





La virtualisation

**Situation ?**

**Imaginez la situation suivante :**

- ✓ vous faites partie d'une petite entreprise de développement informatique, composée d'environ 20 salariés.
- ✓ Chaque employé possède un ordinateur Linux, Mac ou Windows selon ses besoins, auxquels s'ajoutent quelques smartphones sous Android.

### **Vos besoins (problème) :**

Vous souhaitez déployer une application de stockage de données sur tout votre parc informatique.

- Plusieurs applications répondent aux besoins de l'entreprise 😊

### **Mais avant d'en choisir une :**

- vous souhaitez (devez) tester si elles sont bien complètement fonctionnelles sur Windows, Linux, Mac et Android.

**Solution :**

La solution la plus fiable pour tester cela serait de le faire directement sur des machines de l'entreprise.

➤ Mais elles sont toutes utilisées !!

Et il n'est pas question de monopoliser les machines de vos collègues pour faire vos tests.

Heureusement, des solutions existent pour vous simplifier la vie et vous permettre de faire tous vos tests sur votre propre ordinateur :

- Tout ça grâce à la **virtualisation**.

**Intérêt(s) ?**



**On peut déjà lister les cas suivants :**

1. tester un OS sans l'installer sur votre machine physique ;
2. s'assurer de la compatibilité d'une application avec différents OS ;
3. utiliser une application dans un environnement particulier ;
4. utiliser les fonctionnalités spécifiques d'un OS ;
5. simuler une 2e machine et faire des tests de communication simples ;
6. créer un petit réseau de plusieurs machines ;
7. simuler des environnements multi-OS ;
8. mutualiser des serveurs physiques et faire des économies ;
9. partager un serveur physique en plusieurs serveurs virtuels ;
10. rendre une architecture hautement disponible (HA) en améliorant sensiblement sa
11. tolérance aux pannes ;
12. migrer des serveurs virtuels sur de nouveaux serveurs physiques ;
13. déployer un serveur virtuel à la volée ;
14. faciliter les opérations de maintenance.

Vous voyez déjà qu'avec cette liste non exhaustive, la virtualisation a des intérêts multiples, et qu'il serait dommage de s'en passer.

**Qu'est-ce qu'une machine virtuelle ?**

Ce qu'on appelle machine virtuelle ou VM (Virtual Machine), est une machine qui n'est pas physique ;

➤ autrement dit, qui **n'est pas** constituée de **ressources physiques**.

Pour fonctionner, une machine a effectivement besoin de 4 types de ressources principaux :

1. le **CPU** ;
2. la **mémoire** (RAM) ;
3. le **stockage** (disque dur) ;
4. le réseau (**optionnel**).



Comment une machine virtuelle peut-elle donc fonctionner sans ressources physiques ?

**C'est simple :**

elle utilise les ressources d'une machine physique sur laquelle elle est installée :

➤ la **machine hôte**.

Le principe de la virtualisation repose effectivement sur le **partage de ressources**.

La machine hôte n'a pas forcément besoin de 100 % de chaque type de ressource pour fonctionner :

➤ elle peut donc en "**prêter**" ou en **allouer une partie** à une ou **plusieurs** VM.

Pour faire simple, une VM est un **ordinateur dans un ordinateur** (bienvenue dans la matrice).

**L'hyperviseur ?**

**Vous avez donc une machine dans une machine.**

➤ Mais comment se répartissent-elles les ressources ?

**Nativement**, les machines (Windows, Linux ou autre) **ne sont pas** capables de partager les ressources physiques qu'elles utilisent.

**C'est là qu'intervient l'hyperviseur.**

Derrière ce mot un peu barbare se cache simplement une **catégorie d'applications** capables de partager les ressources de la machine sur laquelle elles sont installées.

**L'hyperviseur a 2 rôles majeurs :**

- ✓ créer des ressources virtuelles propres à chaque VM ;
- ✓ répartir ses propres ressources.

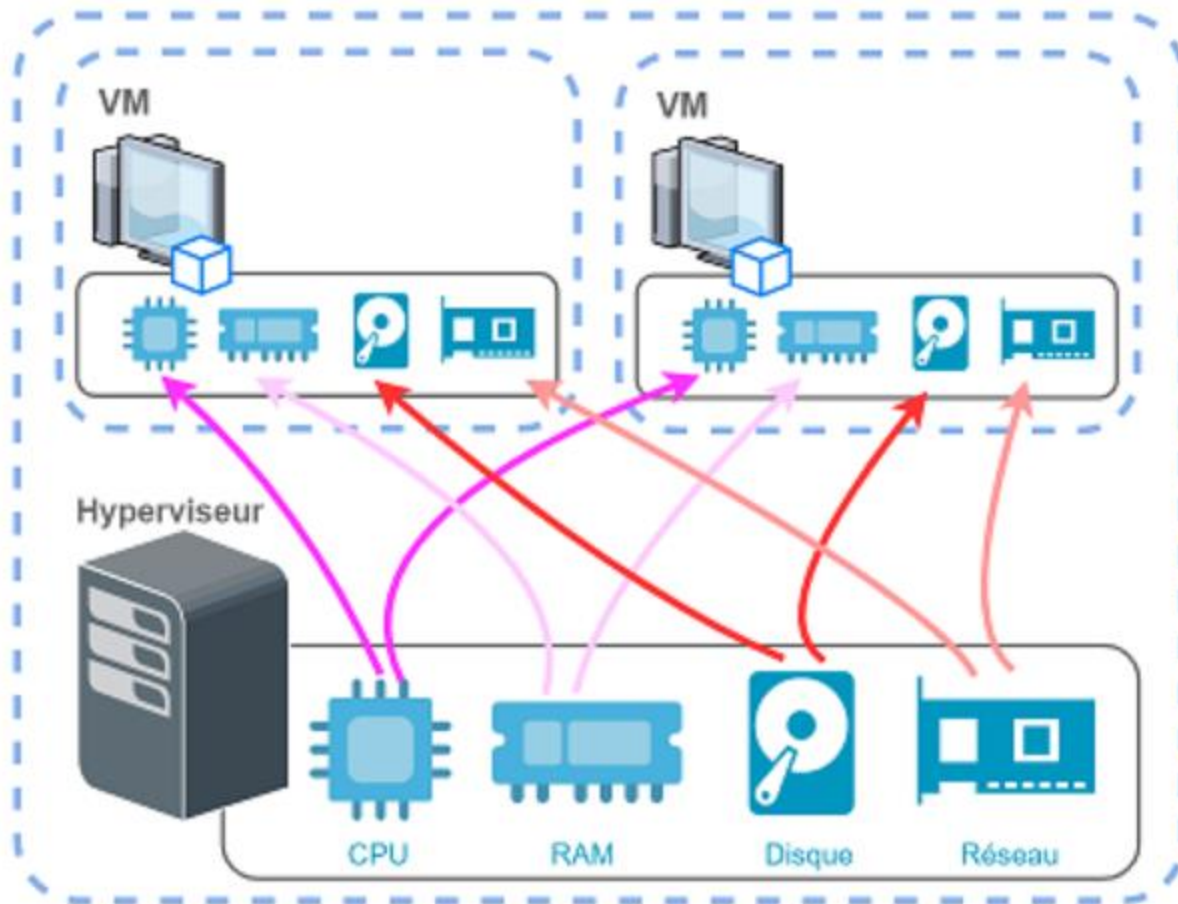
## **Créer des ressources virtuelles propres à chaque VM :**

Dans un premier temps, il crée pour chaque VM des ressources virtuelles.

- En effet, les VM ne vont pas aller se servir directement dans les ressources de la machine hôte, tout simplement car elles n'ont pas conscience d'être des VM hébergées sur un hôte.
- Elles vont donc aller piocher dans leurs ressources virtuelles.

Et comme n'importe quelle machine physique, une VM aura donc son propre disque dur, sa mémoire, son processeur et ses périphériques, à la différence près que tout cela sera virtuel.

La même en visuel ☺





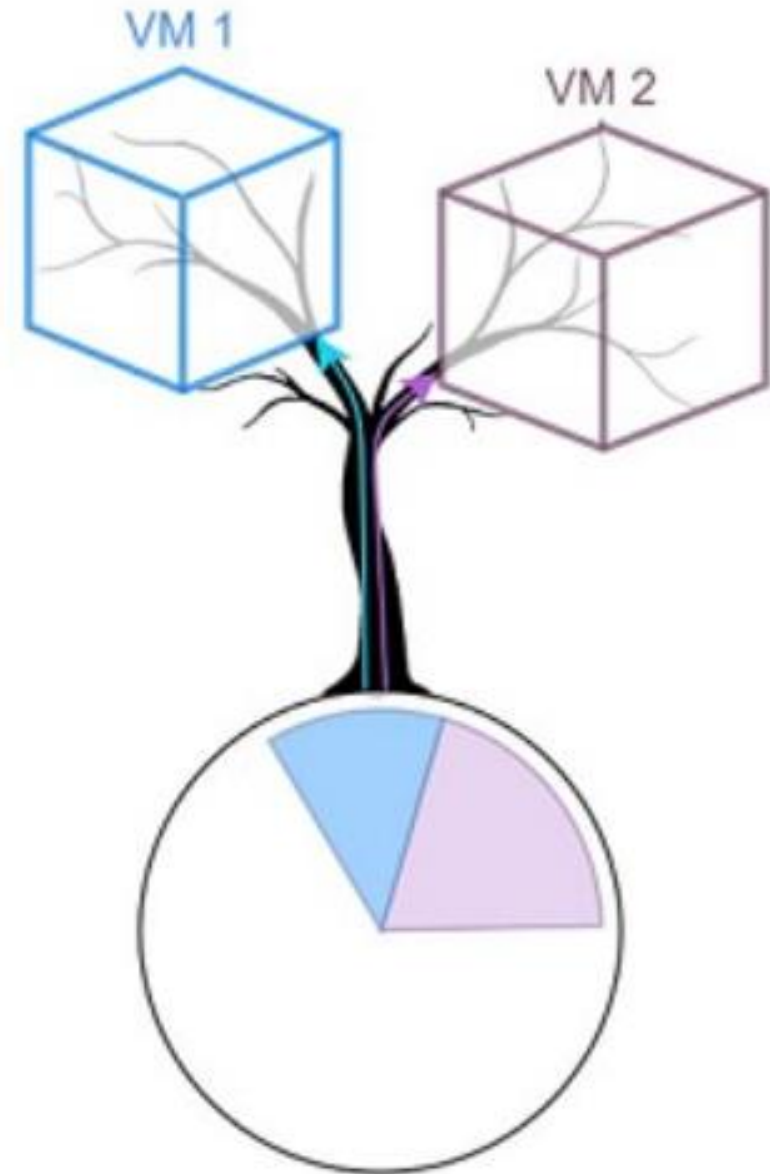
## Répartir ses propres ressources :

Dans un second temps, l'hyperviseur joue le rôle de **chef d'orchestre** :

- Il **alloue** à **chaque** VM les ressources dont elles ont **besoin** :
  - ✓ au bon moment
  - ✓ et dans les bonnes quantités (mais dans une certaine limite qui aura été fixée).

**On peut assimiler cela à un organisme vivant :**

comme un arbre, qui dispose de ressources (eau, nutriments) puisées dans le sol, et va les dispatcher vers ses branches, en fonction de leurs besoins respectifs.



**Distinguez les hyperviseurs selon vos besoins ?**

**Il existe 2 grandes catégories d'hyperviseurs qui s'utilisent dans des cas très différents.**

Si vous cherchez un moyen de transport pour relier Strasbourg à sa banlieue, vous allez opter pour le vélo, la voiture, le tram ou le bus :

➤ c'est ce qui correspond à un hyperviseur de **type 2**.

Ils sont effectivement adaptés pour les **petites infrastructures**.

De façon générale, les hyperviseurs de type 2 sont adaptés dans le cas où vous avez une seule machine et que vous voulez faire des tests multiplateformes (d'application, d'OS, de communication...)

## En revanche :

si vous cherchez un moyen de transport pour relier Paris à Montréal, vous allez plutôt opter pour l'avion.

- c'est ce qui correspond à un **hyperviseur** de **type 1**.

Les hyperviseurs de type 1 sont particulièrement utiles dans des **grosses architectures réseaux d'entreprise**, qui nécessitent :

- ✓ des optimisations de coûts et de maintenance,
- ✓ tout en améliorant la robustesse face aux pannes.

**Ces 2 moyens de transport ne sont pas vraiment interchangeables.**

En effet :

- il est peu probable que vous puissiez prendre l'avion entre Strasbourg et sa banlieue, car d'une part :
  - ✓ il n'y a pas les infrastructures prévues pour (pistes d'atterrissage, aéroport...)
  - ✓ et d'autre part, ça ne vous ferait absolument pas gagner de temps et ça vous coûterait bien plus cher.

Et ne parlons même pas du trajet Paris-Montréal en bus.



Pour les hyperviseurs, c'est la même chose : ils correspondent à deux usages distincts et ne sont donc pas interchangeables. Gardez donc bien en tête que ce n'est pas vous qui allez choisir votre type d'hyperviseur, mais que c'est votre besoin qui va le définir.

**Leurs différences d'un point de vue technique ?**



L'hyperviseur de **type 2**, ou hosted hypervisor (**hyperviseur hébergé**), est le plus facile à mettre en place.

- Il s'installe simplement comme n'importe quelle application sur votre système et permet, une fois installé, de créer des VM indépendantes de l'OS hôte.



Il est donc considéré comme n'importe quelle application et n'a **aucune priorité** sur les ressources de l'hôte. Si une application décide soudainement qu'elle a besoin de 95 % des ressources de votre machine pour ouvrir un fichier extrêmement volumineux, votre hyperviseur va se retrouver avec moins de 5 % des ressources pour faire tourner toutes ses VM.



Pourquoi **moins de 5 % et pas 5 % ?**

Simplement car votre OS, qui est le pilier central, a lui aussi besoin de ressources pour fonctionner.

➤ Il va donc se servir en premier pour :

1. alimenter les processus systèmes,
2. puis l'application prendra sa part,
3. et votre hyperviseur par la suite.

**En revanche :**

- chaque VM est **autorisée**, par l'hyperviseur, à utiliser une **certaine quantité** de ressources (**définie au départ**) et **ne peut donc pas dépasser** cette limite.

L'OS hôte ne pourra donc pas être ralenti par les VM :

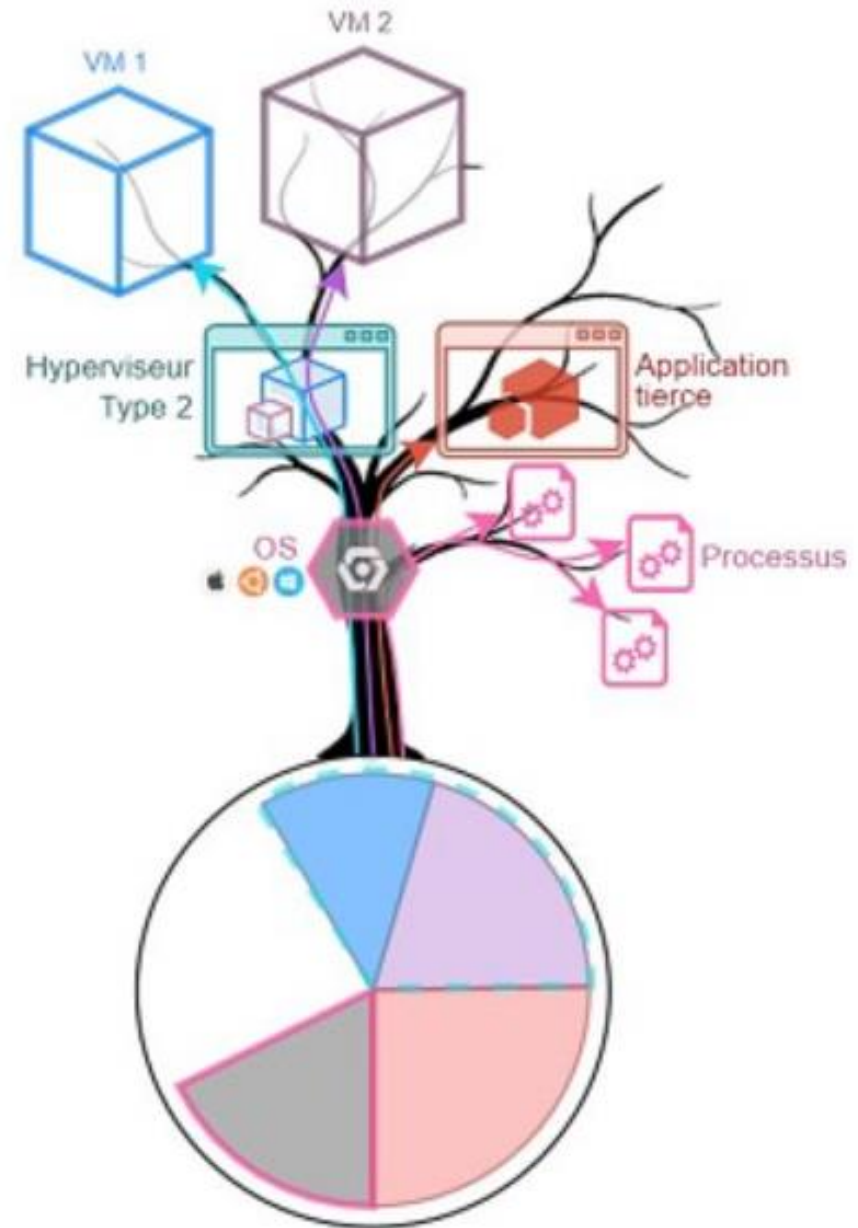
- ✓ si elles ont été configurées raisonnablement.

**Si l'on reprend l'exemple de notre arbre :**

l'OS est le tronc qui répartit les ressources à toutes les branches, c'est-à-dire les applications.

Il garde aussi des ressources pour son propre fonctionnement.

Chaque application (branche, donc) a des besoins spécifiques au fil de la journée, mais n'a a priori aucune priorité sur les autres applications (branches, toujours).

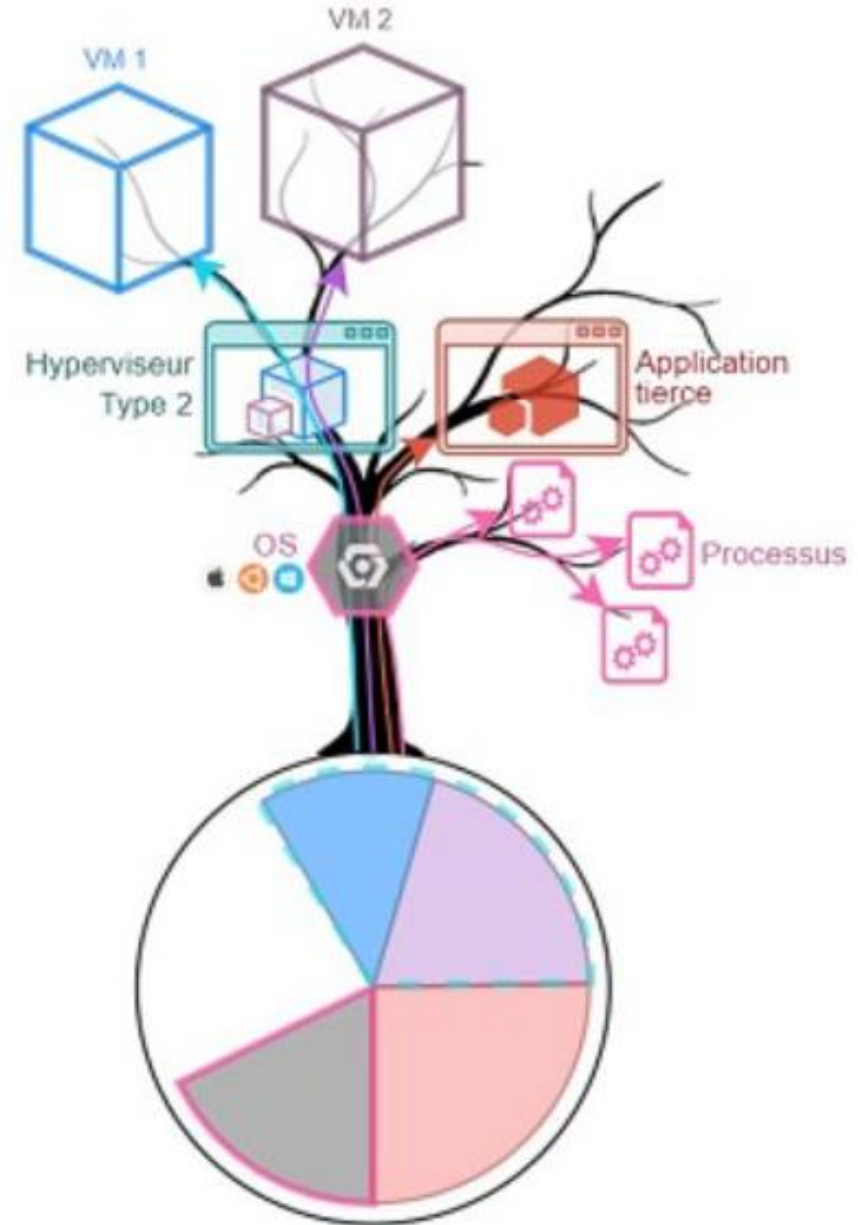


**Si l'on reprend l'exemple de notre arbre :**

L'hyperviseur de type 2 est lui aussi l'une de ces branches.

Il va répartir à son tour les ressources dans ses branches supérieures : les VM.

Ces branches supérieures devront utiliser les ressources qui leur sont distribuées et ne pourront en aucun cas aller chercher des ressources dans les étages inférieurs.





Un hyperviseur de type 2 est une application qui est assez gourmande en mémoire et CPU. Assurez-vous donc de fermer un maximum d'applications lorsque vous voulez en utiliser un.

## Les utilisations d'un hyperviseur de type 2 sont multiples.

- Ils sont **assez faciles** à mettre en place et **très efficaces** pour répondre aux besoins du type :
  - ✓ tester un OS sans formater votre machine physique ;
  - ✓ tester ou utiliser régulièrement une application sur un OS en particulier ;
  - ✓ simuler une 2e machine et faire des tests de communication simples ;
  - ✓ créer un petit réseau de plusieurs VM pour tester des protocoles réseau, des règles de pare-feu, configurer un serveur de supervision ou autre.



Du fait d'être installés sur un OS hôte, ces hyperviseurs ont des **performances limitées** et hébergent des VM qui peuvent par conséquent être **instables**.

Ces hyperviseurs ne sont pas faits pour créer des VM de production, mais uniquement pour du **développement**, des **tests** et de **l'usage personnel**.



**Comparez les différents hyperviseurs de type 2 ?**

**Les principaux hyperviseurs de type 2 sont les suivants :**

- ✓ Oracle VirtualBox ;
- ✓ VMWare Workstation (Player et Pro) et VMware Fusion (pour Mac) ;
- ✓ Microsoft VirtualPC (plus maintenu depuis 2011).

**Identifiez un hyperviseur de type 1 ?**

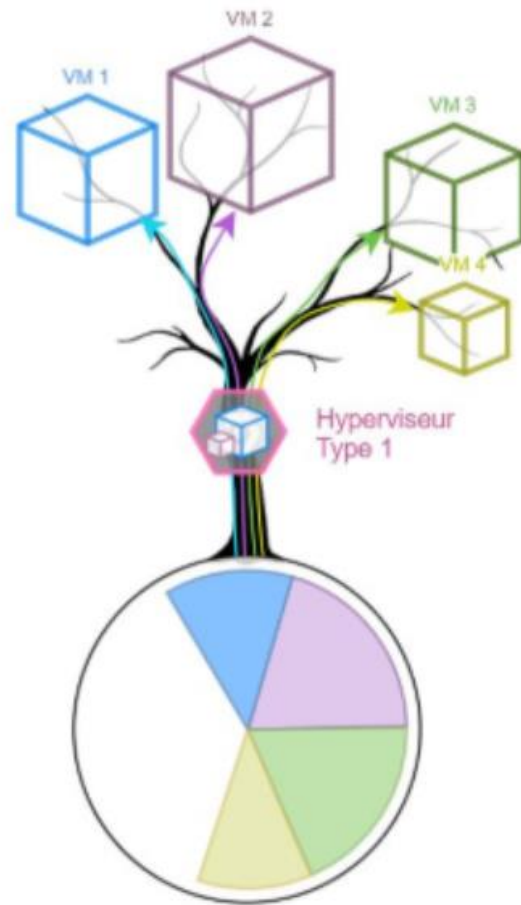
L'hyperviseur de **type 1** ou hyperviseur natif, est, **contrairement** au type 2 :

- **installé directement** sur le matériel, **sans** OS intermédiaire.

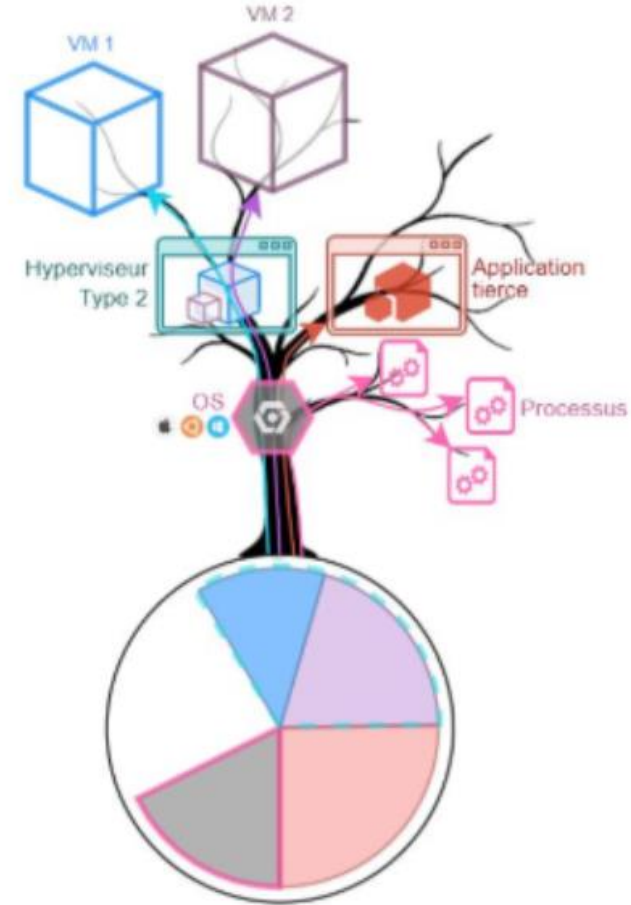
Cela signifie qu'avec un hyperviseur de type 1 :

- les *ressources* de l'hôte sont **directement gérées** par l'hyperviseur et non plus par l'OS qui disparaît ou est relégué au statut de VM.

Reprenons notre schéma de l'arbre, et voyez comme un hyperviseur de type 1 simplifie les choses par rapport à son homologue de type 2 :



*Fonctionnement d'un hyperviseur de type 1*



*Fonctionnement d'un hyperviseur de type 2*

## Avec un hyperviseur de type 1 :

- la machine hôte est **dédiée** à la création de VM et ne sait faire plus que cela.

Si ce type d'hyperviseur s'utilise dans un contexte complètement différent de l'hyperviseur de type 2, cela vient principalement de ses **performances bien supérieures**, rendues possibles car :

- ✓ d'une part, l'hyperviseur a un **accès direct** aux ressources (sans passer par un OS) ;
- ✓ d'autre part, la **totalité** des **ressources** est **dédiée** aux VM.



Ce type d'hyperviseur est destiné à héberger entre une dizaine et une vingtaine de VM. Il nécessite donc des serveurs très robustes et performants, et le coût de la licence pour les utiliser est souvent très élevé.

Les hyperviseurs de type 1 sont utilisés en entreprise pour **plusieurs raisons**, comme par exemple :

- ✓ réduire les coûts matériels et de maintenance ;
- ✓ optimiser les ressources physiques ;
- ✓ répartir la charge dynamiquement ;
- ✓ permettre la haute disponibilité des serveurs ;
- ✓ créer des VM de préproduction pour les tester en environnement réel avant de les mettre en production.

Il est assez rare, en tant que professionnel de l'informatique, de devoir choisir parmi ces solutions, tout simplement car :

- lorsque vous arriverez dans une entreprise, une solution de virtualisation sera déjà en place.

C'est donc celle-ci que vous utiliserez ;

- les choix se font plus souvent sur des critères politiques que technologiques :

Si votre entreprise a déjà signé des contrats chez certains fournisseurs qui sont en partenariat de près ou de loin avec un des professionnels de la virtualisation, il est très probable que c'est cette solution qui soit choisie.



**En résumé :**  
**le besoin détermine le type d'hyperviseur ?**

Contexte	Exemples de cas métiers	Hyperviseur	Profil des utilisateurs de l'hyperviseur
En entreprise ou pour un usage personnel	<ul style="list-style-type: none"> <li>• Tester un OS</li> <li>• Créer un environnement de test</li> <li>• Développer une application et la tester sur différents systèmes</li> </ul>	type 2	Profils multiples : <ul style="list-style-type: none"> <li>• développeur</li> <li>• ingénieur/administrateur réseau</li> <li>• technicien</li> <li>• profil non technique pour utiliser un outil spécifique</li> </ul>
En entreprise, dans des architectures en production	<ul style="list-style-type: none"> <li>• Créer un serveur</li> <li>• Remplacer des machines physiques par des VM (réduction des coûts)</li> <li>• Tests en environnement de pré-production</li> </ul>	type 1	Profils orientés réseau : <ul style="list-style-type: none"> <li>• ingénieur réseaux/cloud</li> <li>• administrateur réseau</li> </ul>

**Questions ??**

Prochain chapitre: installation et test d'une machine virtuelle.