Mon premier nom Namecoin enregistré

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 19 janvier 2014. Dernière mise à jour le 2 mars 2016

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

Il y a depuis plusieurs années une énorme activité de recherche et de programmation pour développer des outils de communication sur l'Internet échappant ou limitant la censure et/ou l'espionnage. Ces outils reposent en général sur les idées de communication pair-à-pair et l'absence d'un élément jouant un rôle particulier (pas de racine, pas d'AC, pas de tiers de confiance, etc), puisqu'un tel élément, même s'il est sympa au début, peut toujours devenir un facteur de censure et/ou d'espionnage. La grande majorité de ces projets ne sont pas allés très loin, et beaucoup étaient même ridicules dès le début. Assurer les fonctions d'un élément particulier, notamment les fonctions de sécurité, en pair-à-pair pur est vraiment difficile ! Parmi les rares survivants d'un processus de sélection très sévère, il y a les techniques fondées sur une chaîne de blocs publique, vérifiable par tous, chaîne popularisée par Bitcoin. Par exemple, un motif d'insatisfaction parfois exprimé vis-à-vis du système de noms de domaines est sa structure arborescente, qui fait dépendre, même si c'est très indirectement, d'une racine. Peut-on se passer d'une telle structure ? C'est hautement non trivial mais on peut essayer et Namecoin est une technique prometteuse. Je viens de l'utiliser pour mes premiers noms Namecoin, `bortzmeyer` et `d/bortzmeyer` (et vous allez devoir lire l'article jusqu'au bout pour savoir à quoi rime ce `d/`).

Un bon exemple du problème posé par l'actuel système de noms de domaine est donné dans un article récent de TechDirt <http://www.techdirt.com/articles/20140110/12140025836/pharmacy-lobbyists.shtml>. Les lobbies qui veulent artificiellement maintenir un modèle d'affaires menacé par l'Internet sont très prompts à réclamer la suppression (« "*take down*" ») des noms de domaines utilisés pour des activités qu'ils n'aiment pas (comme le partage des œuvres culturelles). Le risque de censure est donc élevé car, dans de nombreux cas, ces lobbies ont eu gain de cause <http://torrentfreak.com/us-resume-file-sharing-domain-seizures-110201/>. Beaucoup de titulaires de noms de domaines ne se sentent pas protégés contre l'arbitraire de ces suppressions ou détournements de noms.

Comment est-ce que Namecoin prétend résoudre ce problème ? Je ne vais pas ici détailler son fonctionnement technique (l'article du Wikipédia anglophone cité plus haut, ainsi que les différents articles cités dans cet article vous en diront plus), mon but est plutôt de raconter une expérience vécue. Disons que Namecoin repose sur un **livre des opérations**, une chaîne publique de blocs, comme Bitcoin (c'est d'ailleurs à l'origine le même code, mais la chaîne est différente et on ne peut pas acheter des noms avec des bitcoins). (Et merci à Nathanaelle <https://twitter.com/ns_m> pour le terme de livre

des opérations.) On l'oublie souvent, mais les transactions Bitcoins incluent un programme, écrit dans un langage simple et limité, exécuté pour valider la transaction. Dans Bitcoin, ce langage est très limité, notamment pour des raisons de sécurité. Il est un peu plus riche dans Namecoin, il comporte notamment des méthodes pour enregistrer un nom. L'existence d'un nom se vérifie donc en validant toute la chaîne et en relevant la création d'un nom, pas trop ancienne (les noms sont enregistrés pour une certaine période). On a donc, sinon vaincu, du moins sérieusement endommagé le triangle de Zooko : on a des noms sympas (on choisit le nom qu'on veut), sûrs (tout le monde peut vérifier l'intégrité du livre des opérations) et uniques (de même qu'avec Bitcoin, tout le monde peut vérifier qu'un bitcoin n'a pas été dépensé deux fois, avec Namecoin, tout le monde peut vérifier qu'un nom n'a pas été enregistré deux fois).

Bon, et qu'est-ce qui empêche d'enregistrer plein de noms ? Ah, mais je n'ai pas dit que Namecoin était gratuit. Il faut payer, en namecoins. Cette monnaie s'obtient, comme les bitcoins, en minant, ou bien en l'achetant à quelqu'un d'autre. Cette seconde voie est plus facile, j'ai donc acheté mes namecoins sur Kraken <https://www.bortzmeyer.org/bitcoin-marches.html>.

Maintenant, place à la pratique. Pour créer un nom Namecoin, il faut installer un nœud Namecoin. Je n'ai pas l'impression qu'il existe, comme il y a pour Bitcoin, un logiciel permettant de faire un client léger, ne minant pas et ne validant pas. J'ai donc installé un nœud complet (la chaîne est bien plus courte que celle de Bitcoin et il y a moins de mineurs donc le travail est bien plus raisonnable). Une fois téléchargé <https://github.com/namecoin/namecoin-core> et compilé, le démon s'utilise comme le classique nœud Bitcoin. Notez l'option `-gen` qui lui dit de participer au minage (et qui fait donc tourner votre CPU en permanence). Vérifions que le démon tourne bien :

```
% namecoin-cli getinfo
{
    "version" : 37300,
    "balance" : 0.11000000,
    "blocks" : 157803,
    "timeoffset" : -1,
    "connections" : 8,
    "proxy" : "",
    "generate" : false,
    "genproclimit" : -1,
    "difficulty" : 1546423251.74634910,
    "hashespersec" : 0,
    "testnet" : false,
    "keypoololdest" : 1389730666,
    "keypoolsize" : 101,
    "paytxfee" : 0.00000000,
    "mininput" : 0.00010000,
    "errors" : ""
}
```

On en est au bloc 157803 (on aurait pu avoir juste le numéro de bloc avec `namecoin-cli getblockcount`). On peut vérifier sur un explorateur public du Livre des Opérations, comme <http://explorer.dot-bit.org/>, à quel bloc lui en est. La première fois qu'on lance le nœud, on est évidemment loin derrière. Personnellement, mon PC ordinaire à la maison a mis trois heures pour rattraper la chaîne publique, trois heures pendant lesquelles je faisais des `namecoin-cli getblockcount` de temps en temps pour vérifier si ça avançait.

```
% date; namecoin-cli getblockcount
Thu Jan 16 10:01:01 CET 2014
84566

% date; namecoin-cli getblockcount
Thu Jan 16 10:11:11 CET 2014
92468
```

Mais la vitesse de progression de cette chaîne de blocs n'est pas du tout prévisible, il vaut mieux être patient. Au bout du compte, une fois qu'on est synchrone (ou même avant mais, dans ce cas, on ne verra pas ses propres transactions), on peut commencer à travailler.

D'abord, le fric ! Créons une adresse pour recevoir des namecoins, en indiquant un nom de compte (vous pouvez en avoir plusieurs, cela ne sert que localement, pour gérer vos namecoins) :

```
% namecoin-cli getnewaddress bortzmeyer
Myw9PZkBDjjKpaCjSMnWNGrVd7AnDpQoBY

% namecoin-cli getreceivedbyaddress Myw9PZkBDjjKpaCjSMnWNGrVd7AnDpQoBY
0.00000000
```

(Vous avez vu, j'ai subtilement affiché mon adresse : si vous aimez cet article, vous pouvez y envoyer des namecoins. Mais rappelez-vous que tout est public, cela se verra, par exemple avec l'explorateur <http://explorer.dot-bit.org/a/Myw9PZkBDjjKpaCjSMnWNGrVd7AnDpQoBY>.) J'envoie ensuite depuis Kraken <https://www.bortzmeyer.org/bitcoin-marches.html> des namecoins à cette adresse :

```
% namecoin-cli getreceivedbyaddress Myw9PZkBDjjKpaCjSMnWNGrVd7AnDpQoBY
0.09500000
```

Les sous sont arrivés, on va pouvoir créer des noms. (Si on était sérieux, on commencerait par faire une sauvegarde des fichiers en `/.namecoin`. Comme avec Bitcoin, si on perd sa clé privée, on perd tout, et sans recours possible.) Commençons avec un nom au hasard :

```
% namecoin-cli  name_new bortzmeyer
[
    "abd0330ed6d82e425...65d77bca37b3",
    "3789f...a9d31"
]
```

Notez bien les deux nombres qui serviront par la suite. À ce stade, n'est enregistré dans le livre des opérations qu'un condensat du nom (pour éviter qu'un malin ne voit passer votre demande et ne l'enregistre aussitôt, avant que plusieurs pairs n'aient pu valider votre transaction). Dès que l'explorateur public a reçu le nouveau bloc (sa page d'accueil affiche le dernier bloc reçu), vous pouvez vérifier que la transaction a eu lieu (utilisez comme critère de recherche le premier nombre affiché ci-dessus, abd033...) mais une recherche par nom échouera puisque seul le condensat est présent. La FAQ dit d'attendre douze blocs. Au rythme d'aujourd'hui, cela peut faire plusieurs heures. Puis « abattez vos cartes » et indiquez le vrai nom :

```
% namecoin-cli name_firstupdate bortzmeyer 3789fa452d9a9d31 abd0330ed6d82e4250a17d0bc8c709461c3a7218d59958efd2d9
02cd4cc2283bbba6586e988ce141e1b6cfa3ceabc7752617fb0a99c15efafb93
```

(J'ai bien dit `name_firstupdate`, et pas `first_update`, la FAQ est fausse.) Cette fois, une fois l'explorateur synchronisé au nouveau bloc, une recherche par nom va fonctionner (testez <http://explorer.dot-bit.org/n/140495>). Félicitations, vous avez un nom Namecoin, qui vous assurera l'admiration de vos ami(e)s, le désir de vos amant(e)s et une augmentation par votre entreprise.

À noter que, si vous aviez fait un `name_new` pour un nom déjà existant, c'est au moment du `name_firstupdate` que la collision serait détectée. Sinon, le dernier argument de la commande est la valeur associée au nom. Ici, j'ai juste mis 1 pour uniquement réserver le nom. Mais, en fait, on peut associer des valeurs à des attributs, en indiquant des données en JSON. Ici, je mets mon adresse de courrier (il y a une liste temporaire de valeurs possibles <http://dot-bit.org/Namespace:Domain_names_v2.0>) :

```
% namecoin-cli name_update  bortzmeyer  '{"email": "stephane+namecoin@bortzmeyer.org"}'
```

On peut voir que cela a bien été pris en compte. Du fait que le livre des transactions est public, tout le monde peut voir toutes les données :

```
% namecoin-cli name_filter bortzmeyer
[
    {
        "name" : "bortzmeyer",
        "value" : "{\"email\": \"stephane+namecoin@bortzmeyer.org\"}",
        "expires_in" : 35962
    },
...
```

Attention, le JSON envoyé n'est pas testé et des erreurs de syntaxe idiotes peuvent donc empêcher son interprétation. Il peut être prudent de vérifier d'abord, par exemple avec <http://dot-bit.org/tools/domainCheck.php>.

Prêtez attention au membre `expires_in` (noté en nombre de blocs). Avec Namecoin, les noms ne sont réservés que pour une période donnée. Pensez à les renouveler. Et mettez en place une supervision, par exemple avec Name Alert <http://namealert.mvps.eu/edit>. Ainsi, vous serez prévenus lorsqu'un nom approchera de l'expiration, par un message du genre :

```
Date: Thu, 22 Jan 2015 09:45:02 -0000
From: alert@namealert.mvps.eu
To: stephane+namealert@bortzmeyer.org
Subject: Your domain d/bortzmeyer will soon expire

One of your Namecoin domains, d/bortzmeyer
will expire in 96 blocks.

Current rate is 150.027387087 blocks per day, so your domain
will expire in approximately 0 days.
```

(En fait, c'est théorique, Name Alert ne marche apparemment pas bien et j'ai perdu <https://www.bortzmeyer.org/maman-j-ai-perdu-mon-namecoin.html> `d/bortzmeyer` ainsi. Pas de renouvellement, pas d'alerte, et, une fois le nom détruit, quelqu'un d'autre l'a enregistré et Name Alert ne m'a prévenu que plusieurs mois après. On peut voir cette histoire en <https://namecha.in/name/id/bortzmeyer>. En raison d'une bogue, <http://explorer.namecoin.info/> n'arrive pas à l'afficher.)

Bref, Namecoin reprend un des problèmes les plus énervants du système des noms de domaine, l'obligation de veiller à l'expiration et au renouvellement <http://www.numerama.com/magazine/24072-le-ministere-de-la-culture-oublie-de-payer-son-nom-de-domaine-un-citoyen-le-fai
html>. Pour renouveler, vous n'avez pas de commande spéciale, vous faites juste un `name_update` en indiquant les mêmes valeurs (malheureusement, il n'y a pas de solution plus simple et moins risquée) :

```
% namecoin-cli name_update d/bortzmeyer  '{"ip":"204.62.14.153", "ip6":"2605:4500:2:245b::42", "email": "ste
3393a1f8b428053c2b343deec3f3ff5ec10f360e9a7a82ae91606052c5edab76
```

Bien, nous avons maintenant une base de données fiable, à l'intégrité vérifiable, et qui contient nos jolis noms. Nous pouvons associer à ces noms des informations intéressantes comme l'adresse de courrier ou comme des adresses IP. Mais tout le monde n'a pas forcément un résolveur Namecoin chez lui et dans toutes ses applications. Pour l'instant, le protocole nettement dominant pour résoudre les noms, c'est le DNS. Si on ne peut pas utiliser le DNS pour résoudre les noms Namecoin, personne n'utilisera Namecoin (problème classique des nouvelles techniques). Mais, heureusement, on peut. Le principe est d'utiliser un TLD dédié, `.bit` (non officiellement enregistré, attention, des problèmes pourront survenir). Il faut monter un serveur DNS faisant autorité pour `.bit` et/ou configurer ses résolveurs pour utiliser des serveurs de `.bit`. Les différentes méthodes possibles sont documentées en ligne <http://dot-bit.org/HowToBrowseBitDomains>. Commençons par le plus trivial, configurer le résolveur Unbound pour interroger les serveurs existants. On met dans la configuration d'Unbound :

```
domain-insecure: "bit"

stub-zone:
       name: "bit"
       stub-addr: 178.32.31.41
       stub-addr: 78.47.86.43
       stub-addr: 95.211.195.245
       stub-addr: 2001:1af8:4020:a037:1::1000
       stub-addr: 162.243.56.54
```

La première ligne est nécessaire puisque la racine du DNS est signée avec DNSSEC mais que `.bit` n'y est pas enregistré. Les dernières lignes donnent les adresses de serveurs publics faisant autorité pour `.bit`. On trouve une liste en ligne <http://dot-bit.org/HowToBrowseBitDomains> mais la liste est peu à jour, et compte pas mal de serveurs qui ne marchent pas. Bon, Unbound est assez intelligent pour n'utiliser que ceux qui répondent et :

```
% dig  ANY explorer.bit

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> ANY explorer.bit
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25144
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;explorer.bit. IN ANY

;; ANSWER SECTION:
explorer.bit. 86162 IN SOA ns0.web-sweet-web.net. root.srv01.web-sweet-web.net. (
2014010900 ; serial
21600      ; refresh (6 hours)
3600       ; retry (1 hour)
604800     ; expire (1 week)
86400      ; minimum (1 day)
)
explorer.bit. 86162 IN NS ns1.web-sweet-web.net.
explorer.bit. 86162 IN NS ns0.web-sweet-web.net.
explorer.bit. 86162 IN TXT "v=spf1 a mx ?all"
explorer.bit. 86162 IN A 178.32.102.200
explorer.bit. 86162 IN MX 5 srv01.web-sweet-web.net.

;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sun Jan 19 18:29:24 2014
;; MSG SIZE  rcvd: 202
```

C'est parfait, tout marche. Notez que cela laisse deux gros problèmes de sécurité : ces serveurs publics faisant autorité pour `.bit` ne sont pas forcément de confiance (on trouve de tout parmi eux) et, même s'ils sont de confiance, il n'y a rien qui protège la liaison réseau entre vous et eux. C'est d'autant plus embêtant que les outils de débogage manquent. Par exemple, on ne peut pas se fier au numéro de série dans l'enregistrement SOA de `.bit` : chaque serveur le met à sa guise selon un algorithme différent.

Avant de revenir à ces problèmes de sécurité et de montrer la solution, un mot sur la publication. Est-ce qu'il suffit d'enregistrer le nom `bortzmeyer` pour qu'il soit publié dans `.bit` ? Non. Il existe une convention qui partitionne Namecoin en plusieurs espaces de nommage <https://dot-bit.org/Namespace>. Pour être publié dans `.bit`, le nom doit être préfixe par `d/`. On va donc enregistrer `d/bortzmeyer`. Au `name_firstupdate`, il va bien être publié dans `.bit` (attention, les serveurs publics ne se mettent pas à jour en temps réel, on peut avoir à patienter des heures). Voici un exemple, en publiant des adresses IP :

```
% namecoin-cli name_update d/bortzmeyer '{"ip":"204.62.14.153", "ip6":"2605:4500:2:245b::42"}'
46c5e87df04a491166d9c4a407007ed32b2a149362f52b7c5a65504362b84f70
```

Si vous avez vous-même un résolveur qui gère les `.bit`, vous pouvez vérifier que cela marche :

```
% dig  AAAA bortzmeyer.bit

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> AAAA bortzmeyer.bit
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42151
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;bortzmeyer.bit. IN AAAA

;; ANSWER SECTION:
bortzmeyer.bit. 86357 IN AAAA 2605:4500:2:245b::42

;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sun Jan 19 18:40:44 2014
;; MSG SIZE  rcvd: 71
```

Et comme j'ai en prime ajouté une directive `ServerAlias` dans Apache, vous pouvez, si votre résolveur gère `.bit`, visiter <http://bortzmeyer.bit/>.

Comme on a une copie de toute la base des noms, des opérations comme de trouver les données sur un nom sont triviales :

```
% namecoin name_filter d/bortzmeyer
[
    {
        "name" : "d/bortzmeyer",
        "value" : "{\"ip\":\"204.62.14.153\",\"ip6\":\"2605:4500:2:245b::42\"}",
        "expires_in" : 35666
    }
]
```

(Notez le JSON dans le JSON...) On peut faire la même interrogation en ligne via l'explorateur public : <http://explorer.dot-bit.org/n/140687> ou via n'importe quelle passerelle comme DNSchain <http://dns.dnschain.net/d/bortzmeyer>. Contrairement au système actuel des noms de domaines, qui utilise deux protocoles complètement différents pour distribuer les données, le DNS et whois, Namecoin n'a qu'un mécanisme pour tout. (Vous pouvez aussi utiliser un moteur de recherche public <http://dot-bit.org/tools/domainSearch.php>.) Voici un exemple avec un domaine qui, au lieu de mettre directement des données (comme les adresses IP) dans Namecoin, utilise un mécanisme de délégation, ici vers deux serveurs DNS :

```
% namecoin name_filter d/explorer
[
    {
        "name" : "d/explorer",
        "value" : "{\"info\":{\"registrar\":\"http://register.dot-bit.org\"},\"email\": \"register@dot-bit.org\"
        "expires_in" : 3859
    }
]
```

Nous avions vu que la solution pour accéder aux domaines `.bit` en utilisant des serveurs publics n'était pas idéale, question de sécurité. La seule solution sûre est d'avoir un serveur `.bit` chez soi. Cela peut se faire, par exemple, avec le programme NamecoinToBind <https://github.com/khalahan/NamecoinToBind/>. Lorsqu'on le fait tourner, il fabrique un fichier de zone à la syntaxe du RFC 1035 [1] (contrairement à ce que son nom pourrait faire croire, il n'a rien de spécifique à BIND), en se connectant au nœud Namecoin et en chargeant toute la liste des noms :

```
% php namescan.php
PHP Warning:  PHP Startup: Unable to load dynamic library '/usr/lib/php5/20100525+lfs/suhosin.so' - /usr/lib/php
New blocks : 1
New names : 22
Write : /home/stephane/namecoin/NamecoinToBind/cache/getinfo_seri
New domains : 1
Write : /home/stephane/namecoin/NamecoinToBind/cache/names_block_seri
Write : /home/stephane/namecoin/NamecoinToBind/cache/bind_tree_seri
PHP Notice:  Undefined variable: templateFile in /home/stephane/namecoin/NamecoinToBind/namescan.php on line 106
Write : /etc/bind/dotbit/db.namecoin.bit
PHP Notice:  Undefined variable: statDir in /home/stephane/namecoin/NamecoinToBind/namescan.php on line 130
```

On peut ensuite dire au serveur de noms de charger ce fichier `/etc/bind/dotbit/db.namecoin.bit`. ici, pour BIND :

```
zone "bit" {
    type master;
    file "/etc/bind/dotbit/db.namecoin.bit";
};
```

Attention, plusieurs noms dans Namecoin ont une syntaxe incorrecte (pas pour le DNS, qui est tolérant, mais pour les règles plus strictes que BIND suit par défaut) :

```
Jan 18 17:29:17 ludwigVII named[23967]: /etc/bind/dotbit/db.namecoin.bit:15421: jdt_test.bit: bad owner name (ch
```

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc1035.txt`

Il doit y avoir une option de BIND pour être plus laxiste mais je ne l'ai pas sous la main, j'ai donc supprimé manuellement. Évidemment, en production, il faudra trouver une solution plus stable (et faire tourner `php namescan.php` depuis cron).

Notez qu'il existe une solution pratique pour accéder aux sites Web en `.bit` sans configurer son résolveur, mais en utilisant un relais en `bit.pe`. Essayez, par exemple <http://bortzmeyer.bit.pe>.

Parmi les fonctions avancées (et, à ma connaissance, absolument pas déployées) de Namecoin, on peut noter un équivalent de DANE <https://www.bortzmeyer.org/jres-dane-2011.html>, documenté en <https://github.com/namecoin/wiki/wiki/Domain-Name-Specification-2.0> et discuté en <http://blog.mediocregopher.com/namecoind-ssl.html> (discussion bien vieille ?) ou encore <http://dot-bit.org/forum/viewtopic.php?f=5&t=1137>.

Merci notamment aux participants au canal IRC `#namecoin` sans lesquels je n'y serais pas arrivé. Si vous voulez davantage d'informations, commencez par le site « officiel » <http://namecoin.info/>.