RFC 5732 : Extensible Provisioning Protocol (EPP) Host Mapping

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 13 octobre 2009

Date de publication du RFC : Août 2009

https://www.bortzmeyer.org/5732.html

La représentation des **serveurs de noms** ("*host*", dans ce contexte) dans un registre de noms de domaine a toujours été une source de confusion et de désaccords. Le protocole EPP d'avitaillement ("*provisioning*") d'un registre a tranché arbitrairement et décidé que la bonne méthode était d'avoir des objets « serveur de noms » ("*host*") explicitement mis dans le registre. C'est ce que normalise notre RFC, successeur du RFC 4932 [1], qui lui-même succédait au RFC 3732.

Un registre de noms de domaine stocke en effet au moins deux classes d'objets : les domaines, bien sûr, et les contacts, les entités (personnes physiques ou organisations) qui gèrent les domaines. Mais cela laisse ouverte la question des serveurs de noms. Pour pouvoir déléguer un domaine, le registre a besoin de ces serveurs, qui se retrouveront en partie droite des enregistrements de type NS, comme ici, dans le registre de .org :

```
example.org.      IN     NS    ns1.example.org.
                  IN     NS    galadriel.lothlorien.net.
```

Comme souvent lors de l'élaboration d'un schéma de données, on peut se poser la question : objet ou attribut ? Les serveurs de noms doivent-ils être des objets « de première classe », gérés en tant que tels, accessibles via whois ou bien doivent-ils être de simples **attributs** des objets de la classe domaine ?

Les deux approches sont possibles. .com utilise la première. Un serveur de noms est un objet de première classe, vous pouvez le voir avec whois :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4932.txt

```
% whois ns.kimsufi.com
...
   Server Name: NS.KIMSUFI.COM
   IP Address: 213.186.33.199
   Registrar: OVH
```

D'autres registres ont choisi de faire des serveurs de noms de simples attributs. Quelle approche fallait-il retenir pour le protocole d'avitaillement EPP, normalisé dans le RFC 5730 ? Celui-ci sépare le protocole proprement dit de la définition des classes d'objets (classes nommées, dans EPP, "*mappings*". Il existe une classe (un "*mapping*") pour les domaines (RFC 5731), une pour les contacts (RFC 5733) et notre RFC 5732 pour les serveurs de noms. Toutes sont optionnelles. Un registre n'est pas obligé de mettre en œuvre tous ces "*mappings*" et peut donc, s'il ne gère pas les objets "*hosts*", ignorer le RFC 5732.

Si un registre choisit, par contre, de gérer des objets « serveur de noms » comme dans ce RFC, la section 1 décrit les relations entre ces objets et les domaines. Ainsi, tout serveur de noms est subordonné à un domaine (le parent) : `ns1.example.org` est subordonné à `example.org` et la relation doit être conservée par EPP (par exemple, l'objet "*host*" ne peut être créé que par le client EPP qui gère l'objet domaine parent). À noter que le parent peut être **externe** au registre (par exemple `galadriel.lothlorien.net` pour le registre de `.org`).

La section 2 de ce RFC énumère ensuite les attributs de l'objet « serveur de noms ». Le serveur a un **nom** (par exemple `ns2.example.net`), conforme aux règles syntaxiques du RFC 1123. Comme tous les objets manipulés avec EPP, il a un identificateur unique spécifique à EPP, le "*client identifier*" (voir le RFC 5730). Il a aussi un **état** ("*status*"), qui peut être une combinaison par exemple `ok` combiné avec `linked` (qui indique qu'il est utilisé dans au moins un domaine).

Il a enfin une adresse IP **facultative**. Le RFC recommande de ne la stocker que si elle est nécessaire pour publier la **colle**, les enregistrements qui permettent de trouver l'adresse IP d'un serveur de noms qui sert la zone dont il fait lui-même partie par exemple dans :

```
example.org.      IN     NS    ns1.example.org.
                  IN     NS    galadriel.lothlorien.net.
```

Ici, `ns1.example.org` est dans la zone qu'il sert (`example.org`), il faut donc transmettre au registre son adresse IP, pour qu'il puisse publier la colle :

```
ns1.example.org.          IN   AAAA    2001:db8:314::1:53
```

alors que cela n'est pas nécessaire pour `galadriel.lothlorien.net`. Les RFC 2874 et RFC 3596 contiennent des détails sur cette question. La section 3.2.1 de notre RFC, sur la commande `<create>` revient également sur ce point en insistant que cette commande n'impose pas de transmettre des adresses IP, bien au contraire.

Le cœur d'EPP est décrit dans le RFC 5730, qui inclus une description des commandes possibles (comme `<create>` ou `<delete>`). Toutes ne s'appliquent pas à tous les objets et chaque norme d'un "*mapping*" doit donc décrire quelles commandes ont un sens pour lui. C'est ici l'objet de la section 3. Par exemple (section 3.1.3), le passage d'un "*registrar*" à un autre (`transfer`) n'a pas de sens pour un objet « serveur de noms » et n'est donc pas défini. L'espace de noms XML du "*host mapping*", de notre classe « serveur de noms » est `urn:ietf:params:xml:ns:host-1.0` (voir section 6).

Les commandes `<check>` et `<info>` ont leur sens habituel (section 3.1), celui de récupérer des informations sur un objet, ici en donnant son nom. Voici l'exemple donné par le RFC pour la réponse à une commande `<info>` pour le serveur `ns1.example.com` :

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <host:infData
       xmlns:host="urn:ietf:params:xml:ns:host-1.0">
        <host:name>ns1.example.com</host:name>
        <host:roid>NS1_EXAMPLE1-REP</host:roid>
        <host:status s="linked"/>
        <host:status s="clientUpdateProhibited"/>
        <host:addr ip="v4">192.0.2.2</host:addr>
        <host:addr ip="v4">192.0.2.29</host:addr>
        <host:addr ip="v6">1080:0:0:0:8:800:200C:417A</host:addr>
        <host:clID>ClientY</host:clID>
        <host:crID>ClientX</host:crID>
        <host:crDate>1999-04-03T22:00:00.0Z</host:crDate>
        <host:upID>ClientX</host:upID>
        <host:upDate>1999-12-03T09:00:00.0Z</host:upDate>
        <host:trDate>2000-04-08T09:00:00.0Z</host:trDate>
      </host:infData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>54322-XYZ</svTRID>
    </trID>
  </response>
</epp>
```

Les informations spécifiques à notre classe sont dans l'espace de noms `urn:ietf:params:xml:ns:host-1.0` dont le préfixe est ici `host`.

La syntaxe formelle complète de cette classe figure dans la section 4, sous la forme d'un schéma W3C.

L'annexe A rassemble les changements depuis le RFC 4932. Les changements, à part la mise à jour des RFC cités en référence, consistent surtout en une nouvelle licence pour le schéma XML et une précision sur le code de retour 2201 (permission refusée).