RFC 8905 : The 'payto' URI scheme for payments

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 26 octobre 2020

Date de publication du RFC : Octobre 2020

https://www.bortzmeyer.org/8905.html

Le paiement de services ou de biens est un problème crucial sur l'Internet. En l'absence de mécanisme simple, léger et respectant la vie privée, on n'a aujourd'hui que des solutions centralisées dans des gros monstres, comme Amazon pour vendre des objets physiques, ou YouTube pour monétiser ses vidéos. Ce RFC ne propose pas de solution magique mais il spécifie au moins une syntaxe pour indiquer un mécanisme de paiement : le plan d'URI `payto:`, qui permettra peut-être un jour de faciliter les paiements.

Le paysage d'aujourd'hui est le suivant. Si vous êtes un créateur (d'articles, de vidéos, de dessins, peu importe) et que vous voulez être payé pour vos créations (ce qui est tout à fait légitime), vous n'avez comme solutions que de faire appel à du financement participatif (style Liberapay ou Ulule) ou bien de passer par une grosse plate-forme comme YouTube, qui imposera ses règles, comme la captation de données personnelles. Sans compter le contrôle du contenu, qui fait qu'une vidéo parlant de sujets sensibles sera démonétisée. (Voyez par exemple cette vidéo de Charlie Danger <https://www.youtube.com/watch?v=QONJ8uBrn-s> où, à partir de 10 :45 et surtout 11 :45, elle explique comment YouTube n'a pas aimé sa vidéo sur l'IVG et ce qui en est résulté.) Mais cette solution d'hébergement sur un GAFA est sans doute la plus simple pour le créateur, et elle ne dépend pas de la bonne volonté des lecteurs/spectacteurs. Lorsqu'on discute avec un·e vidéaste de son hébergement chez Google et qu'on lui propose d'utiliser plutôt un service libre et décentralisé fait avec PeerTube, la réponse la plus fréquente est « mais je perdrais la monétisation, et je ne peux pas vivre seulement d'amour et d'eau fraîche ». Je l'ai dit, il n'existe pas encore de solution parfaite à ce problème. Pour un cas plus modeste, celui de ce blog, j'ai tenté Flattr <https://www.bortzmeyer.org/flattr.html> et Bitcoin <https://www.bortzmeyer.org/bitcoin-blog.html> mais avec très peu de succès.

Ce nouveau RFC ne propose pas une solution de paiement, juste un moyen de faire des URI qu'on pourra mettre dans ces pages Web pour pointer vers un mécanisme de paiement. Par exemple, va indiquer qu'il faut envoyer 10 milli-bitcoins à cette adresse (c'est la mienne), avec un gentil message. Si votre navigateur Web gère le plan d'URI `payto:` (ce qu'aucun ne fait à l'heure actuelle), que vous cliquez puis confirmez, je recevrai 0,01 bitcoin. (Notez qu'il existe une syntaxe spécifique à Bitcoin, pour un

paiement, décrite dans le BIP 0021 <https://en.bitcoin.it/wiki/BIP_0021>, et qui ressemble beaucoup à celle du RFC.)

Un peu de technique maintenant : les plans d'URI ("*scheme*") sont définis dans le RFC 3986 [1], section 3.1. Vous connaissez certainement des plans comme `http:` ou `mailto:`. Le plan `payto:` est désormais dans le registre IANA des plans <https://www.iana.org/assignments/uri-schemes/uri-schemes.xml#uri-schemes-1>. Après le plan et les deux barres se trouve l'autorité. Ici, elle indique le type de paiement. Notre RFC en décrit certains (comme `bitcoin` montré dans l'exemple) et on pourra en enregistrer d'autres. L'idée est de pouvoir présenter à l'utilisateur un mécanisme uniforme de paiement, quel que soit le type de paiement. (À l'heure actuelle, si vous acceptez les paiements par Flattr <https://www.bortzmeyer.org/flattr.html> et PayPal, vous devez mettre deux boutons différents sur votre site Web, et qui déclencheront deux expériences utilisateur très différentes. Sans compter les traqueurs qu'il y a probablement derrière le bouton de Paypal.)

Question interface utilisateur, le RFC recommande que le navigateur permette ensuite à l'utilisateur de choisir le compte à utiliser, s'il en a plusieurs et, bien sûr, lui demande clairement une confirmation claire. Pas question qu'un simple clic déclenche le paiement ! (Cf. section 8 du RFC, qui pointe entre autres le risque de "*clickjacking*".) Notez aussi que le RFC met en garde contre le fait d'envoyer trop d'informations (par exemple l'émetteur) dans le paiement, ce qui serait au détriment de la vie privée.

On peut ajouter des options à l'URI (section 5 du RFC). Par exemple la quantité à verser (`amount`, cf. l'exemple Bitcoin plus haut, le code de la monnaie, s'il a trois lettres, devant être conforme à ISO 4217, le Bitcoin n'ayant pas de code officiel, j'ai utilisé un nom plus long), `receiver-name` et `sender-name`, `message` (pour le destinataire) et `instruction`, ce dernier servant à préciser le traitement à faire par l'organisme de paiement.

Voici un autre exemple d'URI `payto:`, après Bitcoin, IBAN (ISO 20022) : . L'exemple est tiré du RFC. Je n'ai pas mis mon vrai numéro IBAN car je ne suis pas expert en sécurité des IBAN et je ne sais pas s'il n'y a pas des inconvénients à le rendre public. Un expert pour confirmer ? À part ça, avec le type `iban`, l'option `message` correspond à ce que SEPA appelle "*unstructured remittance information*" et `instruction` au "*end to end identifier*".

Puisqu'il y a une option permettant d'envoyer un message, la section 6 du RFC note qu'il n'y a pas de vrai mécanisme d'internationalisation, entre autres parce qu'on ne peut pas garantir de manière générale comment ce sera traité par les établissements de paiement.

Outre Bitcoin et IBAN, déjà vus, notre RFC enregistre plusieurs autres types de mécanismes de paiement. ACH, BIC/Swift et UPI appartiennent au monde bancaire classique. Le type `ilp` vient, lui, du monde Internet, comme Bitcoin, est pour les paiements via InterLedger, s'appuyant sur les adresses InterLedger <https://interledger.org/rfcs/0015-ilp-addresses/>. Et il y a aussi un type `void`, qui indique que le paiement se fera en dehors du Web, par exemple en liquide.

Cette liste n'est pas fermée, mais est stockée dans un registre évolutif <https://git.gnunet.org/gana.git/tree/payto-payment-target-types/registry.rec>, registre peuplé selon la politique « Premier arrivé, premier servi ». Vous noterez que ce registre n'est pas géré par l'IANA mais par GANA <https://gana.gnunet.org/>.

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc3986.txt`

La section 7 indique les critères d'enregistrement souhaitables d'un nouveau type (les enregistrements existants peuvent servir d'exemple) : fournir des références précises, trouver un nom descriptif, préférer un nom neutre à celui d'un organisme particulier, etc. J'ai regardé ces critères pour le cas de PayPal, mécanisme de paiement très répandu sur l'Internet, mais ce n'était pas évident. Même si on spécifie des URI du genre `payto://paypal/smith@example.com`, il ne serait pas évident pour le navigateur de les traduire en une requête PayPal (PayPal privilégie l'installation de son propre bouton actif, ou bien le système PayPal.me). Bref, je n'ai pas continué mais si quelqu'un de plus courageux veut enregistrer le type `paypal`, il n'est pas nécessaire d'être représentant officiel de PayPal, et ce n'est pas trop difficile. Idem pour d'autres systèmes de paiement par encore spécifiés comme Liberapay. Notez que le système conçu par les auteurs du RFC, Taler, n'est pas encore enregistré non plus.