de ces investissements. Mais leurs investissements initiaux ne nous ont amenés que là où nous sommes aujourd'hui.

Nous ne sommes qu'au commencement de l'histoire de la transformation de l'humanité par le logiciel. Marc Andreessen, co-fondateur de Netscape et reconnu comme capital-risqueur derrière la société Andreessen Horowitz, remarque en 2011 que *«le logiciel dévore le monde» (source)*. Depuis lors, cette pensée est devenue un mantra pour l'ère moderne.

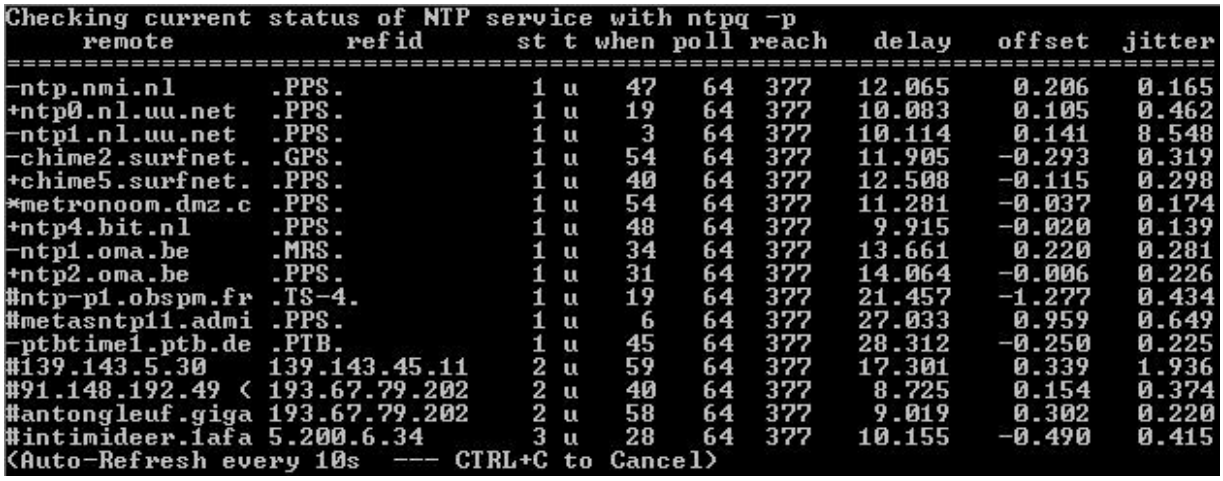
Le logiciel touche tout ce que l'on fait : non seulement les frivolités et les loisirs, mais aussi les choses obligatoires et critiques. OpenSSL, le projet décrit au début de cet essai,

le démontre bien. Dans une interview téléphonique, Steve Marquess explique qu'OpenSSL n'était pas utilisé seulement par les utilisateurs de sites web, mais aussi par les gouvernements, les drones, les satellites et tous *«les gadgets que vous entendez bipper dans les hôpitaux» [Entretien téléphonique avec Steve Marquess, NdA.]*.

Le Network Time Protocol *[protocole de temps par le réseau, NdT]*, maintenu par Harlan Stenn, synchronise les horloges utilisées par des milliards de périphériques connectés et touche tout ce qui contient un horodatage. Pas seulement les applications de conversations ou les courriels, mais aussi les marchés financiers, les enregistrements médicaux et la production de produits chimiques.

Et pourtant, Harlan note:

> *Il y a un besoin de soutenir l'infrastructure publique libre. Mais il n'y a pas de source de revenu disponible à l'heure actuelle. Les gens se plaignent lorsque leurs horloges sont décalées d'une seconde. Ils disent, «oui nous avons besoin de vous, mais nous ne pouvons pas vous donner de l'argent». (source)*



Durant ces cinq dernières années, l'infrastructure *open source* est devenue une couche essentielle de notre tissu social. Mais

tout comme les *startups* ou la technologie elle-même, ce qui a fonctionné pour les 30 premières années de l'histoire de l'*open source* n'aidera plus à avancer. Pour maintenir notre rythme de progression, nous devons réinvestir dans les outils qui nous aident à construire des projets plus importants et de meilleure qualité.

Trouver un moyen de soutenir l'infrastructure numérique peut sembler intimidant, mais il y a de multiples raisons de voir le chemin à parcourir comme une opportunité.

Premièrement, l'infrastructure est déjà là, avec une valeur clairement démontrée. Ce rapport ne propose pas d'investir dans une idée sans plus-value. L'énorme contribution sociale de l'infrastructure numérique actuelle ne peut être ignorée ni mise de côté, comme cela est déjà arrivé dans des débats tout aussi importants sur les données, la vie privée, la neutralité du net, ou l'opposition entre investissement privé et investissement public. Il est dès lors plus facile de faire basculer les débats vers les solutions.

Deuxièmement, il existe déjà des communautés *open source* engagées et prospères avec lesquelles travailler. De nombreux développeurs s'identifient par le langage de programmation qu'ils utilisent (tels que Python ou JavaScript), la fonction qu'ils apportent (telles qu'analyste ou développeur opérationnels), ou un projet important (tels que Node.js ou Rails). Ce sont des communautés fortes, visibles, et enthousiastes.

Les constructeurs de notre infrastructure numérique sont connectés les uns aux autres, attentifs à leurs besoins, et techniquement talentueux. Ils ont déjà construit notre ville ; nous avons seulement besoin d'aider à maintenir les lumières allumées, de telle sorte qu'ils puissent continuer à faire ce qu'ils font de mieux.

Les infrastructures, qu'elles soient physiques ou numériques,

ne sont pas faciles à comprendre, et leurs effets ne sont pas toujours visibles, mais cela doit nous encourager à les suivre plus en détail, pas moins. Quand une communauté a parlé si ouvertement et si souvent de ses besoins, tout ce que nous devons faire est d'écouter.

Remerciements

Merci à tous ceux qui ont courageusement accepté d'être mentionnés dans cet ouvrage, ainsi qu'à ceux dont les réponses honnêtes et réfléchies m'ont aidée à affiner ma pensée pendant la phase de recherche :
André Arko, Brian Behlendorf, Adam Benayoun, Juan Benet, Cory Benfield, Kris Borchers, John Edgar, Maciej Fijalkowski, Karl Fogel, Brian Ford, Sue Graves, Eric Holscher, Brandon Keepers, Russell Keith-Magee, Kyle Kemp, Jan Lehnardt, Jessica Lord, Steve Marquess, Karissa McKelvey, Heather Meeker, Philip Neustrom, Max Ogden, Arash Payan, Stormy Peters, Andrey Petrov, Peter Rabbitson, Mikeal Rogers, Hynek Schlawack, Boaz Sender, Quinn Slack, Chris Soghoian, Charlotte Spencer, Harlan Stenn, Diane Tate, Max Veytsman, Christopher Allan Webber, Chad Whitacre, Meredith Whittaker, Doug Wilson.

Merci à tous ceux qui ont écrit quelque chose de public qui a été référencé dans cet essai. C'était une partie importante de la recherche, et je remercie ceux dont les idées sont publiques pour que d'autres s'en inspirent.

Merci à Franz Nicolay pour la relecture et Brave UX pour le design de ce rapport.

Enfin, un très grand merci à Jenny Toomey et Michael Brennan pour m'avoir aidée à conduire ce projet avec patience et enthousiasme, à Lori McGlinchey et Freedman Consulting pour leur retours et à Ethan Zuckerman pour que la magie opère.

Framasoft remercie chaleureusement les 40 traducteurs et

traductrices du groupe **Framalang** qui depuis septembre ont contribué à vous proposer cet ouvrage (qui sera disponible en **Framabook…** quand il sera prêt) :

Adélie, AFS, alien spoon, Anthony, Asta (Gatien Bovyn), astraia_spica, Bam92 (Abel Mbula), Bidouille, Bromind (Martin Vassor), Ced, dominix, Edgar Lori, flo, glissière de sécurité, goofy, goudron, Julien / Sphinx, jums, Laure, Luc, Lumibd, lyn, Mika, MO, Opsylac (Diane Ranville), pasquin, Penguin, peupleLà, Piup, roptat, Rozmador, salade, serici, teromene, Théo, urlgaga, woof, xi (Juliette Tibayrenc), xXx