

# Docker

Simplon - DATA IA

# Objectifs

- Découvrir les containers avec Docker
- Maîtriser la base de l'exécution de Docker
-

# Programme

- Executer son premier container
- Les containers ?
- Docker Desktop vs Docker CE
- Construire son propre container

# Exécuter son 1er container

- Télécharger Docker Desktop sur [docker.com/products/docker-desktop](https://docker.com/products/docker-desktop)
- Installer
- Lancer
  - Activer Hyper-V si besoin

# Exécuter son 1er container

- Lancer son terminal
- `$ docker pull ubuntu:18.04`
- `$ docker run -it ubuntu:18.04`



Conteneurs ?





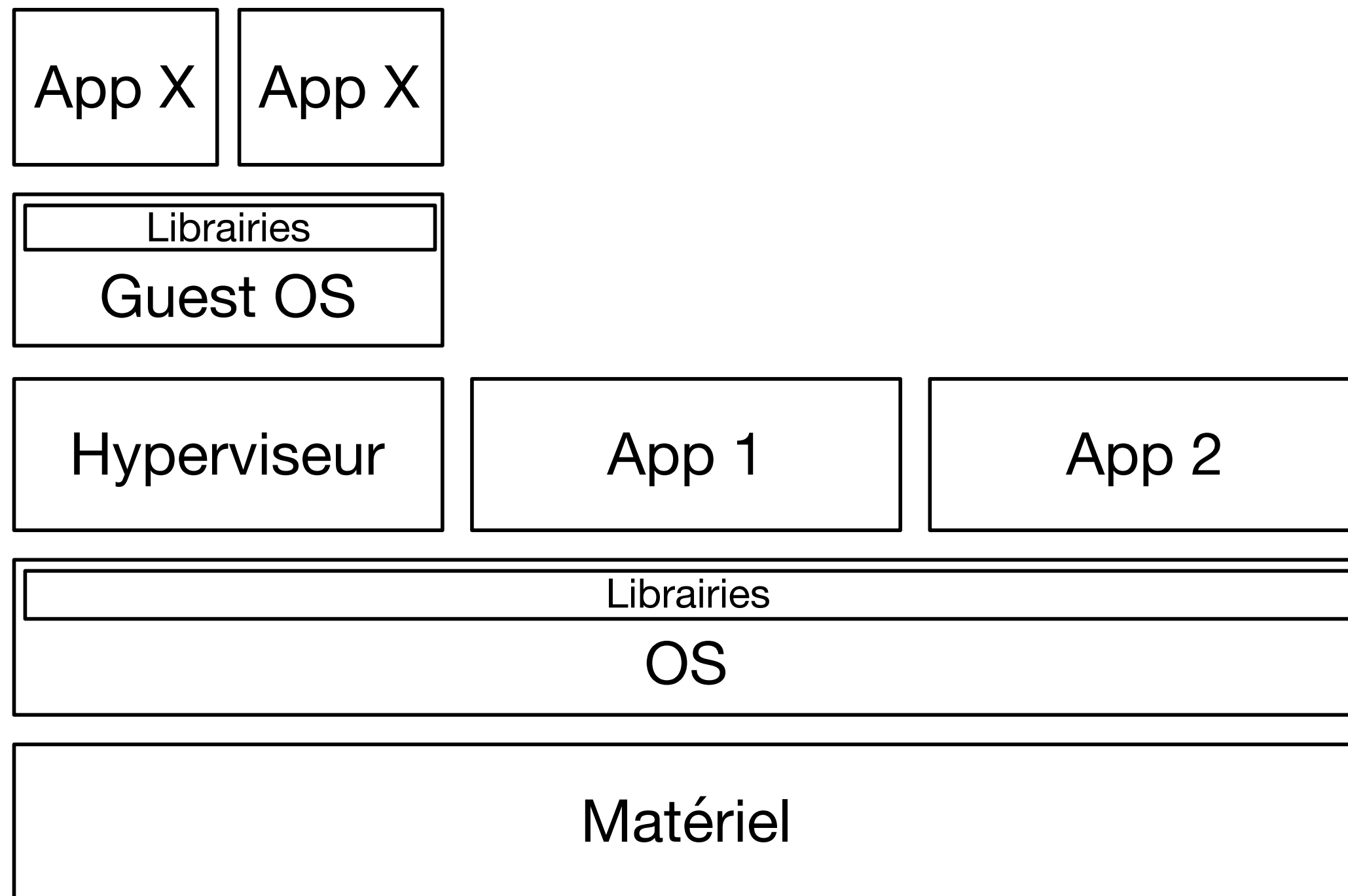


# VM vs Containers

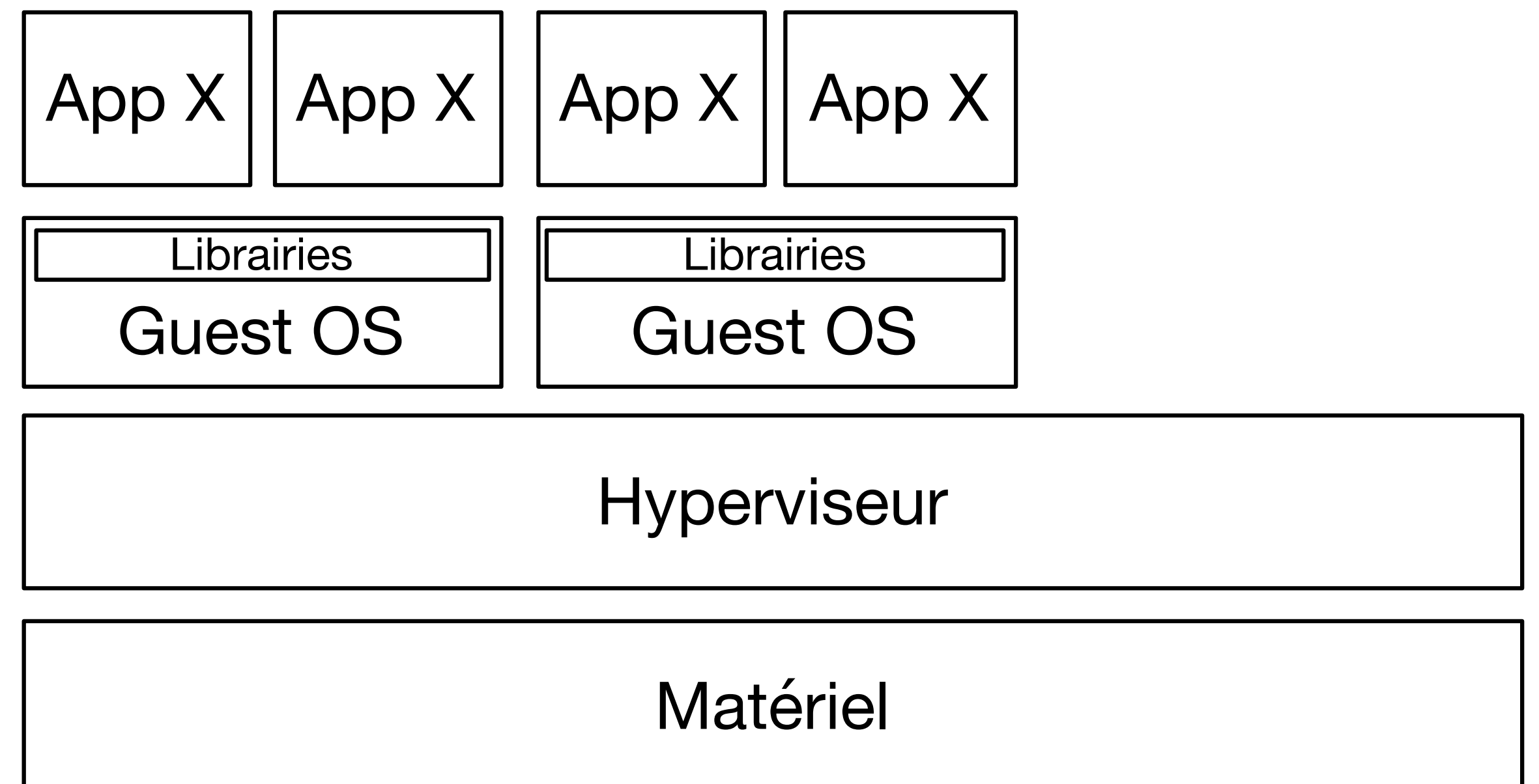
- Hyperviseurs Type 2 - hyperviseurs hébergés
  - VMWare Fusion, VirtualBox, ...
- Hyperviseurs Type 1 - hyperviseurs natifs
  - VMWare ESX, Microsoft Hyper-V, ...
- Containers
  - Docker, Kubernetes (k8s), OpenVZ, ...



# Machines Virtuelles



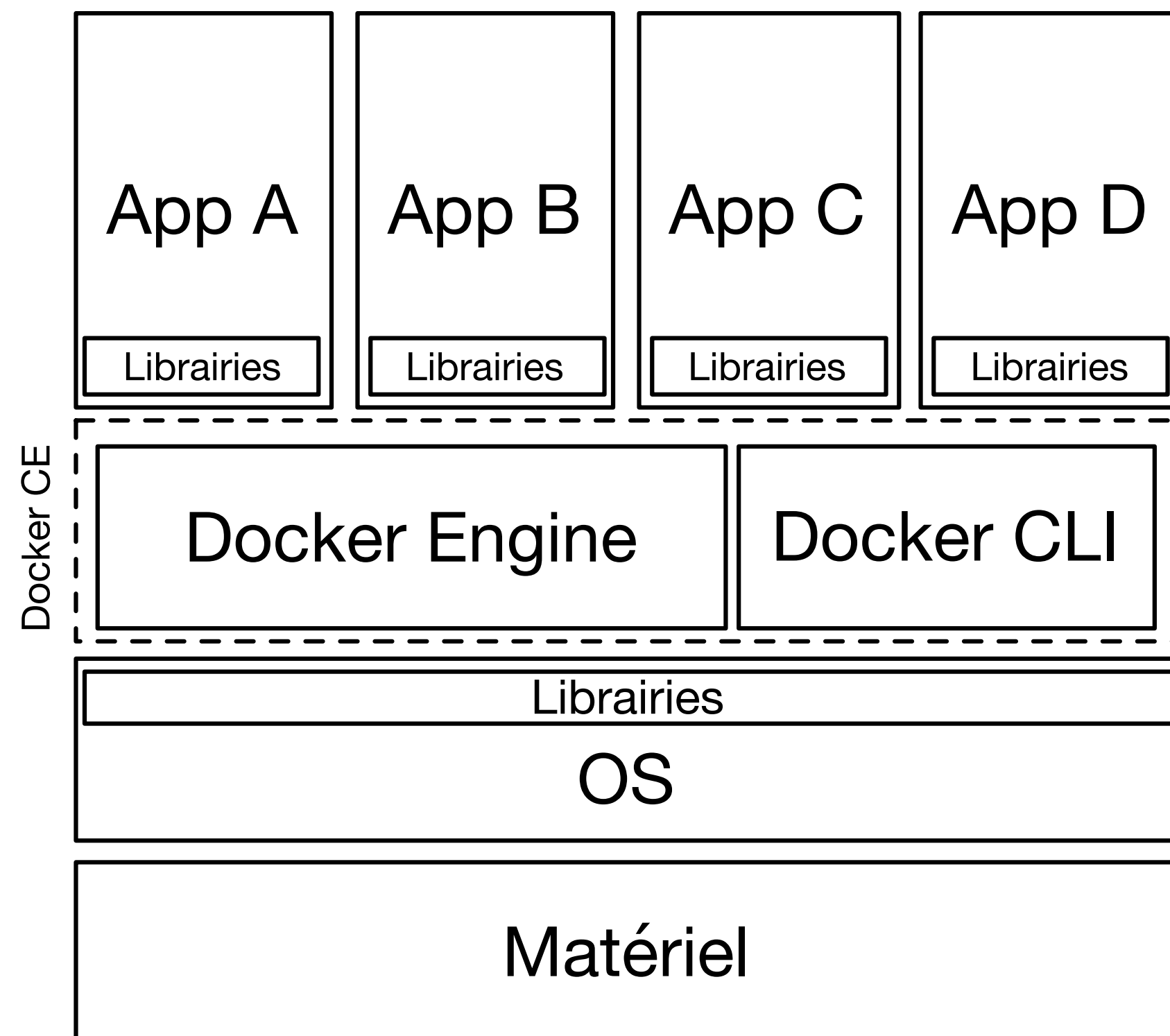
Hyperviseur Type 2



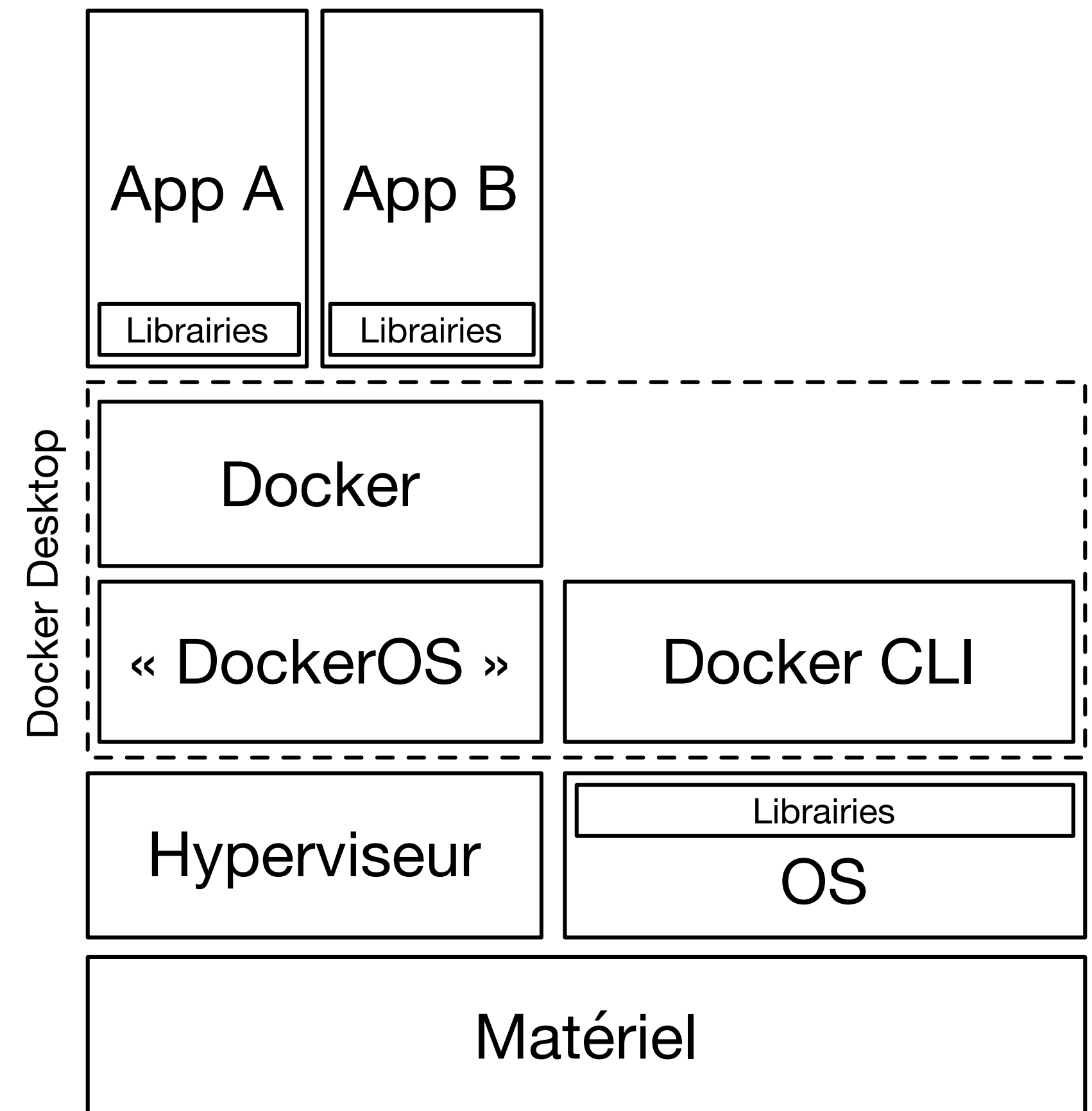
Hyperviseur Type 1



# Containers



Containers sur GNU/Linux



Containers sur Windows ou macOS



# Pourquoi des containers

- N'est pas seulement package l'application
- Inclus toutes les librairies nécessaires
- 1 container = 1 fonctionnalité
- Dépendances à d'autres services possible via le réseau
- Ne pas sauvegarder des données dans le container
  - Utiliser des « volumes »



# Vocabulaire

- HUB
  - Registre permettant de télécharger des images docker gratuitement
  - Permet d'héberger ses propres images docker publique (gratuit) ou privées (payant)
  - <https://hub.docker.com>

# Vocabulaire

- Image
  - Contient l'ensemble des fichiers et des paramètres d'exécution par défaut du programme souhaité
  - Est « versionné » avec des tags
  - Syntaxe : [registry/][user/]image\_name[:tag]
    - Tag par défaut : « latest »



# Vocabulaire

- Container
  - Instance d'une image Docker
  - Peut être personnalisé avec différents paramètres (réseau, volumes, ...)
  - Chaque container est isolé des autres, sauf on les associes explicitement (même réseau, volume partagé)

# Les commandes

- Obtenir la liste des commandes
  - `$ docker --help`
- Obtenir l'aide sur une commande
  - `$ docker <command> --help`



# Les commandes

- Récupérer une image depuis le hub
  - `$ docker pull <image>[:tag]`
- Lister les images locales
  - `$ docker images`
- Supprimer une image locale
  - `$ docker rmi <id|name>`

# Les commandes

- Créer un container à partir d'une image
  - `$ docker run [options...] <image>[:tag] [command]`
  - `$ docker run -d --name mysql-test mysql:latest`
  - `$ docker run -it ubuntu:18.03 bash`



# Les commandes

- Lister les containers
  - \$ docker ps [-a]
    - Option -a : lister également les containers stoppés

# Les commandes

- Stopper un container
  - \$ docker stop <id|name>
- Redémarrer un container stoppé
  - \$ docker start <id|name>
- Supprimer un container
  - \$ docker rm <id|name>



# Les commandes

- Executer une commande dans un container en cours d'exécution
  - `$ docker exec <options> <id|name> <command...>`
  - `$ docker exec -it mysql-test bash`
  - `$ docker exec -it mysql-test mysql -u root -p`

# Les volumes

- Par défaut, toutes les données sont stockée dans le container
- Si container supprimé, données supprimées
- Concept de « volume » : les données sont extériorisées
  - Par défaut, sur système de fichier de l'hôte
  - Avec plugins : AWS S3, NFS, ...

# Les volumes

- Volumes « anonymes »
  - Créés par défaut si définit dans l'image mais non référencé lors de l'exécution
- Volumes hôtes
  - `$ docker run -v "/home/arnaud/data:/var/data" [...] image_name`
  - `$ docker run -v "C:\Users\Arnaud\data:/var/data" [...] image_name`
- Volumes nommés
  - `$ docker volume create volume_name`
  - `$ docker run -v "volume_name:/var/data" [...] image_name`



# Les réseaux

- Docker crée des réseaux virtuels
- Plusieurs containers sur un même réseau peuvent interagir ensemble
  - Condition : avoir le port exporté dans l'image
- `$ docker network create network_name`
- `$ docker run --network network_name image_name`
- Publication de ports
  - Par défaut, les ports sont « fermés » depuis l'extérieur
  - Il faut « publier » le port
  - `$ docker run -p "port_externe:port_interne" image_name`