



pas la collaboration sur une même carte. Mais vous pouvez à la sortie l'imprimer, en faire une image et donc aussi l'enregistrer localement (soit directement dans votre navigateur soit comme fichier au format [JSON](#)<sup>[2]</sup>) pour une utilisation ultérieure.

Pour ce qui concerne les usages, Framindmap pourra être utile en situation de brainstorming, prise de notes, formation, classification, structuration d'un projet, etc.

Comme d'habitude, il repose sur une solution libre que nous avons adaptée et francisée, en l'occurrence [mindmaps](#), du JavaScript sous licence [AGPL v3](#). Et comme d'habitude vous êtes cordialement invité(e) à l'installer sur votre propre serveur pour faire respirer le Web (et vos données personnelles) en le décentralisant.

-> [Découvrir et utiliser Framindmap](#)

*PS : Nous sortons de nombreux projets actuellement, merci de les relayer si vous le jugez bon car toute notre équipe marketing commerciale est en vacances actuellement (en fait ça fait 10 ans que ça dure). Merci également de faire en sorte que cela ne soit pas notre chant du cygne [en nous soutenant](#) !*  
□

## Notes

[1] Mais plutôt rudimentaire si on le compare à un traditionnel logiciel de mind mapping *en dur* sur votre ordinateur comme [Freeplane](#). Nous savons où aller pour de futures améliorations □

[2] Format non lu directement par [Freeplane](#) par exemple, mais ce serait un chouette plugin à développer □

---

# Les services en ligne ne sont ni libres ni privés ; ils posent d'autres problèmes

Facebook est-il « libre » ? Gmail est-il « libre » ? Le nébuleux [« cloud computing »](#) est-il « libre » ?

Lorsqu'il s'agit de services en ligne la question est mal posée et mérite d'être précisée, nous affirme ici [Richard Stallman](#) en passant en revue les différents et complexes cas de figure<sup>[1]</sup>.



Les services en ligne ne sont ni libres

# ni privateurs ; ils posent d'autres problèmes

[URL d'origine du document \(GNU.org\)](#)

[Richard Stallman](#) – version du 15 mai 2012 – Gnu.org – Creative Commons By-Nd

(Traduction : Céline Libéral, Mathieu Adoutte, Caner, pour Framalang)

**Les programmes et les services sont deux choses différentes. Un programme est une œuvre que vous pouvez exécuter, un service est une activité avec laquelle vous pouvez interagir.**

Pour les programmes, nous établissons une distinction entre libre et non libre (privateur, ou propriétaire). Plus précisément, cette distinction s'applique à un programme dont vous avez une copie : soit vous [disposez des quatre libertés](#), soit vous n'en disposez pas.

Une activité (telle qu'un service) n'existe pas sous forme de copies. Il n'est donc pas possible d'en avoir une ni d'en faire. Ainsi, les quatre libertés qui définissent un logiciel libre n'ont pas de sens pour les services.

Pour utiliser une analogie culinaire, même si j'ai appris à cuisiner en vous regardant, ma cuisine ne peut pas être une copie de la vôtre. Il se peut que j'aie une copie de la recette que vous utilisez pour cuisiner, et que je m'en serve, car les recettes, comme les programmes, sont des œuvres qui peuvent exister en plusieurs exemplaires. Mais la recette et la manière de cuisiner sont deux choses différentes (et les plats obtenus en sont une troisième).

Avec la technologie actuelle, les services sont souvent implémentés en faisant tourner des programmes sur des ordinateurs mais ce n'est pas le seul moyen (en fait, il existe des services en ligne qui sont implémentés en demandant

à des êtres humains en chair et en os de saisir des réponses à des questions). Dans tous les cas, l'implémentation n'est pas visible par les utilisateurs du service, donc cela n'a aucun impact sur eux.

Un service en ligne peut poser aux utilisateurs le problème du recours à un logiciel libre ou non libre, par le biais du client requis pour l'utiliser. Si le service nécessite l'utilisation d'un programme client non libre, recourir à ce service suppose que vous abandonniez votre liberté à ce programme. Pour beaucoup de services web, ce logiciel non libre est du [code JavaScript](#), installé discrètement dans le navigateur de l'utilisateur. Le programme [GNU LibreJS](#) permet de refuser facilement d'exécuter ce code JavaScript non libre. Mais le problème du logiciel client est selon toute logique distinct de celui du service lui-même.

Il existe un cas où un service est directement comparable à un programme : celui où le service revient à avoir une copie d'un programme donné et à l'exécuter vous-même. On appelle cela SaaS, ou « logiciel en tant que service », et un tel service constitue toujours une régression au plan éthique. Si vous disposiez du programme équivalent, vous auriez le contrôle sur votre informatique, à supposer que le programme soit libre. Mais lorsque vous utilisez le service de quelqu'un d'autre pour faire la même chose, vous ne contrôlez plus rien.

Recourir au SaaS revient à utiliser un programme non libre avec des fonctionnalités de surveillance et une porte dérobée (backdoor) universelle. Donc [vous devriez le refuser, et le remplacer par un logiciel libre](#) qui fait la même chose.

En revanche, les fonctions principales de la plupart des services sont de communiquer et de publier des informations. Ils n'ont rien en commun avec un logiciel que vous exécuteriez vous-même, ce ne sont donc pas du SaaS. Ils ne peuvent pas non plus être remplacés par votre copie d'un programme, car un programme s'exécutant sur vos propres ordinateurs, utilisé

uniquement par vous, n'est pas suffisant en lui-même pour communiquer avec d'autres personnes.

Les services non SaaS peuvent nuire à leurs utilisateurs d'autres manières. Les problèmes qui leur sont liés peuvent être aussi bien la mauvaise utilisation des données que vous leur avez envoyées, que la collecte d'autres données (surveillance). Le [Franklin Street Statement](#)<sup>[2]</sup> a tenté d'aborder cette question, mais nous n'avons pas de position bien établie pour le moment. Ce qui est clair, c'est que les problèmes liés aux services sont différents de ceux qui concernent les programmes. Ainsi, par souci de clarté, il est préférable de ne pas appliquer les termes « libre » et « non libre » à un service.

Supposons qu'un service soit fourni par le biais d'un logiciel : l'opérateur du serveur a des copies de nombreux programmes, et les exécute pour fournir le service. Ces copies peuvent être des logiciels libres, ou non. Si c'est l'opérateur qui a développé les programmes, et qu'il les utilise sans en distribuer de copie, alors ils sont libres (sans que cela signifie grand chose) puisque tout utilisateur (il n'y en a qu'un) détient les quatre libertés.

Si certains programmes ne sont pas libres, cela n'affectera pas directement les utilisateurs du service. Ce ne sont pas eux qui exécutent ces programmes, c'est l'opérateur. Dans certaines situations bien particulières, ces programmes peuvent indirectement affecter les utilisateurs : si le service détient des informations confidentielles, ses utilisateurs peuvent s'inquiéter de la présence de portes dérobées dans les programmes non libres, permettant à d'autres d'accéder à leurs données. En fait, les programmes non libres du serveur obligent les utilisateurs à faire confiance à leurs développeurs ainsi qu'à l'opérateur du service. Le risque concret dépend de chaque cas particulier, et notamment de ce que font ces programmes non libres.

Cependant, il y a une personne qui est *certainement* affectée par les programmes non libres qui fournissent le service, c'est l'opérateur du serveur lui-même. Nous ne lui reprochons pas d'être à la merci de logiciels non libres, et nous n'allons certainement pas le boycotter pour ça. Au contraire, nous nous faisons du souci pour sa liberté, comme pour celle de tout utilisateur de logiciel non libre. Quand nous en avons l'occasion, nous essayons de lui expliquer qu'il met en péril sa liberté, en espérant qu'il bascule vers le logiciel libre.

Inversement, si un opérateur de service utilise GNU/Linux ou d'autres logiciels libres, ce n'est pas une bonne action, c'est tout simplement dans son intérêt. Nous n'allons pas l'encenser, ni le remercier, simplement nous le félicitons d'avoir fait le choix le plus avisé. S'il publie certains de ses logiciels libres, contribuant ainsi à faire avancer la communauté, là nous avons une raison de le remercier. Nous lui suggérons de diffuser ces programmes sous la [GNU Affero GPL](#) (Licence publique générale Affero de GNU), puisqu'ils sont destinés à des serveurs.

### [Pourquoi la GPL Affero ?](#)

Ainsi, nous n'avons pas de règle qui dise que des systèmes libres ne devraient pas utiliser (ou s'appuyer sur) des services (ou des sites) fournis par le biais de logiciels non libres. Cependant, ils ne devraient pas dépendre de services de type SaaS, ni inciter à les utiliser. Les SaaS doivent être remplacés par des logiciels libres. Et, toutes choses égales par ailleurs, il est préférable de privilégier les fournisseurs de service qui apportent leur contribution à la communauté en publiant des logiciels libres utiles. Il est également souhaitable de privilégier les architectures pair-à-pair plutôt que les communications centralisées reposant sur un serveur.

## Notes

[1] Crédit photo : [Maurizio Scorianz](#) (Creative Commons By-Nc)

[2] Franklin Street Statement on Freedom and Network Services : déclaration de Franklin Street sur la liberté et les services en ligne.

---

# Les sous-titres faciles d'Universal Subtitles



[Framatube](#), comme les lecteurs assidus de ce blog le savent déjà, est une initiative lancée il y a environ deux ans, dans le but de regrouper des vidéos en rapport avec le logiciel libre et sa culture, pour en favoriser la diffusion.

Framatube s'appuie pour cela aujourd'hui sur la plateforme Blip.tv et, pas de mystère, sur le travail de bénévoles chargés de produire des sous-titres lorsque la vidéo est dans une langue étrangère, l'équipe [Framalang](#) entrant alors dans la danse <sup>[1]</sup>.

Dans ce dernier cas, le travail nécessaire pour proposer la vidéo sous-titrée va au delà de la simple traduction : il faut préalablement transcrire la bande-son pour la traduire, puis



synchroniser les sous-titres obtenus avec la bande-son, incruster les sous-titres et finalement encoder le tout dans un format libre. Cela demande un travail important, qui nécessite une motivation de longue haleine et plus de travail que les habituelles 6 ou 7 heures par traduction d'article publiée sur le Framablog.

D'autant que, les sous-titres étant au final incrustés dans la vidéo, tout ce travail ne peut être réutilisé pour proposer de nouvelles traductions dans d'autres langues.

Pour prendre l'analogie du logiciel, le code source qui pourrait permettre d'améliorer ou de modifier facilement le résultat n'est pas fourni : paradoxal pour un projet Framasoft !

Heureusement, la [Participation Culture Foundation](#) (fondation à but non lucratif à laquelle on doit notamment le logiciel de télévision par Internet [Miro](#)) développe « [Universal Subtitles](#) » un outil en-ligne qui simplifie grandement la tâche.

Ce projet se place sous le parrainage de la Mozilla Foundation, autre fondation à but non lucratif à qui l'on doit le fameux navigateur web [Firefox](#) et qui a initié le projet Mozilla Drumbeat pour promouvoir des initiatives comme celle-ci, visant à garder le Web Ouvert.

L'objectif d'Universal Subtitles est pour le moins ambitieux, offrir à chaque vidéo disponible sur le web un support de sous-titrage collaboratif, où chacun puisse participer au sous-titrage dans sa langue d'une vidéo rencontrée sur le web.

La solution retenue est habile puisqu'elle consiste, à partir d'une vidéo préexistante sur le Web :

1. à proposer des outils permettant de réaliser un sous-titrage de manière communautaire dont les données sont hébergées sur les serveurs du projet,

2. à générer un simple bout de code mélangeant JavaScript et HTML5 à ajouter sur son site pour afficher un lecteur vidéo personnalisé pour afficher les sous-titres à la volée en surimpression de la vidéo.

Les avantages sont nombreux :

- en associant la transcription et ses traductions aux vidéos, il devient possible d'indexer, analyser et traduire l'information contenue dans ces vidéos de façon automatique (les vidéos pourront être indexées par les moteurs de recherche aussi finement que les textes);
- le travail manuel de sous-titrage se trouve facilité et le résultat peut à tout moment être modifié, complété et amélioré par la communauté;
- tout cela contribuant à généraliser le sous-titrage et la traduction des vidéos, bénéficiant au plus grand nombre d'une part et augmentant d'autant l'accessibilité de ce support de communication. Les personnes atteintes de handicaps visuels ou auditifs pouvant à nouveau (comme aux débuts du web) bénéficier de versions textes lisibles à son rythme ou par un vocalisateur.

Et puis, cela promet de belles batailles concernant les sous-titres communautaires de clips musicaux...

Yostral, l'un des piliers de Framalang, a testé le système et publié un journal sur [LinuxFr.org](http://LinuxFr.org) que nous reproduisons ici avec son aimable autorisation :

*Je viens de tester le site [UniversalSubtitles.org](http://UniversalSubtitles.org), qui se veut le « Wikipédia des sous-titres ».*

*Le but est simple : faciliter le sous-titrage dans n'importe-quelle langue de n'importe-quelle vidéo se trouvant sur le web.*

*Voyant les problèmes que nous avons à [Framalang](http://Framalang) pour être efficace en sous-titrage vidéo, chacun faisant une étape dans*

son coin, sauf la traduction qui se fait sur wiki, pas adapté à ça, j'ai donc fait un petit test sur une simple vidéo, en français, pour voir les possibilités de travail collaboratif de traduction et de sous-titrage.

**La première étape** consiste bien sûr à « importer » une vidéo. En fait on travaille directement sur une vidéo déjà en ligne. Donc pas d'upload sur le site, pas de transcodage, rien. Il suffit simplement de donner l'adresse de la vidéo souhaitée. Pour le moment, les formats supportés sont Ogg, WebM, FLV et donc les sites Youtube et Blip.TV (Dailymotion et Vimeo sont en cours).

Cette vidéo apparaît ensuite dans leur « Widget » de traduction. Une page je suppose remplie de HTML5 et de javascript, qui offre une interface relativement conviviale, claire et avec des raccourcis clavier. C'est succinct, mais suffisant pour faire ce qu'on lui demande. C'est donc ici que se déroule la première des trois étapes qui mènent au sous-titrage original de la vidéo : taper les sous-titres correspondant à la vidéo. Pour ceux qui ont 42 doigts, pas de soucis, vous laissez filer la vidéo et vous entrez le texte en temps réel... pour les autres, on peut toujours faire des pauses et naviguer avant/arrière tout simplement.

**La seconde étape** est la synchronisation du texte que vous avez tapé précédemment. Là, la vidéo défile et on doit appuyer sur une touche à chaque fois que l'on veut que le sous-titrage passe à la ligne suivante. Ici, faut être concentré ! Bien sûr, on peut mettre en pause, revenir en arrière, mais le mieux est de faire défiler et synchroniser au fur et à mesure. Faisable, mais pas évident, surtout sur la longueur.

**La troisième étape** est le peaufinage, où on peut ajuster les sous-titres plus précisément. Si on s'est un peu endormi à l'étape précédente, c'est ici qu'on se réveille.

Au final on nous donne un lien permanent vers la vidéo et ses sous-titres, ainsi qu'un bout de code pour l'intégration dans une page web. La vidéo est toujours la même au même endroit et seuls les sous-titres, qu'on peut se télécharger en .srt, sont sur le site d'[UniversalSubtitles.org](http://UniversalSubtitles.org).

À partir de cette page, on peut rajouter simplement des sous-titres dans d'autres langues, calés sur la VO. Ça veut dire qu'une fois la transcription faite et synchronisée, en anglais ou autre, la traduction en français est vraiment simplifiée : on a juste à la rentrer dans les cases en dessous de chaque ligne de sous-titres de la langue originale. On peut bien sûr la stopper puis la reprendre plus tard. On voit également sur cette page les langues disponibles liées à la vidéo et l'avancement des traductions.

Bref, c'est vraiment du sous-titrage collaboratif. Les outils sont minimalistes mais efficaces. On peut tout ré-éditer, sauvegarder, exporter. Et on peut même uploader un fichier .srt, pour ceux qui ont plus souvent l'occasion de traduire hors ligne plutôt que connectés à Internet.

Comme je l'ai déjà dit, les outils sont, pour le moment en tout cas, assez minimalistes. Par exemple on ne peut pas régler les sous-titres : la police, la couleur, l'emplacement... limitation du HTML5 ou pas encore implanté dans ce programme ? On verra. Mais ça risque de poser parfois des gros problèmes de visibilité des sous-titres. Il m'est arrivé à plusieurs reprises, sur les vidéos qu'on a placées sur [Framatube](http://Framatube), de devoir rajouter une bande noire en-bas, ou les placer en haut, pour pouvoir lire convenablement les sous-titres pour des raisons de contraste avec le fond, qui peut varier...

Un autre bémol, c'est qu'il faut s'enregistrer : il faut soit un compte Twitter, soit un compte OpenID, soit un compte Google... En fait ça ne crée pas de compte à proprement parler, mais se sert de l'existant, ce qui est vraiment un moindre

mal.

*En tout cas c'est un bel outil qui va nous rendre bien des services pour nos sous-titrages.*

*Test fait avec une vidéo de Fred Couchet, de l'April, vidéo hébergée sur Blip.tv :  
<http://universalsubtitles.org/videos/JFyBckvaWyCE/fr/>*

## Notes

[1] Le Framablog de Framasoft, qui publie les travaux des projets Framatube et Framalang, tout cela n'est pas sans rappeler un certain Batman, avec sa Batmobile garée dans sa Batcave... Mais nous nous éloignons déjà du sujet.

---

# Ouvrir ses logiciels mais fermer ses données à l'ère du cloud computing

Voici une courte traduction qui aborde furtivement deux sujets selon nous intéressants. Le premier n'est pas nouveau puisqu'il évoque la traditionnelle [différence d'approche](#) entre le logiciel libre cher à [Richard Stallman](#) et l'open source, à ceci près que l'avènement du cloud computing lui donne un nouvel éclairage.



Le second est peut-être plus original puisqu'il met en parallèle les logiciels et les données pour constater un mouvement opposé.

Nous sommes nombreux à souhaiter que les logiciels deviennent de plus en plus libres. Mais des Google et des Facebooks ont également envie que nos données suivent le même chemin pour pouvoir les manipuler tout à leur guise. C'est même fondamental pour eux puisque c'est tout leur business model qui est construit sur cela.

Or nous nous inquiétons chaque jour davantage du devenir de nos données, et si nous les souhaitons « libres » c'est avant tout libres de ne pas être contrôlées et exploitées sans notre consentement. Liberté et ouverture n'ont donc clairement pas le même sens chez les uns et chez les autres<sup>[1]</sup>.

Il faut dire que dans les nuages : logiciels, formats, fichiers et données s'entrechoquent. Quand par exemple vous faites du traitement de texte directement en ligne (Google Docs, Zoho, etc.), c'est un peu tout à la fois qui est sollicité, sans qu'on n'arrive plus trop bien à les distinguer.

« Ouvrons » nos logiciels mais « fermons » nos données ? C'est en résumé, la question brutale que pose ce billet.

## **Libérez mes logiciels, pas mes données**

### **[Open source my software but not my data](#)**

*Dana Blankenhorn – 27 avril 2010 – ZDNet (Blog Linux and Open Source)*

*(Traduction Framalang : Kovalsky, Barbidule et Goofy)*

Comme Google avant lui, Facebook fait l'objet d'une [attention accrue](#) pour son interprétation du terme « ouvert » dans le monde en ligne.

Que les logiciels soient libres est une bonne chose. Mais que les données soient ouvertes ? Peut être pas tant que ça.

L'affirmation classique concernant le logiciel est qu'à moins que vous utilisiez l'[AGPL](#), à moins que tout ne soit ouvert y compris vos sources secrètes, vous n'êtes pas vraiment ouvert, vous prétendez seulement l'être. Ouvert serait juste un autre mot pour dire que vous n'avez rien à cacher.

Je n'y ai jamais cru. L'open source n'est pas la même chose que le logiciel libre, c'est une des premières leçons qu'on m'a apprises quand j'ai commencé ce combat. (Richard Stallman s'en est chargé personnellement.)

L'open source est un continuum de choix, allant de l'idéal des logiciels libres de Stallman jusqu'au code de Microsoft sous restrictions serrées. L'open source est né en réaction logiciel libre de Stallman, et parfois en opposition à celui-ci.

Précédemment, j'ai mis au point une [courbe de l'open source](#), pour illustrer l'étendue des choix disponibles. Plus vous avez besoin d'une participation de la communauté, plus vous êtes en bas de la courbe. Plus votre contrôle de la propriété du code augmente, plus vous êtes en haut.

Plus tard j'ai modifié cela en élaborant une [courbe du développement open source](#), prenant en compte différents modèles de développement.

Ce qui est notable à propos de l'essentiel du code conçu pour être utilisé en ligne, c'est qu'il n'est généralement pas en bas de la courbe. Même Google n'est pas en bas de la courbe, bien qu'il soit un membre de la communauté open source tout à fait respectable. Google ne soutient pas l'AGPL.

Mais qu'en est il des données ? Qui décide du statut des données en ligne ? Est ce que la décision vous appartient, ou revient-elle aux entreprises qui hébergent les données ?

Facebook a [assimilé les données à du logiciel](#), et il se permet alors de les diffuser dans la nature, en affirmant qu'il ne fait que suivre les principes de l'open source.

Quand vous comparez libre et propriétaire dans le monde logiciel, le libre semble formidable. Mais comparez-les sous l'angle des données, sur le mode « vos données seront ouvertes sauf si vous dites non », et les Sénateurs vont y voir [une violation de la vie privée](#). En particulier si, comme Facebook, vous vous étiez vous-même défini jusqu'à récemment comme un réseau privé sans risque pour les enfants, et non comme un classique espace ouvert du Web.

Il est facile pour les logiciels de se déplacer vers le haut ou le bas de la courbe de l'open source. Pour les données cela se révèle problématique.

## Notes

[1] Crédit photo : [Katayun](#) (Creative Commons By)

---

# Le CRM libre et citoyen CiviCRM adopté par la FSF

CRM est l'acronyme anglais de Customer Relationship Management, que l'on traduit chez nous par [Gestion de la relation client](#).





Dans la communauté du libre, il y a un CRM qui a le vent en poupe actuellement, c'est CiviCRM. Déjà utilisé par Creative Commons ou la Fondation Wikimedia, c'est aujourd'hui la Free Software Foundation (FSF) de Richard Stallman qui a décidé de l'adopter (et nous invite à faire autant), en nous disant tout le bien qu'elle en pense dans un communiqué traduit ci-dessous.

Remarque : On est beaucoup plus ici dans la gestion de la relation de *membres* d'associations que dans la gestion de la relation de *clients*.

*PS : La prochaine étape sera-t-elle la Paypal libération ?*

## **Pour les associations à but non lucratif, il est temps d'abandonner les outils propriétaires de collecte de dons**

*Matt Lee – 14 avril 2010 – FSF.org  
(Traduction Framalang : Eric Moreau)*

**La Free Software Foundation (FSF) annonce aujourd'hui que CiviCRM a obtenu son agrément en tant que système complet de gestion de donations et de relations clients pour les associations à but non-lucratif.**

Dans le cadre de sa campagne [hautement prioritaires](#), la FSF avait souligné la nécessité d'une solution libre dans ce secteur. En parallèle à ce communiqué, la FSF adopte [CiviCRM](#) pour ses propres besoins, et encourage vivement les autres associations sans but lucratif à en faire autant.

Depuis toujours, les associations sans but lucratif sont dépendantes de solutions propriétaires ou « SaaS » (*NdT : Software as a Service, logiciel en tant que service en ligne*) pour la collecte de dons, comme Blackbaud's Raiser's Edge ou eTapestry. Les organisations qui les utilisent en sont prisonnières, ont peu de contrôle sur les fonctionnalités du

logiciel, et sont à la merci des caprices d'une seule entreprise. Ces associations doivent également assumer le coût de la migration si elles souhaitent passer à un système propriétaire différent, et ne parviennent pas à gagner leur indépendance. Dans ces conditions, les outils censés améliorer leur efficacité finissent en réalité par réduire leur capacité à accomplir leurs missions sociales.

CiviCRM, au contraire, partage le code de son logiciel pour permettre aux associations de comprendre son fonctionnement, offre la possibilité à tout un chacun de lui apporter des améliorations, et peut aussi l'héberger sur ses serveurs sécurisés. Le code et le format de données étant libres, utiliser ce programme n'implique pas qu'on en devienne prisonnier. Puisqu'il est disponible pour le système d'exploitation libre GNU/Linux, il permet aussi de s'affranchir d'un autre logiciel souvent nécessaire à l'utilisation des solutions propriétaires de collecte de dons : Microsoft Windows.

« Les avantages qu'offre CiviCRM satisferont les associations qui souhaitent organiser leurs relations avec les donateurs, les sympathisants et les médias. En plus de sa fonction de répertoire de contacts, il prend en charge la collecte de fonds en ligne, l'inscription à des manifestations, la gestion des adhésions et l'envoi de courriers personnalisés, électroniques ou traditionnels. Cerise sur le gâteau, ce logiciel libre est distribué sous la licence GNU Affero General Public License : les associations peuvent l'héberger elles-mêmes et garder la liberté nécessaire afin de mener leur activité sans entraves », a déclaré John Sullivan, le directeur des opérations de la FSF.

Les idéaux du logiciel libre encourageant le partage et la modification sont au cœur du développement de CiviCRM, explique le développeur Dave Greenberg. « Le projet CiviCRM a été lancé par un groupe de développeurs et de directeurs de projet qui avaient auparavant collaboré sur une application

propriétaire de gestion des dons. Étant très désireux d'accroître la portée et l'efficacité des associations sans but lucratif, nous nous sommes rendu compte qu'existait le besoin d'une application de GRC capable de répondre aux demandes du secteur associatif. Dès le départ, il nous a paru évident que celle-ci devait être un logiciel libre, développée en collaboration avec une communauté qui en aurait la jouissance. En ce qui me concerne, je trouve l'interaction avec notre communauté d'utilisateurs stimulante et gratifiante intellectuellement. Voir des gens qui possèdent une expertise dans un domaine particulier donner de leur temps et nous soumettre des idées pour nous aider à améliorer le produit, c'est très excitant. »

En adoptant ce nouveau service, la FSF rejoint d'autres organisations telles qu'Amnesty International, Creative Commons et la Fondation Wikimédia, qui utilisent elles aussi CiviCRM.

Le directeur exécutif Peter Brown décrit comment la FSF utilise logiciel et a l'intention de la donner publiquement en exemple : « Je me réjouis d'encourager les autres associations sans but lucratif à échapper à leur logiciels propriétaires actuels ou à leurs systèmes de « logiciels en tant que service » et à essayer plutôt CiviCRM. La FSF gère plus de 40 000 contacts et 15 000 dons par an, une branche éditoriale, une boutique en ligne et plusieurs sites de campagne auxquels sont associées des listes de diffusions – le tout avec des logiciels libres.

Un système complet de gestion des dons et de relations clients constituera la dernière pièce du puzzle pour les associations caritatives qui désirent n'utiliser que des logiciels libres. Nous prévoyons de publier un guide destiné à faire partager notre expérience à d'autres associations soucieuses de l'aspect éthique de leurs logiciels.

Nathan Yergler, responsable des ressources informatiques de

Creative Commons, fait lui aussi l'éloge de cet outil : « CiviCRM est une composante cruciale de l'infrastructure de Creative Commons. Nous avons vu cette application mûrir et s'améliorer au fil du temps, offrir de nouvelles fonctionnalités et des performances meilleures à chaque nouvelle version. La communauté de développeurs de CiviCRM est accessible et réactive, et se met en quatre pour aider ses utilisateurs lorsque c'est nécessaire. Je recommande chaudement CiviCRM aux associations semblables à Creative Commons qui cherchent une solution de GRC. »

Piotr Szotkowski, membre de l'équipe principale de CiviCRM, indique que malgré la maturité du projet, il reste encore du travail gratifiant à accomplir : « Tous ceux qui souhaitent nous aider sont les bienvenus. Contribuer au développement de CiviCRM est source de nombreuses satisfactions, par exemple lorsqu'on sait que le code qu'on a écrit a servi à aider les victimes de l'ouragan Katrina, qu'il aide les associations comme Amnesty International ou Front Line à se battre pour les droits de l'Homme, ou la Fondation Wikimedia à mieux organiser son excellent travail sur Wikipedia et tous ses autres projets. »

Pour savoir où télécharger CiviCRM, comment l'utiliser et comment y contribuer, rendez-vous à l'adresse : <http://civicrm.org>. Vous trouverez une discussion ayant pour sujet la comparaison des solutions disponibles pour les bases de données libres sur le wiki LibrePlanet de la FSF.

Pour connaître les dangers que représente l'utilisation des « logiciels en tant que service », consulter [Who does that server really serve?](#) (NdT : *De qui ce serveur est-il vraiment le serviteur ?*).

---

# Cloud computing, logiciel libre et service public

En ce début d'investiture Obama, de nombreux internautes expriment souhaits et desiderata à la nouvelle administration.



Le vœu de William Hurley a retenu notre attention. « Je suggère que vous créiez un [cloud computing](#) financé par l'État, mis à la disposition des facultés et des universités », nous dit-il. Nous n'en saurons pas vraiment plus car il n'est pas entré dans les détails mais nous avons néanmoins saisi la perche pour en faire un billet où nous pourrions nous interroger ensemble sur les relations entre *l'informatique dans les nuages*, le logiciel libre et un éventuel rôle de la puissance publique.

Placer nos données personnelles, nos fichiers, nos liens, etc. sur le *Grand Internet* présente en effet de nombreux avantages pratiques à l'heure où l'on peut « quasiment » se connecter partout tout le temps. Mais ce n'est pas sans poser quelques questions pour ne pas dire quelques problèmes. Je pense par exemple à la mésaventure récente de ce pauvre [Marc L\\*\\*\\*<sup>\[1\]</sup>](#) que l'on pouvait suivre au jour le jour à la trace via Facebook et consorts. Je pense également à notre récent billet [La tête dans les nuages mais les pieds sur terre](#). Je pense enfin à la toile tissée méthodiquement par Google avec [tous ses services en ligne](#) (Gmail, Reader, Maps, Earth, Picasa, Calendar, Docs, YouTube...) que nous sommes nombreux à utiliser au quotidien.

Et puisqu'il est question d'éducation supérieure et

d'université<sup>[2]</sup>, je dois bien vous avouer que, l'année dernière, lorsqu'il a été question de créer en deux-trois coups de cuillère à pot des adresses de messagerie (avec ou sans chat) pour tous les enseignants et étudiants de mon établissement scolaire, de se doter d'agendas partagés et d'une suite bureautique en ligne (fichiers tableurs et traitements de texte potentiellement accessibles en lecture écriture, selon les droits, à toute la communauté, et disponibles au [format ODF](#)), je me suis tourné vers... [Google Apps Education](#) sans avoir « malheureusement » à le regretter.

En tant que responsable [TICE](#) de mon lycée J'ai donc osé confier les données de mes élèves à Google ! Est-ce grave docteur ? Oui ça l'est ! J'en ai bien conscience et je compte bien un jour me soigner (quand bien même Google n'affiche bien entendu pas de publicités pour ce service spécifique au monde éducatif). Il faut tout de même dire, à ma décharge, que par rapport à ma problématique et à mes besoins (temps limité, budget nul et situation particulière d'un lycée à l'étranger) je ne pouvais raisonnablement pas m'appuyer sur un quelconque [Espace Numérique de Travail \(ENT\)](#) dont je doute de plus en plus de leur pertinence et efficacité à mesure que le temps passe (surtout si l'on continue à s'obstiner à les développer académie par académie).

Les partisans du logiciel libre peuvent sensibiliser sur les risques encourus à confier nos documents et informations numériques à des services en ligne « gratuits » proposés par des sociétés commerciales « web 2.0 » qui n'offrent pas de garanties sur l'avenir et sont souvent opaques au niveau de nos droits et des formats. Ils peuvent pousser à ce que des licences plus adaptées et plus transparentes soient adoptées (telle la [licence AGPL](#)). Mais, contrairement à un Microsoft où il « suffisait » de proposer des alternatives logicielles libres, ils ne peuvent absolument pas concurrencer un Google sur son terrain, c'est-à-dire justement le cloud computing, qui nécessite des investissements très très lourds ne serait-

ce que pour pour installer et maintenir les batteries de serveurs disséminés un peu partout sur le réseau. Et alors je crois effectivement que le politique et le [secteur public](#) (national ou supra-national) peuvent nous être d'un grand secours pour modifier la donne (si tant est que l'on juge que la donne mérite modification).

C'est certainement l'alliance « logiciel libre + secteur public » qui pourra faire en sorte de ne pas laisser le champ libre aux seules sociétés privées. Ne privatisons pas la totalité du cloud computing (surtout dans le domaine éducatif), voilà par extrapolation, le sujet du billet du jour. Un peu comme ce qu'a voulu faire le projet de bibliothèque numérique européenne [Europeana](#) pour contrarier [Google Books](#) avec pour le moment [le succès que l'on sait...](#)

## **L'enseignement supérieur a besoin d'un nuage informatique national**

[Higher education needs a national computing cloud](#)

*William Hurley – 26 janvier 2009 – InfoWorld  
(Traduction Framalang : Don Rico)*

Le cloud computing (ou *informatique dématérialisée* ou *informatique dans les nuages*), est vital pour l'avenir de l'enseignement supérieur aux États-Unis, et j'invite le Président Obama à agir

Le 26 janvier 2009  
M. le Président Barack Obama  
La Maison Blanche  
Washington, DC 20500-0001

M. le Président,

Je tiens à vous adresser mes plus sincères félicitations, Monsieur, pour votre récente investiture à la fonction de 44ème Président des États-Unis d'Amérique. Votre victoire est

la preuve de la grandeur de notre démocratie, mais aussi de la capacité de transformation de cette même démocratie. Comme des millions de mes semblables du monde entier, j'ai regardé avec une grande fierté votre prestation de serment, lors de laquelle vous êtes devenu un exemple vivant de l'impact que peut avoir un seul citoyen américain.

Vous avez déclaré « Le monde change, et nous devons changer avec lui ». Je suis sûr que nous ne pouvons plus d'accord, M. le Président, et je crois que la politique que mènera votre administration sera une fontaine d'innovation. Je sais que le vice-président et vous êtes profondément favorables au développement des initiatives de recherche dans les instituts d'enseignement supérieur qui sont au cœur de l'innovation américaine. Pour ces instituts, l'avenir est déjà là. Mais, comme l'a écrit William Gibson « il n'est pas encore tout à fait équitablement réparti ».

Nous avons laissé le coût de la technologie entraver notre capacité à innover. Les chercheurs ne sont plus limités par le manque d'idées ou de connaissances, mais plutôt par les moyens informatiques nécessaires pour conduire des expériences et en analyser les résultats.

Je suggère donc que vous créiez un cloud computing financé par l'État, mis à la disposition des facultés et des universités. Une telle ressource aurait le mérite de niveler le terrain universitaire. Les chercheurs qui travaillent d'arrache-pied dans des milliers d'instituts de taille modeste auraient alors accès à une puissance informatique qui n'est pour l'instant accessible qu'à une poignée de privilégiés. Il nous est impossible de prédire d'où viendra la prochaine grande innovation, mais des ressources informatiques dématérialisées publiques amélioreraient de façon extraordinaire nos moyens de coopérer et d'innover au niveau national.

Les grandes avancées technologiques et sociales peuvent se produire presque simultanément. En septembre 1962, un jeune



chercheur publiait ce qui donnerait naissance aux plus grandes avancées technologiques de notre temps, et au même moment, un autre jeune homme originaire de Kosciusko, dans le Missouri, amorçait un itinéraire personnel qui aboutirait à un moment charnière dans la lutte pour les droits civiques. D'aucuns peuvent y voir une coïncidence, mais en ce qui me concerne j'y vois la providence. Les recommandations de [Paul Baran](#) en faveur d'une structure nationale publique destinée à transporter des données informatiques et l'entrée de [James Meredith](#) à l'université du Mississippi ont, du point de vue technologique et social, changé les États-Unis en profondeur.

Votre administration, par sa connaissance des nouvelles technologies, tient l'occasion d'accomplir un autre grand bond en avant. On ne peut comparer ma lettre à l'article de Baran, mais j'espère suggérer cette idée au moment opportun. Une idée trop en avance sur son temps a aussi peu de valeur qu'une idée avancée après que des engagements ont déjà été pris. J'espère donc attirer votre attention maintenant, avant que vos projets de réformes pour l'éducation aient été élaborés, et tant qu'il reste du temps pour prévoir le financement d'un cloud computing ayant le potentiel de transformer des chercheurs dispersés et inégaux en une locomotive d'innovation la plus puissante du monde.

Encore une fois, M. le Président, je tiens à vous féliciter, votre équipe et vous, pour votre victoire grandiose acquise grâce aux nouvelles technologies, ainsi que pour votre investiture inattendue, exaltante et triomphante.

## Notes

[1] J'en profite pour saluer ici le [très libre](#) curieux magazine curieux Le Tigre.

[2] Crédit photo : [Per Ola Wiberg](#) (Creative Commons By)

---

# Le réseau Framasoft en info continue !

Pour vous récompenser d'avoir été sages en 2008, nous vous offrons un cadeau pour bien commencer l'année : un canal de [microblogging](#), pour encore plus de Framasoft dans votre PC ou tout autre objet capable de se connecter au Net !



Plus sérieusement, pourquoi le microblogging ? Parce que c'est un outil qui nous permettra d'ajouter une nouvelle corde à notre arc et de diffuser autrement des infos sur Framasoft et le monde du libre<sup>[1]</sup>.

Ce sera pour nous un moyen d'assurer un suivi de nos billets, d'exercer une veille des infos concernant le monde du libre, et de vous tenir au courant de l'actualité de Framasoft (non, vous n'y apprendrez pas ce qu'aKa a pris au petit-déj ni quelle couleur de chaussettes j'ai choisi de porter).

Côté technique, nous diffuserons nos mini-billets sur deux canaux : [identi.ca](#) et [Twitter](#). Twitter, parce qu'étant le réseau le plus fréquenté c'est celui qui nous permettra de toucher le plus large public, et d'attirer de nouveaux lecteurs dans nos filets. Mais nos billets n'arriveront sur Twitter que par rebond. Je m'explique...

Nous ne pouvons nous contenter d'un outil fermé, aussi sympathique soit-il. Nous avons donc opté pour le service

[identi.ca](http://identi.ca), similaire à Twitter dans le principe, mais complètement libre. Il est basé sur le logiciel [Laconica](http://laconica.com), qui est Open Source, et placé sous licence [AGPL](http://www.gnu.org/licenses/agpl.html), comme préconisé par Richard Stallman (cf. [notre billet sur Clipperz](#)). Les mini-billets publiés via cette plateforme sont quant à eux sous licence [Creative Commons Paternité 3.0](http://creativecommons.org/licenses/by/3.0/). Les flux sont compatibles avec le réseau Jabber, et on peut donc les recevoir directement dans [Pidgin](http://pidgin.im), par exemple. Et les billets publiés à partir de [laconica](http://laconica.com) sont directement répercutés sur notre compte Twitter. Qui plus est, mais c'est personnel, je trouve l'interface de Laconica plus sympa...

À présent, venons-en au fait ! Voici où vous pourrez suivre nos flux de micro-blogging<sup>[2]</sup> :

- sur [identi.ca](http://identi.ca) : <http://identi.ca/framasoft/all>
- sur Twitter : <http://twitter.com/framasoft>

Nous espérons que vous serez nombreux à nous lire !



Hé Momo, approche c'est l'heure des grosses tronches, ils ont invité Stallman et Nine Inch Nails :

## Notes

[1] Crédit photo : [Ali Brohi](#) (Creative Commons By)

[2] Crédit illustration : [Harrypopof](#) (Licence Art Libre)

---

# La tête dans les nuages mais les pieds sur terre

GMail, Google Apps, Zoho, Flickr, Del.icio.us, Box.net, Wuala, DropBox, Plaxo... Ça vous parle ? Vous savez, ces applications en ligne (ou Web Apps) pratiques et séduisantes qui poussent comme des champignons aux quatre coins de la Toile, souvent accompagnées de la mention *beta* pour faire hi-tech. On a même trouvé un terme pour englober tout ce petit monde, le très à la mode [Cloud Computing](#), soit l'informatique *dans les nuages* ou *dématérialisée*<sup>[1]</sup>.



Que celui qui n'a pas un compte chez l'un de ces services en ligne me jette la première pierre. Reconnaissons qu'il est fort commode, notamment pour qui travaille sur plusieurs postes ou de façon nomade, de pouvoir accéder à ses courriels, à ses documents et à certaines données depuis n'importe quel PC (voire téléphone mobile) à condition de disposer d'une connexion Internet.

Mais n'est-on pas en droit de s'inquiéter de savoir que tant de nos données se baladent on ne sait trop où dans l'espace virtuel ? Plaxo, par exemple, avait été critiqué à ses débuts car assez porté sur le spam et l'exploitation peu scrupuleuse des carnets d'adresses qu'on lui confiait (même si ce service a depuis redressé la barre). Quid des documents créés avec Google Docs, des fichiers conservés chez Box.net, des mails échangés avec GMail ? De plus en plus de grands groupes ou de start-ups se lancent sur ce marché apparemment juteux et se battent pour posséder nos données.

Il y a quelque temps déjà, plusieurs voix s'élevaient contre ces services en ligne : [Larry Ellison](#), le fondateur d'Oracle, [qualifiait le Cloud Computing de mode](#), et [Richard Stallman](#), dans un entretien accordé au quotidien anglais [The Guardian](#), allait plus loin [en taxant ces services en ligne de pièges](#). Stallman mettait en garde les utilisateurs contre ces Web Apps et le stockage de données personnelles sur les serveurs d'entreprises commerciales : selon lui, confier ses données à de tels services revient à en perdre le contrôle et pose donc les mêmes problèmes que l'utilisation des logiciels propriétaires.

Stallman, qui n'a pas l'habitude de faire dans la nuance, recommandait donc de n'utiliser aucun de ses services et de leur préférer nos bonnes vieilles applis *en dur* sur lesquelles nous gardons tout contrôle. (Nul doute qu'il pensait par là à la [Framakey](#)...)

Tim O'Reilly, dans son blog, s'est lui aussi penché sur la question et a publié un [billet](#) assez fourni, dans lequel il estime qu'il faudrait appliquer au Cloud Computing les principes de l'Open Source et rappelle qu'il avait déjà mis en garde contre le verrouillage du Web, pourtant basé sur des programmes et outils Open Source, par des applications Web 2.0.

Cette question de la confidentialité, du contrôle des données et de l'indépendance de l'utilisateur face au logiciel concerne tous ceux qui ont un usage intensif du Web et de l'outil informatique, mais semble cruciale pour les adeptes du logiciel libre, très sensibles à ces questions. Comme dans le software classique, certains acteurs du Cloud Computing proposent des solutions libres. C'est le cas de [Clipperz](#), qui a développé un gestionnaire de mots de passe et d'informations personnelles sous licence GPL.

Sur le blog de Clipperz , un des auteurs appelle les utilisateurs et les développeurs à agir pour préserver la

liberté et la confidentialité *dans les nuages*, et propose quelques mesures pour que les applications Web 2.0 soient en accord avec les valeurs du libre, du point de vue des licences et du comportements des navigateurs Internet par exemple. On y retrouve par billet interposé Richard Stallman, avec qui l'auteur s'est entretenu et qui y va lui aussi de ses conseils.

Histoire de redescendre un peu sur terre après tant de temps passé dans les nuages, nous vous présentons donc la traduction de ce billet, réalisée par notre équipe Framalang.

## **Liberté et protection de la vie privée en ligne : agissez !**

### [Freedom and Privacy in the Cloud – a call for action](#)

*Marco – 30 mai 2008 – Clipperz*

*(Traduction Framalang : Olivier, Burbumpa et Don Rico)*

Ce message traite de la liberté. La liberté de posséder vos données et la liberté d'utiliser des logiciels libres. Vous devriez aussi pouvoir exiger ces libertés et en jouir quand vous utilisez des applications web.

Si vous soutenez le [mouvement du logiciel libre](#), vous pouvez facilement opter pour Gimp au lieu de Photoshop, pour Firefox au lieu d'Internet Explorer. Vous pouvez également protéger le caractère privé de vos données en utilisant les outils de cryptage disponibles ([GPG](#), [TrueCrypt](#)...). Mais dès qu'il s'agit d'applications web, tout se complique.

Les avantages des applications web – ou web apps – (accessibles partout et tout le temps, mises à jour transparentes, stockage fiable, ...) sont nombreux, mais bien souvent les utilisateurs perdent la liberté d'étudier, de modifier et de discuter du code source qui fait tourner ces web apps.

De plus, nous sommes contraints de confier nos données aux fournisseurs de ces web apps (marque-pages, documents rédigés, copies des discussions, informations financières et désormais... [dossiers médicaux](#)) qui ne résident alors plus sur nos disques durs mais qui sont rangés quelque part *dans les nuages*. Ce n'est pas vraiment une situation confortable de devoir choisir entre aspect pratique et liberté.

Que l'on soit clair : les web apps sont formidables et je les adore. Mais je pense que le moment est venu de réclamer plus de liberté et de confidentialité. Voilà comment nous pouvons obtenir ces deux résultats en trois étapes.

## 1. Choisissez l'AGPL

Quelle est l'importance de l'[AGPL](#) ? Si vous êtes un fournisseur de services et que vos services s'appuient sur des logiciels placés sous licence AGPL, vous devez rendre le code source disponible à toute personne utilisant ce service. La FSF suggère dans ses [directives](#) de placer un lien *Source* qui renvoie à une archive contenant le code source directement dans l'interface de l'application web.

(Ne me demandez pas pourquoi la communauté des logiciels libres a mis tant de temps à réagir !)

### Mesures

- Aider Clipperz à mettre au point une *suite AGPL* : un ensemble d'applications web répondant aux besoins les plus courants.
- Cette suite devrait comprendre : un traitement de texte, un logiciel de discussion, un gestionnaire de mots de passe, un carnet d'adresses, un pense-bête, un calendrier, un gestionnaire de marque-pages ... Et chaque web app devra être soumise à la licence AGPL ! Vous pourrez alors oublier Google, del.icio.us, Plaxo, Meebo ... à moins qu'ils ne se mettent à l'AGPL aussi.
- Nous avons déjà deux candidats pour certains postes

([Ajax Chat](#) pour les discussions en ligne et, bien sûr, [Clipperz](#) pour le gestionnaire de mots de passe), mais la plupart des places sont encore à pourvoir !

- Aider Clipperz à diffuser les bienfaits de l'AGPL auprès des développeurs de projets web open-source. Demandez-leur de se convertir à l'AGPL.

## 2. Ajoutez-y une pointe de divulgation nulle de données

Les développeurs Web, comme les utilisateurs, connaissent encore assez peu les possibilités offertes par le chiffrement via un navigateur pour rendre les applications web aussi sécurisées et confidentielles que les logiciels classiques.

Chez Clipperz, nous voulons apporter une nouvelle vision que nous appelons *les web apps à divulgation nulle de données* ([description plus détaillée ici](#)) qui associe l'idée d'un hébergement auquel même l'hébergeur n'a pas accès et un ensemble de règles basées sur le credo *confidentialité absolue*.

Ce nom est aussi bien un hommage au chiffrement (une *garantie de divulgation nulle de données* est un protocole de chiffrement standard) que la promesse d'une relation particulière entre l'utilisateur et le fournisseur d'application. Le serveur hébergeant la web app peut ne rien savoir sur ses utilisateurs, pas même leurs identifiants ! Clipperz applique cette vision pour mettre en œuvre son [gestionnaire de mots de passe en ligne](#).

### Mesures

- Appliquer les techniques *divulgation nulle de données* à chaque composant de la *suite AGPL*. Convertir une application web à l'architecture *divulgation nulle* n'est pas simple, mais chez Clipperz nous avons développé un savoir-faire important et nous serons heureux de partager aussi bien ces connaissances que le code de



base.

Nous pourrions ainsi finalement jouir d'un traitement de texte en ligne qui ne pourra pas lire nos documents, un logiciel de discussion qui n'enregistrera pas nos conversations, un wiki sur lequel on pourra conserver sans crainte des données importantes, etc.

- Établir et maintenir à jour une liste des Fournisseurs de Service d'Application (*NdT : ASP pour Application Service Provider en anglais*) qui hébergent la suite complète sous AGPL. Cette référence sera utile à tous ceux qui attachent de l'importance aux logiciels libres et à la confidentialité mais ne possèdent pas les compétences et les ressources pour faire tourner des web apps sur leur propre serveur.

### **3. Créer un navigateur plus intelligent**

On y est presque, mais il nous reste encore à fournir aux utilisateurs de web apps un environnement encore plus flexible et sécurisé. Dans la pratique, du fait de l'architecture des web apps à *divulgation nulle de données*, le serveur réalise de façon générale les tâches suivantes :

- charger le code Javascript dans le navigateur de l'utilisateur (charger le programme) ;
- authentifier l'utilisateur (optionnel et par un protocole à *divulgation nulle de données*)
- rapatrier et stocker les données chiffrées demandées par le navigateur de l'utilisateur.

Logiciel libre est synonyme de contrôle total de ce qui se passe sur mon ordinateur. Se posent alors deux questions :

- Comment faire tourner une version modifiée du code Javascript à la place de celui chargé par le serveur ?
- Comment être alerté des modifications apportées au code Javascript que le serveur envoie à mon navigateur ?

J'ai récemment eu l'immense honneur d'échanger mes idées avec [Richard Stallman](#) lui-même au sujet de ces problèmes, et il a suggéré une solution futée pour les résoudre tous les deux.

Stallman propose d'ajouter une fonctionnalité au navigateur qui permette à l'utilisateur de dire : « Quand tu charges l'URL X, utilise le code Javascript de l'URL Y comme s'il venait de l'URL X ». Si l'utilisateur fait appel à cette fonctionnalité, il peut utiliser sa propre copie du code Javascript et peut toujours échanger des données avec le serveur hébergeant l'application web.

Un navigateur possédant cette capacité pourrait aussi facilement vérifier si le script Java de l'URL X est différent du script Java sauvegardé à l'URL Y. Si l'utilisateur fait confiance à la version courante du code Javascript de l'URL X, il peut en faire une copie à l'URL Y et sera ainsi alerté de tout changement. Cette solution protège l'utilisateur du code malveillant qu'il pourrait exécuter sans le savoir dans son navigateur, du code qui pourrait voler ses données et détruire l'architecture à divulgation nulle d'information.

### **Mesures :**

- Écrire des extensions pour les principaux navigateurs libres (Mozilla, Webkit, ...) qui mettent en œuvre l'idée de Stallman.

Militer pour l'adjonction de la "suite AGPL" et des navigateurs améliorés pré-cités dans les distributions GNU/Linux.

- Continuez à lire ce blog où je posterai de nouveaux articles régulièrement.
- [Faites-moi part](#) de vos commentaires et suggestions.
- Faites passer le message au travers de vos blogs, de vos messages sur les forums, ...
- Faites un [don](#).

Et le meilleur pour la fin : comment nommeriez-vous cet ambitieux projet ? Faites-moi part de vos idées dans les commentaires !

## Notes

[1] Crédit photo : [Nicholas T](#) (Creative Commons By)