

LOGICIELS LIBRES...MAIS LIBRES DE QUOI ?

Communiqué

Été 2025

Tous les jours on utilise, sans plus s'en émouvoir, d'innombrables logiciels dont on se passe très bien de comprendre le fonctionnement. C'est bienheureux car c'est là le signe que l'informatique s'est démocratisée : chacun a désormais la possibilité, à travers les interfaces *user-friendly*, d'effectuer sur sa machine des tâches auparavant réservées à la minorité de connaisseurs capables de les programmer. Par exemple quiconque lisant cet article qui voudra vérifier mes dires n'aura qu'à ouvrir son navigateur, taper « logiciel libre » dans son moteur de recherche et aura ses réponses sans avoir eu à s'intéresser plus que cela aux *web crawlers* ou aux algorithmes de *page ranking*.

Mais au-delà de leur fonctionnement-même, il demeure un aspect des logiciels qui reste trop souvent méconnu. Celui-ci ne relève pas d'aspects techniques mais de préoccupation citoyenne. Il s'agit du cadre légal dans lequel s'inscrivent les logiciels. Les questions comme : « Quel est le prix de ce logiciel ? », « Quelle est son efficacité ? », « À quel point est-il performant ? » effacent bien souvent celles comme : « À qui appartient ce logiciel ? », « Comment a-t-il été développé ? », « Qu'implique mon utilisation de ce logiciel vis-à-vis de la communauté du Web ? ». Dans le cas de notre lecteur·ice consciencieux·euse, s'est-iel posé la question du navigateur qu'il allait utiliser, et du moteur de recherche ? Cette question peut paraître insignifiante mais nous allons voir qu'elle implique de fait une prise de position sur la façon dont les connaissances, les compétences et la propriété s'articulent dans le monde numérique.

Alors qu'est-ce qui fait la différence entre *Firefox* et *Microsoft Edge* ? Entre *Bluesky*, *Mastodon* et *Twitter/X* ?

Il était une fois le logiciel libre

Juridiquement parlant les uns sont propriétaires - au sens où leur code et leur distribution relève de la décision exclusive d'un propriétaire - là où les autres ne le sont pas : certains d'entre eux sont libres. Mais un logiciel libre ne se définit pas seulement par l'absence de propriétaire. Il doit respecter des contraintes précises qui peuvent se résumer en quatre point-clé :

- o. Liberté d'exécuter le programme dans n'importe quels desseins.
- 1. Liberté de modifier le programme à la guise de l'utilisateur¹.
- 2. Liberté de redistribuer des copies (gratuitement ou non).
- 3. Liberté de diffuser des versions modifiées du programme.

On l'aura compris : le logiciel libre va de pair avec « liberté ». La terminologie anglophone *free software* peut entraîner une confusion avec la gratuité qui, elle, n'est pas forcément de mise et, surtout, serait très réductrice. Cette déclaration des droits a été proclamée par la *Free Software Foundation* (FSF), la fondation qui soutient le logiciel libre et garantit son cadre juridique au moyen de licences spéciales. Son histoire est intrinsèquement liée à l'émergence du logiciel libre. Celle-ci commence en 1983, date à laquelle Richard Stallman, alors programmeur au MIT, refuse de se plier aux exigences de confidentialité et d'exclusivité industrielle de l'entreprise *Symbolics*. Cette position serait en effet inconciliable avec sa vision de l'informatique qui doit allier partage et diffusion des connaissances. À cette époque, le matériel informatique commence déjà à se privatiser. Le système Unix notamment connaît un succès grandissant. Face à cette tendance à la privatisation des logiciels, Stallman

¹Cette règle implique l'accès au code source.

décide de s'engager dans la conception de matériels qui échapperaient à la logique environnante exclusive promue à des fins commerciales. Il lance ainsi le projet GNU, GNU pour GNU is Not Unix², un immense projet destiné à doter le monde informatique des logiciels phares et notamment à écrire un OS libre³. D'abord seul dans son immense entreprise, Stallman est rejoint par d'autres développeur·euses. Petit à petit le projet prend de l'ampleur si bien que Stallman crée la FSF⁴ ainsi que toutes les licences garantissant le caractère libre de ces logiciels.

Copy...left

À cet égard, il faut mentionner le *copyleft*, la création juridique qui protège un logiciel libre. Là où le *copyright* assoit le droit de propriété, son contrepied, le *copyleft*, interdit l'appropriation d'un logiciel libre. Ainsi, une fois qu'un projet est mis sous *copyleft* il ne peut désormais être incorporé qu'à des projets libres. Cela peut sembler restrictif mais imaginons un instant ce que signifierait un logiciel libre dépourvu de cette protection. Supposons qu'une entreprise s'empare du code, l'améliore un petit peu, puis le revend. Tout le succès et tout le bénéfice lui reviendrait alors. À l'inverse, si le·a développeur·euse libre du logiciel veut à présent faire une retouche sur le nouveau produit, cela pourrait lui être refusé puisque c'est maintenant la propriété de l'entreprise (mais de toute façon iel ne le veut plus car le nouveau logiciel qu'iel créerait ne serait plus libre). On sent que dans de telles circonstances, le caractère restrictif de la définition du libre se retournerait contre lui-même et étoufferait de nombreux projets. Le *copyleft* agit donc précisément comme un moyen de défense contre les clauses propriétaires en propageant le caractère libre d'un projet initial à toutes ses transformations ultérieures. Ainsi, de la même manière que, dans l'esprit du développement d'un logiciel libre, il est strictement interdit d'incorporer, ne serait-ce qu'une ligne de code provenant d'un projet non libre, les projets propriétaires, à cause du *copyleft*, ne peuvent s'emparer du code des logiciels libres. Un cas de figure différent encore est celui des logiciels *open source* qui eux, plus souples, laissent leur code se faire allègrement accaparer par des projets propriétaires. (Un exemple ? Prenez votre smartphone, et, en cherchant un peu, vous trouverez dans les paramètres une page d'informations légales qui répertorie les licences des logiciels *open source* qui y sont utilisés.) C'est d'ailleurs en ceci que réside la différence notoire entre libre et *open source*...mais une explication s'impose!

Libre ou open source ?

Une confusion de taille, malheureusement très répandue, est de mélanger ces deux notions. Si un logiciel libre est toujours *open source*, la réciproque est fautive⁵. Précisons tout de suite que, tout comme le terme *free software* donne lieu à des erreurs d'interprétations, le terme *open source* ne doit pas être réduit à la seule accessibilité du code source (on dirait alors « *source available* »). On pourrait citer la définition que l'*Open Source Initiative* (OSI) - le pendant de la FSF pour l'*open source* - articule en dix point-clé mais pour mieux comprendre le terme, revenons-en à son origine.

Le logiciel *open source* est en fait né d'une scission d'avec la communauté du logiciel libre. En 1998, l'OSI est créée afin de prôner une nouvelle façon de développer qui s'inspire directement de celle du projet Linux, grand succès de l'époque. Si les termes « logiciel libre » et « *open source* » renvoient presque au même type de programmes, ils se distinguent par les valeurs qui y sont attachées. Ce qu'il faut comprendre est que l'*open source* est profondément apolitique. Il ne défend aucune des

² Eh oui, on ne finit pas de déplier ce nom récursif...#geekhumour

³ Pour davantage de détails sur cette histoire se référer à <https://www.gnu.org/gnu/thegnuproject.en.html> où Stallman détaille sa position et l'esprit du projet GNU.

⁴ En 2019, Stallman a été exclu de la FSF suite à ses prises de position controversées lors de l'affaire Epstein, une manifestation parmi d'autres de sa misogynie notoire. Il l'a cependant réintégrée en 2021 et y est encore.

⁵ Une façon assez simple de vérifier cela est de constater que les licences reconnues par la FSF come libres sont reconnues par l'OSI mais que celle-ci en reconnaît bien d'autres encore. La *Microsoft Public Licence* par exemple est *open source* mais pas libre.

valeurs de liberté des utilisateurs mais se focalise sur l'efficacité et les avantages pratiques d'un type de développement en communautés ouvertes. Pour se faire une idée de la tonalité des dissensions entre libre et *open source*, écoutons Stallman sur le sujet : « Le mouvement du logiciel libre agit pour la liberté des utilisateurs des programmes; c'est un mouvement pour la liberté et la justice. À l'inverse, l'idée *open source* privilégie d'abord et avant tout l'avantage pratique et ne défend pas de principes. Voici pourquoi nous ne sommes pas d'accord avec l'*open source* et n'employons pas ce terme. »⁶. Si en pratique la nuance est fine, on comprendra qu'il ne faut pas dire à un activiste du libre qu'il est dans de l'*open source*...et réciproquement.

À présent que nous avons en main les définitions, certaines questions se posent d'elles-mêmes.

Pourquoi faire du libre ?

Quel intérêt a un-e développeur-euse à offrir un logiciel libre à la communauté ? En terme d'intérêt financier, aucun, le plus souvent. Les développeur-euses de logiciels libres le font par passion, par engagement ou par simple plaisir d'exercer leur virtuosité du code. Et le plus étonnant est que, même dans un monde mercantile obnubilé par le gain, ce genre d'initiative porte ses fruits. La preuve ? Les logiciels libres célèbres largement utilisés comme *Firefox*, *LibreOffice*, ou *Gimp*.

Des projets sans dirigeants ?

Est-ce une jungle ? À quoi ressemble l'élaboration d'un projet libre/*open source* ? Un développeur du projet *open source* fetchmail ayant suivi de très près les avancées du projet Linux, en donne un témoignage détaillé et très éclairant dans un article *The Cathedral and the Bazaar*⁷, où il compare les deux styles de développements de façon imagée. Ainsi il y a d'une part la « cathédrale », qui représente l'organisation hiérarchisée extrêmement règlementée du développement traditionnel de logiciels propriétaires en entreprise, et d'autre part, le « bazar », pour qualifier la convergence des contributions d'individus hétéroclites autour d'un même projet *open source*. Pour lui il ne fait aucun doute que cette deuxième méthode, la méthode « à la Linux », est la plus prometteuse. Sa force réside en sa capacité à faire converger les égos des développeur-euses autour d'un objectif commun. Pas de hiérarchie mais des degrés d'importance échelonnés par la qualité du code fourni qui seul, permet de gagner de la confiance au sein de la communauté de développement. Elle réunit ainsi les deux ingrédients indispensables au succès d'un projet : diversité des compétences et motivation des contributeur-euses.

Est-ce que ça marche vraiment ?

Ces différences sur le plan du développement et de la diffusion se traduisent-elles par des écarts de performance ? En fait, si l'on a tendance à croire qu'une entreprise est une meilleure garantie pour la qualité d'un logiciel il n'en est rien. Les modèles de développement libre et *open source* ont de nombreux atouts comme par exemple la possibilité de mobiliser un plus grand nombre de contributeur-ices/relecteur-ices, la possibilité d'obtenir plus de retours de la part des communautés de développeur-euses. Le modèle *open source* a connu un tel succès que le monde entrepreneurial, séduit, l'a largement adopté et, à l'heure actuelle, il n'est presque pas de logiciel propriétaire qui n'incorpore, au sein du projet de développements, des parties *open source*. Et ceci inclue les plus grosses entreprises comme *Google*, *Microsoft* et *IBM*⁸.

⁶<https://www.gnu.org/philosophy/open-source-misses-the-point.html>

⁷<http://www.catb.org/esr/writings/homesteading/cathedral-bazaar/>

⁸Google : <https://opensource.google/projects>

Microsoft : <https://opensource.microsoft.com/projects/>

La question de la confidentialité

Au-delà de la performance, d'autres critères surgissent comme la confidentialité ou la sécurité. Il n'est pas de réponse nette à apporter à ces questions, chaque modèle ayant ses défauts. À l'inverse des cas libres et *open source*, l'opacité des logiciels propriétaires empêche dans certains cas de connaître précisément les modalités effectives des applications des politiques de confidentialités, ce qui est d'autant plus problématique que les données personnelles sont souvent au coeur des bénéfices réalisés. À cet égard, on peut rappeler le scandale *Cambridge Analytica* de 2018 qui a révélé que cette firme de consulting avait obtenu de *Facebook* et utilisé pendant 6 ans des données personnelles de 87 millions d'utilisateur.ices du réseau social sans leur accord et en violation directe des politiques de confidentialité.

Quid de la fiabilité ?

Qu'est-ce qui garantit la fiabilité d'un logiciel libre si personne n'en est responsable ? En matière de sécurité, le libre accès au code des programmes est en fait plus souvent un avantage qu'un risque au sens où, tous les utilisateurs ayant intérêt à détecter des failles, le nombre de tests et de vérifications est démultiplié. Bien sûr un esprit malveillant pourrait s'introduire dans un projet et miner intentionnellement ses contributions d'erreurs ou de *malware* mais c'est là que s'exerce la force de la communauté des projets libres et *open source* : il est rare qu'un tel acte de malveillance ne soit pas détecté avant d'avoir pu causer de dommage. Et l'expérience montre que, plus le projet a d'ampleur ou concerne des aspects critiques, plus rapide sera la réaction. Une récente histoire de *xz*, un logiciel *open source* destiné à la compression de données, illustre bien le risque et la résilience de l'*open source*⁹.

Côté pratique, enfin

Tout cela est très beau mais comment faire maintenant pour savoir quel type de logiciel j'utilise et comment trouver un substitut fiable, libre ou *open source* ? Pour la première partie, il suffit de regarder la licence puisque c'est elle qui établit et garantit le cadre juridique. Bien sûr il en existe quelques centaines alors pas si simple de s'y retrouver... mais quelques recherches bien menées du côté des sites de l'OSI ou de la FSF devraient lever toute ambiguïté. Pour ce qui est de la migration vers des logiciels alternatifs, c'est une autre histoire. Et là pas de miracle : il faut fourrager un peu dans le web à la recherche de l'outil dont on a besoin. Mentionnons tout de même à l'intention de notre lecteur.ice attentif.ve un site de l'UE qui en répertorie un grand nombre : <https://european-alternatives.eu/alternatives-to>, ainsi qu'un site gouvernemental : <https://code.gouv.fr/sill/list>.

texte écrit par Nelly Cerf
relu et publié par AliENS

⁹<https://www.redhat.com/en/blog/understanding-red-hats-response-xz-security-incident>