L'agent de transfert de Courrier: EXIM

Grégoire EHOUMI

AFNOG 2014, Djibouti, Djibouti

Fichier de Configuration

- Exim utilise un fichier unique de configuration, qui est divisé en un certain nombre de sections
- La première section contient les options globales
- Les autres sections commencent avec “begin *sectionname”*
- Ils sont optionnels, et peuvent apparaître dans n'importe quel ordre
- *Les commentaires, les macros, et les inclusions sont disponibles*
- *Les paramètres optionnels peuvent se rapporter aux fichiers de données auxiliaires, par exemple, un fichier d’alias (habituellement /etc/aliases)*

Changement de configuration

- Editez **/usr/exim/configure avec votre éditeur de texte favori**
- **Les nouveaux processus de exim prendront le nouveau fichier tout de suite**
- **Vous avez besoin d'envoyer le signal SIGHUP au démon pour le redémarrer**

```
kill -HUP `cat /var/spool/exim/exim-daemon.pid`
```

- **Vérifiez le journal (log) pour voir si exim a redémarré avec succès**

```
tail /var/spool/exim/log/mainlog
```

Les sections du fichier de configuration

- Options globales
  Options générales et d'entrée relatives
- Règles de réécriture d'adresse
  spécifie la réécriture de l'enveloppe et de l'entête des adresses
- Règles de nouvelle tentative
  Contrôle les nouvelles tentatives après des échecs temporaires
- Configuration du chemin (routeur)
  Spécifie le traitement des adresses destinataires
- Configuration du transport
  Spécifie comment les livraisons réelles sont faites
- configuration de l'authentification
  Spécifie les méthodes d'authentification SMTP
- Listes de contrôles d'accès :Acces Control Lists (ACLs)
  Définie les politiques pour les SMTP entrants

Présentation du fichier de configuration par défaut

Paramétrage options globales
`begin ACL`
Listes de contrôles d'accès
`begin routers`
Configuration du routeur(router)
`begin transports`
Configuration du transport
`begin retry`
Règles des tentatives
`begin rewrite`
Règles de réécritures
`begin authenticators`
Configuration d'authentification

Requis pour SMTP entrant

Requis pour la livraison de message

Exemples d 'options globales communes

- Limites des entrées SMTP

```
smtp_accept_max = 200
smtp_accept_queue = 150
smtp_accept_reserve = 10
smtp_reserve_hosts = 192.168.0.0/16
smtp_connect_backlog = 100
```

- Surcharge

```
queue_only_load = 5
deliver_queue_load_max = 7
```

- Limites de taille de message

```
message_size_limit = 10M
return_size_limit = 65535
```

Les routeurs de Exim 4

- Exim contient un certain nombre de différents routeurs

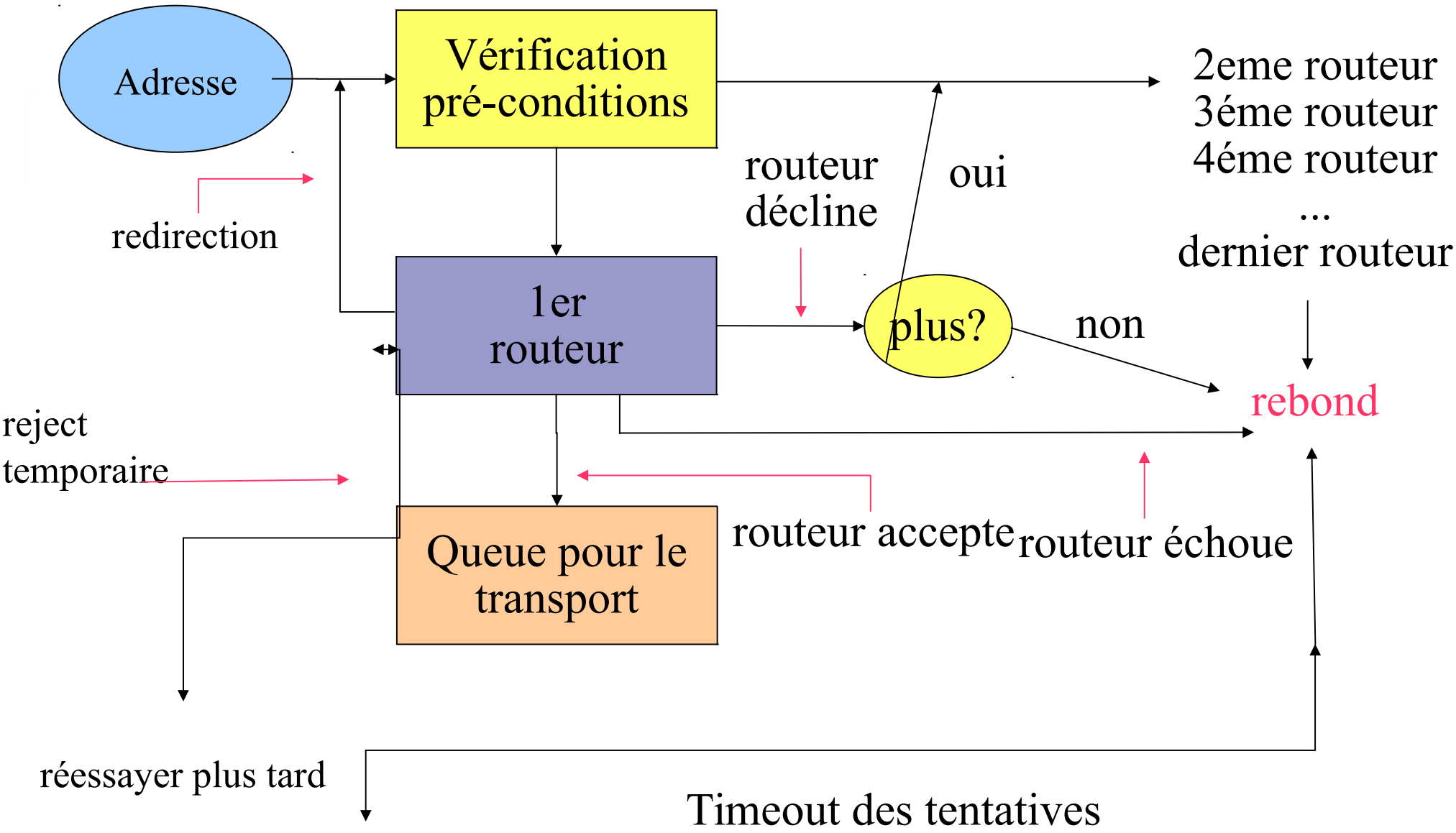
  Exemple: *le routeur dnslookup fait le traitement DNS*

  le routeur redirect fait la redirection d' adresse

  (l'aliasing et le forwarding)
- *La configuration définit quels routeurs sont utilisés, dans quel ordre, et dans quelles conditions*

  Exemple: les routeurs sont souvent limités à des domaines spécifiques
- *Le même routeur peut apparaître plus d'une fois, habituellement avec différentes configurations*
- *L'ordre dans lequel les routeurs sont définis est très important*

L'acheminement (routage) dans Exim 4



Configuration de routage simple

- Vérifiez le domaine nonlocal : exécutez le routeur '*dnslookup*'
  - *Accepter: Queue pour le transport smtp*
  - *Rejeter:Si "no_more" défini => rebond*
- *Vérifiez les aliases système: le routeur 'redirect'*
  - *Accepter: génère de nouvelle(s) adresse(s)*
  - *Rejeter: passé au prochain routeur*
- *Vérifiez les forward des utilisateurs locaux : autre routeur 'redirect'*
  - *Accepter: génère nouvelle(s) adresse(s)*
  - *Rejeter: passé au prochain routeur*
- *Vérifiez les utilisateurs locaux: exécutez le routeur 'accept'*
  - *Accepter: file d'attente pour le transport 'appendfile'*
- *Plus de routeurs à essayer => rebond*

Transports de Exim

- Les transports sont les composants de Exim qui délivrent réellement les copies des messages
  - *Le transport 'smtp' délivre sur TCP/IP aux sites distants*
  - *Le transport 'appendfile' écrit dans un fichier local*
  - *Le transport 'pipe' écrit vers autre processus via un pipe*
  - *Le transport 'lmtp' fait de même, en utilisant LMTP*
  - *Le transport 'Autoreply' est anormal, parce qu'il crée une réponse automatique au lieu de faire une vraie livraison*
- *L'ordre dans lequel des transports sont définis est sans importance*
- *Un transport est utilisé uniquement si référencé par un routeur*
- *Des transports sont exécutés dans des sous-processus, sous leur propre uid, après le routage*

Les routeurs par défaut (1)

- Le premier routeur gère les domaines non locaux

```
dnslookup:
  driver = dnslookup
  domains = ! +local_domains
  ignore_target_hosts = 127.0.0.0/8
  transport = remote_smtp
  no_more
```

- Des pré-conditions verifiées pour un domaine non local
- Des entrées DNS “idiotes” sont ignorées
- Si le domaine est trouvé dans le DNS, mettre en queue pour **remote_smtp**
- Dans le cas contraire, **no_more** transforme le “rejet” en “echec”

Les routeurs par défaut (2)

- Le deuxième routeur manipule les aliases système

```
system_aliases:
  driver = redirect
  data = ${lookup{$local_part}lsearch\
         {/etc/aliases}}
  allow_fail                    allows :fail:
  allow_defer                   allows :defer:

  pipe_transport = address_pipe
  file_transport = address_file
  user = exim
```

- Les lignes du fichier alias ressemblent à ceci

```
postmaster:  pat, james@otherdom.example
retired:     :fail: No longer works here

majordomo:   |/usr/bin/majordom ...
```

Les routeurs par défaut(3)

- Le troisième routeur manipule les fichiers "*.forward*" *des utilisateurs*

```
userforward:
   driver = redirect
   check_local_user
   file = $home/.forward
   no_verify
   pipe_transport = address_pipe
   file_transport = address_file
   reply_transport = address_reply
   allow_filter
```

- ***"data"et "file" sont des options mutuellement exclusives pour"redirect"***

  data s'étend à une liste de redirection

  file s'étend au nom d'un fichier contenant une telle liste

Les routeurs par défaut(4)

- Le routeur final manipule les boîtes aux lettres des utilisateurs locaux :

```
localuser:
  driver = accept
  check_local_user
  transport = local_delivery
```

- Récapitulation - une adresse est routée comme ceci:

  Adresse distante => **remote_smtp transport**

  System_aliases => nouvelle adress(es), fail, defer

  .forward => nouvelles adress(es)

  Utilisateur local => local_delivery transport

  adresse non routable => rebond

- ***Juste un cas de configuration parmi tant d'autres***

Transports par défaut (1)

- Principaux transports

```
remote_smtp:
  driver = smtp

local_delivery:
  driver = appendfile
  file = /var/mail/$local_part
  delivery_date_add
  return_path_add
  envelope_to_add
# group = mail
# mode = 0660
```

- Le défaut suppose un répertoire avec “sticky bit”

  Le paramétrage du groupe et du mode est une approche alternative

Transports par défaut(2)

- Transports auxiliaires

```
address_pipe:
  driver = pipe
  return_output

address_file:
  driver = appendfile
  delivery_data_add
  return_path_add
  envelope_to_add

address_reply:
  driver = autoreply
```

Routage vers les "smarthosts"

- Remplacer le premier routeur par ceci

```
send_to_smarthost:
  driver = manualroute
  domains = ! +local_domains
  route_list = * smart-host1.example:\
                 smart-host2.example
  transports = route_smtp
```

- La règle **route_list** contient trois éléments séparés :

  Le premier représente le domaine : * correspond à n'importe quel domaine

  Le second est une liste de machines pour les domaines correspondants

  Le troisième est "byname" (par défaut) ou bydns

- Mettez **"hosts_randomize"** pour trier les serveurs de façon aléatoire chaque fois

Les domaines virtuels

- Les cas simples sont juste des alias

```
virtual_domains:
  driver = redirect
  domains = lsearch;/etc/virtuals
  data = ${lookup{$local_part}lsearch\
         {/etc/aliases-$domain}}
  no_more
```

- Un alias sans domaine utilise le domaine local qualifié

```
philip:  ph10
jc:      julius@other.domain.com
```

ACLs

- Les ACL s'appliquent seulement aux SMTP entrants
  Mais ils s'appliquent aussi aux SMTP locaux (**bs and bS**)
- **Pour les messages SMTP entrants**
  acl_smtp_rcpt définit le ACL à exécuter pour chaque RCPT
  Le defaut est "deny"
  acl_smtp_data définit le ACL à exécuter après DATA
  Le défaut est "accept"
- **Les tests sur le contenu de message peuvent seulement être faits après DATA**
- **D'autres ACLs peuvent être utilisés pour AUTH, ETRN, EXPN, VRFY**

Un simple ACL

```
acl_smtp_rcpt = acl_check_rcpt

begin acl

acl_check_rcpt:
  accept   local_parts = postmaster
           domains     = +my_domains

  require  verify      = sender

  accept   domains     = +my_domains
           verify      = recipient
```

- Implicitement “deny” à la fin

Les listes nommées

```
domainlist local_domains = @ : plc.com
hostlist   relay_hosts   = 192.168.32.0/24
```

- NB : la liste est spécifiée à un seul endroit
  Les références sont plus courtes et plus faciles à comprendre
- Optimisation: des correspondances dans les listes nommées sont mises en cache
  Exemple: plusieurs routeurs examinant la même liste de domaines
- Une liste nommée est référencée en mettant '+' devant son nom
  `hosts = 127.0.0.1 : +relay_hosts`

- Une liste nommée peut être inversée
  `domains = !+local_domains`
  Ceci n'est pas possible avec les macros

Les déclarations de ACL

- Chaque déclaration contient un « verbe » et une liste de conditions

  verb condition 1 (une par ligne)

  condition 2

  ...

- *Si toutes les conditions sont remplies*

  "accept" Permet l'éxécution de la commande SMTP (dans le cas contraire, "pass" ou "reject" – voir prochain slide)

  "Pass" ou "reject" voir prochain slide)

  "deny" Rejet (sinon passe)

  "require" Passe (sinon rejet)

  "warn" éxécute une action d'avertissement (par exemple : écrire des journaux ou ou ajouter des entêtes) : Passe t oujours

Les modificateurs de ACL

- **Message définissant un message personalisé pour un refus ou un avertissement**

```
deny     message  = You are black listed at \
                    $dnslist_domain
         dnslists = rbl.mail-abuse.org : ...
```

- **log_message définit un message journal personalisé**

```
require log_message = Recipient verify failed
        verify      = recipient
```

- **"endpass" est utilisé avec le verbe "accept" pour des résultats spécifiques**

```
accept  domains  = +local_domains
        endpass
        verify   = recipient
```

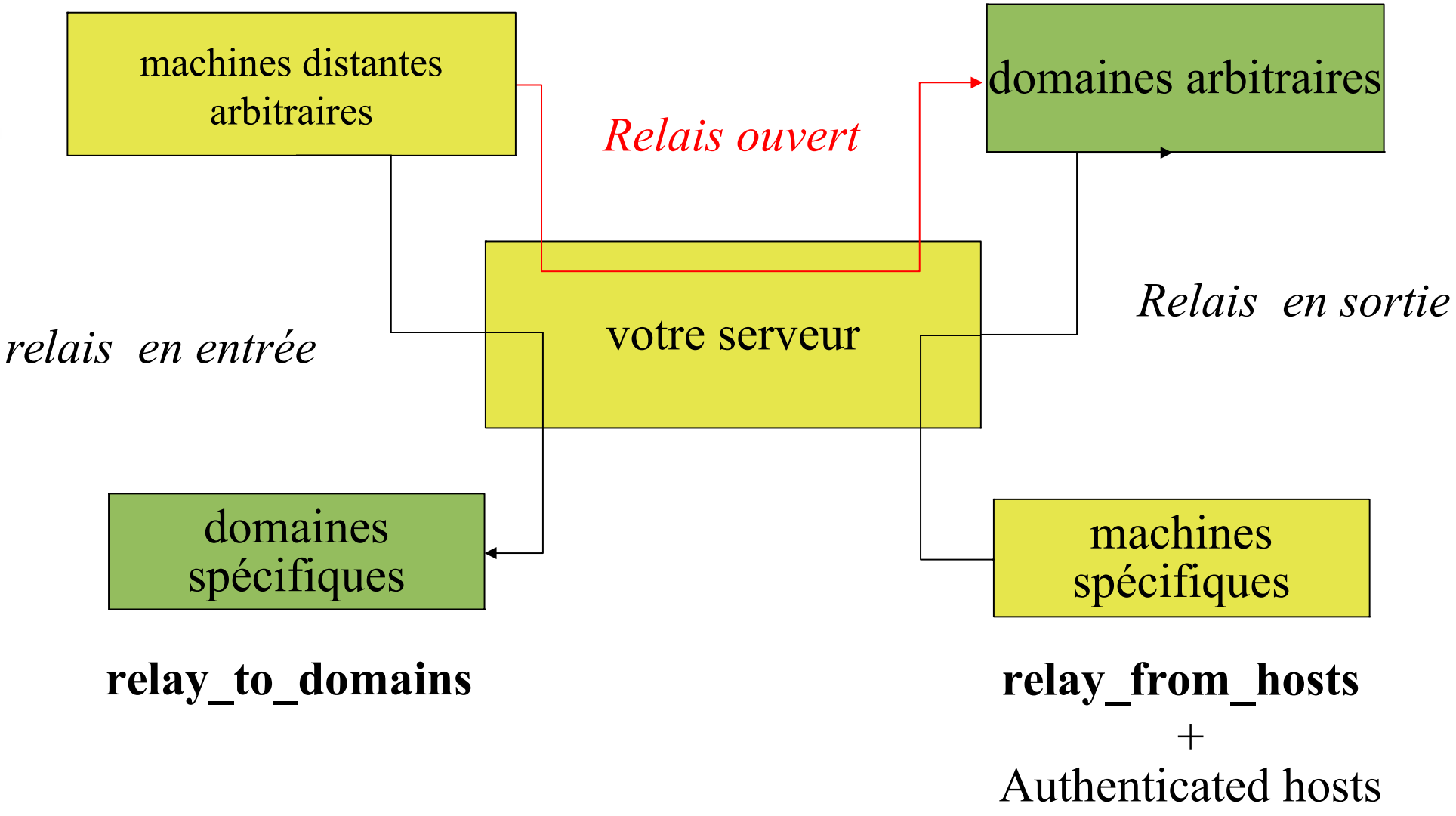
Au dessus de "endpass", l'échec cause l'éxecution de la prochaine déclaration

Au dessous de "endpoint", l'échec cause le rejet

ACLs par défaut

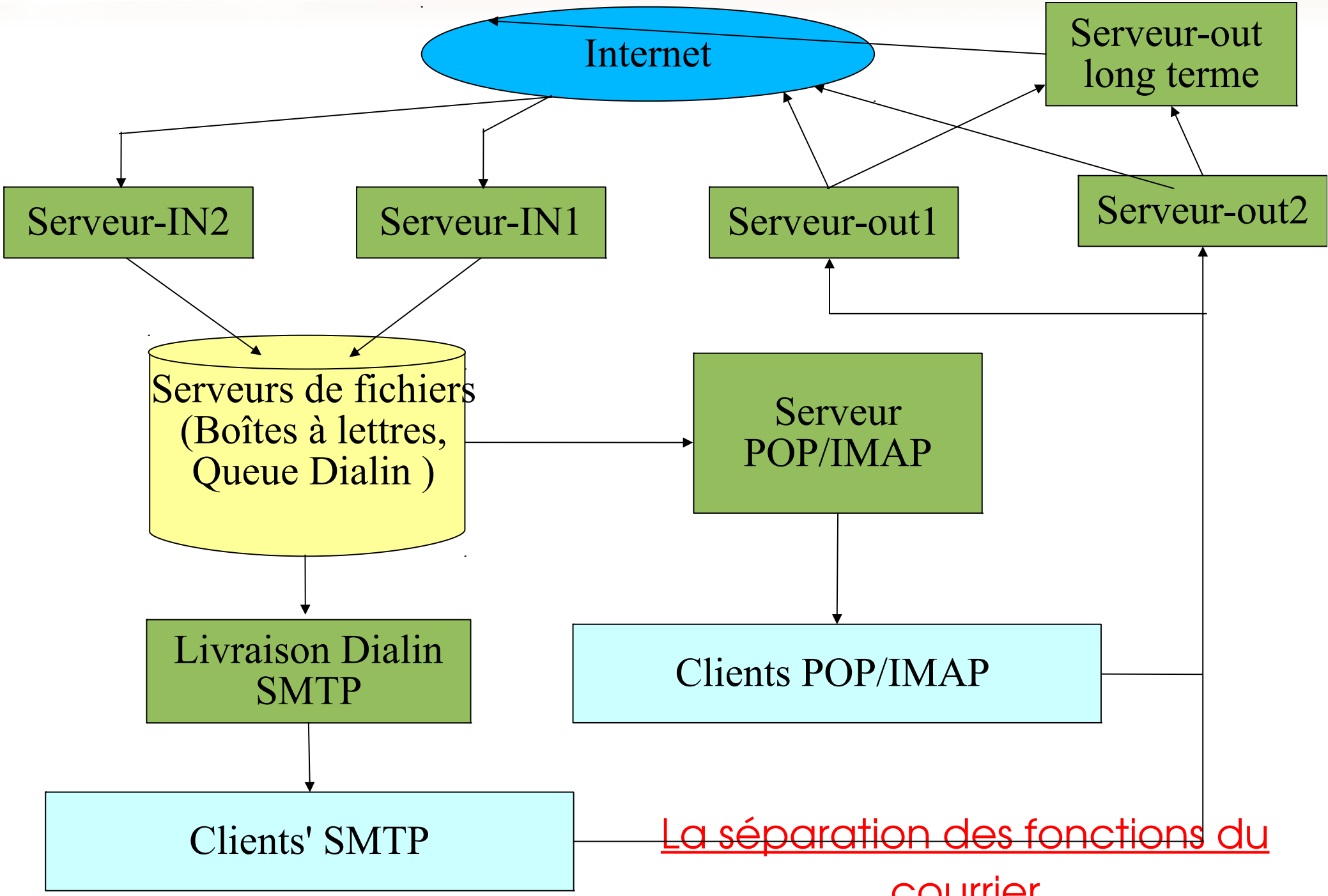
```
acl_check_rcpt:
  accept  hosts         = :
  deny    local_parts   = ^.*[@%!|/] : ^\\.
  accept  local_parts   = postmaster
          domains       = +local_domains
  require verify        = sender
  accept  domains       = +local_domains
          endpass
          message       = unknown user
          verify        = recipient
  accept  domains       = +relay_to_domains
          endpass
          message       = unrouteable address
          verify        = recipient
  accept  hosts         = +relay_from_hosts
  accept  authenticated = *
  deny    message       = relay not permitted
```

Bon et mauvais relais



Grandes installations

- Utilisez un serveur de nom local avec beaucoup de mémoire
- Exim est limité par les entrés/sorties disque
  - Utilisez un système disque rapide
  - Utiliser le **split_spool_directory**
  - Utilisez plusieurs répertoires pour les boîtes aux lettres
- **Évitez les fichiers au mot de passe linéaire**
- **Utilisez le format maildir pour permettre les livraisons parallèles**
- **Projetez d'agrandir le système avec des serveurs parallèles**
  - Ceci aide aussi à ajouter plus de capacité disque
- **Séparez le courrier entrant et sortant**
- **Gardez la file d'attente de sortie aussi courte que possible**
  - Utilisez des serveurs de chute (fallback hosts) et/ou $message_age pour plusieurs niveaux



La séparation des fonctions du courrier