

# Couche applicative (DNS, HTTP, SMTP, IMAP et POP)

Si on fait abstraction de toutes les couches en dessous de la couche application, on trouve le protocole applicatif. Le protocole applicatif définit comment les données de l'application peuvent être demandées et envoyées (par exemple via HTTP pour des sites internet, IMAP pour recevoir des emails ou encore SMTP pour envoyer des emails).

Le protocole applicatif est le langage utilisé par l'application pour communiquer, il décrit donc la forme des messages et le sens des échanges (définition syntaxique et sémantique).

Pour s'identifier, les applications utilisent un port et une IP (IPv4 ou IPv6), l'IP indique la machine et le port définit l'application émettrice ou destinataire (exemple, 80 pour HTTP, 443 pour HTTPS, 22 pour SSH, 53 pour DNS, etc).

## Le DNS

Chaque machine est identifiée au moyen d'une adresse IP (codée sur 32 bits IPv4 ou 128 bits avec l'IPv6).

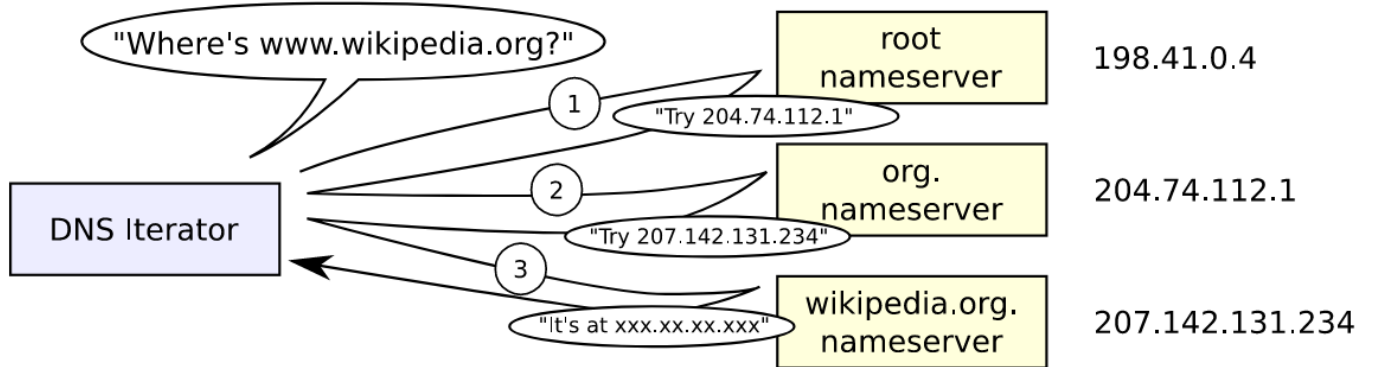
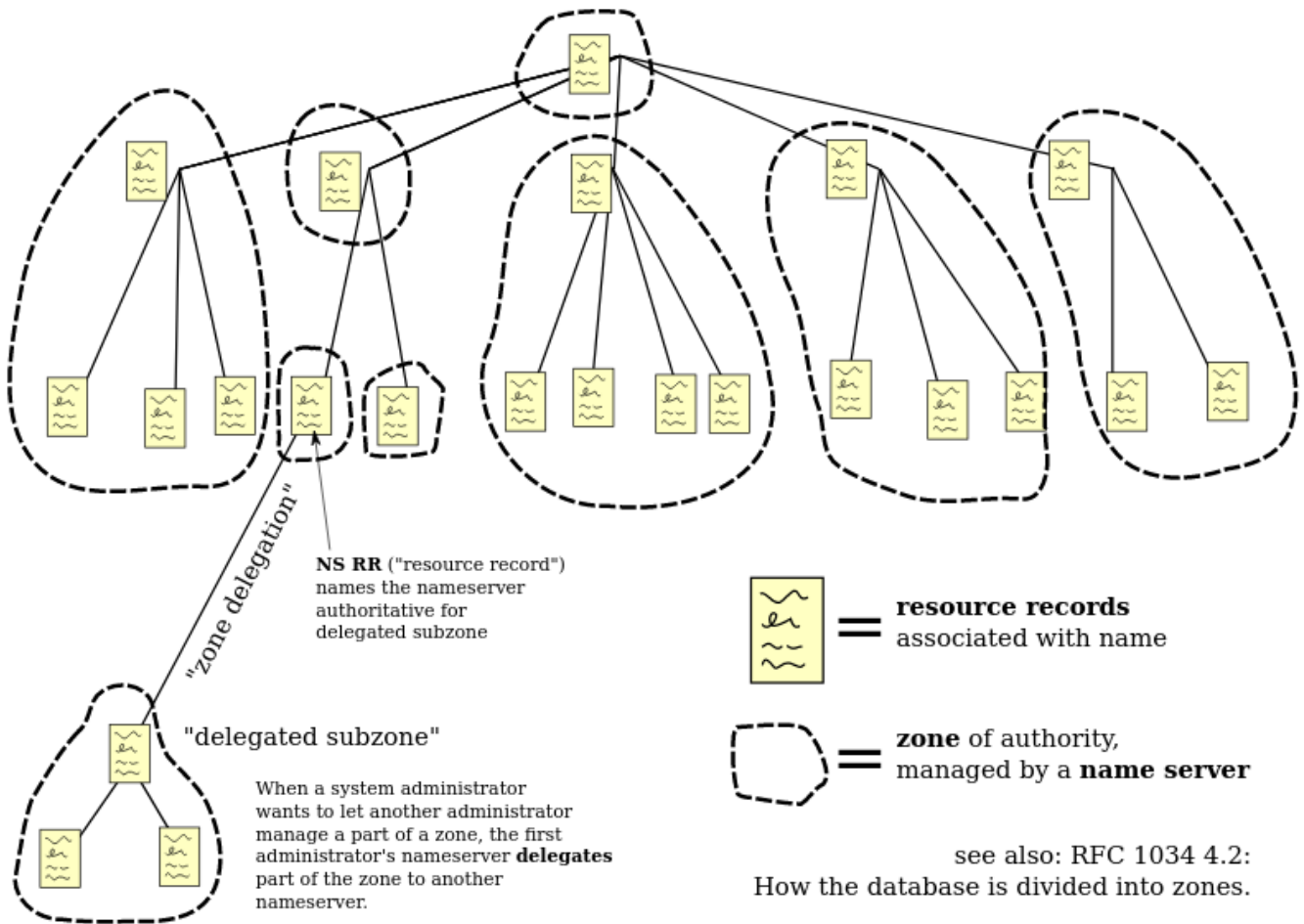
Le but du DNS (Domain Name System) est d'agir comme une sorte d'annuaire, ainsi à la place de devoir retenir des choses tel que `2001:41d0:404:200::597` il suffit de retenir `snowcode.ovh`. Il est donc possible de réserver un nom de domaine (généralement payant).

Une première manière de gérer cela serait d'avoir un fichier texte liant un nom et une adresse IP, par exemple avec `/etc/hosts` qui lie automatiquement `localhost` à l'adresse `127.0.0.1`.

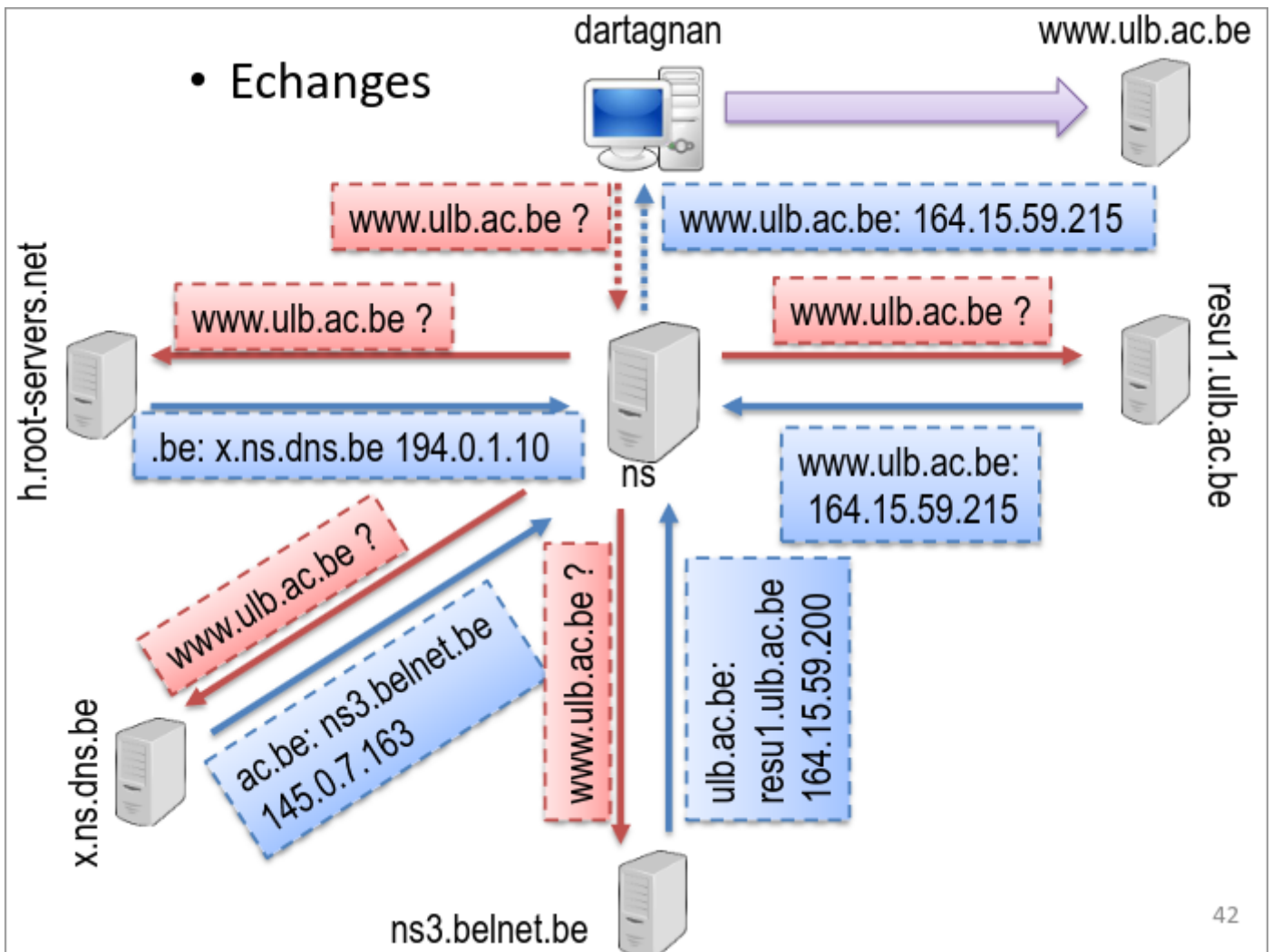
Le problème est qu'il serait impossible de synchroniser une base de données contenant tous les noms de domaines de tout le monde sur tous les appareils. On pourrait alors décider de créer une base de données centralisée, le problème est alors que tout le monde dépendrait d'un unique serveur pour les noms de domaines.

C'est pour cela qu'il y a plusieurs serveurs liés entre eux. Par exemple si on veut aller sur `swilabile.be`, on va d'abord demander au serveur mondial où se trouve le DNS de `be`, ce dernier peut alors ensuite renvoyer où se trouve `swilabile.be`. Il est donc possible d'avoir plus de niveaux d'imbrications.

# Domain Name Space



Il est donc possible de créer son propre DNS pour gérer ses propres sous domaines.



Le serveur qui va effectuer cette recherche est généralement celui qui est proposé par le réseau local (sauf s'il a été spécifiquement été précisé dans la configuration du système) via le protocole DHCP.

## Mail

Les mails utilisent plusieurs protocoles, le SMTP (via le port 25) permet d'envoyer des messages à des serveurs (tel que gmail.com, outlook.com, etc), la réception d'un message (ouvrir sa boîte mail) se fait via les protocoles POP3 (si on veut télécharger tous les mails localement) ou IMAP (si on veut ne pas avoir à télécharger tout localement).

Les mails doivent toujours respecter la RFC 1855 afin de pouvoir être bien reçue et comprise par son destinataire.

Pour transmettre une pièce jointe (binaire) en texte, on peut utiliser le base64 qui va encoder le binaire avec des caractères alphanumériques et quelques caractères spéciaux.

Le problème est que les protocoles de mail sont très vieux, ainsi, il est tout à fait possible de prétendre être quelqu'un d'autre, aussi les mails ne sont pas chiffrés et beaucoup de fournisseurs d'accès internet bloquent le port 25.

Il est cependant indispensable de pouvoir supporter les mails, car les emails sont devenus incontournables au fil du temps.

# Le web

Le web a été développé au CERN à Genève, derrière le "web" il y a plusieurs éléments (normes, protocoles, applications) :

- HTML (description de documents) ;
- Des serveurs web pour les délivrer (exemple apache ou nginx) ;
- Des clients pour les lire (Firefox, Chrome, etc) ;
- Le protocole HTTP qui permet à ces deux éléments de communiquer (versions allant de 1 à 3).
- Définition de l'URL (protocol://machine:port/chemin), par exemple (  
<https://books.snowcode.ovh:8888/hello.html>)

Chaque requête contient une commande (GET, HEAD, POST, PUT, DELETE), un entête (contenant des informations comme un token d'accès pour les cookies, d'où on vient, les métadonnées sur l'appareil, etc), et le corps qui est le contenu de la requête (page ou formulaire).

Ces entêtes peuvent être utilisés pour identifier et tracker des utilisateurs, car leur combinaison permet d'identifier des utilisateurs avec une certaine précision. Cela peut notamment être testé sur [amiunique.org](https://amiunique.org).

---

Revision #1

Created 27 February 2024 16:25:22 by SnowCode

Updated 27 February 2024 16:29:09 by SnowCode