RFC 9874 : Best Practices for Deletion of Domain and Host Objects in the Extensible Provisioning Protocol (EPP)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 5 octobre 2025

Date de publication du RFC : Septembre 2025

https://www.bortzmeyer.org/9874.html

Dans un registre de noms de domaine, il existe une classe d'objets pour les domaines et parfois une pour les serveurs de noms ("*hosts*"). C'est en se basant sur les objets de ces classes que des informations sont ensuite publiées dans le DNS. Que se passe-t-il si on retire un objet de ces classes, alors que d'autres objets en dépendaient ? Va-t-on scier une branche sur laquelle quelqu'un est assis ? Ce RFC fait le point sur les solutions existantes, notant que certaines sont moins bonnes que d'autres, notamment pour la sécurité.

Prenons un exemple simple : le domaine `monbeaudomaine.example` est enregistré auprès du registre du TLD `.example`. Ses serveurs de noms sont `ns1.domaine-d-un-copain.example` et `ns1.hébergeur.example` et on va supposer que le registre de `.example` traite les serveurs de noms comme une classe distincte dans sa base de données. Maintenant, supposons que le domaine `domaine-d-un-copain.exa` soit supprimé parce que son titulaire n'en voit plus l'utilité. Que va devenir le serveur de noms `ns1.domaine-d-un-copa` Il existe de nombreuses façons de traiter ce problème, et c'est le rôle de ce RFC de les analyser toutes.

Ainsi, la section 3.2.2 du RFC 5731 [1] dit que ce n'est pas bien de détruire un domaine si des objets de type serveur de noms sont sous ce domaine. Dans l'exemple précédent, le registre de `.example` aurait refusé la suppression de `domaine-d-un-copain.example`. Mais cela laisse le problème entier : si le titulaire ne veut plus payer et donc veut supprimer le domaine, que faire ?

Un cas similaire se produit si on veut supprimer un serveur de noms. Si le client EPP demande au serveur EPP du registre de `.example` la suppression de l'objet `ns1.domaine-d-un-copain.example`, que doit faire le registre, sachant que ce serveur de noms est utilisé par `monbeaudomaine.example` ?

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc5731.txt`

La section 3.2.2 du RFC 5732 dit que le registre devrait refuser la suppression. C'est d'autant plus gênant que, dans le modèle RRR ("*Registrant-Registrar-Registry*"), domaine et serveur(s) peuvent être gérés par des BE différents, n'ayant pas le droit de toucher aux objets des autres BE.

Vous pouvez trouver des bonnes explications et des exemples réels dans les supports d'une présentation qui avait été faite à l'IETF <https://datatracker.ietf.org/doc/slides-115-irtfopen-risky-bizness

Quelles sont donc les recommandations concrètes de ce RFC ? La section 6 les résume. Au choix :

— Renommer les serveurs de noms vers le nom d'un serveur maintenu par le client EPP (le BE), comme décrit dans la section 5.1.3.4.
— Demander au client EPP de supprimer les serveurs de noms mais avec possibilité de rétablissement (RFC 3915) comme décrit en section 5.2.2.3.
— Renommer les serveurs de noms dans un nom de domaine spécial, dont il est garanti qu'il ne sera jamais utilisé, comme décrit en section 5.1.4.3. Pour éviter d'avoir à créer une nouvelle entrée dans le registre des noms de domaine spéciaux <https://www.iana.org/assignments/special-use-domain-names/special-use-domain-names.xml#special-use-domain>, le nom `sacrificial.invalid`, utilisant un TLD existant, est recommandé.

Toutes les autres techniques sont déconseillées. Mais voyons maintenant les détails.

Le protocole EPP permet de changer le nom d'un serveur de noms (section 3.2.5 du RFC 5732). Une pratique déjà observée est donc de renommer les serveurs de noms. Dans l'exemple ci-dessus, recevant la demande de suppression de `domaine-d-un-copain.example`, le registre (ou le BE s'il le peut) renommerait `ns1.domaine-d-un-copain.example` en, mettons, `ns1.domaine-d-un-copain.renamed.inval` Ici, `.invalid`, TLD réservé par le RFC 6761, ne poserait pas vraiment de problème mais renommer dans un domaine ouvert à l'enregistrement pourrait créer un risque de sécurité, si le domaine de destination n'existe pas, un méchant pouvant alors l'enregistrer et être ainsi en mesure de répondre aux requêtes DNS.

La section 3 du RFC explique brièvement pourquoi les RFC 5731 et RFC 5732 déconseillent fortement de permettre la suppression d'un domaine tant que des serveurs de noms dans ce domaine existent. Il y a deux risques de sécurité si on détruisait le domaine en laissant les serveurs de noms tels quels dans la base de donnés du registre : un de déni de service (le nom ne se résout plus et le serveur va donc être inutile) et un de détournement (si un méchant peut ré-enregistrer le nom de domaine supprimé). Il y a aussi le problème de la colle <https://www.afnic.fr/observatoire-ressources/papier-expert/le-dns-ca-colle-ou-ca-ne-colle-pas/> orpheline, décrit dans le rapport du SSAC, « "*Comment on Orphan Glue Records in the Draft Applicant Guidebook*" <https://itp.cdn.icann.org/en/files/security-and-stability-advisory-committee-ssac-reports/sac-048-en.pdf> ».

Si la clé qui identifie un serveur de noms est un numéro quelconque et pas son nom, on peut renommer le serveur sans changer les délégations, ce qui est particulièrement utile si le client EPP n'a pas le droit de changer des délégations qui appartiennent à un autre client du registre. Le nouveau nom ne va en général pas être associé à un serveur opérationnel : on sacrifie un serveur pour pouvoir supprimer le domaine parent. Mais cela entraine quelques risques (section 4 du RFC et l'article d' Akiwate, G., Savage, S., Voelker, G., et K. Claffy, « "*Risky BIZness : Risks Derived from Registrar Name Management*" <https://doi.org/10.1145/3487552.3487816> »). Si on renomme vers un nom actuellement inexistant, le domaine peut être détourné si un malveillant enregistre ensuite ce domaine.

Compte tenu de tout cela, la section 5 du RFC étudie les différentes pratiques possibles, leurs avantages et leurs inconvénients. Pour les illustrer, je vais utiliser une base de données simple, décrite en SQL (les essais ont été faits avec PostgreSQL). Voici par exemple une création d'une telle base :

```
CREATE TABLE Domains (name TEXT UNIQUE);

-- Ici, la table Nameservers n'offre aucune valeur ajoutée par rapport
-- au fait de tout mettre dans Domains. Mais ce ne sera pas le cas par
-- la suite.
CREATE TABLE Nameservers (name TEXT UNIQUE);

CREATE TABLE Delegation (domain TEXT REFERENCES Domains(name),
                         server TEXT REFERENCES Nameservers(name));

INSERT INTO Domains VALUES ('monbeaudomaine.example');
INSERT INTO Domains VALUES ('domaine-d-un-copain.example');

INSERT INTO Nameservers VALUES ('ns1.domaine-d-un-copain.example');
INSERT INTO Nameservers VALUES ('ns1.hébergeur.example');

INSERT INTO Delegation VALUES ('monbeaudomaine.example', 'ns1.domaine-d-un-copain.example');
INSERT INTO Delegation VALUES ('monbeaudomaine.example', 'ns1.hébergeur.example');
```

Dans ce premier cas très simple, la suppression du domaine `domaine-d-un-copain.example` est triviale :

```
registry=> DELETE FROM Domains WHERE name='domaine-d-un-copain.example';
DELETE 1
```

Mais elle laisse la possibilité de colle orpheline et surtout d'un détournement de `monbeaudomaine.example` si quelqu'un ré-enregistre `domaine-d-un-copain.example`. Ce premier essai n'est pas conforme aux exigences des RFC 5731 et RFC 5732. On va essayer de faire mieux.

Si on interdit (à juste titre) la suppression d'un domaine lorsque des serveurs de noms sont nommés dans ce domaine, on peut arriver à supprimer un domaine en supprimant d'abord les serveurs de noms qui sont nommés dans ce domaine (section 5.2) :

```
CREATE TABLE Domains (name TEXT UNIQUE);

-- Ajout d'une dépendance au domaine parent, pour éviter les suppressions.
CREATE TABLE Nameservers (name TEXT UNIQUE, parent TEXT REFERENCES Domains(name));

CREATE TABLE Delegation (domain TEXT REFERENCES Domains(name),
                         server TEXT REFERENCES Nameservers(name));

INSERT INTO Domains VALUES ('monbeaudomaine.example');
INSERT INTO Domains VALUES ('domaine-d-un-copain.example');
INSERT INTO Domains VALUES ('hébergeur.example');

-- Pour une vraie base, on écrirait du code SQL qui extrait le parent
-- automatiquement.
INSERT INTO Nameservers VALUES ('ns1.domaine-d-un-copain.example', 'domaine-d-un-copain.example');
INSERT INTO Nameservers VALUES ('ns1.hébergeur.example', 'hébergeur.example');

INSERT INTO Delegation VALUES ('monbeaudomaine.example', 'ns1.domaine-d-un-copain.example');
INSERT INTO Delegation VALUES ('monbeaudomaine.example', 'ns1.hébergeur.example');

registry=> DELETE FROM Domains WHERE name='domaine-d-un-copain.example';
ERROR:  update or delete on table "domains" violates foreign key constraint "nameservers_parent_fkey" on table "
DETAIL:  Key (name)=(domaine-d-un-copain.example) is still referenced from table "nameservers".
-- Ce comportement est ce que recommandent les RFC 5731 et 5732.
```

```
-- Cela oblige le client à supprimer les serveurs de noms d'abord, ce qui à
-- son tour nécessite potentiellement de changer les délégations :

registry=> DELETE FROM Delegation WHERE server = 'ns1.domaine-d-un-copain.example';
DELETE 1
registry=>  DELETE FROM Nameservers WHERE name='ns1.domaine-d-un-copain.example';
DELETE 1
registry=>  DELETE FROM Domains WHERE name='domaine-d-un-copain.example';
DELETE 1
```

Ici, il y a eu suppression explicite des serveurs de noms par le client (section 5.2.2.1). Cela peut poser des problèmes de permission, dans le cadre du système RRR, si tous les objets ne sont pas chez le même BE. Mais la suppression explicite est une des trois solutions recommandées, notamment si on ajoute la possibilité de rétablir l'état précédent (commande EPP `<restore>`, RFC 3915).

On peut aussi envisager une suppression implicite (section 5.2.1.1), le registre se chargeant du nettoyage (c'est le rôle de la directive SQL `ON DELETE CASCADE`) :

```
CREATE TABLE Domains (name TEXT UNIQUE);

CREATE TABLE Nameservers (name TEXT UNIQUE,
                          parent TEXT REFERENCES Domains(name) ON DELETE CASCADE);

CREATE TABLE Delegation (domain TEXT REFERENCES Domains(name) ON DELETE CASCADE,
                         server TEXT REFERENCES Nameservers(name) ON DELETE CASCADE);

INSERT INTO Domains VALUES ('monbeaudomaine.example');
INSERT INTO Domains VALUES ('domaine-d-un-copain.example');
INSERT INTO Domains VALUES ('hébergeur.example');

-- Pour une vraie base, on écrirait du code SQL qui extrait le parent
-- automatiquement.
INSERT INTO Nameservers VALUES ('ns1.domaine-d-un-copain.example', 'domaine-d-un-copain.example');
INSERT INTO Nameservers VALUES ('ns1.hébergeur.example', 'hébergeur.example');

INSERT INTO Delegation VALUES ('monbeaudomaine.example', 'ns1.domaine-d-un-copain.example');
INSERT INTO Delegation VALUES ('monbeaudomaine.example', 'ns1.hébergeur.example');

registry=> SELECT Domains.name,Nameservers.name FROM Domains,Delegation,Nameservers WHERE Delegation.domain=
          name          |              name
------------------------+--------------------------------
monbeaudomaine.example | ns1.domaine-d-un-copain.example
monbeaudomaine.example | ns1.hébergeur.example
(2 rows)

registry=>  DELETE FROM Domains WHERE name='domaine-d-un-copain.example';
DELETE 1

-- Serveur de noms et délégation ont été détruits en cascade. Ce n'est
-- pas déraisonnable mais c'est quand même un peu effrayant.

registry=> SELECT Domains.name,Nameservers.name FROM Domains,Delegation,Nameservers WHERE Delegation.domain=
          name          |         name
------------------------+----------------------
monbeaudomaine.example | ns1.hébergeur.example
(1 row)
```

Cette solution est simple et efficace mais détruire implicitement des objets de la base de données peut inquiéter les responsables de cette base. Et cela peut laisser un domaine avec trop peu de serveurs de

noms pour assurer sa continuité de service voire, dans le pire des cas, sans serveurs du tout. Et il serait bon de prévenir le client de cette suppression implcite, par exemple par le mécanisme de "*poll*" d'EPP (RFC 8590).

Si on interdit (à juste titre, le RFC le recommande) la suppression d'un domaine lorsque des serveurs de noms sont nommés dans ce domaine, une solution possible est de renommer les serveurs avant de supprimer le domaine (section 5.1). Le nouveau nom permet d'indiquer clairement la raison du renommage. Mais ce renommage laisse dans la base des « déchets » qu'il faudra nettoyer un jour. Cette catégorie contient de nombreuses variantes. Par exemple, on peut renommer dans un TLD spécial (RFC 6761), ici, `.invalid` :

```
CREATE TABLE Domains (name TEXT UNIQUE);

-- On introduit un identificateur du server de noms qui n'est *pas*
-- son nom, pour permettre le renommage.
CREATE TABLE Nameservers (id SERIAL UNIQUE, name TEXT UNIQUE,
                                parent TEXT REFERENCES Domains(name));

CREATE TABLE Delegation (domain TEXT REFERENCES Domains(name),
                         server INTEGER REFERENCES Nameservers(id));

INSERT INTO Domains VALUES ('monbeaudomaine.example');
INSERT INTO Domains VALUES ('domaine-d-un-copain.example');
INSERT INTO Domains VALUES ('hébergeur.example');
-- Pour le renommage, un nom qui indique clairement le but :
INSERT INTO Domains VALUES ('renamed.invalid');

-- Pour une vraie base, on écrirait du code SQL qui extrait le parent
-- automatiquement.
INSERT INTO Nameservers (name, parent) VALUES ('ns1.domaine-d-un-copain.example', 'domaine-d-un-copain.example')
INSERT INTO Nameservers (name, parent) VALUES ('ns1.hébergeur.example', 'hébergeur.example');

INSERT INTO Delegation VALUES ('monbeaudomaine.example',
     (SELECT id FROM Nameservers WHERE name='ns1.domaine-d-un-copain.example'));
INSERT INTO Delegation VALUES ('monbeaudomaine.example',
     (SELECT id FROM Nameservers WHERE name='ns1.hébergeur.example'));

registry=> SELECT Domains.name,Nameservers.name FROM Domains,Delegation,Nameservers WHERE Delegation.domain=Doma
          name          |              name
------------------------+---------------------------------
monbeaudomaine.example | ns1.domaine-d-un-copain.example
monbeaudomaine.example | ns1.hébergeur.example
(2 rows)

registry=> UPDATE Nameservers SET name='ns1.domaine-d-un-copain.example.renamed.invalid', parent='renamed.invali
UPDATE 1
registry=> DELETE FROM Domains WHERE name='domaine-d-un-copain.example';
DELETE 1

registry=> SELECT Domains.name,Nameservers.name FROM Domains,Delegation,Nameservers WHERE Delegation.domain=Doma
          name          |                      name
------------------------+-------------------------------------------------
monbeaudomaine.example | ns1.domaine-d-un-copain.example.renamed.invalid
monbeaudomaine.example | ns1.hébergeur.example
(2 rows)
```

Par contre, il ne faut pas utiliser `.alt`, qui est explicitement réservé aux protocoles non-DNS (RFC 9476). Notez que certains serveurs EPP peuvent tester le nom des serveurs de noms et refuser des TLD « inconnus ».

Dans la nature, on a pu observer d'autres pratiques, comme de renommer dans un sous-domaine de `as112.arpa`, nom garanti ne pas exister (RFC 7535), mais qui n'est pas censé servir à cela (RFC 6305).

On a vu aussi des renommages vers des résolveurs <https://www.bortzmeyer.org/resolveur-dns.html> DNS publics, ce qui est également une horreur, la délégation doit être faite vers des serveurs faisant autorité <https://www.bortzmeyer.org/serveur-dns-faisant-autorite.html>, pas des résolveurs <https://www.bortzmeyer.org/resolveur-dns.html>.

Certains clients EPP maintiennent des serveurs de noms actifs qui servent pour le renommage. Cela fait davantage de travail mais cela protège contre le détournement. Le DNS va continuer à fonctionner normalement. On pourrait aussi imaginer des serveurs de noms actifs, répondant NXDOMAIN (« ce domaine n'existe pas »), qui soient gérés collectivement (section 5.1.4.4) ; un tel service serait certainement utile. On pourrait même créer un nom de domaine pour ce service (`sacrificial.arpa` ?) mais personne ne l'a encore fait. Pour l'instant, la solution des serveurs maintenus par le client EPP (section 5.1.3.4) fait partie des trois solutions recommandées. Le client prudent doit bien verrouiller le domaine dans lequel ces serveurs sont nommés (enregistrement multi-années, verrouillage par le registre, etc).

On peut aussi renommer les serveurs de noms vers un nom non-existant dans un TLD existant. Ça s'est déjà vu mais il ne faut surtout pas faire cela : un attaquant pourrait enregistrer le nom et capter ainsi le trafic (`.invalid` n'a pas ce problème). Idem si le nom n'est pas sous votre contrôle. Un exemple est donné par le domaine `lenvol.re` : ses serveurs de noms étaient `ns.hostin.io`, `ns.hostin.mu` et `ns.hostin.re`. Lors de la suppression de `hostin.re` en octobre 2024, le dernier serveur de noms a été renommé `host1.renamedbyregistry.com` (et, en dépit du nom, pas par le registre). Ce domaine `renamedbyregistry.com` étant enregistré, et par un autre BE, on voit le risque.

```
%  whois lenvol.re
domain:                         lenvol.re
status:                         ACTIVE
...
nserver:                        host1.renamedbyregistry.com
nserver:                        ns.hostin.io
nserver:                        ns.hostin.mu
```

D'autres noms qui utilisaient ce même serveur ont également le problème :

```
% dig @d.nic.fr savanna.re. NS
...
;; AUTHORITY SECTION:
savanna.re.  3600 IN NS host1.renamedbyregistry.com.
savanna.re.  3600 IN NS ns.hostin.mu.
savanna.re.  3600 IN NS ns.hostin.io.

;; Query time: 8 msec
;; SERVER: 2001:678:c::1#53(d.nic.fr) (UDP)
;; WHEN: Tue Jun 24 14:33:32 CEST 2025
```

En lecture supplémentaire, notre RFC recommande le rapport « *"SSAC 125 "Report on Registrar Nameserver Management""* <https://itp.cdn.icann.org/en/files/security-and-stability-advisory-committee-sac-125-09-05-2024-en.pdf> », ainsi que l'article « *"Risky BIZness : Risks Derived from Registrar Name Management"* <https://doi.org/10.1145/3487552.3487816> ».