# RFC 7745 : XML Schemas for Reverse DNS Management

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 21 janvier 2016

Date de publication du RFC : Janvier 2016

https://www.bortzmeyer.org/7745.html

---

Ce court RFC décrit un schéma XML qu'utilise l'ICANN pour gérer les zones DNS dans les domaines `ip6.arpa` et `in-addr.arpa`. Rien de standard, juste la documentation d'une pratique locale.

Depuis 2011, l'ICANN utilise ce système pour gérer et produire les zones DNS dont elle a la responsabilité sous `.arpa` (`in-addr.arpa` est documenté dans le RFC 1034 [1] et `ip6.arpa` dans le RFC 3152). L'ICANN délègue des zones aux RIR et ceux-ci souhaitent un système le plus fiable, simple et prévisible possible.

La gestion de ces zones se faisait, il y a fort longtemps, à la main, avec un éditeur ordinaire, mais cela n'est évidemment plus possible depuis le déploiement de DNSSEC : les clés changent, les enregistrements DS ("*Delegation Signer*") doivent être mis à jour, etc. Et la plus petite erreur casse la validation cryptographique. Bref, pour ce service comme pour d'autres, DNSSEC pousse fortement à automatiser sérieusement les processus.

Le principe de base (section 3, le gros du RFC) est que le RIR prépare sa demande sous forme d'un fichier XML conforme au schéma décrit dans ce RFC, et l'envoie à l'ICANN par une requête REST. Elle est évidemment en HTTPS et authentifiée par un certificat client (et un pour le serveur, également). L'AC est l'ICANN elle-même. Recevant le XML, l'ICANN met à jour automatiquement les zones sous `.arpa`.

En bonne logique REST, les requêtes HTTP sont `GET`, `PUT` et `DELETE`. `GET` sert à interroger l'état actuel de la base (et le RIR n'utilise donc que l'URI, il n'envoie pas de XML).

Voici un exemple d'une requête en XML qui pourrait être envoyée (requête `PUT`) pour mettre à jour la zone `10.in-addr.arpa`, avec deux enregistrements DS (pour la même clé, la 33682) :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc1034.txt`

```
<zone xmlns="http://download.research.icann.org/rdns/1.1"
     name="10.in-addr.arpa" cust="IANA" ipversion="ipv4"
     version="1.1" modified="2012-01-18T01:00:06"
     state="active" href="https://host.example.org/ipv4/10">
     <nserver>
       <fqdn>BLACKHOLE-1.IANA.ORG.</fqdn>
     </nserver>
     <nserver>
       <fqdn>BLACKHOLE-2.IANA.ORG.</fqdn>
     </nserver>
     <ds>
       <rdata>33682 5 1 ea8afb5fce7caf381ab101039</rdata>
     </ds>
     <ds>
       <rdata>33682 5 2 7d44874f1d93aaceb793a88001739a</rdata>
     </ds>
   </zone>
```

Notez que la modification n'est pas forcément instantanée : il peut y avoir un système de vérification manuelle et d'approbation explicite.

Les URL utilisés par le client sont variables. Par exemple, pour mettre à jour le domaine `10.in-addr.arpa` cité plus haut, l'URL serait `http://icann.example/4/10.in-addr.arpa` (en remplaçant `icann.example` par le vrai nom du serveur REST à l'ICANN). Pour avoir une liste des demandes en attente (rappelez-vous que le système n'est pas synchrone), le client ferait un GET sur `http://icann.example/queuelist`, etc.

Les annexes A et B du RFC donnent le schéma Relax NG complet des éléments XML possibles. Par exemple, une zone DNS se définit ainsi :

```
# A single zone record
zone = element zone {
  # The zone record's name, eg 10.in-addr.arpa
  attribute name { text },
  ...
  # The administrative state of the zone (optional)
  attribute state { "active" | "pending" | "error" }?,
  # The last modified timestamp in UTC (optional)
  attribute modified  { xsd:dateTime }?,
  ...
  # A zone NS RRset MUST have at least two NS records
  nserver,
  nserver+,
  # It MAY contain some DS records
  ds*
}
```

À noter que ce schéma ne permet pas d'indiquer de la colle (adresses IP de serveurs de noms situés dans la zone elle-même) puisqu'il ne sert que pour les zones sous `.arpa` (on ne voit jamais de serveurs de noms nommés ainsi).

Pourquoi développer un tel système plutôt que d'utiliser la norme EPP ? Le RFC ne le dit pas mais on peut supposer que c'est parce qu'EPP est bien plus complexe.