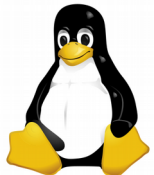


SERVEUR DHCP

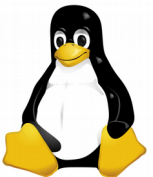


Jean-Louis Gouwy



Plan

- Qu'est-ce que DHCP ?
- Le protocole DHCP
- Fonctionnement
- Atelier
- Le serveur DHCP
- Le client DHCP
- Exercices
- L'agent relais DHCP
- Exercice
- Références



Qu'est-ce que DHCP ?

- **DHCP**

- Dynamic Host Configuration Protocol

- **OBJECTIFS**

- Protocole permettant d'attribuer dynamiquement une configuration IP aux machines clientes (au minimum: @IP / Netmask)
- Les @IP sont prises dans une plage spécifiée ou fixées en dur.
- Elles peuvent être attribuées durant un certain temps au delà duquel le client devra refaire une requête.

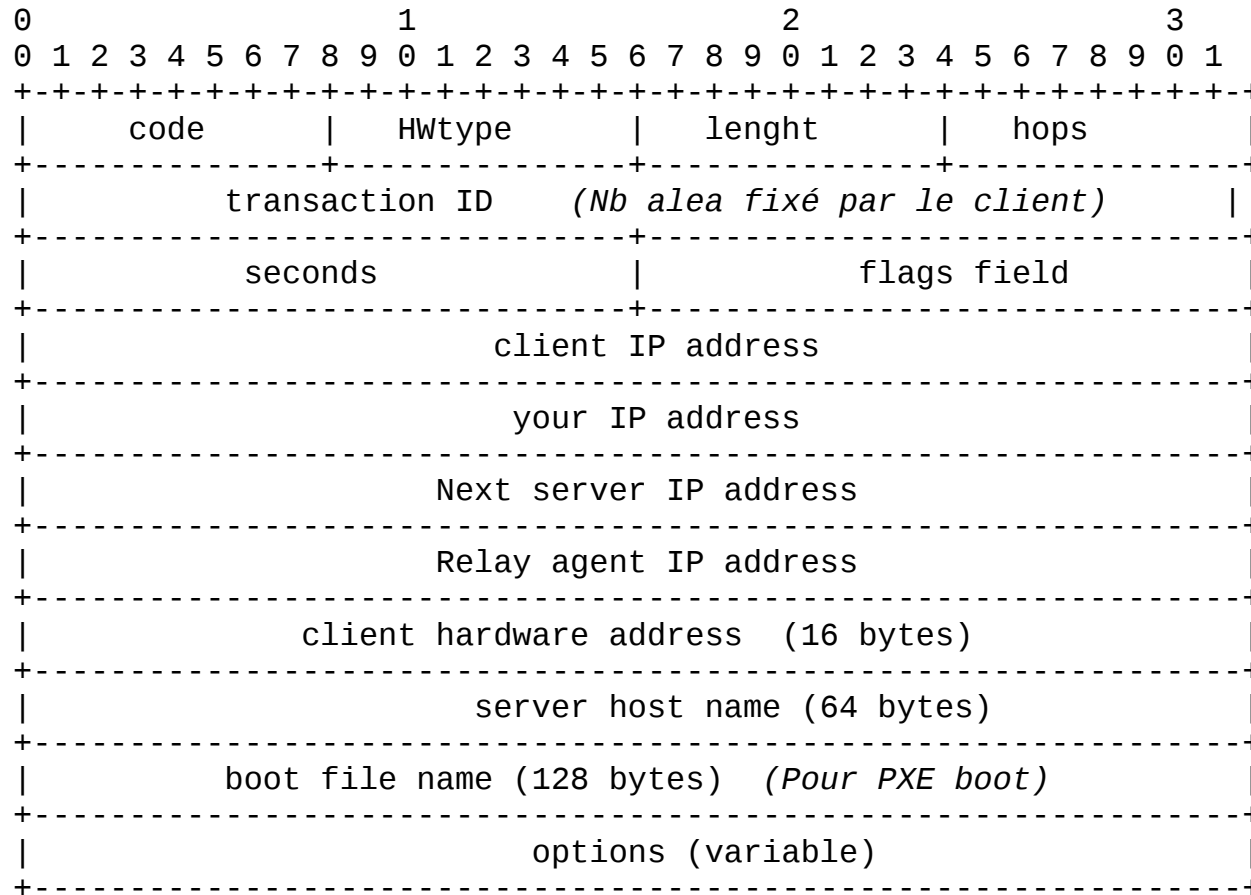
- **INTERETS**

- Centralisation de la configuration des interfaces des postes.
- Pour les réseaux à topologie très variables (portables ...)

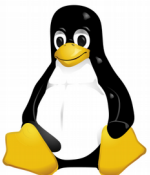


Le protocole DHCP

DHCP Message Format (RFC 2131)



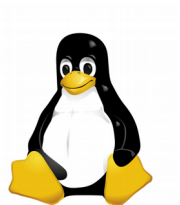
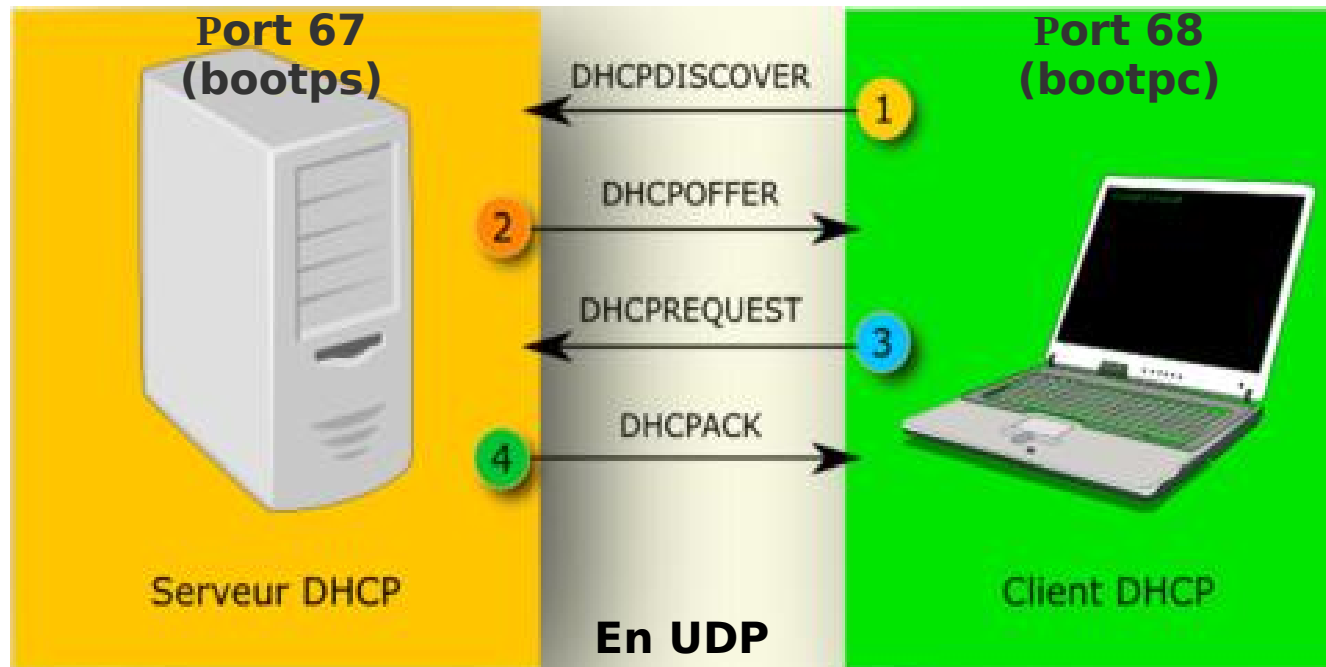
Plus d'info: <http://ylescop.free.fr/mrim/protocoles/rfc-fr/rfc2131.htm>



Fonctionnement

Lorsqu'un client DHCP n'a encore aucune connaissance du réseau et qu'un serveur DHCP est disponible, le mécanisme classique d'une attribution d'une configuration IP s'articule autour de l'échange de 4 trames:

DHCP DISCOVER -> DHCP OFFER -> DHCPREQUEST-> DHCPACK



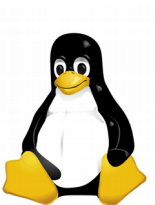
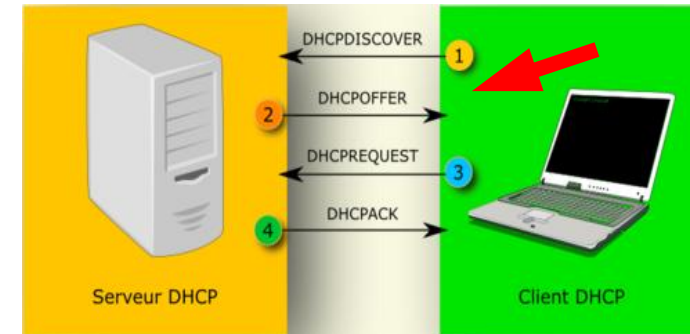
Fonctionnement (suite)

1. DHCP DISCOVER

Un message de *découverte d'un serveur* est envoyé en broadcast Ethernet sur le LAN et est destiné à trouver un serveur DHCP disponible.

Contenu:

- **client IP@:** 0.0.0.0 (le client n'est pas encore configuré)
- **@Mac client:** le serveur pourra retoucher par la suite le client via cette seule adresse.
- **N° transaction:** pour identifier la transaction (car plusieurs transactions sont possibles en même temps).
- ...



Fonctionnement (suite)

2. DHCP OFFER

Le(s) serveur(s) répondent en émettant un message *d'offre de bail* en broadcast Ethernet ou non selon la demande formulée dans la trame DISCOVER (*Bootpflags : unicast*)..

Contenu:

- **Proposition de configuration:**

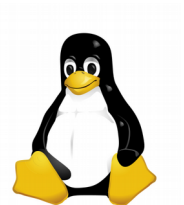
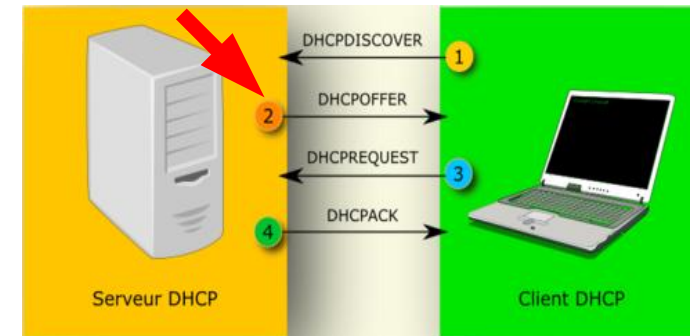
- . une @IP (your IP Address)
- . un bail
- . etc

- **@Mac client:** pour pouvoir toucher le client concerné.

- **Next srv IP@:** si la proposition est acceptée, le client mémorisera l'@IP du serveur.

- **N° transaction**

- ...



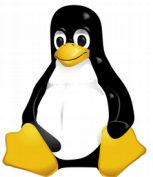
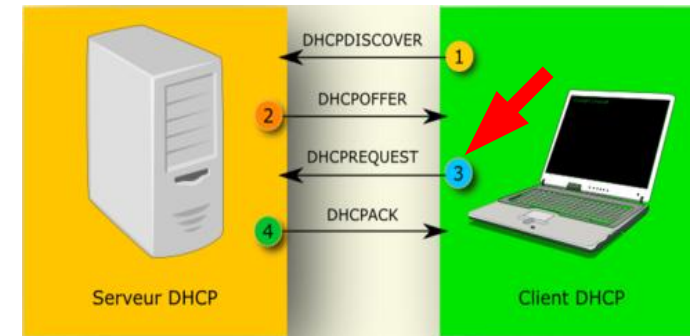
Fonctionnement (suite)

3. DHCP REQUEST

Le client envoie alors *son choix* à tous les serveurs et donc toujours en broadcast. Ceci pour indiquer l'offre qu'il accepte (généralement la 1ère reçue).

Contenu:

- **Serveur identifier (dans les options):** c'est l'@Ip du serveur retenu.
- **N° transaction**
- ...



Fonctionnement (suite)

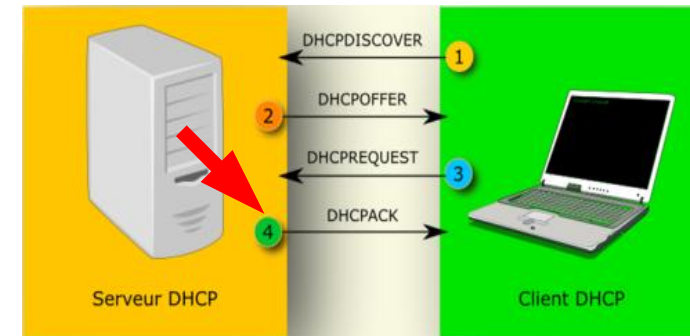
4. DHCP ACK

Le serveur concerné répond en unicast définitivement par un *accusé de réception* qui constitue une confirmation du bail.

L'adresse du client est alors marquée comme utilisée et ne sera plus proposée à un autre client pour toute la durée du bail.

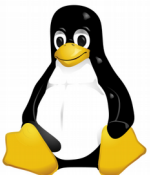
+

Mise à jour des BD d'attribution des bails (côté client et côté serveur)



Contenu:

- **Paramètres de configuration IP:** qui seront utilisés par le client pour configurer l'interface.
- **Bail et indicateurs de renouvellement**
- **Autres paramètres:** @IPPasserelle, @IPDns, @IPWins...
- **N° transaction**
- ...



AUTRES MESSAGES

DHCPNACK	Srv → Client pour signaler que l'adresse Ip demandée a été réassignée (bail expiré) ou qu'elle n'est plus actuellement valide (le client a été physiquement déplacé sur un autre réseau).
DHCPRELEASE	Client → Srv pour signaler qu'il libère sa configuration IP et annule son bail.
DHCPINFORM	Client → Srv pour recevoir le reste de sa configuration alors qu'il dispose déjà d'une @IP (configurée manuellement par exemple).
DHCPDECLINE	Client → Srv pour refuser l'offre proposée (l'@IP offerte est déjà utilisée par une autre machine).



Analysons maintenant, par un renifleur, une capture de trames correspondant à un dialogue initial DHCP.

Modèle de travail:

- Lançons un renifleur sur la machine serveur DHCP:

```
# tshark -i ifname port 67 > sniffdhcp.txt
```

- Libérons l'adresse sur la machine cliente:

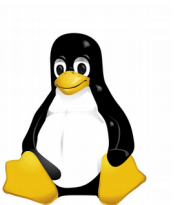
Prendre la commande qui a permis de lancer le client et remplacer les options "-l" et "-q" par l'option "-r" (release).

```
# /sbin/dhclient -r -lf /var/lib/dhclient/dhclient-ifname.leases  
-pf /var/run/dhclient-ifname.pid eth0
```

- Renouvelons l'adresse sur la machine cliente:

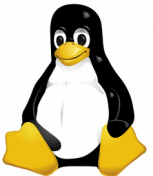
Prendre la commande qui a permis de libérer l'adresse et supprimer l'option "-r".

```
# /sbin/dhclient -lf /var/lib/dhclient/dhclient-ifname.leases  
-pf var/run/dhclient-ifname.pid eth0  
  
ou  
# systemctl restart network
```



Atelier

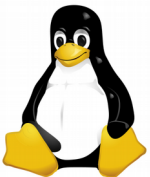
- Sur le serveur: `# mcedit sniffdhcp.txt`
- Analysons ensemble ...



Serveur DHCP

L'Internet Software Consortium (<http://www.isc.org>) développe un serveur DHCP pour le monde du logiciel libre. C'est le serveur DHCP le plus répandu, et celui qui "suit" au mieux les Rfcs.

- Daemon du serveur DHCP: `dhcpcd` *(man dhcpcd)*
- Fichier de configuration: `/etc/dhcp/dhcpcd.conf` *(man dhcpcd.conf)*
- Base des concessions d'@IP: `/var/lib/dhcpcd/dhcpcd.leases` *(man dhcpcd.leases)*



Serveur DHCP (suite)

LE FICHIER DE CONFIGURATION: **dhcpd.conf**

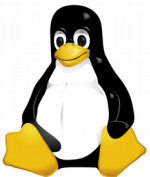
Il est composé de plusieurs sections, chacune limitée par des accolades.

Exemple simple

```
# Les options globales (applicables à toutes les sections et redéfinissables)
default-lease-time 259200;  Bail proposé (ici 3 jours)
max-lease-time 518400 ;     Bail maximum proposé si le client est gourmand

subnet 192.168.1.0 netmask 255.255.255.0  Réseau pour lequel dhcpd intervient
{
    range 192.168.1.10 192.168.1.245;      La réserve d'adresses dynamiques
    option subnet-mask 255.255.255.0;      Le masque
    option broadcast-address 192.168.1.255; Adresse de broadcast
    option routers 192.168.1.1;            La passerelle par défaut
    option domain-name-servers 192.168.1.6; Le serveur DNS

    host monserveur  Pour le host dont l'@Mac est renseignée, lui attribuer
    {                une adresse fixe.
        hardware ethernet 00:80:c8:85:b5:d2;
        fixed-address 192.168.1.1;  L'adresse fixe ne doit pas appartenir au range !!
    }
}
```



Serveur DHCP (suite)

BASE DES CONCESSIONS: **dhcpd.leases**

- Doit exister vide (le créer si nécessaire via *touch*) après l'installation.
- C'est la base de données d'attribution des clients DHCP (destinataire, date de début et de fin et l'@Mac de la carte).

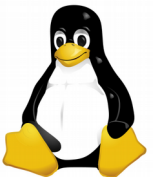
Exemple

```
...  
lease 192.168.1.243 {  
    starts 1 2010/08/30 15:51:18;  
    ends 4 2010/09/02 15:51:18;  
    tstp 4 2010/09/02 15:51:18;  
    binding state active;  
    next binding state free;  
    hardware ethernet 00:22:15:56:b7:32;  
    uid "\001\000"\025\226\24TJ";  
}  
...
```

Le client d'@Mac 00:22:15:56:b7:32 a reçu l'adresse 192.168.1.243 et ce pour 3 jours.

Remarques:

- Il s'agit de l'heure universelle GMT.
- 0: dimanche, 1: lundi .. 6: samedi.



Serveur DHCP (suite)

LES INTERFACES D'ECOUTE

Par défaut, DHCP écoute sur toutes les interfaces.

Pour en sélectionner certaines d'entre elles, il faut modifier la façon dont le service dhcpd est lancé par systemd:

→ Modification du service dhcpd.service

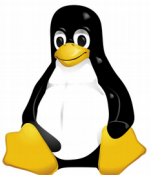
```
# cp /lib/systemd/system/dhcpd.service /etc/systemd/system/
```

```
/etc/systemd/system/dhcpd.service
```

```
→ ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf  
-user dhcp -group dhcp --no-pid enp0s3
```

Il n'écoute que sur l'interface
enp0s3

```
# systemctl --system daemon-reload → pour relire la configuration des services  
# systemctl restart dhcpd
```



Serveur DHCP (suite)

LANCEMENT/ARRET/REDEMARRAGE

```
# systemctl start/stop/restart dhcpd
```

ACTIVATION AU DEMARRAGE

```
# systemctl enable dhcpd
```

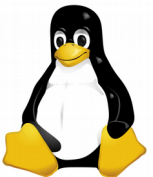
REMARQUES

- Bien souvent un seul serveur DHCP par LAN.
- Le client dhcp peut changer la configuration du fichier `/etc/resolv.conf`.



Client DHCP

- Son rôle est de rechercher sur le réseau un serveur DHCP et de négocier avec lui une configuration IP cohérente.
- Le plus en vogue est *dhclient* (*man dhclient*) mais il en existe bien d'autres (dhcpcd, pump, dhcpd ...).
- Chaque client possède son propre fichier de configuration: dhclient.conf, pump.conf , ... (hors cadre du cours).



Client DHCP (suite)

BAILS OBTENUS: `/var/lib/dhclient/dhclient--ifname.leases`

Exemple

```
lease {  
    interface "enp0s3";  
    fixed-address 192.168.1.243;  
    option subnet-mask 255.255.255.0;  
    option routers 192.168.1.1;  
    option dhcp-lease-time 259200;  
    option domain-name-servers 192.168.1.6;
```

```
    option dhcp-server-identifier 192.168.1.2;
```

Serveur DHCP d'origine

```
renew 3 2018/09/01 3:51:47;
```

1er essai de renouvellement d'adresse (~ ½ du bail)

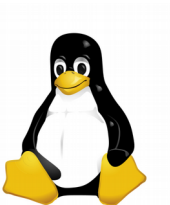
```
rebind 3 2018/09/01 21:05:37;
```

2eme essai de renouvellement d'adresse (~¾ du bail)

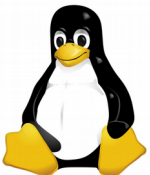
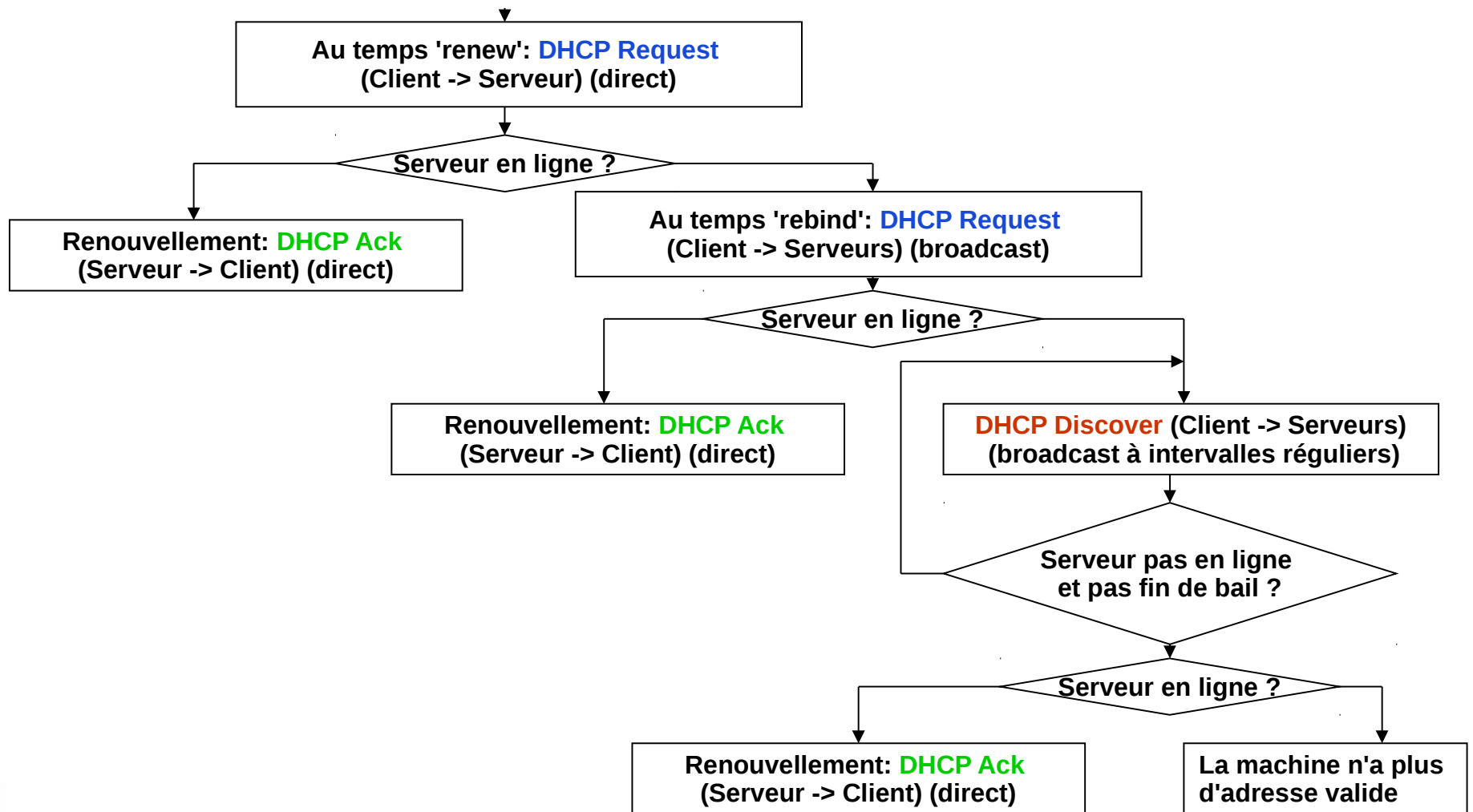
```
expire 4 2018/09/02 6:51:18;
```

Expiration du bail (~7/8 du bail)

```
}
```



MECANISME DE RENOUVELLEMENT DU BAIL (Suite)

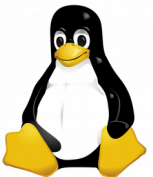


REMARQUES

- Grâce aux informations conservées dans ce fichier `dhclient.leases`, le client *dhclient* adopte un comportement un peu particulier, que l'on ne retrouve pas dans celui de Microsoft, par exemple.

Lorsqu'un hôte a obtenu un premier bail de la part du DHCP, l'adresse du serveur DHCP est conservée et, même après extinction et redémarrage de l'hôte au bout d'un temps bien supérieur à la durée de son bail, le client commencera par envoyer directement un DHCP request au serveur qu'il connaît.

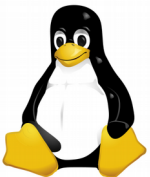
- Sous RedHat, une configuration dynamique d'adresses IP entraîne automatiquement le lancement du client *dhclient*. Ce qui n'est pas le cas pour une configuration statique.



Client DHCP (suite)

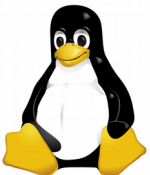
REFLEXION

Peut-on avoir un client et un serveur DHCP qui tournent sur la même machine ? Oui ou non et pourquoi ?



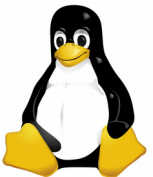
Exercice 1

- **Constituez un petit réseau indépendant de 3 machines**
 - une des machines jouera le rôle de serveur DHCP
 - les 2 autres seront des clientes
- **Installation**
 - installez le serveur DHCP sur la machine serveur
 - installez, si nécessaire, le client DHCP sur les 2 autres machines
- **Configuration de la machine serveur**
 - lui attribuer une adresse IP statique (192.168.0.1/24)
 - configurez le serveur DHCP
 - . plage d'adresses 192.168.0.10 à 192.168.0.20
 - . une adresse fixe (192.168.0.21) pour une des deux machines
 - . un bail de 4 heures pour la première machine et un de 8 heures pour la seconde
 - . attribution automatique d'un hostname à la machine d'adresse fixe (voir 'option host-name' ou 'use-host-decl-names')



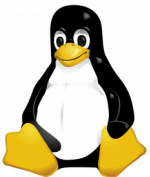
Exercice 1 (suite)

- **Configuration de la 1ère machine cliente**
 - attribution dynamique de ses paramètres IP
 - vérifiez que le serveur DHCP ne tourne pas
 - lancez, si nécessaire, le client DHCP (s'il ne tourne pas déjà)
 - redémarrez le réseau
 - vérifiez si l'interface est correctement configurée
- **Configuration de la 2ème machine cliente**
 - Idem 1ère machine



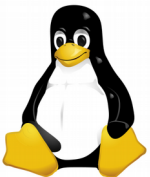
Exercice 1 (suite)

- Comment connaître l'@Mac d'une interface distante à l'aide d'un serveur DHCP ?
- Examinez et vérifiez le fichier `dhcpd.leases` et `dhclient.ifname.leases` après une connexion d'un client.
- Examinez et vérifiez le fichier `dhcpd.leases` après une déconnexion d'un client.
- Examinez et vérifiez le fichier `dhcpd.leases` après la libération de l'adresse par le client.
- Comment tester qu'un client redemande une adresse en fin de bail ?
Faites l'expérience au laboratoire.



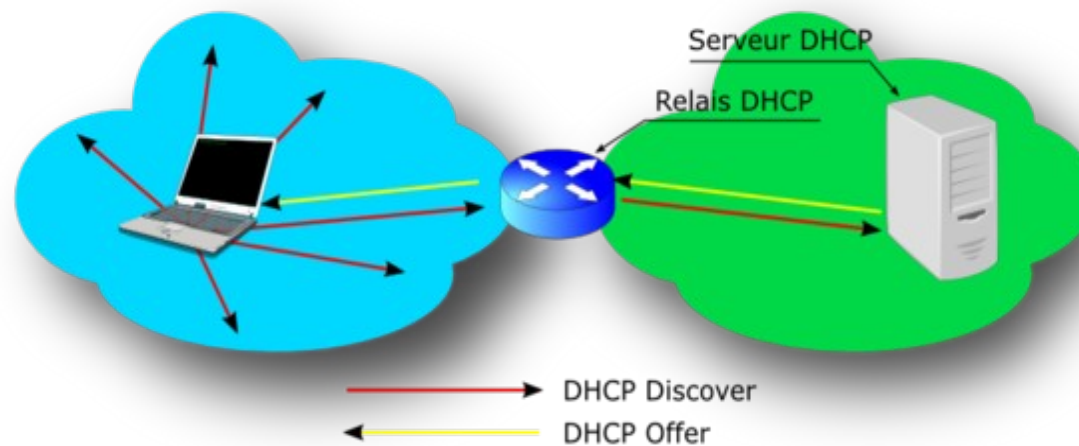
Exercice 2

- **Constituez deux petits réseaux interconnectés par un routeur**
 - le serveur DHCP est installé sur le routeur
 - 1^{er} réseau : 192.168.30.0/24
 - 2^{eme} réseau : 192.168.40.0/24
- **Configuration du DHCP**
 - Il attribuera, par range (10 à 20) des adresses pour 4 heures à chaque machine de chaque réseau

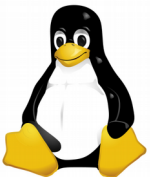


L'agent relais DHCP

- Comme les clients contactent les serveurs DHCP à l'aide d'une diffusion, dans un inter-réseau, vous devriez théoriquement installer un serveur DHCP par sous-réseau.
- C'est ici qu'intervient l'agent relais DHCP dont le rôle est d'intercepter les requêtes en broadcast et les transmettre à un serveur DHCP connu de cet agent.
- Si votre routeur prend en charge la RFC 1542, il peut faire office d'agent de relais DHCP, et ainsi relayer les diffusions de demande d'adresse IP des clients DHCP dans chaque sous-réseau.

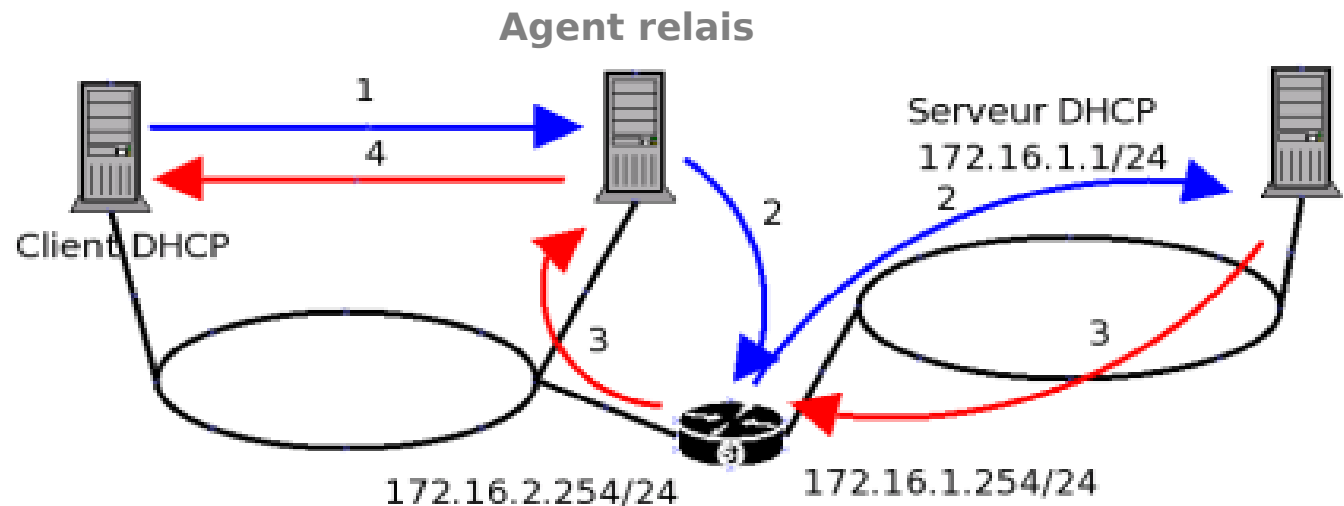


Dhcp (JL Gouwy)

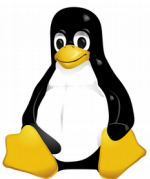


L'agent relais DHCP (suite)

- Dans le cas contraire, une machine peut être configurée comme agent de relais DHCP.

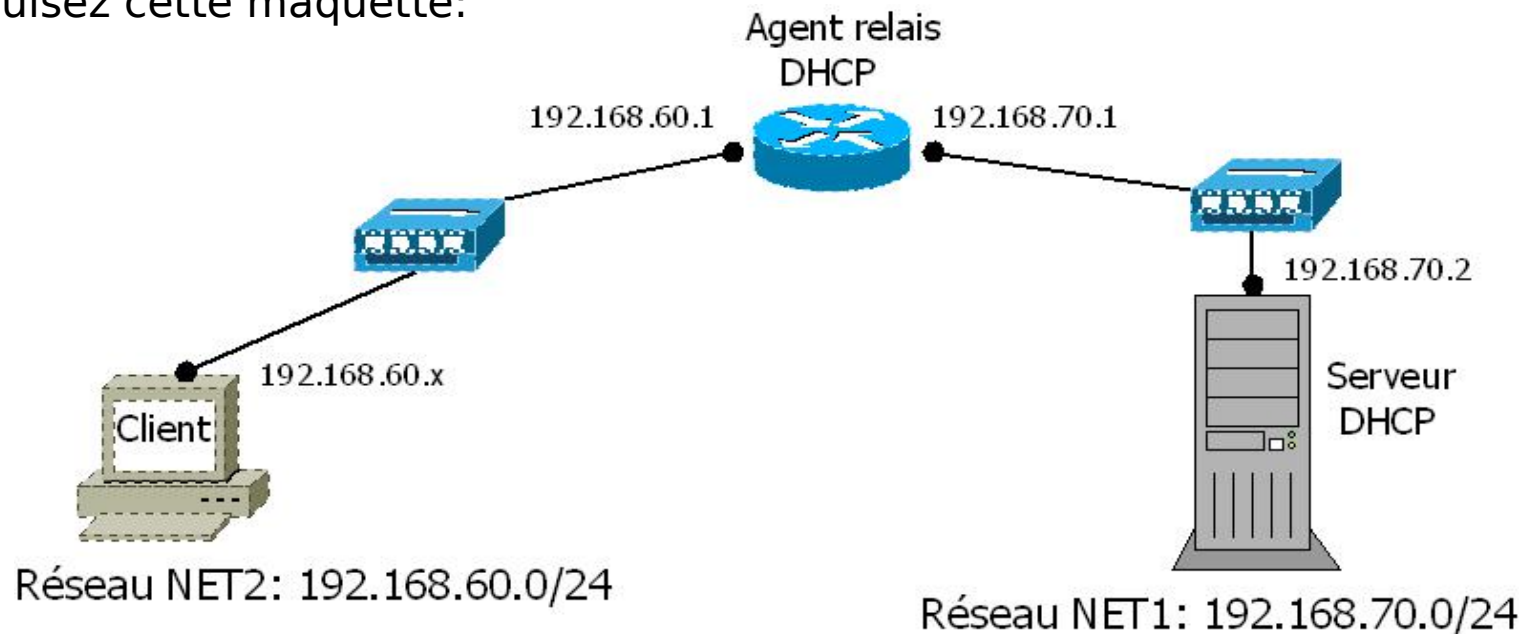


- (1) Après avoir envoyé une trame de broadcast, le client DHCP dialogue avec l'agent de relais en unicast.
- (2) L'agent demande une configuration IP au serveur DHCP dont il connaît l'adresse.
- (3) Le serveur retourne une configuration à l'agent.
- (4) Celle-ci est donnée au client DHCP par l'agent.



Exercice 3

Construisez cette maquette:



- L'installation du serveur dhcp inclut celle de l'agent relais (dhcrelay).
- Le serveur dhcp ne doit pas tourner sur l'agent relais.
- Par défaut l'agent relais écoute sur toutes ses interfaces.
- Démarrage de l'agent: `# systemctl start dhcrelay`
- Activation manuelle de l'agent: `commande dhcrelay (man dhcrelay)`

☑ *En vue de réaliser l'exercice sur 3 machines, nous installons ici l'agent directement sur la passerelle.*



Références

WEBOGRAPHIE

<http://www.frameip.com/dhcp>
<http://christian.caleca.free.fr/dhcp.html>
<http://www.linux-france.org/prj/edu/archinet/systeme/ch29.html>

BIBLIOGRAPHIE

Red Hat Enterprise Linux 7
Guide de Gestion des réseaux
Last Updated: 2018-04-17

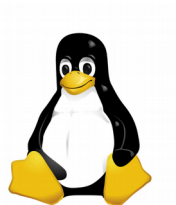
TCP/IP
Par Joe Casad & Bob Willsey (Edition CampusPress)

CentOS Bible
By Timothy Boronczyk and Christopher Negus - Edition 2009 - Wiley Publishing, Inc.

Le Campus - Linux: Installation, configuration et applications
Par Michael Kofler – 8ème Edition 2009 – Pearson Education France

The DHCP HandBooks (2ème édition) - Par Ralph Droms & Tedemon

ADMINISTRATION UNIX: ASPECTS RESEAUX Par Xavier Bogaert
Technofutur3



RESOLUTION LOCALE DE NOMS

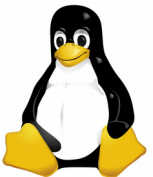


Jean-Louis Gouwy



Plan

- La résolution de noms
- La résolution locale
- Références

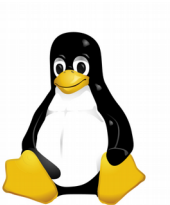


La résolution de noms

- Pour établir une correspondance compréhensible par les systèmes entre des noms de machines et leur adresse.

Ex: DNS, WINS, NIS ...

- La correspondance peut être connue localement (statiquement) ou être disponible sur le réseau.
- Résolution locale (statique) => /etc/hosts
 - ☹ Un fichier /etc/hosts à gérer pour chacune des machines du réseau
- Solution plus générale (locale + globale - internet) => système DNS (voir plus loin)
 - ☺ Un serveur DNS à configurer sur une seule machine



La résolution locale

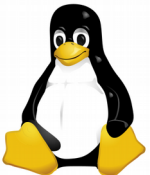
1^{ère} étape: Choisir le service de résolution de noms

/etc/nsswitch.conf ⁽¹⁾ (extrait / prioritaire si les 2 existent)

```
...  
hosts: files nis dns  
...
```

—————→

(1) Prise en compte immédiate des modifications apportées ...



La résolution locale

2^{ème} étape: Configurer le(s) services(s) choisi(s)

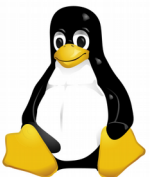
files: **/etc/hosts**⁽¹⁾

```
127.0.0.1      theti localhost.localdomain  localhostdomain
198.197.56.141 theti theti.isat.be    papyrus
198.197.56.9   mapasserelle
198.197.56.67  fileserver
```

dns: **/etc/resolv.conf** ⁽¹⁾

```
nameserver 193.190.156.67
nameserver 193.190.159.19
```

⁽¹⁾ Prise en compte immédiate des modifications apportées ...



La résolution locale

- Commandes utiles

hostnamectl pour montrer ou changer le nom de la machine

uname -n pour montrer le nom de la machine

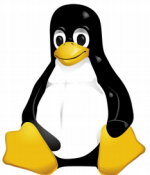
- Manuel *man /etc/nsswitch.conf*

- Fichiers de configuration

/etc/nsswitch.conf

/etc/hosts

/etc/resolv.conf



BIBLIOGRAPHIE

ADMINISTRATION UNIX: ASPECTS RESEAUX
Par Xavier Bogaert
Technofutur3

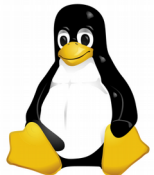
LINUX Red Hat Fedora
Par Bill Ball & Hoyt Duff
CampusPress (Paris)



LE SYSTEME DNS

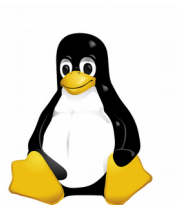


Jean-Louis Gouwy



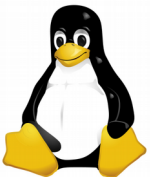
Plan

- But et historique
- Plan du DNS
 - Systeme de nommage / Domaine / Zone / Autorité / Délégation / Serveurs racines / Redondance
- Résolution et résolution inverse
- Parcours d'une requête
- Debug DNS
- Serveur de cache
 - Architecture / Utilités / Configuration minimale
- Bind
 - Présentation / Packages / Composants / Serveur cache
- Exercices: Serveur de cache
 - Exercice 1: Serveur de cache / Exercice 2: Serveur forward esclave
- Serveur autoritaire
 - Architecture / Remarques
- Bind
 - Gérer un domaine / Serveur autoritaire récursif / Serveur autoritaire itératif / Les fichiers de zone



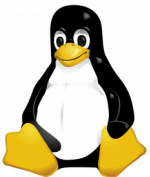
Plan (suite)

- Exercice: Serveur autoritaire
Exercice 3: Serveur autoritaire de cache
- Délégation et sous domaine
- Exercice: Délégation et sous domaine
Exercice 4
- Redondance
- Bind
Configuration du serveur primaire / Configuration du serveur secondaire /
Synchronisation
- Exercice: Redondance
Exercice 5
- Références



But et historique

- Pour traduire des noms DNS (FQDN - Fully Qualified Domain Name) en adresse IP et l'inverse
 - www.helha.be → 193.190.66.18
 - 193.190.66.18 → www.helha.be
- Service essentiel pour tout réseau relié au monde extérieur
- Très utile aussi pour un réseau local (remplacement de /etc/hosts)
- Car le fichier /etc/hosts est très limité
 - doit être recopié sur toutes les machines
 - mises à jour, ajouts et suppressions synchronisées fastidieuses
- En 1984, Paul Mockapetris mit au point un système de nommage: le DNS (décrit dans les RFC 883 et 884, puis 1034 et 1035).



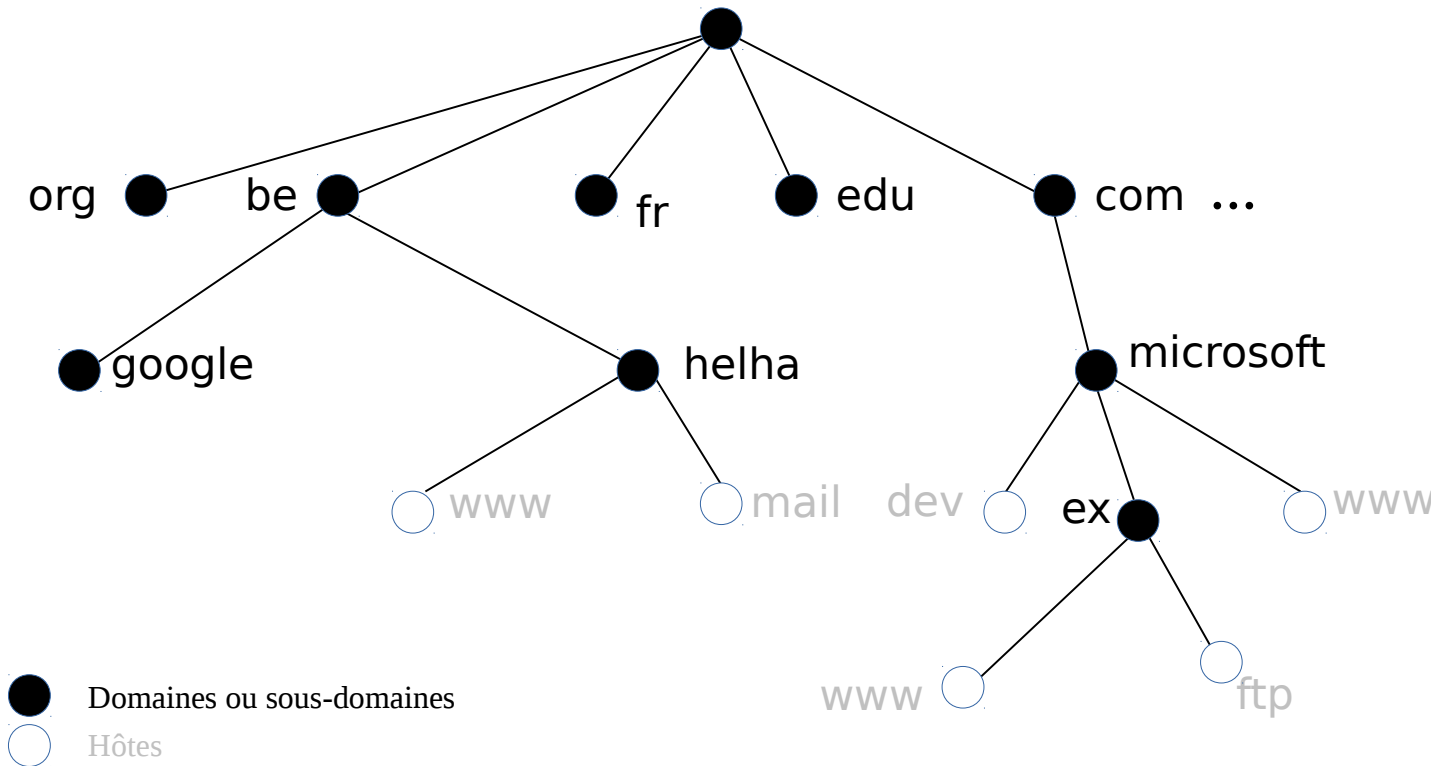
Plan du DNS: Système de nommage

Racine

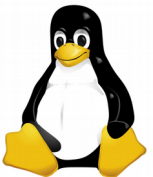
Top Level Domains

2nd Level Domains

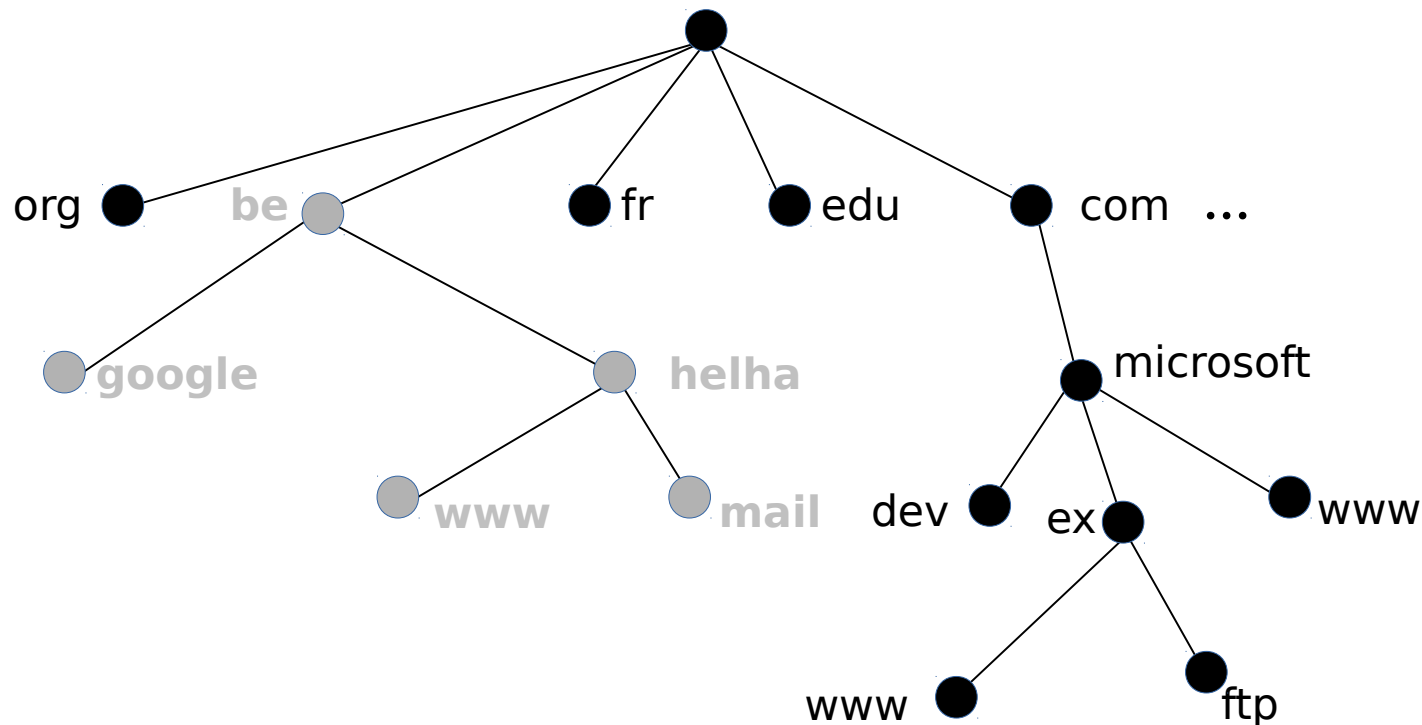
Host names and subdomains



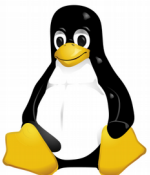
- Basé sur un modèle en arborescence similaire à celui des systèmes de fichiers.
- La dénomination d'un nom commence par le bas pour se terminer à la racine.
(ex. **ftp.ex.microsoft.com**. → le dernier point étant optionnel)



Plan du DNS: Domaine

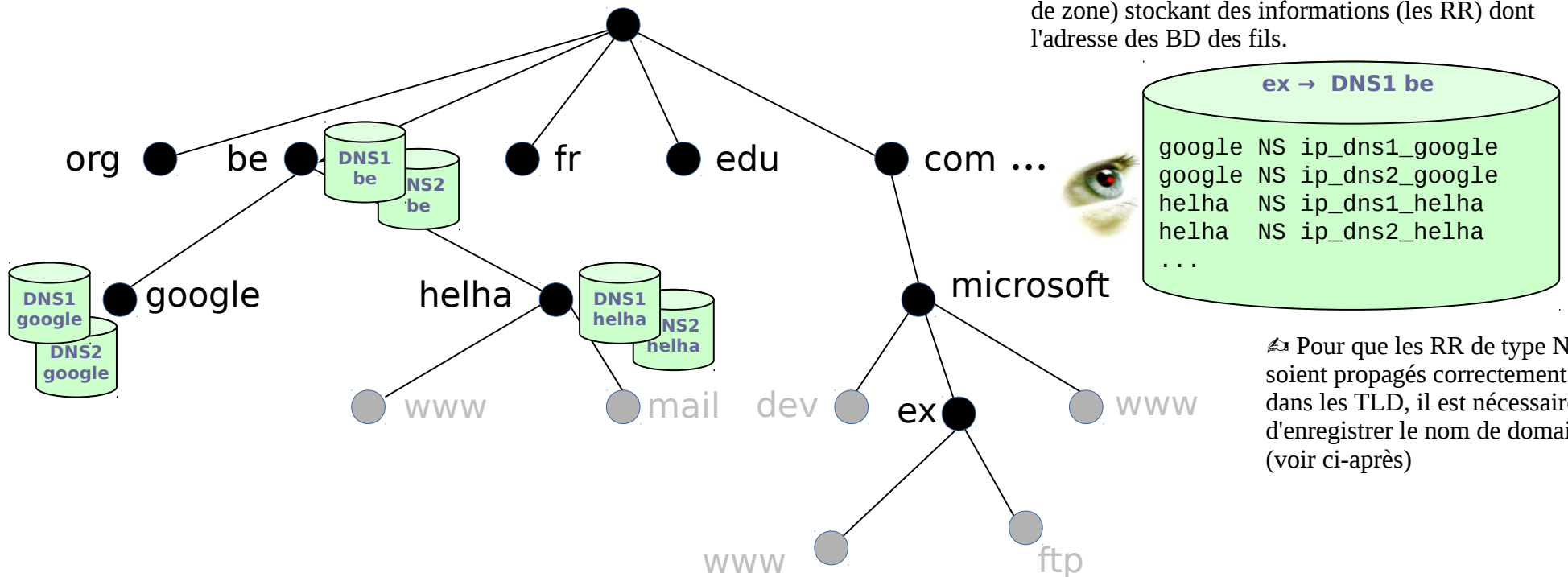


- Un domaine est l'ensemble d'une sous arborescence.
Exemple: Le domaine "**be.**" rassemble toute la sous-arborescence à partir du noeud be.
- Il y a un domaine racine "." et des domaines fils: ex. "org."
- Un nom de domaine est utilisé dans les URL et les adresses de messagerie (ex. Soit le domaine "helha.be"
→ URL : <http://www.helha.be>
→ @ : toto@helha.be



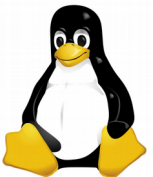
Plan du DNS: Zone

Chaque nœud contient une (ou plusieurs) BD (fichiers de zone) stockant des informations (les RR) dont l'adresse des BD des fils.

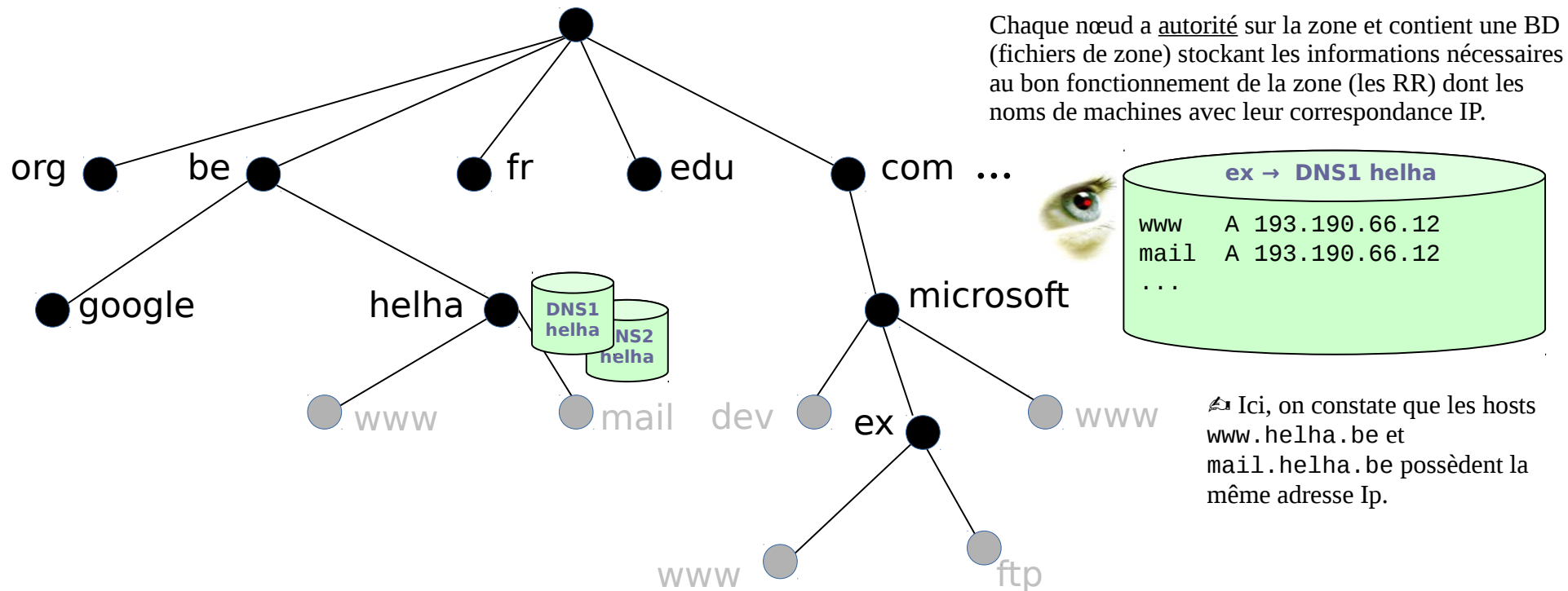


✎ Pour que les RR de type NS soient propagés correctement dans les TLD, il est nécessaire d'enregistrer le nom de domaine (voir ci-après)

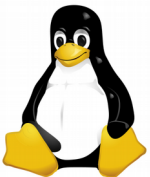
- Une zone est la partie descriptive pour un niveau donné.
 - Elle est restreinte à un nœud → une zone est constituée de la base de données décrivant un nœud.
- ✎ Un RR (Ressource Record) de type NS renseigne l'adresse Ip d'un serveur Dns d'un sous-domaine.



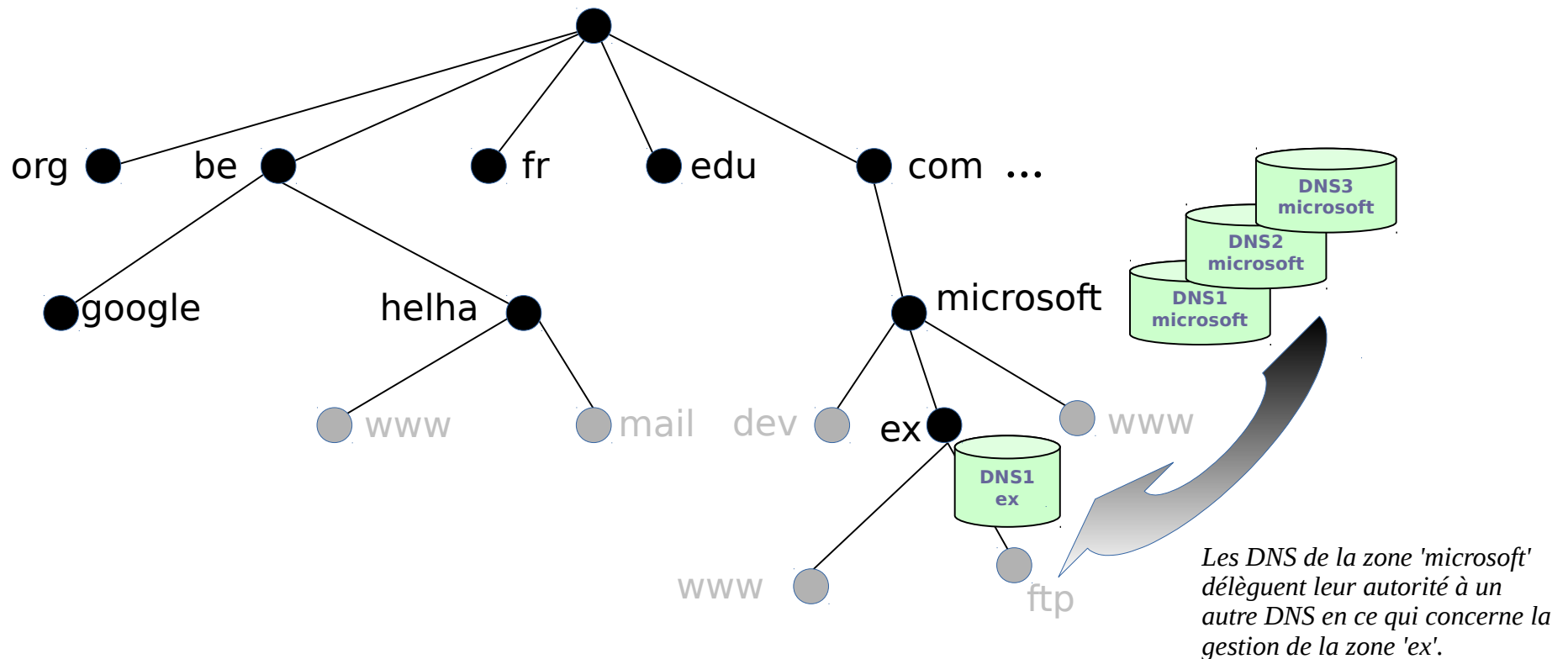
Plan du DNS: Autorité



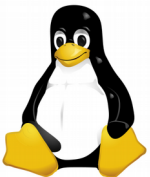
- 👁 Un DNS particulier s'occupe d'un nœud sur lequel il a autorité. On dit que le serveur gère une zone d'autorité. C'est à dire qu'il gérera l'attribution des noms et résoudra les noms via un fichier de zone (BD) distinct pour chaque nœud.
- 👁 Un RR de type A établit une correspondance entre un nom de host et son adresse Ip.



Plan du DNS: Délégation

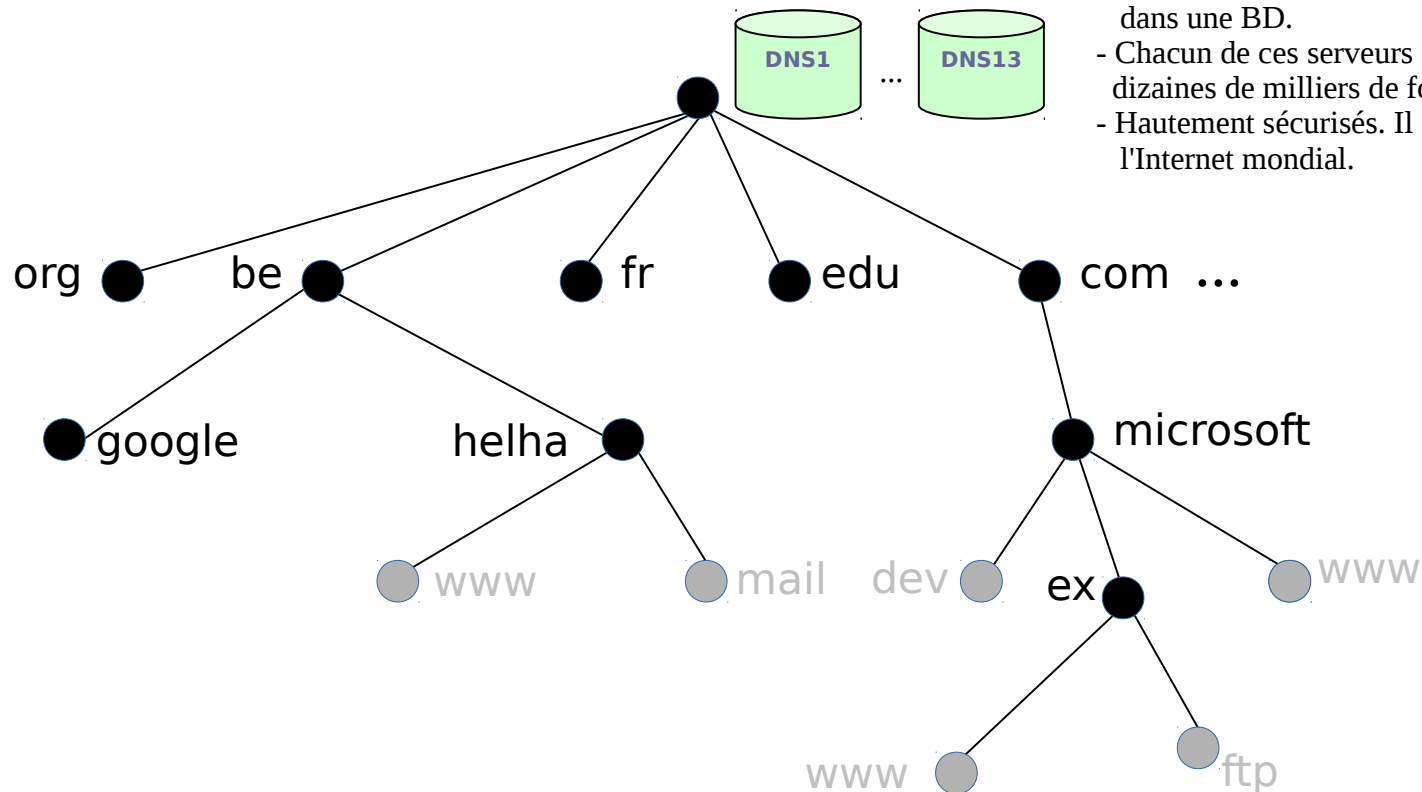


- Un serveur faisant autorité sur une zone peut déléguer la gestion de ses sous-domaines à d'autres serveurs de nom.
- Des fichiers de zone pour chaque sous-domaine doivent donc être créés et les fichiers de zone du domaine parent devront être modifiés en conséquence.



Plan du DNS: Serveurs racines

- 13 serveurs répartis dans le monde et gérés par 12 organisations indépendantes, chacun contenant les références de tous les serveurs de premier niveau dans une BD.
- Chacun de ces serveurs est consulté plusieurs dizaines de milliers de fois par heure.
- Hautement sécurisés. Il en va de l'utilisation de l'Internet mondial.

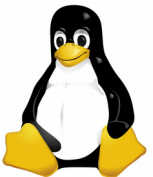


👉 Tout serveur DNS doit connaître les adresses IP des serveurs racines.



Plan du DNS: Serveurs racines (suite)

- Ils contiennent la même information grâce à un système de réplication.
- Ils sont identifiés par les lettres de A à M et appartiennent tous au même domaine ROOT-SERVERS.NET.



Plan du DNS: Serveurs racines (suite)

- Plusieurs de ces serveurs correspondent à plusieurs serveurs répartis dans le monde: <http://root-servers.org/>

The screenshot shows the 'Root Servers' website. At the top, there is a navigation bar with letters A through M. The letter 'K' is highlighted with a blue circle. Below this, the 'Operator' is listed as 'RIPE NCC', which is also circled in blue. The 'Locations' section shows 'Sites: 56', also circled in blue. Below this, a list of 56 locations is displayed, each with a globe icon and the location name. The locations are: Amsterdam, NL; Ashland, US; Athens, GR; Barcelona, ES; Beirut, LB; Belgrade, RS; Berlin, DE; Brisbane, AU; Bucharest, RO; Budapest, HU; Buenos Aires, AR; Calgary, CA; Cork, IE; Doha, QA; Frankfurt, DE; Gdynia, PL; Geneva, CH; Hamburg, DE; Helsinki, FI; Johannesburg, ZA; Kansas City, US; Karlsruhe, DE; Kuwait City, KW; London, UK; Miami, US; Milan, IT; Montevideo, UY; Moscow, RU; Mumbai, IN; Noida, IN; Novosibirsk, RU; Ottawa, CA; Paris, FR; Poznan, PL; Prague, CZ; Reno, US; Reykjavik, IS; Richmond, US; Riga, LV; Saint Petersburg, RU; San Jose, CR; Santiago, CL; Semey, KZ; Sofia, BG; St George, US; Tbilisi, GE; Tehran, IR; Tokyo, JP; Vienna, AT; Yerevan, AM; Zurich, CH. At the bottom, the IP addresses are listed: IPv4: 193.0.14.129 and IPv6: 2001:7fd::1, both circled in blue.

- Ici, le serveur K.ROOT-SERVERS.NET, accessible par une adresse Ipv4 et Ipv6 et dupliqué 56 fois dans le monde, est géré par l'organisation RIPE NCC.
- Le serveur A.ROOT-SERVERS.NET est le serveur d'origine. Les autres (B → M) sont des serveurs miroirs de celui-ci.



Plan du DNS: Serveurs racines (suite)

- Un serveur racine est en fait constitué d'un ensemble de serveurs dupliqués et configurés en 'anycast'. Autrement dit, chacun de ceux-ci possède la même adresse Ip mais un seul répondra à la requête DNS.

Lorsqu'un routeur reçoit une demande pour joindre une adresse 'anycast', il la route généralement vers le serveur géographiquement le plus proche.

→ nécessité de pouvoir configurer des routeurs supportant un protocole de routage dynamique - ex. BGP (Hors cadre du cours)

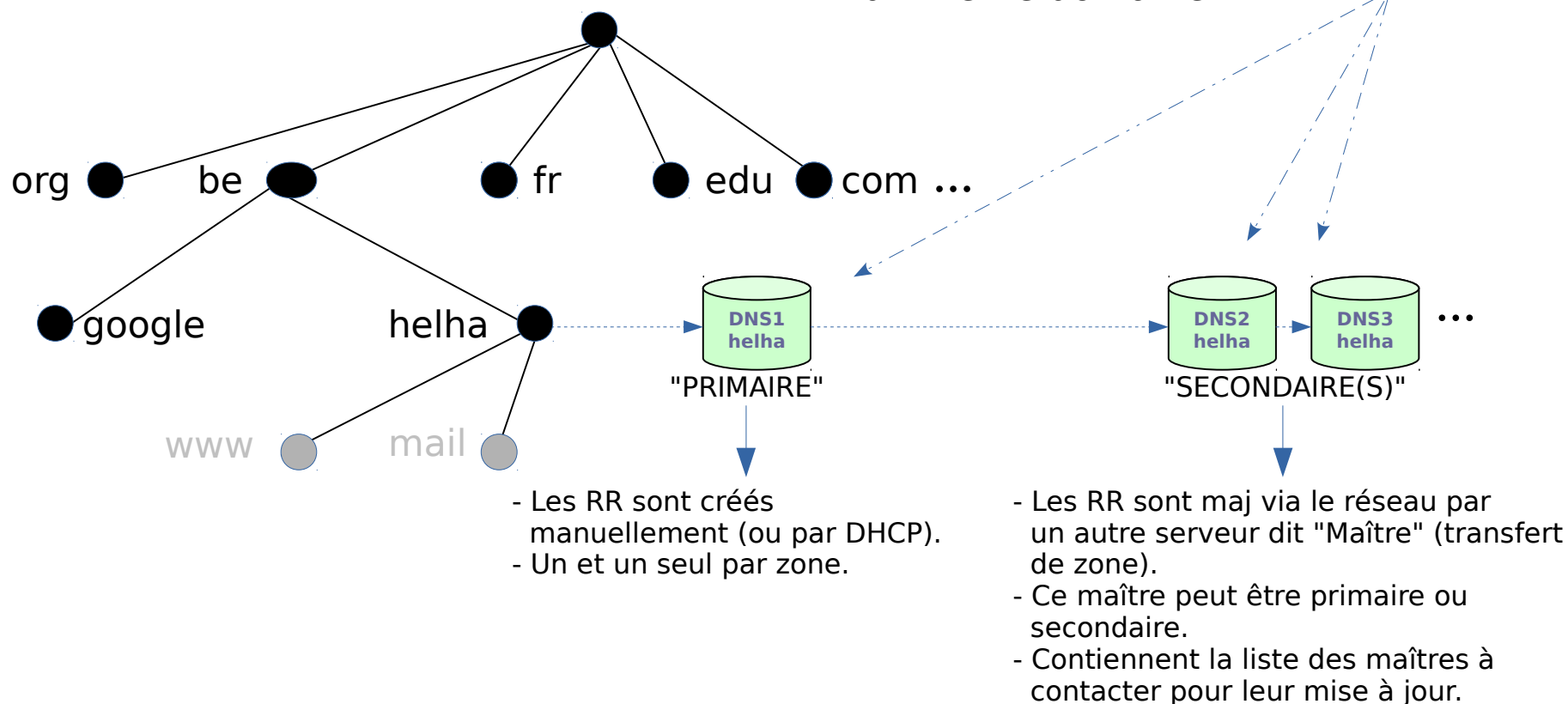
Plus d'info. sur l'anycast et le protocole BGP :

<https://vincent.bernat.im/fr/blog/2011-dns-anycast>

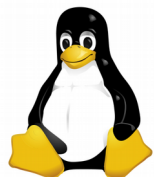


Plan du DNS: Redondance

Dans la réalité, plusieurs serveurs de noms font très souvent autorité pour un même domaine.

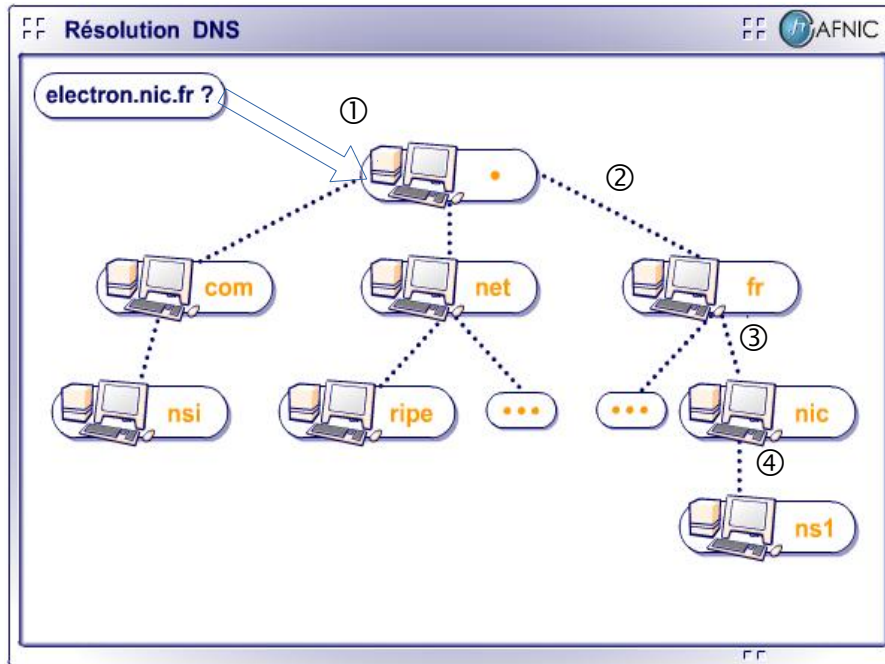


- Backup (meilleure tolérance aux pannes)
- Loadbalancing (répartition des charges possible)

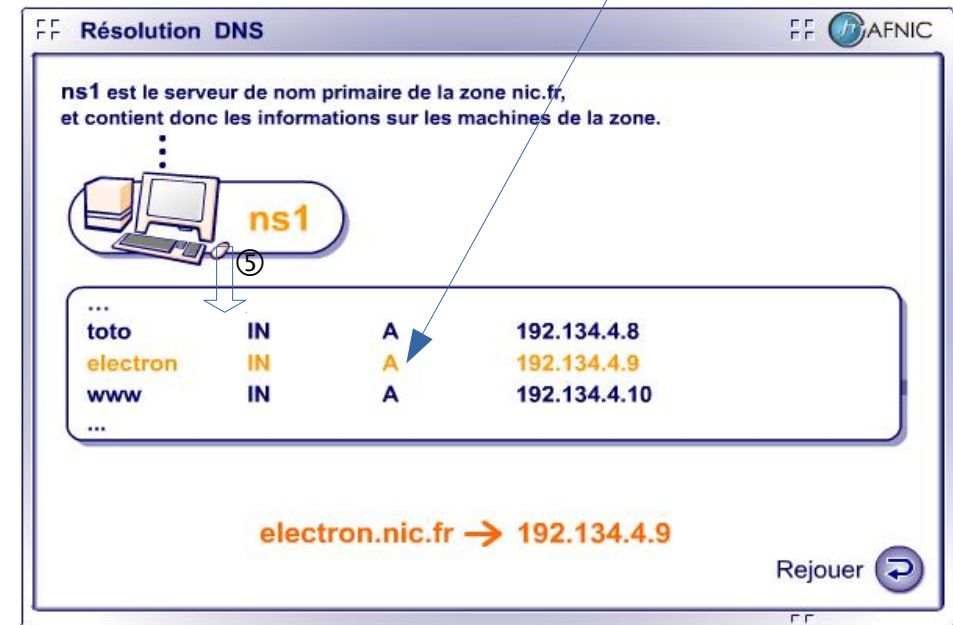


Résolution et résolution inverse

- Comment fait le DNS pour retrouver une adresse Ip à partir d'un nom de machine ?

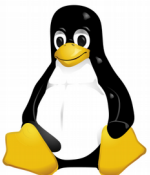


Ressource record (RR) de type A



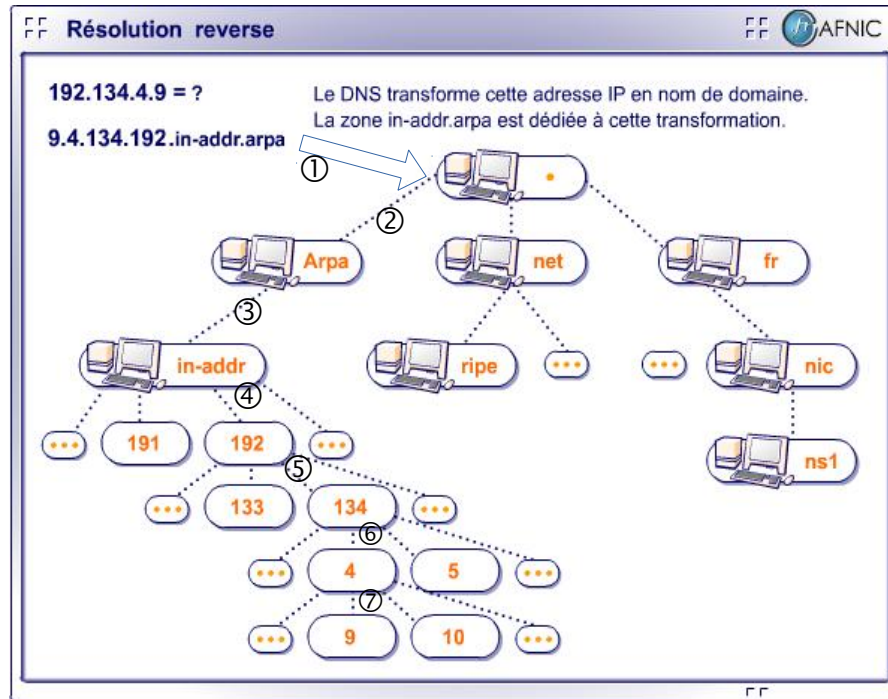
<https://www.afnic.fr/ext/dns/html/cours241.html>

Cours complet: [cours239.html](#) → [cours248.html](#)

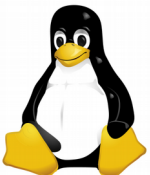
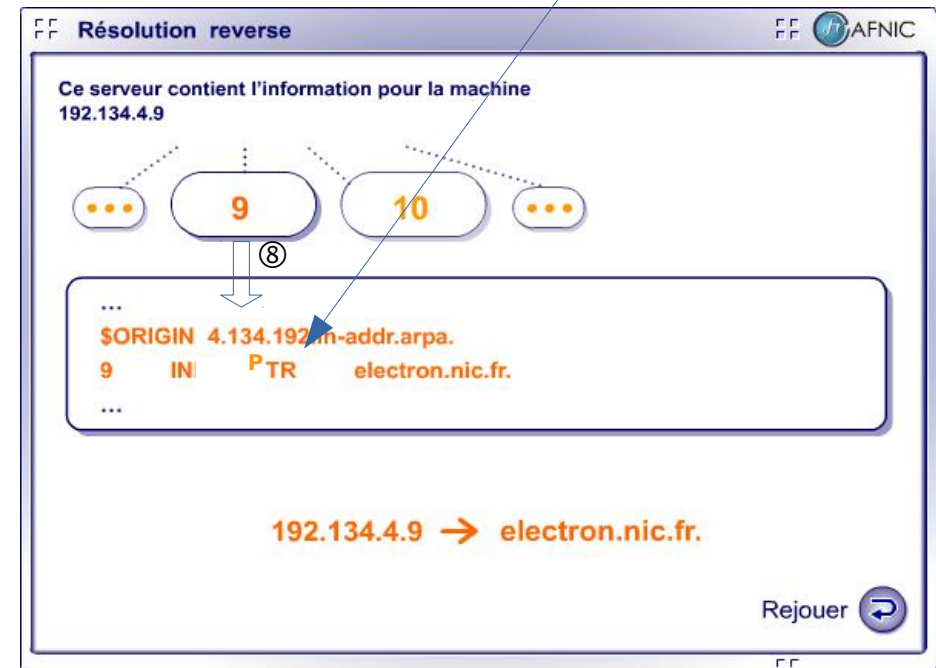


Résolution et résolution inverse

- Comment fait le DNS pour retrouver un nom de machine à partir d'une adresse Ip ?



Ressource record (RR) de type PTR



Résolution et résolution inverse

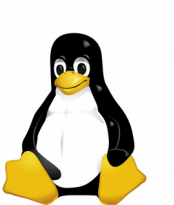
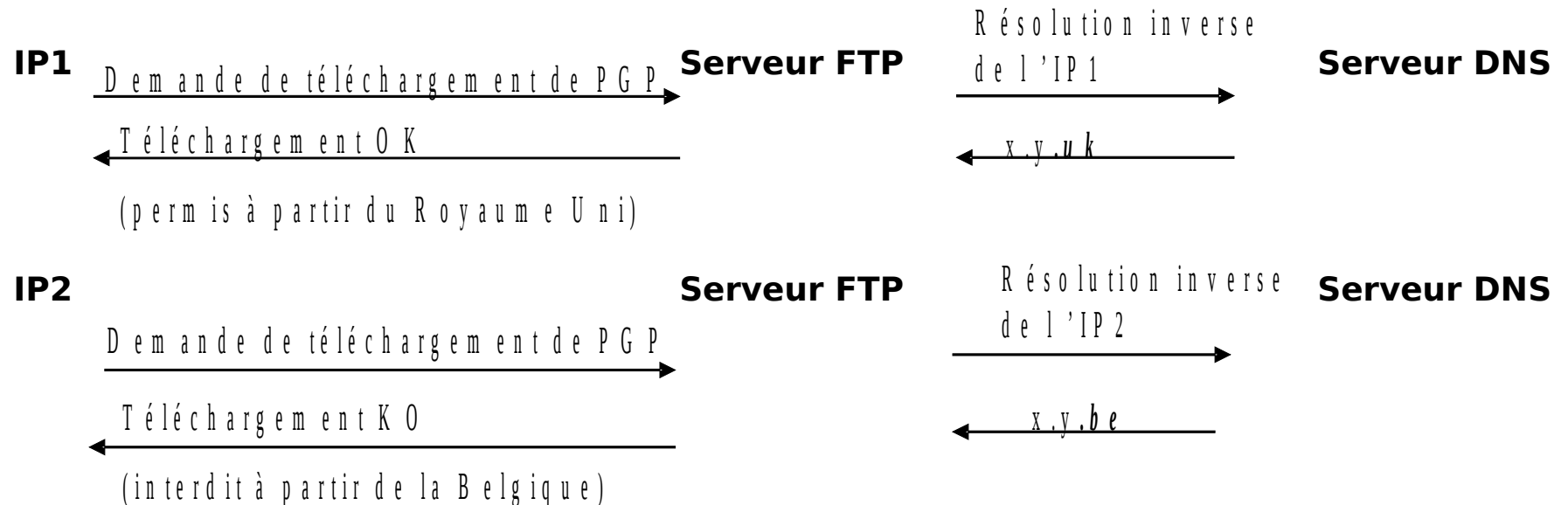
Résolution inverse: Utilités

- Utile pour restreindre l'accès à des services Internet, permettre les règles anti-spam des serveurs de messagerie ...

- Exemple

Soit un logiciel de cryptographie PGP installé sur un serveur FTP.
FTP est configuré pour que ce logiciel ne soit téléchargeable qu'à partir de certains pays ...

La résolution inverse sera donc sollicitée par le serveur FTP au serveur DNS.



Parcours d'une requête

- Le "resolver" permet de communiquer avec les serveurs DNS
- 2 modes d'interrogation:

Récuratif: Le client envoie une requête à un serveur, ce dernier devant interroger tous les autres serveurs nécessaires pour renvoyer la réponse complète au client (mode utilisé par les machines clientes en général).

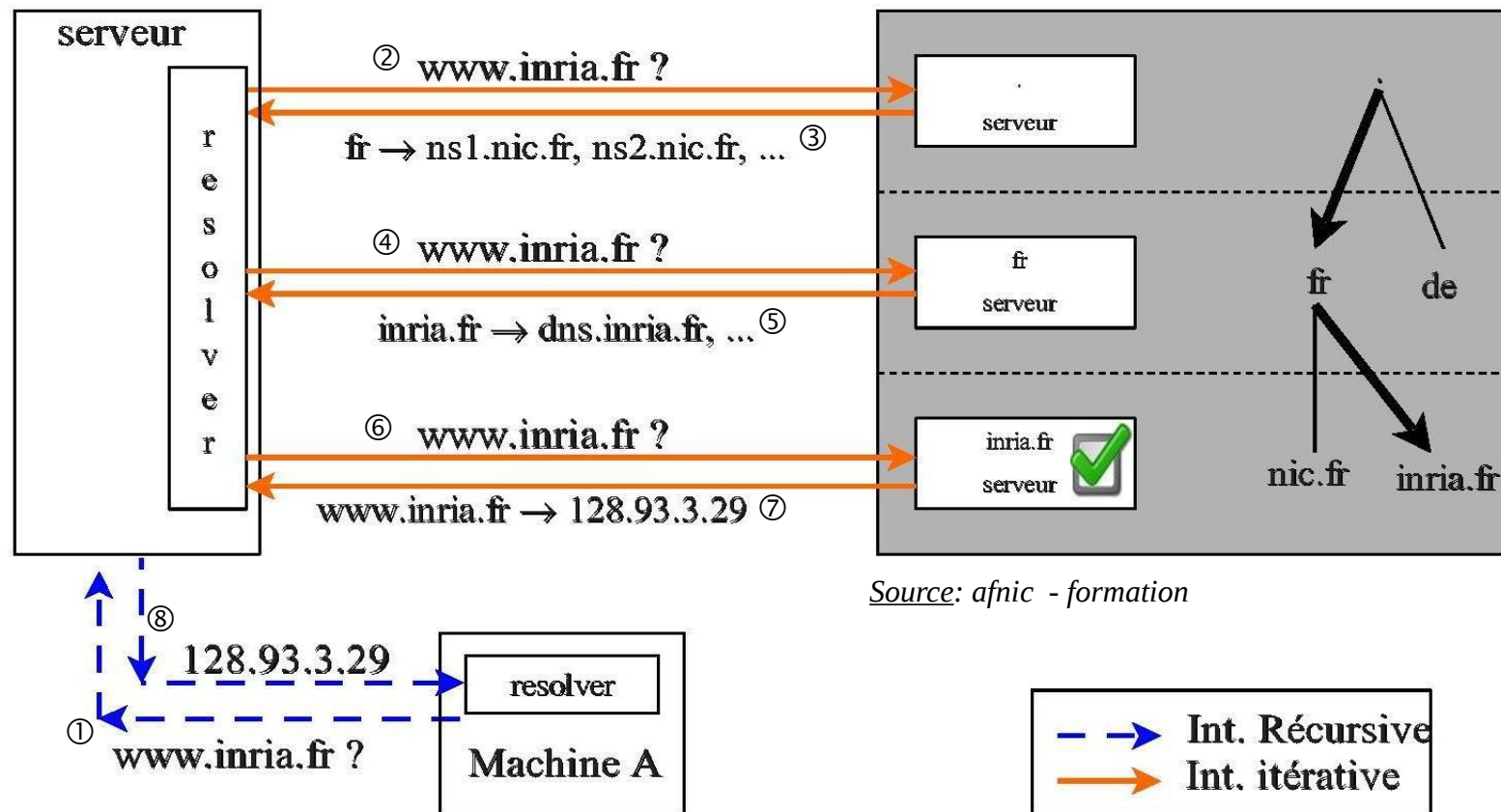
Une requête récurative attend une réponse définitive à une résolution de noms.

Itératif: Le client envoie une requête à un serveur, ce dernier renvoyant la réponse si il la connaît, ou le nom d'un autre serveur qu'il suppose plus renseigné pour résoudre cette question (mode utilisé par le resolver des serveurs en général).

Une requête itérative attend pour réponse la réponse elle-même ou bien une référence vers un autre serveur DNS.

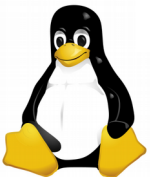


Parcours d'une requête



Source: afnic - formation

- Ici, il y a eu 4 interrogations pour résoudre `www.inria.fr`
- Mécanisme accélérateur: le cache (voir plus loin)

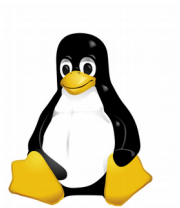


Remarque: DNS récursif ouvert

- Il est fortement conseillé de ne pas laisser votre DNS récursif ouvert.
- C'est-à-dire ne pas permettre la récursion sur votre DNS à partir d'Internet.
- Votre DNS n'acceptera de résoudre des noms qu'à partir votre réseau local.



Ne pas être DNS relais.
Accroissement de la sécurité (hors cadre du cours)



Debug DNS

dig (**D**omain **I**nformation **G**roper)

man dig

dig → donne la liste des serveurs racines

dig @server name type → donne les informations concernant une ressource (name) d'un certain type (type) d'un certain serveur Dns (@server).

Atelier 1 (dig):

Suivre la chaîne des délégations entre les zones à partir de la racine jusqu'à l'atteinte du nom Dns demandé soit ici `www.reseaucerta.org` pour connaître son Ip.

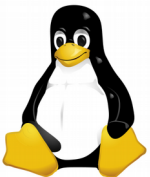
dig @e.root-servers.net www.reseaucerta.org A → on choisit un serveur racine

...

dig @d0.org... www.reseaucerta.org A → on choisit un serveur ayant autorité sur org.

dig @a.dns.gandi.net www.reseaucerta.org A → on choisit un serveur ayant autorité sur reseaucerta.org.

`www.reseaucerta.org 86400 IN A 194.254.4.9` → adresse ip recherchée



Debug DNS

Atelier 2 (dig):

Toujours en suivant la chaîne des délégations entre les zones à partir de la racine, tentez de résoudre un nom qui n'existe pas.

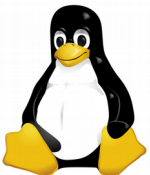
```
# dig @e.root-servers.net www.xxyyzz.be A → on choisit un serveur racine  
...
```

```
# dig @brussels.ns.dns.be www.xxyyzz.be A → on choisit un serveur ayant autorité  
... sur be.
```

```
...  
;; ->> HEADER ... status : NXDOMAIN → Non eXistant DOMAIN
```

Autres cas

```
# dig www.helha.be  
# dig +trace www.helha.be → Recherche à partir de la racine  
# dig lesoir.be MX  
# dig -x 204.13.162.123 → Recherche inverse
```



nslookup

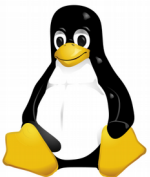
man nslookup

Cette commande peut être utilisée en mode interactif

Atelier 3 (nslookup):

Suivre la chaîne des délégations entre les zones à partir de la racine jusqu'à l'atteinte du nom Dns demandé soit ici www.reseaucerta.org pour connaître son Ip.

```
# nslookup
> server e.root-servers.net      → on choisit un serveur racine
> set type=NS                   → on s'intéresse aux records de type NS
> org.                           → quels sont les dns qui gèrent org. ?
...
> server d0.org...               → on passe sur un de ces serveurs
> reseaucerta.org.              → quels sont les dns qui gèrent reseaucerta.org. ?
...
> server a.dns.gandi.net         → on passe sur un de ces serveurs
> set type=A                     → on s'intéresse aux records de type A
> www.reseaucerta.org            → quelle est l'Ip de www.reseaucerta.org ?
...
```

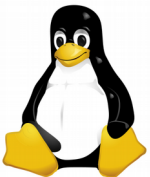


Debug DNS

> set type=MX	→ on s'intéresse aux records de type MX
> reseaucerta.org.	→ quels sont les serveurs de mail de reseaucerta.org ?
...	
> set type=A	→ on s'intéresse aux records de type A
> smtp.reseaucerta.org.	→ quelle est l'ip du serveur smtp de reseaucerta.org ?
...	
> set type=ANY	→ on s'intéresse à tout
> reseaucerta.org.	
...	

Autres cas

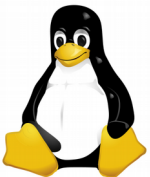
# nslookup www.lelibre.be	→ C'est le Dns par défaut qui est choisi pour résoudre le nom.
# nslookup www.lelibre.be 109.88.203.3	→ C'est un autre Dns qui est choisi.
# nslookup 91.121.208.164	→ Recherche inverse résolue avec le Dns par défaut.



Atelier 4 (nslookup):

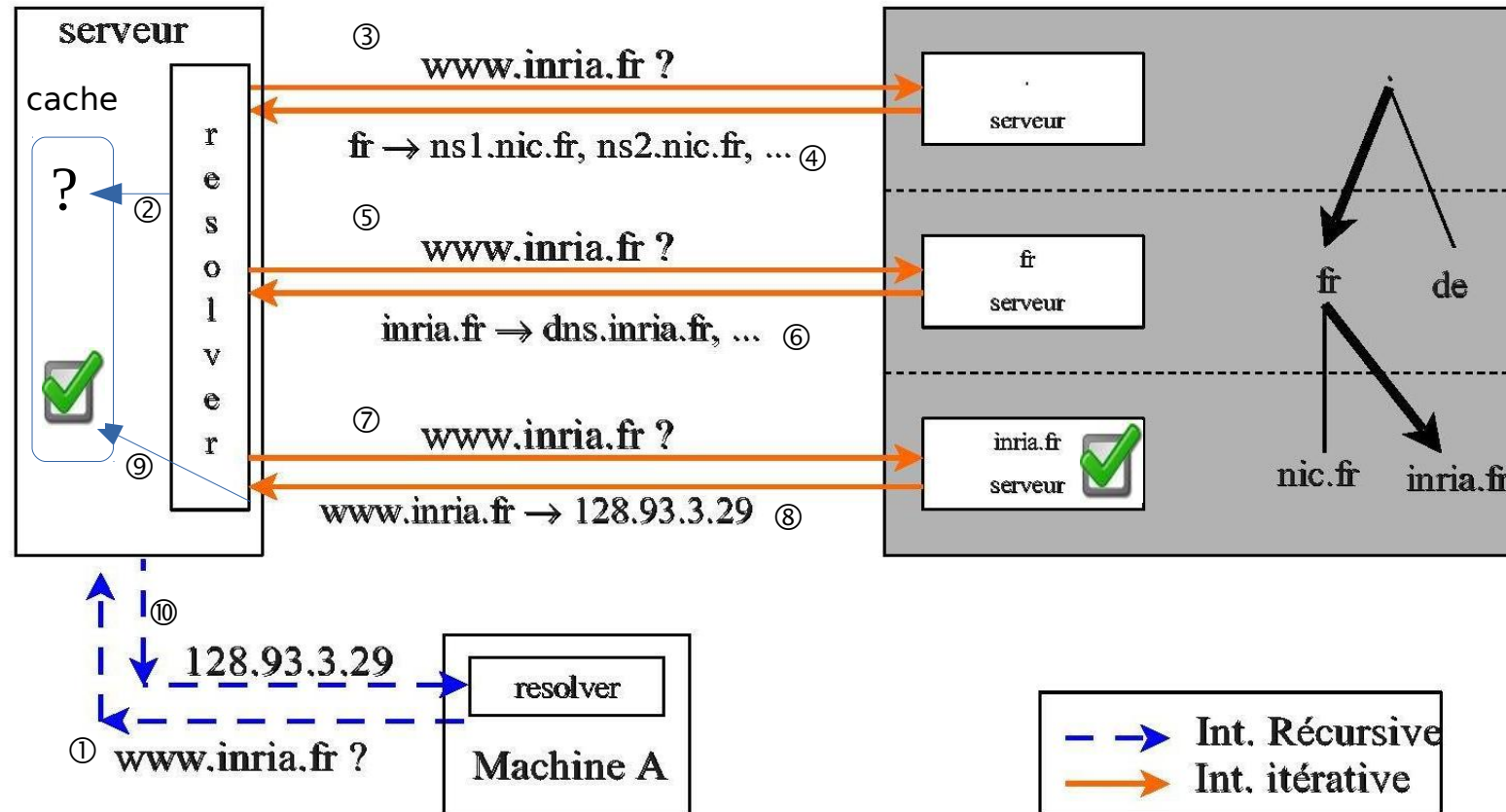
Toujours en suivant la chaîne des délégations entre les zones à partir de la racine, tentez de réaliser une recherche inverse (soit résoudre 91.121.208.164)

# nslookup	
> server e.root-servers.net	→ on choisit un serveur racine
> set type=PTR	→ on s'intéresse aux records de type PTR
> 91.121.208.164	→ y a-t-il un RR de ce type dans sa zone in-addr.arpa ?
...	→ Et le serveur ne me répond pas directement, mais
...	m'envoie une liste de serveurs ont autorité sur les
...	adresses qui commencent par 91
> server dns12.ovh.net	→ on en choisit un au hasard ...
> 91.121.208.164	→ ... et on lui repose la même question
...	
...	→ Ici, on a déjà la réponse. Cela signifie que dns12.ovh.net
	a autorité sur toutes les Ips du domaine 91.in-addr.arpa
	car il s'agit d'un réseau de classe A



Serveur de cache

Architecture

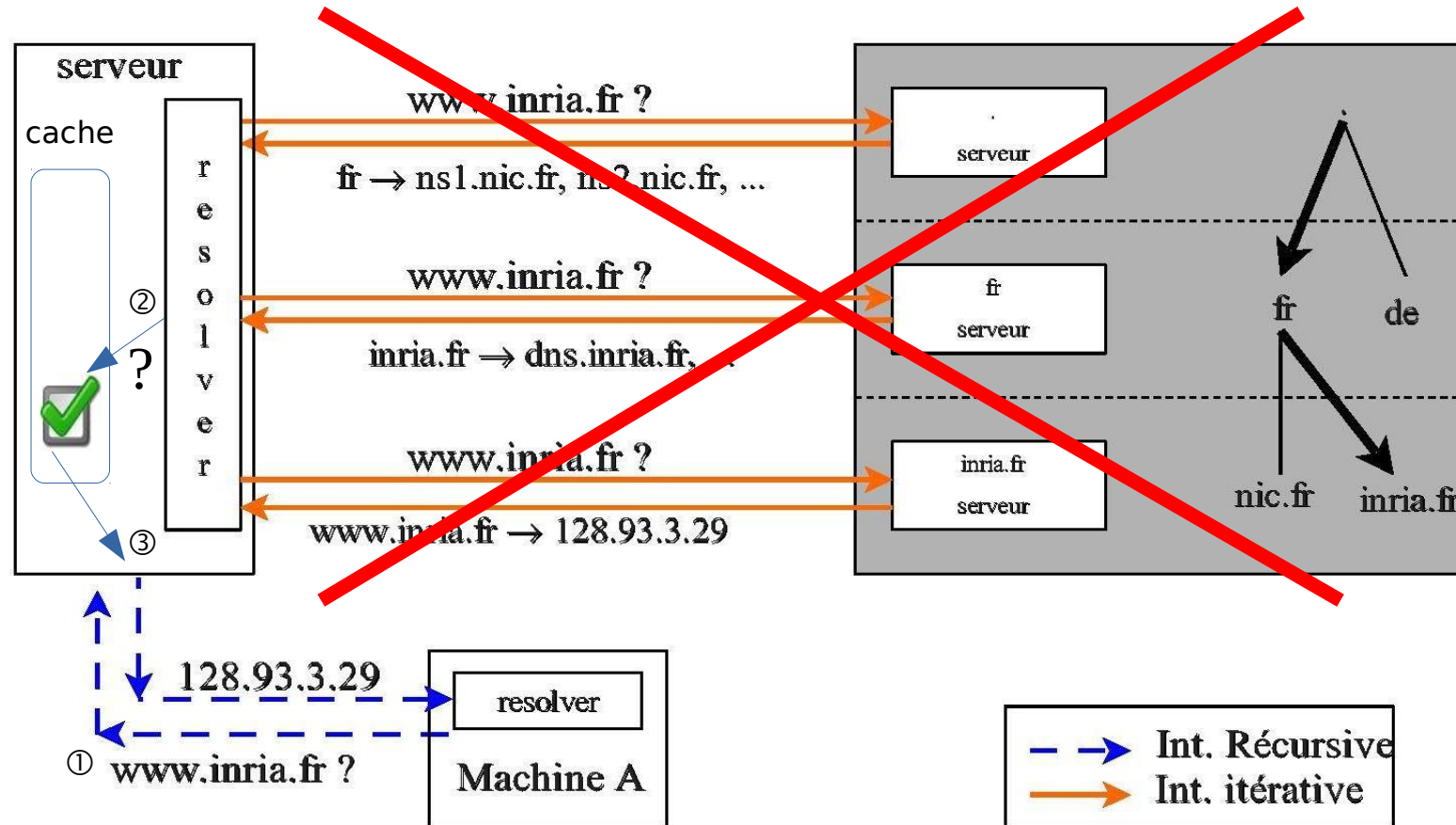


- Ici, le resolver du dns interroge d'abord son cache
- Si celui-ci est vide, il interroge alors les dns extérieurs de manière itérative ou fera appel à des 'forwarders' afin de résoudre la requête (voir plus loin)
- Une fois résolue, l'association Ip/Nom est mise en cache

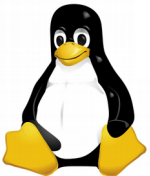


Serveur de cache

Architecture: suite



- Lorsque la réponse à la requête est déjà cache, l'extérieur n'est pas sollicité.
 - Le resolver du dns la transmet au client à partir de ce cache.
- Il s'agit d'une Non-authoritative answer.



Serveur de cache

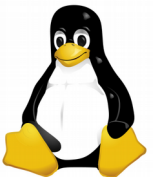
Avantages

- éviter la surcharge inutile du réseau
- supprimer les délais du réseau
- amoindrir la charge des autres serveurs

→ tout serveur possède en général au minimum un cache

Inconvénient

- ne pas oublier de sécuriser le cache



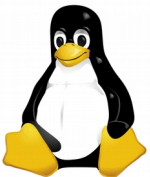
Configuration minimale

- Base de données nécessaire
 - adresses des serveurs de la racine
 - reverse du loopback 1.0.0.127.in-addr.arpa (c'est elle qui fait office de zone de cache)
- Les données du cache possèdent une durée de vie limitée (Time To Live - ttl) afin de permettre son rafraîchissement et la prise en compte des modifications.
- Il s'enrichit au fur et à mesure par les données récoltées pour résoudre les requêtes des clients.
 - une requête déjà demandée est résolue à partir du cache du serveur
- Ce type de serveur n'a autorité sur aucune zone.



Présentation

- Bind (Berkeley Internet Name Daemon)
- Serveur de noms le plus utilisé sur Internet.
- Bind 9 supporte l'IPv6, les noms de domaine unicode, le multithread et de nombreuses améliorations de sécurité.
- Maintenu actuellement par Paul Vixie avec l'Internet Software Consortium.
<http://www.isc.org>
- Dernière version stable: Bind 9.14

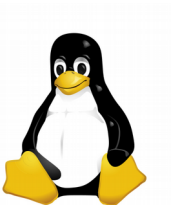


Packages

- bind – bind-utils – bind-libs
- bind-chroot: pour travailler dans un environnement "chrooté" (hors cadre)

Composants

- Le daemon *named*
 - c'est le service Dns → # `systemctl start/stop/restart named`
 - port d'écoute (udp 53)
- Fichiers de configuration
 - */etc/named.conf* → fichier de configuration principal
 - */var/named* → dossier par défaut qui contient les fichiers de zones
- Outils de debuggage:
 - dig, nslookup, host ...* (inclus dans le package *bind-utils*)
- Points d'entrée: *man named, man /etc/named.conf*



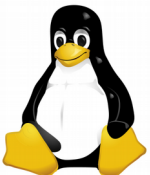
Serveur cache

named.conf

```
options {  
    listen-on port 53 {127.0.0.1 ; 192.1.0.2 ; } ; // Port d'écoute, ip admises  
    directory "/var/named"; // répertoire des fichiers de zones  
};  
  
zone "." IN {  
    type hint;  
    file "named.ca"; // cache des serveurs racines  
};  
  
zone "1.0.0.127.in-addr.arpa" IN {  
    type master;  
    file "named.loopback"; // zone primaire du reverse loopback  
};  
  
zone "localhost" IN {  
    type master;  
    file "named.localhost";  
};
```

// Zone primaire du loopback.
// Facultative sauf si on veut faire résoudre le nom 'localhost' par le serveur.

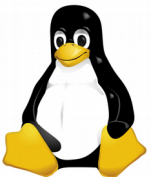
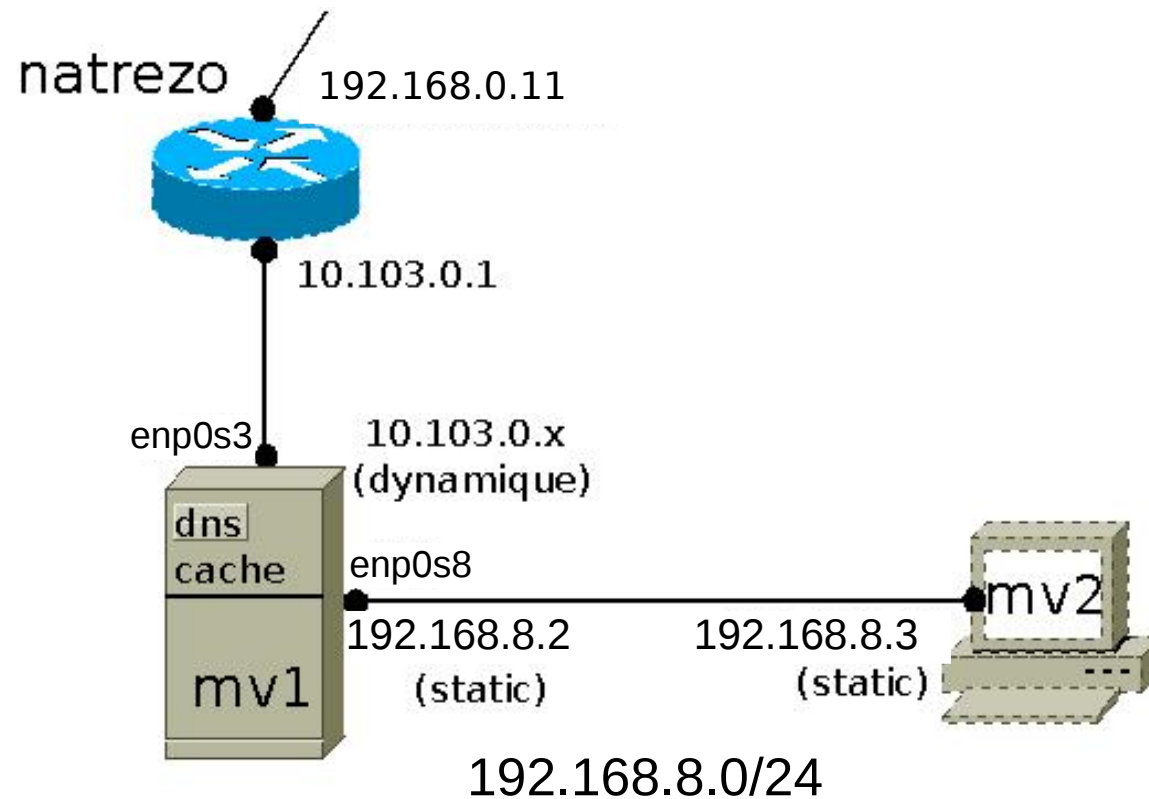
Ces 3 fichiers de zones seront automatiquement créés lors de l'installation du package 'bind'...



Exercices: Serveur de cache

Exercice 1: Serveur de cache

- Réalisez la maquette suivante :



Exercices: Serveur de cache

Exercice 1: Serveur de cache (suite)

- Installez les packages de Bind sur MV1.
- Configurez le fichier `/etc/named.conf` sur MV1.
- Configurez les resolvers des 2 machines.

Sur MV1:

```
/etc/resolv.conf  
nameserver 127.0.0.1
```

Sur MV2:

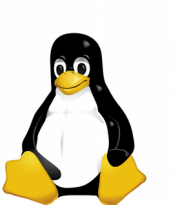
```
/etc/resolv.conf  
nameserver 192.168.8.2
```

↓ ↓
Liste des serveurs à contacter pour résoudre un nom - max 3.

- Vérifiez le fichier `/etc/nsswitch.conf` des 2 machines.

Sur MV1 & MV2

```
/etc/nsswitch.conf  
...  
hosts : files dns ...
```



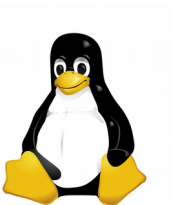
Exercices: Serveur de cache

Exercice 1: Serveur de cache (suite)

- Vérifiez le fichier `/etc/hosts` des 2 machines.

Sur MV1 & MV2
`/etc/hosts`
`127.0.0.1 localhost`

- Lancez votre dns + vérification des logs dans `/var/log/messages`
+ vérification via `named-checkconf` et `named-checkzone`
- Vérifiez le bon fonctionnement de votre dns à l'aide de `nslookup` et `dig`.
(Videz le cache entre vos manipulations `nslookup` et `dig` ...)
- Tentez de résoudre une requête dns à partir de MV2.
- Lancez `tshark` sur MV2 et espionnez une requête dns.
- Lancez `tshark` sur MV1 et espionnez une requête dns.
 - a) qui ne se trouve pas encore en cache
 - b) qui se trouve déjà en cache



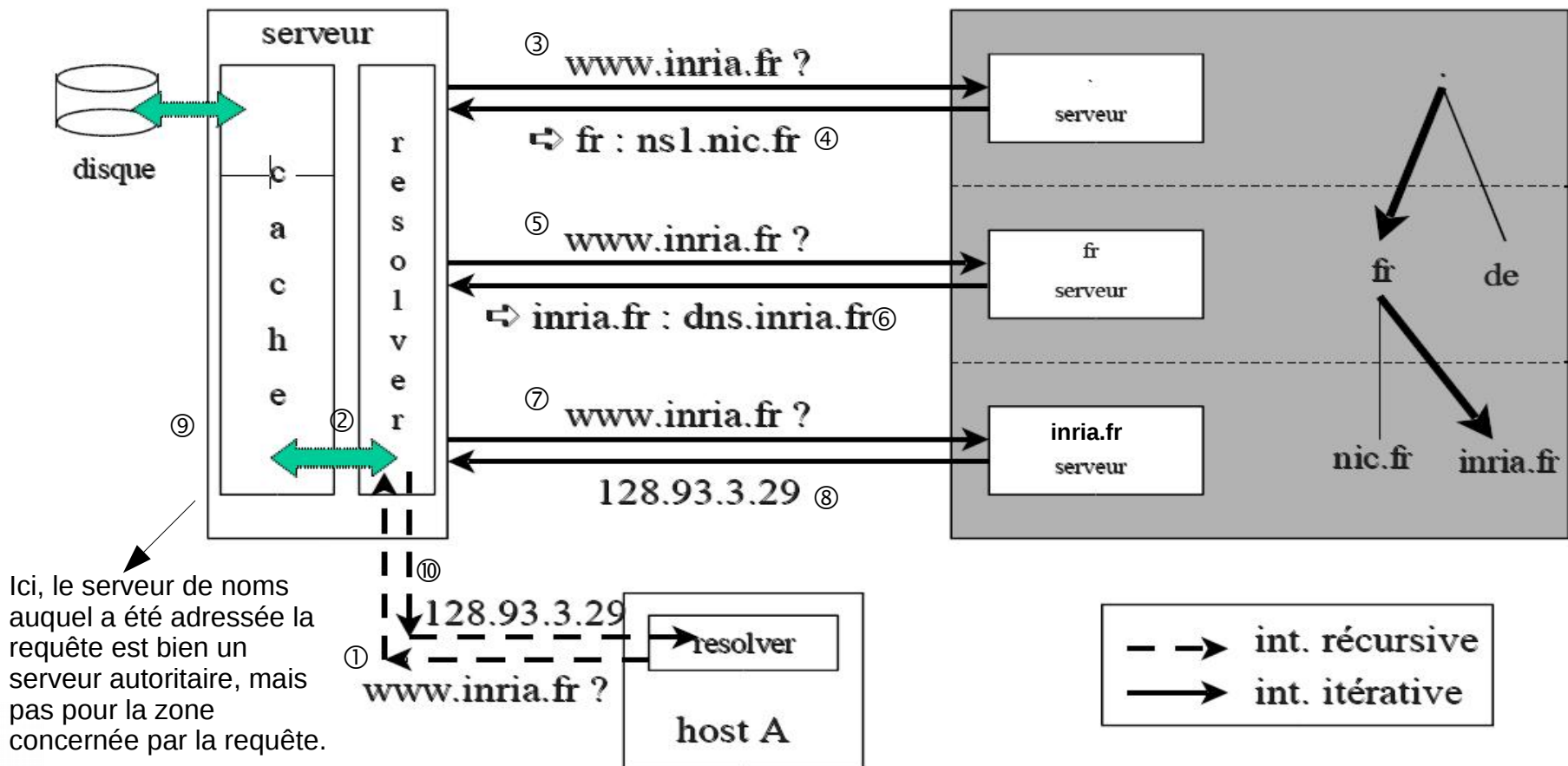
Serveur autoritaire

Architecture

Résolution d'une requête récursive par une suite de requêtes itératives envoyées vers les serveurs autoritaires des zones 'root' ('.'), 'fr' et 'inria.fr'.

Serveur ayant autorité sur le domaine afnic.fr

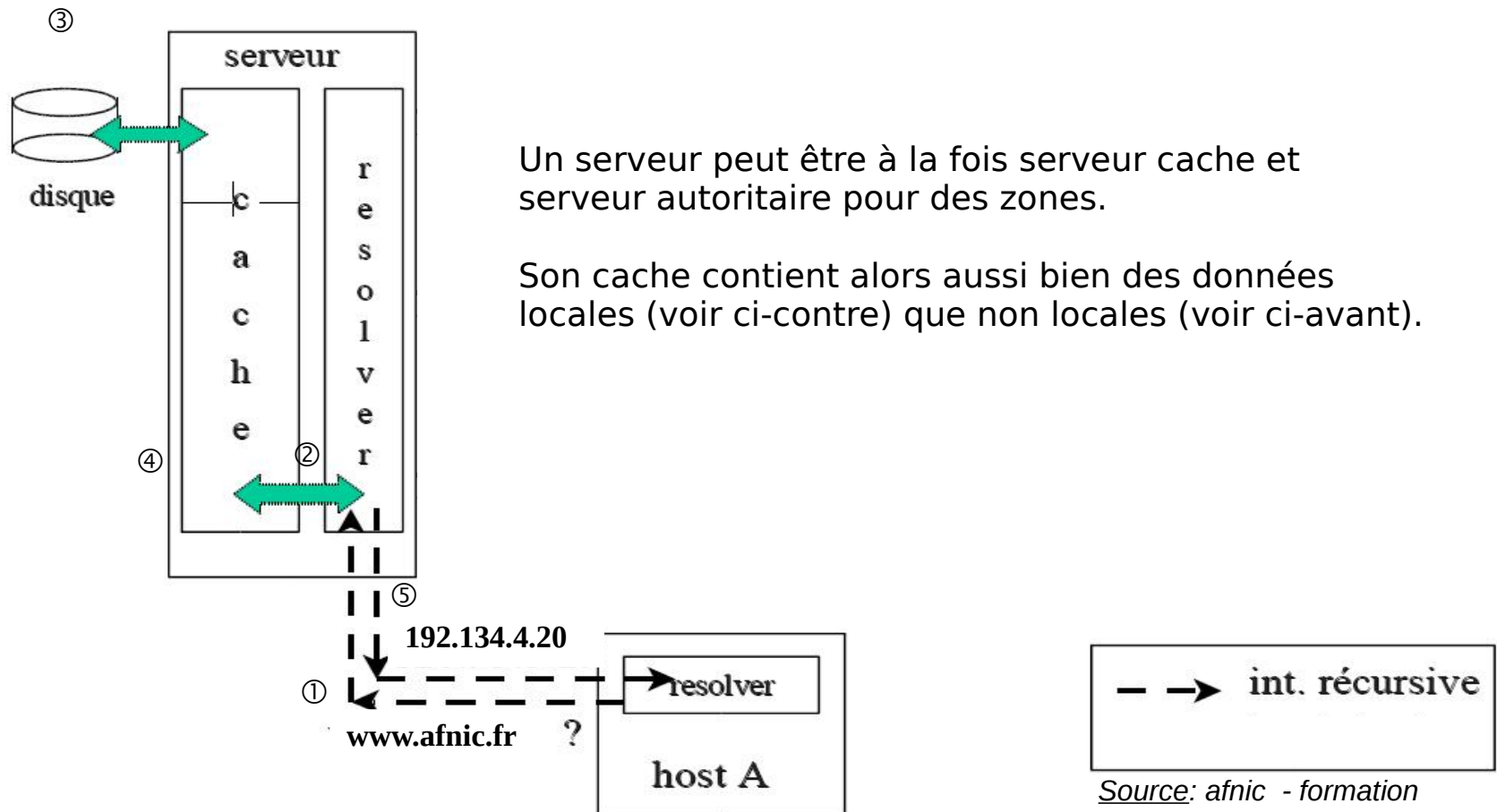
Source: afnic - formation



Serveur autoritaire

Architecture (suite)

Résolution d'une requête récursive envoyée directement vers le serveur autoritaire la zone concernée.



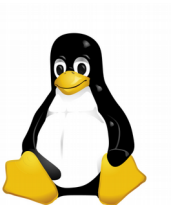
Remarques

Récuratif

- le serveur résout les requêtes récursives des clients et garde les informations obtenues dans son cache
 - le cache stocke des informations pour lesquelles le serveur n'a pas nécessairement autorité
- serveurs cache de campus par exemple

Itératif

- il répond toujours en fonction des données qu'il possède localement
 - ne construit pas de cache pour des données non locales .
 - une machine cliente (d'utilisateur final) ne doit jamais pointer sur un serveur de ce type comme serveur par défaut.
- mode permettant de limiter la charge d'un serveur (il ne résout pas toute la requête)
 - serveurs de la racine, serveurs ayant autorité pour un grand nombre de zones.

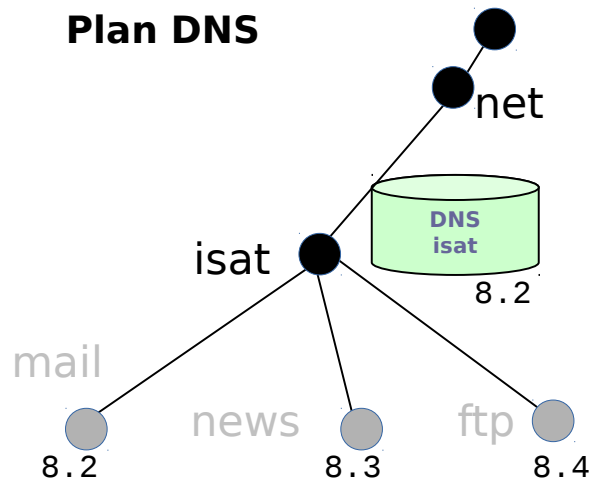


Bind

Gérer un domaine

Soit construire un serveur dns gérant le domaine 'isat.net' suivant:

Plan DNS



Chaque machine devra être capable de répondre à une requête via son nom dns.

ex. ping ftp.isat.net devra fonctionner au sein de votre réseau 192.168.8.0/24

Chaque machine devra toujours être capable d'utiliser l'internet.

Les services

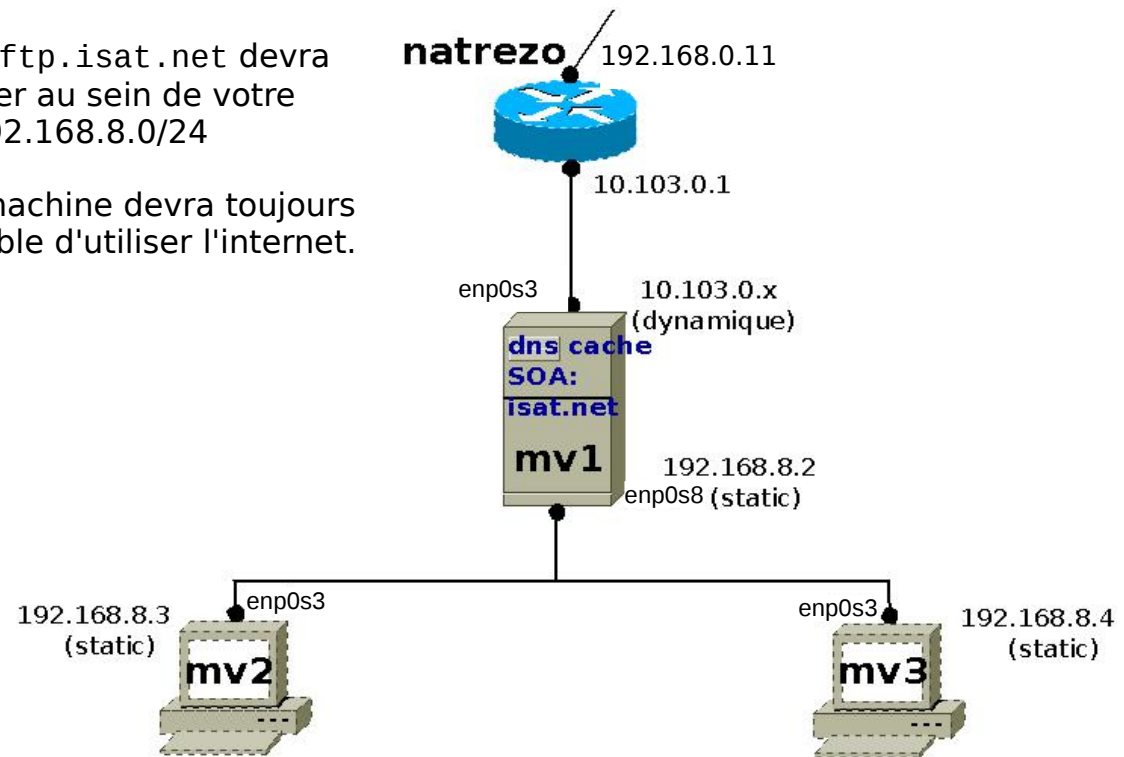
mail.isat.net sur 192.168.8.2
news.isat.net sur 192.168.8.3
ftp.isat.net sur 192.168.8.4

Alias sur les noms canoniques

r2d2.isat.net sur 192.168.8.2
yoda.isat.net sur 192.168.8.3
lea.isat.net sur 192.168.8.4

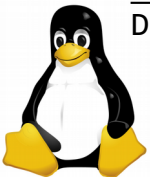
Le DNS

De la zone isat.net sur 192.168.8.2



192.168.8.0/24

Dns (JL Gouwy)



Serveur autoritaire récursif

named.conf

```
options {  
    listen-on port 53 {127.0.0.1 ; 192.168.8.2 ; } ; // Port d'écoute, ip admises  
    directory "/var/named";  
    recursion yes;           //récursif (yes par défaut)  
};
```

```
zone "." IN {  
    type hint;  
    file "named.ca";  
};
```

```
zone "1.0.0.127.in-addr.arpa" IN {  
    type master;  
    file "named.loopback";  
};
```

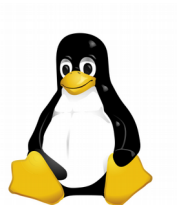
```
zone "localhost" IN {  
    type master;  
    file "named.localhost";  
};
```

→ (Suite)

(Suite)

```
zone "isat.net" IN {  
    type master;  
    file "db.isat.net";  
}; // zone primaire isat.net
```

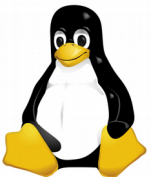
```
zone "8.168.192.in-addr.arpa" IN {  
    type master;  
    file "db.isat.net-rev";  
}; // zone primaire du reverse isat.net
```



Serveur autoritaire itératif

named.conf

```
options {  
    listen-on port 53 {127.0.0.1 ; 192.168.8.2 ; } ; // Port d'écoute, ip admises  
    directory "/var/named";  
    recursion no;           // le serveur n'accepte aucune requête récursive mais uniquement les  
                           // requêtes itératives  
};  
  
...  
idem ci-avant  
...
```



Bind

Syntaxe d'un RR:

nom|@ **[TTL]** **[classe]** **type** **données** **[commentaire]**

Nom DNS auquel la ressource considérée est associée. Dans un fichier de zone, les noms se terminant par "." sont des FQDN tandis que les autres sont relatifs au nom de la zone.

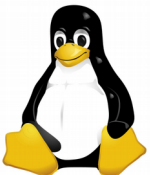
@ : utilise la valeur courante de \$ORIGIN comme nom (ou le nom de zone indiquée dans named.conf).

Nombre spécifiant la durée pendant laquelle le RR peut être conservé en cache. Si aucune valeur n'est indiquée, c'est le TTL par défaut de la zone qui est appliquée (en sec. Ou M-H-D-W).

Type de l'enregistrement (A - NS - MX - SOA ...).

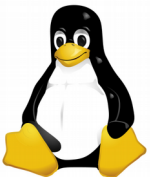
Données spécifiques au type d'enregistrement.

Internet (pour des raisons pratiques, c'est la seule classe importante). Si elle est absente, c'est IN par défaut qui est employée.

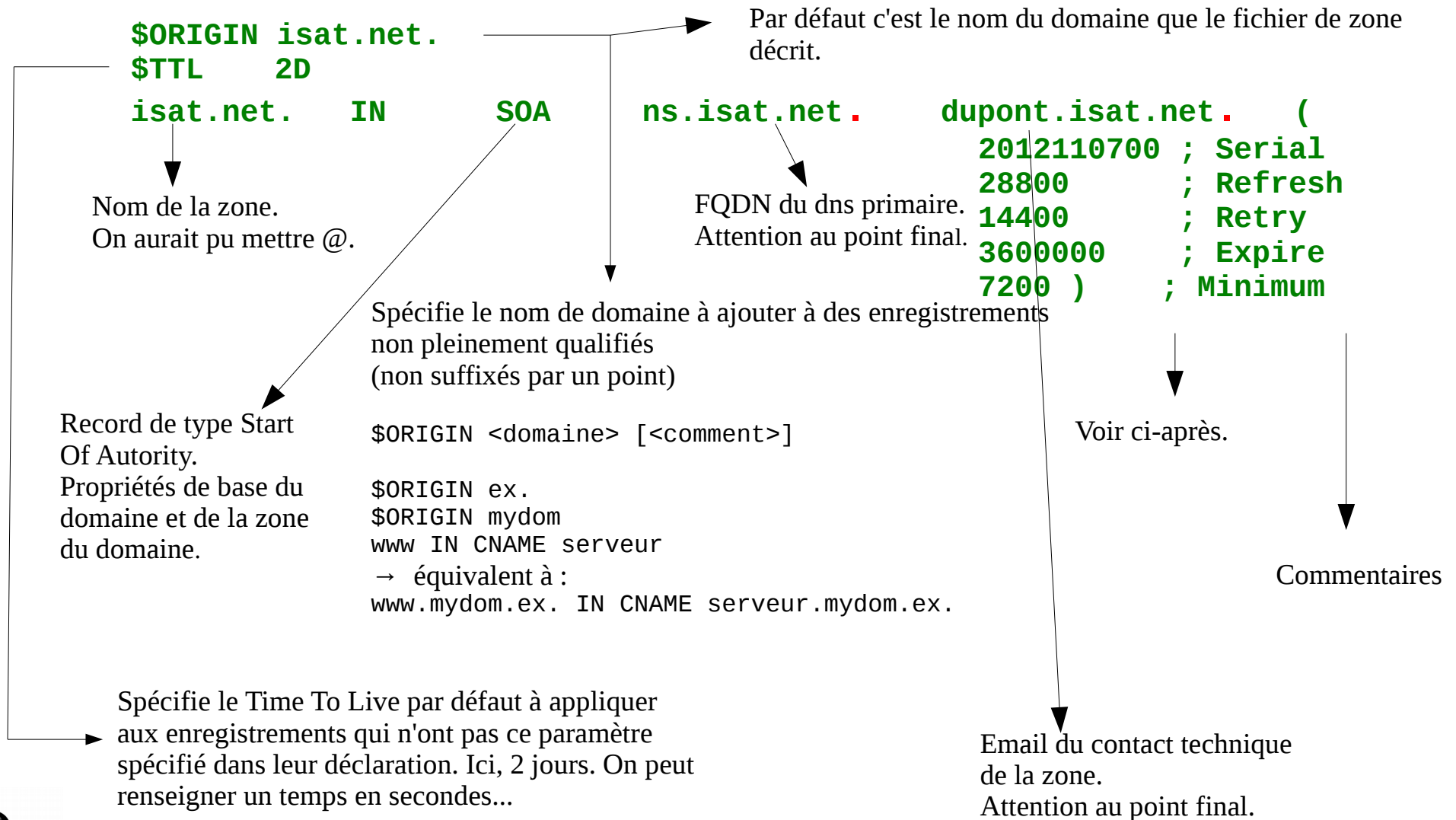


Les fichiers de zone: celui de la zone isat.net

```
$ORIGIN isat.net.  
$TTL      2D  
isat.net.  IN      SOA      ns.isat.net.  root.isat.net.  (  
                                2012110700 ; Serial  
                                28800      ; Refresh  
                                14400      ; Retry  
                                3600000    ; Expire  
                                7200 )     ; Minimum  
  
                                IN      NS      ns.isat.net.  
  
ns         IN      A        192.168.8.2  
mail       IN      A        192.168.8.2  
news      IN      A        192.168.8.3  
ftp       IN      A        192.168.8.4  
  
r2d2      IN      CNAME    mail  
lea       IN      CNAME    ftp  
yoda      IN      CNAME    news  
  
isat.net.  IN      MX       10      mail
```



Les fichiers de zone: celui de la zone isat.net (suite)



Les fichiers de zone: celui de la zone isat.net (suite)

```
$ORIGIN isat.net.
```

```
$TTL      2D
```

```
isat.net.      IN          SOA          ns.isat.net.      dupont.isat.net.  (  
                2012110700 ; Serial  
                28800      ; Refresh  
                14400      ; Retry  
                3600000     ; Expire  
                7200 )      ; Minimum
```

Serial: Spécifie la version des données de la zone.
A incrémenter à chaque modification (nécessaire pour à
synchronisation des serveurs secondaires)
Conseil : YYYYMMDDxx → max. 99 modif./jour

Refresh: Intervalle, ici en sec., entre 2 vérifications du serial
number par les secondaires.

Retry: Intervalle, ici en sec., entre 2 vérifications du serial
number par les secondaires si la 1ere vérification a
échoué.

Expire: Temps, ici en sec., après lequel le secondaire détruit les
données de la zone qu'il possède et arrête de répondre aux
requêtes pour cette zone s'il ne parvient pas à
contacter le serveur primaire.

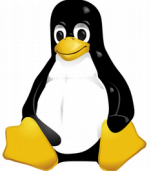
retry<<refresh<<expire

Minimum:

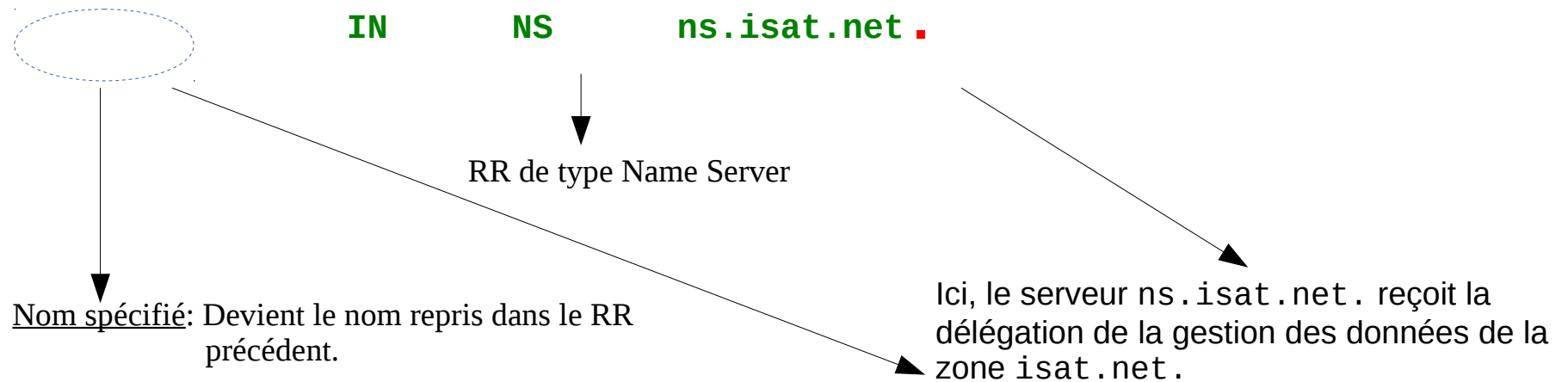
Temps que doit rester dans le cache une
réponse négative suite à une question sur ce
domaine.

Deux types de réponses négatives :

- NXDOMAIN : aucun RR ayant le nom
demandé dans la classe (IN) n'existe dans cette
zone.
- NODATA : aucune donnée pour le triplet
(nom, type, classe) demandé
n'existe ; il existe d'autres records possédant
ce nom, mais de type différent.

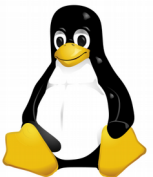


Les fichiers de zone: celui de la zone isat.net (suite)



```
zone      IN  NS  serveur-nom1.domaine.  
          IN  NS  serveur-nom2.domaine.  → serveur secondaire
```

Il faut spécifier les serveurs de noms de la zone que l'on décrit
(associée au SOA)



Les fichiers de zone: celui de la zone isat.net (suite)

ns	IN	A	192.168.8.2
mail	IN	A	192.168.8.2
news	IN	A	192.168.8.3
ftp	IN	A	192.168.8.4

RR de type IPv4 Address

Indique l'adresse IPv4 associée à un nom..
mail.isat.net. IN A 192.168.8.2

AAAA: Adresse IPv6

Les noms non pleinement qualifiés (ne se terminant pas par un point) sont relatifs au nom du domaine spécifié dans la directive \$ORIGIN

Donc, ici, les écritures :

mail.isat.net. IN A 192.168.8.2

et

mail IN A 192.168.8.2

sont équivalentes.



Les fichiers de zone: celui de la zone isat.net (suite)

```
r2d2      IN      CNAME  mail
lea       IN      CNAME  ftp
yoda      IN      CNAME  news
```

RR de type 'Canonical name'

Indique que le nom est un alias
vers un autre nom (le nom canonique)

Remarques:

alias IN CNAME nom.canonique.

- plusieurs alias différents peuvent pointer vers le même nom canonique

```
alias1 IN CNAME relais
alias2 IN CNAME relais
```

😊 si l'Ip associée au nom canonique change, seule l'Ip correspondant à ce nom canonique dans son RR de type A doit être changée.

- un nom canonique ne doit pas pointer vers un alias déjà créé

```
alias IN CNAME relais1
alias IN CNAME relais2
```



- quand un nom est déjà lié à un CNAME, il est interdit de faire figurer d'autres types de RR pour ce nom.

```
alias IN CNAME relais
alias IN NS serveur
```



Les fichiers de zone: celui de la zone isat.net (suite)

isat.net. IN MX 10 mail



RR de type 'Mail Exchanger'

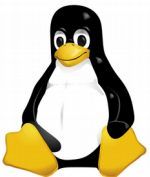
Spécifie un serveur de messagerie pour la zone : email à quelqu-un@nom

On cherche dans le DNS un MX indiquant la machine sur laquelle il faut envoyer le courrier pour nom.

Un paramètre précise le poids relatif de l'enregistrement MX:

Si plusieurs MX existent, le courrier est envoyé en 1er à la machine ayant le poids le plus bas, puis dans l'ordre croissant des poids en cas d'échec

```
nom IN MX 10 nom.relais1.  
      IN MX 20 nom.relais2.  
      IN MX 30 nom.relais3.
```



Les fichiers de zone: celui de la zone reverse 8.168.192.in-addr.arpa

```
$ORIGIN 8.168.192.in-addr.arpa.  
$TTL      2D  
8.168.192.in-addr.arpa.  IN      SOA      ns.isat.net. root.isat.net. (  
                                2012110700 ; Serial  
                                28800      ; Refresh  
                                14400      ; Retry  
                                3600000    ; Expire  
                                7200      ) ; Minimum  
  
                                IN      NS      ns.isat.net.  
  
2  IN      PTR      mail.isat.net.  
3  IN      PTR      news.isat.net.  
4  IN      PTR      ftp.isat.net.
```



RR de type 'Pointeur'

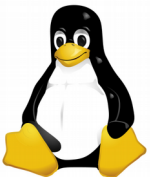
Indique le nom associé à un numéro IP dans l'arborescence in-addr.arpa (ip6.arpa)

```
2.8.168.192.in-addr.arpa. IN PTR mail.isat.net.
```



Les fichiers de zone: celui du reverse 0.0.127.in-addr.arpa

- Personne n'a la responsabilité de ce reverse pour le numéro 127.0.0.1 dans la hiérarchie in-addr.arpa.
- Doit toujours être configuré sous peine de comportement anormal du DNS.

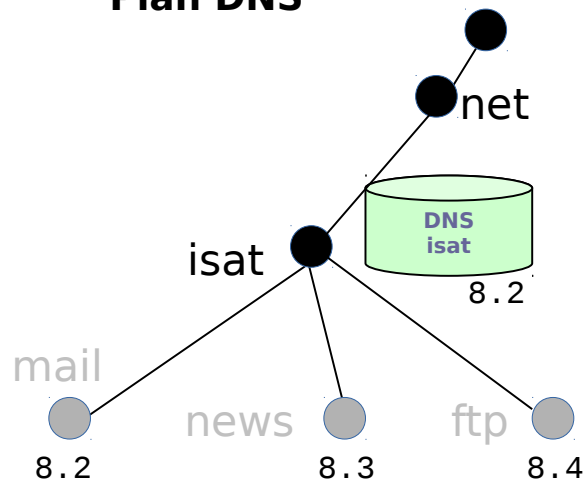


Exercice: Serveur autoritaire

Exercice 3: Serveur autoritaire de cache

- Construire le serveur dns gérant le domaine 'isat.net.' exposé ci-avant.

Plan DNS



Les services

mail.isat.net sur 192.168.8.2
news.isat.net sur 192.168.8.3
ftp.isat.net sur 192.168.8.4

Alias sur les noms canoniques

r2d2.isat.net sur 192.168.8.2
yoda.isat.net sur 192.168.8.3
lea.isat.net sur 192.168.8.4

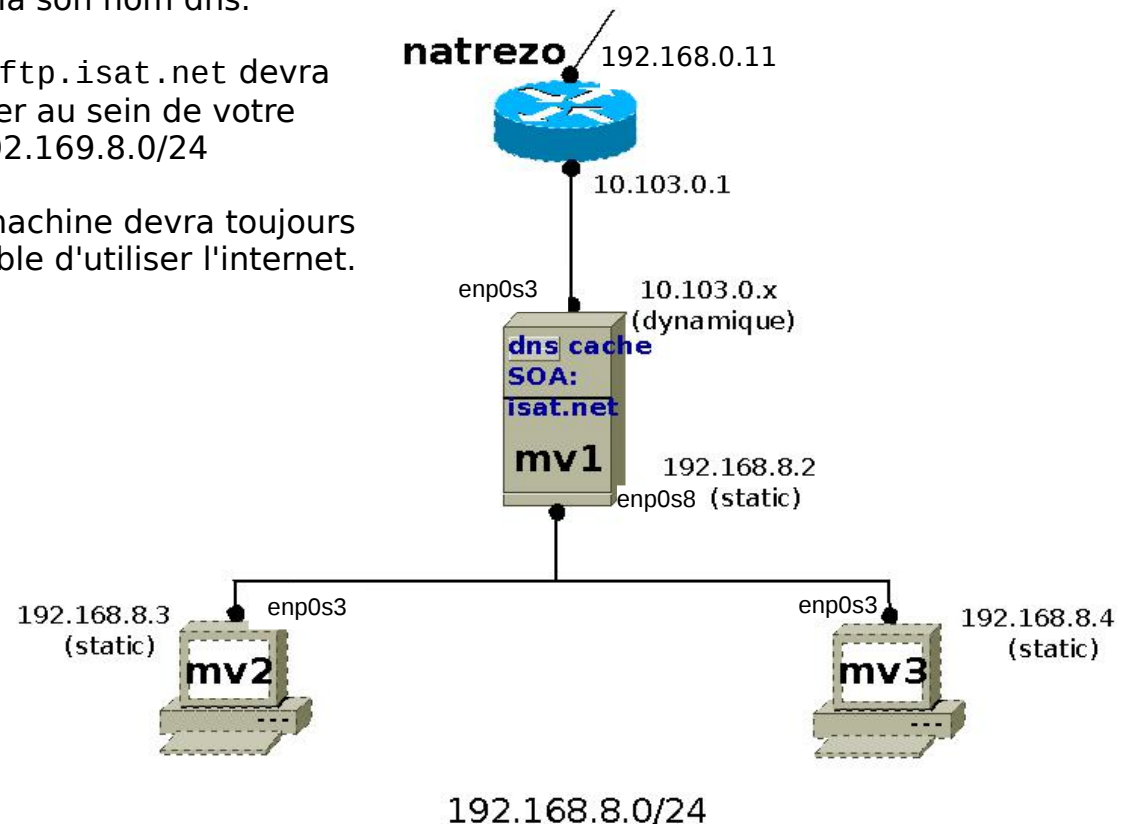
Le DNS

De la zone "isat.net" sur 192.168.8.2

Chaque machine devra être capable de répondre à une requête via son nom dns.

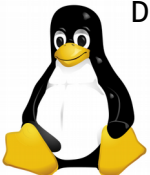
ex. ping ftp.isat.net devra fonctionner au sein de votre réseau 192.169.8.0/24

Chaque machine devra toujours être capable d'utiliser l'internet.



Dns (JL Gouwy)

192.168.8.0/24



Exercice: Serveur autoritaire

Exercice 3: Serveur autoritaire de cache

- Configurez votre dns.
- Fonctionnement (ping + outils de debugage):
 - . chaque mv doit pouvoir se toucher par son nom ou son alias.
 - . chaque mv a toujours l'accès à l'internet.
- Quelques essais et compléments
 - a) Enlevez 192.168.8.2.; de la directive listen-on et relancez named.
Que constatez-vous ?
 - b) Rajoutez à nouveau ce réseau et ajoutez la directive recursion no et relancez named. Que constatez-vous ?
 - c) Remplacez la directive recursion no par:
recursion yes → *facultatif car par défaut*
allow-recursion {127.0.0.1 ; 192.168.8.0/24;} ;
allow-query-cache {127.0.0.1 ; 192.168.8.0/24;} ;

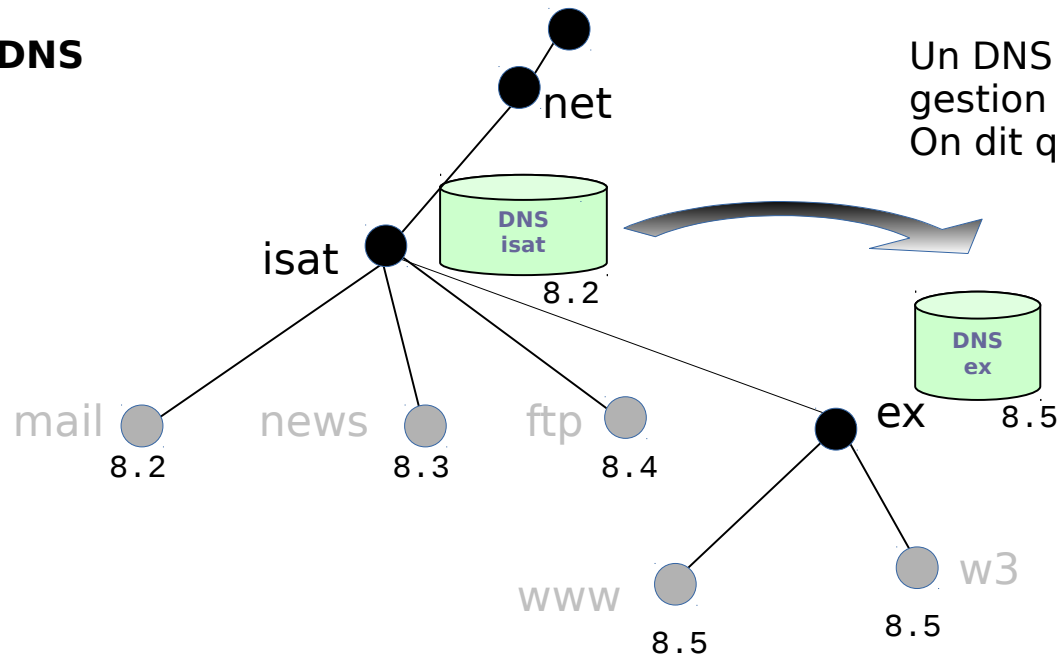
et relancez named. Que constatez-vous ?
 - d) Rajoutez la directive version "DNS ISAT" et relancez named.
Quelle pourrait-être son utilité ?



Délégation et sous domaine

Soit une nouvelle zone 'ex':

Plan DNS



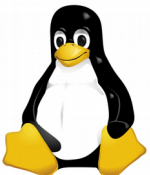
Les services

mail.isat.net sur 192.168.8.2
news.isat.net sur 192.168.8.3
ftp.isat.net sur 192.168.8.4

Les DNS

De la zone "isat.net" sur 192.168.8.2
De la zone "ex.isat.net" sur 192.168.8.5

www.ex.isat.net sur 192.168.8.5
w3.ex.isat.net sur 192.168.8.5



Délégation et sous domaine

Bind: Le fichier de la zone parente

```
$ORIGIN isat.net.  
$TTL      2D  
isat.net.      IN      SOA      ns.isat.net.  root.isat.net.  (  
                2017110700 ; Serial  
                28800      ; Refresh  
                14400      ; Retry  
                3600000    ; Expire  
                7200 )      ; Minimum  
  
                IN      NS      ns.isat.net.  
ex             IN      NS      ns.ex.isat.net.  
  
ns             IN      A       192.168.8.2  
mail          IN      A       192.168.8.2  
news          IN      A       192.168.8.3  
ftp           IN      A       192.168.8.4  
ns.ex         IN      A       192.168.8.5  
  
isat.net.     IN      MX      10      mail
```

La délégation de zone est déclarée dans le fichier de zone du domaine parent par un RR de type NS.

Et un RR de type A est ensuite nécessaire pour la correspondance entre l'adresse IP et le nom.

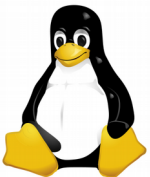


Délégation et sous domaine

Bind: Le fichier de la zone fille

```
$ORIGIN ex.isat.net.  
$TTL      2D  
ex.isat.net.  IN      SOA      ns.ex.isat.net.  root.ex.isat.net.  (  
                                2017110700 ; Serial  
                                28800      ; Refresh  
                                14400      ; Retry  
                                3600000    ; Expire  
                                7200 )      ; Minimum  
                                IN      NS      ns.ex.isat.net.  
  
ns           IN      A        192.168.8.5  
www          IN      A        192.168.8.5  
w3           IN      A        192.168.8.5
```

Le fichier de zone du sous-domaine est un fichier de zone classique.

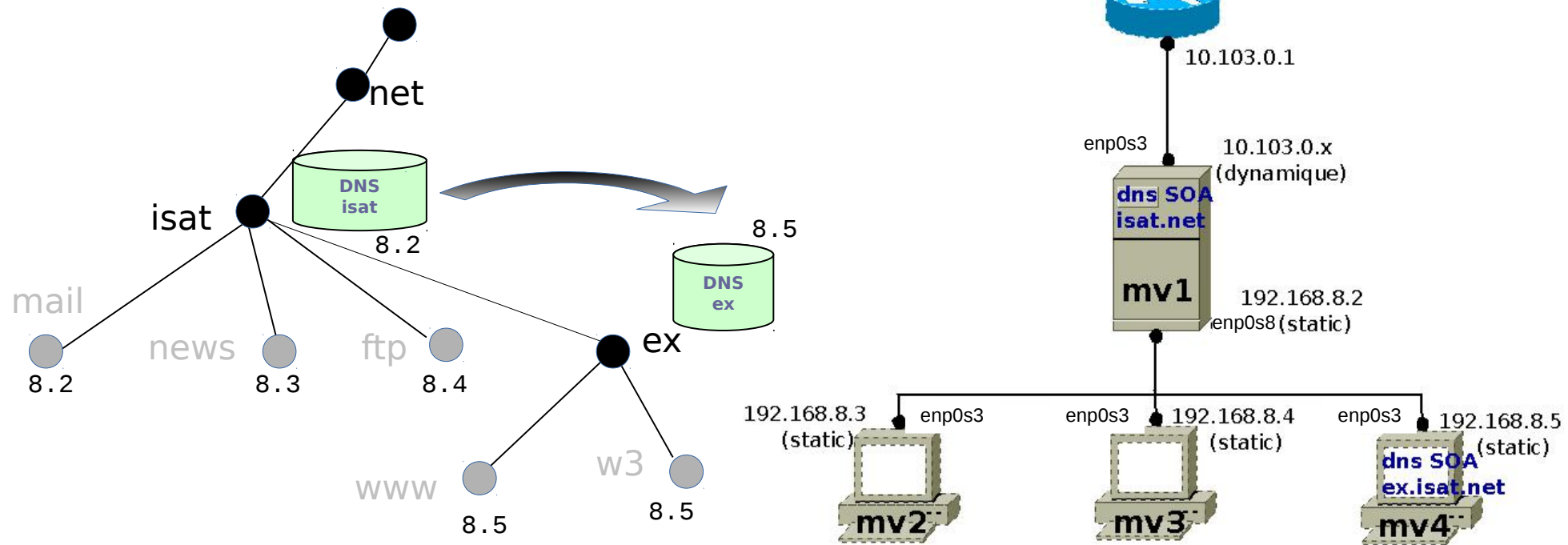


Exercice: Délégation et sous domaine

Exercice 4

- Construire les serveurs dns gérant les domaines isat.net. et ex.isat.net :

Plan DNS et architecture:

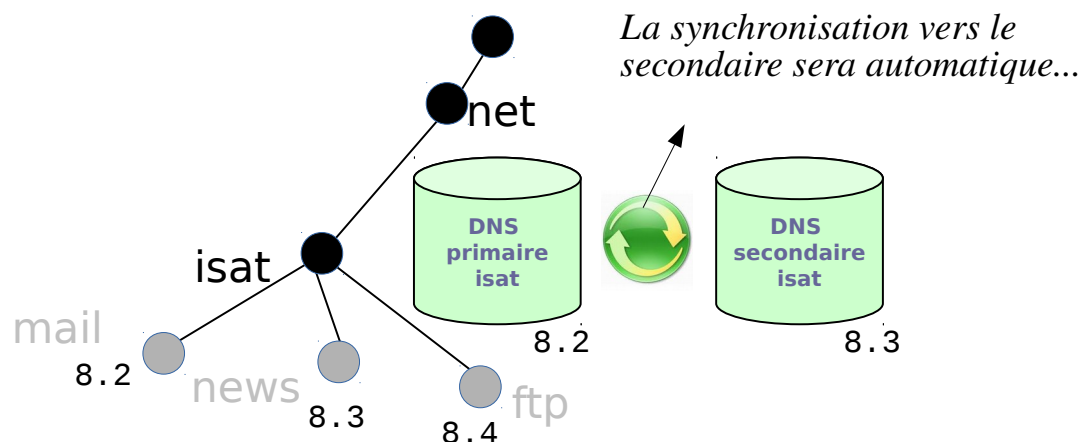
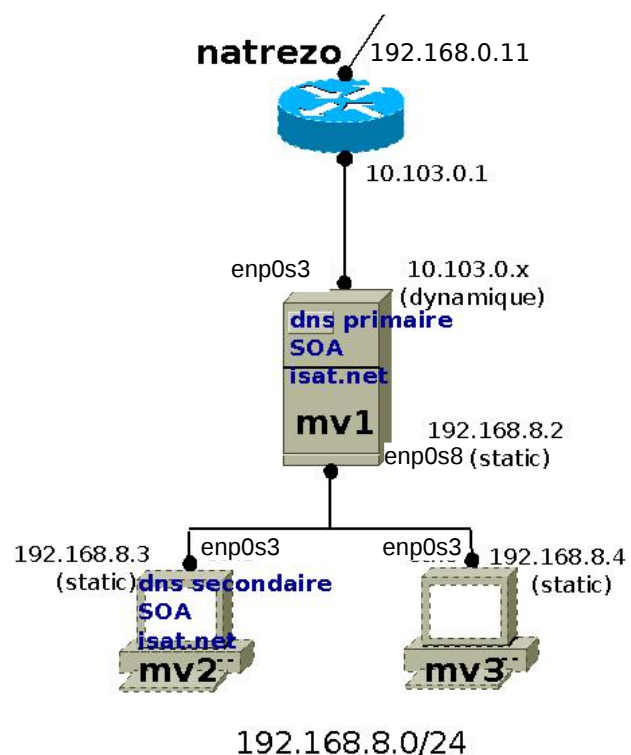


- Fonctionnement (ping + outils de debugage):
 - . chaque mv doit pouvoir se toucher par son nom ou son alias.
 - . chaque mv a toujours l'accès à l'internet.



Redondance

Soit ajouter un serveur dns secondaire pour le domaine 'isat.net':



- Un dns secondaire ne gère pas directement les informations sur les zones mais les obtient à partir du primaire de la zone (ou d'un autre secondaire) via le réseau (transfert de zone).
- Ce dns ne peut modifier des données de la zone mais il a lui aussi autorité sur la zone.
- Cette redondance permet une meilleure tolérance aux pannes et une réduction de la charge de travail des dns principaux.

2 façons pour mettre à jour un secondaire:

- en fonction de la valeur '*refresh*' définie dans le SOA de la zone ;
- ou lorsqu'il reçoit une notification du primaire

Quand il démarre, un secondaire doit connaître son maître pour entamer un transfert de zone avec lui.



Redondance

Configuration du serveur primaire: **named.conf**

```
options {  
    listen-on port 53 {127.0.0.1 ; 192.168.8.2 ; } ;  
    directory "/var/named";  
};
```

```
zone "." IN {  
    type hint;  
    file "named.ca";  
};
```

*Ces esclaves ont le droit de
demander le transfert de
ces zones au maître.*

```
zone "1.0.0.127.in-addr.arpa" IN {  
    type master;  
    file "named.loopback";  
};
```

```
zone "localhost" IN {  
    type master;  
    file "named.localhost";  
};
```

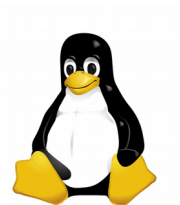
(Suite)

*Le maître a le devoir de signaler les
changements sur ces zones aux
esclaves.*

(Suite)

```
zone "isat.net" IN {  
    type master;  
    notify yes;  
    also-notify {192.168.8.3;} ;  
    allow-transfer {192.168.8.3;} ;  
    file "db.isat.net";  
}; // zone primaire isat.net
```

```
zone "8.168.192.in-addr.arpa" IN  
{    type master;  
    notify yes;  
    also-notify {192.168.8.3;} ;  
    allow-transfer {192.168.8.3;} ;  
    file "db.isat.net-rev";  
}; // zone primaire du reverse isat.net
```



Redondance

Configuration du serveur primaire: **db.isat.net**

```
$ORIGIN isat.net.
```

```
$TTL 2D
```

```
isat.net.  IN      SOA      ns.isat.net.  root.isat.net.  (
2017110701 ; Serial
28800      ; Refresh
14400      ; Retry
3600000    ; Expire
7200      ) ; Minimum
```

On "déclare" ns2.isat.net comme serveur secondaire.
Attention, le RR du serveur primaire (ns.isat.net) doit rester en première position (voir après).

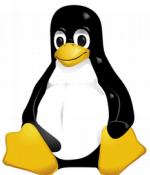
```
IN      NS      ns.isat.net.
IN      NS      ns2.isat.net.
```

```
ns      IN      A      192.168.8.2
ns2     IN      A      192.168.8.3
mail    IN      A      192.168.8.2
news    IN      A      192.168.8.3
ftp     IN      A      192.168.8.4
```

```
r2d2    IN      CNAME   mail
lea     IN      CNAME   ftp
yoda    IN      CNAME   news
```

```
isat.net.  IN      MX      10      mail
```

Chaque fois qu'une zone du maître est modifiée, ne pas oublier d'incrémenter le numéro de série (utilisé pour la synchronisation)



Redondance

Configuration du serveur primaire: **db.isat.net-rev**

```
$ORIGIN 8.168.192.in-addr.arpa.
```

```
$TTL 2D
```

```
8.168.192.in-addr.arpa. IN SOA ns.isat.net. root.isat.net. (
                                2017110701 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                7200      ) ; Minimum
```

```

      IN NS ns.isat.net.
3  IN NS ns2.isat.net.

2  IN PTR mail.isat.net.
3  IN PTR news.isat.net.
4  IN PTR ftp.isat.net.
```

On "déclare" également le serveur secondaire dans la zone reverse sans oublier d'incrémenter le numéro de série.



Redondance

Configuration du serveur secondaire: **named.conf**

```
options {  
    listen-on port 53 {127.0.0.1 ; 192.168.8.3 ; } ;  
    directory "/var/named";  
};
```

```
zone "." IN {  
    type hint;  
    file "named.ca";  
};
```

*Les zones transférées seront stockées
dans /var/named/slaves.
Le process named devra avoir la
permission d'y écrire.
Ce dossier sera garni par le serveur lui-même.*

```
zone "1.0.0.127.in-addr.arpa" IN {  
    type master;  
    file "named.loopback";  
};
```

```
zone "localhost" IN {  
    type master;  
    file "named.localhost";  
};
```

(Suite)

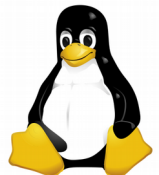
*Ces zones dites "non-vivantes" ne risquent pas changer.
ns2 peut alors en être maître. Elles sont identiques à celles du
primaire.*

*Liste des serveurs maîtres (utile pour
la maj par 'refresh').*

(Suite)

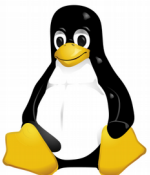
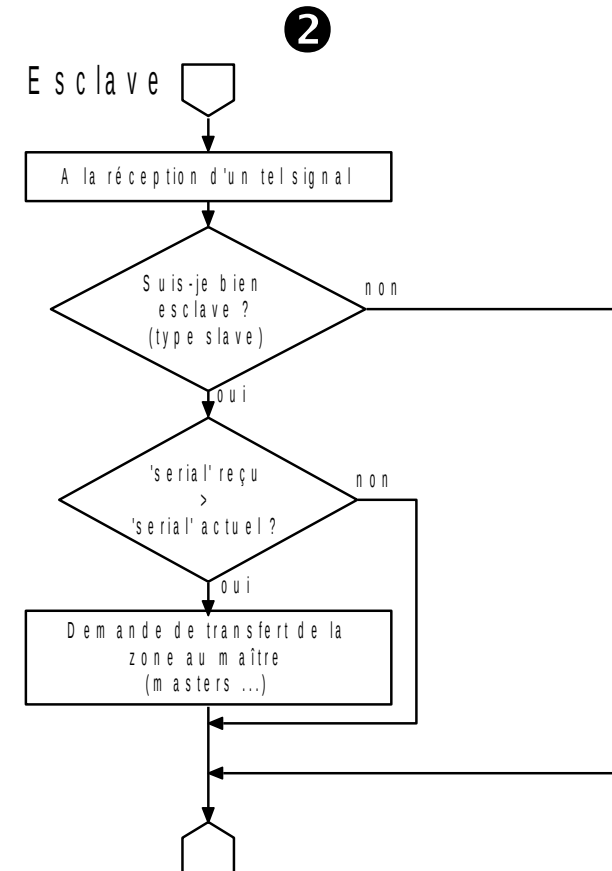
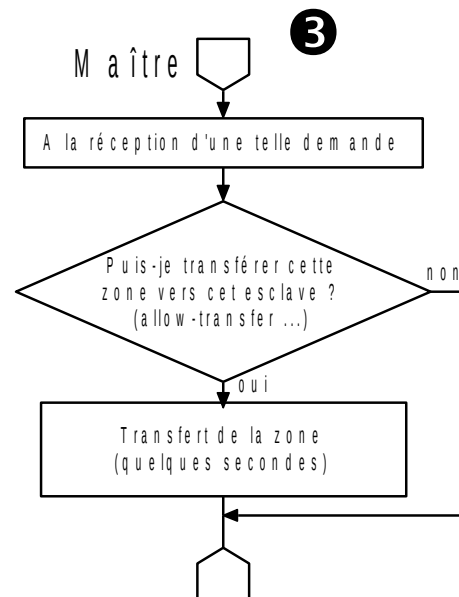
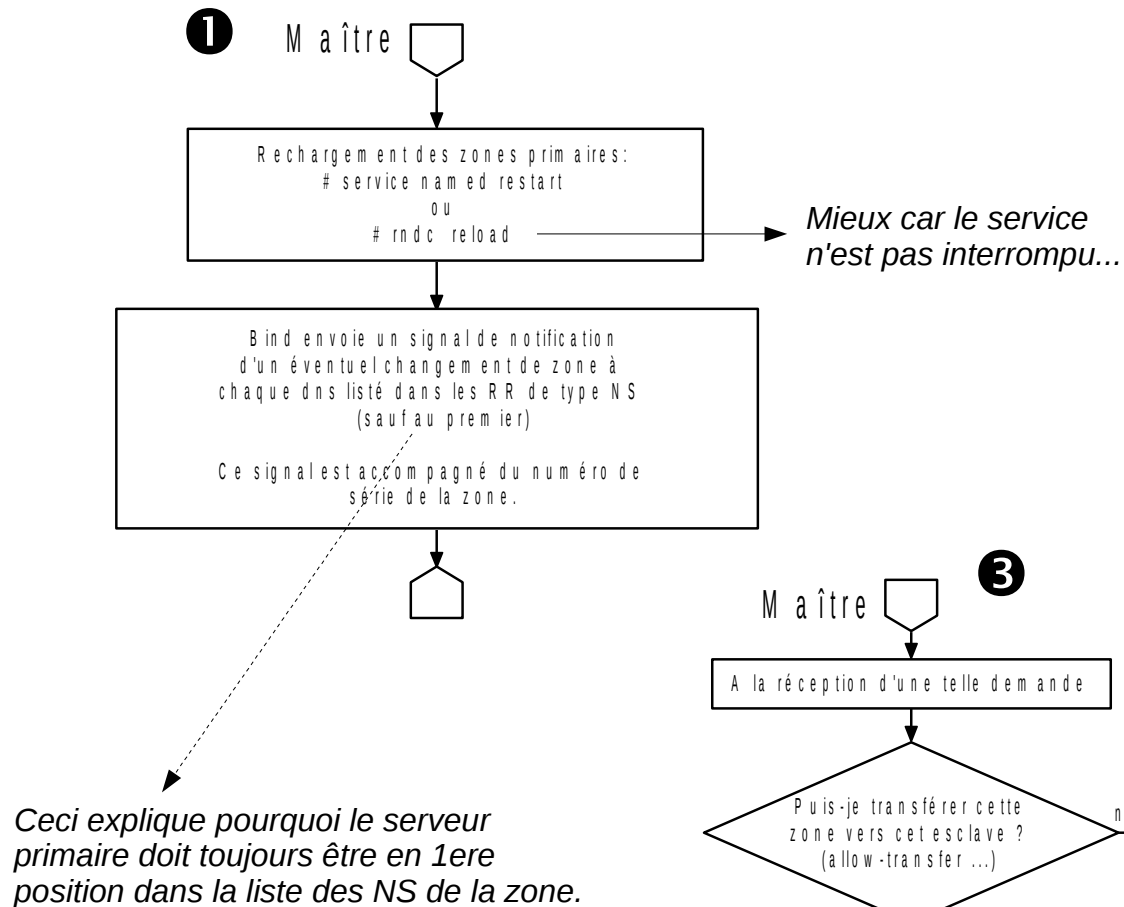
```
zone "isat.net" IN {  
    type slave;  
    masters { 192.168.8.2; } ;  
    file "slaves/db.isat.net";  
}; // zone secondaire isat.net
```

```
zone "8.168.192.in-addr.arpa" IN {  
    type slave;  
    masters { 192.168.8.2; } ;  
    file "slaves/db.isat.net-rev";  
}; // zone secondaire du reverse isat.net
```



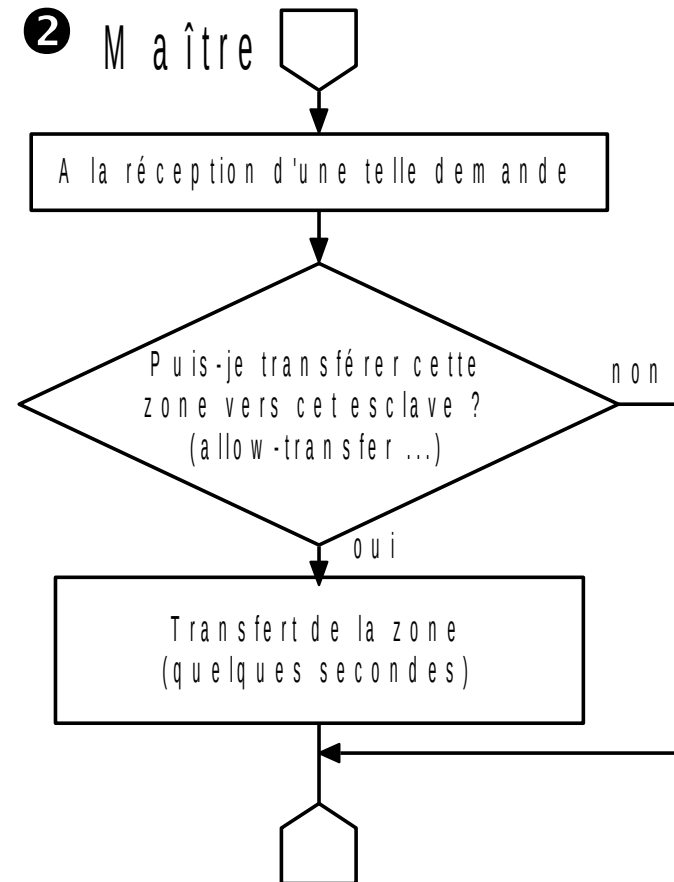
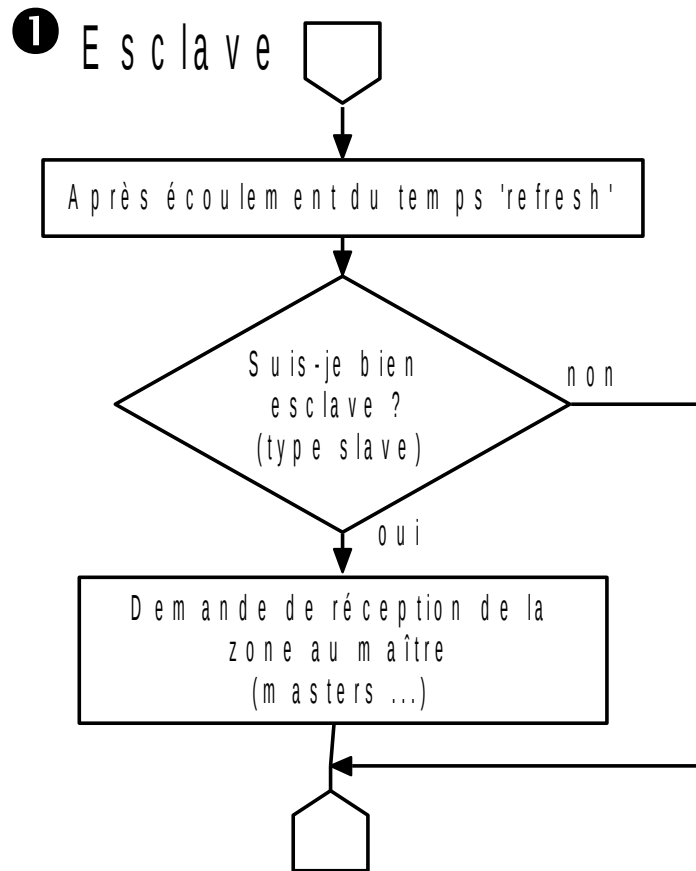
Redondance

Synchronisation (1): Le maître notifie



Redondance

Synchronisation (2): Le maître ne notifie pas



- ✍ Ici, et par souci de clarté, les temps 'retry' et 'expire' sont volontairement écartés de l'explication ...
- ✍ Cette technique de synchronisation est de moins en moins utilisée car la synchronisation est trop tardive par rapport à la technique par 'notification'.

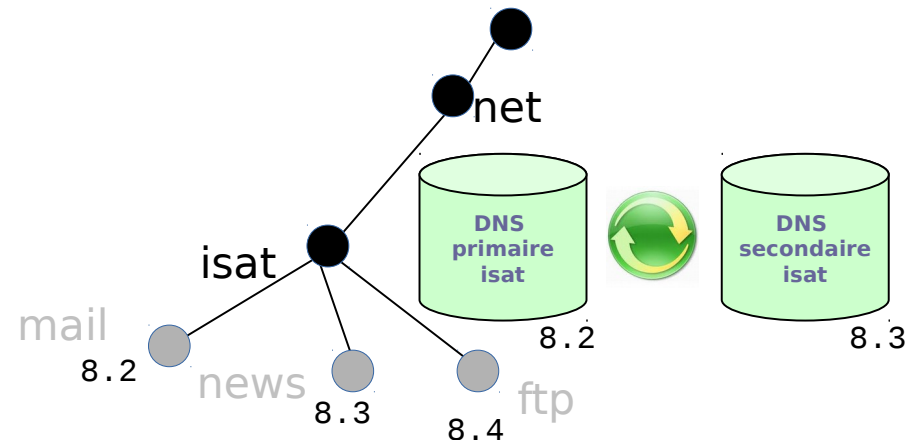
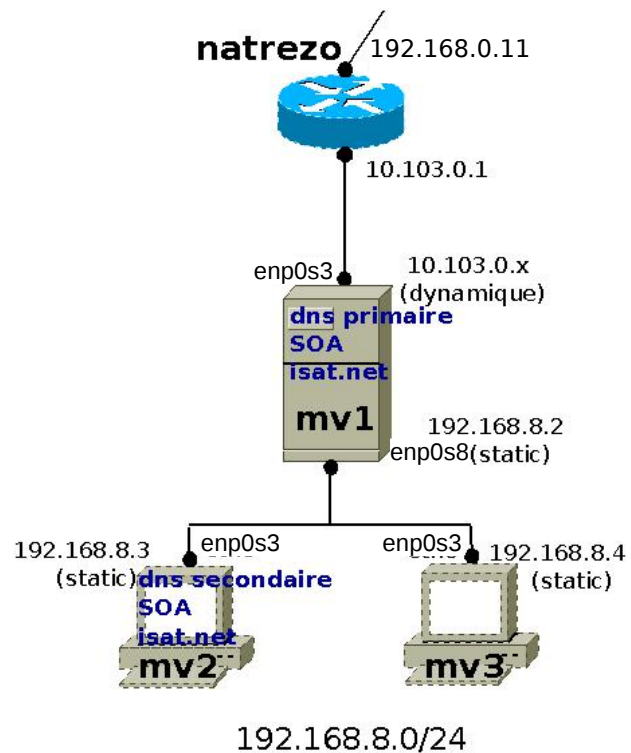


Exercice: Redondance

Exercice 5

- Construire les serveurs dns primaires et secondaires de 'isat.net':

Plan DNS et architecture:



Les services

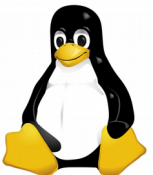
mail.isat.net sur 192.168.8.2
news.isat.net sur 192.168.8.3
ftp.isat.net sur 192.168.8.4

Alias sur les noms canoniques

r2d2.isat.net sur 192.168.8.2
yoda.isat.net sur 192.168.8.3
lea.isat.net sur 192.168.8.4

Les DNS

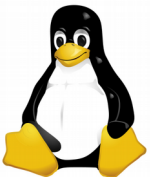
Primaire du domaine "isat.net" sur 192.168.8.2
Secondaire du domaine "isat.net" sur 192.168.8.3



Exercice: Redondance

Exercice 5 (suite)

- Lancez un shell sur MV1 et MV2 pour surveiller l'évolution des logs.
Sur un autre shell, démarrez bind sur le primaire puis sur le secondaire.
Que constatez-vous ?
- Incrémentez le numéro de série et ajoutez un RR factice dans les 2 zones du primaire.
Rechargez les zones du primaire.
Que constatez-vous ?
- Sur MV3, déclarez MV1 et MV2 comme dns à contacter pour résoudre des noms et tentez une résolution de noms.
Stoppez bind sur MV1.
Retentez une résolution de noms à partir de MV3.
Que constatez-vous ?



Références

WEBOGRAPHIE

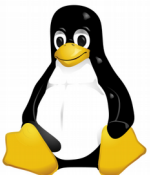
<http://www.afnic.fr//ext/dns/index.html> (autoformation)
<https://www.afnic.fr/ext/dns/html/cours239.html> à [cours248.html](#)
<http://www.certa.ssi.gouv.fr>
https://www.reseaucerta.org/sites/default/files/ccDNS_v1.2.pdf
<http://www.zytrax.com/books/dns>
<http://zero202.free.fr/cs61-dns/html/ar01s03.html>

BIBLIOGRAPHIE

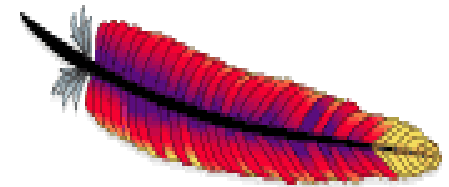
LINUX - Préparation à la certification LPIC-1 (LPI 101 LPI 102) [2e édition]
Par Sébastien ROHAUT – ENI Editions

CentOS Bible
By Timothy Boronczyk and Christopher Negus
Edition 2009 - Wiley Publishing, Inc.

ADMINISTRATION UNIX: ASPECTS RESEAUX
Par Xavier Bogaert - Technofutur3



Serveur Web Apache



Jean-Louis Gouwy



Plan

- Introduction (Historique / Caractéristiques)
- Le modèle client-serveur (Les échanges)
- Fonctionnalités
- Architecture (Vue d'ensemble / Noyau / Modules / Filtres)
- Installation (Par les sources / Par les binaires / Architecture sous Fedora 26)
- Configuration (Structure de l'httpd.conf / Structure de l'httpd.conf sous Fedora 26 /
Le contexte des directives)
- Environnement principal (ServerName / ServerRoot / DocumentRoot / ServerAdmin /
ServerTokens/ Listen / ErrorDocument)
- Exercice 1
- Contrôler Apache (L'arbre des processus / Création des instances de httpd
Les directives: MinSpareServers - MaxSpareServers - StartServers -
MaxRequestWorkers - ServerLimit - User - Group)
- Les sites perso
- Les redirections simples
- Les index de répertoires
- Exercice 2



Plan

- Hébergement virtuel (Introduction / Principe / Explication par l'exemple / Remarques)
- Exercice 3
- Les fichiers journaux (Le journal des erreurs / Le journal des accès / Les hôtes virtuels)
- Exercice 4
- Références



Introduction

• HISTORIQUE

- 1965: Naissance en Suisse de l'idée originale de l'hypertexte (réseau d'un ensemble de documents informatiques liés entre eux) puis de l'httpd (http daemon).
- 1992: 26 serveurs Web.
- 1995: Démarrage du projet Apache (version 1.0).
- 1996: Apache devient le serveur Web le plus répandu.
- 1999: Constitution de l'ASF (Apache Software Foundation).
<http://www.apache.org> (ASF)
<http://httpd.apache.org> (serveur Apache)

Le serveur Apache tient son nom d'une des plus fières tribus indiennes dont la vigueur et la faculté d'adaptation n'était plus à prouver.



Introduction

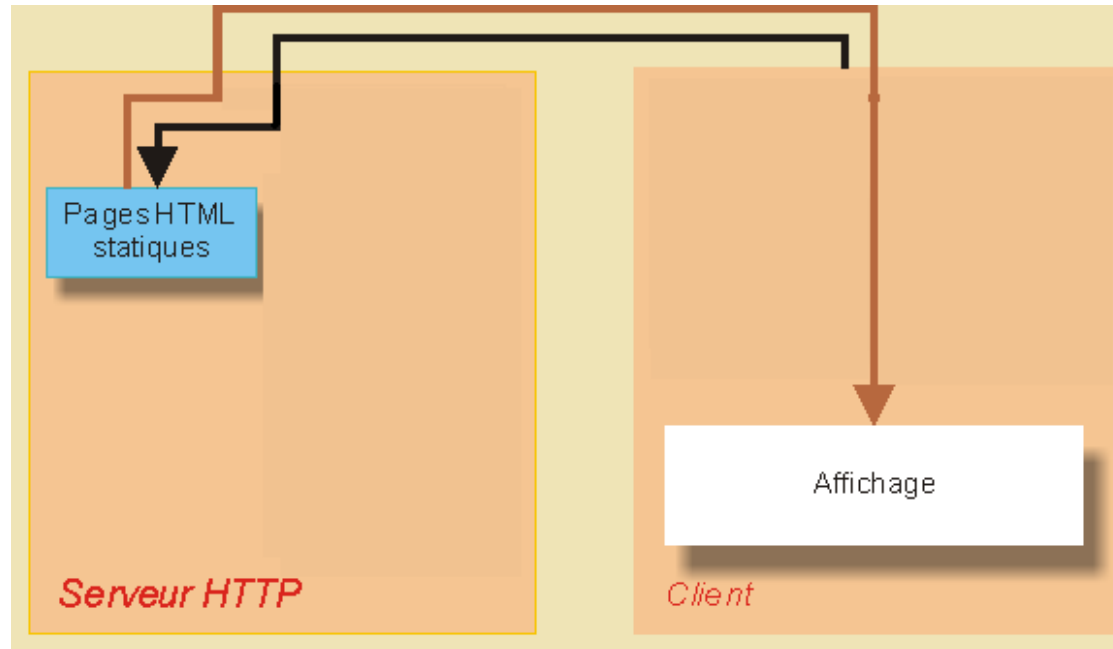
• CARACTERISTIQUES

- Open Source et gratuité.
- Multi-plateformes (Unix- Unix Like - Linux - Windows - MacOS).
- C'est le serveur web 'opensource' le plus répandu.
Source: *<http://www.netcraft.com>*
- Modulable, fiable, performant, sécurisé et extensible.
- Supporte les protocoles: HTTP / HTTPS (le plus souvent), POP3, FTP ...



Le modèle client-serveur

- **LES ECHANGES: Demande d'une page html 'statique'**



Ici, tout le contenu est défini dans la page HTML demandée.
Le serveur l'envoie tel quel au client qui n'a plus qu'à l'interpréter et l'afficher.



HELHa
Haute École Louvain en Hainaut

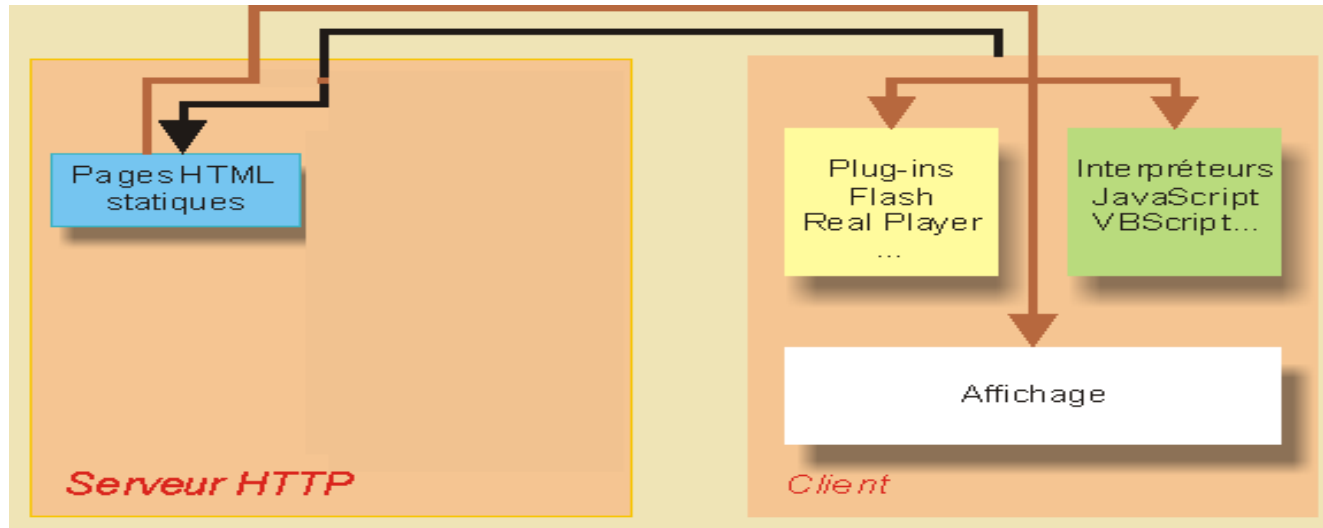
-
- The diagram illustrates the architecture of a web server and client interaction. It is divided into two main sections: **Serveur HTTP** (HTTP Server) on the left and **Client** on the right.
- Serveur HTTP:**
- Contains a box labeled **Générateurs de pages dynamiques PHP, ASP, scripts CGI...** (Dynamic page generators).
 - Below it is a box labeled **Base de données** (Database).
 - Arrows indicate a bidirectional flow between the generators and the database.
 - A warning icon (a sad face) is placed next to the text **Surcharge possible du serveur** (Server overload possible), indicating a potential bottleneck or performance issue.
- Client:**
- Contains a box labeled **Affichage** (Display).
- Connections:**
- A thick red arrow points from the **Générateurs de pages dynamiques** box to the **Affichage** box, representing the delivery of content to the client.
 - A thin black line connects the top of the **Client** section back to the top of the **Serveur HTTP** section, representing the return path for requests.

- Demande d'un résultat calculé par le serveur et dont les données sont fournies par le client.
- Demande de données qui se trouvent dans une base de données hébergée côté serveur.



Le modèle client-serveur

- **LES ECHANGES: Demande d'une page html 'dynamique' (client side)**



- ☺ Pas de surcharge sur le serveur.
Pages animées.
- ☹ Incompatibilité des navigateurs.
Failles de sécurité possibles côté client.

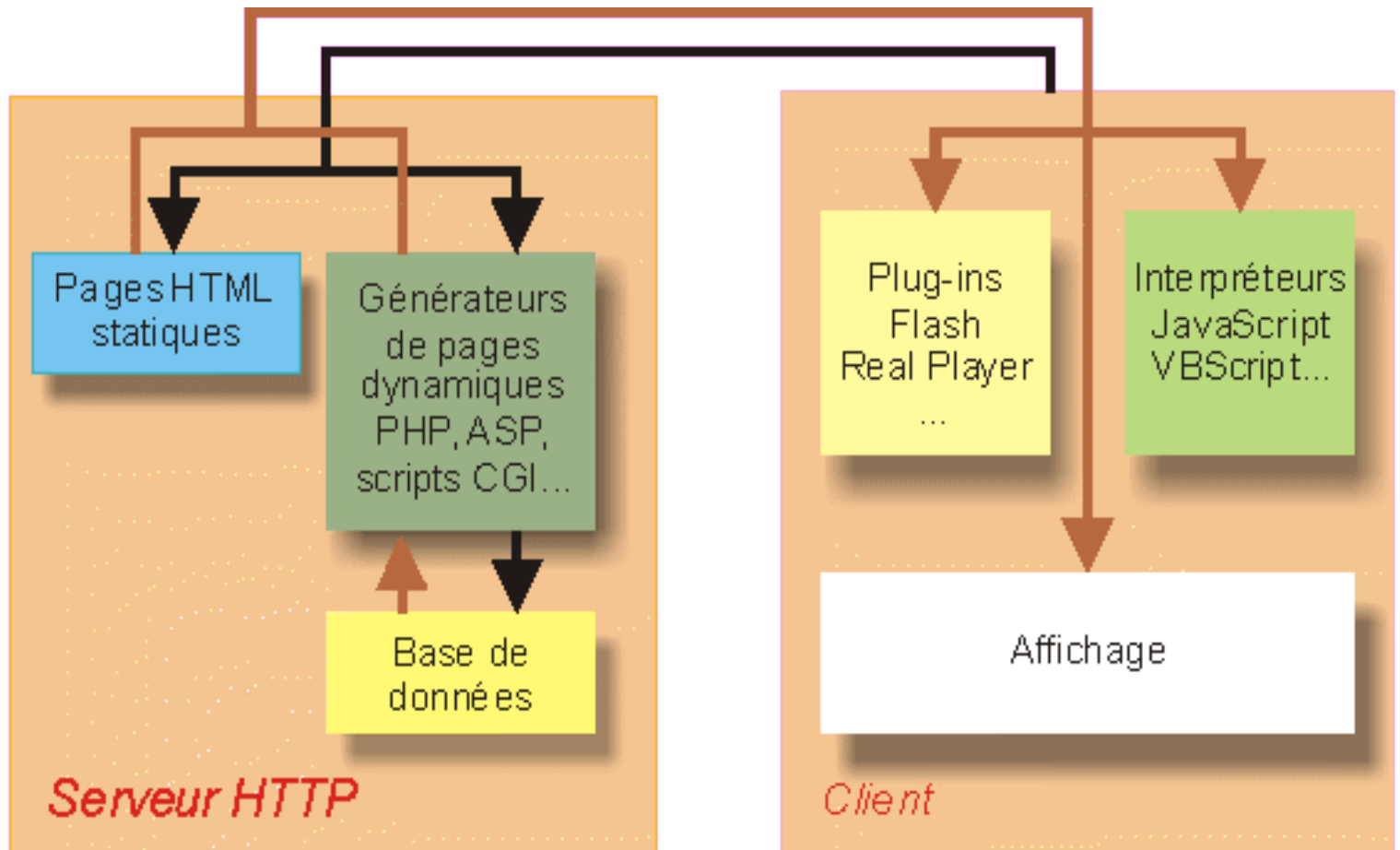
Exemples:

- Contrôler la validité de données avant de les envoyer au serveur.
- Effectuer un traitement local pour afficher un résultat.
- Animations quelconques.



Le modèle client-serveur

- **LES ECHANGES: Modèle complet**



Serveur HTTP

Client



- **Conformité aux standards:**
Entièrement conforme aux standards HTTP/1.1 – RFC 2616).
- **Scalabilité:**
Permet d'héberger un grand nombre de sites web sur une même machine sans voir les performances diminuer de manière cruciale.
- **Objets dynamiques partagés (shared objects):**
Modules pouvant être compilés séparément du noyau et activés au démarrage d'Apache.
- **Personnalisation:**
Programmation (en C ou en Perl) de modules personnels via l'API d'Apache.
- **Programmation:**
Permet la programmation côté serveur (PHP, Perl, servlets Java, Java Server Page, Active Server Pages, CGI, FastCGI, Server-Side Includes)



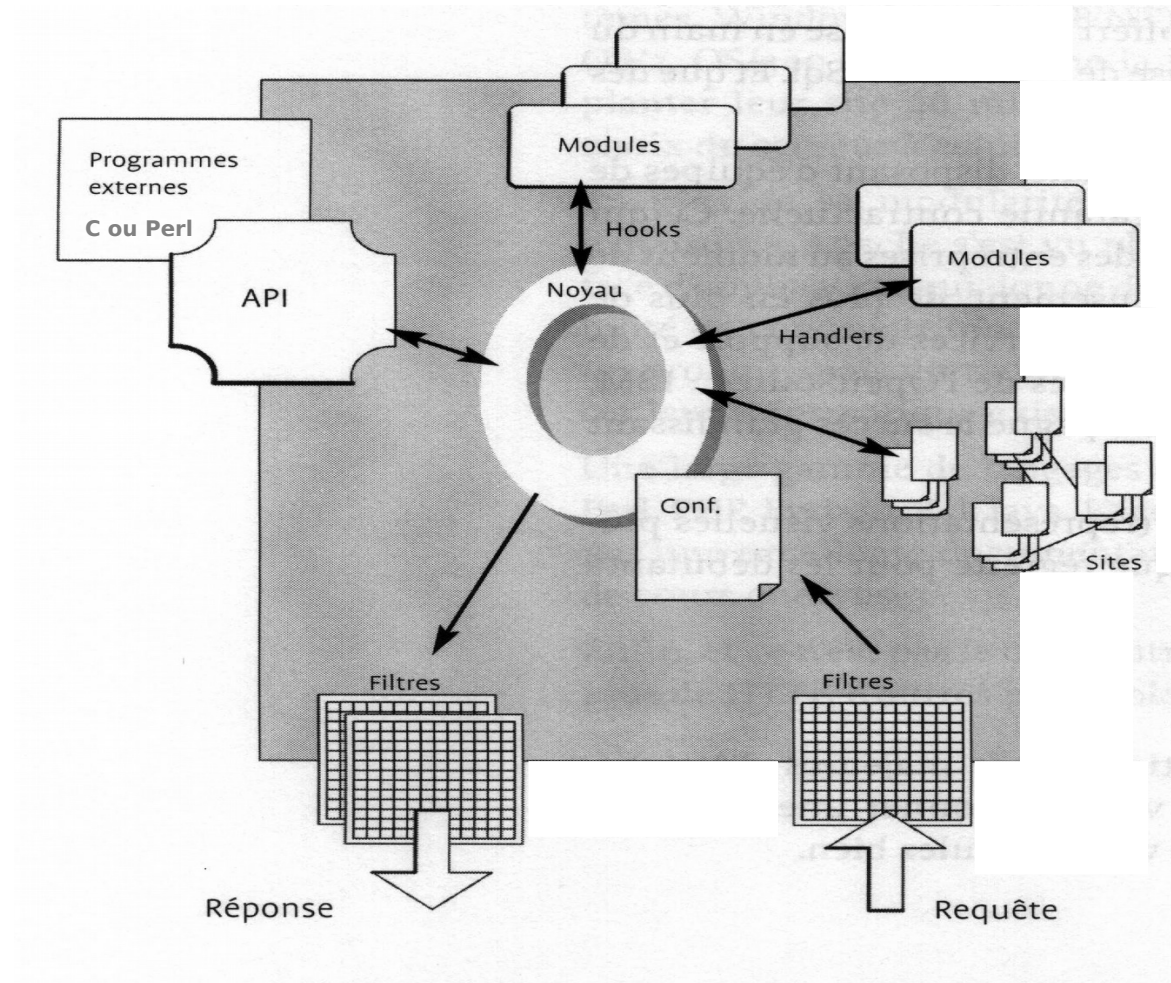
- **Serveur mandataire:**
Apache peut être configuré en proxy serveur à l'aide de son module `mod_proxy`.
- **Sécurité:**
Par différentes méthodes d'authentification de l'utilisateur (fichier plats ou bases de données dbm-mysql-...)

Par l'autorisation du chiffrement sur Internet via l'échange de certificats numériques (SSL)



Architecture

- **VUE D'ENSEMBLE**



- **Le noyau**

- Contient toutes les fonctionnalités de base du serveur (ex. directive Listen).
- Il distribue le travail aux modules ou aux programmes externes.
- Les documents sont extraits du ou des sites dont il a la charge.
- Il reçoit et comprend une requête.
- Il prépare et envoie une réponse.
- Il connaît ses ressources disponibles (modules disponibles, liste et localisation des sites à gérer ...) grâce à son fichier de configuration (httpd.conf).
- Pour chaque module qu'il va solliciter, il dispose d'un jeu de directives.



Architecture

- **Les modules**

- Ils étendent les fonctionnalités du noyau.

- Ils peuvent être :

- . Standards: Maintenus par l'ASF (ex. `mod_auth_basic`)
Font partie de la distribution d'Apache
(<http://httpd.apache.org/docs/2.4/fr/mod>)

- . Tiers: Maintenus par des tiers (ex. `mod_auth_oracle`)
Disponibles sur la toile.

- Ils peuvent être installés:

- . Lors de la compilation (ou de l'installation) de la distribution d'Apache
(modules standards)

- . Lors d'une compilation séparée (modules tiers) via l'utilitaire *apxs*
`# path_to/apxs -cia module.c` (*-cia : compile, installe et active*)



- **Les modules (suite)**

- Le choix des modules à utiliser se fait dans le fichier de configuration principal. Deux directives sont disponibles pour gérer les modules :

`LoadModule` : qui permet de charger un module au démarrage

`<IfModule>` : qui permet d'activer certaines parties du fichier de configuration si un module a été chargé.

- `httpd -l` Pour voir la liste des modules compilés dans le cœur d'Apache. Ce n'est pas la liste des modules chargés dynamiquement via la directive `LoadModule`.

Compiled in modules:

```
core.c  
http_core.c  
mod_so.c
```

Doivent toujours être présents.

Permet le bon fonctionnement de la directive `LoadModule` (chargement de modules de type `Shared Object`)

`httpd -L | grep -i userdir`

Pour connaître le module correspondant à chaque directive (ici la directive `'UserDir'`).



- **Les filtres**

Les filtres sont des modules standards ou tiers que vous choisissez d'incorporer ou non à Apache.

- . soit à l'entrée, sur le chemin des demandes entrantes
- . soit en sortie, sur le chemin des réponses sortantes

Par exemple un module de compression des données qui:

- . en entrée décompresse des requêtes compressées par un browser
- . en sortie compresse l'envoi de tous les documents du site



Installation

- **Par les sources**

- Hors cadre du cours.
- Plus d'info: <http://httpd.apache.org/docs/2.4/fr/install.html>



- **Par les binaires**

- Via la commande d'installation automatique de la distribution.
 - Ce sera la retenue pour la suite du cours malgré que cette installation est moins souple que par compilation des sources (ciblage des modules plus difficile)...
- ... par contre, elle offre un outil de désinstallation.

CentOS 6 : `yum install httpd -y` (Apache version 2.2)

CentOS 7 : `yum install httpd -y` (Apache version 2.4.6)

Fedora 26 Server avec Apache embarqué.
(Apache version 2.4.25) :

Nous retiendrons cette distribution car :

- la version d'Apache offerte est de la dernière branche 2.x
- de plus, le protocole HTTP 2 est correctement supportée à partir d'Apache 2.4.23.



Installation

- **Architecture sous Fedora 26**

/etc/httpd/ → dossier contenant l'ensemble des fichiers de configuration.

└─ **conf/httpd.conf** → fichier principal de configuration.

└─ **conf.d** → dossier contenant les fichiers secondaires de configuration.

└─ **conf.modules.d** → dossier contenant les fichiers de lancement des modules.

/var/www/ → dossier contenant les données du site par défaut.

└─ **cgi-bin**: dossier (vide) contenant les scripts.

└─ **html**: dossier (vide) contenant les pages du site par défaut.

/var/log/httpd/ → dossier contenant les journaux

└─ **access_log** → journal des accès aux pages traitées par le serveur.

└─ **error_log** → journal des erreurs.



Installation

- **Architecture sous Fedora 26**

/usr/share/httpd

- error → dossier contenant les pages affichées en cas d'erreur
- icons → dossier contenant quelques icônes
- manual → dossier contenant la documentation

/usr/lib64/httpd/modules → dossier contenant le binaire des modules (.so)

/usr/sbin/httpd → le daemon Apache

Remarques:

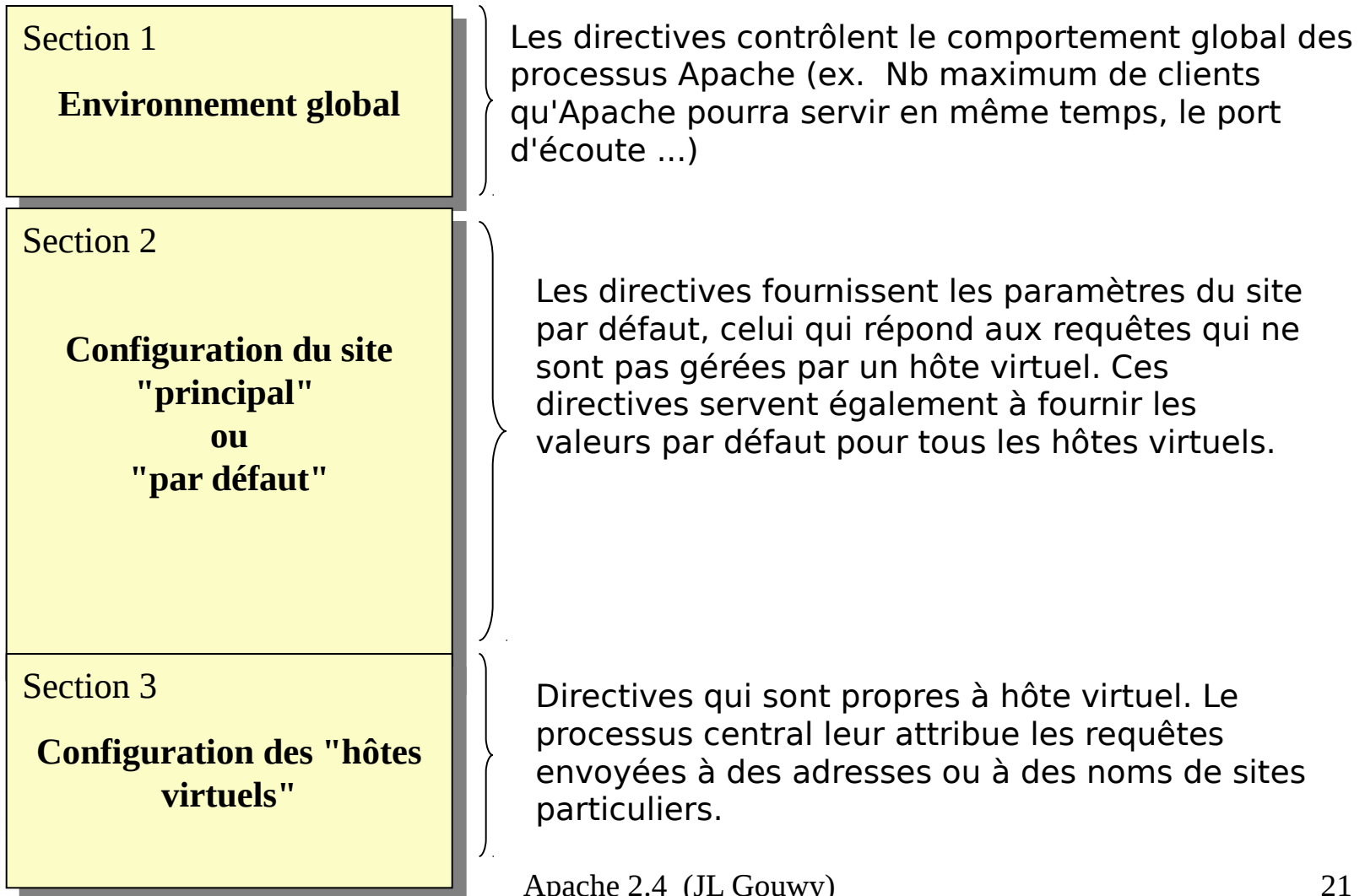
```
# systemctl start httpd.service  
→ lancement d'Apache (port d'écoute par défaut: 80)
```

```
# systemctl enable httpd.service  
→ sera lancé au démarrage du système
```



Configuration

- **Structure de l'httpd.conf**



Configuration

- **Structure de l'httpd.conf sous Fedora 26**

```
...  
...  
Include conf.modules.d/*.conf ●  
...  
# 'Main' server configuration  
...  
...  
IncludeOptional conf.d/*.conf ●
```

Chargés par ordre
alphabétique

Il est préférable de ne pas le modifier.

On préférera la création d'un fichier spécifique à nos besoins dans le dossier /etc/httpd/conf.d.

Cela permet de ne pas bloquer les mises à jour éventuelles du fichier principal et simplifiera grandement les migrations (il suffit de récupérer notre fichier de configuration).



Configuration

- **Le contexte des directives**

L'aide officielle d'Apache classe les directives dans 4 catégories

<http://httpd.apache.org/docs/2.4/fr/mod/quickreference.html>

s	server config
v	virtual host
d	directory
h	.htaccess



Configuration

- **Le contexte des directives (suite)**

- Contexte général du serveur (server config - s)

Agit sur tout le serveur.

ex.	StartServers	(uniquement dans ce contexte)
	ServerName	(dans ce contexte mais peut être redéfinie pour chaque hôte virtuel)

- Contexte hôte virtuel (virtualhost - v)

De nombreuses directives d'hôte virtuel (définies dans le conteneur `<VirtualHost>`) surchargent celles qui sont générales au serveur.

ex.	ServerName
	DocumentRoot



Configuration

- **Le contexte des directives (suite)**

- Contexte conteneur (directory - d)

Inclut les directives qui ne peuvent s'appliquer que dans un des 3 conteneurs (<Directory>, <Files> et <Location>) et dont la portée est limitée à ce conteneur.

ex. Require

- Contexte .htaccess (.htaccess - h)

Sont traitées comme celles d'un conteneur <Directory> de l'httpd.conf. La principale différence est que les directives d'un fichier htaccess peuvent être désactivées au moyen de la directive AllowOverride dans l'httpd.conf.



L'environnement principal

- **La directive ServerName**

Apache doit toujours pouvoir déterminer le nom d'hôte de la machine sur laquelle il tourne car il l'utilise pour créer des URL d'autoréférence.

Exemple: `ServerName www.mysite.be`

- **La directive ServerRoot**

Répertoire dans lequel les fichiers de configuration, les logs et les modules sont gardés. Ce nom de dossier servira à préfixer tout chemin relatif rencontré dans l'`httpd.conf`.

Exemple: `ServerRoot "/etc/httpd"`



L'environnement principal

- **La directive DocumentRoot**

Dossier dans lequel les documents du site par défaut sont déposés.
C'est donc ce dossier qui contient les fichiers qu'Apache fournit lorsqu'il reçoit des requêtes avec l'URL /.

Exemple: `DocumentRoot "/var/www/html"`

☞ Le changement de nom du DocumentRoot doit aussi être effectué dans Son conteneur `<Directory ...>`, qui regroupe toutes les directives s'appliquant à DocumentRoot et ses sous-répertoires.

- **La directive ServerAdmin**

Adresse mail du webmaster qui pourrait s'afficher sur certains documents construits par le serveur et renvoyés au client en cas d'incidents.
Valable si la directive **ServerSignature** est à l'état on permettant alors l'ajout de cette information en bas de page.

Exemple: `ServerAdmin webmaster@mysite.be`



L'environnement principal

- **La directive ServerTokens**

Permet de contrôler le contenu de l'en-tête Server inclus dans la réponse envoyée au client.

Exemples:

ServerTokens Prod[uctOnly]

➔ Le serveur renvoie (par ex.): Server: Apache

ServerTokens Major

➔ Le serveur renvoie (par ex.): Server: Apache/2

ServerTokens Minor

➔ Le serveur renvoie (par ex.): Server: Apache/2.0

ServerTokens Min[imal]

➔ Le serveur renvoie (par ex.): Server: Apache/2.0.41

ServerTokens OS

➔ Le serveur renvoie (par ex.): Server: Apache/2.0.41 (Unix)

ServerTokens Full (valeur par défaut)

➔ Le serveur renvoie (par ex.): Server: Apache/2.0.41 (Unix) PHP/4.2.2



L'environnement principal

- **La directive Listen**

Pour définir les adresses IP et les numéros de ports sur lesquels Apache attend et reçoit les connexions des clients.

Exemples:

`Listen 80` → Apache écoute sur toutes les interfaces sur le port 80.

`Listen 10.0.0.7:80` → Apache écoute sur le port 80 sur l'interface d'IP 10.0.0.7.

`Listen 80` → Apache écoute sur le port 80 et 8080 sur toutes les interfaces.
`Listen 8080`

`Listen 192.168.1.1:80` → Apache répond aux requêtes http sur l'interface interne
`Listen 216.180.25.168:443` et aux requêtes https (connexions SSL) sur l'interface publique.



L'environnement principal

- **La directive ErrorDocument**

Pour remplacer les pages d'erreur standards envoyées au client en cas de problème.

Exemples:

```
ErrorDocument 403 "Vous n'êtes pas autorisé à lire cette page !"
```

Ici on affiche simplement un texte adapté à l'erreur.

```
ErrorDocument 401 /missing.html
```

Ici on affiche une page html sensée se trouver à la racine du site web.

```
ErrorDocument 500 http://www.bidon.com/erreur.html
```

Ici on affiche une page html extérieure au site.



ErrorDocument 401 nécessite toujours une URL interne.



Exercice 1

• Un serveur simple

[Préfixez tous les fichiers d'extension .conf du dossier /etc/httpd/conf.d par '01-' et créez un fichier 00-1main.conf et 00-0server.conf]

a) Configurez le serveur pour:

- . qu'il écoute sur le port 80 sur toutes les interfaces
- . qu'il présente une page d'accueil index.html lors d'une requête vers l'URL de ce site (inventez son contenu)
- . qu'il affiche une page html personnalisée en cas d'erreur 404

b) Vérifiez la configuration du serveur.

c) Testez votre serveur pour vérifier si la page d'accueil est bien offerte:

- . à l'aide de l'utilitaire telnet
- . à l'aide d'un navigateur quelconque

d) Reconfigurez Apache pour qu'il écoute cette fois sur le port 8080:

e) Testez votre serveur:

- . à l'aide de l'utilitaire telnet
- . à l'aide d'un navigateur quelconque

f) Compilez, installez et testez un module tiers (mod_pony)



Contrôler Apache

- **L'arbre des processus**

Apache lance plusieurs daemons en parallèle; ceux-ci se trouvent en permanence à l'écoute du réseau afin de pouvoir répondre rapidement à un grand nombre de requêtes simultanées.

```
# service httpd restart
# ps -ef
```

```
...
root      2116  1      S      0:00 /usr/sbin/httpd
apache    2119  2116  S      0:00 /usr/sbin/httpd
apache    2120  2116  S      0:00 /usr/sbin/httpd
apache    2121  2116  S      0:00 /usr/sbin/httpd
apache    2122  2116  S      0:00 /usr/sbin/httpd
apache    2123  2116  S      0:00 /usr/sbin/httpd
apache    2124  2116  S      0:00 /usr/sbin/httpd
apache    2125  2116  S      0:00 /usr/sbin/httpd
apache    2126  2116  S      0:00 /usr/sbin/httpd
```

*Processus principal.
C'est lui qui reçoit les requêtes
et les distribue à ses fils.*

*Maximum
256 instances (fils).*



Contrôler Apache

- **Création des instances de httpd**

Plus d'info.: <http://httpd.apache.org/docs/2.4/fr/misc/perf-tuning.html>

Les instances sont créées en fonction du type de module Multi-Processus (MPM) chargé :

- **Le MPM prefork:** chaque processus enfant possède un seul thread et chaque processus gère une seule connexion à la fois.
 - 😊 Aussi rapide qu'en 'worker'.
Stable et universel (utilisable avec les modules tiers qui ne supportent pas le threading et compatible avec des logiciels ou OS anciens).
 - 😞 Plus gourmand en mémoire qu'en 'worker'.
- **Le MPM worker:** chaque processus enfant possède plusieurs threads et chaque thread gère une seule connexion à la fois.
 - 😊 Moins gourmand en mémoire qu'en 'prefork'.
 - 😞 Pas universel
- **le MPM event** utilise les threads, mais il a été conçu pour traiter davantage de requêtes simultanément.
 - 😊 Moins gourmand en mémoire qu'en 'prefork' et plus rapide qu'en 'worker'
 - 😞 Pas universel



Contrôler Apache

- Les directives **MinSpareServers** / **MaxSpareServers** / **StartServers** / **MaxRequestWorkers** / **ServerLimit**

MinSpareServers: Nombre minimal d'instances de serveurs.

MaxSpareServers: Nombre maximum d'instances de serveurs.

StartServers: Nombre de serveurs supplémentaires créés au démarrage d'Apache.

MaxRequestWorkers / ServerLimit:

Limite le nombre de processus qui peuvent tourner simultanément (chaque connexion cliente en utilise un).

Exemple: Voir exercice 2



Contrôler Apache

- **Les directives User et Group**

Utilisateur et groupe Linux sous lequel s'exécuteront les processus fils d'Apache chargés de répondre aux requêtes des clients.
Le processus maître doit être lancé sous le compte root pour pouvoir changer le user et le group de ses processus fils.

Exemples:

User apache
Group apache



Les sites perso

- Il est possible de permettre aux utilisateurs disposant d'un compte sur le serveur de posséder leur propre site.
- Pratique très courante parmi les FAI qui proposent l'hébergement de pages web de leurs clients.
- Cette fonctionnalité est fournie par le module standard `mod_userdir`.

```
LoadModule userdir_module ...
```

- Elle est activée par la directive `UserDir`.

```
<IfModule mod_userdir.c>  
    #UserDir disabled  
    UserDir public_html  
</IfModule>
```

Si le module 'userdir' est chargé, alors un utilisateur du système pourra y héberger son site Web dont la racine se trouvera dans le dossier 'public_html' de sa home directory.

- Le site sera accessible via l'URL: `http://ip_serveur/~login_utilisateur`
Exemple: `http://www.mysite.be/~jean`



Les sites perso

- Autres formes de la directive UserDir :

```
UserDir disabled <user1 user2 ...>
```

Désactive la gestion des sites perso pour la liste des utilisateurs indiquée.

```
UserDir disabled
```

Désactive la gestion de tous les sites perso.
Souvent utilisée avant une directive UserDir enabled.

```
UserDir enabled <user1 user2 ...>
```

Active la gestion des sites perso pour la liste des utilisateurs indiquée.

<u>Exemple:</u>	UserDir disabled
	UserDir enabled jean louis



Les redirections simples

- La directive standard `Alias` permet d'accéder facilement à des documents HTML en dehors de l'arborescence `DocumentRoot`.
- Cette fonctionnalité est fournie par le module standard `mod_alias`.

```
LoadModule alias_module ...
```

- Soit accéder à la page HTML `/usr/share/doc/HTML/index.html` via le site `www.mysite.be` par redirection:

```
Alias /CentOS "/usr/share/doc/HTML/"
```

```
<Directory "/usr/share/doc/HTML">  
    Require all granted  
</ Directory>
```

*Car la cible est dans un dossier situé en dehors de l'arborescence du site web
→ permettre explicitement l'accès à ce dossier.*

Cette page sera accessible via l'URL:

```
http://www.mysite.be/CentOS
```



Les index de répertoires

- **Recherche d'une page d'accueil**

Apache est capable de retrouver la page `index.html` sans pour autant que celle-ci soit indiquée dans l'URL.

Exemple:

`http://www.mysite.be/cours`

équivalent à:

`http://www.mysite.be/cours/index.html`



Les index de répertoires

- **Recherche d'une page d'accueil (suite)**

- C'est le module standard `mod_dir` qui recherche et fournit une page (`index.html` par défaut) susceptible de se trouver dans le répertoire indiqué dans l'URL.
- Si la directive `DirectoryIndex` est présente, elle permettra d'ajouter des références à d'autres pages.

```
DirectoryIndex index.html index.htm index.php
```



Les index de répertoires

- **Recherche d'une page d'accueil (suite)**

- Si aucune page d'accueil n'est trouvée, le module `mod_autoindex` (piloté par sa directive `IndexOptions`) crée un index des fichiers du répertoire concerné.

Exemples:

`IndexOptions None` (indexation classique)

`IndexOptions FancyIndexing` (indexation au look plus agréable)

`IndexOptions FancyIndexing VersionSort`

(idem + tri des entrées contenant des numéros de versions)

- Pour supprimer l'indexation sur un dossier particulier

```
<Directory directory_name>
```

```
    Options -Indexes
```

```
</Directory>
```



Exercice 2

- a) Reprendre l'exercice 1 et reconfigurez le serveur pour qu'il écoute sur le port 80 sur votre interface.
- b) Quel est le Module Multi-Processus (MPM) utilisé actuellement ?
- c) Combien de processus enfants y a-t-il actuellement ?
- d) Configurez-le pour:
 - qu'il lance 7 processus enfant en mode 'prefork' au démarrage et qu'il s'assure qu'il en reste toujours au moins 3 en réserve. En cas de montée en charge, 20 processus enfants maximum seront créés pour satisfaire les requêtes.Lorsque le serveur aura satisfait toutes les requêtes de cette montée, il se stabilisera avec 10 processus enfants.

Testez tout cela, à l'aide d'outils adéquats.



Exercice 2 (Suite)

- que la requête `http://www.mysite.be/pamauthor` présente la page `/usr/share/doc/pam/html/sag-author.html`
- qu'il donne la possibilité aux utilisateurs jean, louis et bernard de disposer d'un site perso dont la page d'accueil puisse être `index.html` ou `accueil.html` (inventez leur contenu).
Le site perso de bernard ne sera pas accessible pour le moment.
- qu'un index des entrées du dossier 'cours' soit présenté lors de la requête `http://www.mysite.be/cours`

e) Testez vos configurations.



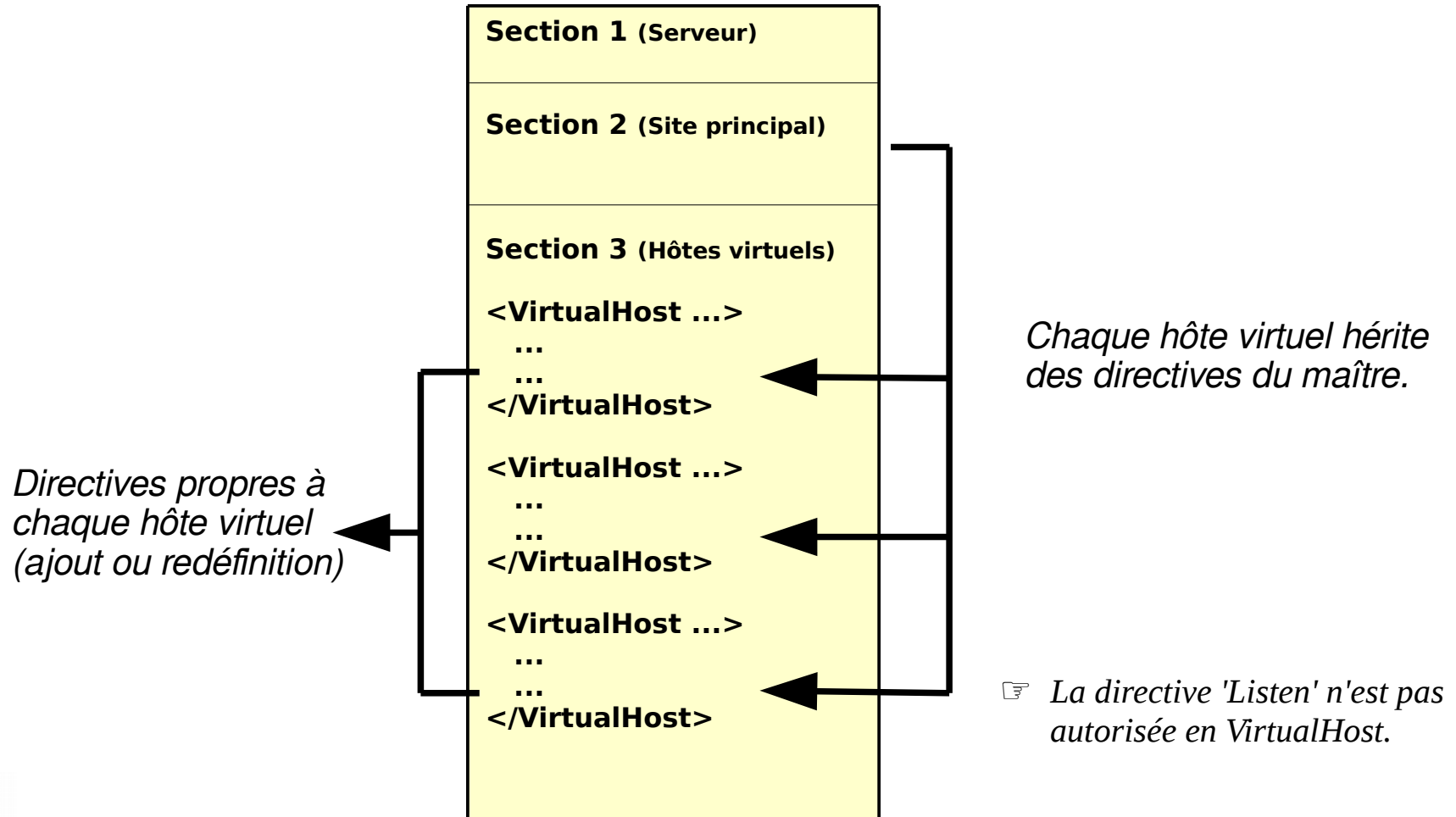
- **Introduction**

- Un seul service httpd pour gérer plusieurs sites web.
- Chaque Ip du serveur peut être dédiée à la gestion d'un ou plusieurs sites web (HTTP/1.1).



Hébergement virtuel

- Principe



- Explication par l'exemple

httpd.conf

Section 2

```
ServerName www.mysite.be
DocumentRoot /var/www/html
...
```

Section 3

```
<VirtualHost 192.168.0.2>
  ServerName vh2.mysite.be
  DocumentRoot /var/www/vh2
</VirtualHost>
```

```
<VirtualHost 192.168.0.3>
  ServerName vh3.mysite.be
  DocumentRoot /var/www/vh3
</VirtualHost>
```

↓

```
<VirtualHost 192.168.0.1>
  ServerName www.mysite.be
  DocumentRoot /var/www/html
</VirtualHost>
```

```
<VirtualHost 192.168.0.1>
  ServerName jean.mysite.be
  DocumentRoot /var/www/jean
</VirtualHost>
```

```
<VirtualHost 192.168.0.1>
  ServerName louis.mysite.be
  DocumentRoot /var/www/louis
</VirtualHost>
```

```
<VirtualHost 192.168.0.1>
  ServerName gouwy.mysite.be
  DocumentRoot /var/www/gouwy
</VirtualHost>
```



- **Explication par l'exemple (suite)**


A la lecture de l'httpd.conf, Apache crée une table contenant la liste des hôtes virtuels (servername) déclarés pour chaque Ip.

192.168.0.1	www.mysite.be	jean.mysite.be	<u>louis.mysite.be</u>	<u>gouwy.mysite.be</u>
192.168.0.2	vh2.mysite.be			
192.168.0.3	vh3.mysite.be			



- **Explication par l'exemple (suite)**

Lorsque Apache reçoit une requête sur une Ip se trouvant dans la table, il recherche le ServerName qui correspond à l'en-tête Host de la requête pour cette Ip...


				
	①	②	③	④
192.168.0.1	<u>www.mysite.be</u>	<u>jean.mysite.be</u>	<u>louis.mysite.be</u>	<u>gouwy.mysite.be</u>
192.168.0.2	⑤ <u>vh2.mysite.be</u>			
192.168.0.3	⑥ <u>vh3.mysite.be</u>			




URL	Champ Host dans l' <u>http request</u>	Site desservi
<u>http://www.mysite.be</u>	<u>www.mysite.be</u>	①
<u>http://jean.mysite.be</u>	<u>jean.mysite.be</u>	②
<u>http://louis.mysite.be</u>	<u>louis.mysite.be</u>	③
<u>http://gouwy.mysite.be</u>	<u>gouwy.mysite.be</u>	④
<u>http://vh2.mysite.be</u>	<u>vh2.mysite.be</u>	⑤
<u>http://vh3.mysite.be</u>	<u>vh3.mysite.be</u>	⑥



- **Explication par l'exemple (suite)**

Si aucune correspondance n'est trouvée, c'est le premier hôte virtuel de la liste qui sera sélectionné. Ainsi, pour qu'il soit toujours accessible, on indique souvent le site maître en tant que premier de liste dans l'entrée correspondant à son Ip.

				
	①	②	③	④
192.168.0.1	<u>www.mysite.be</u>	<u>jean.mysite.be</u>	<u>louis.mysite.be</u>	<u>gouwy.mysite.be</u>
192.168.0.2	⑤ vh2.mysite.be			
192.168.0.3	⑥ vh3.mysite.be			


URL	Champ Host dans l'http request	Site desservi	
<u>http://192.168.0.1</u>	192.168.0.1	Pour cette <u>Ip</u> , aucune correspondance dans >>> ①	
<u>http://192.168.0.2</u>	192.168.0.2	Pour cette <u>Ip</u> , aucune correspondance dans >>> ⑤	
<u>http://192.168.0.3</u>	192.168.0.3	Pour cette <u>Ip</u> , aucune correspondance dans >>> ⑥	



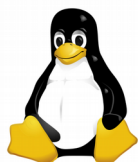
Hébergement virtuel

- **Explication par l'exemple (suite)**

Lorsque Apache reçoit une requête sur une Ip sur laquelle il écoute mais qu'aucun site virtuel n'est attachée à celle-ci, alors c'est le site principal défini en section 2 de l'httpd.conf qui sera desservi.

				
	①	②	③	④
192.168.0.1	<u>www.mysite.be</u>	<u>jean.mysite.be</u>	<u>louis.mysite.be</u>	<u>gouwy.mysite.be</u>
192.168.0.2	⑤ vh2.mysite.be			
192.168.0.3	⑥ vh3.mysite.be			

URL	Champ Host dans l'http request	Site desservi			
<u>http://192.168.0.4</u>	192.168.0.4	Apache écoute sur cette Ip mais aucun site n'y est attaché >>> <u>www.mysite.be</u> (de la section 2 de l'httpd.conf)			



- **Remarques**

- La création d'un alias sur une interface iface se fait par la création d'un fichier :

```
/etc/sysconfig/network-scripts/ifcfg-iface:0
```

```
...  
IPADDR=nouvelle_ip  
NETMASK=nouveau_masque  
ONBOOT=yes  
NAME=iface:0  
DEVICE=iface:0
```

↖
Numéro d'alias

- Pour créer d'autres alias sur cette même interface, il suffit de créer d'autres fichiers en incrémentant le numéro d'alias (...:**1**, ...:**2**, etc) et en adaptant les valeurs des variables NAME et DEVICE en conséquence.



Exercice 3

Configurez Apache pour qu'il puisse en même temps gérer de l'hébergement par ip et par nom.

- 192.168.27.10 permettra d'héberger l'unique site `jean.mysite.be`
- 192.168.27.11 permettra d'héberger l'unique site `louis.mysite.be`
- 192.168.27.2 permettra d'héberger les sites `vh2.mysite.be` et `vh3.mysite.be`. Ce dernier ne sera accessible qu'aux users repris dans `/var/www/securite/pwd`.
- 192.168.27.1 permettra d'héberger le site `vh1.mysite.be`
- Le site maître sera toujours accessible par l'ip 192.168.27.1 ou par `www.mysite.be`. Ce sera aussi le site par défaut si une requête arrive via une interface associée à aucun site.

Remarque: Les fonctionnalités des exercices précédents doivent toujours être opérationnelles.

Relancez Apache et testez sa configuration



Pour véritablement gérer un serveur web, il est nécessaire de disposer d'un retour d'informations à propos de l'activité et des performances du serveur, ainsi que de tout problème qui pourrait survenir.

- **Le journal des erreurs** (Directives ErrorLog & LogLevel)

Directive ErrorLog : Le nom et la localisation du journal.

Exemples

ErrorLog logs/errors-logs
Enregistrement vers un fichier particulier.

ErrorLog "|/usr/local/bin/erreurs_httpd"
Traitement de l'erreur par un binaire.

ErrorLog syslog:user
Traitement de l'erreur par le daemon syslogd.



- **Le journal des erreurs**

Directive `LogLevel` : Indique quels sont les messages à écrire dans le fichier journal.

Exemple `LogLevel warn`

Tableau des criticités en ordre croissant

<code>emerg</code>	Urgences - le serveur est inutilisable.
<code>alert</code>	Des mesures doivent être prises immédiatement.
<code>crit</code>	Conditions critiques (accès réseau impossible par ex.).
<code>error</code>	Erreurs dans les pages, les scripts.
<code>warn</code>	Avertissements (pages mal codées, erreurs non bloquantes dans un script...
<code>notice</code>	Événement important mais normal.
<code>info</code>	Informations.
<code>debug</code>	Enregistre TOUT ce qui se passe sur le serveur.

☞ *Lorsqu'un niveau particulier est spécifié, les messages de tous les autres niveaux de criticité supérieure seront aussi enregistrés (ex. le niveau `crit` enregistre en plus les messages de niveau `alert` et `emerg`).*



Les fichiers journaux

- **Le journal des erreurs**

Format

[Wed Oct 11 14:32:52 2015] [error] [client 127.0.0.1] client denied by server configuration:
/export/home/live/ap/htdocs/test

*Date et l'heure
du message.*

*Sévérité de
l'erreur
rapportée
(directive
LogLevel)*

*Adresse IP du
client qui a
généralé l'erreur.*

*Le message proprement dit, qui indique
dans ce cas que le serveur a été configuré
pour interdire l'accès au client. Le
serveur indique le chemin système du
document requis (et non son chemin
web).*



Les fichiers journaux

- **Le journal des accès**

Directives : CustomLog, LogFormat & SetEnvIf
Modules: mod_log_config & mod_setenvif

Directive CustomLog : La localisation du journal.

Exemple

CustomLog logs/access_log common



Les logs seront enregistrés selon l'alias 'common' défini dans la directive LogFormat (voir ci-après).



Les fichiers journaux

- **Le journal des accès**

Directive LogFormat : Le formatage des records du journal.

Exemple

LogFormat "%h %l %u %t \"%r\" %>s %b" common

Enregistrement des entrées de journalisation selon le format "Common Log Format" (CLF). Ce format standard peut être produit par de nombreux serveurs web différents et lu par de nombreux programmes d'analyse de journaux.

127.0.0.1 - frank [10/Oct/2015:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326

(%h)
Adresse IP du client qui a envoyé la requête.

- **(%l)**
Information non disponible.

(%u)
Identifiant de la personne qui a demandé le document, issu d'une authentification HTTP. (tiret si absent)

(%t)
L'heure à laquelle la requête a été reçue.

(\"%r\")
Requête du client

(%>s)
Code de statut que le serveur retourne au client.

(%b)
Taille de l'objet retourné, en-têtes non compris. (tiret si aucun contenu retourné).



Les fichiers journaux

- **Le journal des accès**

- Autres types de journaux d'accès possibles :

Journalisation combinée.
Journalisation multiple.
Journalisation conditionnelle.

- Autres tuning possibles :

Rotation des journaux via des journaux redirigés.

- **Les hôtes virtuels**

- Même journal pour tous les logs de tous les hôtes virtuels.
- Un journal séparé pour chaque hôte virtuel.
- Un journal unique mais pouvant être parser via un programme tel que `split-logfile`.

Plus d'info : <https://httpd.apache.org/docs/2.4/fr/logs.html>



Exercice 4

Continuez la configuration d'Apache pour qu'il réponde aux exigences suivantes :

	Site maître	Sites virtuels
Journal des erreurs		
Localisation	<code>logs/error_log</code>	<code>log/error_log</code>
Niveau de criticité	<code>debug</code>	<code>warn</code>
Journal des accès		
Localisation	<code>logs/access_log</code>	<code>logs/access_log.prefixe</code>
Format	<code>combined</code>	<code>common</code>

Qu'est-ce que le format combiné ? → Voir l'aide sur le site d'Apache

Testez votre nouvelle configuration.



WEBOGRAPHIE

<http://irp.nain-t.net/doku.php/210http:start>
<http://www.linux-france.org/prj/edu/archinet/systeme>
<http://www.commentcamarche.net/contents/crypto>
<http://httpd.apache.org/docs/2.4/>

BIBLIOGRAPHIE

Guide de référence 'Apache 2' - JM CULOT - OEM Eyrolles

Apache 2.0 - Guide de l'administrateur Linux - C. AULDS - Eyrolles

Apache 2.4 - Installation et configuration (Nicolas Martinez)- Juin 2015 Edition: ENI

