



# Système d'exploitation avancé

## Introduction à la virtualisation serveur x86

Pierre LEROY – pierre1.leroy@orange.com

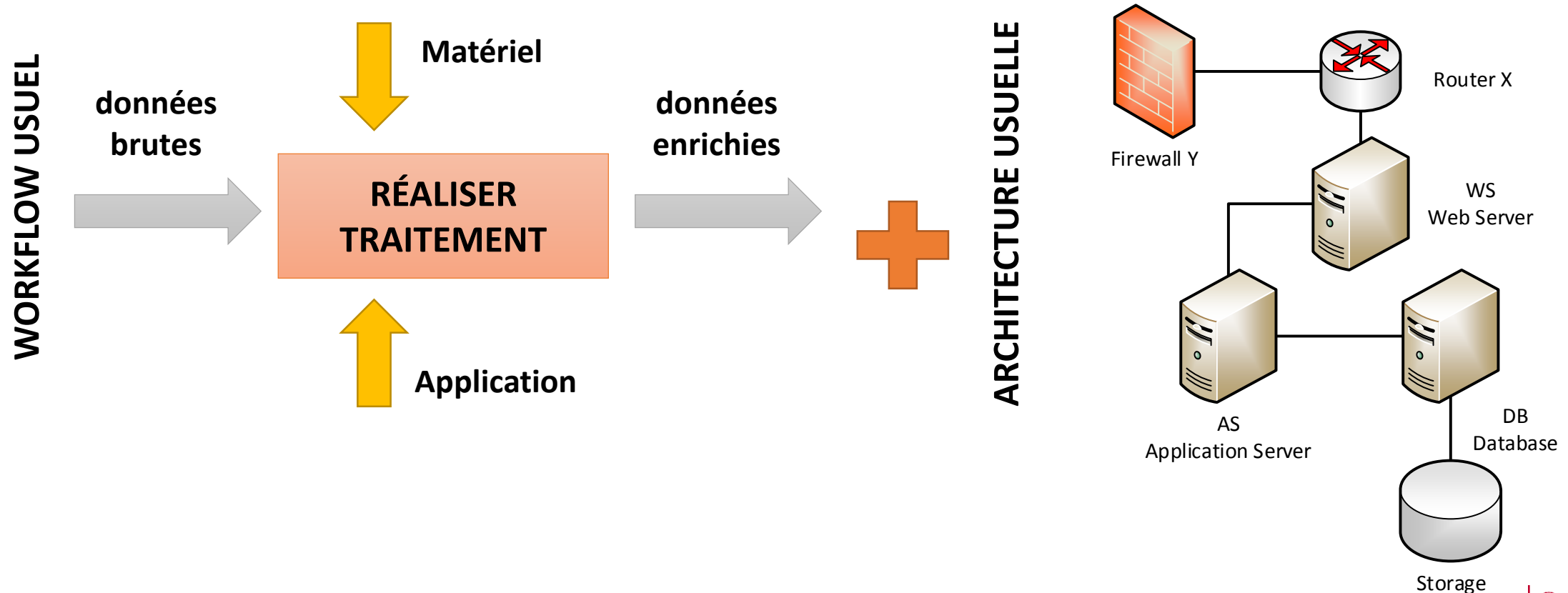
# Sommaire

---

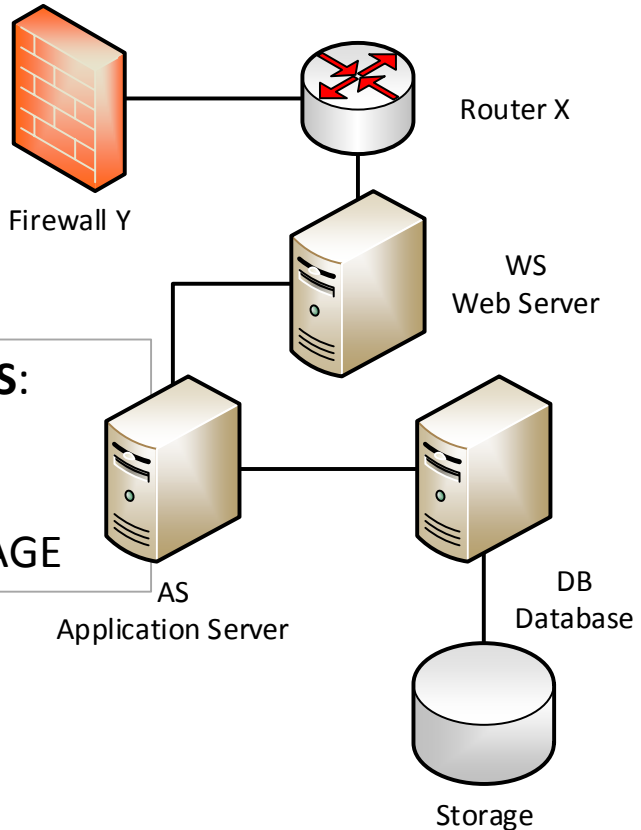
- I. Contexte & périmètre
- II. Mise en œuvre
- III. Aspect développement
- IV. Offre de service & perspectives d'évolution

# Constat

- Application LAMBDA :



# Constat



- Perte de ressources matérielles :
  - ✓ +50% de la puissance globale dédiée
- Pertes de ressources immatérielles :
  - ✓ Monétaires **OPEX & CAPEX**

# Constat

---

## TENDANCE

- Nombre d'applications en augmentation
- Ouverture aux évolutions d'architecture : déménagement / évolution projet
- Disposer d'environnements : développement / qualification / pré-production

## RESULTATS

- Hausse de la demande des ressources : CPU / RAM / Stockage
- Hausse des demandes liées à l'exploitation des applications
- Augmentation de la complexité technique :
  - ✓ IE: garantir les PRA associés; la HA haute disponibilité

## INCOVENIENTS

- Les coûts liés au provisioning et à l'exploitation des serveurs
- L'utilisation des ressources liés à l'environnement

# Constat

---

## ELEMENTS REPONSE

- Mutualisation des infrastructures
- Meilleur dimensionnement des serveurs livrés aux applications

## INCOVENIENTS SOUS-JACENTS

- L'isolation inter-application n'est pas garantie
  - ✓ Introduction de **SPOF** (Single Point Of Failure) :
    - Quid des changes/màj ?
  - ✓ Introduction de contentions liés aux ressources:
    - Utilisation 100% **CPU/RAM** par une seule application
    - Saturation d'un lien **ETHERNET/SAN**
  - Parades techniques possibles: positionnement **ulimit** etc.
- Les possibilités évolutions sont très complexes et peu flexibles

# Constat

---

## BESOIN CONSTATE

- Nécessité d'une technologique permettant :
  - ✓ La mutualisation des infrastructures (optimisation des ressources)
  - ✓ L'isolation des serveurs et ressources
  - ✓ Permettant la gestion des changes/màj
  - ✓ Une flexibilité d'évolution

 **LA VIRTUALISATION tente d'apporter une réponse cohérente**

# Définition

---

## VIRTUALISATION

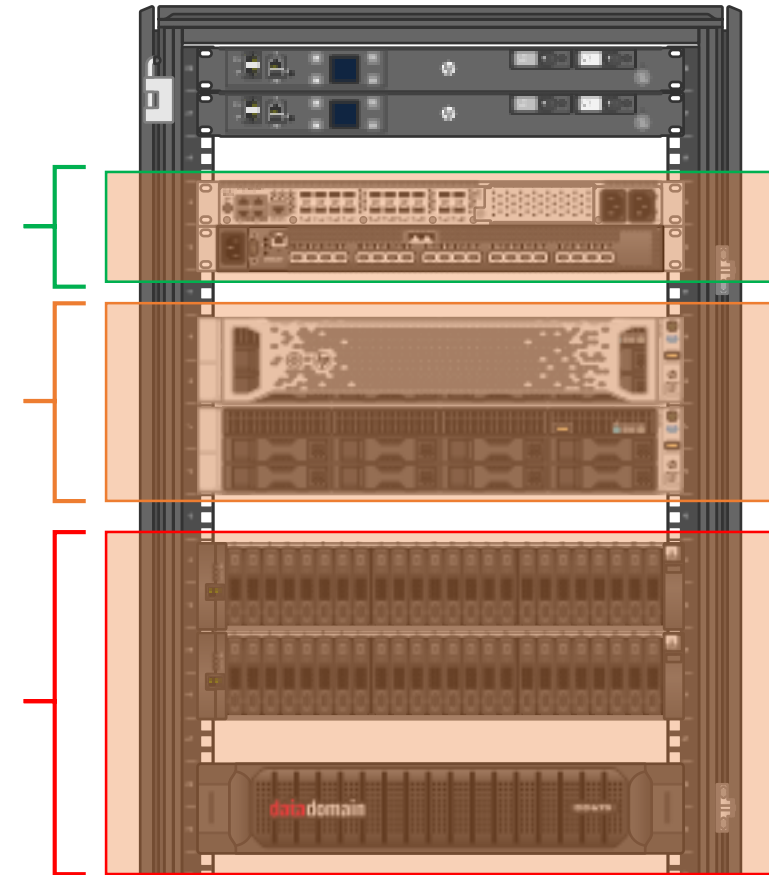
**Concept** reposant sur l'« **abstraction** » des objets (ou « choses »). Plutôt que de vous obliger à avoir une version physique réelle de chaque objet (ie: un serveur ou un périphérique de stockage), la technologie de *virtualisation* simule ces objets, vous permettant ainsi de créer des serveurs ou des périphériques de stockage « virtuels » au sein de votre configuration matérielle.



# Domaine d'application

## APPLICABILITE

- La virtualisation peu être implémentée :
  - ✓ Serveur
  - ✓ Réseau
  - ✓ Stockage
- Une partie conséquente des éléments physiques de l'infrastructure est éligible à la virtualisation





# Questions ?

# Sommaire

---

- I. Contexte & périmètre
  - II. Mise en œuvre
  - III. Aspect développement
  - IV. Offre de service & perspectives d'évolution
-

# Définitions

---

## VM - Machine Virtuelle

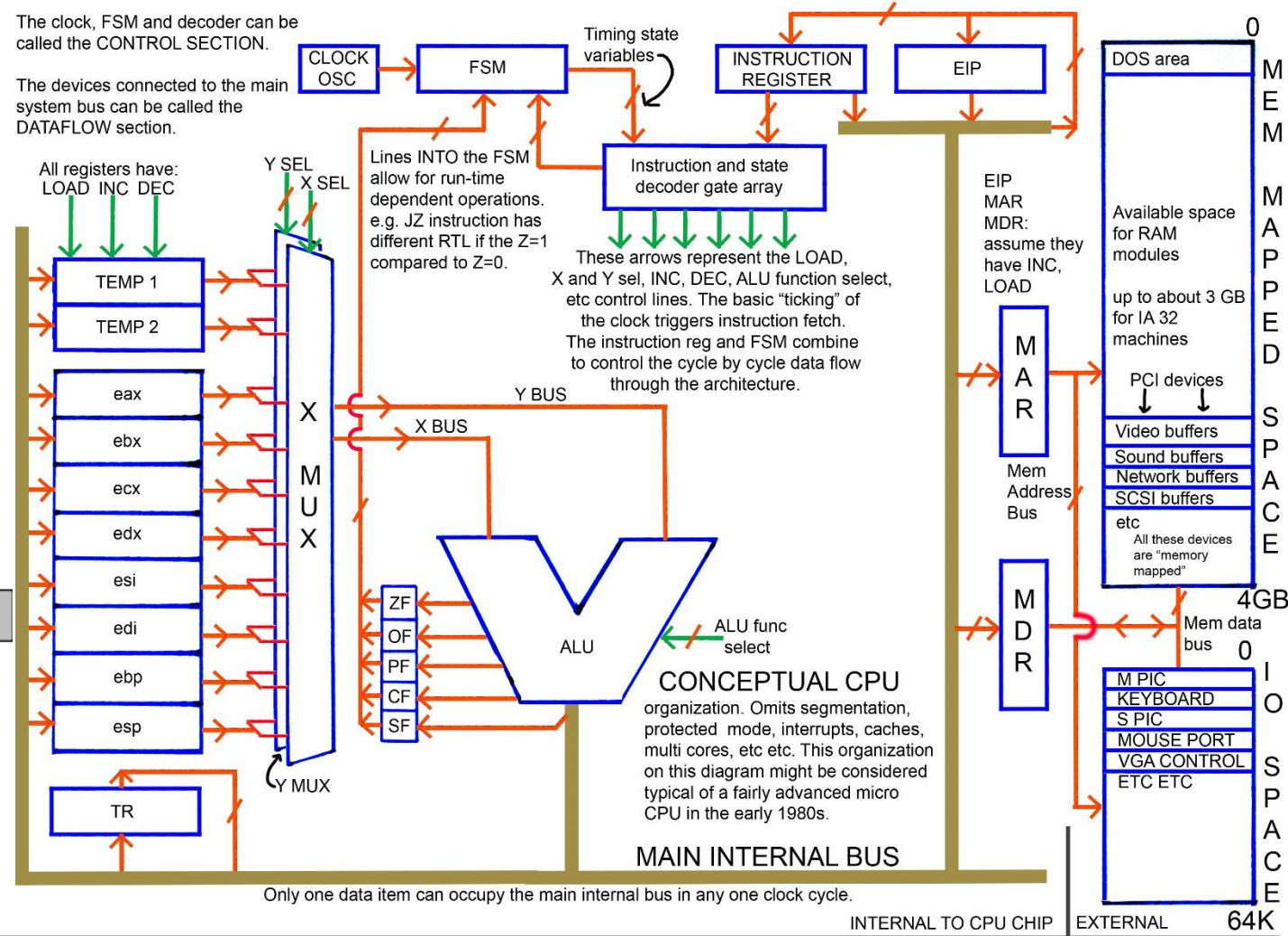
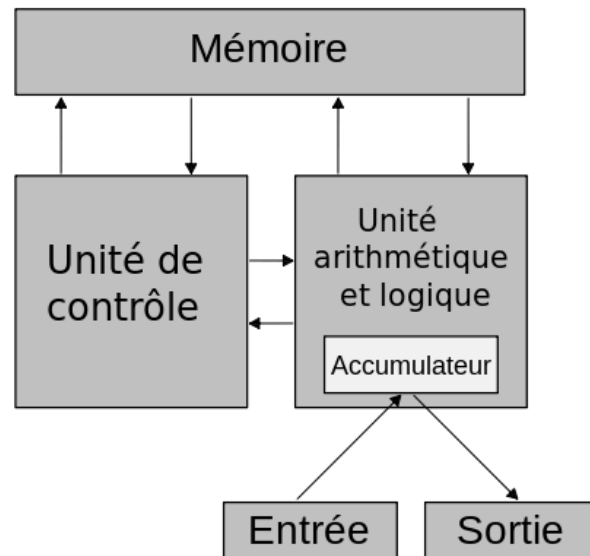
Un item donnant l'illusion d'un appareil informatique créée par un logiciel d'émulation appelé hyperviseur. L'intérêt premier de la machines virtuelle réside dans le fait de pouvoir s'abstraire des caractéristiques de la machine physique utilisée.

## Hyperviseur

L'hyperviseur simule la présence de ressources matérielles et logicielles telles que la **mémoire**, le **processeur**, le **disque dur**, le **système d'exploitation** et les pilotes, permettant d'exécuter des programmes dans les mêmes conditions (architecture) que celles de la machine simulée. Il existe deux principaux types d'hyperviseurs appelés type « 1 » et « 2 ».

# Définitions

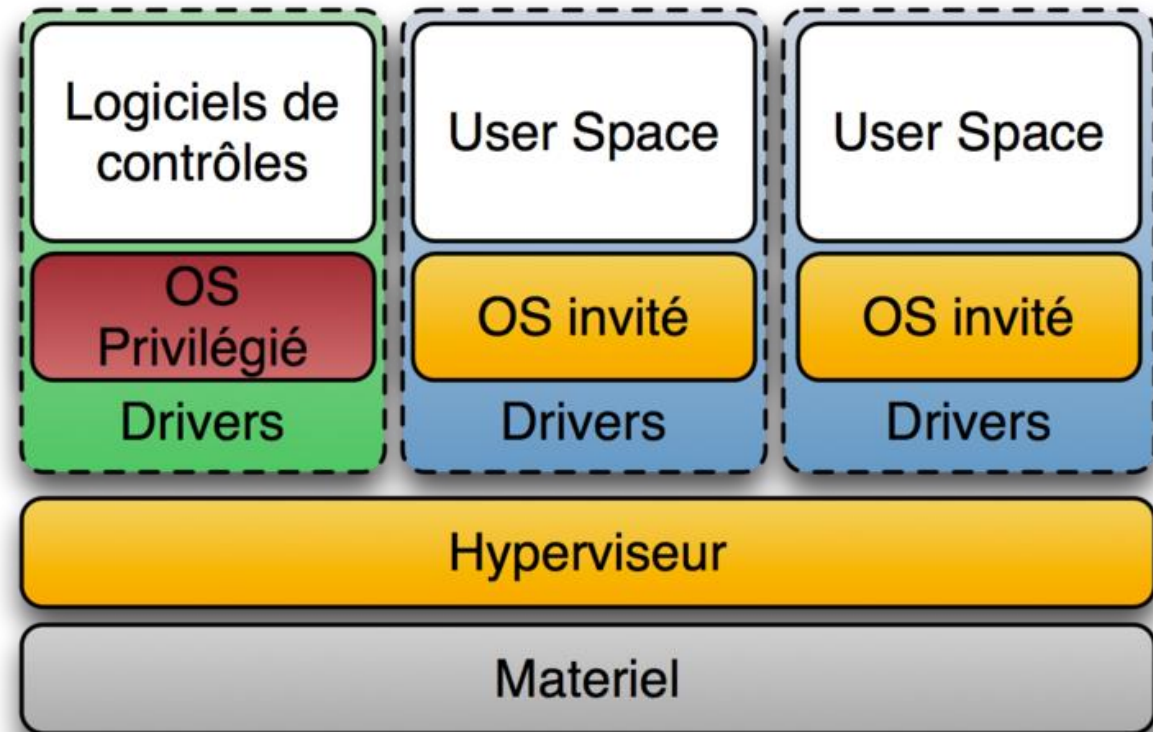
## Emulation



# Définitions

## Hyperviseur TYPE 1

- Exécution sur le matériel
- Noyau dédié à l'usage
  - ✓ allégé & optimisé
- Présence d'«overhead»
- Principaux *vendors* actuels:



# Définitions

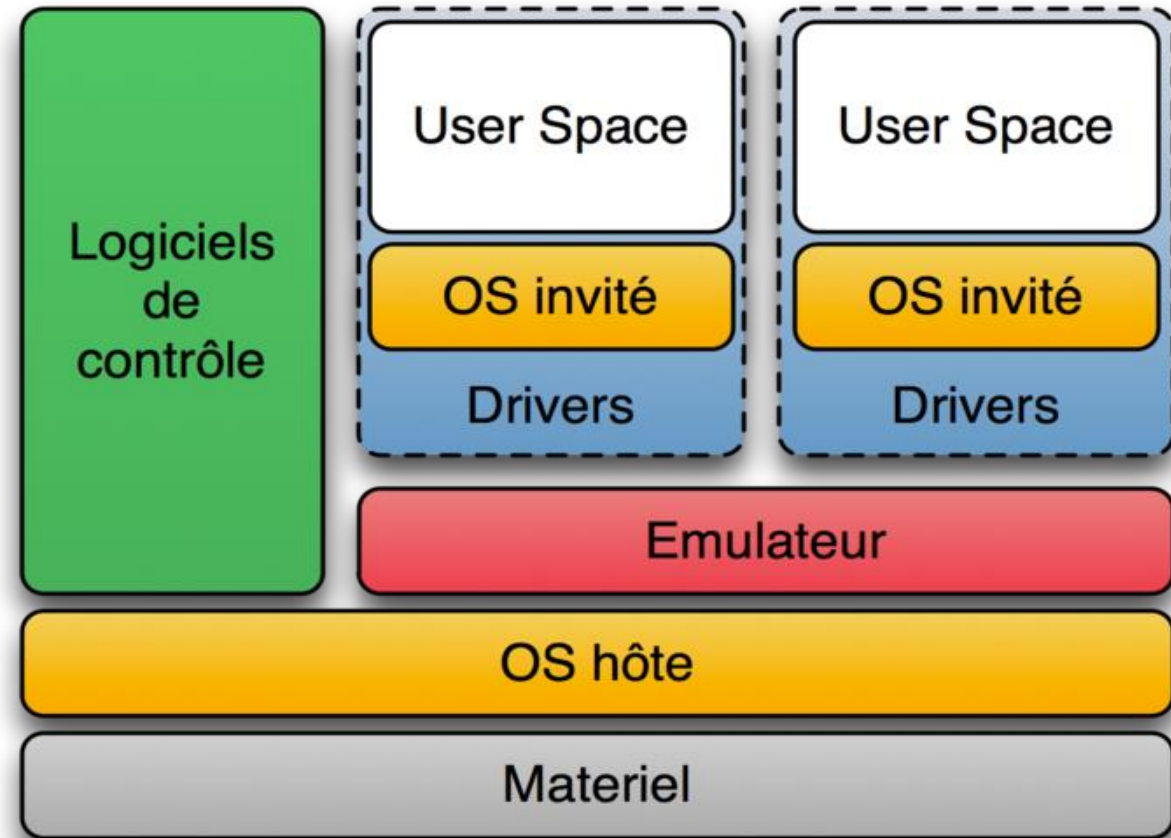
## Hyperviseur TYPE 2

- Exécution sur un OS préalable
- Présence d'une brique logicielle d'émulation
- Performance en retrait /TYPE1
- Principaux *vendors* actuels:



VirtualBox

Parallels™

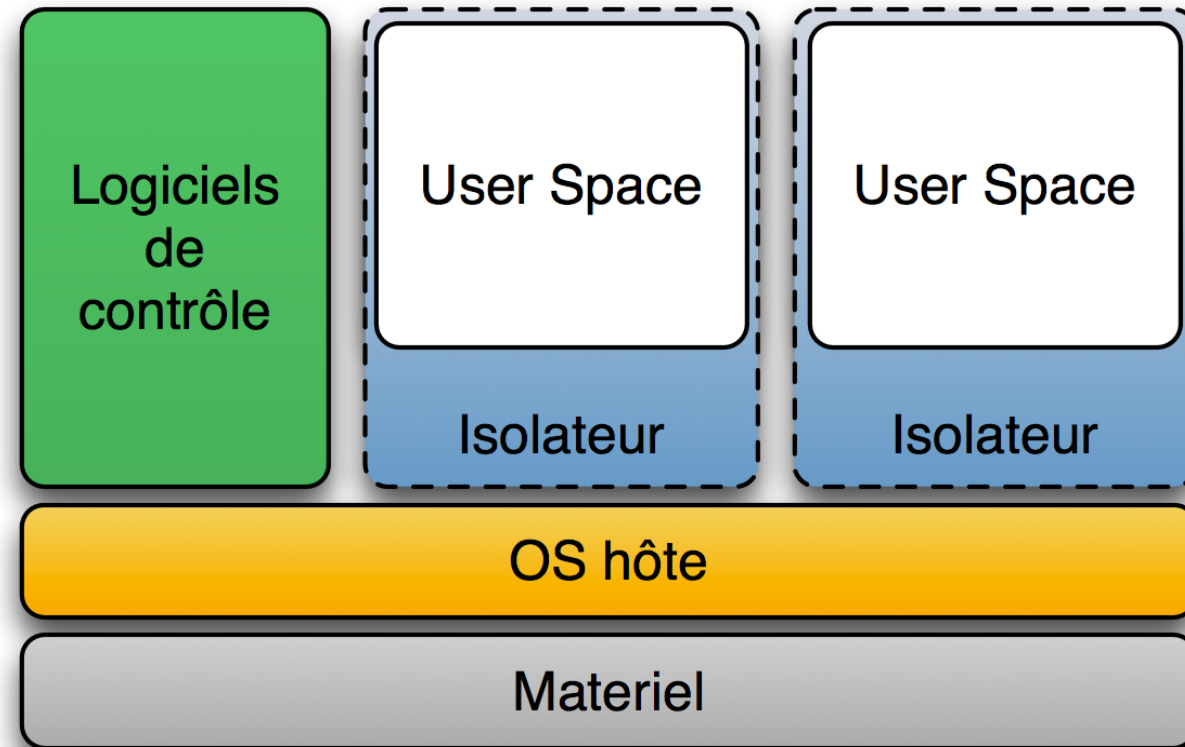
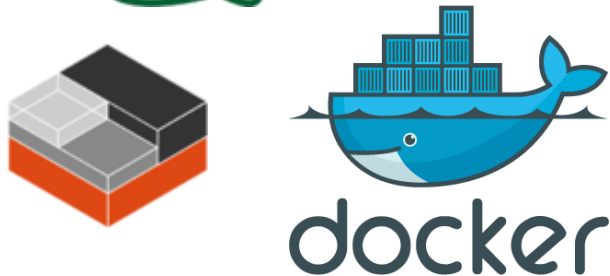




# Définitions

## Isolateur

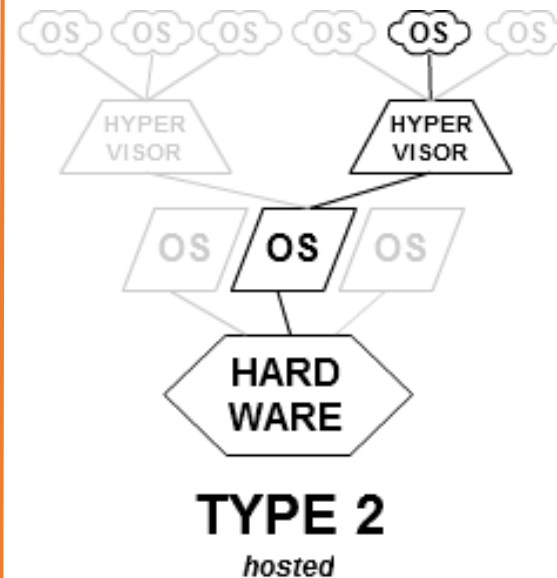
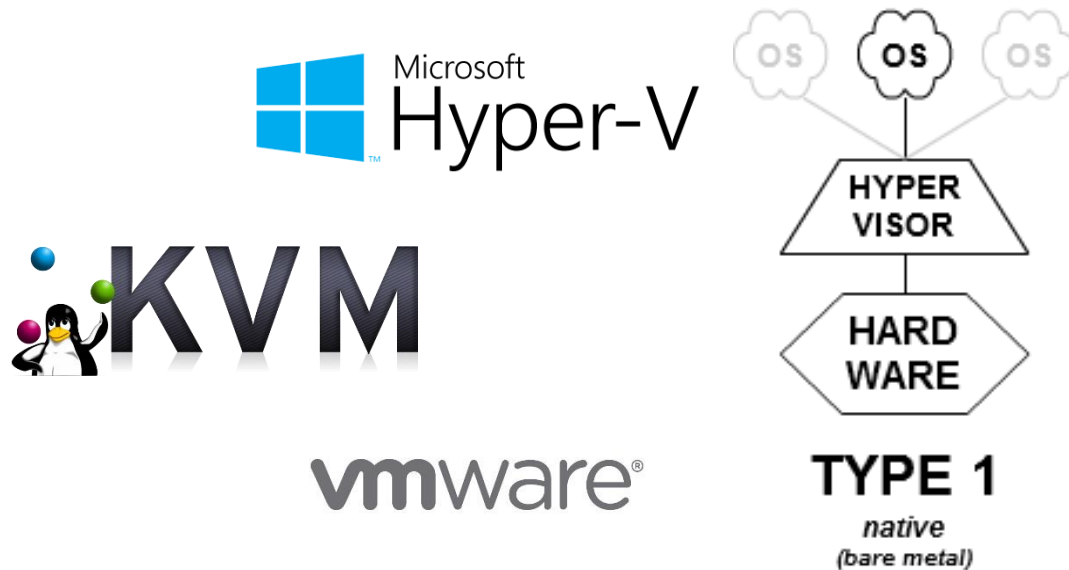
- Isolation « OS »
  - ✓ Isolation noyau
  - ✓ Performance
  - ✓ Flexibilité
- Principales solutions actuelles:





# Définitions

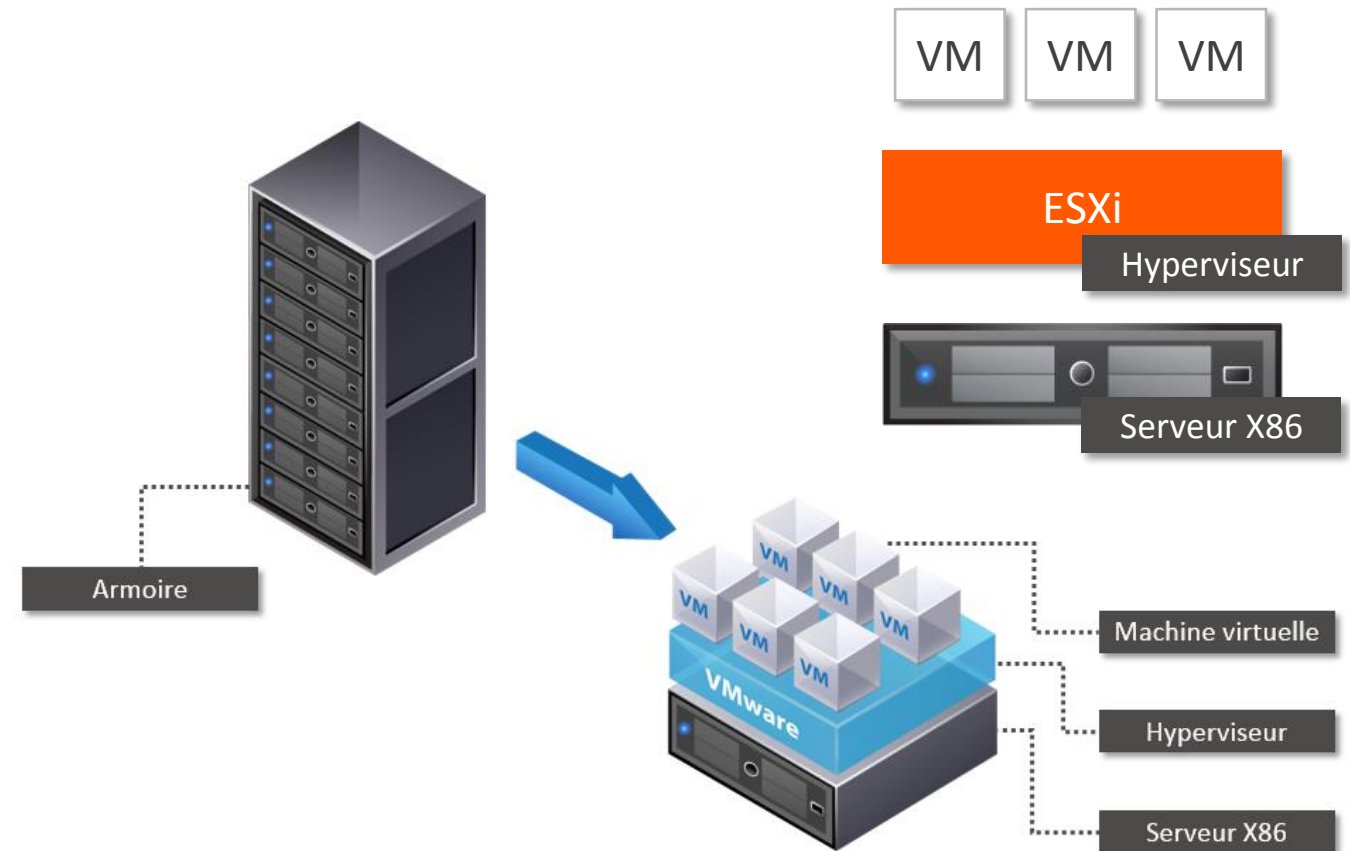
## Hyperviseur

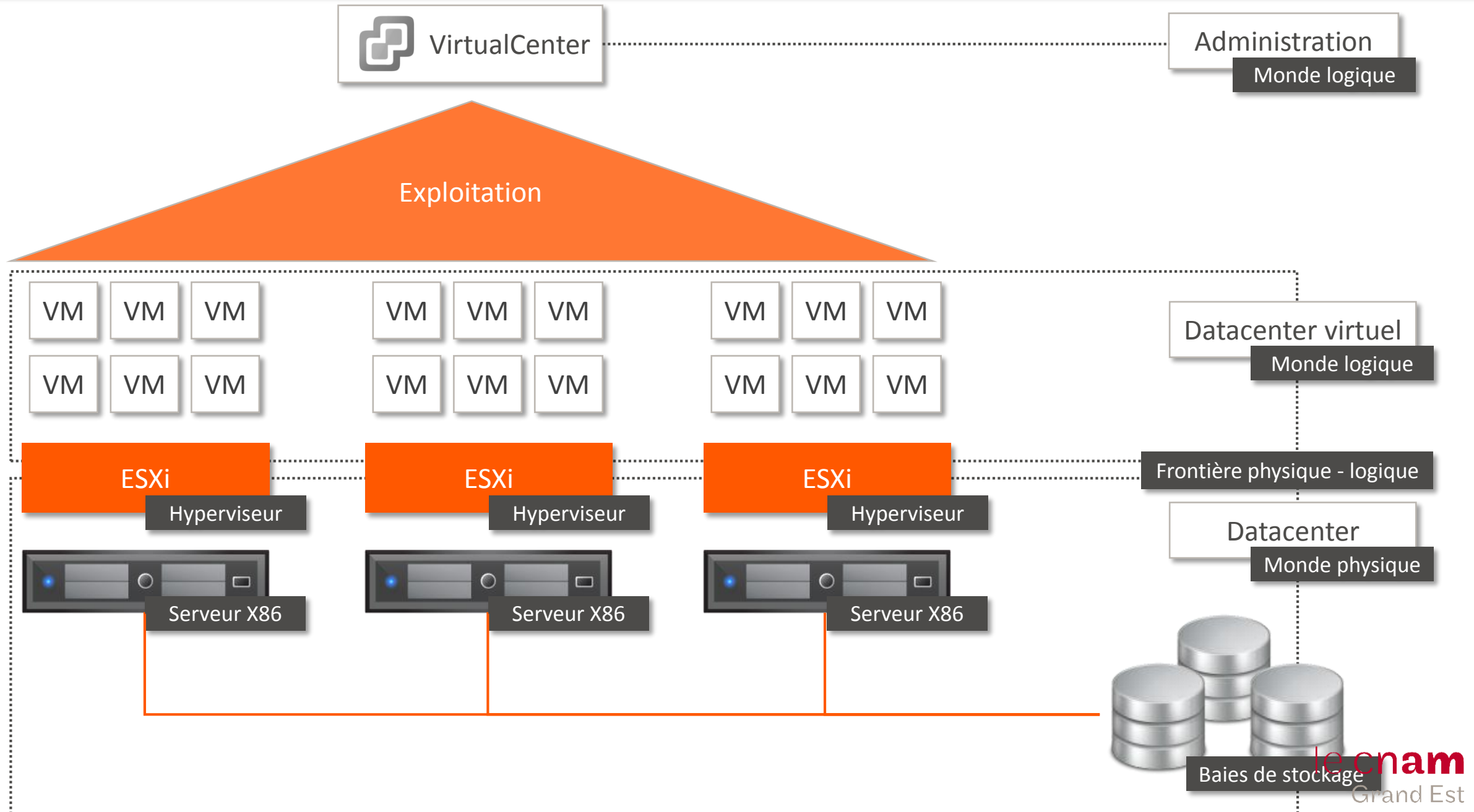


# Mise en pratique **vmware®**

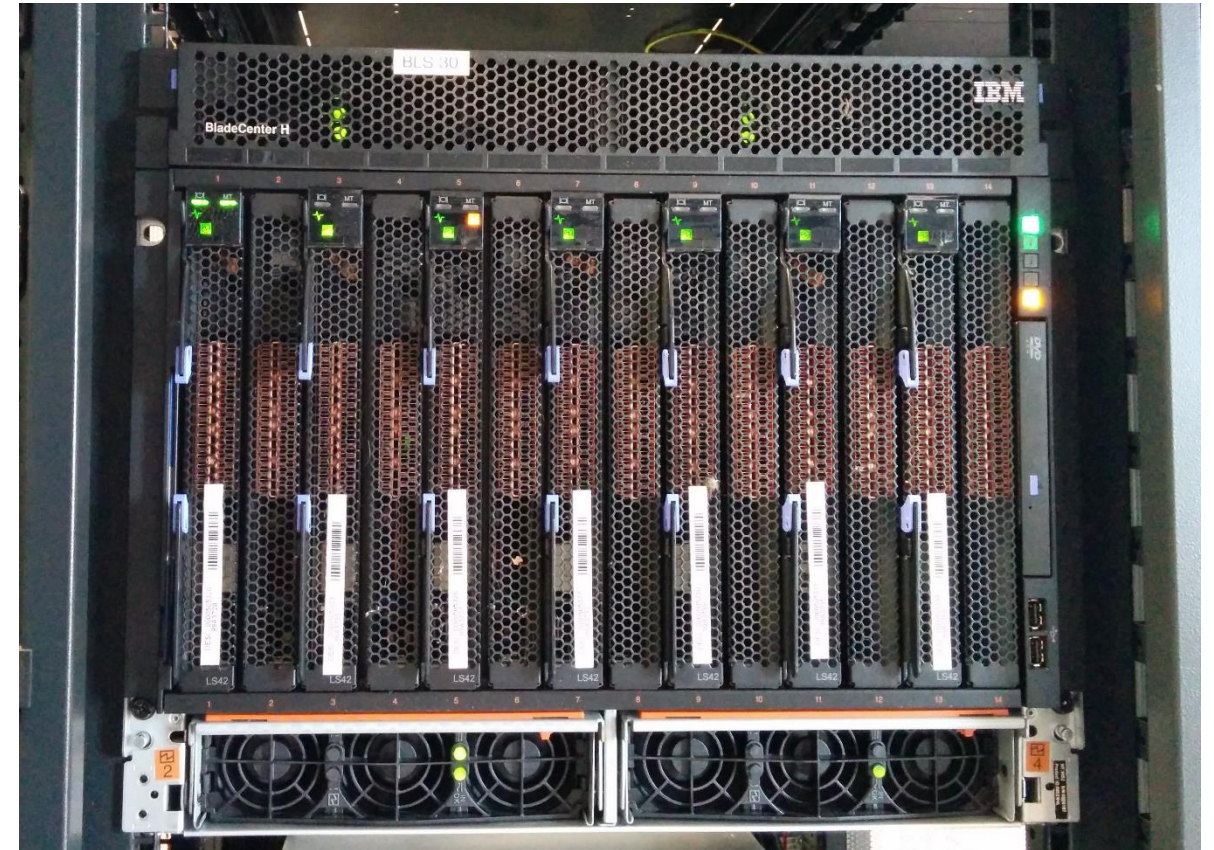
## Hyperviseur TYPE 1

- Virtualisation depuis 1999
- Suite logicielle vSphere
- Implémente les mécanismes:
  - ✓ HA – Haute Disponibilité
  - ✓ DRS – Répartition de charge

























# Mise en pratique

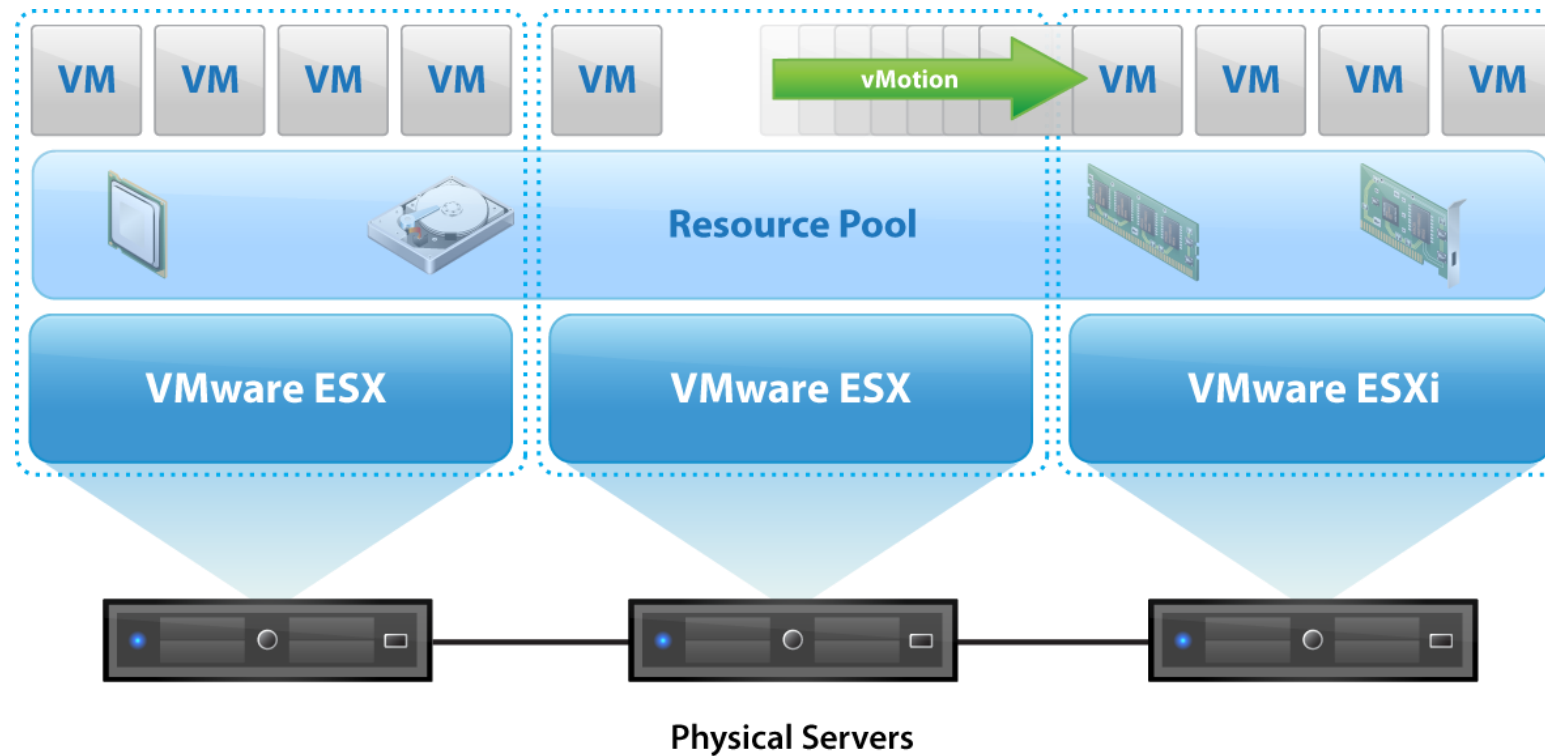
## Implémentation d'une VM

- Structuration formalisée:
  - ✓ Enveloppe VM: description de la configuration CPU/RAM/Stockage
  - ✓ Représentation BIOS
  - ✓ Représentation RAM
  - ✓ Fichier d'échange SWAP
  - ✓ Fichier(s) image disque(s)
  - ✓ + Fichiers liés aux snapshots
- Standards d'enveloppe existants -> OVF

 CentOS_IVdev.vmsd	VMware snapshot metadata
 CentOS_IVdev.vmx	VMware Team Member
 CentOS_IVdev.vmdk	VMware virtual disk file
 CentOS_IVdev-s001.vmdk	VMware virtual disk file
 CentOS_IVdev-s002.vmdk	VMware virtual disk file
 CentOS_IVdev-s003.vmdk	VMware virtual disk file
 CentOS_IVdev-s004.vmdk	VMware virtual disk file
 CentOS_IVdev-s005.vmdk	VMware virtual disk file
 CentOS_IVdev-s006.vmdk	VMware virtual disk file
 CentOS_IVdev-s007.vmdk	VMware virtual disk file
 CentOS_IVdev-s008.vmdk	VMware virtual disk file
 CentOS_IVdev-s009.vmdk	VMware virtual disk file
 CentOS_IVdev-s010.vmdk	VMware virtual disk file
 CentOS_IVdev-s011.vmdk	VMware virtual disk file
 CentOS_IVdev.vmx	VMware virtual machine configuration
 CentOS_IVdev.nvram	VMware Virtual Machine nonvolatile RAM

# Mise en pratique **vmware®**

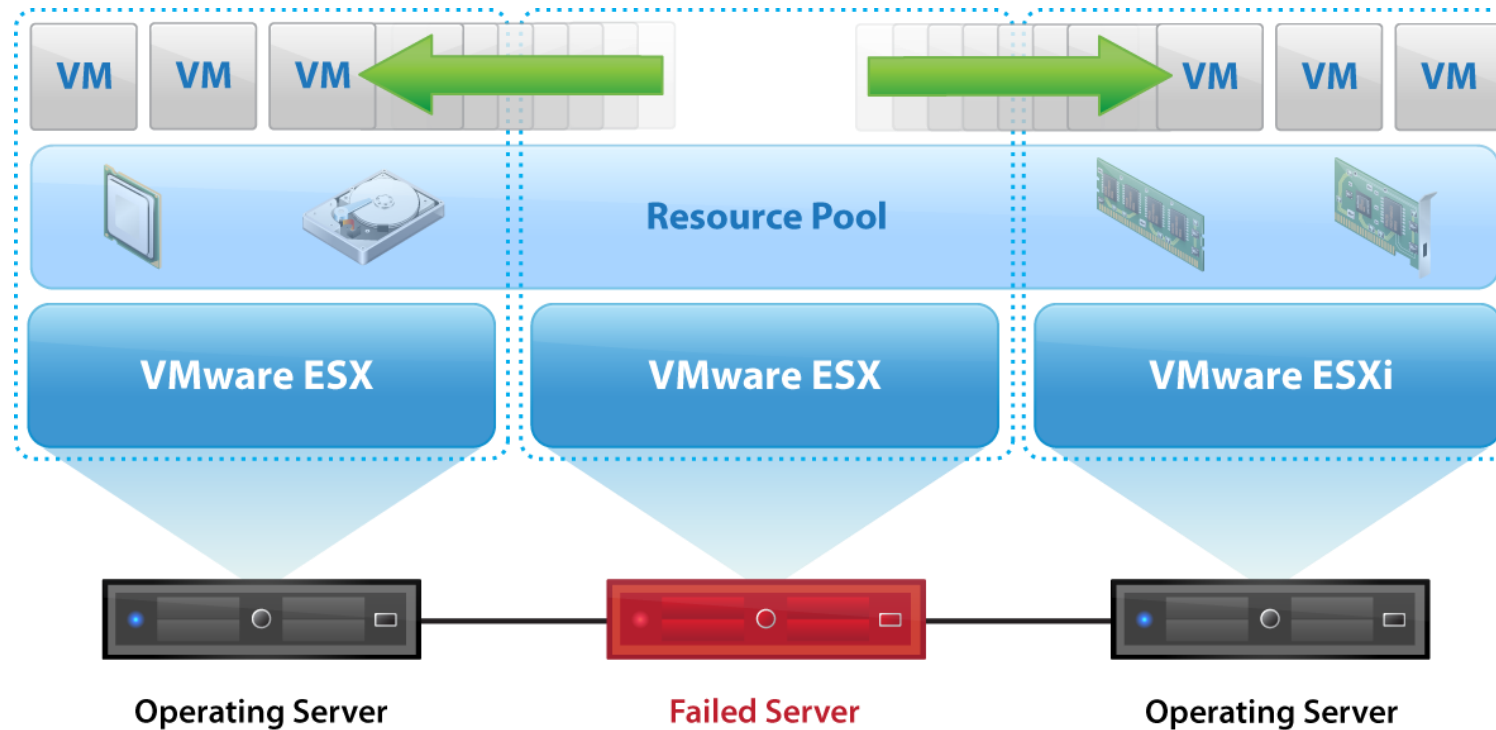
## DRS – Distributed Resources Scheduling





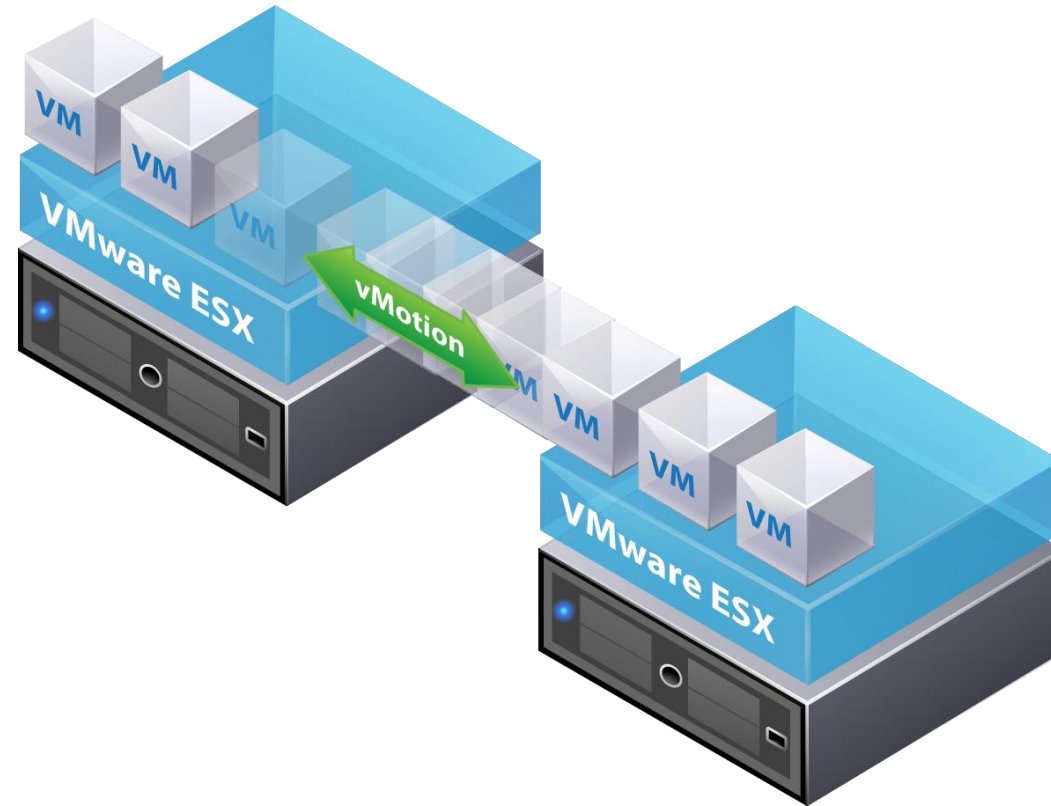
# Mise en pratique **vmware®**

## HA – High Availability



# Mise en pratique **vmware®**

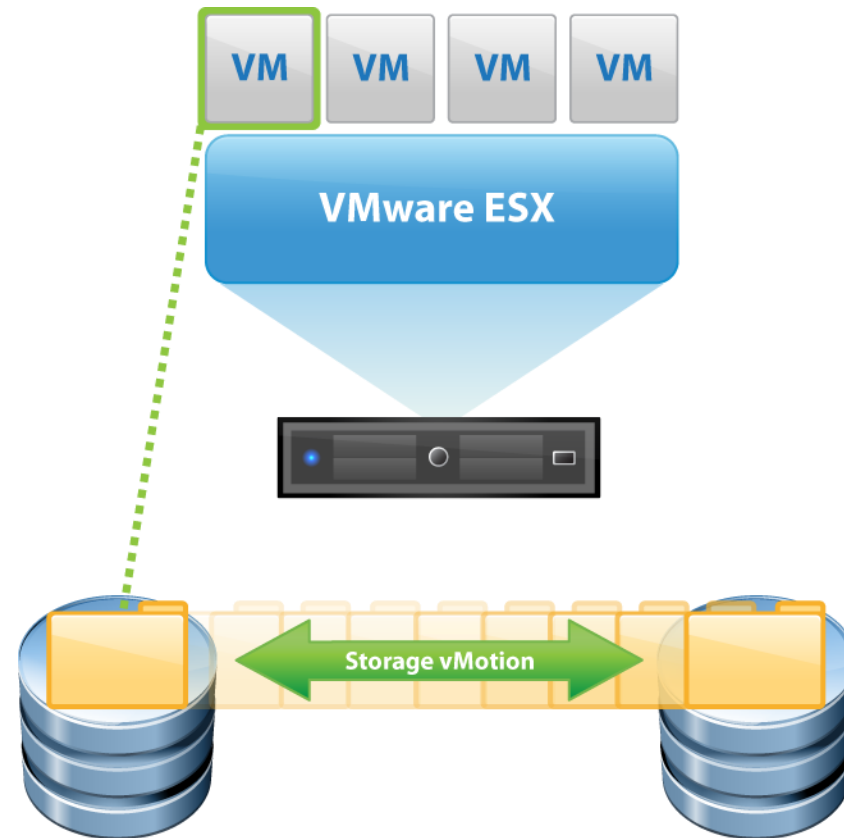
## vMotion





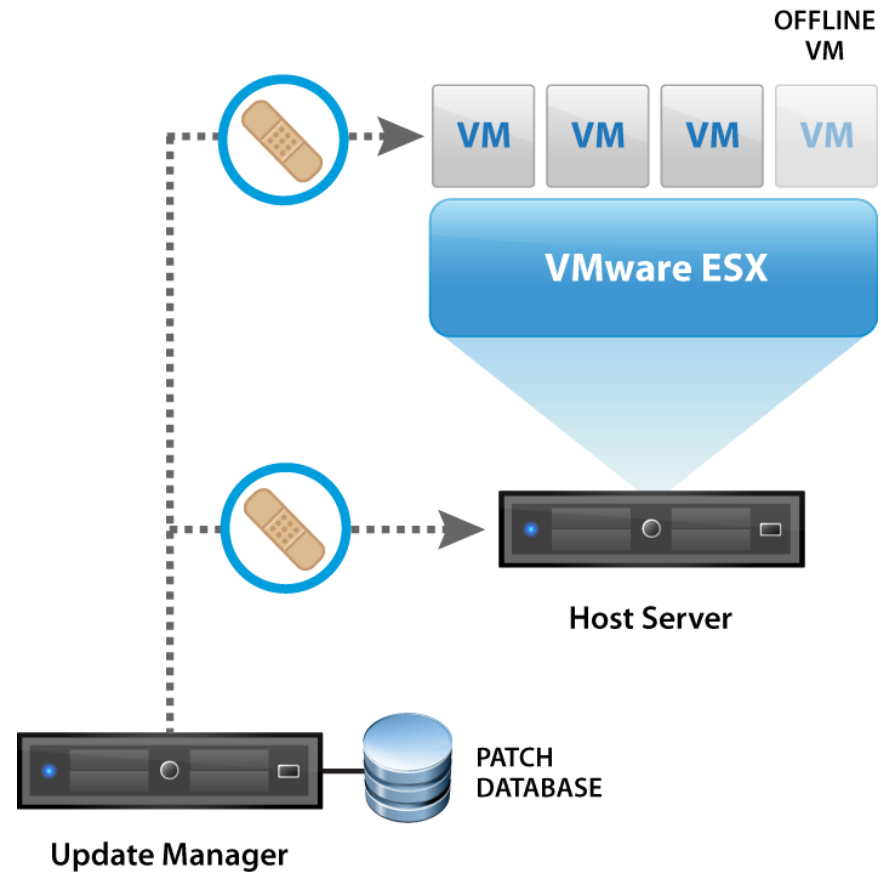
# Mise en pratique **vmware®**

## Storage vMotion



# Mise en pratique **vmware®**

## Update Manager



# Mise en pratique

---

## Démonstration

- Mise en place d'un conteneur et d'une VM
- Illustration des mécanismes d'isolation
- Représentation des objets au sein de la solution

# Mise en pratique

---

## Démonstration

- Mise en place d'un conteneur et d'une VM
- Illustration des mécanismes d'isolation
- Représentation des objets au sein de la solution

# Implémentation

---

## Isolation

- Mécanisme basé sur l'implémentation des **cgroups**
  - ✓ Fonctionnalité du noyau linux kernel > v2.6.24
- Les cgroups fournissent une réponse permettant de réaliser:
  - ✓ Limitations + priorités sur les ressources **CPU / RAM / stockage / réseau**
  - ✓ Isolation par **NAMESPACE**
  - ✓ Isolation complete : **processus / réseau / filesystems**

- [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/html/Resource\\_Management\\_Guide/index.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Resource_Management_Guide/index.html)
- <https://wiki.archlinux.org/index.php/cgroups>



# Questions ?

# Sommaire

---

- I. Contexte & périmètre
  - II. Mise en œuvre
  - III. Aspect développement
  - IV. Offre de service & perspectives d'évolution
-

# Besoin applicatifs

---

## EXPLOITATION

- La virtualisation implique des besoins applicatifs sous-jacents du fait:
  - ✓ De la potentielle complexité de l'environnement associé
  - ✓ Des tâches d'exploitation à réaliser
  - ✓ Du volume croissant à exploiter
  
- En contrepartie la plus part des solutions proposent des API pour développer
  - ✓ Scripts d'automatisation
  - ✓ Portail self-care exploitant
  - ✓ ...

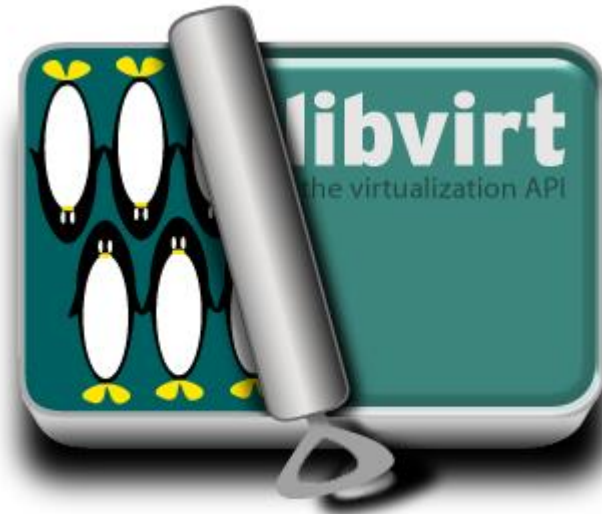


# Intégration

---

## API

- Catalogue d'API dédié à l'interfaçage application <> *solution vendor* :
  - ✓ Libvirt
  - ✓ Vjjava
  - ✓ PowerCLI
  - ✓ ...



# Plateforme

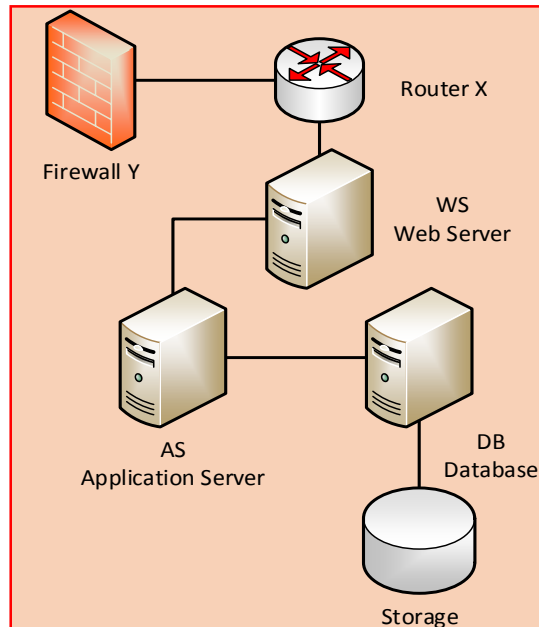
## ENVIRONNEMENT

- Disposer d'environnements : développement / qualification / production

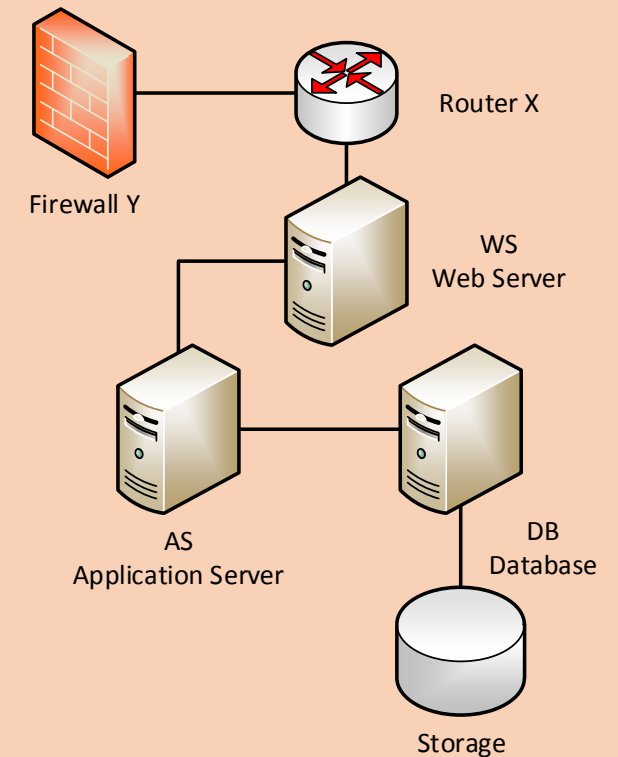
OUTILS DEVELOPPEMENT



QUALIFICATION



PRODUCTION



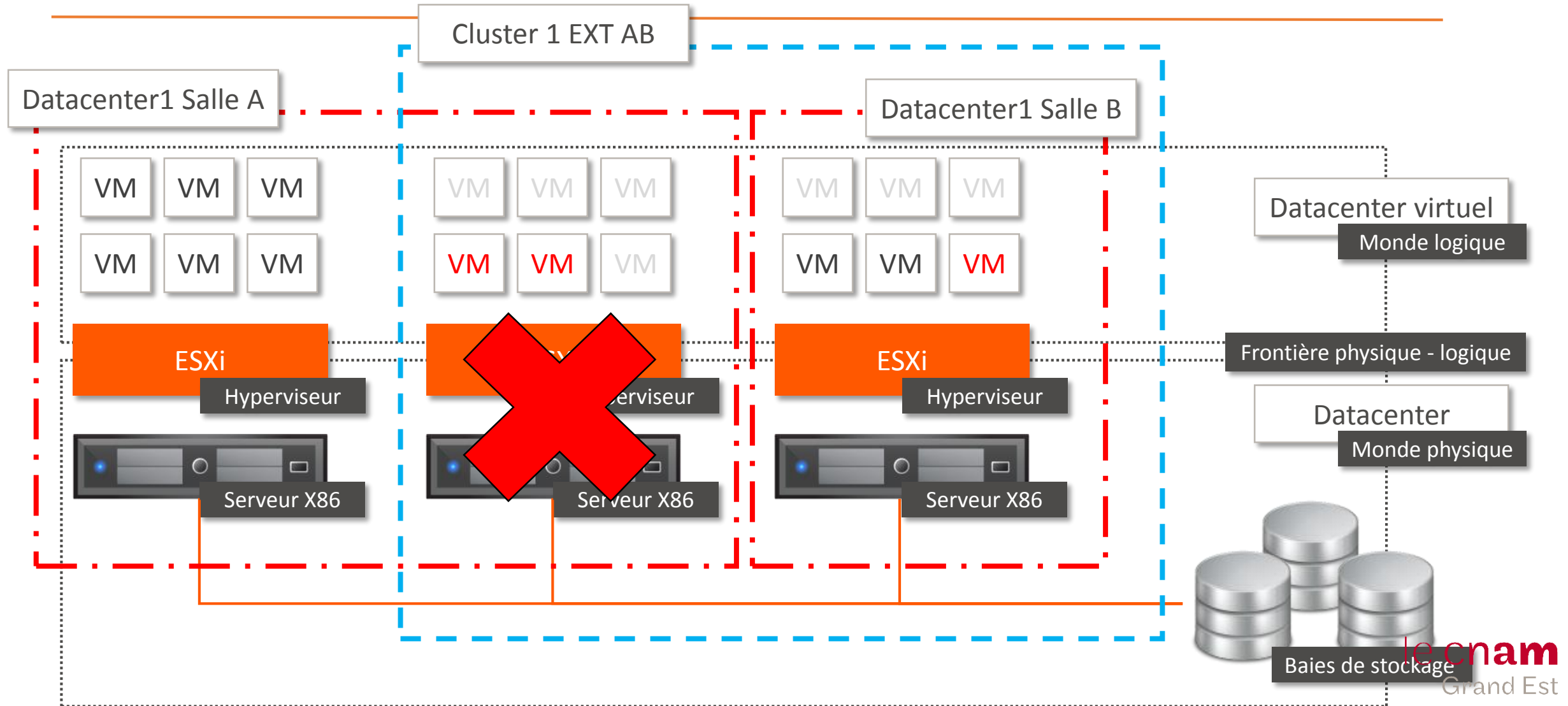
# Architecture

---

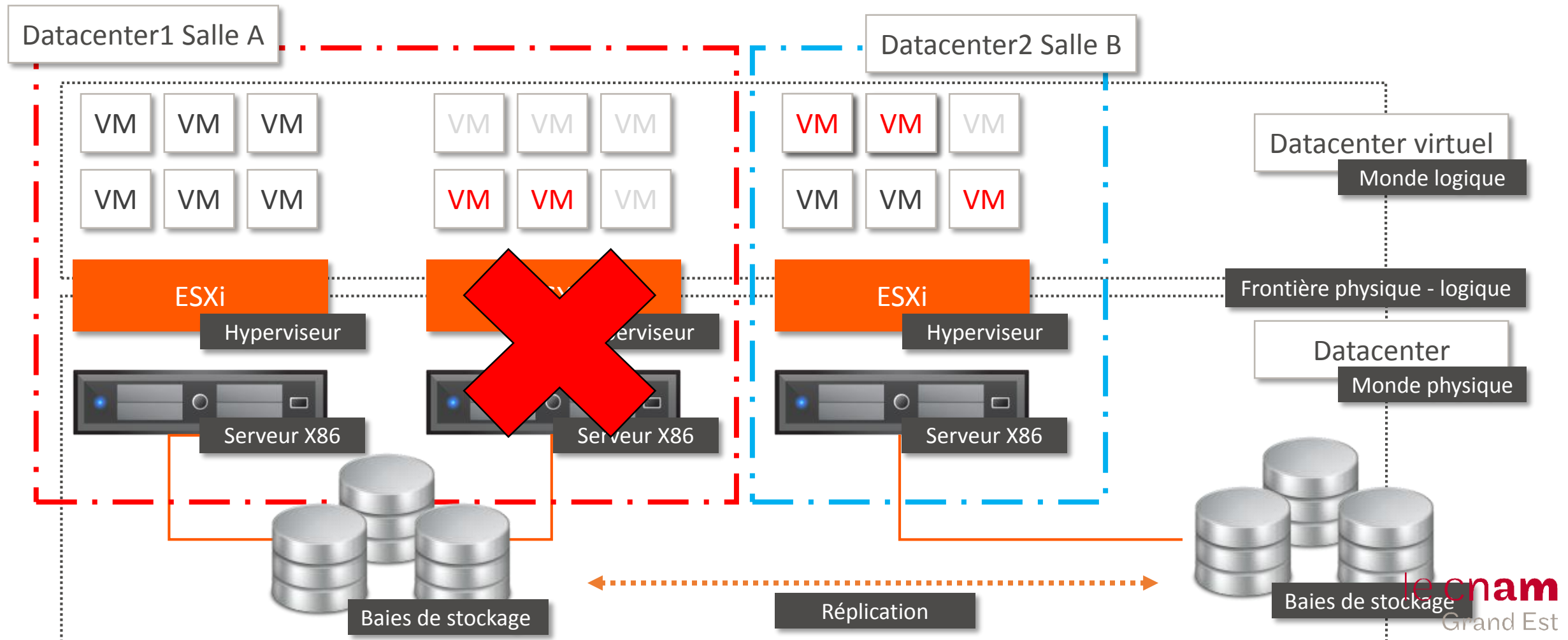
## ENVIRONNEMENT

- Simplification des déploiement applicatifs
- Diminution de la complexité de déploiement des applications « sensibles »:
  - Inter-salle DC ;
  - Extra DC;
- Diminution de la complexité des architectures soumises à PRA :
  - Basculement de salle;
  - Basculement de DataCenter;
- Focus sur les équipements connexes : Loadbalancer / baie de stockage etc.

# Architecture



# Architecture





# Questions ?

# Sommaire

---

- I. Contexte & périmètre
- II. Mise en œuvre
- III. Aspect développement
- IV. Offre de service & perspectives d'évolution

# Cloud computing

---

## Définition

Ou l'**informatique en nuage** ou **nuagique** ou encore l'**infonuagique** (au Québec), est l'exploitation de la puissance de calcul ou de stockage de serveurs informatiques distants par l'intermédiaire d'un réseau, généralement internet.

## Principes

- **Ressources en libre-service** : et adaptation automatique à la demande. La capacité de stockage et la puissance de calcul sont adaptées automatiquement au besoin d'un consommateur. La demande est automatique et la réponse est immédiate.
- **Ouverture** : les services sont mis à disposition sur l'Internet, et utilisent des techniques standardisées qui permettent de s'en servir sur différents supports.
- **Mutualisation** : agrégation des ressources hétérogènes (matériel, logiciel, trafic réseau) en vue de servir plusieurs consommateurs à qui les ressources sont automatiquement attribuées.
- **Paieement à l'usage** : la quantité de service consommée dans le *cloud* est mesurée, à des fins de contrôle, d'adaptation des moyens techniques et de facturation.



# Offre [X]aaS

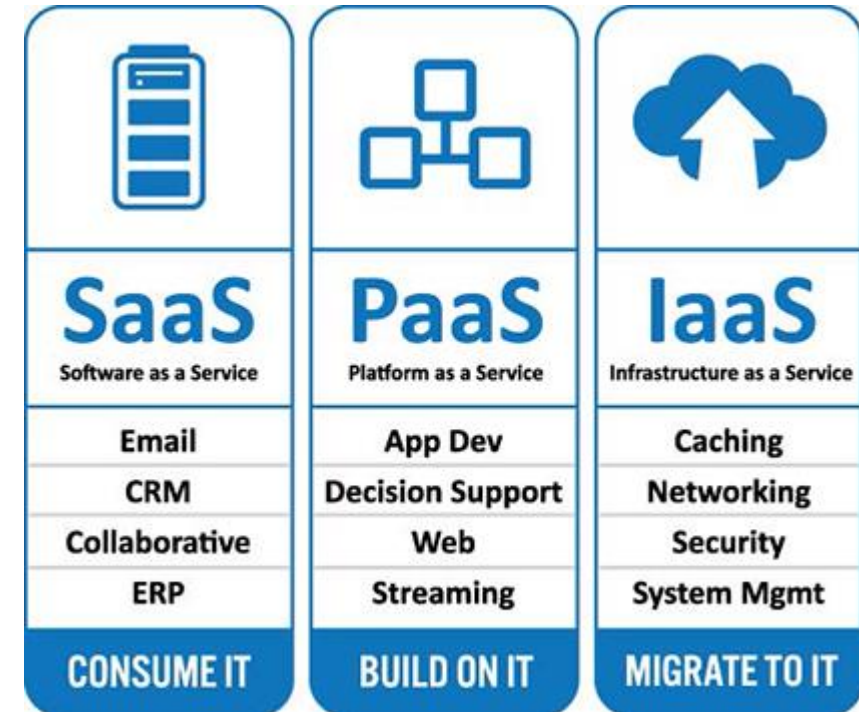
## Implémentation

- L'avènement du cloud computing se concrétise par le biais d'offres de service:

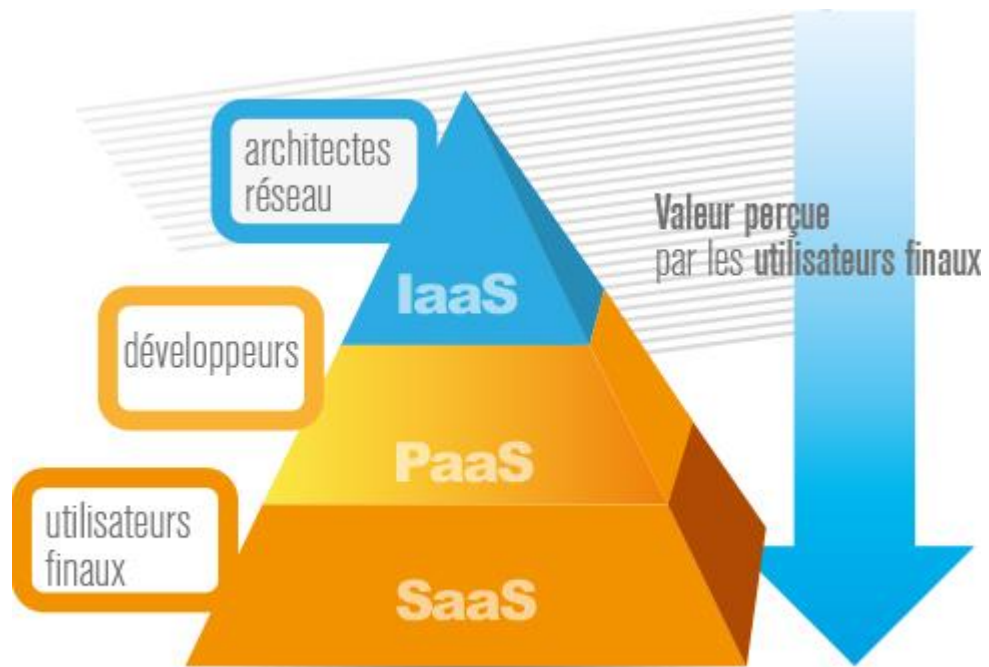
- ✓ SaaS: **Software** as a Service
- ✓ PaaS: **Platform** as a Service
- ✓ IaaS : **Infrastructure** as a Service

- Ces dernières sont destinées à des usages différents et impliquent:

- ✓ Des **responsabilités** différentes
- ✓ Des **connaissances** différentes



# Offre [X]aaS



Service Models	Cloud Stack	Stack Components	Who is Responsible		
SAAS PAAS IAAS	User	<div>Login</div> <div>Registration</div> <div>Administration</div>	Customer	Customer	Customer
	Application	<div>Authentication</div> <div>User Interface</div> <div>Reports</div> <div>Authorization</div> <div>Transactions</div> <div>Dashboard</div>			
	Application Stack	<div>LAMP (Linux, Apache, MySQL, PHP/Perl/Python)</div> <div>redis</div> <div>RabbitMQ</div> <div>mongoDB</div> <div>New Relic</div>	Vendor	Vendor	Vendor
	Infrastructure	<div>Data Center</div> <div>Servers</div> <div>Network</div> <div>Disk Storage</div> <div>Firewall</div> <div>Load Balancer</div>			

# Perspectives

---

- Desktop as a service (DaaS)
  - ✓ « bureau virtuel » ou « bureau virtuel hébergé » est l'externalisation d'une virtual desktop infrastructure auprès d'un fournisseur de services.
- Network as a service (NaaS)
  - ✓ Correspond à la fourniture de services réseaux, suivant le concept de software defined networking (SDN).
- Storage as a service (STaaS)
  - ✓ Fourniture d'offre de service de stockage.

# Vendors « cloud »

---

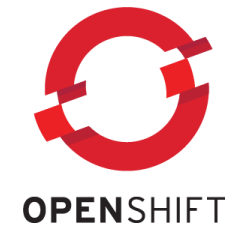
## IaaS



## SaaS



## PaaS





# Questions ?



# Conclusion



# Annexes