

Architecture de serveurs virtualisés pour la communauté mathématique

Jacquelin Charbonnel

Journées ARAMIS - Lyon, juin 2012
version 1.1



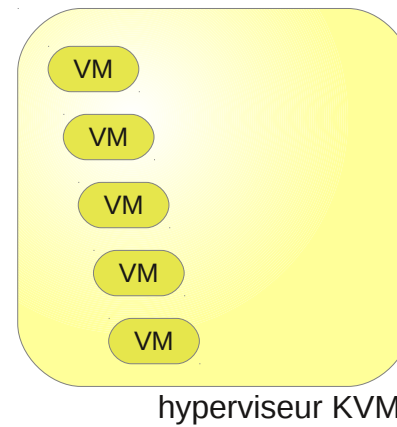
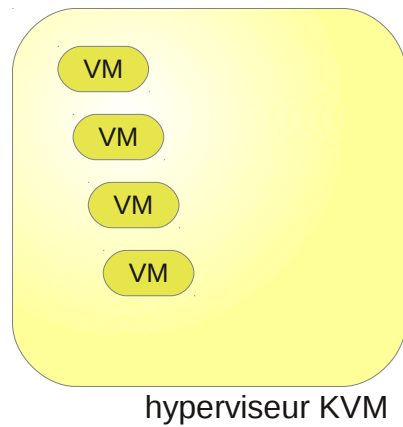
Plan

- KVM pour un laboratoire
- KVM pour la Plateforme en Ligne pour les Mathématiques
- Exemple : plateforme d'hébergement web
- Le projet PLACO

KVM pour un laboratoire

- LAREMA
 - laboratoire de mathématiques
 - une demi-centaine d'utilisateurs
- une dizaine de serveurs physiques
 - hyperviseurs
 - serveurs applicatifs interactifs
 - serveurs d'infrastructures (NFS, LDAP, DNS, DHCP, ntp)
 - serveurs de calcul
 - serveur de sauvegarde

- 2 hyperviseurs
 - KVM sous CentOS 6
 - une quinzaine de VM





Name

CPU usage

coquille1 (QEMU)



base
Shutoff



base_centos6
Shutoff



dellmonitor
Shutoff



icinga_nfs
Running



laremagw
Running



svn
Running



tonton
Running

coquille2 (QEMU)



alceste_nfs
Running



base_centos5
Shutoff



ds389
Shutoff



icinguang
Running



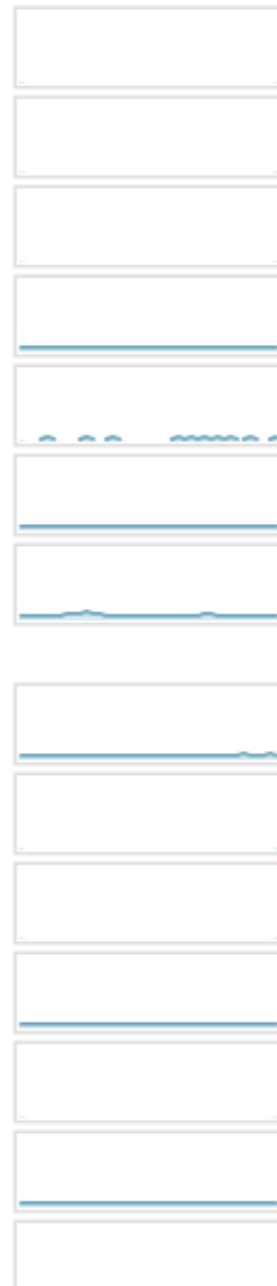
ittp2
Shutoff



roundcube
Running



sogo
Running



Candidats à la virtualisation

- messagerie SMTP+IMAP (postfix+dovecot)
- web + webmail (apache, squirrelmail, roundcube)
- firewall + reverse proxy + MX principal (iptables, apache, postfix)
- monitoring (icinga)
 - monitoring des ressources du laboratoire
 - monitoring des ressources de la DSI
- serveurs ssh
 - tunnels pour les chercheurs du labo
 - tunnels pour les chercheurs de la fédération des Pays de Loire
- divers
 - générateur d'images LTSP

Jugés non virtualisables

- serveurs applicatifs
 - pour raison de performance
- serveur de sauvegarde
 - pour raison d'isolation
- serveur de calcul
 - pour raison de performance (à approfondir)
- serveurs d'infrastructure : nfs, ntp, ldap, dhcp, dns
 - services utilisés par les hyperviseurs

Principe #1 : couches de serveurs

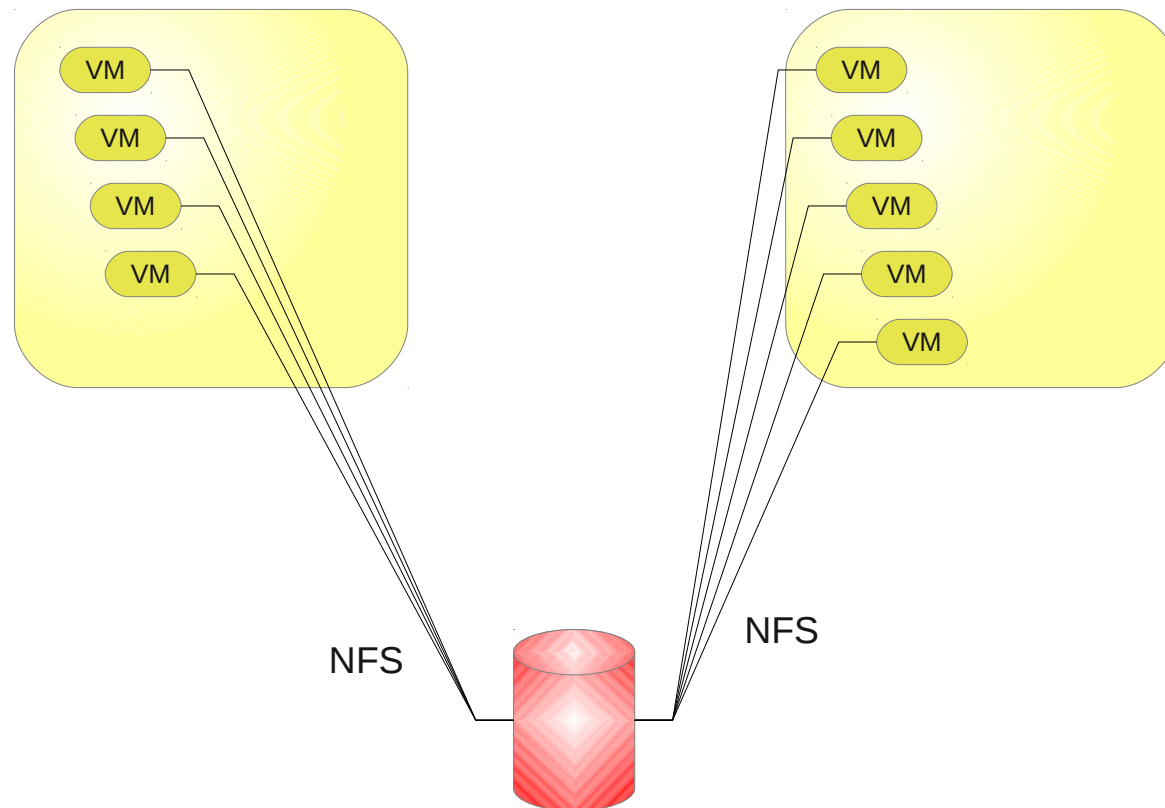
- 2 couches de serveurs
 - les hyperviseurs appartiennent à la couche #1
 - les VM appartiennent à la couche #2
- principe :
 - la couche #1 ne doit pas utiliser les services de la couche #2
 - un hyperviseur ne doit pas dépendre de services hébergés sur une quelconque VM
- pourquoi ?
 - écarter les interdépendances (bloquantes au démarrage)

Principe #2 : les data

- data = homedir, mailboxes, htdocs, etc.
- aucun gros volume de données sur disques virtuels (containers)
 - data exportées par NFS depuis un serveur physique dédié
- pourquoi ?
 - 1 gros disque = 1 gros fichier sur l'hyperviseur (plusieurs Go)
 - lourd à copier, sauvegarder, déplacer, comparer

Principe #2 : les data

- data exportées par NFS depuis un serveur physique dédié



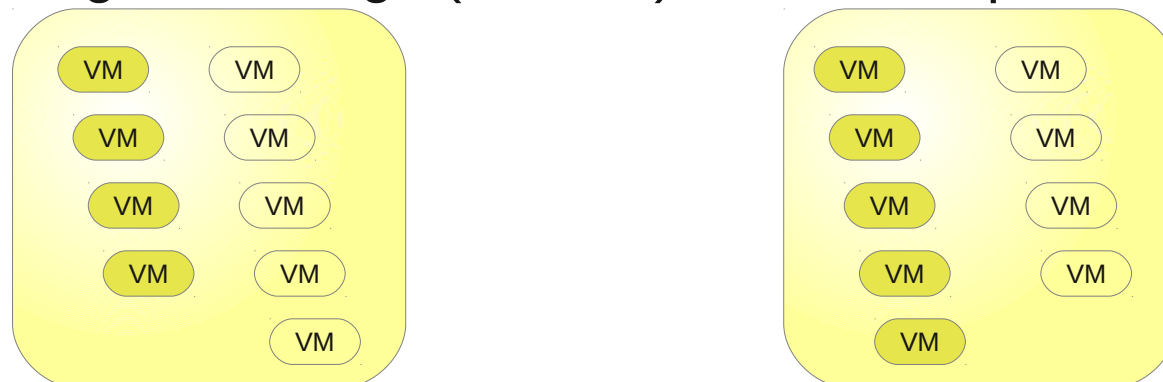
Principe #3 : les VM

les VM sont présentes sur chacun des 2 hyperviseurs

- VM non attachées à l'hyperviseur (*déplaçables*)
- chaque VM est
 - en production sur 1 host
 - en backup sur l'autre

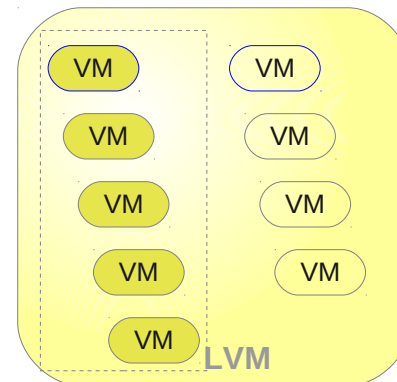
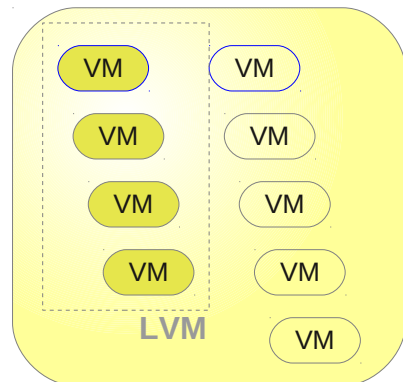
pourquoi ?

- PRA simple
- équilibrage de charge (manuel) des VM en production



Principe #4 : les containers

- container = disque de VM
- les containers sont sur LVM sur l'hyperviseur
- pourquoi ?
 - bénéficier des snapshots au niveau du fs

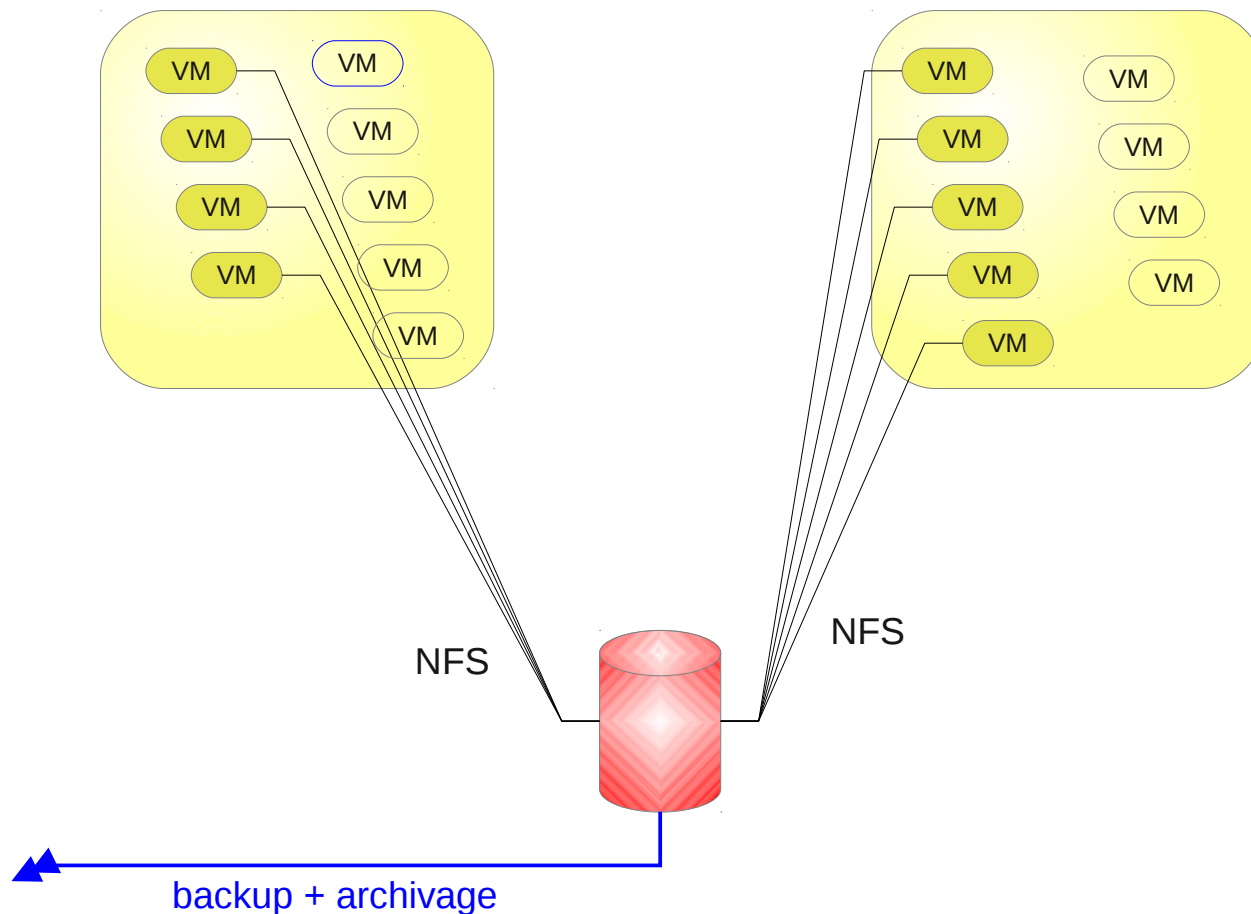


Backups

- backups des data
- backup des systèmes des VM
- backup des containers

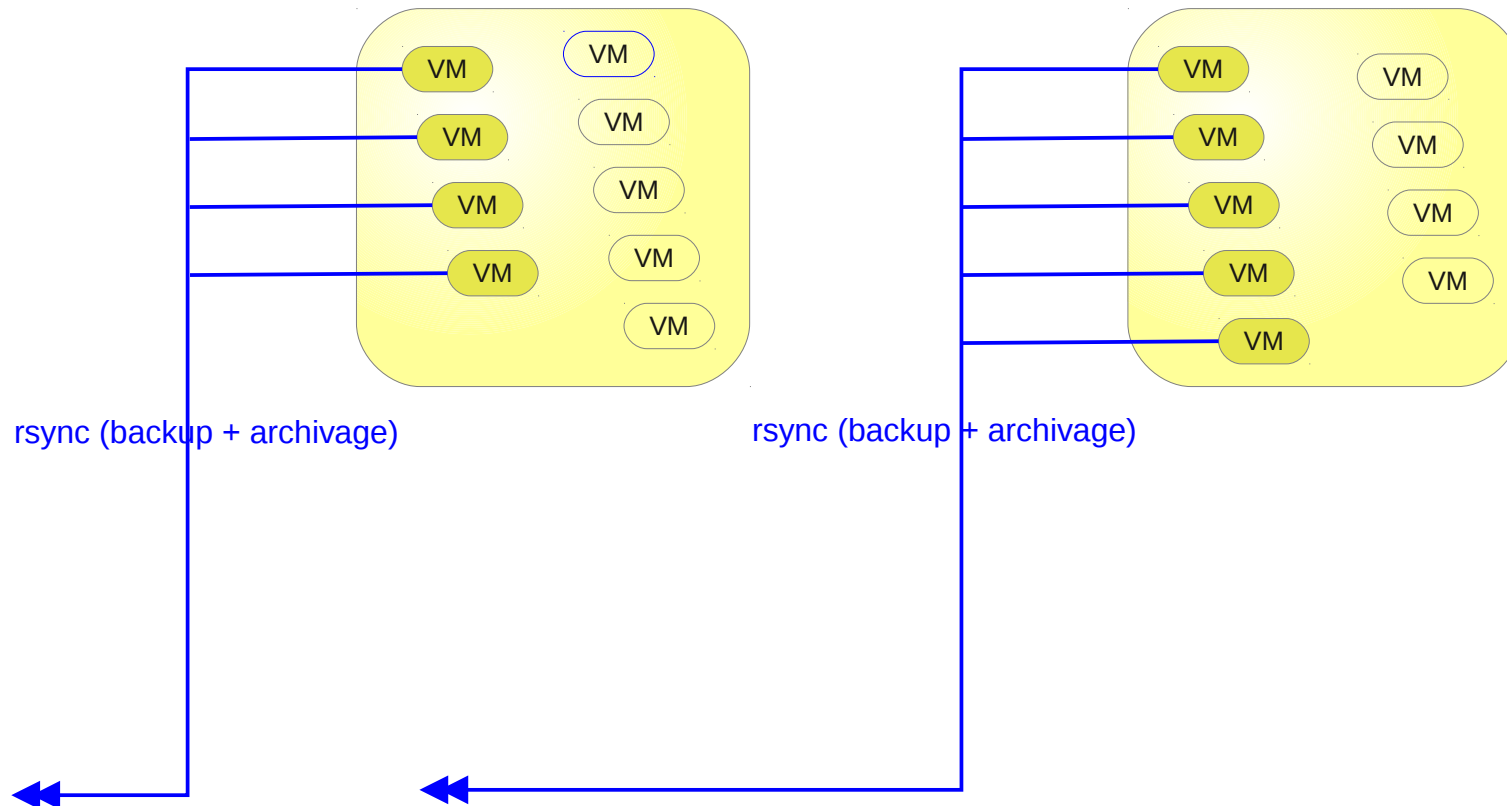
Backup des data

- backup du serveur NFS
 - intégré au système de backup général
 - archivage *daily, weekly, monthly*



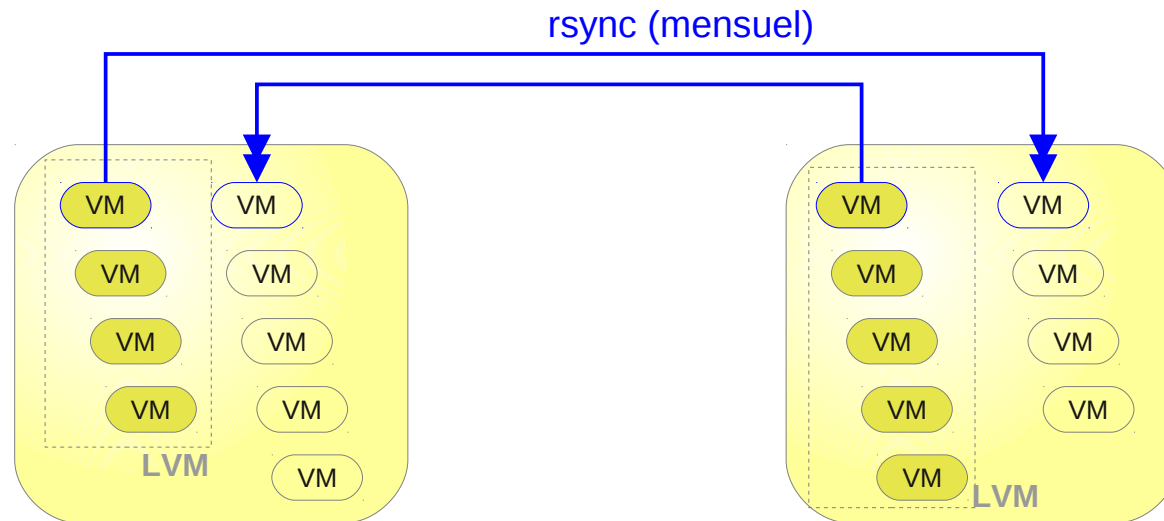
Backup des systèmes

- systèmes sauvegardés via chaque VM
- intégré au système de backup général des serveurs
- archivage *daily, weekly, monthly*



Backup des containers

- sauvegardés 1 fois par mois
- sur le second host
- sans archivage



Backup des containers

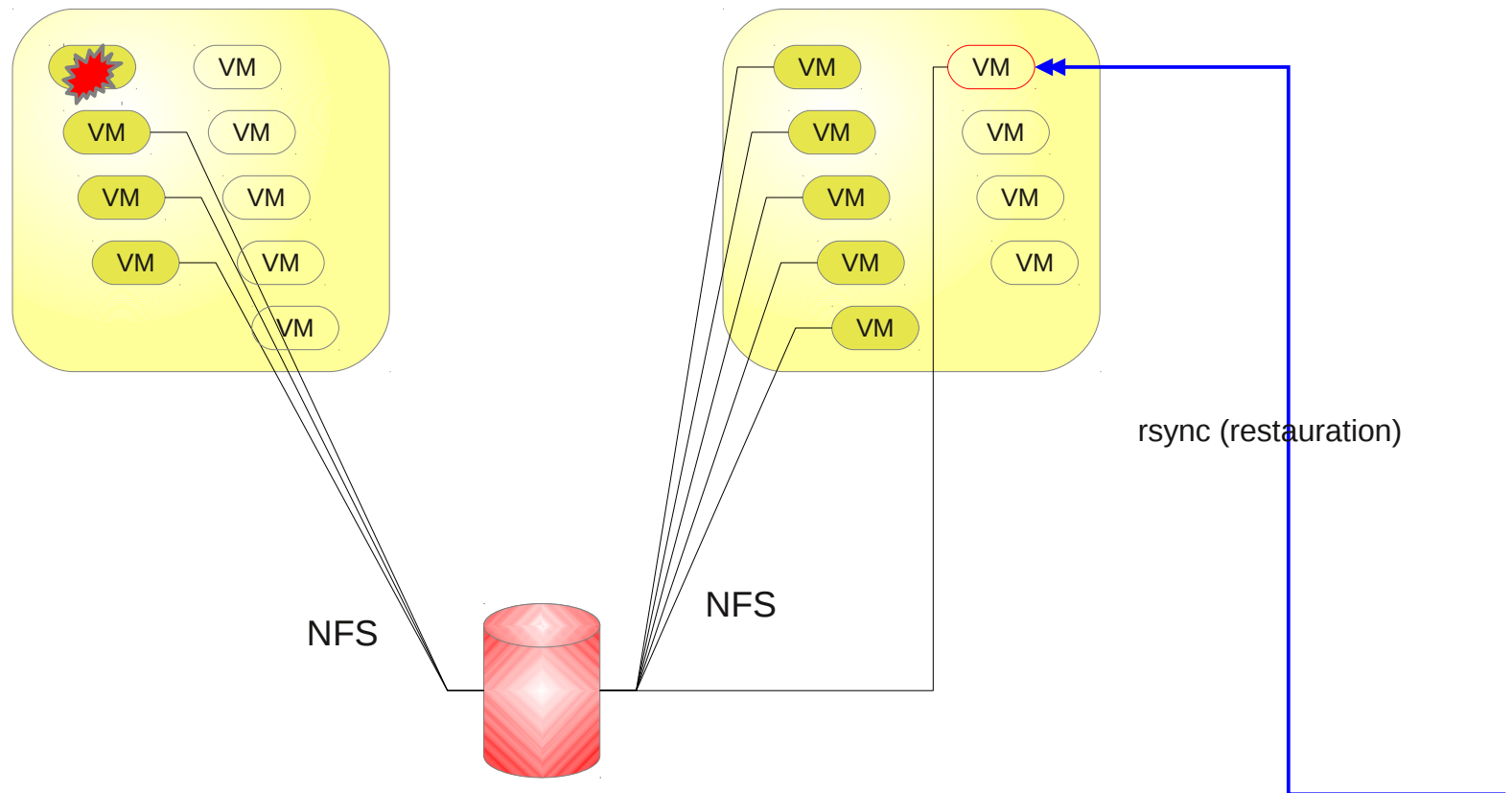
processus :

- pause de toutes les VM (virsh suspend)
 - snapshot LVM du LV contenant les disques (lvcreate --snapshot)
 - redémarrage de toutes les VM (virsh resume)
-
- rsync des disques du snapshot sur l'autre hyperviseur (rsync)
 - suppression du snapshot LVM (lvremove)
 - rsync des définitions de VM (.xml)

moins d'1 sec

Reprise d'activités

- démarrer la VM sur l'autre hyperviseur
- restauration du dernier backup du système (backup)
- reboot

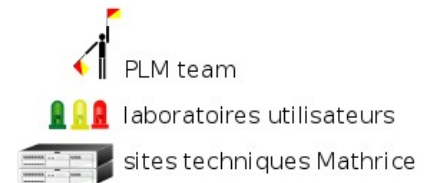
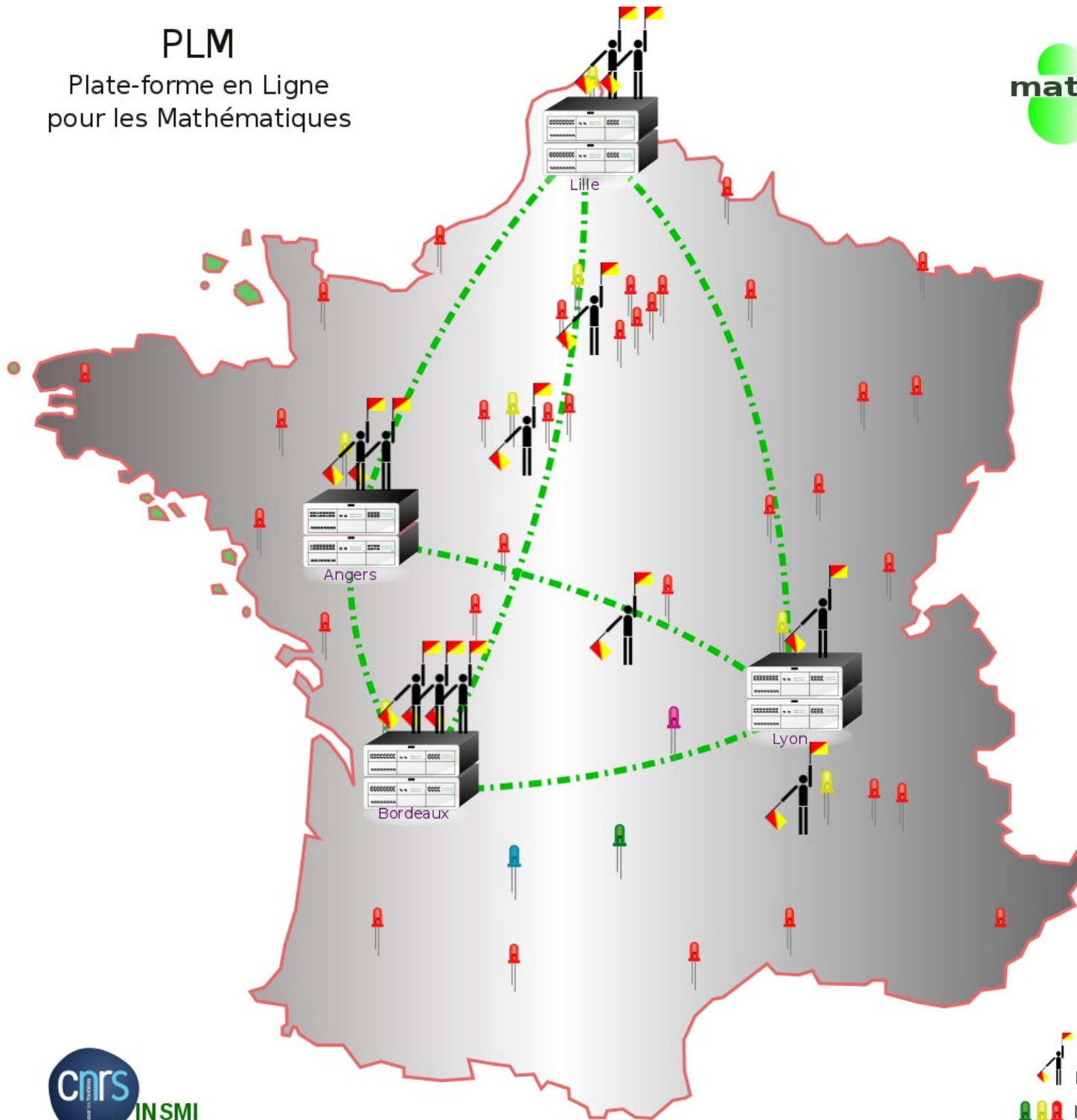


La PLM

- Plate-forme en ligne pour les mathématiques
 - infrastructure répartie géographiquement
 - utilisée par 61 laboratoires (2168 utilisateurs)
- 4 sites techniques (Bordeaux, Lille, Angers et Lyon)
 - 8 hyperviseurs sous CentOS 6
 - 50 VM

PLM

Plate-forme en Ligne
pour les Mathématiques



Les services

- serveurs de licences logicielles
- proxy de consultation des revues scientifiques en ligne
- un annuaire de la communauté mathématique française
- gestion de noms de domaine (`math.cnrs.fr`, `resinfo.org`, etc.)
- messagerie (`@math.cnrs.fr`)
- outils de production et d'organisation personnelle (webmail, agenda, carnet d'adresses)
- serveurs interactifs, de calcul
- outils de travail collaboratif (hébergement web, listes de diffusion, partages réseaux, subversion, web-conférence)

Historique

- initialement sous VMware server 1.x
- 2009-2011 : migration vers KVM
- 2012 : normalisation de l'infrastructure

Pourquoi migrer, pourquoi KVM ?

- VMware 1.x
 - fin de vie de VMware Server 1.x (2009)
 - 1 petit bug jamais résolu au niveau des snapshots
- VMware 2.x
 - pas de console *native*
 - console via un navigateur web
- KVM
 - opensource
 - bien supporté par RedHat (-> natif dans CentOS)
 - communauté très active, développement rapide

Bilan

- meilleures performances
- installation + facile
 - intégré à la distribution CentOS
 - rien à recompiler à chaque nouveau kernel
 - pas de tools à installer sur les VM

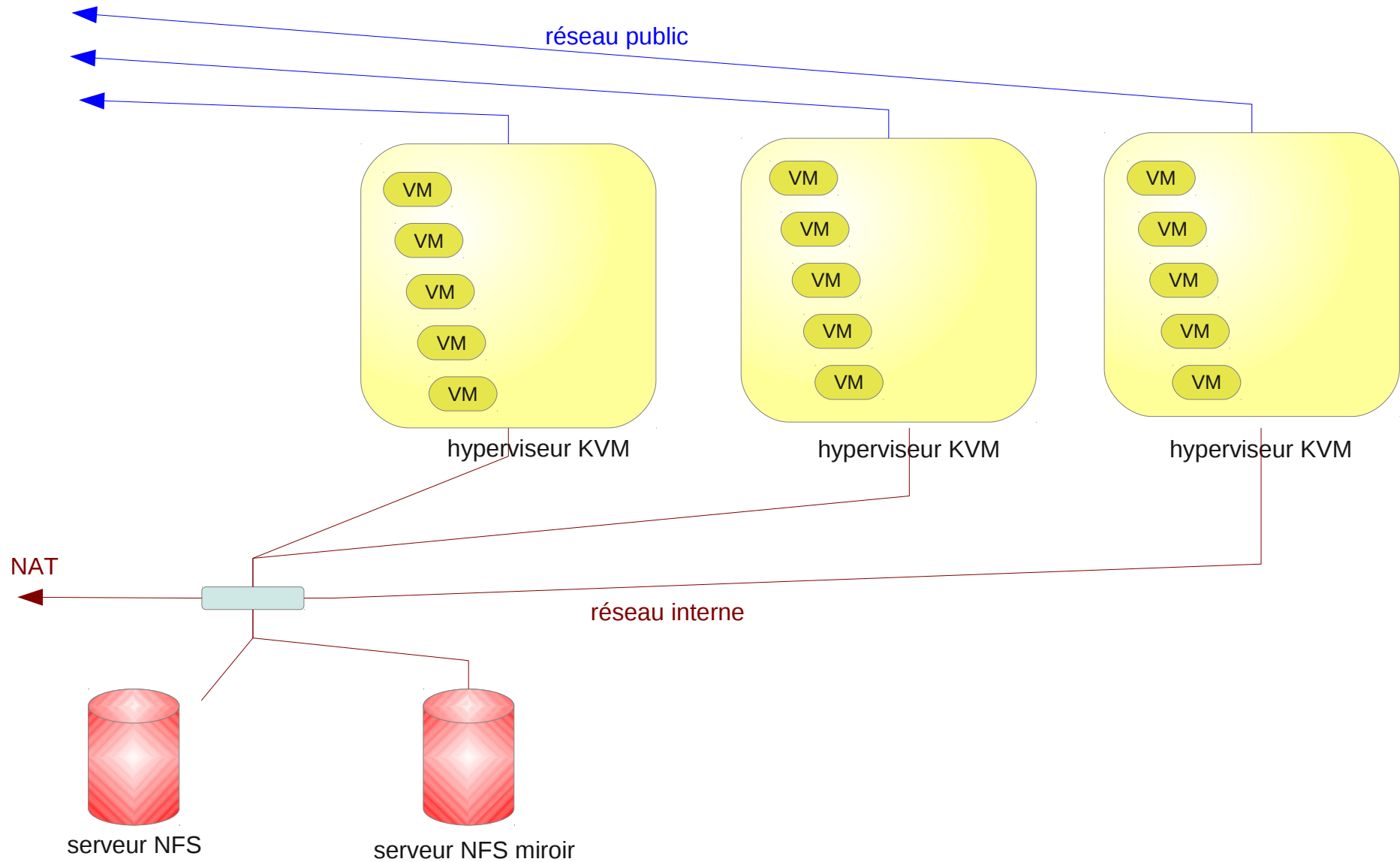
2012 : la normalisation

- simplifier au maximum
 - supprimer les dépendances entre sites
- dissocier les fonctions
 - virtualisation
 - stockage
 - backup et archivage
- sur chaque site :
 - banaliser les hyperviseurs
 - support d'un nombre quelconque d'hyperviseurs

2012 : la normalisation

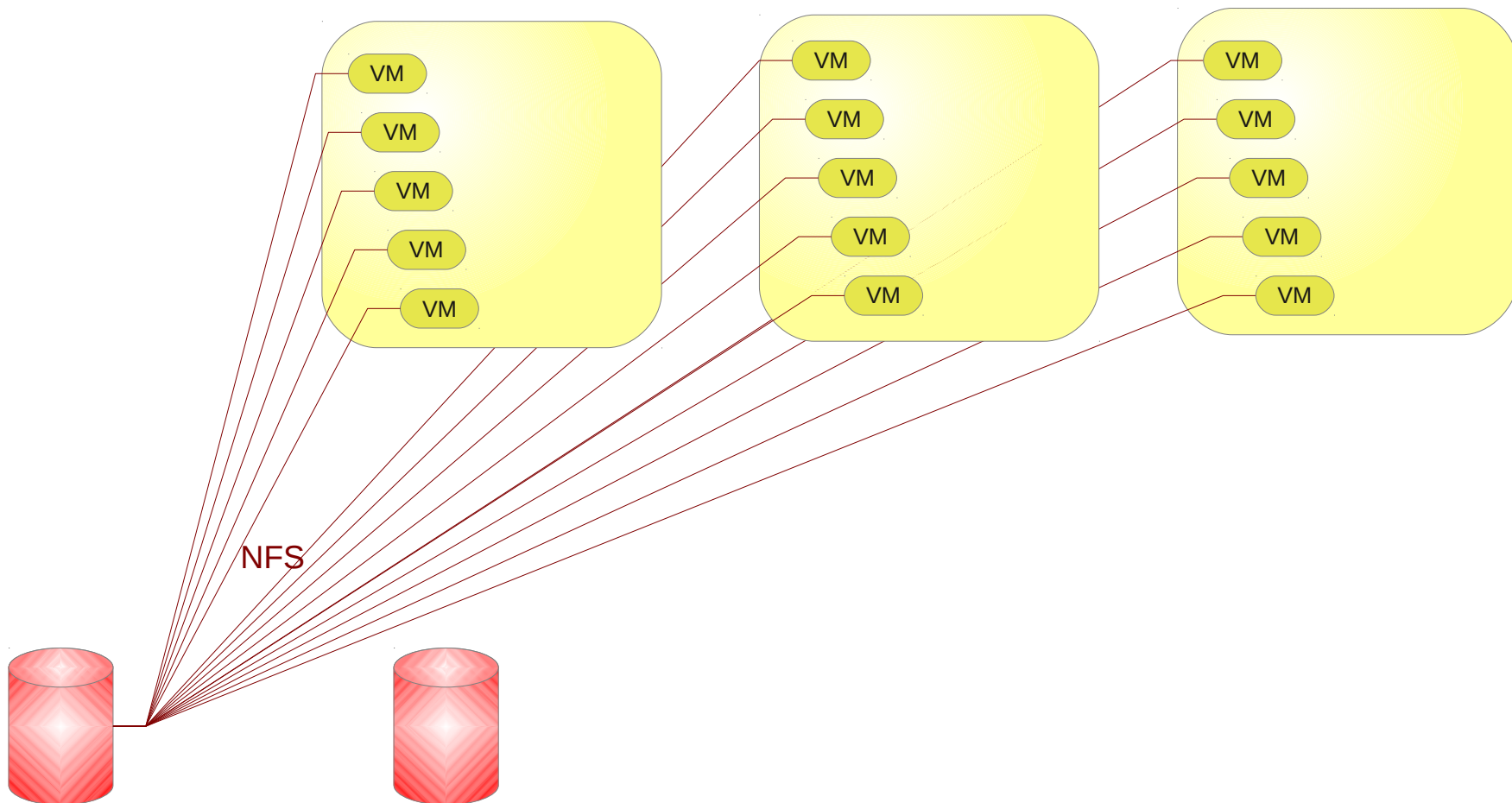
- 1 site =
 - n hyperviseurs banalisés
 - 1 serveur NFS
 - 1 serveur pour PRA (miroir du serveur NFS)
 - 1 vrai réseau interne privé physique
- 1 VM =
 - 1 interface réseau interne NATée (ntp, dns, dhcp, NFS, log, ssh d'admin, system update)
 - 1 interface publique si le service rendu est public
 - déplaçable d'un hyperviseur à l'autre
- configuration gérée par Puppet
- 1 nouveau site pour le backup délocalisé (Grenoble)

site de la PLM



Principe #1 : les data

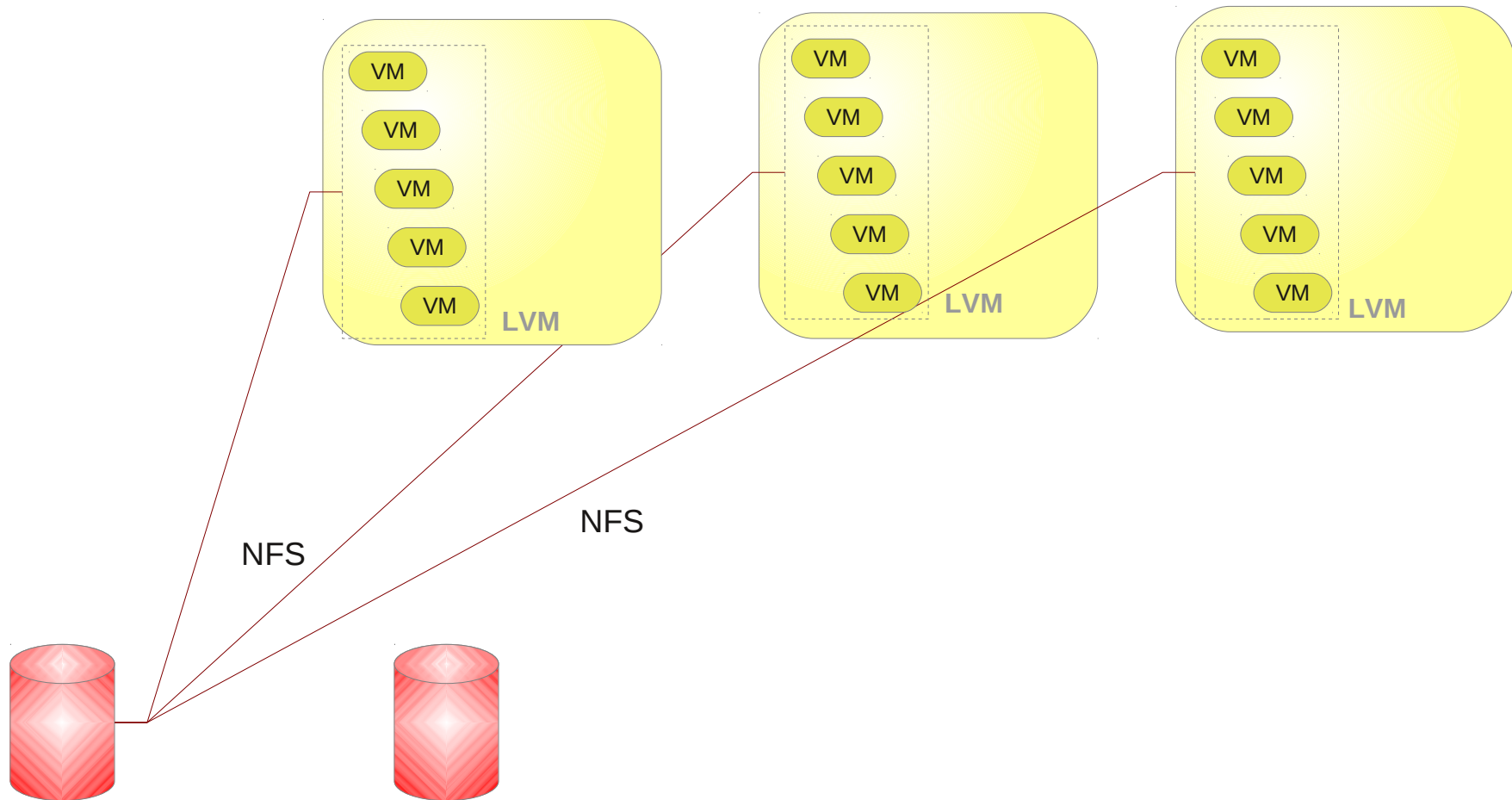
- data = homedir, mailboxes, htdocs, etc.
- aucun gros volume de données dans les containers
 - exportées via NFS depuis un serveur physique dédié



Principe #2 : VM et containers

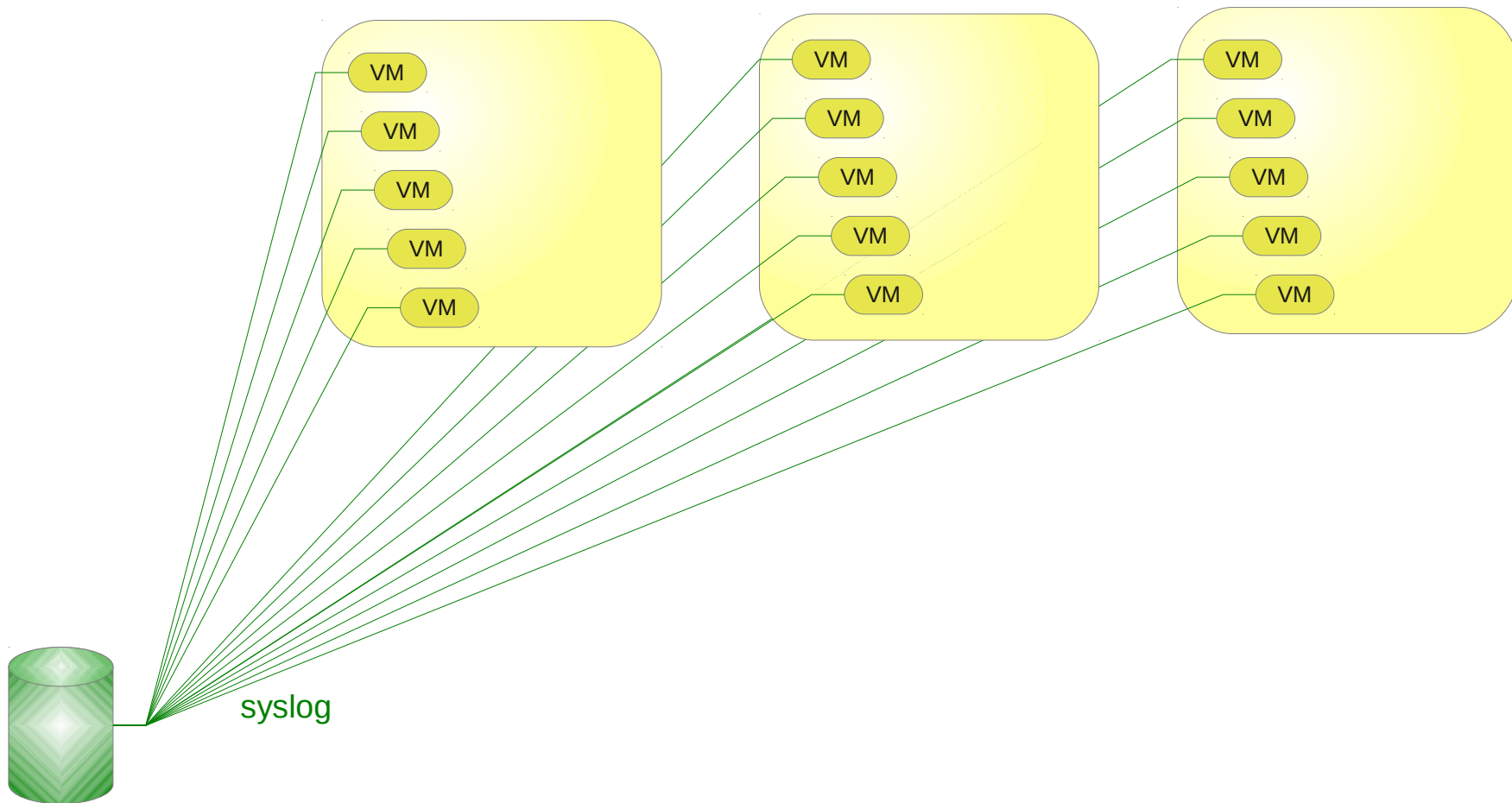
les VM et leurs containers sont sur LVM exporté par NFS

VM non attachée à l'hyperviseur (*déplaçable*)



Trafic NFS

- pour minimiser le trafic NFS
 - pas de log en local sur les VM

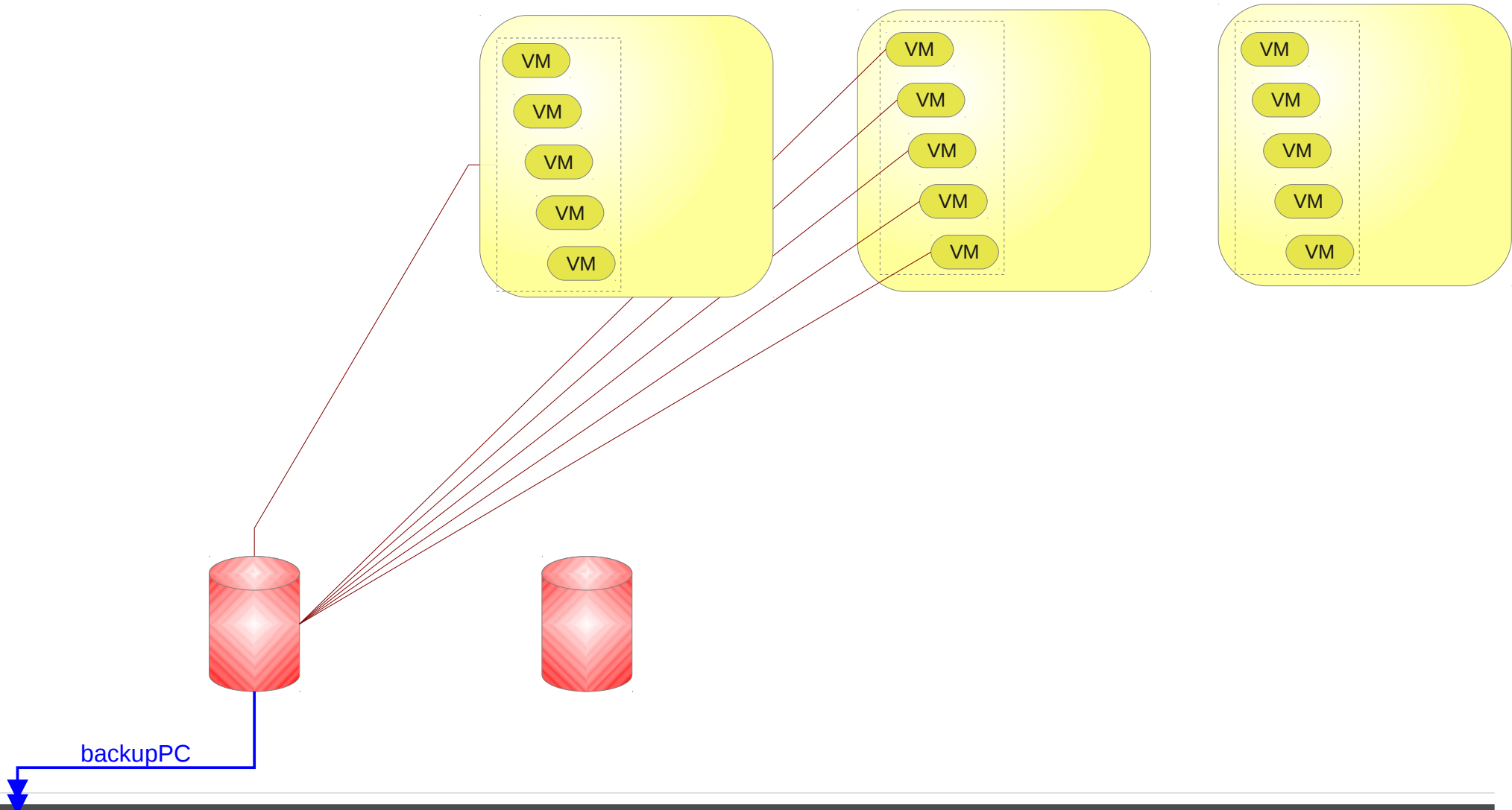


Backups

- backups des data
- backup des systèmes des VM
- backup des containers

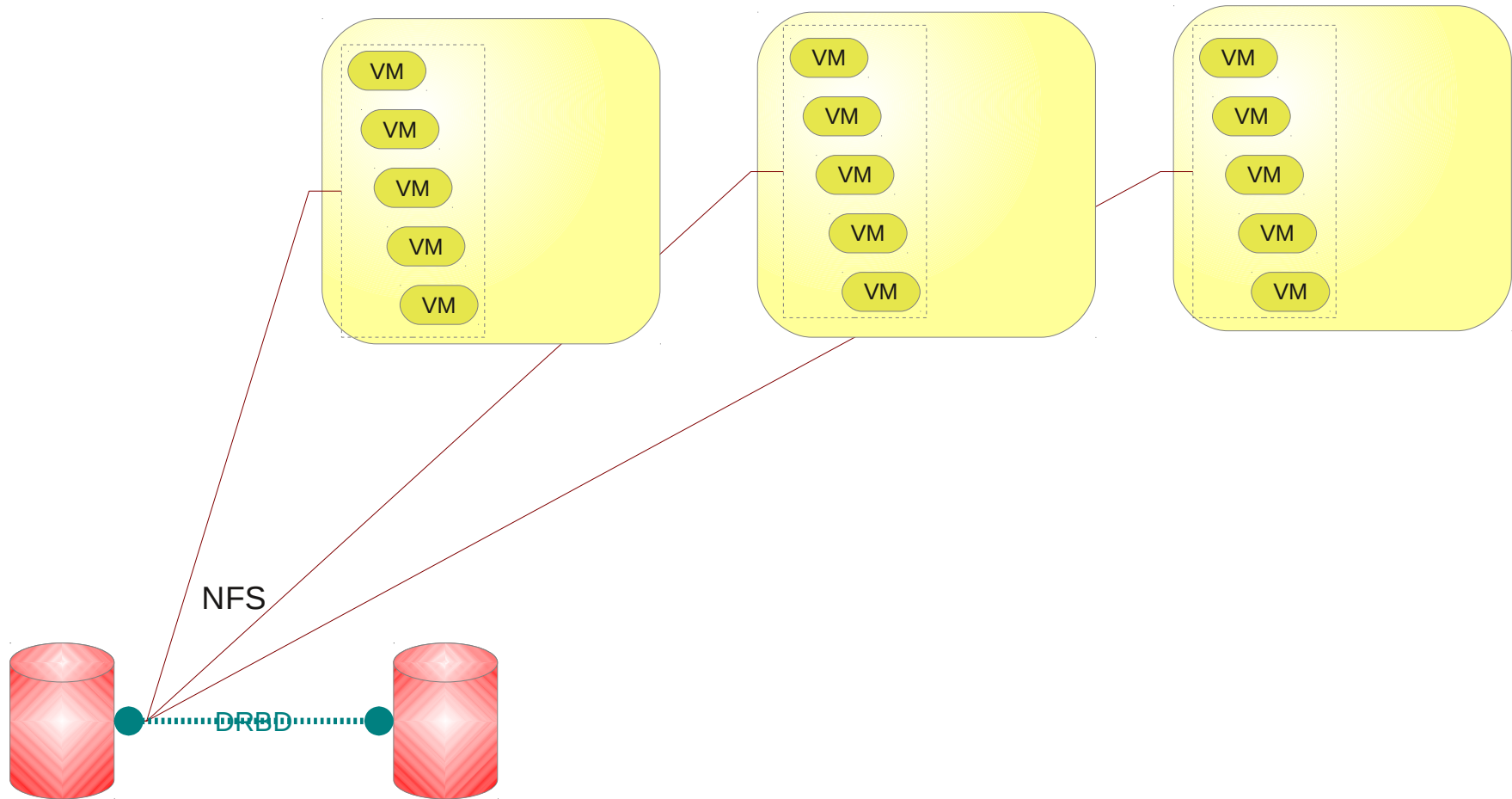
Backup data + backup systèmes

- backup du serveur NFS (backupPC)

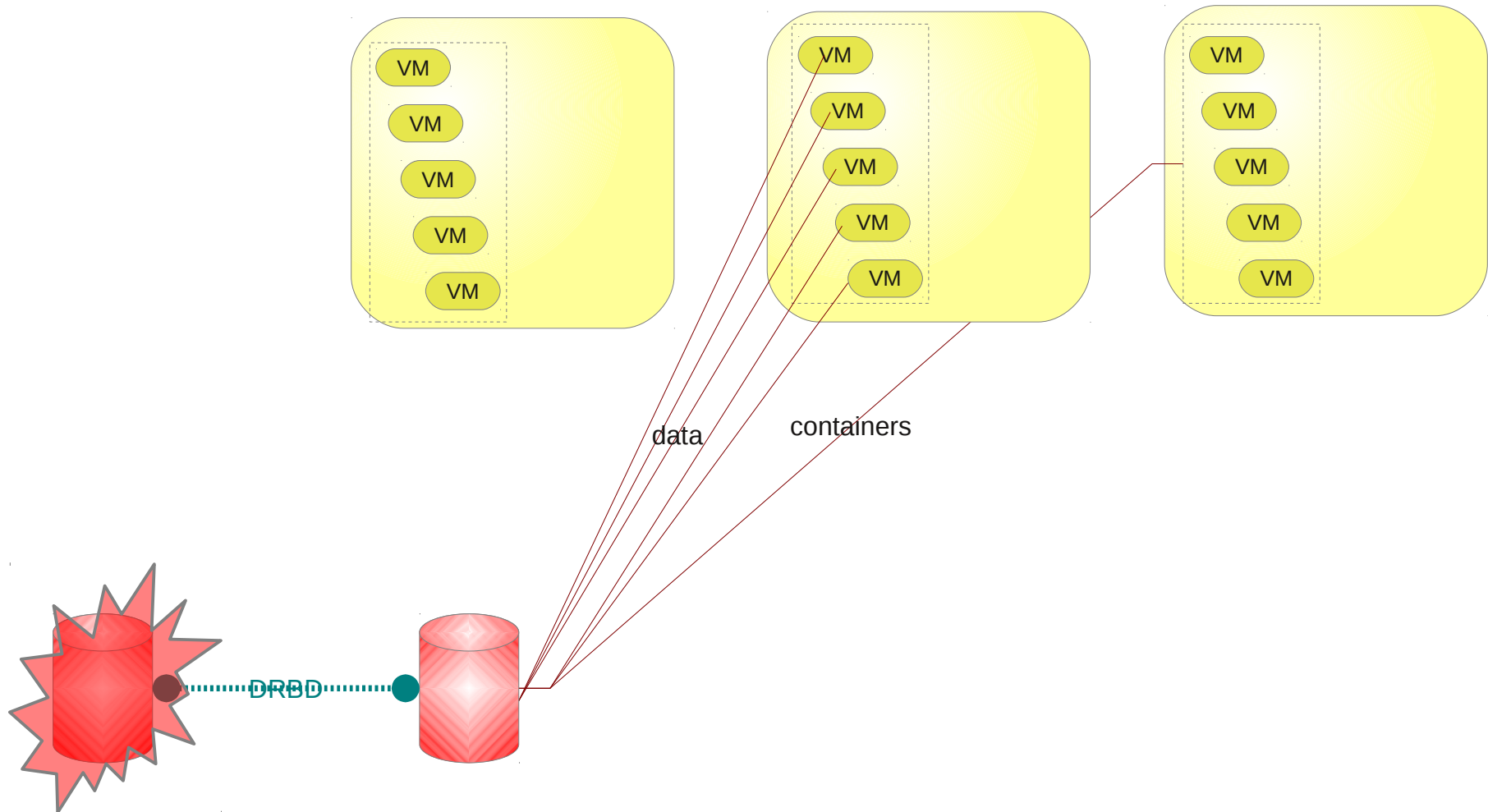


Backup des containers

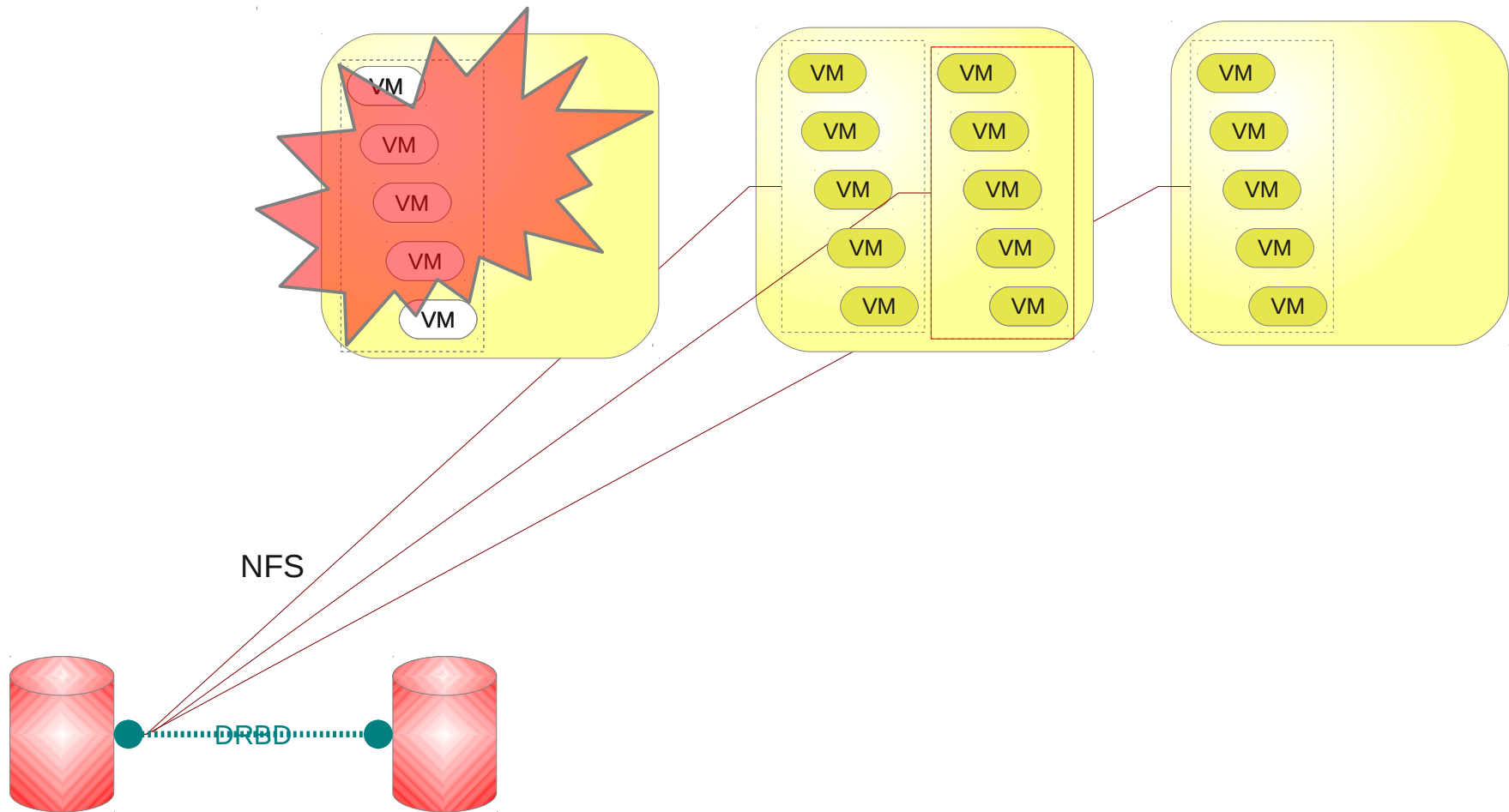
- miroir avec DRBD



Plan de Reprise d'Activité



Plan de Reprise d'Activité



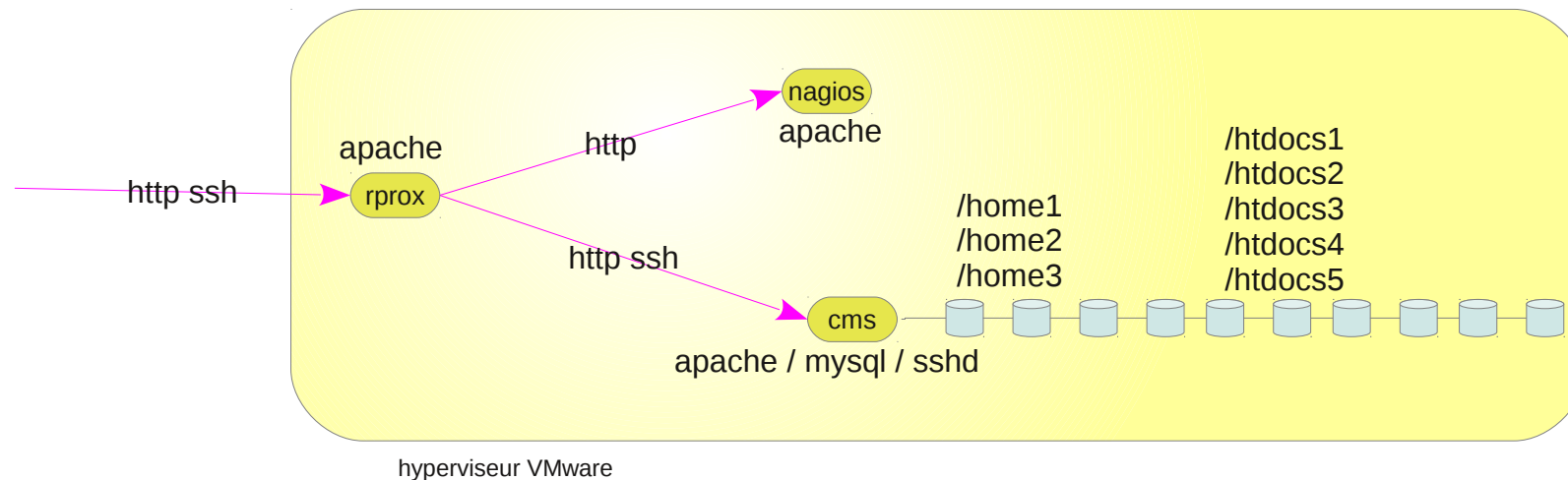
Exemple : l'hébergement web

ou comment la virtualisation peut améliorer les perfs

- 70 sites web
- sites statiques, CMS (SPIP, drupal, etc.)
- quelques sites gourmands en ressources
 - générateurs d'images à la volée
- quelques sites très fréquentés

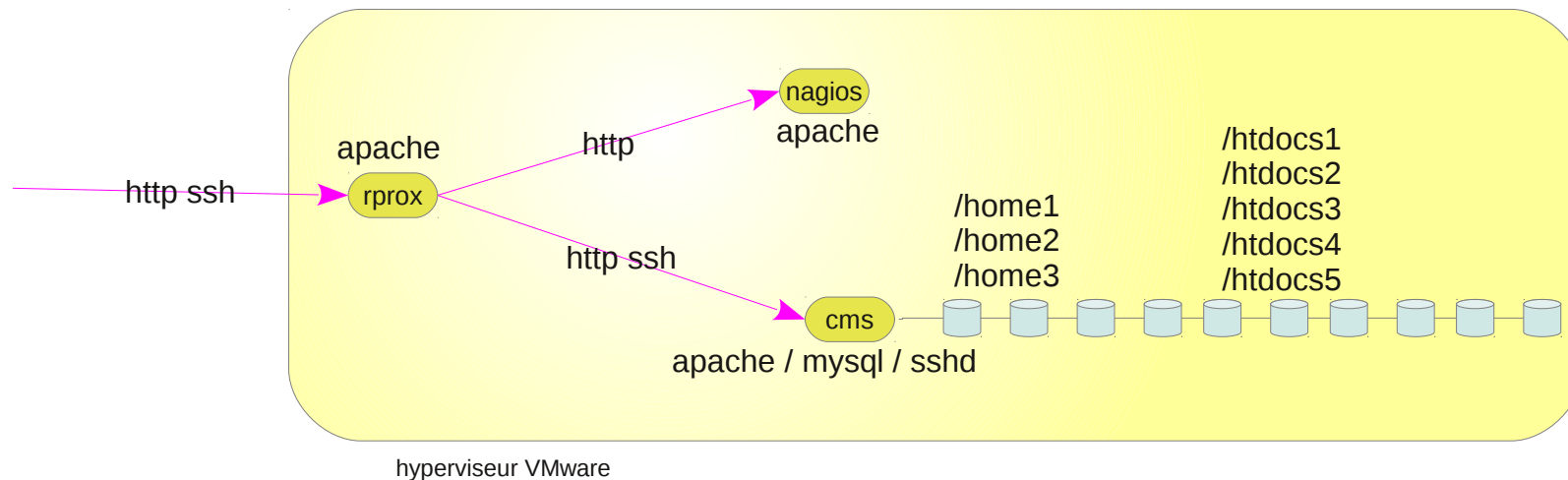
Architecture initiale

- 1 front-end + 1 back-end
- 1 site = 1 virtual host
- 10 disques virtuels de 4Go



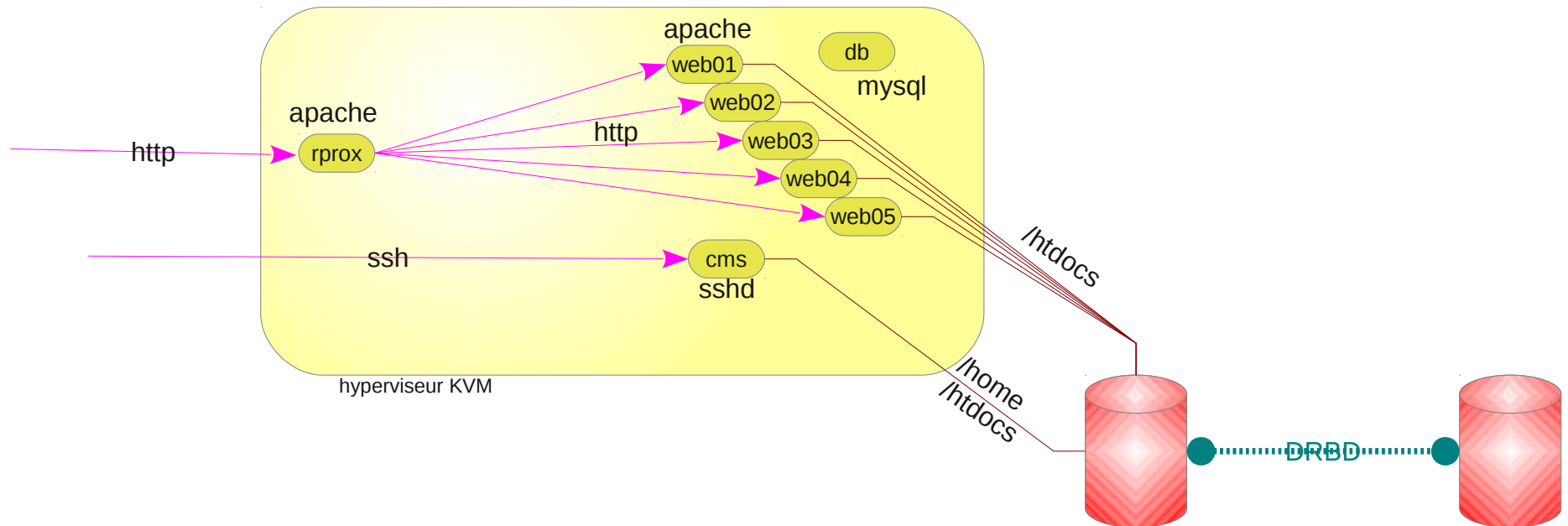
Architecture initiale

- monolithique : 1 site peut impacter la performance des autres
- gros disques virtuels (4Go) : lourd à gérer



Achitecture actuelle

- RAM de chaque webxx : 2Go
- granularité CPU / IO / RAM
- plusieurs versions possibles (php, etc.)

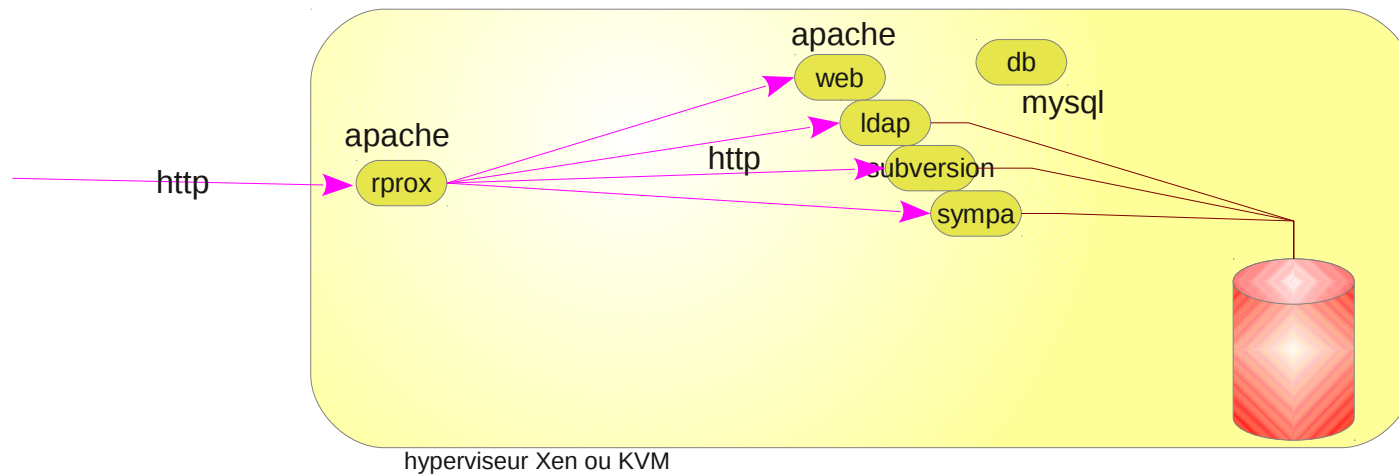


Le projet PLACO

Architecture de PLACO

- générateur de plates-formes collaboratives
- les services :
 - listes de diffusion (sympa)
 - environnement personnel (horde)
 - agenda, carnet d'adresse, webmail
 - hébergement web (apache)
 - partage réseau (apache webdav)
 - versionning (subversion)
- base d'authentification unique : OpenLDAP
- 2 hyperviseurs possibles : Xen et KVM
- 2 OS possible : Debian et CentOS

Plateforme collaborative générée



Mode d'emploi

- installer le générateur

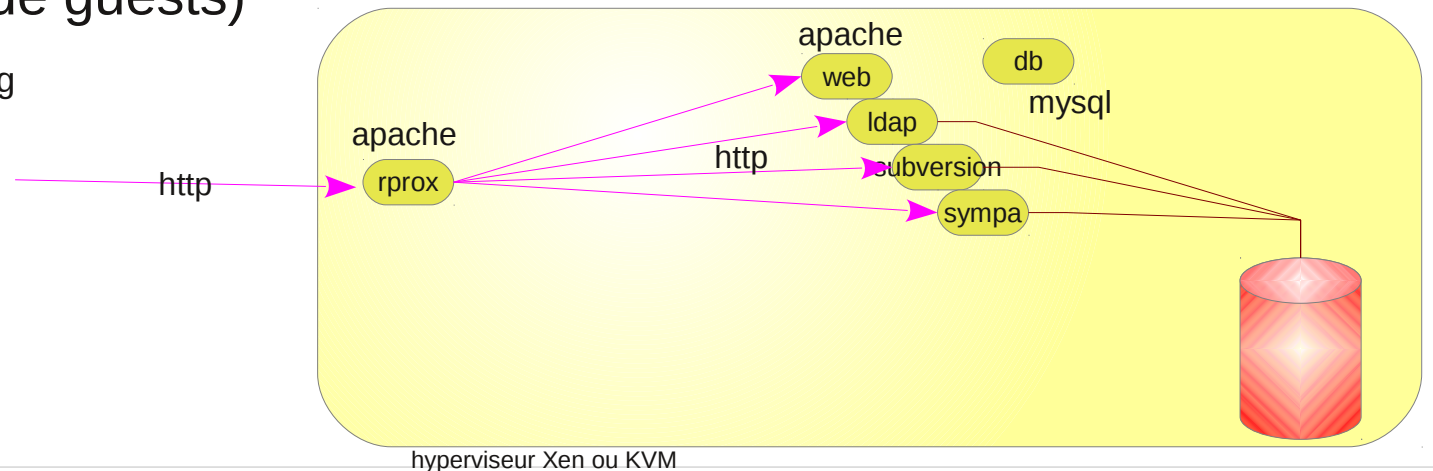
```
# wget --no-check-certificate -P /tmp https://svn.math.cnrs.fr/placodev/tags/stable/install.sh  
# ./tmp/install.sh
```

- créer une plate-forme minimale (annuaire+reverse proxy)

```
# placosh init_platform
```

- personnaliser (ajout de guests)

```
# placosh install_web_hosting  
# placosh install_sympa  
# placosh install_svn  
...
```



En savoir plus...

<http://placodev.mathrice.fr>