

Le Web est-il devenu trop compliqué ?

Le Web, tout le monde s'en sert et beaucoup en sont très contents. Mais, même parmi ceux et celles qui sont ravis de l'utiliser, il y a souvent des critiques. Elles portent sur de nombreux aspects et je ne vais pas essayer de lister ici toutes ces critiques. Je vais parler d'un problème souvent ressenti : le Web n'est-il pas devenu trop compliqué ?



Je ne parle pas de la complexité pour l'utilisateur, par exemple des problèmes qu'il ou elle peut avoir avec telle ou telle application Web, ou tel formulaire incompréhensible ou excluante. Non, je parle de la complexité des nombreuses technologies sous-jacentes. Alors, si vous n'êtes pas technicien·ne, vous avez peut-être envie d'arrêter votre lecture ici en pensant qu'on ne parlera que de technique. Mais ce n'est pas le cas, cet article est pour tous et toutes. (Ceci dit, si vous arrêtez votre lecture pour jouer avec le chat, manger un bon plat, lire un livre passionnant ou faire des câlins à la personne appropriée, cela ne me dérange pas et je vous souhaite un agréable moment.)

Mais revenons à l'objection « OK, les techniques utilisées dans le Web sont compliquées mais cela ne concerne que les développeuses et développeurs, non ? » Eh bien non car cette complication a des conséquences pour tous et toutes. Elle se traduit par des logiciels beaucoup plus complexes, donc elle réduit la concurrence, très peu d'organisation pouvant aujourd'hui développer un navigateur Web. Elle a pour conséquence de rendre l'utilisation du Web plus lente : bien que les machines et les réseaux aient nettement gagné en performance, le temps d'affichage d'une page ne cesse d'augmenter. Passer à la fibre ou à la 5G ne se traduira pas forcément par un gain de temps, puisque ce sont souvent les calculs nécessaires à l'affichage qui ralentissent la navigation. Et enfin cette complication augmente l'empreinte environnementale du Web, en imposant davantage d'opérations aux machines, ce qui pousse au remplacement plus rapide des terminaux.

L'insoutenable lourdeur du Web

Une page Web d'aujourd'hui n'est en effet pas une simple description d'un contenu. Elle inclut la « feuille de style », rédigée dans le langage CSS, qui va indiquer comment présenter la page, du JavaScript, un langage de programmation qui va être exécuté pour faire varier le contenu de la page, des vidéos, et d'autres choses qui souvent distraient du contenu lui-même. Je précise que je ne parle pas ici des applications tournant sur le Web (comme une application d'accès au courrier électronique, ou une application de gestion des événements ou l'application maison utilisée par les employés d'une organisation pour gérer leur travail), non, je parle des pages Web de contenu, qui ne devraient pas avoir besoin de toute cette artillerie.

Du fait de cette complexité, il n'existe aujourd'hui que quatre ou cinq navigateurs Web réellement distincts. Écrire un navigateur Web aujourd'hui est une tâche colossale, hors de

portée de la très grande majorité des organisations. La concurrence a diminué sérieusement. La complexité technique a donc des conséquences stratégiques pour le Web. Et ceci d'autant plus qu'il n'existe derrière ces navigateurs que deux moteurs de rendu, le cœur du navigateur, la partie qui interprète le langage HTML et le CSS et dessine la page. Chrome, Edge et Safari utilisent le même moteur de rendu, WebKit (ou l'une de ses variantes).

Et encore tout ne tourne pas sur votre machine. Derrière votre écran, l'affichage de la moindre page Web va déclencher d'innombrables opérations sur des machines que vous ne voyez pas, comme les calculs des entreprises publicitaires qui vont, en temps réel, déterminer les « meilleures » publicités à vous envoyer dans la figure ou comme l'activité de traçage des utilisateurs, notant en permanence ce qu'ils font, d'où elles viennent et de nombreuses autres informations, dont beaucoup sont envoyées automatiquement par votre navigateur Web, qui travaille au moins autant pour l'industrie publicitaire que pour vous. Pas étonnant que la consommation énergétique du numérique soit si importante. Et ces calculs côté serveur ont une grande influence sur la capacité du serveur à tenir face à une charge élevée, comme on l'a vu pendant les confinements Covid-19. Les sites Web de l'Éducation Nationale ne tenaient pas le coup, même quand il s'agissait uniquement de servir du contenu statique.

La surveillance coûte cher

La complexité du Web cache en effet également cette activité de surveillance, pratiquée aussi bien par les entreprises privées que par les États. Autrefois, acheter un journal à un kiosque et le lire étaient des activités largement privées. Aujourd'hui, toute activité sur le Web est enregistrée et sert à nourrir les bases de données du monde de la publicité, ou les fichiers des États. Comme exemple des informations envoyées par votre navigateur, sans que vous en ayez

clairement connaissance, on peut citer bien sûr les fameux cookies. Ce sont des petits fichiers choisis par le site Web et envoyés à votre navigateur. Celui-ci les stockera et, lors d'une visite ultérieure au même site Web, renverra le cookie. C'est donc un outil puissant de suivi de l'utilisateur. Et ne croyez pas que, si vous visitez un site Web, seule l'organisation derrière ce site pourra vous pister. La plupart des pages Web incluent en effet des ressources extérieures (images, vidéos, boutons de partage), pas forcément chargés depuis le site Web que vous visitez et qui ont eux aussi leurs cookies. La loi Informatique et Libertés (et, aujourd'hui, le RGPD) impose depuis longtemps que les utilisateurs soient prévenus de ce pistage et puissent s'y opposer, mais il a fallu très longtemps pour que la CNIL tape sur la table et impose réellement cette information des utilisateurs, le « bandeau cookies ». Notez qu'il n'est pas obligatoire. D'abord, si le site Web ne piste pas les utilisateurs, il n'y a pas d'obligation d'un tel bandeau, ensuite, même en cas de pistage, de nombreuses exceptions sont prévues.

Le Monde et des tiers sélectionnés, notamment des [partenaires publicitaires](#), utilisent des cookies ou des technologies similaires. Les cookies nous permettent d'accéder à, d'analyser et de stocker des informations telles que les caractéristiques de votre terminal ainsi que certaines données personnelles (par exemple : adresses IP, données de navigation, d'utilisation ou de géolocalisation, identifiants uniques).

Ces données sont traitées aux fins suivantes : analyse et amélioration de l'expérience utilisateur et/ou de notre offre de contenus, produits et services, mesure et analyse d'audience, interaction avec les réseaux sociaux, affichage de publicités et contenus personnalisés, mesure de performance et d'attractivité des publicités et du contenu.

Pour plus d'information, consulter notre [politique de confidentialité](#).

Vous pouvez librement donner, refuser ou retirer votre consentement à tout moment en accédant à notre outil de [paramétrage des cookies](#) et/ou, en ce qui concerne la publicité, au [panneau des préférences publicitaires](#). Si vous ne consentez pas à l'utilisation de ces technologies, nous considérerons que vous vous opposez également à tout dépôt de cookie fondé sur un intérêt légitime.

Vous pouvez consentir à l'utilisation de ces technologies en cliquant sur « accepter »

Accepter

Paramétrer les cookies

Un bandeau cookies. Notez qu'il n'y a pas de bouton Refuser.

Les bandeaux cookies sont en général délibérément conçus pour qu'il soit difficile de refuser. Le but est que l'utilisateur clique rapidement sur « Accepter » pour en être débarrassé,

permettant ainsi à l'entreprise qui gère le site Web de prétendre qu'il y a eu consentement.

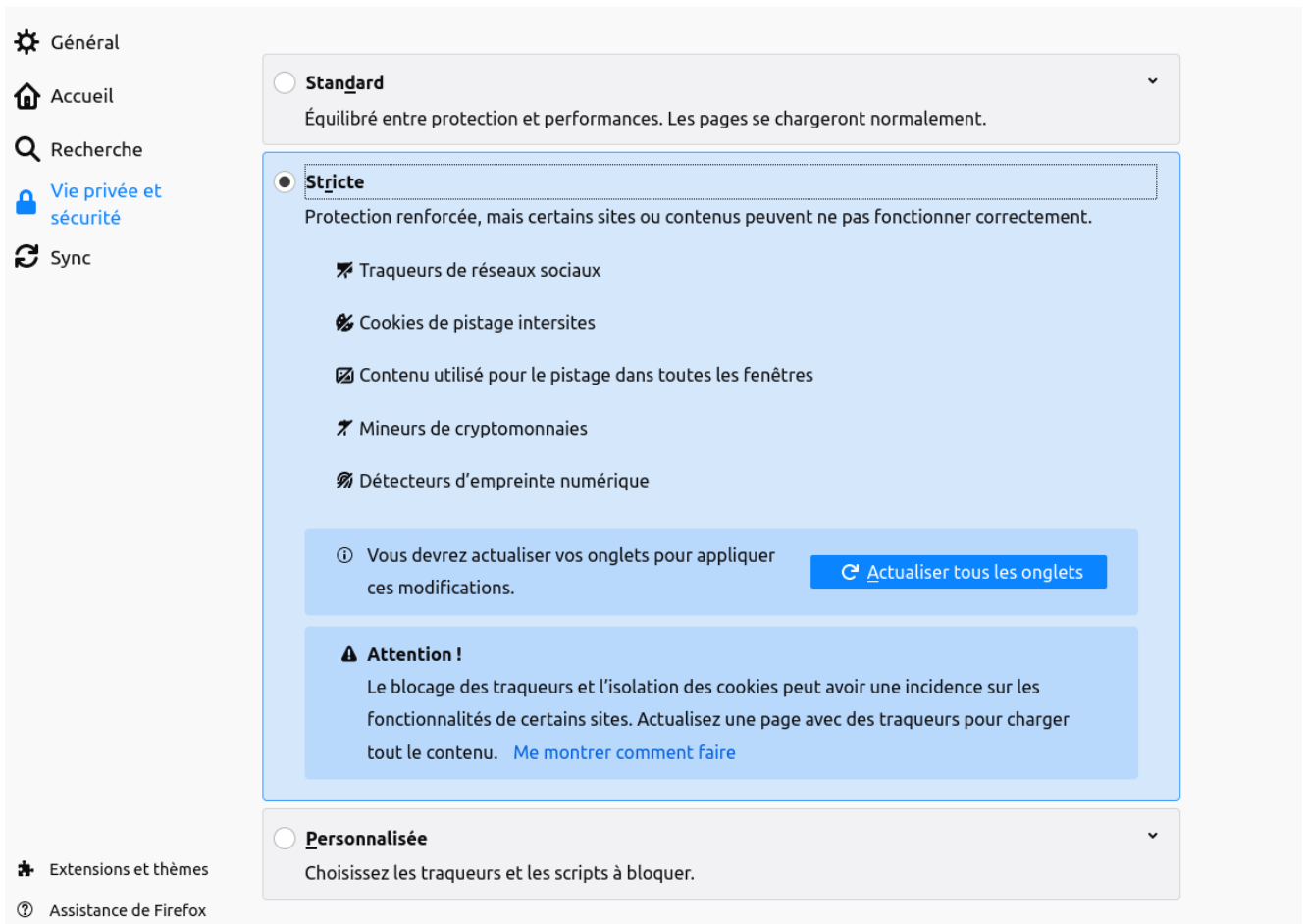
Désolé de la longueur de ce préambule, d'autant plus qu'il est très possible que, en tant que lectrice ou lecteur du Framablog, vous soyez déjà au courant. Mais il était nécessaire de revenir sur ces problèmes du Web pour mieux comprendre les projets qui visent à corriger le tir. Notez que les évolutions néfastes du Web ne sont pas qu'un problème technique. Elles sont dues à des raisons économiques et politiques et donc aucune approche purement technique ne va résoudre complètement le problème. Cela ne signifie pas que les techniciens et techniciennes doivent rester les bras croisés. Ils et elles peuvent apporter des solutions partielles au problème.



Bloquer les saletés

Première approche possible vers un Web plus léger, tenter de bloquer les services néfastes. Tout bon navigateur Web permet ainsi un certain contrôle de l'usage des cookies. C'est par

exemple ce que propose Firefox dans une rubrique justement nommée « Vie privée et sécurité ».



Le menu de Firefox pour contrôler notamment les cookies

On peut ainsi bloquer une partie du système de surveillance. Cette approche est très recommandée mais notez que Firefox vous avertit que cela risque d'« empêcher certains sites de fonctionner ». Cet avertissement peut faire hésiter certains utilisateurs, d'autant plus qu'avec les sites en question, il n'y aura aucun message clair, uniquement des dysfonctionnements bizarres. La plupart des sites Web commerciaux sont en effet développés sans tenir compte de la possibilité que le visiteur ait activé ces options. Si le site de votre banque ne marche plus après avoir changé ces réglages, ne comptez pas sur le support technique de la banque pour vous aider à analyser le problème, on vous dira

probablement uniquement d'utiliser Google Chrome et de ne pas toucher aux réglages. D'un côté, les responsables du Web de surveillance disent qu'on a le choix, qu'on peut changer les réglages, d'un autre côté ils exercent une pression sociale intense pour qu'on ne le fasse pas. Et puis, autant on peut renoncer à regarder le site Web d'un journal lorsqu'il ne marche pas sans cookies, autant on ne peut guère en faire autant lorsqu'il s'agit de sa banque.

De même qu'on peut contrôler, voire débrayer les cookies, on peut supprimer le code Javascript. À ma connaissance, Firefox ne permet pas en standard de le faire, mais il existe une extension nommée NoScript pour le faire. Comme avec les cookies, cela posera des problèmes avec certains sites Web et, pire, ces problèmes ne se traduiront pas par des messages clairs mais par des dysfonctionnements. Là encore, peu de chance que le logiciel que l'entreprise en question a chargé de répondre aux questions sur Twitter vous aide.

Enfin, un troisième outil pour limiter les divers risques du Web est le bloqueur de publicité. (Personnellement, j'utilise uBlock Origin.)

+ Tout	
lemonde.fr	++ -
www.lemonde.fr	++ -
akamaiedge.net	+ -
batch.com	+ -
fastly.net	++ -
lubenda.com	+ -
lemde.fr	++ -
stackpathdns.com	+ -
youtube.com	-

www.lemonde.fr

Bloqués sur cette page
3, soit 1%

Domaines connectés
7 sur un total de 8

Bloqués depuis l'installation
689 520, soit 8%

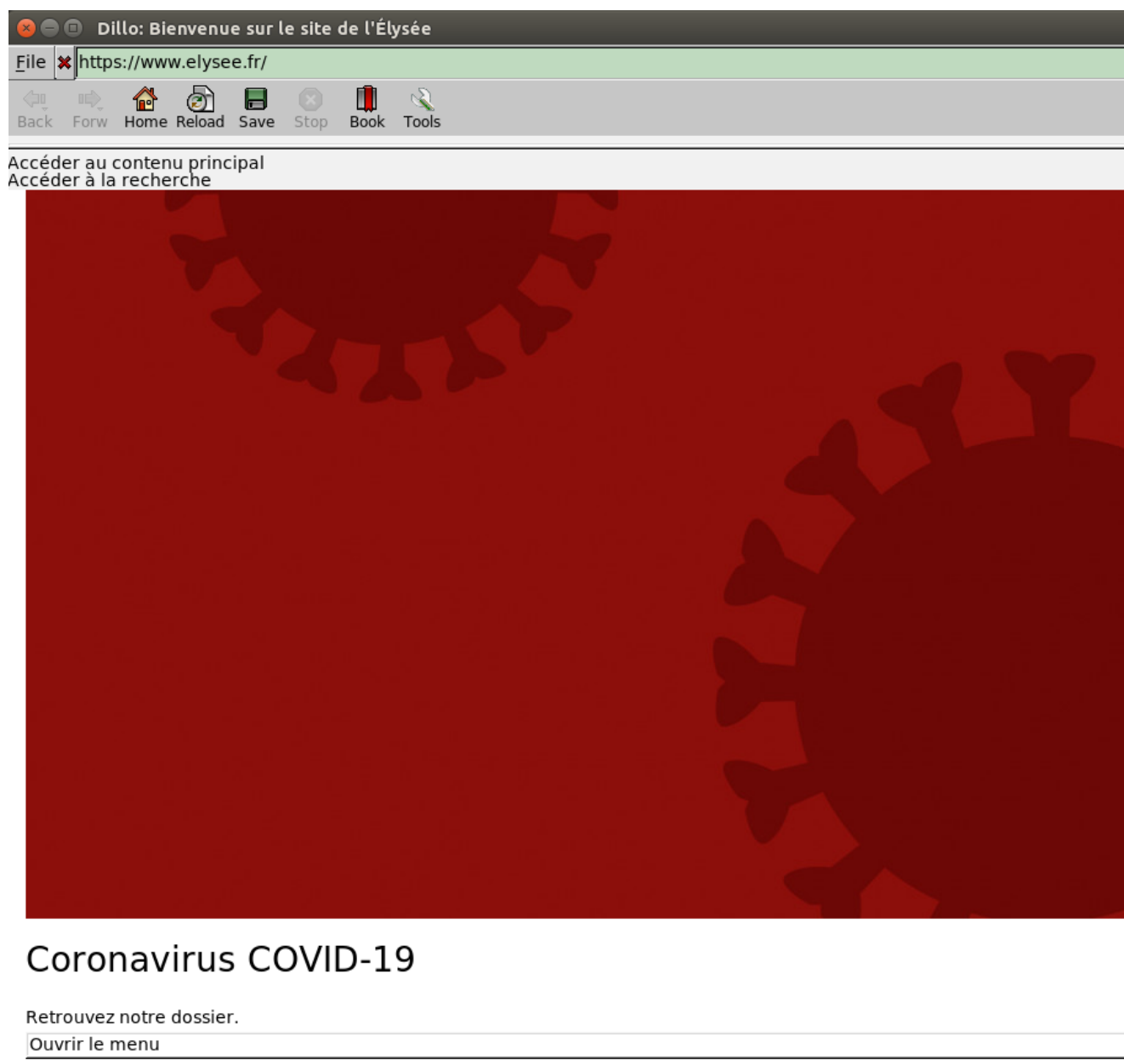
uBlock Origin sur le site du Monde, où il a bloqué trois pisteurs et publicités. On voit aussi à gauche la liste des sites chargés automatiquement par votre navigateur quand vous regardez Le Monde.

Absolument indispensable à la fois pour éviter de consacrer du temps de cerveau à regarder les publicités, pour la sécurité (les réseaux de distribution de la publicité sont l'endroit idéal pour diffuser du logiciel malveillant) et aussi pour l'empreinte environnementale, le bloqueur empêchant le chargement de contenus qui feront travailler votre ordinateur pour le profit des agences de publicité et des annonceurs.

Un navigateur Web léger ?

Une solution plus radicale est de changer de navigateur Web. On peut ainsi préférer le logiciel Dillo, explicitement conçu pour la légèreté, les performances et la vie privée. Dillo marche parfaitement avec des sites Web bien conçus, mais ceux-

ci ne sont qu'une infime minorité. La plupart du temps, le site sera affiché de manière bizarre. Et on ne peut pas le savoir à l'avance ; naviguer sur le Web avec Dillo, c'est avoir beaucoup de mauvaises surprises et seulement quelques bonnes (le Framablog, que vous lisez en ce moment, marche très bien avec Dillo).



Le site de l'Élysée vu par Dillo. Inutilisable (et pas par la faute de Dillo mais par le choix de l'Élysée d'un site spectaculaire plutôt qu'informatif).

Autre navigateur « alternatif », le Tor Browser. C'est un

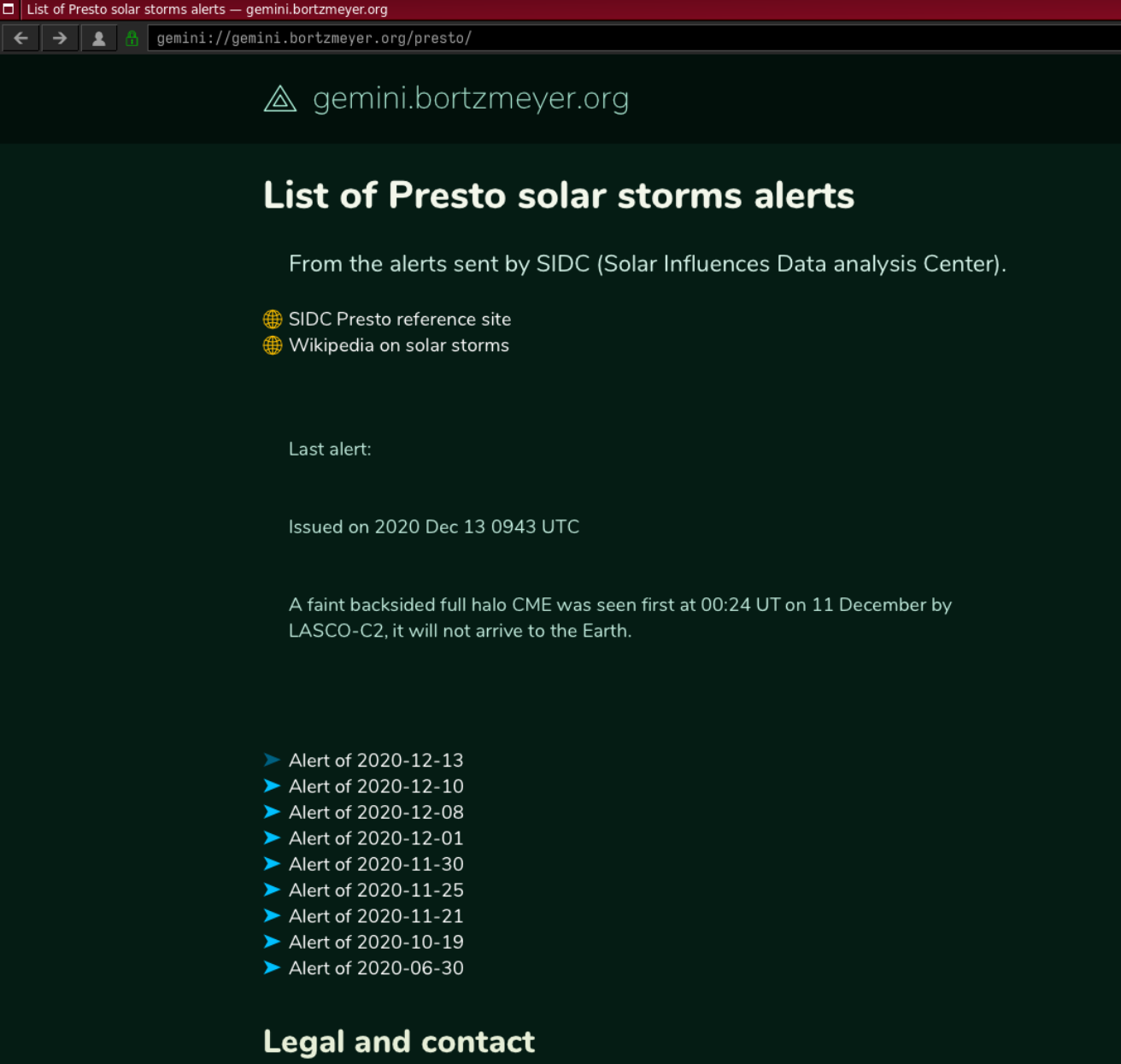
Firefox modifié, avec NoScript inclus et qui, surtout, ne se connecte pas directement au site Web visité mais passe par plusieurs relais du réseau Tor, supprimant ainsi un moyen de pistage fréquent, l'adresse IP de votre ordinateur. Outre que certains sites ne réagissent pas bien aux réglages du Tor Browser, le passage par le réseau Tor se traduit par des performances décrues.

Toutes ces solutions techniques, du bloqueur de publicités au navigateur léger et protecteur de la vie privée, ont un problème commun : elles sont perçues par les sites Web comme « alternatives » voire « anormales ». Non seulement le site Web risque de ne pas fonctionner normalement mais surtout, on n'est pas prévenu à l'avance, et même après on n'a pas de diagnostic clair. Le Web, pourtant devenu un écosystème très complexe, n'a pas de mécanismes permettant d'exprimer des préférences et d'être sûr qu'elles sont suivies. Certes, il existe des techniques comme l'en-tête « Do Not Track » où votre navigateur annonce qu'il ne souhaite pas être pisté mais il est impossible de garantir qu'il sera respecté et, vu le manque d'éthique de la grande majorité des sites Web, il vaut mieux ne pas compter dessus.




Gemini, une solution de rupture

Cela a mené à une approche plus radicale, sur laquelle je souhaitais terminer cet article, le projet Gemini. Gemini est un système complet d'accès à l'information, alternatif au Web, même s'il en reprend quelques techniques. Gemini est délibérément très simple : le protocole, le langage parlé entre le navigateur et le serveur, est très limité, afin d'éviter de transmettre des informations pouvant servir au pistage (comme l'en-tête User-Agent du Web) et il n'est pas extensible. Contrairement au Web, aucun mécanisme n'est prévu pour ajouter des fonctions, l'expérience du Web ayant montré que ces fonctions ne sont pas forcément dans l'intérêt de l'utilisateur. Évidemment, il n'y a pas l'équivalent des

cookies. Et le format des pages est également très limité, à la fois pour permettre des navigateurs simples (pas de CSS, pas de Javascript), pour éviter de charger des ressources depuis un site tiers et pour diminuer la consommation de ressources informatiques par le navigateur. Il n'y a même pas d'images. Voici deux exemples de navigateurs Gemini :



The screenshot shows a Gemini browser window with the following content:

- Address bar: `gemini://gemini.bortzmeyer.org/presto/`
- Page title: `List of Presto solar storms alerts — gemini.bortzmeyer.org`
- Logo:  `gemini.bortzmeyer.org`
- Section header: **List of Presto solar storms alerts**
- Text: From the alerts sent by SIDC (Solar Influences Data analysis Center).
- Links:
 -  [SIDC Presto reference site](#)
 -  [Wikipedia on solar storms](#)
- Text: Last alert:
- Text: Issued on 2020 Dec 13 0943 UTC
- Text: A faint back-sided full halo CME was seen first at 00:24 UT on 11 December by LASCO-C2, it will not arrive to the Earth.
- List of alerts:
 - [Alert of 2020-12-13](#)
 - [Alert of 2020-12-10](#)
 - [Alert of 2020-12-08](#)
 - [Alert of 2020-12-01](#)
 - [Alert of 2020-11-30](#)
 - [Alert of 2020-11-25](#)
 - [Alert of 2020-11-21](#)
 - [Alert of 2020-10-19](#)
 - [Alert of 2020-06-30](#)
- Section header: **Legal and contact**

Le client Gemini Lagrange

List of Presto solar storms alerts

From the alerts sent by SIDC (Solar Influences Data analysis Center).

- [SIDC Presto reference site](#)
- [Wikipedia on solar storms](#)

Last alert:

Issued on 2020 Dec 13 0943 UTC

A faint backside full halo CME was seen first at 00:24 UT on 11 December by LASCO-C2, it will not arrive to the Earth.

- [Alert of 2020-12-13](#)
- [Alert of 2020-12-10](#)
- [Alert of 2020-12-08](#)
- [Alert of 2020-12-01](#)
- [Alert of 2020-11-30](#)
- [Alert of 2020-11-25](#)
- [Alert of 2020-11-21](#)
- [Alert of 2020-10-19](#)
- [Alert of 2020-06-30](#)

Legal and contact

This Gemini mirror

U:%*- *elpher* Top L1 (elpher Fill)

Le même site Gemini, vu par un client différent, Elpher

Gemini est un système récent, s'inspirant à la fois de systèmes anciens (comme le Web des débuts) et de choses plus récentes (ainsi, contrairement au Web, le chiffrement du trafic, pour compliquer la surveillance, est systématique). Il reprend notamment le concept d'URL donc par exemple le site d'informations sur les alertes de tempêtes solaires utilisé plus haut à titre d'exemple est

[gemini://gemini.bortzmeyer.org/presto/](http://gemini.bortzmeyer.org/presto/). Gemini est actuellement en cours de développement, de manière très ouverte, notamment sur la liste de diffusion publique du projet. Tout le monde peut participer à sa définition. (Mais, si vous voulez le faire, merci de lire la FAQ d'abord, pour ne pas recommencer une question déjà discutée.) Conformément aux buts du projet, écrire un client ou un serveur Gemini est facile et des dizaines de logiciels existent déjà. Le nom étant une allusion aux missions spatiales étatsuniennes Gemini, mais signifiant également « jumeaux » en latin, beaucoup de ces logiciels ont un nom qui évoque le spatial ou la gémellité. Pour la même raison spatiale, les sites Gemini se nomment des capsules, et il y en a actuellement quelques centaines opérationnelles. (Mais, en général, avec peu de contenu original. Gemini ressemble pour l'instant au Web des débuts, avec du contenu importé automatiquement d'autres services, et du contenu portant sur Gemini lui-même.)

On a vu que Gemini est une solution très disruptive et qui ne sera pas facilement adoptée, tant le marketing a réussi à convaincre que, sans vidéos incluses dans la page, on ne peut pas être vraiment heureux. Gemini ne prétend pas à remplacer le Web pour tous ses usages. Par exemple, un CMS, logiciel de gestion de contenu, comme le WordPress utilisé pour cet article, ne peut pas être fait avec Gemini, et ce n'est pas son but. Son principal intérêt est de nous faire réfléchir sur l'accès à l'information : **de quoi avons-nous besoin pour nous informer ?**

- - -

- Pour en savoir plus sur Gemini, cet autre article un peu plus technique du même auteur : Le protocole Gemini, revenir à du simple et sûr pour distribuer l'information en ligne ?

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 parlais ». 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 nombreuses-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 GAFAs 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.