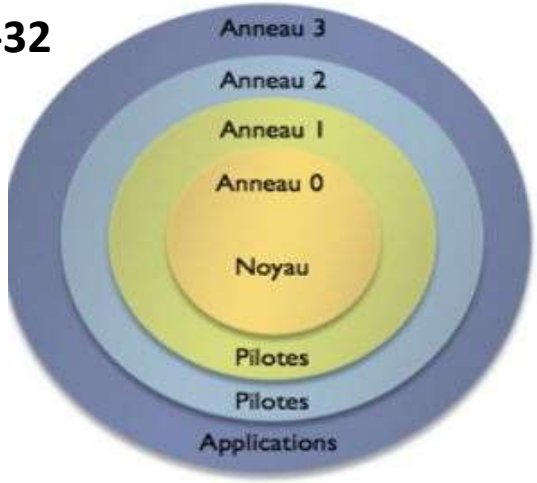
Virtualisation assistée au niveau matériel (Hardware Assisted Virtualization).

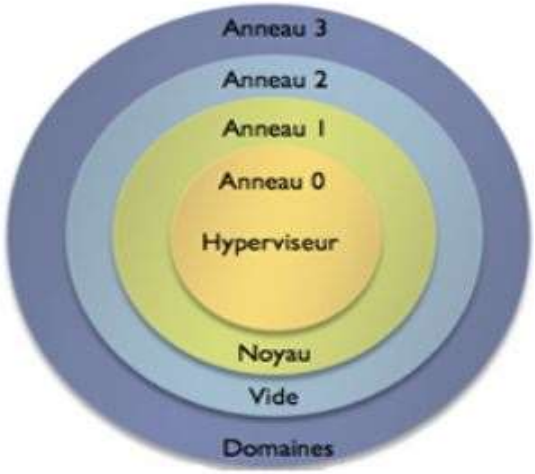
Source: voir références

Rappel : Anneaux de protection système

Architecture x86-32

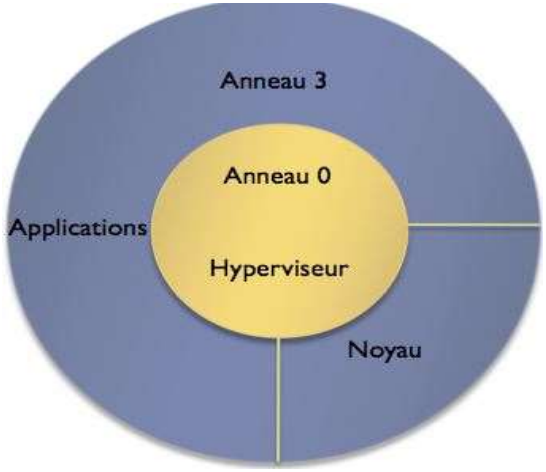


Sans virtualisation

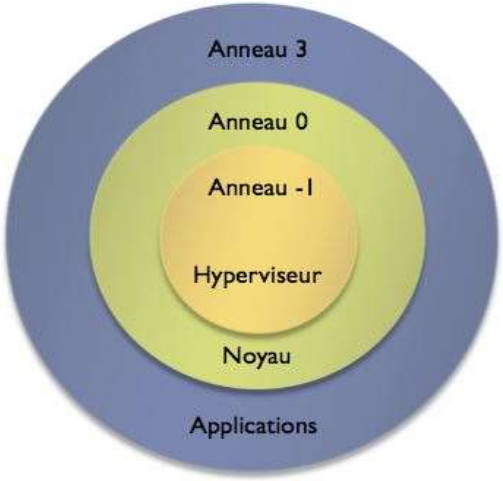


Avec virtualisation

Architecture 64-bit



Projet Xen



Projet AMD et Intel

Virtualisation assistée au niveau matériel (Hardware Assisted Virtualization).

Objectif :

Faire fonctionner des systèmes invités dont les **OS peuvent être différents** mais **non modifiés**

➢Cette technologie consiste à **séparer les ressources matérielles au niveau de la carte mère** de la machine.

➢Le support de la virtualisation peut être **intégré ou assistée par le processeur** qui se charge de :

- Virtualisation des **accès mémoire**
- **Protection du processeur** physique des accès bas niveaux

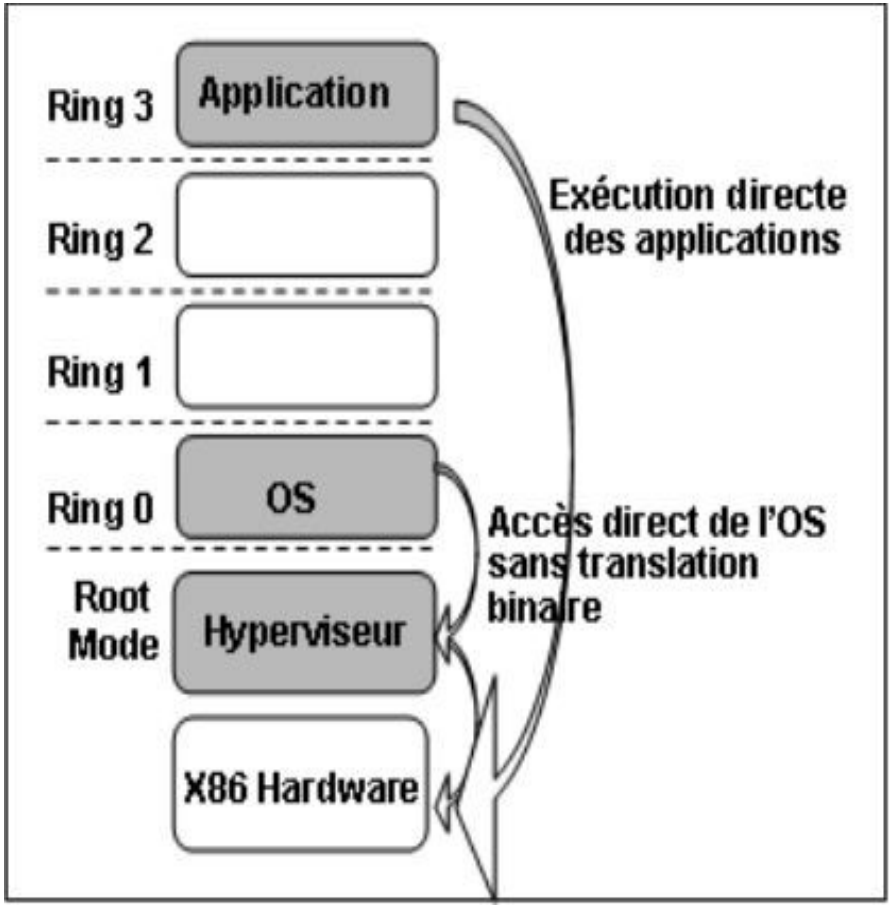
Source: voir références

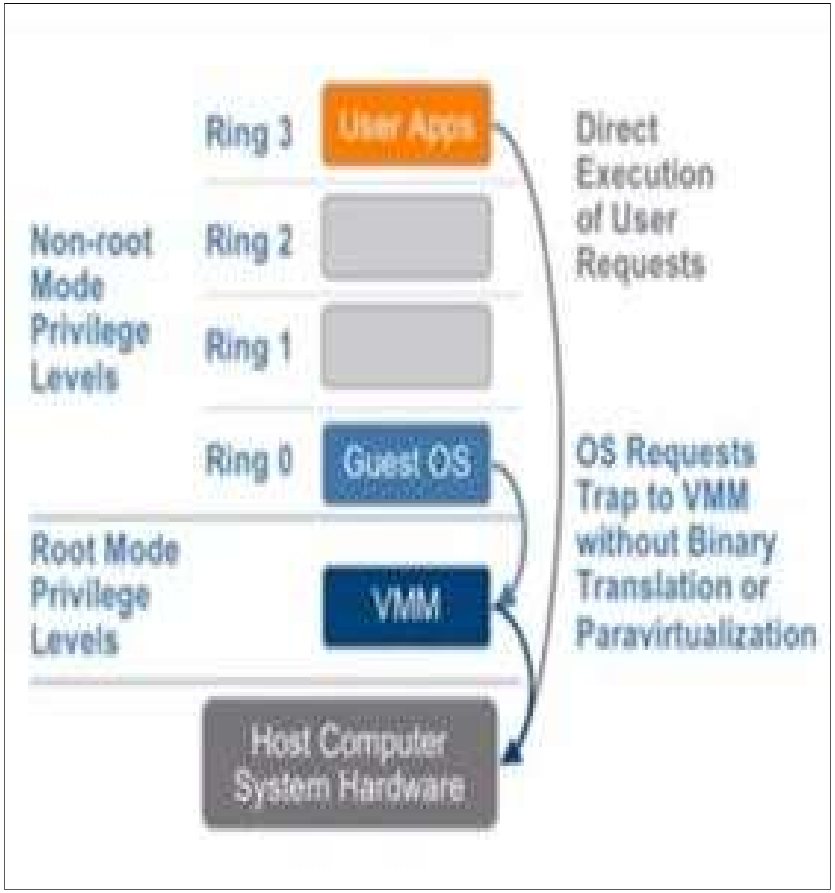
Virtualisation assistée au niveau matériel (Hardware Assisted Virtualization).

➢ Afin d'éviter de mettre **l'OS sur un Ring qui n'est pas conçu** pour lui ou **modifier du kernel de l'OS**, les processeurs **Intel VT** (Vritualization Technology) et **AMDV** (Virtualization) implémentent un **nouveau mode d'exécution** appelé **virtualisation assistée au niveau matériel. (Simplifier la tache de l'hyperviseur)**

➢ La virtualisation assistée par le matériel présente au **niveau du processeur** une « **extension** » qui lui permet de **communiquer à la fois avec le système d'exploitation de la machine physique et d'autres systèmes d'exploitation** par l'intermédiaire d'une puce dédiée aux tâches de virtualisation

Virtualisation assistée au niveau matériel (Hardware Assisted Virtualization).





- Il comporte un niveau racine (**Root**), correspondant à des Rings inférieurs à 0, et un niveau normal, correspondant aux anciens Rings de 0 à 3.
  Ce niveau privilégié accède directement au matériel

Source: voir références

Virtualisation assistée au niveau matériel (Hardware Assisted Virtualization).

- L'hyperviseur fonctionne en **mode Racine (Root Mode)** avec le niveau de contrôle le plus élevé.
- OS invités fonctionnent sur le **Ring 0.** Ils occupent bien l'emplacement pour lequel ils ont été conçus (**Placement du OS Guest dans son anneau traditionnel**.)
- Les machines virtuelles gèrent leurs **propres interruptions** et le **matériel de gère directement les zones de mémoire vive** disponibles au machines virtuelle
- Il n'est plus besoin de **modifier les Guest OS**, **ni d'utiliser de translation binaire** (la translation binaire est cependant toujours nécessaire pour certains jeux d'instructions).
- L'ajout d'un **degré d'intelligence au niveau du matériel** qui sera **conscient de l'utilisation d'une couche de virtualisation**

Source: voir références

Virtualisation assistée au niveau matériel : Bienfaits

- Ce nouveau **niveau racine** réduit considérablement **l'overhead**
- **Fluidité du partage des ressources physiques** entre les machines virtuelles
- **Augmentation des performances**
- Il **n'est plus besoin de modifier les Guest OS, ni d'utiliser de translation binaire**
- Simplifie la virtualisation logicielle
- Les architectures X86 s'affranchissent de certaines barrières techniques

Source: voir références

Virtualisation assistée au niveau matériel (Hardware Assisted Virtualization): Acteurs

1) **Xen** permet d'exécuter **plusieurs systèmes d'exploitation (et leurs applications) de manière isolée** sur une même machine physique

2) **KVM (**Kernel-based Virtual Machine) est un module de noyau chargeable pour Linux qui tire parti de technologies de virtualisation matérielle comme Intel VT et AMD-V.

Chaque machine virtuelle dispose d'une UC, d'une RAM et d'une interface réseau qui lui est propres.

Source: voir références

Virtualisation assistée au niveau matériel (Hardware Assisted Virtualization): Acteurs

3) VSphere ESXi 6 et 7 est un hyperviseur de **Type 1** et permet de gérer et virtualiser des serveurs

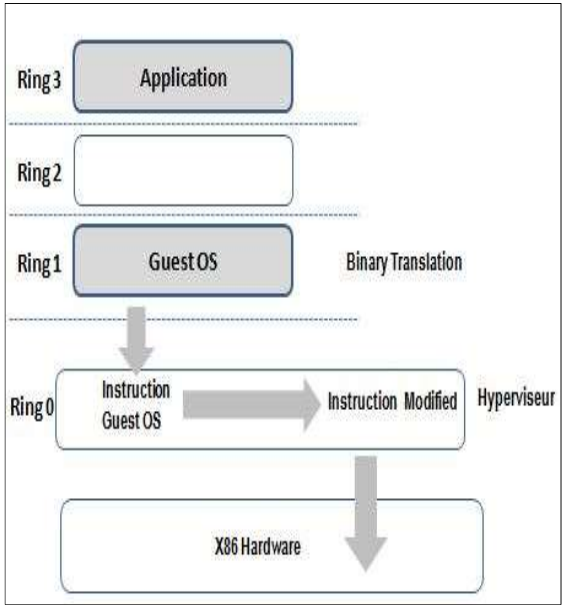
vSphere ESXi dispose de nombreuses fonctionnalités qui permettent de gérer au mieux les différentes VMs :

- **vMotion** : vMotion est une fonctionnalité permettant la migration à chaud (sans avoir à éteindre la VM) entre 2 hôtes **ESXi**
- **vSphere HA** : High Avaibaility est une fonctionnalité permettant un redémarrage automatique des VM après une panne sur l'hôte
- **vShield Endpoint** : c'est un système d'antivirus/antimalware permettant de sécuriser les VM sur l'hôte.

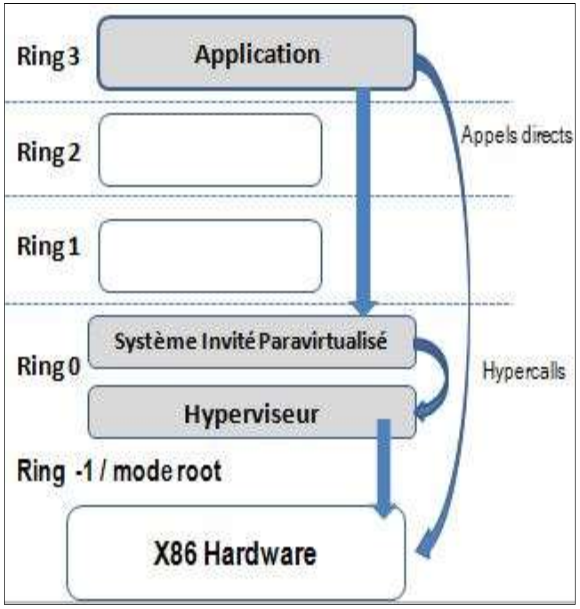
4) Hyper-V : est un système de virtualisation basé sur un hyperviseur 64 bits de la version de Windows Server

Source: voir références

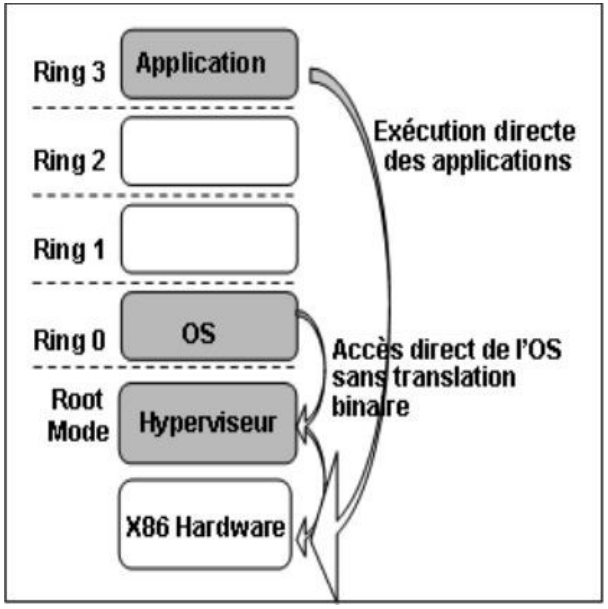
Comparaison Full virtualisation, Paravirtualisation
Virtualisation assistée au niveau matériel



Full virtualisation



Paravirtualisation



Hardware Assisted Virtualization

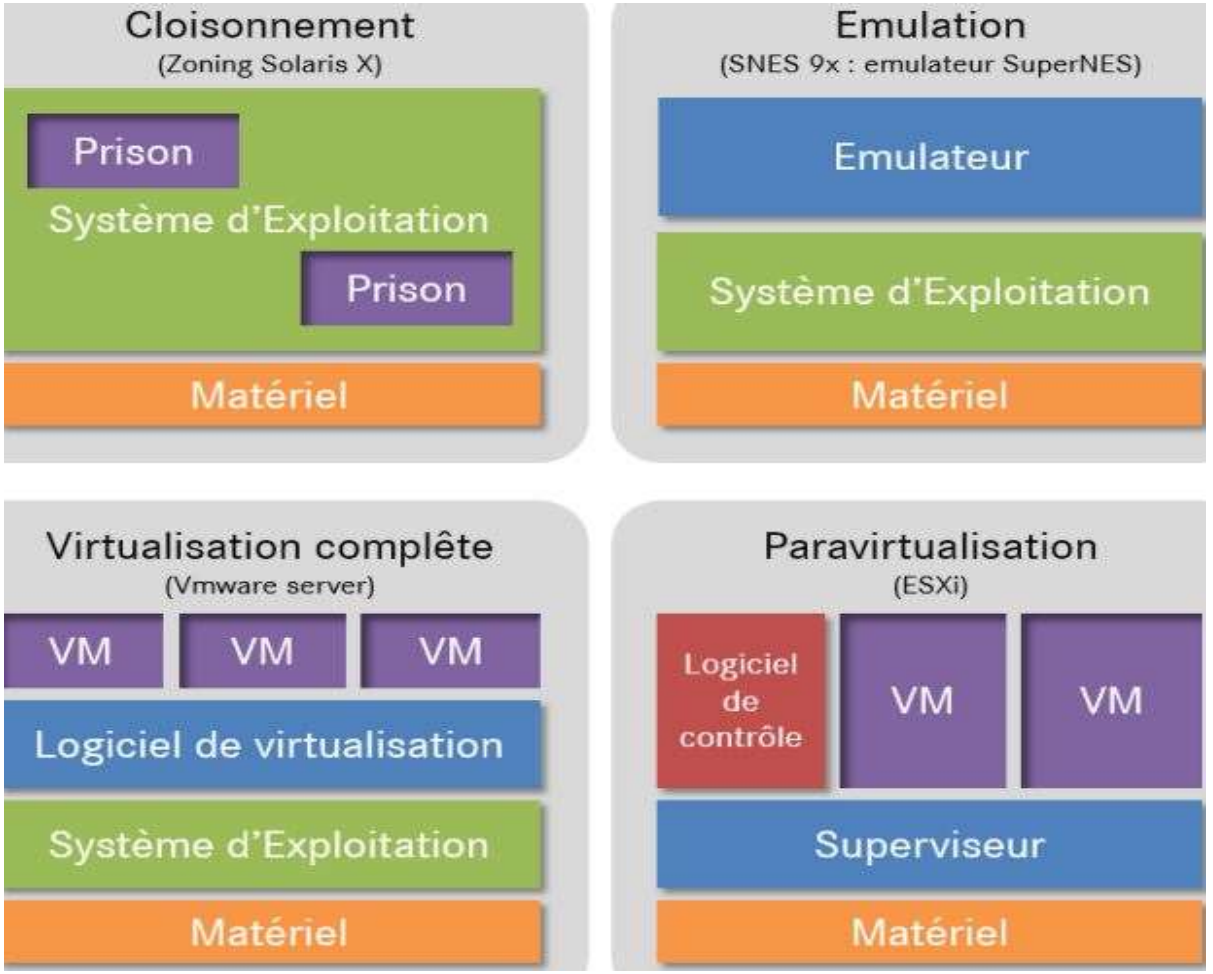
N.B : ESXi fonctionne en mode Paravirtualisation et Hardware Assisted Virtualisation

Source: voir références

Comparaison : Cloisonnement , Emulation, Full virtualisation, Paravirtualisation

Allouer définitivement des ressources

Zone Solaris



Transformer des instructions

Source: voir références

Virtualisation au niveau applicatif

(Application Level virtualization)

Virtualisation au niveau applicatif (Application Level virtualization)

Objectif :

Dissocier l'application du système d'exploitation hôte, des autres applications présentes

afin **d'éviter les conflits** en les simulant sous forme de **plusieurs services.**

Source: voir références

Virtualisation au niveau applicatif (Application Level virtualization)

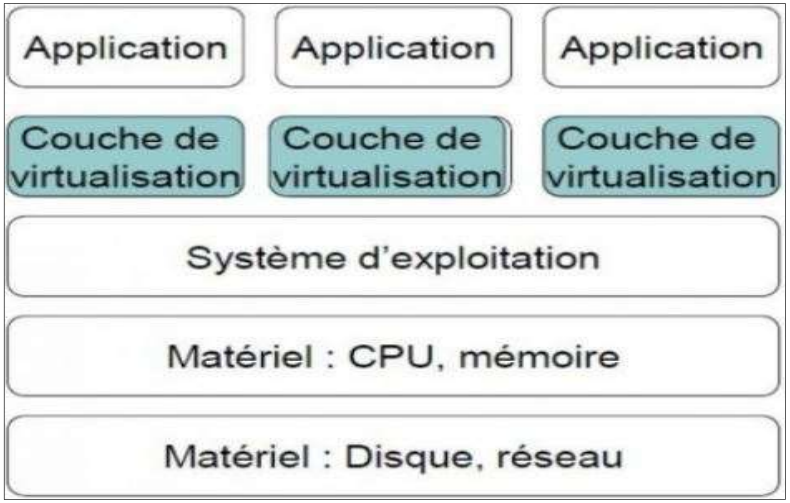
Principe :

Encapsuler dans un **même package** l'**application** et son **environnement système** de manière **imperméable** au système d'exploitation sur lequel l'application s'exécute.

- De cette manière, même si une application est défectueuse, elle ne va **pas amputer les autres applications ni le système d'exploitation**.
- Les applications sont considérées comme des **services virtuels** et n'ont donc pas besoin d'être installées sur chaque ordinateur.
- Les applications sont installées dans un **emplacement isolé** du centre de calcul de l'entreprise où elles sont séparées du système d'exploitation sous-jacent et des autres applications.

Source: voir références

Virtualisation au niveau applicatif (Application Level virtualization)



➢ Les applications sont transformées en **services virtuels**, **administrés de façon centralisée**, **sans nécessiter d'installation** sur chaque ordinateur .

1) **Si la virtualisation est réalisée côté serveur,** il s'agit de la **virtualisation de session**. L'utilisateur ouvre une **session sur un serveur distant.** Les applications sont délivrées aux utilisateurs de façon centralisée
2) **Si la virtualisation est réalisée sur le poste de travail (~~Virtual Desktop Integration, VDI~~):** Il consiste à afficher sur des centaines (voire des milliers) de **postes physiques**, une **image virtuelle du poste** utilisateur qui est en fait réellement exécutée sur un **serveur distant** .

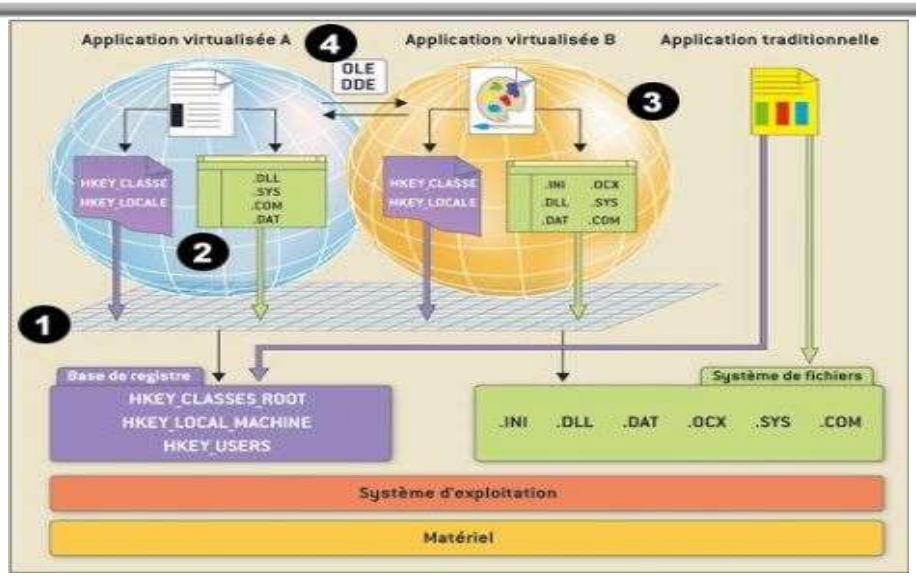
   L'utilisateur pouvait accéder aux meilleurs logiciels, **sans avoir une machine locale puissante,** nécessaire pour les faire fonctionner.

Source: voir références

Virtualisation au niveau applicatif (Application Level virtualization) : comparaison

(2) A chaque application ses fichiers système

(1) Une couche d'intégration



1) La virtualisation des applications **ajoute une couche** entre les **programmes virtualisés** et le **système d'exploitation** qui **intercepte les appels systèmes aux base de registre et applications**.

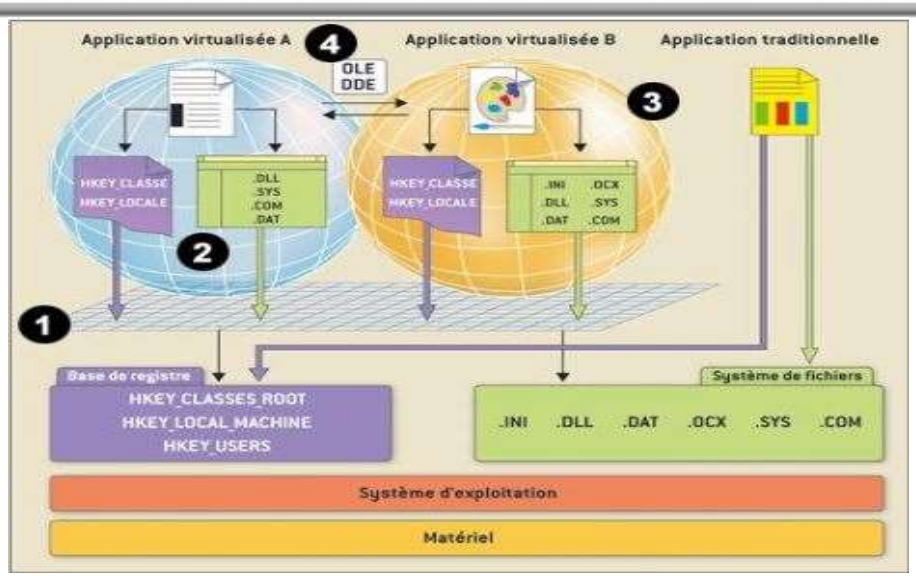
2) Le **système de fichiers** et la **base de registre virtuels** ne sont pas des **copies** de ceux du système d'exploitation.
Ils regroupent uniquement les **modifications effectuées par l'application** pour qu'elle puisse fonctionner.

Source: voir références

Virtualisation au niveau applicatif (Application Level virtualization)

(2) A chaque application ses fichiers système

(1) Une couche d'intégration



(3) Des bulles étanches

(4) tests réduits lors des changements d'OS

3) Si l'application veut **altérer la configuration système**, elle ne le fera que dans **sa copie de la base de registre.**

Elle n'aura accès qu'à ses **propres versions de DLL** et **de fichiers de configuration** système.

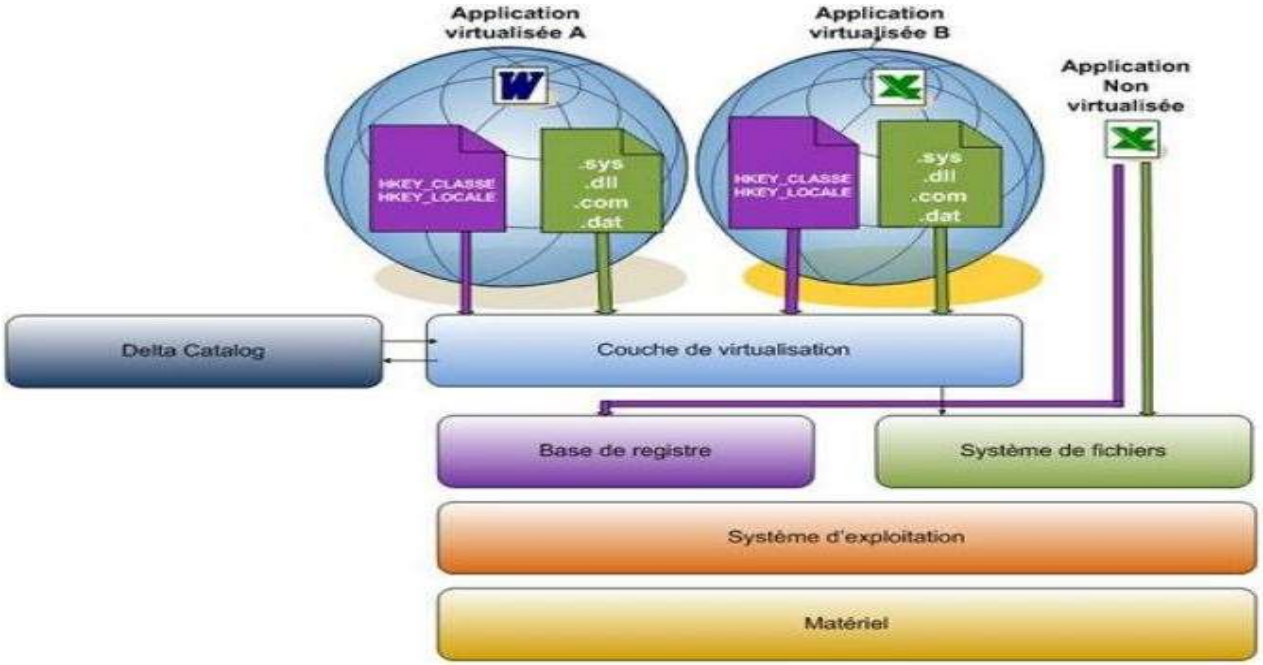
Donc, Il **n'y a pas de conflit** avec les autres applications. C'est ce qu'est appelé le **concept de bull**

4) Plusieurs applications peuvent opérer dans différentes bulles et elles demeurent **indépendantes les unes des autres**.

Une telle garantie d'indépendance entre les applications **limite fortement le volume des tests de régression nécessaires** en cas de **changement de système d'exploitation**.

Source: voir références

Virtualisation au niveau applicatif : Exemple sous windows



Concept :

- Intercepter les opérations de l'application
  - Sur le système de fichier
  - Sur la base de registre (windows)
  - Sur les variables d'environnement
- Enregistrer les modifications
  - « delta catalog »
- Stocker les données de l'applications dans un endroit virtualisé
  - Arborescence spécifique à l'application

Virtualisation au niveau applicatif : Exemples

Exemple 1:

- Installation d'un logiciel de virtualisation d'application sur serveur classique ou virtuel comme **Citirix Xen**
- Installation de l'application Office 2016 sur le serveur
- Les utilisateurs qui se connectent à l'application qui s'exécute sur le serveur ont l'impression que l'application tourne localement >>>> ce qui est faux

Exemple 2:

- Hôtes virtuels avec le serveur Web Apache.
- Domaines virtuels avec Postfix

Source: voir références

Virtualisation au niveau applicatif : Exemples d'application

- **VMWare ThinApp (Windows)** :
  - Solution qui permet d'exécuter quasiment tous les types d'applications sur la plupart des environnements d'exploitation Windows, **sans aucun conflit**.
  - Les utilisateurs peuvent ainsi, par exemple, exécuter Internet Explorer **(IE) 6 et IE 7** sur le **même système d'exploitation sans interrompre** leurs opérations en cour.
  - Il consiste à générer un exécutable (fichier .exe) à partir du programme d'installation du logiciel à virtualiser qui **s'installe de manière isolée par rapport au système d'exploitation**.

- **Citrix XenApp** :
  - Il s'agit d'un logiciel serveur permettant de **déployer des applications ou des services sur un réseau et d'y accéder à distance** à partir de clients légers. On parle de « solution d'infrastructure d'accès »
  - L'installation d'une application se fait sur le serveur et se lance sur celui-ci

- **Microsoft APP-V**

Virtualisation au niveau applicatif : Avantages/Inconvénients

Avantages:

- Les logiciels d'application peuvent être déployés, gérés et maintenus de manière centralisée.
- En isolant l'application, le système sous-jacent est protégé des codes malveillants et **l'incompatibilité est pallié**.
- Création des copies de ressources partagées pour chaque application.

Inconvénients :

- Les applications qui sont intégrées étroitement au système d'exploitation ou qui nécessitent l'accès à des pilotes périphériques spécifiques ne peuvent pas être virtualisées
- La virtualisation des applications soulève des questions de licence.

Source: voir références