# C’est quoi, l’interopérabilité, et pourquoi est-ce beau et bien ?

*Protocole, HTTP, interopérabilité, ça vous parle ? Et normes, spécifications, RFC, ça va toujours ? Si vous avez besoin d’y voir un peu plus clair, l’article ci-dessous est un morceau de choix rédigé par Stéphane Bortzmeyer qui s’est efforcé de rendre accessibles ces notions fondamentales.*

---

## Protocoles

Le 21 mai 2019, soixante-neuf organisations, dont Framasoft, ont signé un appel à ce que soit imposé, éventuellement par la loi, un minimum d’**interopérabilité** pour les gros acteurs commerciaux du Web.

« Interopérabilité » est un joli mot, mais qui ne fait pas forcément partie du vocabulaire de tout le monde, et qui mérite donc d’être expliqué. On va donc parler d’interopérabilité, de protocoles, d’interfaces, de normes, et j’espère réussir à le faire tout en restant compréhensible (si vous êtes informaticien·ne professionnel·le, vous savez déjà tout cela ; mais l’appel des 69 organisations concerne tout le monde).

Le Web, ou en fait tout l’Internet, repose sur des **protocoles** de communication. Un protocole, c’est un ensemble de règles qu’il faut suivre si on veut communiquer. Le terme vient de la communication humaine, par exemple, lorsqu’on rencontre quelqu’un, on se serre la main, ou bien on se présente si

l’autre ne vous connaît pas, etc. Chez les humains, le protocole n’est pas rigide (sauf en cas de réception par la reine d’Angleterre dans son palais, mais cela doit être rare chez les lectrices et lecteurs du Framablog). Si la personne avec qui vous communiquez ne respecte pas exactement le protocole, la communication peut tout de même avoir lieu, quitte à se dire que cette personne est bien impolie. Mais les logiciels ne fonctionnent pas comme des humains. Contrairement aux humains, ils n’ont pas de souplesse, les règles doivent être suivies exactement. Sur un réseau comme l’Internet, pour que deux logiciels puissent communiquer, chacun doit donc suivre exactement les mêmes règles, et c’est l’ensemble de ces règles qui fait un protocole.

Un exemple concret ? Sur le Web, pour que votre navigateur puisse afficher la page web désirée, il doit demander à un serveur web un ou plusieurs fichiers. La demande se fait obligatoirement en envoyant au serveur le mot GET (« donne », en anglais) suivi du nom du fichier, suivi du mot « HTTP/1.1 ». Si un navigateur web s’avisait d’envoyer le nom du fichier avant le mot GET, le serveur ne comprendrait rien, et renverrait plutôt un message d’erreur. En parlant d’erreurs, vous avez peut-être déjà rencontré le nombre 404 qui est simplement le code d’erreur qu’utilisent les logiciels qui parlent HTTP pour signaler que la page demandée n’existe pas. Ces codes numériques, conçus pour être utilisés entre logiciels, ont l’avantage sur les textes de ne pas être ambigus, et de ne pas dépendre d’une langue humaine particulière. Cet exemple décrit une toute petite partie du protocole nommé HTTP (pour *Hypertext Transfer Protocol*) qui est le plus utilisé sur le Web.

Il existe des protocoles bien plus complexes. Le point important est que, derrière votre écran, les logiciels communiquent entre eux en utilisant ces protocoles. Certains servent directement aux logiciels que vous utilisez (comme HTTP, qui permet à votre navigateur Web de communiquer avec le

serveur qui détient les pages désirées), d'autres protocoles relèvent de l'infrastructure logicielle de l'Internet ; vos logiciels n'interagissent pas directement avec eux, mais ils sont indispensables.

Le protocole, ces règles de communication, sont indispensables dans un réseau comme l'Internet. Sans protocole, deux logiciels ne pourraient tout simplement pas communiquer, même si les câbles sont bien en place et les machines allumées. Sans protocole, les logiciels seraient dans la situation de deux humains, un Français ne parlant que français, et un Japonais ne parlant que japonais. Même si chacun a un téléphone et connaît le numéro de l'autre, aucune vraie communication ne pourra prendre place. Tout l'Internet repose donc sur cette notion de protocole.

Le protocole permet l'**interopérabilité**. L'interopérabilité est la capacité à communiquer de deux logiciels différents, issus d'équipes de développement différentes. Si une université bolivienne peut échanger avec une entreprise indienne, c'est parce que toutes les deux utilisent des protocoles **communs**.

Un exemple classique d'interopérabilité : la prise électrique. Kae [CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/

3.0)], via Wikimedia Commons

Seuls les protocoles ont besoin d'être communs : l'Internet n'oblige pas à utiliser les mêmes logiciels, ni à ce que les logiciels aient la même interface avec l'utilisateur. Si je prends l'exemple de deux logiciels qui parlent le protocole HTTP, le navigateur Mozilla Firefox (que vous êtes peut-être en train d'utiliser pour lire cet article) et le programme curl (utilisé surtout par les informaticiens pour des opérations techniques), ces deux logiciels ont des usages très différents, des interfaces avec l'utilisateur reposant sur des principes opposés, mais tous les deux parlent le même protocole HTTP. Le protocole, c'est ce qu'on parle avec les autres logiciels (l'interface avec l'utilisateur étant, elle, pour les humain·e·s.).

La distinction entre protocole et logiciel est cruciale. Si j'utilise le logiciel A parce que je le préfère et vous le logiciel B, tant que les deux logiciels parlent le même protocole, aucun problème, ce sera juste un choix individuel. Malgré leurs différences, notamment d'interface utilisateur, les deux logiciels pourront communiquer. Si, en revanche, chaque logiciel vient avec son propre protocole, il n'y aura pas de communication, comme dans l'exemple du Français et du Japonais plus haut.

# Babel

Alors, est-ce que tous les logiciels utilisent des protocoles communs, permettant à tout le monde de communiquer avec bonheur ? Non, et ce n'est d'ailleurs pas obligatoire. L'Internet est un réseau à « permission facultative ». Contrairement aux anciennes tentatives de réseaux informatiques qui étaient contrôlés par les opérateurs téléphoniques, et qui décidaient de quels protocoles et quelles applications tourneraient sur leurs réseaux, sur

l’Internet, vous pouvez inventer votre propre protocole, écrire les logiciels qui le parlent et les diffuser en espérant avoir du succès. C’est d’ailleurs ainsi qu’a été inventé le Web : Tim Berners-Lee (et Robert Cailliau) n’ont pas eu à demander la permission de qui que ce soit. Ils ont défini le protocole HTTP, ont écrit les applications et leur invention a connu le succès que l’on sait.

Cette liberté d’innovation sans permission est donc une bonne chose. Mais elle a aussi des inconvénients. Si chaque développeur ou développeuse d’applications invente son propre protocole, il n’y aura plus de communication ou, plus précisément, il n’y aura plus d’interopérabilité. Chaque utilisatrice et chaque utilisateur ne pourra plus communiquer qu’avec les gens ayant choisi le même logiciel. Certains services sur l’Internet bénéficient d’une bonne interopérabilité, le courrier électronique, par exemple. D’autres sont au contraire composés d’un ensemble de **silos** fermés, ne communiquant pas entre eux. C’est par exemple le cas des messageries instantanées. Chaque application a son propre protocole, les personnes utilisant WhatsApp ne peuvent pas échanger avec celles utilisant Telegram, qui ne peuvent pas communiquer avec celles qui préfèrent Signal ou Riot. Alors que l’Internet était conçu pour faciliter la communication, ces silos enferment au contraire leurs utilisateurs et utilisatrices dans un espace clos.

La situation est la même pour les réseaux sociaux commerciaux comme Facebook. Vous ne pouvez communiquer qu’avec les gens qui sont eux-mêmes sur Facebook. Les pratiques de la société qui gère ce réseau sont déplorables, par exemple en matière de captation et d’utilisation des données personnelles mais, quand on suggère aux personnes qui utilisent Facebook de quitter ce silo, la réponse la plus courante est « je ne peux pas, tou·te·s mes ami·e·s y sont, et je ne pourrais plus communiquer avec eux et elles si je partais ». Cet exemple illustre très bien les dangers des protocoles liés à une

entreprise et, au contraire, l'importance de l'interopérabilité.

« La tour de Babel  », tableau de Pieter Bruegel l'ancien. Domaine public (Google Art Project)

Mais pourquoi existe-t-il plusieurs protocoles pour un même service ? Il y a différentes raisons. Certaines sont d'ordre technique. Je ne les développerai pas ici, ce n'est pas un article technique, mais les protocoles ne sont pas tous équivalents, il y a des raisons techniques objectives qui peuvent faire choisir un protocole plutôt qu'un autre. Et puis deux personnes différentes peuvent estimer qu'en fait deux services ne sont pas réellement identiques et méritent donc des protocoles séparés, même si tout le monde n'est pas d'accord.

Mais il peut aussi y avoir des raisons commerciales : l'entreprise en position dominante n'a aucune envie que des acteurs plus petits la concurrencent, et ne souhaite pas permettre à des nouveaux entrants d'arriver. Elle a donc une

forte motivation à n'utiliser qu'un protocole qui lui est propre, que personne d'autre ne connaît.

Enfin, il peut aussi y avoir des raisons plus psychologiques, comme la conviction chez l·e·a créat·eur·rice d'un protocole que son protocole est bien meilleur que les autres.

Un exemple d'un succès récent en termes d'adoption d'un nouveau protocole est donné par le **fédivers**. Ce terme, contraction de « fédération » et « univers » (et parfois écrit « fédiverse » par anglicisme) regroupe tous les serveurs qui échangent entre eux par le protocole ActivityPub, que l'appel des soixante-neuf organisations mentionne comme exemple. ActivityPub permet d'échanger des messages très divers. Les logiciels Mastodon et Pleroma se servent d'ActivityPub pour envoyer de courts textes, ce qu'on nomme du micro-blogging (ce que fait Twitter). PeerTube utilise ActivityPub pour permettre de voir les nouvelles vidéos et les commenter. WriteFreely fait de même avec les textes que ce logiciel de blog permet de rédiger et diffuser. Et, demain, Mobilizon utilisera ActivityPub pour les informations sur les événements qu'il permettra d'organiser. Il s'agit d'un nouvel exemple de la distinction entre protocole et logiciel. Bien que beaucoup de gens appellent le fédivers « Mastodon », c'est inexact. Mastodon n'est qu'*un des logiciels* qui permettent l'accès au fédivers.

Le terme d'ActivityPub n'est d'ailleurs pas idéal. Il y a en fait un ensemble de protocoles qui sont nécessaires pour communiquer au sein du fédivers. ActivityPub n'est que l'un d'entre eux, mais il a un peu donné son nom à l'ensemble.

Tous les logiciels de la mouvance des « réseaux sociaux décentralisés » n'utilisent pas ActivityPub. Par exemple, Diaspora ne s'en sert pas et n'est donc pas interopérable avec les autres.

# Appel

Revenons maintenant l'appel cité au début, Que demande-t-il ? Cet appel réclame que l'interopérabilité soit imposée aux GAFA, ces grosses entreprises capitalistes qui sont en position dominante dans la communication. Tous sont des silos fermés. Aucun moyen de commenter une vidéo YouTube si on a un compte PeerTube, de suivre les messages sur Twitter ou Facebook si on est sur le fédivers. Ces GAFA ne changeront pas spontanément : il faudra les y forcer.

Il ne s'agit que de la communication externe. Cet appel est modéré dans le sens où il ne demande pas aux GAFA de changer leur interface utilisateur, ni leur organisation interne, ni leurs algorithmes de sélection des messages, ni leurs pratiques en matière de gestion des données personnelles. Il s'agit uniquement d'obtenir qu'ils permettent l'interopérabilité avec des services concurrents, de façon à permettre une réelle liberté de choix par les utilisateurs. Un tel ajout est simple à implémenter pour ces entreprises commerciales, qui disposent de fonds abondants et de nombreu·ses-x programmeur·e·s compétent·e·s. Et il « ouvrirait » le champ des possibles. Il s'agit donc de défendre les intérêts des utilisateurs et utilisatrices. (Alors que le gouvernement, dans ses commentaires, n'a cité que les intérêts des GAFA, comme si ceux-ci étaient des espèces menacées qu'il fallait défendre.)

# Qui commande ?

Mais au fait, qui décide des protocoles, qui les crée ? Il n'y a pas de réponse simple à cette question. Il existe plein de protocoles différents et leurs origines sont variées. Parfois, ils sont rédigés, dans un texte qui décrit exactement ce que doivent faire les deux parties. C'est ce que l'on nomme une **spécification**. Mais parfois il n'y a pas vraiment de spécification, juste quelques vagues idées et un programme qui

utilise ce protocole. Ainsi, le protocole BitTorrent, très utilisé pour l'échange de fichiers, et pour lequel il existe une très bonne interopérabilité, avec de nombreux logiciels, n'a pas fait l'objet d'une spécification complète. Rien n'y oblige développeurs et développeuses : l'Internet est « à permission facultative ». Dans de tels cas, celles et ceux qui voudraient créer un programme interopérable devront lire le code source (les instructions écrites par le ou la programmeur·e) ou analyser le trafic qui circule, pour essayer d'en déduire en quoi consiste le protocole (ce qu'on nomme la rétro-ingénierie). C'est évidemment plus long et plus difficile et il est donc très souhaitable, pour l'interopérabilité, qu'il existe une spécification écrite et correcte (il s'agit d'un exercice difficile, ce qui explique que certains protocoles n'en disposent pas).

Parfois, la spécification est adoptée formellement par un organisme dont le rôle est de développer et d'approuver des spécifications. C'est ce qu'on nomme la **normalisation**. Une spécification ainsi approuvée est une **norme**. L'intérêt d'une norme par rapport à une spécification ordinaire est qu'elle reflète a priori un consensus assez large d'une partie des acteurs, ce n'est plus un acte unilatéral. Les normes sont donc une bonne chose mais, rien n'étant parfait, leur développement est parfois laborieux et lent.

Écrire des normes correctes et consensuelles peut être laborieux. Codex Bodmer – Frater Rufillus (wohl tätig im Weißenauer Skriptorium) [Public domain]

Toutes les normes ne se valent pas. Certaines sont publiquement disponibles (comme les normes importantes de l'infrastructure de l'Internet, les RFC – Request For Comments), d'autres réservées à ceux qui paient, ou à ceux qui sont membres d'un club fermé. Certaines normes sont développées de manière publique, où tout le monde a accès aux informations, d'autres sont créées derrière des portes soigneusement closes. Lorsque la norme est développée par une organisation ouverte à tous et toutes, selon des procédures publiques, et que le résultat est publiquement disponible, on parle souvent de normes ouvertes. Et, bien sûr, ces normes ouvertes sont préférables pour l'interopérabilité.

L’une des organisations de normalisation ouverte les plus connues est l’IETF (*Internet Engineering Task Force*, qui produit notamment la majorité des RFC). L’IETF a développé et gère la norme décrivant le protocole HTTP, le premier cité dans cet article. Mais d’autres organisations de normalisation existent comme le W3C (*World-Wide Web Consortium*) qui est notamment responsable de la norme ActivityPub.

Par exemple, pour le cas des messageries instantanées que j’avais citées, il y a bien une norme, portant le doux nom de XMPP (*Extensible Messaging and Presence Protocol*). Google l’utilisait, puis l’a abandonnée, jouant plutôt le jeu de la fermeture.

# Difficultés

L’interopérabilité n’est évidemment pas une solution magique à tous les problèmes. On l’a dit, l’appel des soixante-neuf organisations est très modéré puisqu’il demande seulement une ouverture à des tiers. Si cette demande se traduisait par une loi obligeant à cette interopérabilité, tout ne serait pas résolu.

D’abord, il existe beaucoup de moyens pour respecter la lettre d’un protocole tout en violant son esprit. On le voit pour le courrier électronique où Gmail, en position dominante, impose régulièrement de nouvelles exigences aux serveurs de messagerie avec lesquels il daigne communiquer. Le courrier électronique repose, contrairement à la messagerie instantanée, sur des normes ouvertes, mais on peut respecter ces normes tout en ajoutant des règles. Ce bras de fer vise à empêcher les serveurs indépendants de communiquer avec Gmail. Si une loi suivant les préconisations de l’appel était adoptée, nul doute que les GAFA tenteraient ce genre de jeu, et qu’il faudrait un mécanisme de suivi de l’application de la loi.

Plus subtil, l’entreprise qui voudrait « tricher » avec les

obligations d'interopérabilité peut aussi prétendre vouloir « améliorer » le protocole. On ajoute deux ou trois choses qui n'étaient pas dans la norme et on exerce alors une pression sur les autres organisations pour qu'elles aussi ajoutent ces fonctions. C'est un exercice que les navigateurs web ont beaucoup pratiqué, pour réduire la concurrence.

Jouer avec les normes est d'autant plus facile que certaines normes sont mal écrites, laissant trop de choses dans le vague (et c'est justement le cas d'ActivityPub). Écrire une norme est un exercice difficile. Si on laisse beaucoup de choix aux programmeuses et programmeurs qui créeront les logiciels, il y a des risques de casser l'interopérabilité, suite à des choix trop différents. Mais si on contraint ces programmeuses et programmeurs, en imposant des règles très précises pour tous les détails, on empêche les logiciels d'évoluer en réponse aux changements de l'Internet ou des usages. La normalisation reste donc un art difficile, pour lequel on n'a pas de méthode parfaite.

# Conclusion

Voilà, désolé d'avoir été long, mais les concepts de protocole et d'interopérabilité sont peu enseignés, alors qu'ils sont cruciaux pour le fonctionnement de l'Internet et surtout pour la liberté des citoyen·ne·s qui l'utilisent. J'espère les avoir expliqués clairement, et vous avoir convaincu·e de l'importance de l'interopérabilité. Pensez à soutenir l'appel des soixante-neuf organisations !

# Après

Et si vous voulez d'autres informations sur ce sujet, il y a :

- Si vous aimez le son plutôt que l'écrit, le podcast de l'APRIL a parlé de l'interopérabilité.