RFC 8244 : Special-Use Domain Names Problem Statement

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 octobre 2017

Date de publication du RFC : Octobre 2017

https://www.bortzmeyer.org/8244.html

Le RFC 6761 [1] créait un registre <https://www.iana.org/assignments/special-use-domain-names/special-use-domain-names.xml> de « noms de domaines d'usage spécial », où l'IETF enregistrait des noms qui pouvaient nécessiter un traitement spécial par une des entités de la chaîne d'avitaillement et de résolution des noms de domaines. Un exemple est celui des noms de domaine qui n'utilisaient pas le DNS, comme le `.onion` du RFC 7686. Certaines personnes ont émis des réserves sur ce registre, par exemple parce qu'il marchait sur une pelouse que l'ICANN considère comme sienne. Ce nouveau RFC, très controversé, fait la liste de tous les reproches qui ont été faits au RFC 6761 et à son registre des noms spéciaux.

La résolution de noms - partir d'un nom de domaine et obtenir des informations comme l'adresse IP - est une des fonctions cruciales de l'Internet. Il est donc normal de la prendre très au sérieux. Et les noms ont une valeur pour les utilisateurs (la vente record semble être celle de `business.com`). Souvent, **mais pas toujours**, cette résolution se fait avec le DNS. Mais, je l'ai dit, pas toujours : il ne faut pas confondre les noms de domaines (une organisation arborescente, une syntaxe, les composants séparés par des points, par exemple `_443._tcp.www.bortzmeyer.org.`, `réussir-en.fr.` ou `sjnrk23rmcl4ie5atmz664v7o7k5nkk4jh7mm6lor2n4hxz2tos3eyid.onion.`) et le DNS, un protocole réseau particulier (section 2 du RFC, pour la terminologie). La section 2 de ce RFC, suivant la définition du RFC 8499, ne considère pas qu'un nom de domaine utilise forcément le DNS. (Le problème n° 20, en section 3 revient sur cette question.)

Le registre des noms de domaines spéciaux <https://www.iana.org/assignments/special-use-domain-nar
special-use-domain-names.xml> a été créé en 2013 par le RFC 6761. Contrairement à ce que croient beaucoup de gens, il ne contient pas que des TLD (par exemple, il a aussi `a.e.f.ip6.arpa` ou bien `example.org`). Depuis sa création, plusieurs noms ont été ajoutés, le plus spectaculaire ayant été

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc6761.txt`

le .onion en 2015, via le RFC 7686. Beaucoup d'autres noms ont été proposés, sans être explicitement rejetés mais sans être acceptés non plus (voir les "*Internet-Drafts*" draft-grothoff-iesg-special-use-p2p-gns et draft-chapin-additional-reserved-tlds). Ces enregistrements ne se sont pas très bien passés, avec beaucoup d'engueulades, notamment pour les TLD.

En effet, la bagarre a toujours fait rage pour la gestion des noms dans la racine du DNS. Il y a cinq sortes de TLD :

— Ceux qui sont mis dans la racine publique du DNS, comme .bo, .net, .guru, etc. Contrairement à ce que dit le RFC, ils ne sont pas forcément délégués par l'ICANN (.de a été créé bien avant l'ICANN, et .su résiste depuis longtemps à toutes les tentatives de l'ICANN de le supprimer).
— Ceux gérés par l'IETF « pour des raisons techniques », et qui sont parfois délégués dans la racine publique, le plus connu étant .arpa. Les domaines du registre des noms spéciaux en font partie. Les fanas de gouvernance noteront qu'ils ne passent jamais par l'ICANN.
— Ceux que l'ICANN bloque, pour des raisons variées (cf. leurs règles <https://newgtlds.icann.org/en/applicants/agb/guidebook-full-04jun12-en.pdf>, notamment sections 2.2.1.2.1 ou 2.2.1.4.1).
— Ceux qui sont utilisés en dehors de la racine publique du DNS, comme .eth pour Ethereum ou .bit pour Namecoin. Cela inclut aussi des noms très utilisés localement comme .home ou .corp.
— Et ceux qui sont libres, que personne n'utilise encore. Demain, peut-être que .macron sera utilisé par les fans du Président ?

Cette classification est imparfaite, comme toutes les classifications <http://www.paperblog.fr/1962135/borges-classer-avec-classe/>. Par exemple, un nom peut être sur la liste noire de l'ICANN et être quand même utilisé par des organisations locales, ou via un logiciel particulier.

Ce RFC veut présenter une liste exhaustive des problèmes posés lors de l'enregistrement de ces noms spéciaux. Comme elle est exhaustive (« "*an unfiltered compilation of issues*" »), elle n'est pas consensuelle, loin de là ! Plusieurs des problèmes ne sont pas considérés comme tels par tout le monde, ce que le RFC note bien dans les sections 1 et 3.

Le gros du RFC est donc cette liste de problèmes, en section 3. Je ne vais pas tous les citer. Notez tout de suite que ce RFC décrit des problèmes, ou perçus comme tels par certains, pas des solutions. Il ne propose pas de réforme du RFC 6761. Notez aussi que les problèmes ne sont pas classés (leur ordre d'exposition, et le numéro qu'ils reçoivent, n'est pas un ordre d'importance.)

Premier problème cité, la coordination avec l'ICANN. Les cinq catégories de noms de TLD cités plus haut coexistent dans le même espace de noms. S'il y a deux .macron, il y aura un problème (voir par exemple le rapport SAC 090 <https://www.icann.org/en/system/files/files/sac-090-en.pdf>, mais en se rappelant que c'est l'ICANN qui s'auto-défend). L'IETF et l'ICANN ont un mécanisme d'information réciproque <https://www.ietf.org/liaison/> mais pas de processus bureaucratique formel de coordination. Si l'IETF veut réserver .home (RFC 7788), il n'y a officiellement pas de mécanisme pour communiquer cette décision à l'ICANN et obtenir d'elle la confirmation que ce nom n'est pas par ailleurs en cours de réservation. (Cf. 4.1.4 qui revient sur ce mécanisme - « "*liaison*" » en anglais.)

(Personnellement, cet argument me fait plutôt rire : beaucoup de gens sont actifs à la fois à l'IETF et à l'ICANN, les deux organisations opèrent de manière assez publique, surtout l'IETF, et la possibilité qu'elles réservent chacune un nom sans savoir ce qu'a fait l'autre est purement théorique.)

Le problème 2 est celui de la définition de « pour des raisons techniques », introduit dans le RFC 2860, et qui fonde le droit de l'IETF à réserver des TLD sans passer par la caisse de l'ICANN (« "*assignments of domain names for technical uses [...] are not considered to be policy issues, and shall remain subject to*

the provisions of this Section 4" ≫). Le problème est que ces raisons techniques ne sont définies nulle part et que personne ne sait trop ce que cela veut dire.

Les problèmes 3 et 4 sont qu'il n'y a pas de Directeur Chef de l'Internet. Ni l'ICANN, ni l'IETF (ni évidemment l'UIT) ne sont reconnus, ni en droit, ni en fait, comme ayant une autorité quelconque (le raccourci journalistique de présenter l'ICANN comme ≪ le régulateur mondial de l'Internet ≫ est ridiculement faux). Imaginons qu'une développeuse crée un nouveau logiciel qui utilise le TLD `.zigzag` pour son nommage, et diffuse le logiciel en question : personne ne peut l'en empêcher, même si l'ICANN souffre de la perte de revenus potentiels, même si des grognons à l'IETF murmurent bruyamment que c'est du "*squatting*" (voir aussi le début de la section 4). Et heureusement que c'est ainsi : c'est le côté ≪ innover sans autorisation ≫ (≪ "*permissionless innovation*" ≫ qui a été si essentiel pour le succès de l'Internet). Un autre exemple est bien sûr le projet .42 <`https://linuxfr.org/news/lexp%C3%A9rience-42-un-tld-hors-de-la-tutelle-de-licann`>, aujourd'hui abandonné mais qui illustrait ce côté décentralisé de l'Internet.

La définition du problème 5 note que les organisations (ou les individus) utilisent des TLD (ou des domaines quelconques, mais c'est surtout sur les TLD que se focalise la discussion) sans suivre les procédures. Elles ont plusieurs raisons pour agir ainsi :

— Elles ne savent pas qu'il existe des procédures (c'est sans doute le cas le plus fréquent).
— Elles savent vaguement qu'il existe des procédures mais elles se disent que, pour une utilisation purement locale, ce n'est pas important. Le trafic sur les serveurs de noms de la racine montre au contraire que les fuites sont importantes : les noms locaux ne le restent pas. Sans compter le risque de ≪ collisions ≫ entre un nom supposé purement local et une allocation par l'ICANN : `.corp` et `.home` sont des TLD locaux très populaires et il y a des candidatures (actuellement gelées "*de facto*") pour ces TLD à l'ICANN
— Les procédures existent, l'organisation les connait, mais ce n'est pas ouvert. C'est par exemple le cas des TLD ICANN, pour lesquels il n'y a actuellement pas de cycle de candidature, et on n'espère pas le prochain avant 2019 ou 2020.
— Les procédures existent à l'ICANN, l'organisation les connait, c'est ouvert, mais le prix fait reculer (185 000 $ US au précédent cycle ICANN, et uniquement pour déposer le dossier).
— Les procédures existent à l'IETF, l'organisation les connait, c'est ouvert, mais l'organisation refuse délibérement de participer (cas analogue à CARP).
— Les procédures existent, l'organisation les connait, c'est ouvert, mais l'organisation estime, à tort ou à raison, que sa candidature sera refusée (c'est actuellement le cas à l'IETF <`https://www.ietf.org/blog/2015/09/onion/`>).

Ensuite, le problème 6 : il y a plusieurs protocoles de résolution de nom, le DNS n'étant que le principal. En l'absence de métadonnées indiquant le protocole à utiliser (par exemple dans les URL), se servir du TLD comme ≪ aiguillage ≫ est une solution tentante (`if tld == "onion" then use Tor elsif tld == "gnu" then use GnuNet else use DNS...`)

Le registre des noms de domaines spéciaux est essentiellement en texte libre, sans grammaire formelle. Cela veut dire que le code spécifique qui peut être nécessaire pour traiter tous ces domaines spéciaux (un résolveur doit, par exemple, savoir que `.onion` n'utilise pas le DNS et qu'il ne sert donc à rien de l'envoyer à la racine) doit être fait à la main, il ne peut pas être automatiquement dérivé du registre (problème 7). Résultat, quand un nouveau domaine est ajouté à ce registre, il sera traité ≪ normalement ≫ par les logiciels pas mis à jour, et pendant un temps assez long. Par exemple, les requêtes seront envoyées à la racine, ce qui pose des problèmes de vie privée (cf. RFC 7626).

Comme noté plus haut, certains candidats à un nom de domaine spécial sont inquiets du temps que prendra l'examen de leur candidature, et du risque de rejet (problème 8). Ils n'ont pas tort. L'enregistrement du `.local` (RFC 6762) a pris dix ans, et les efforts de Christian Grothoff <`https://grothoff.org/christian/`> pour enregistrer `.gnu` se sont enlisés dans d'épaisses couches de bureaucratie. Ce

n'est peut-être pas un hasard si Apple a fini par avoir son `.local` (non sans mal) alors que des projets de logiciel libre comme GNUnet se sont vu fermer la porte au nez.

C'est que l'IETF n'est pas toujours facile et qu'un certain nombre de participants à cette organisation ont souvent une attitude de blocage face à tout intervenant extérieur (problème 9). Ils n'ont pas tort non plus de se méfier des systèmes non-DNS, leurs raisons sont variées :

— Absence d'indication du mécanisme de résolution donc difficulté supplémentaire pour les logiciels (« c'est quoi, `.zkey` ? »). De plus, un certain nombre de participants à l'IETF estiment qu'il ne faut de toute façon qu'un seul protocole de résolution, afin de limiter la complexité.
— Le problème de la complexité est d'autant plus sérieux que les autres protocoles de résolution de noms n'ont pas forcément la même sémantique que le DNS : ils sont peut-être sensibles à la casse, par exemple.
— Conviction que l'espace de nommage est « propriété » de l'IETF et/ou de l'ICANN, et que ceux qui réservent des TLD sans passer par les procédures officielles sont de vulgaires squatteurs.
— Sans compter le risque, s'il existe un moyen simple et gratuit de déposer un TLD via l'IETF, que plus personne n'alimente les caisses de l'ICANN.
— Et enfin le risque juridique : si Ser Davos fait un procès à l'IETF contre l'enregistrement de `.onion`, cela peut durer longtemps et coûter cher. L'ICANN court le même risque mais, elle, elle a des avocats en quantité, et de l'argent pour les payer.

Le problème 11 est davantage lié au RFC 6761 lui-même. Ce RFC a parfois été compris de travers, comme pour l'enregistrement de `ipv4only.arpa` (RFC 7050, corrigé ensuite dans le RFC 8880) et surtout celui de `.home` (RFC 7788, qui citait le RFC 6761 mais n'avait tenu aucune de ses obligations). Voir aussi le problème 16.

Les problèmes 12 et 13 concernent les TLD existants (au sens où ils sont utilisés) mais pas enregistrés officiellement (voir par exemple le rapport de l'ICANN sur les « collisions » <`https://www.icann.org/en/system/files/files/name-collision-02aug13-en.pdf`>). Cela serait bien de documenter quelque part cette utilisation, de façon à, par exemple, être sûrs qu'ils ne soient pas délégués par l'ICANN. Mais cela n'a pas marché jusqu'à présent.

Le problème 14 concerne le fait que des noms de domaine spéciaux sont parfois désignés comme spéciaux par leur écriture dans le registre, mais parfois simplement par leur délégation dans le DNS.

Le problème 15 est celui de la différence entre enregistrer un usage et l'approuver. Avec la règle « géré par l'IETF pour un usage technique » du RFC 2860, il n'y a pas moyen d'enregistrer un nom sans une forme d'« approbation ». (Pas mal d'articles sur l'enregistrement de `.onion` avaient ainsi dit, à tort, que « l'IETF approuvait officiellement Tor ».)

Le problème 17 est un bon exemple du fait que la liste de Prévert <`https://fr.wiktionary.org/wiki/inventaire_%C3%A0_la_Pr%C3%A9vert`> qu'est ce RFC est en effet « non filtrée » et que toute remarque soulevée y a été mise, quels que soient ses mérites. Il consiste à dire que l'utilisation du registre des noms spéciaux est incohérente, car les enregistrements donnent des règles différentes selon le nom. Cela n'a rien d'incohérent, c'était prévu dès le départ par le RFC 6761 (section 5) : il n'y a pas de raison de traiter de la même façon `.onion` (qui n'utilise pas le DNS du tout) et `.bit` (passerelle entre le DNS et Namecoin).

Le problème 19 découle du fait que les noms de domaine ne sont pas de purs identificateurs techniques comme le sont, par exemple, les adresses MAC. Ils ont un sens pour les utilisateurs. Bien que, techniquement parlant, les développeurs de Tor auraient pu choisir le nom `.04aab3642f5` ou `onion.torproject.` comme suffixe pour les services en oignon, ils ont préféré un nom d'un seul composant, et compréhensible, `.onion`. Ce désir est bien compréhensible (une proposition à l'IETF est de reléguer tous les noms spéciaux sous un futur TLD `.alt`, qui ne connaitra probablement aucun succès même s'il est créé un

jour). Mais il entraine une pression accrue sur la racine des noms de domaine : si deux projets réservent `.zigzag`, lequel des deux usages faut-il enregistrer ?

Enfin, le dernier problème, le 21, est davantage technique : il concerne DNSSEC. Si un TLD est enregistré comme domaine spécial, faut-il l'ajouter dans la racine du DNS et, si oui, faut-il que cette délégation soit signée ou pas ? S'il n'y a pas de délégation, le TLD sera considéré comme invalide par les résolveurs validants. Par exemple, si je fais une requête pour `quelquechose.zigzag`, la racine du DNS va répondre :

```
% dig @k.root-servers.net A quelquechose.zigzag

; <<>> DiG 9.10.3-P4-Debian <<>> @k.root-servers.net A quelquechose.zigzag
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 7785
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;quelquechose.zigzag. IN A

;; AUTHORITY SECTION:
zero. 86400 IN NSEC zip. NS DS RRSIG NSEC
zero. 86400 IN RRSIG NSEC 8 1 86400 (
20171017050000 20171004040000 46809 .
wyKfrNEygGCbDscCu6uV/DFofs5DKYiV+jJd2s4xkkAT
...
. 86400 IN NSEC aaa. NS SOA RRSIG NSEC DNSKEY
. 86400 IN RRSIG NSEC 8 0 86400 (
20171017050000 20171004040000 46809 .
kgvHoclQNwmDKfgy4b96IgoOkdkyRWyXYwohW+mpfG+R
...
. 86400 IN SOA a.root-servers.net. nstld.verisign-grs.com. (
2017100400 ; serial
1800       ; refresh (30 minutes)
900        ; retry (15 minutes)
604800     ; expire (1 week)
86400      ; minimum (1 day)
)
. 86400 IN RRSIG SOA 8 0 86400 (
20171017050000 20171004040000 46809 .
GnTMS7cx+XB+EmbMFWt+yEAg29w17HJfUaOqvPsTn0eJ
...

;; Query time: 44 msec
;; SERVER: 2001:7fd::1#53(2001:7fd::1)
;; WHEN: Wed Oct 04 12:58:12 CEST 2017
;; MSG SIZE  rcvd: 1036
```

Et l'enregistrement NSEC prouvera qu'il n'y a rien entre `.zero` et `.zip`, amenant le résolveur validant à considérer que `.zigzag` ne peut pas exister. Si le nom devait être traité par le DNS (et, par exemple, résolu localement comme ceux du RFC 6303, c'est une réponse correcte : la requête n'aurait pas dû aller à la racine). Dans d'autres cas, cela peut être gênant. De toute façon, le débat est assez théorique : l'IETF n'a aucun pouvoir sur la racine du DNS, et aucun moyen d'y ajouter un nom.

Après cet examen des possibles et potentiels problèmes, la section 4 du RFC examine les pratiques existantes. Plusieurs documents plus ou moins officiels examinent déjà ces questions. Mais je vous préviens tout de suite : aucun ne répond complètement au·x problème·s.

Commençons par le RFC 2826 sur la nécessité d'une racine unique. C'est un document IAB et qui n'engage donc pas l'IETF, même s'il est souvent cité comme texte sacré. Sa thèse principale est qu'il faut une racine unique, et que `.zigzag`, `.pm` ou `.example` doivent donc avoir la même signification partout, sous peine de confusion importante chez l'utilisateur. Cela n'interdit pas des noms à usage local à condition que cela reste bien local. Comme les utilisateurs ne vont pas faire la différence, ces noms locaux vont forcément fuiter tôt ou tard. (Par exemple, un utilisateur qui regarde `http://something.corp/` ne va pas réaliser que ce nom ne marche qu'en utilisant les résolveurs de l'entreprise, et va essayer d'y accéder depuis chez lui. Un autre exemple serait celui d'un utilisateur qui essaierait de taper `ping sjnrk23rmcl4ie5atmz664v7o7k5nkk4jh7mm6lor2n4hxz2tos3eyid.onion` depuis la ligne de commande, sans se rendre compte que ping ne connait pas Tor.)

Bref, le RFC 2826 dit clairement qu'une racine unique est nécessaire, et que les noms locaux sont casse-gueule.

Le second RFC à lire est évidemment le RFC 6761, qui a créé le registre des noms de domaine spéciaux `<https://www.iana.org/assignments/special-use-domain-names/special-use-domain-names.xml>`. Lui, c'est un document IETF sur le Chemin des Normes. Quelques points qui sont souvent oubliés par les lecteurs du RFC 6761 méritent d'être répétés :

— Certains noms spéciaux ne le sont en fait pas tellement, et sont résolus par le DNS de la manière habituelle (c'est le cas de `in-addr.arpa`, par exemple, et rappelez-vous bien à son sujet que les noms de domaine spéciaux ne sont pas forcément des TLD).
— Parfois, les noms spéciaux sont résolus par le DNS mais de manière inhabituelle, comme `10.in-addr.arpa` (pour les adresses IP du RFC 1918), qui doit être traité par le résolveur, sans jamais interroger la racine (cf. RFC 6303).
— D'autres noms spéciaux sont gravement spéciaux et ne doivent pas utiliser le DNS du tout. Ils servent de « commutateurs » pour indiquer à la bibliothèque de résolution de noms (la GNU libc sur Ubuntu ou Mint par exemple) qu'il faut changer de protocole. C'est le cas de `.onion` (qui doit utiliser Tor) ou de `.local` (qui doit utiliser mDNS, RFC 6762).
— Et tous ces cas sont valides et normaux (même si certains traditionnalistes à l'IETF rechignent devant le dernier).

Notez qu'à l'heure actuelle, tous les noms enregistrés comme noms de domaine spéciaux sont des TLD **ou bien** sont résolus avec le DNS.

Troisième RFC à lire absolument avant de dire des bêtises sur les noms de domaine spéciaux, le RFC 2860, qui fixe le cadre des relations compliquées entre l'IETF et l'ICANN. En gros, la règle par défaut est que l'ajout (ou le retrait) de TLD dans la racine est une prérogative de l'ICANN **sauf** les « noms de domaine à usage technique » (notion non définie…) où l'IETF décide. Notez que ce document concerne uniquement les relations entre l'IETF et l'ICANN. Si le W3C, ou la développeuse du logiciel ZigZag, veut créer un TLD, que se passe-t-il ? Ce point n'est pas traité dans le RFC 2860. Certains exégètes estiment que cela veut dire que ces tiers sont implicitement exclus.

Il y a aussi d'autres documents mais moins cruciaux. Le RFC 6762 qui normalise mDNS est celui qui a réservé `.local` et c'est donc un exemple d'un enregistrement réussi (mais qui fut laborieux, plus de douze années de développement furent nécessaires, cf. l'annexe H du RFC 6762).

Autre exemple réussi, le RFC 7686 sur le `.onion`. `.onion` était utilisé depuis longtemps quand le RFC 6761 a créé le registre des noms de domaine spéciaux. L'enregistrement "*a posteriori*" a réussi,

malgré de vigoureuses oppositions mais il faut noter que le consensus approximatif de l'IETF a été facilité par une décision du CA/B Forum de ne plus allouer de certificats pour des TLD internes <https://www.digicert.com/internal-names.htm>.

Encore un autre RFC à lire, le RFC 6303, qui décrit les noms qui devraient idéalement être résolus localement, c'est-à-dire par le résolveur de l'utilisateur, sans demander aux serveurs faisant autorité. C'est par exemple le cas des `in-addr.arpa` correspondant aux adresses IPv4 privées du RFC 1918. Il ne sert à rien de demander à la racine l'enregistrement PTR de `3.2.1.10.in-addr.arpa` : ces adresses IP étant purement locales, il ne peut pas y avoir de réponse intelligente de la racine. Les noms en `10.in-addr.arpa` doivent donc être résolus localement, et sont donc, eux aussi, des « noms de domaine spéciaux ». Par contre, contrairement à `.local` ou à `.onion`, ils sont résolus par le DNS.

Pas fatigué·e·s ? Encore envie de lire ? Il y aussi l'étude d'Interisle <https://www.icann.org/en/system/files/files/name-collision-02aug13-en.pdf> sur les « collisions ». Derrière ce nom sensationnaliste conçu pour faire peur, il y a un vrai problème, le risque qu'un TLD récent masque, ou soit masqué par, un TLD alloué localement sans réfléchir (comme `.dev`). L'étude montrait par exemple que `.home` était celui posant le plus de risques.

Sur un sujet proche, il y a aussi une étude du SSAC <https://www.icann.org/en/system/files/files/sac-090-en.pdf>, un comité ICANN.

On a dit plus haut que les noms de domaine « spéciaux » n'étaient pas forcément des TLD. C'est par exemple le cas d'un nom utilisé pour certaines manipulations IPv6, `ipv4only.arpa`, créé par le RFC 7050, mais qui, par suite d'un cafouillage dans le processus, n'avait pas été ajouté immédiatement au registre des noms de domaine spéciaux. Dommage : ce nom, n'étant pas un TLD et n'ayant pas de valeur particulière, n'avait pas posé de problème et avait été accepté rapidement.

Enfin, un dernier échec qu'il peut être utile de regarder, est la tentative d'enregistrer comme noms de domaine spéciaux des TLD très souvent alloués localement, et qu'il serait prudent de ne pas déléguer dans la racine, comme le `.home` cité plus haut. Un projet avait été rédigé <https://datatracker.ietf.org/doc/draft-chapin-additional-reserved-tlds/> en ce sens, mais n'avait jamais abouti, enlisé dans les sables procéduraux.

Si vous n'avez pas mal à la tête à ce stade, vous pouvez encore lire la section 5, qui rappelle l'histoire tourmentée de ce concept de noms de domaine spéciaux. Lorsque le DNS a été produit (RFC 882 et RFC 883) pour remplacer l'ancien système `HOSTS.TXT` (RFC 608), la transition ne s'est pas faite sans douleur, car plusieurs systèmes de résolution coexistaient (le plus sérieux étant sans doute les "*Yellow Pages*" sur Unix, mais il y avait aussi "*NetBIOS name service/WINS*", qui ne tournait pas que sur Windows). Encore aujourd'hui, des anciens systèmes de résolution fonctionnent toujours. Le `HOSTS.TXT` survit sous la forme du `/etc/hosts` d'Unix (et de son équivalent Windows). Les systèmes d'exploitation ont en général un « commutateur » qui permet d'indiquer quel mécanisme de résolution utiliser pour quel nom. Voici un exemple d'un `/etc/nsswitch.conf` sur une machine Debian qui, pour résoudre un nom de domaine, va utiliser successivement `/etc/hosts`, LDAP, puis le DNS :

```
hosts:   files ldap dns
```

Le concept de TLD privé, connu uniquement en local, a été (bien à tort) recommandé par certaines entreprises comme Sun ou Microsoft. Il a survécu à la disparition des technologies qui l'utilisaient, comme "*Yellow Pages*". Aujourd'hui, c'est une source d'ennuis sans fin, et bien des administrateurs réseau ont

maudit leur prédécesseur pour avoir configuré ainsi tout le réseau local, allumant ainsi une bombe qui allait exploser quand le TLD « privé » s'est retrouvé délégué.

La discussion à l'IETF, notamment dans son groupe de travail DNSOP <https://datatracker.ietf.org/wg/dnsop> a été très chaude. Un premier document avait été élaboré, draft-adpkja-dnsop-special puis l'ancêtre de ce RFC avait été écrit, le premier document étant abandonné (il était très proche du point de vue de l'ICANN comme quoi seule l'ICANN devrait pouvoir créer des TLD, les autres acteurs n'étant que de vilains squatteurs).