



**HAL**  
open science

## ¡Viva la libertad!

Alban Mancheron

► **To cite this version:**

| Alban Mancheron. ¡Viva la libertad!. GNU/Linux Magazine, 2021, 254, pp.14-20. lirmm-03508567

**HAL Id: lirmm-03508567**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-03508567>**

Submitted on 3 Jan 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

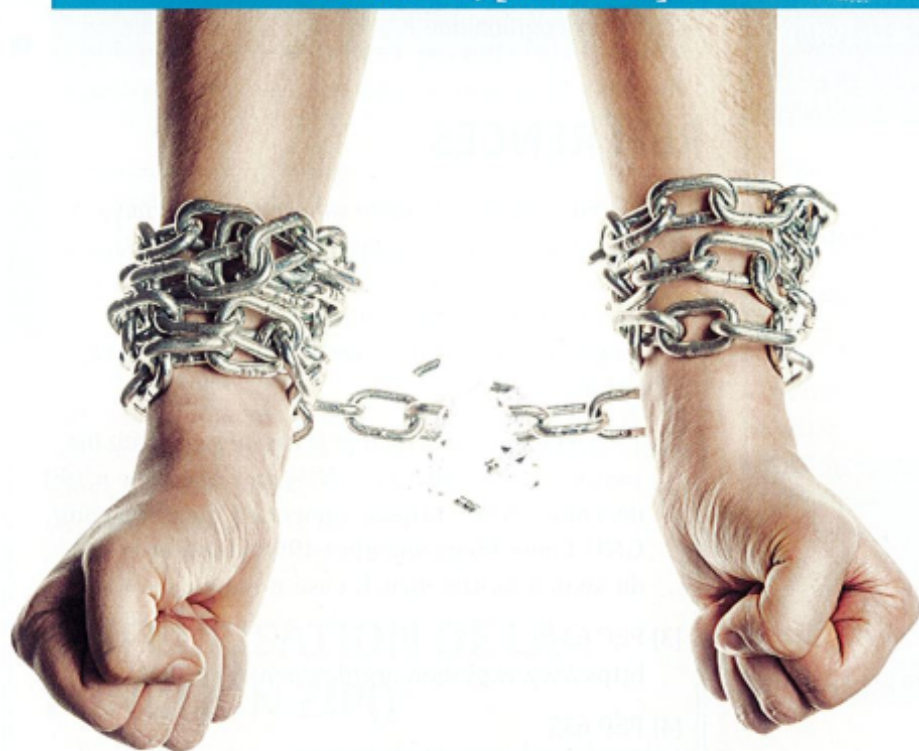
Copyright

# ¡ VIVA LA LIBERTAD !

ALBAN MANCHERON

[Enseignant-chercheur en bioinformatique à l'Université de Montpellier, linuxien depuis 1997 (convaincu depuis 1998)]

MOTS-CLÉS : LOGICIEL LIBRE, PROTECTION DES DONNÉES, RÉSEAUX SOCIAUX, [MATRIX]



Vous avez certainement entendu parler du changement des conditions d'utilisation de l'application WhatsApp et de la nécessité de se tourner vers une solution alternative. Je vous propose à travers ce faux problème de comprendre d'une part les stratégies techniques et technologiques sur lesquelles s'appuient les différents outils, mais également les modèles économiques sur lesquels ces outils sont basés. À l'issue de cet article, peut-être trouverez-vous l'application qui vous convient.

Les réseaux sociaux ! Voici un fléau auquel j'ai tenté d'échapper, mais auquel je commence à me résigner. Tout le monde en parle, tout le monde les utilise et si comme moi vous faites de la résistance, eh bien vous finissez par vous exclure de la communauté, du groupe, de « la vague » [1].

Depuis longtemps, mes enfants me sollicitaient pour pouvoir installer et utiliser, entre autres, **WhatsApp**, **Instagram** et **Discord**. Je m'y étais farouchement opposé. J'avais installé et testé **Jami** (que j'avais présenté dans [2]), mais celui-ci ne propose pas (pour le moment) de discussions de groupe et d'espaces collaboratifs. Alors j'ai cédé à mes enfants, puis j'ai oublié jusqu'à ce que j'entende à la radio le scandale lié aux nouvelles conditions d'utilisation de WhatsApp qui devaient prendre effet au 8 février 2021, ne laissant aux utilisateurs que deux solutions, accepter et conserver le service offert ou bien refuser et arrêter de l'utiliser.

## 1. LE SCANDALE DES « NOUVELLES » CONDITIONS GÉNÉRALES

Le journaliste radiophonique (ben oui, ça s'appelle comme ça) explique que ces nouvelles conditions octroieront le droit à l'application de transmettre les informations des usagers à la plateforme **Facebook**. Bien évidemment, c'est une honte et contre cette prise d'otage, il convient de changer d'application et de passer par exemple à **Signal** (<https://www.signal.org/>) ou encore à **Telegram** (<https://telegram.org/>).

L'argument de poids pour ces deux alternatives relève du professionnalisme du journaliste radiophonique qui se base sur les propos d'Edward Snowden (que tout le monde connaît grâce au film éponyme et qui est responsable des bouts de scotch et de patafix sur les webcams intégrées des ordinateurs de beaucoup de mes collègues... qui vont par ailleurs aux toilettes avec leur smartphone :( ). On retrouve par ailleurs cette information relayée par de nombreux médias, dont la notoriété et le sérieux ne sauraient être contestés.

Revenons à nos moutons conditions générales [3]. Je regarde le passage en question, et plus généralement la différence entre ces nouvelles conditions et les anciennes. Bon d'accord, ils ont bien écrit que l'application pouvait transmettre les informations collectées aux partenaires du groupe Facebook. Ce qui est amusant, c'est que WhatsApp appartient à ce groupe depuis 2014. Quelqu'un de mécréant pourrait dire que ce qui change, c'est donc qu'ils écrivent dans leurs conditions générales ce qu'ils font des données qu'ils collectent. Est-ce une raison pour changer d'application ? Non. Est-ce une raison pour avoir toujours refusé d'utiliser cette application ? Clairement oui ! Peut-on y voir un coup bas de concurrents pour dévoyer une partie de leurs utilisateurs vers un service plus ou moins équivalent ? Probablement.

Et sur ce dernier point, cela semble fonctionner puisque je suis tout à coup régulièrement sollicité pour savoir par quoi remplacer WhatsApp (ce qui est assez amusant vu que les gens qui me connaissent m'assimilent plutôt à un ours, à la rigueur un mi-ours/mi-sanglier). Le problème principal, c'est que la question est finalement assez mal formulée. En effet, WhatsApp représente à la fois une

application, un protocole de communication et une architecture réseau sous-jacente ; donc selon le point de vue que l'on observe, la question admet différentes réponses. Et c'est à ce moment-là que les choses se compliquent, car pour pouvoir répondre à cette question, il faut comprendre d'une part le fonctionnement d'Internet et d'autre part, les modèles économiques qui sous-tendent ces outils de communication.

## 2. LES COMMUNICATIONS SUR LE RÉSEAU INTERNET

L'objectif ici n'est pas de faire un cours d'histoire sur la création de ce réseau ni un cours de réseau présentant le modèle OSI à 7 couches (un vrai modèle d'incontinence), mais d'expliquer le vocabulaire et les architectures sur lesquelles s'appuient les services de communication (en clair, si vous connaissez déjà bien tout ça, rendez-vous directement à la prochaine section).

### 2.1 L'organisation du grand réseau

Commençons donc par « Internet ». Il s'agit tout simplement d'un réseau de réseaux informatiques permettant l'échange d'informations entre les ordinateurs qui le composent. Il n'y a pas de serveur (machine) central. Internet représente donc l'interconnexion directe et indirecte de machines. Il faut par conséquent que chaque machine ou réseau de machines dispose d'un identifiant spécifique donné dans une forme que tout le monde comprendra. On appellera cet identifiant l'adresse et cette adresse sera établie en suivant le protocole d'Internet (en anglais : *Internet Protocol*, dont l'acronyme est **IP**). La version 4 du protocole prévoit qu'une adresse est composée d'une suite de 4 octets (donc 4 entiers allant de 0 à 255), quant à la version 6, un jour peut-être... ou pas !

Au sein d'un réseau, chaque machine possède une adresse IP. Par exemple, dans un réseau domestique géré avec une box fournie par votre FAI préféré, la box est une machine du réseau, qui a donc également une IP au sein du réseau domestique, et qui généralement s'occupe d'attribuer les adresses IP pour une durée définie aux machines qu'elle gère. En règle général, les adresses IP fournies sont de la forme **192.168.0.x** ou **192.168.1.x**. Ces adresses ne sont valides qu'à l'intérieur du réseau domestique et donc

le protocole IP prévoit que les adresses commençant par **192.168** sont nécessairement des adresses réservées à des sous-réseaux (il ne peut pas y avoir de machine connectée directement à Internet dont l'adresse IP commencerait par **192.168**). On parle dans ce cas d'adresses locales au réseau. Cette box fait généralement également office de passerelle vers l'extérieur (donc c'est elle qui fait le relais avec Internet). Elle possède donc également une adresse IP visible depuis l'extérieur qui est fournie par votre FAI (et qui ne peut donc pas commencer par **192.168**). Ce mécanisme d'adressage va permettre de gérer deux types de communications : celles qui se déroulent entre deux machines d'un même réseau local et celles qui se déroulent entre deux machines de deux réseaux différents. En effet, au sein d'un même réseau, c'est assez facile. Il suffit à la machine **A** de crier qu'elle veut parler à la machine **B** (les autres machines n'ont qu'à se boucher les oreilles). Entre deux réseaux, il faut que la machine **A** du premier réseau informe la machine passerelle de ce réseau qu'elle veut communiquer avec une machine **B** d'un autre réseau. La machine passerelle va alors crier aux autres machines de son réseau public qu'elle veut communiquer avec la machine **B** et la passerelle va faire office de relais entre **A** et **B** (notez qu'on ne sait pas quelle machine se cache derrière **B**). Il faut donc qu'entre les machines, il y ait un protocole de communication commun, peu importe que l'information soit émise d'un ordinateur (**GNU/Linux**, **Mac**, autre), d'un smartphone, etc., peu importe que l'information soit transmise par un câble, des ondes, des signaux lumineux... C'est le protocole de contrôle de la transmission sur Internet (en anglais : *Transmission Control Protocol*, dont l'acronyme est **TCP**).

## 2.2 Le client, le serveur et le service

Que vous fassiez une recherche sur votre moteur de recherche préféré, une impression sur une imprimante en réseau, un partage de fichier, une visio... vous passez nécessairement par la combinaison TCP/IP. On peut imaginer que cette combinaison, c'est le fait de pouvoir mettre un contenu (lettre, revue, cadeau, vêtements, etc.) dans un contenant (enveloppe ou carton) et de l'envoyer d'un point à un autre. On comprend aisément que ce qui va changer, c'est le contenu des messages échangés. Et comme les messages sont des communications entre machines, il faut bien que chaque machine communique selon une même

convention, un même protocole pour un même service. Certains de ces protocoles sont standardisés, tels que les protocoles **HTTP**, **FTP**, **SMTP**... Il s'agit des spécifications (modalités) qui régissent l'échange d'informations, le dialogue entre applications. Il faut voir ces protocoles comme une convention, un rituel de communication tel que :

*[Le fils] - Bonjour Papa.*

*[Le père] - Bonjour mon fils.*

*[Le fils] - Dis papa, est-ce que tu veux bien me préparer mon petit déjeuner ?*

*[Le père] - Oui je m'en occupe. Attends 5 minutes.*

*[Le fils] - Papa, est-ce que c'est prêt ?*

*[Le père] - Non, attends encore 1 minute.*

*[Le fils] - Papa, est-ce que c'est prêt ?*

*[Le père] - Mon fils, ton petit déjeuner est prêt.*

*[Le fils] - Merci Papa.*

On retrouve un signal de demande de dialogue, un accusé de réception confirmant que la liaison est établie, une demande, une confirmation de la prise en charge de la demande, une information que la demande a été correctement traitée, une fin de communication. Dans cet exemple, le fils est ce que l'on appelle (en réseau) le **client**, tandis que le père est le **serveur** ; quant à l'association entre le client et le serveur, elle constitue le **service**.

Dans une communication réseau, le client (logiciel) est l'application (programme) qui va se charger de communiquer en se basant sur un protocole prédéfini. Par exemple, **Firefox** est un client qui utilise (entre autres) les protocoles **HTTP** et **HTTPS** pour échanger des informations. **Thunderbird** est un client qui utilise les protocoles **POP/POP3/IMAP/IMAPS** pour rapatrier/synchroniser des mails stockés sur un serveur et **SMTP/SMTPS** pour envoyer des mails.

Pour revenir sur les dénominations des protocoles, **HTTP** est l'acronyme de *Hyper Text Transfert Protocol*, **FTP** celui de *File Transfert Protocol*, **SMTP** correspond à *Simple Mail Transfert Protocol*, **IMAP** quant à lui correspond à *Internet Message Access Protocol* (eh oui, ils ne finissent pas tous en ...TP :D), **SSH** c'est *Secure SHell* (et des fois, il n'y a même pas le ...P de *Protocol*), **SSL** c'est *Secure Socket Layer* (qui permet de chiffrer/déchiffrer les données lors des communications), **HTTPS** c'est la combinaison de **HTTP** + **SSL**.

## 2.3 Protocoles ouverts... ou pas

J'ai écrit que les protocoles HTTP, FTP, SMTP... étaient standardisés et ce mot est important, car il signifie *de facto* que le protocole est publié et connu de tous. Sans être standardisé, il est possible de porter un protocole à la connaissance de tout le monde. Il est alors question de protocole ouvert. *A contrario*, un protocole fermé est un protocole qui est caché aux autres. Les protocoles peuvent également être propriétaires (c'est-à-dire qu'il est interdit de les exploiter sans l'accord de leur propriétaire). Les protocoles propriétaires sont souvent fermés et réciproquement (pour info, RMS préfère le terme « privateur » au terme « propriétaire », mais je ne suis pas vraiment d'accord, car la propriété est un droit alors que le terme privateur suggère une atteinte à la liberté).

Dans certains cas, les protocoles de communication ne sont pas standardisés (parce que pas connus, trop peu utilisés/trop spécifiques/moins faciles à exploiter/...). Parfois, c'est parce qu'ils ne sont pas ouverts. Pour ces protocoles fermés, ils nécessitent en général d'utiliser un client spécifique. C'est typiquement le cas de **Skype** ou de **WhatsApp**. En effet, ces clients utilisent leurs propres protocoles pour communiquer et il est donc généralement impossible d'utiliser une autre application (un autre client) pour échanger les informations.

Les avantages du protocole ouvert sont nombreux. L'ouverture permet d'avoir plusieurs applications basées sur ces protocoles (ceci entraîne généralement un peu de concurrence, ce qui a tendance à améliorer les services). Par exemple, **Firefox** vs **Chrome** vs **Internet Explorer** vs **Opera** vs **Safari**. La concurrence permet de faire évoluer rapidement les fonctionnalités (et souvent également les protocoles). Cela permet également, lorsque les protocoles présentent des vulnérabilités, de bénéficier d'une grande communauté pour proposer des correctifs.

Les avantages du protocole fermé (question de point de vue) sont notamment que chaque société peut imposer son outil et ainsi asservir les utilisateurs. Cela permet aussi de masquer les potentielles failles de sécurité (les failles existent, mais sont plus difficiles à déceler).

La notion de client ayant été abordée, voyons maintenant le concept de serveur. Le serveur est une application qui rend un service accessible via un protocole. Par exemple, un serveur *web* (également appelé serveur HTTP par abus



Chez votre  
marchand de journaux  
et sur [www.ed-diamond.com](http://www.ed-diamond.com)



en kiosque



sur [www.ed-diamond.com](http://www.ed-diamond.com)

**CONNECT**  
LA DOCUMENTATION TECHNIQUE DES PROS DE L'IT

sur [connect.ed-diamond.com](http://connect.ed-diamond.com)

de langage) sert les pages qu'il héberge lorsqu'un client les demande en utilisant le protocole HTTP ou HTTPS. Un serveur de mail réceptionne et envoie les mails de ses utilisateurs en utilisant les protocoles POP/POP3/POP3S/IMAP/IMAPS/SMTP/SMTPS (tous les serveurs ne proposent pas nécessairement tous les protocoles). Le serveur peut-être centralisé ou bien au contraire décentralisé. Un serveur centralisé est un programme qui s'exécute sur une machine (ou un groupe de machines apparaissant de l'extérieur comme s'il n'y en avait qu'une). Un serveur décentralisé est une application qui s'exécute sur plusieurs machines distribuées sur le réseau. Chaque instance du programme (il y a au moins une instance par machine physique) communique avec les autres.

## 2.4 Serveur centralisé vs distribué

Les avantages du serveur centralisé par rapport au décentralisé (ou distribué) : dans le cas où le service est fourni par une seule machine, il n'est pas besoin de gérer la synchronisation des données entre machines, il n'est pas non plus besoin de balancer la charge entre diverses machines et enfin, il n'est pas besoin de gérer l'ajout ou le retrait de machines. Dans le cas où le service est centralisé, mais géré en interne par plusieurs machines, cela facilite malgré tout l'administration du parc (qui est souvent relativement homogène).

Les avantages du serveur décentralisé par rapport au centralisé sont ceux que je viens de présenter comme des désavantages : la possibilité d'ajouter ou de retirer une machine facilement sans rupture de service, la redondance

d'informations et donc la résilience aux pannes et aux pertes d'informations. Bien évidemment, il faut que le service le permette, ce qui peut nécessiter un effort d'ingénierie non négligeable lors de la conception du service.

Dans certains cas, il y a une architecture hybride : par exemple, la gestion des noms de domaines (qui fait que quand on tape [duckduckgo.com](http://duckduckgo.com) dans la barre d'adresse, on tombe sur la machine « immatriculée » **40.114.177.156** sur Internet) utilise 13 serveurs centralisés qui communiquent entre eux un peu à la façon des serveurs décentralisés.

## 2.5 Petite synthèse

Maintenant que les bases sont posées, organisons des exemples de services (familles d'applications) dans un tableau :

	Serveur centralisé	Serveur décentralisé
Protocole ouvert	e-mails Signal	<b>Kadmelia</b> <b>eDonkey/eMule</b> (hybride) <b>BitTorrent</b> (jusqu'en 2007) <b>[matrix]</b>
Protocole fermé	WhatsApp Telegram	Skype Discord BitTorrent (depuis 2007)

C'est un petit peu le bazar en termes de nomenclature, car parfois il y a confusion entre les noms des serveurs, les noms des clients et les noms des services. Par exemple, Jami (<https://jami.net/>) est basé sur Kadmelia (qui est l'infrastructure, donc le serveur), de même **Element** (<https://element.io/>) est basé sur [matrix]. Skype ou WhatsApp reprennent à la fois les noms des clients, des serveurs et du service. Quant à e-mails, il y a tellement de FAI qui proposent un service de messagerie électronique que j'ai tout regroupé sous un nom générique.

Intuitivement, opter pour un protocole fermé, c'est clairement une question de confiance. On peut faire confiance à l'instant **t**, le problème c'est qu'il est difficile de revenir sur ses choix lorsque la relation de confiance évolue et que souvent, il est un peu tard. De même, opter pour un serveur centralisé, cela revient à mettre tous ses œufs (donc ses données) dans le même panier et encore une fois, dépend de la confiance que l'on accordera au propriétaire des infrastructures. Sur les serveurs décentralisés, comme par nature il n'est pas possible d'accorder sa confiance aux propriétaires des machines, la sécurisation des données (cryptographie) est généralement à la base du service.

### 3. LE FINANCEMENT DE LA GRATUITÉ

Vous connaissez certainement l'adage : « Si c'est gratuit, c'est que c'est vous le produit ». C'est plus ou moins vérifié. Une chose est certaine, c'est qu'un service développé par des professionnels (donc généralement qui perçoivent une rémunération), ça coûte vite un bras. S'il faut ajouter à cela de l'infrastructure matérielle, le second bras pourra vite être réclamé. Donc, la question de comprendre le modèle économique sous-jacent est importante. Comment est financé le service proposé ?

La solution la plus simple consiste à faire payer le service par l'octroi d'une licence d'utilisation ou un abonnement (c'est ce que font bon nombre de sociétés très implantées sur le marché de l'informatique, telles que **Microsoft, Adobe...**).

Cela se fait généralement lorsque les sociétés ont un avantage technique ou un quasi-monopole et donc les clients ont le choix entre rien, pas grand-chose ou une usine à gaz (en principe) performante. Parfois, il existe des alternatives libres et gratuites et tout aussi performantes, voire meilleures, mais bien souvent les utilisateurs ont du mal à accepter la transition (p. ex. **Windows** vs **GNU/Linux**, **PaintShop Pro** vs **Gimp**, etc.).

Certains produits sont proposés gratuitement à la communauté (**GNU/Linux**, **Gimp...**), car les porteurs des projets n'avaient pas nécessairement envisagé d'en faire un bénéfice au départ et vivent d'autres subsides (souvent en lien avec la notoriété acquise par leurs projets).

#### NOTE

Lorsque je tente d'expliquer mon problème éthique vis-à-vis des logiciels non vertueux (qui collectent et tirent profit des données personnelles), je suis confronté à deux types de réponses. Les premiers sont ceux qui m'expliquent qu'ils n'ont rien à cacher, et les seconds sont ceux qui s'étonnent de savoir quels profits il est possible de réaliser avec leurs données (et qui souvent m'expliquent qu'en réalité la collecte des données, c'est pour proposer un service plus adapté à leur besoin).

- Aux premiers, je raconte l'anecdote suivante :

Je suis enseignant à la fac et en début d'année, lors d'un cours magistral, j'ai demandé à mes étudiants qui avait un smartphone. La totalité des mains était levée. J'ai alors demandé qui avait installé l'application Tik Tok. J'ai eu quelques mains levées, dont une étudiante au premier rang. Je lui ai donc demandé si elle voulait bien me prêter son téléphone. Elle me l'a tendu et me demandant pourquoi faire exactement. Je lui ai répondu en toute honnêteté que je souhaitais regarder la liste de ses contacts, regarder son historique de navigation sur Internet, regarder les noms des fichiers qui sont stockés sur son mobile et bien évidemment regarder ses photos et vidéos prises de son téléphone. Elle s'est soudainement ravisée et à retiré son bras pour que je n'attrape pas son portable. Je lui ai alors exprimé ma profonde déception de me voir refuser les droits qu'elle avait accordés à une société basée en Chine, alors que je lui offre un service qui me semble bien plus important que la possibilité de partager des vidéos : **l'instruction**.

- Aux seconds, je leur donne un exemple simple :

Si je sais que tels utilisateurs aiment les sodas au Cola, et qu'ils ont tendance à faire une pause à 16 h 30, eh bien je peux leur rendre un super service en leur envoyant une publicité voire un bon de réduction à 16 h 25 pour mon partenaire industriel (qui me rémunère pour cela). Comme je sais également que consommer des sodas, ce n'est pas bon pour la santé et que régulièrement les utilisateurs en question consultent des vidéos de gym à la maison et des sites pour perdre 2 kg sans effort, je peux également leur proposer des formules d'essai ou des abonnements à prix réduit dans des clubs de fitness (proposés par des partenaires qui me rémunèrent aussi pour cela). Il n'est pas difficile de savoir qui est le client et qui est le produit, et surtout que le service n'est pas rendu au produit, mais bel et bien au client. Cela fait un peu penser à **Cypher** dans le film **Matrix** qui explique « je sais que ce steak n'existe pas, je sais que lorsque je le mets dans ma bouche, c'est la Matrice qui dit à mon esprit que ce steak est saignant et délicieux... Au bout de neuf ans, vous savez ce que j'ai compris ? Les ignorants sont bénis ! ».

Pour conclure, et parce que personne n'a jamais rien à cacher, si vous estimez que votre conjoint, vos parents, vos enfants, vos amis, vos voisins, vos collègues peuvent accéder aux données contenues dans vos appareils, alors effectivement, il n'y a pas de problème. Si au contraire cela vous gêne, alors protégez vos données et ne les partagez pas pour un peu de commodité.

Certains services sont proposés gratuitement pour un usage privé et des services adaptés aux collectivités et aux entreprises sont proposés contre licence, mécénat ou même support matériel et humain.

Pour financer leurs services, certains proposent parfois de les entrecouper de publicités pour être rémunérés.

Enfin, il y a les services qui sont proposés gratuitement, mais qui exploitent les informations que les utilisateurs partagent (plus ou moins consciemment) pour générer du bénéfice.

Je ne suis pas exhaustif (la preuve : je n'ai pas mentionné les logiciels payants qui exploitent vos données en prime), mais l'idée générale, c'est que hormis les produits de la communauté du libre (qui tournent autour de notre OS préféré, le seul, le vrai, le grand, j'ai nommé GNU/Linux !), le reste se paye.

C'est pourquoi depuis quelque temps, en plus de faire la promotion des logiciels libres, je distingue également les logiciels que je qualifie de vertueux (c'est-à-dire qui ne se servent pas d'un service avec comme principal objectif de générer des profits).

Vous comprendrez que WhatsApp, même avant ses nouvelles conditions générales, n'est pas un service vertueux (au passage, je vous invite à lire les conditions générales de **Tik Tok**, c'est un véritable régal ; ils ont écrit ce que même les GAFAM n'avaient pas osé jusqu'alors).

## CONCLUSION

Si je devais choisir une application offrant des services similaires à WhatsApp ou Discord, je choisirais dans les outils vertueux dont je contrôle l'hébergement ou bien sur une architecture décentralisée, utilisant un protocole ouvert, et éventuellement payant si j'estime que le service en vaut la peine. Cela pourrait être de passer par **Mattermost** (orienté entreprise, <https://mattermost.com/>) ou les clients basés sur [matrix] (tel que Element). J'exclurai donc *de facto* les services comme Telegram dont le serveur est opaque et hébergé dans un pays où j'ai modérément confiance dans le respect du RGPD (c'est certes subjectif, mais c'est pour ce même genre de raisons que je ne fais pas de saut à l'élastique : il n'y a peut-être aucun risque, mais je n'ai pas confiance) ou Signal (dont le serveur est hébergé aux États-Unis d'Amérique, où la juridiction fédérale est visiblement peu concernée par le respect de notre beau RGPD européen ; toujours le principe du saut à l'élastique). ■

## RÉFÉRENCES

- [1] T. STRASSER, « *La Vague* », 1981.
- [2] A. MANCHERON, « *Communiquer en sécurité avec Jami* », *Linux Pratique* n° 120, juillet/août 2020 : <https://connect.ed-diamond.com/Linux-Pratique/lp-120/communiquer-en-securite-avec-jami>
- [3] Conditions générales du service *WhatsApp* (en français) : <https://www.whatsapp.com/legal/?lang=fr>

## POUR ALLER PLUS LOIN

J'ai testé plusieurs clients basés sur [matrix] : Element (pour Android, pour iOS et pour GNU/Linux, je n'ai pas testé les versions macOS et Windows), son *fork* SchildiChat (pour GNU/Linux, <https://schildi.chat/>) et Fluffy Chat (pour iOS, je n'ai pas testé les versions Android et GNU/Linux, <https://fluffychat.im/>). Ça semble plutôt bien fonctionner. Les autres clients que j'ai testés n'ont pas bien fonctionné ou offraient des fonctionnalités trop restreintes, donc je ne vais pas les accabler ici... Le client Element (et donc SchildiChat) semble offrir globalement plus de fonctionnalités, mais j'en ai trouvé certaines bien sympathiques avec Fluffy Chat.

J'utilise également Jami qui fonctionne globalement, même si la qualité des appels vidéo est encore très incertaine. Et puis j'ai un réseau social riche de moins de 10 personnes, dont la moitié vivent sous mon toit, donc cela relève plus de l'expérimentation que de satisfaire à un besoin.