

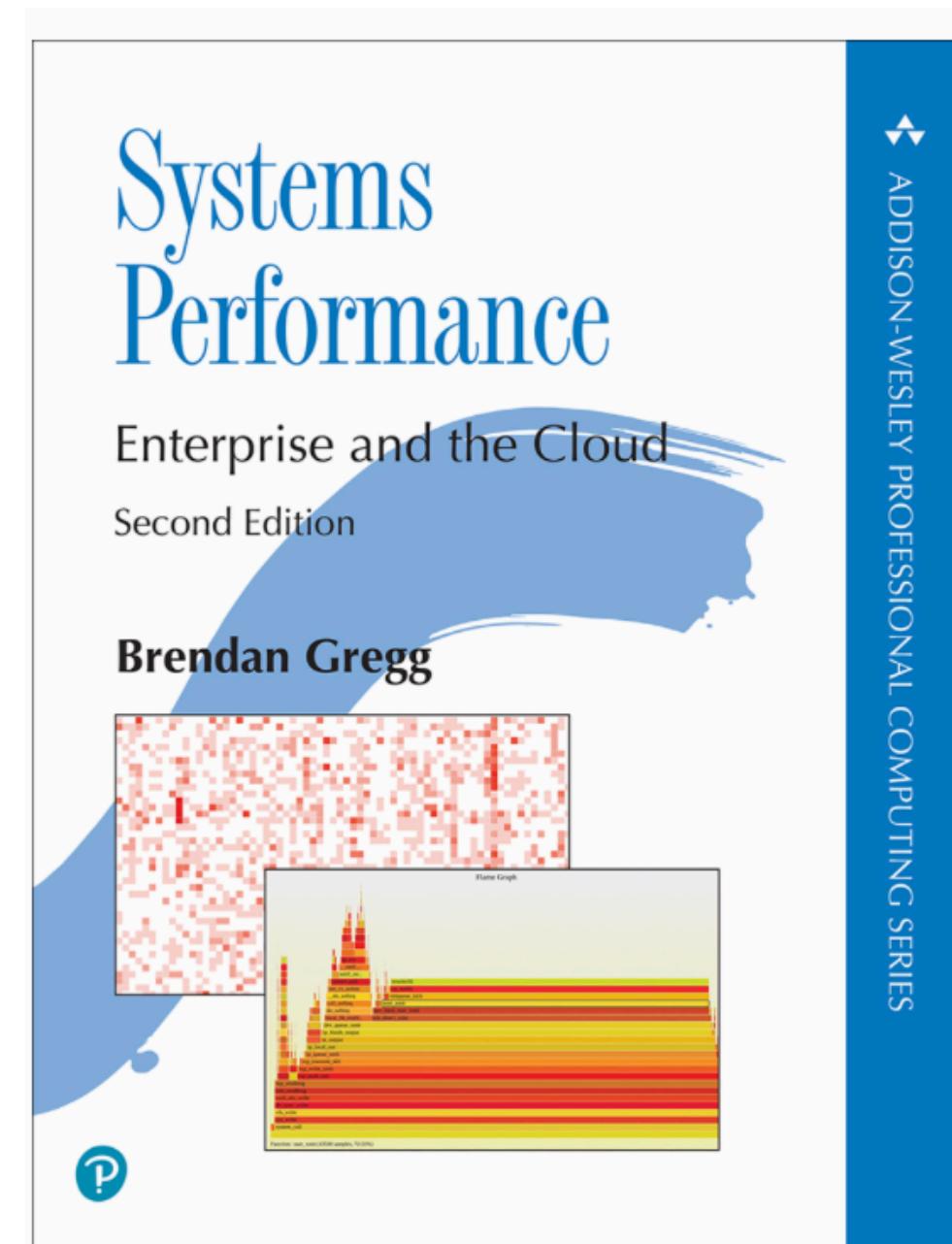


# Plan de la séance

Les architectures de cloud  
Virtualisation (IaaS)  
les différents types  
les différentes ressources

Une pile Cloud globale : OpenStack

Plusieurs schémas  
issus de

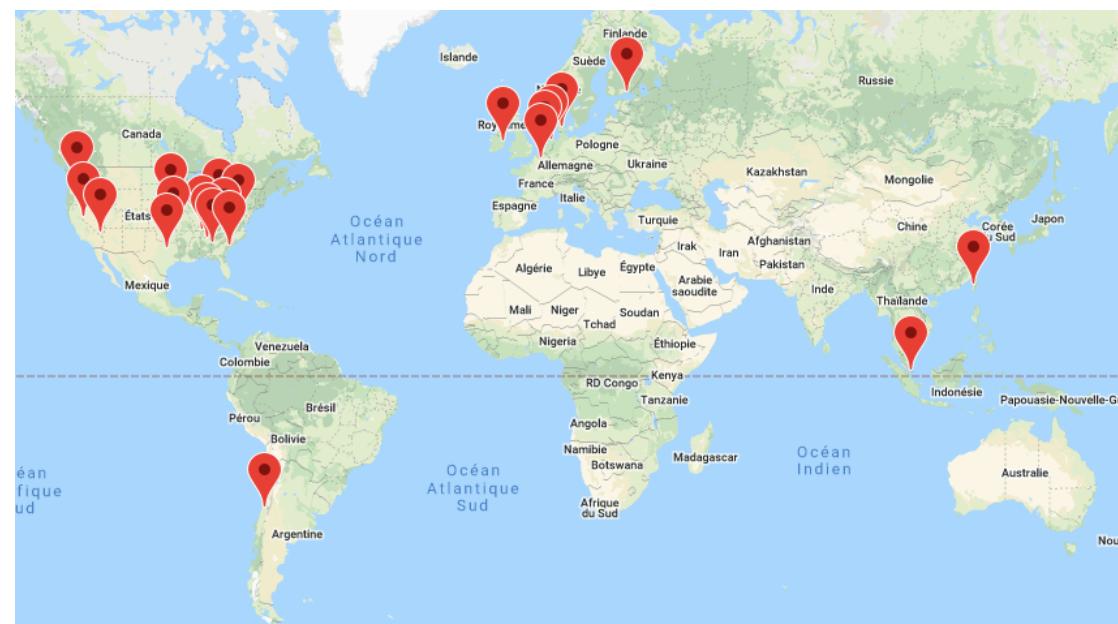


# Présence mondiale

AWS



Google



# Architecture physique



# Photos Google



# Des nuages dans l'océan...



8 fois moins de pannes que  
sur terre (humain, azote)

# Ressources physiques

Réseau

Ethernet 10/1G

interconnexion en propre ou partagé

Noeuds (serveur ou calcul)

processeur

mémoire

GPU

Stockage

SSD

Disques durs

Bandes

Doivent être partagées  
Concept clé : la **virtualisation**

# La virtualisation

Concept très ancien

Découplage entre le matériel et ce qu'en voient  
les systèmes  
les applications

Virtualisation faite par le matériel et/ou le logiciel

Vous connaissez déjà la mémoire virtuelle

# Partage des ressources

## Réseau

partage par des routeurs (permettant éventuellement) de la réservation  
dans un data center : par réseau local virtuel

Noeuds : notion de machines virtuelles/Container  
Mémoire, CPU, I/O

## Stockage

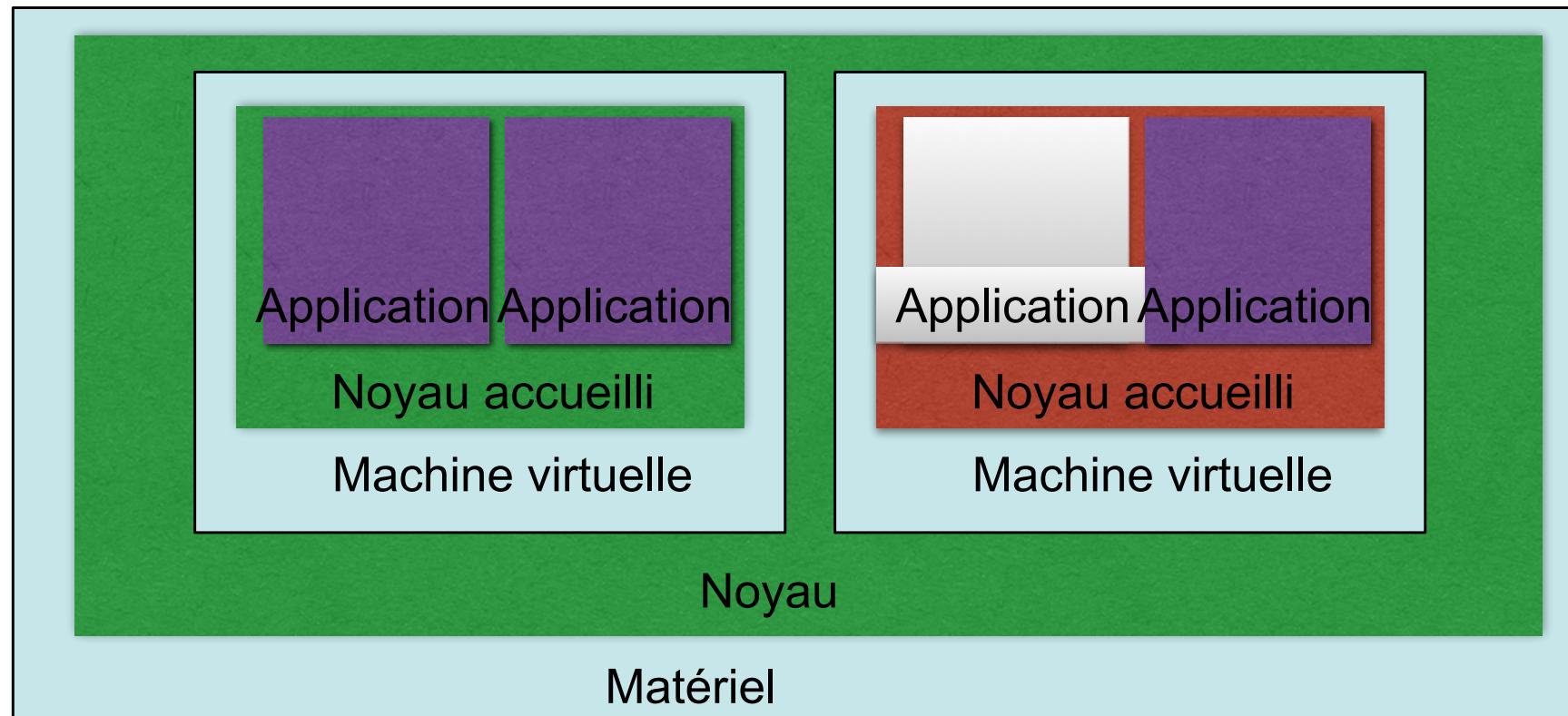
espaces de blocs avec isolation logique assurée par le système de Cloud  
Système de fichiers propre (OVH: ZFS)

# Virtualisation complète

La machine virtuelle est un processus dans une autre machine

VirtualBox, VMWare, QEMU...

Généralement coûteux (performances faibles)



# Séparation « physique »

S'appuie sur un composant physique, plus entièrement en logiciel

Utilisation de capacités particulières des processeurs pour séparer les machines virtuelles

chacune peut avoir sa pile complète

Noyau

Middleware

Applications

Utilisation d'un **hyperviseur**

en charge de la gestion des machines virtuelles

création

partage de ressources

suppression

peut être matériel ou logiciel

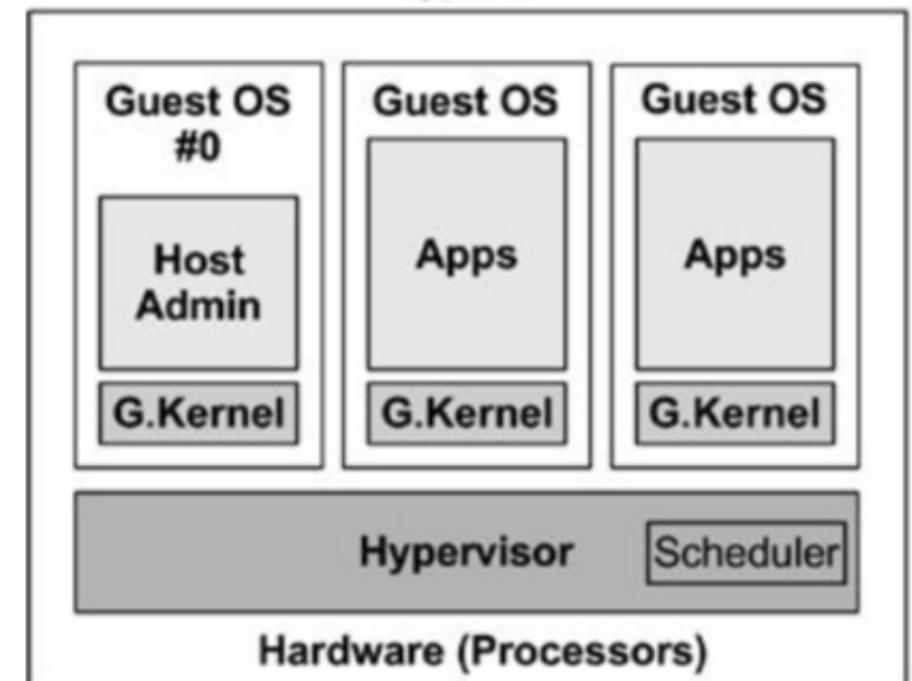
# Hyperviseur de type 1

Entre le matériel et les machines virtuelles  
appelé aussi *native* ou *bare metal*  
exécuté directement sur le matériel

Peut utiliser une VM particulière pour  
l'administration

Ordonnanceur spécifique entre les VMs

Exemple : Xen (2003)  
suivant le processeur, besoin de code spécifique dans l'OS invité



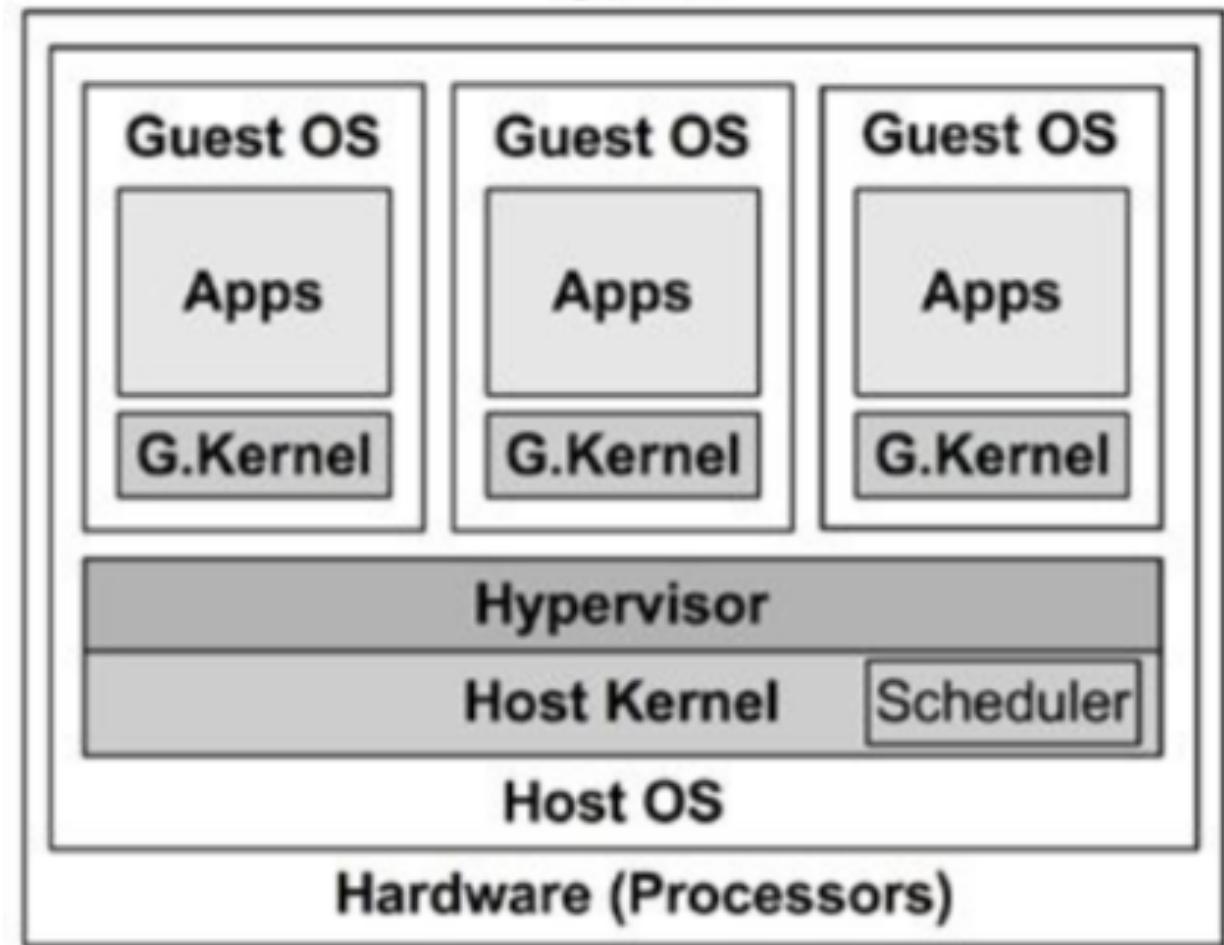
# Hyperviseur de type 2

Exécuté au-dessus du noyau de l'hôte avec des extensions spécifiques  
Proche de la virtualisation applicative mais avec une meilleure efficacité

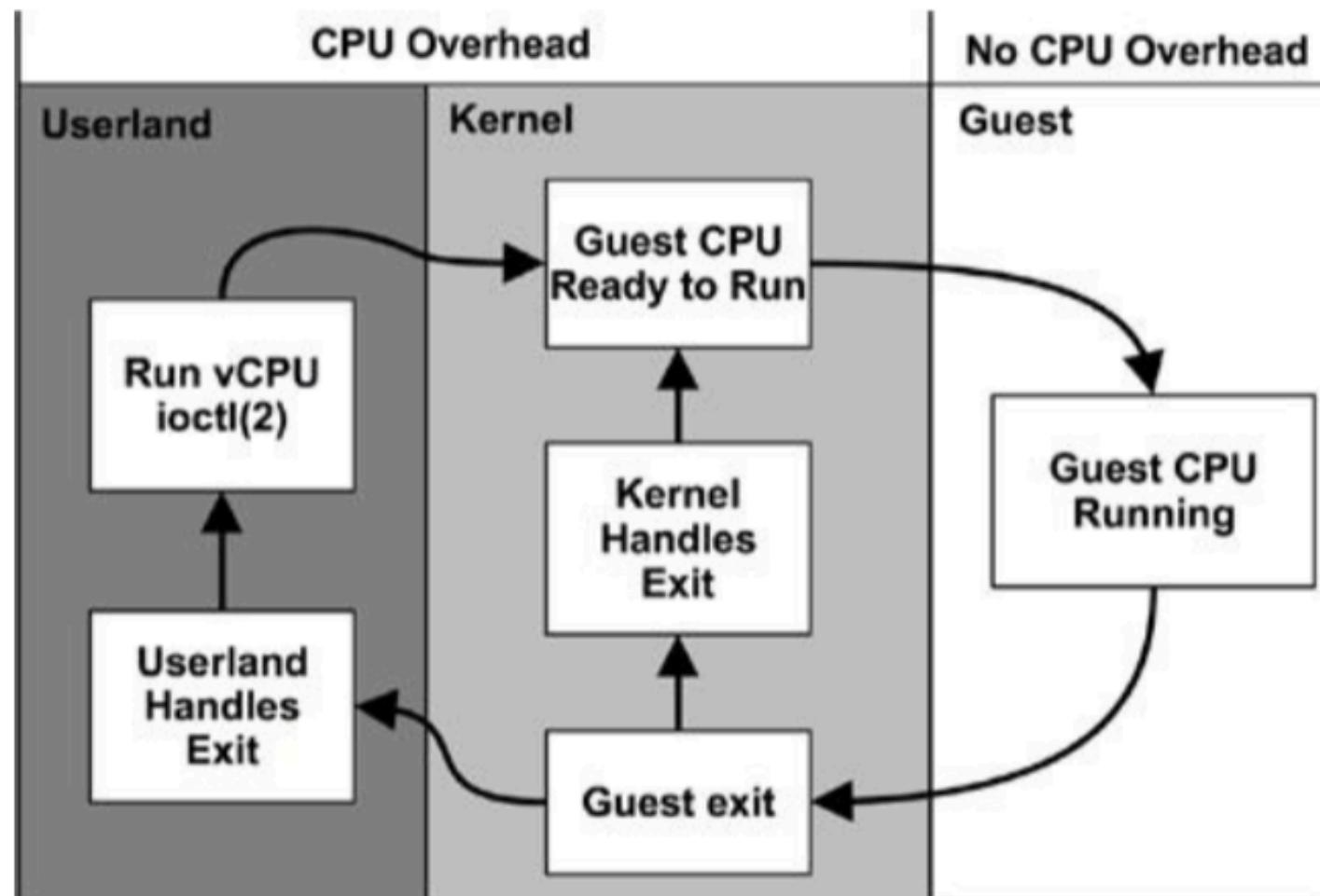
Exécution des machines par des modules au niveau du noyau couplé avec des processus utilisateurs

Plus de flexibilité car l'hôte est une machine complète

Ex : KVM (2007)



Chaque passage par l'hyperviseur ajoute un surcoût (guest exit)



# The Storage Latency Hierarchy

Technology	Latency	Size (e.g.)
L1 CPU Cache	4 cycles (~1 nsec)	32K
L2 CPU Cache	10 cycles (3 nsec)	256K
LLC CPU Cache	40 cycles (13 nsec)	1 MB
DRAM	240 cycles (80 nsec)	16 GB
NVRAM	1200 cycles (400 nsec)	128 GB
RDMA Read	6K cycles (2 usec)	16 GB
FLASH Read	150K cycles (50 usec)	128 GB
FLASH Write	1500K cycles (500 usec)	128 GB
HDD Write min	1500K cycles (500 usec)*	4 TB
HDD Read min	15000K cycles (5 msec)	4 TB
HDD Read max	75000K cycles (25 msec)	4 TB
Tape File Access	15000000K cycles (50 sec)	6 TB

2017-8

\* Write to track cache

# Paravirtualisation

La machine virtuelle a des points d'accès directs au matériel pour plus d'efficacité

réseau  
disque

Support nécessaire dans le système d'exploitation hôte système d'exploitation invité pour une meilleure efficacité

Sur architectures cloud existantes paravirtualisation peut être transparente pour le client (prise en charge par le noyau de l'OS)

Avantage : plus grande efficacité utilisé pour serveurs de données, noeuds de calcul intensifs etc.  
Xen et KVM l'utilisent beaucoup

# Paravirtualisation (2)

## Inconvénients

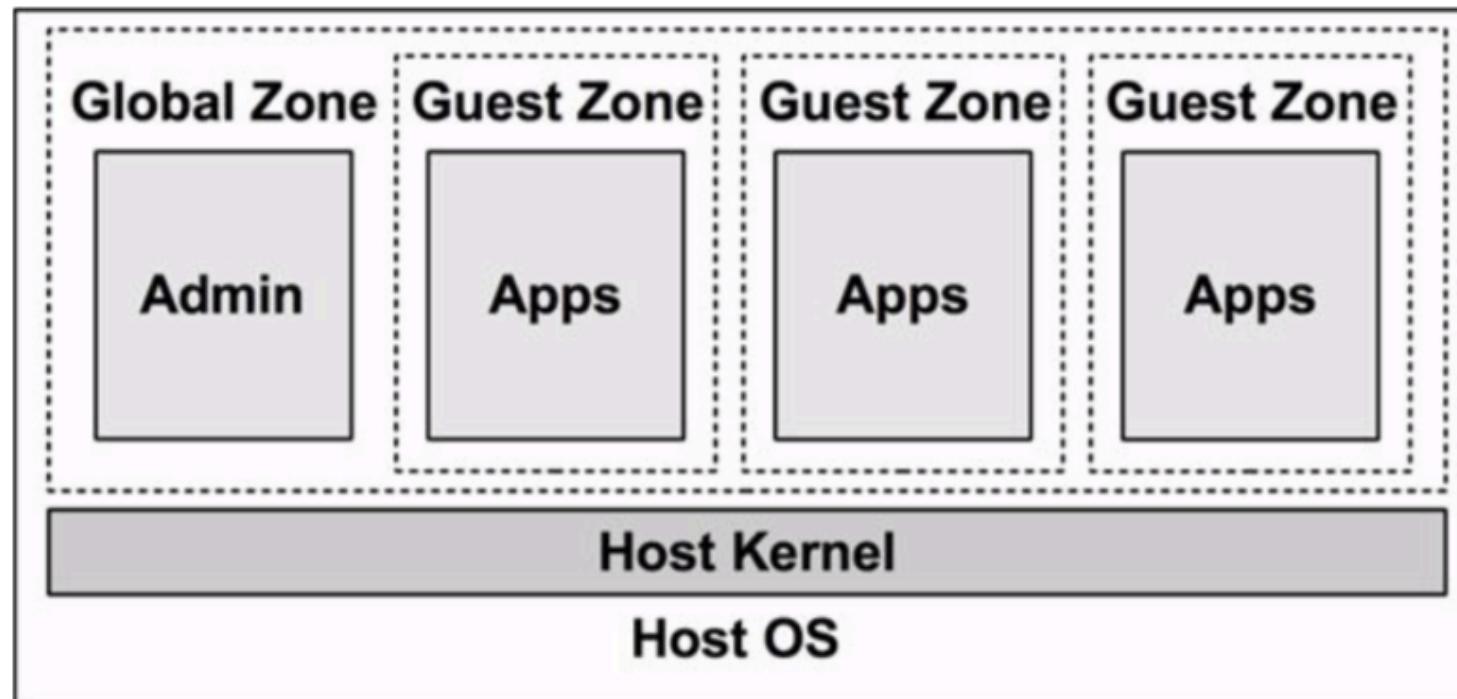
OS hôte et invités doivent le supporter  
et être compatibles!

## Tendances

abandon progressif de la paravirtualisation explicite  
car  
amélioration des performances des hyper viseurs  
support CPU pour raccourcir les chemins  
développement de piles logicielles et matérielles ad-hoc

# Virtualisation au niveau de l'OS

Séparation entre les applications assurée par le noyau de l'OS  
exemples d'implémentation : chroot, LXC, OpenVZ  
proche des containers



# Mémoire virtuelle

Traduction d'adresses virtuelles -> physiques  
faite par la MMU, cachée dans la TLB  
impacte réellement les performances

Machines virtuelles : 2 traductions  
virtuelle -> physique de l'invité (MMU de la VM)  
physique de l'invité -> physique de l'hôte (MMU de l'hyperviseur)  
=> ensuite cachée dans la TLB

Extensions dans les processeurs pour garder des traductions virtuel->  
physique de l'invité  
Intel : Extended Page Table (EPT)  
AMD : Nested Page Table (NPT)

Et augmentation de la taille des TLBs

# La virtualisation dans AWS

Historiquement Xen

Type 1  
et paravirtualisation éventuelle

Depuis 2017 développement d'un hyper viseur maison Nitro

Type 2  
basé sur KVM

utilise des processeurs Intel spécifiques pour le partage CPU, réseau,  
disque, etc.

plus besoin de processus utilisateurs pour les VMs  
performances proches du matériel direct  
impact favorable sur la consommation aussi

# Stockage

## Schéma classique

stockage local sans garantie de persistance mais performant

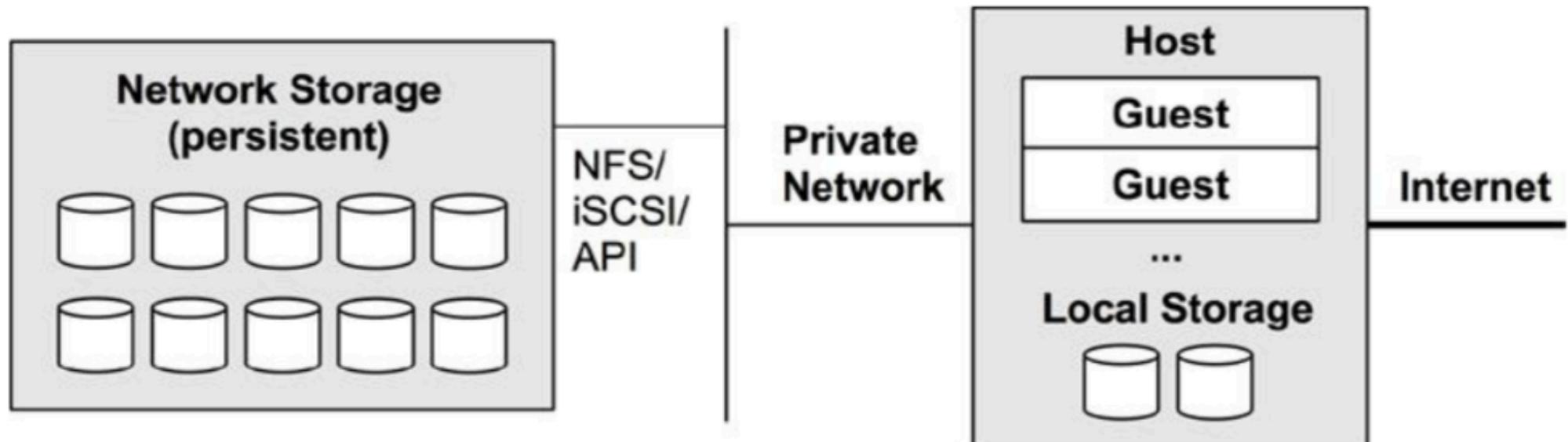
stockage distant sur un NAS

vu comme un block device

système de fichiers

base (clé, valeur)

(R)DBMS



# Les performances du stockage

Espace disponible (=  $f(\text{coût})$ )

Nombre d'opérations par seconde maximal pour des SSDs

Débit minimal garanti peut mettre en place de la réPLICATION

Longueur de la file d'attente des requêtes permet d'identifier les goulets d'étranglement arbitrage possible entre les VMs d'une machine physique

# Exemple du stockage sur AWS

S3 : système clé/valeur

Elastic Block Store (EBS)

accessible au niveau bloc (comme un disque local)

possibilité de

SSD ou HDD

avec qualité de service garantie ou non

prise d'images sauvages sur S3

Peut être un volume séparé qui persistera indépendamment de l'instance EC2 à laquelle il est rattaché

Elastic File System (EFS)

système de fichiers extensible (qui croît avec les besoins)

peut être monté et partagé par des machines EC2

# Réseau

Structuration en VLAN

Répartition en utilisant des affectations d'adresses associée aux découpages en VLAN

Configuration des équipements réseau pour partager la bande passante

Administration extérieure à la machine  
problèmes de désignation à prendre en compte dès la conception (K8s)

# Les serveurs privés virtuels

Proposés par OVH

Regroupent une configuration de serveur standard avec comme caractéristiques

CPU : nombre de cores, fréquence

Mémoire : taille

Disque : espace et vitesse d'accès (technologie), système de fichiers

réseau : volume de données échangées avec l'extérieur

système d'exploitation au choix

Peuvent être physiquement dédiés

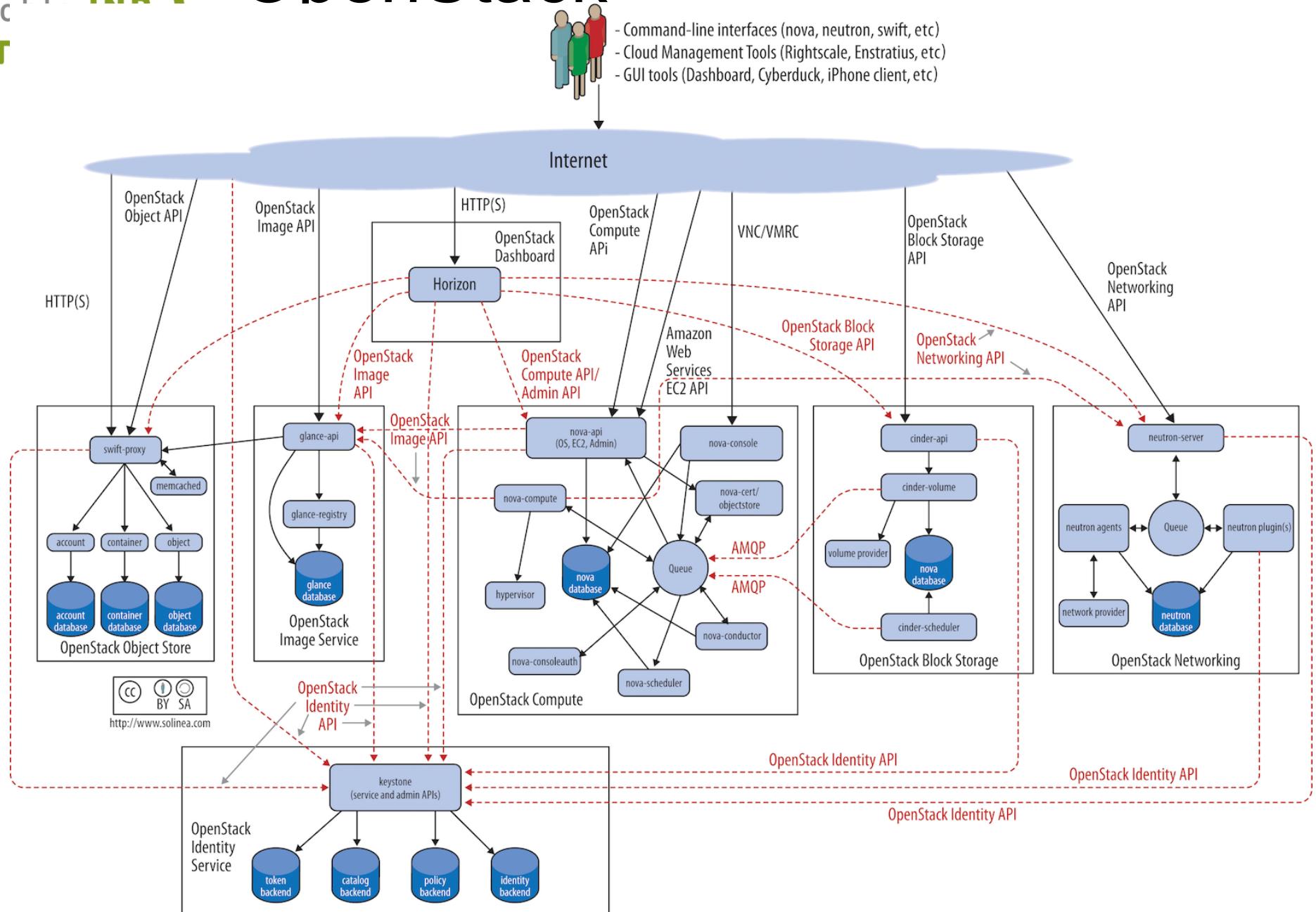
# OpenStack

Standard pour la conception de Cloud  
Géré par une fondation  
Initialement orienté IaaS  
avec des implémentations Open Source (python)  
protocole de communication

Montre l'ensemble des composants nécessaires à la création d'un cloud



# OpenStack



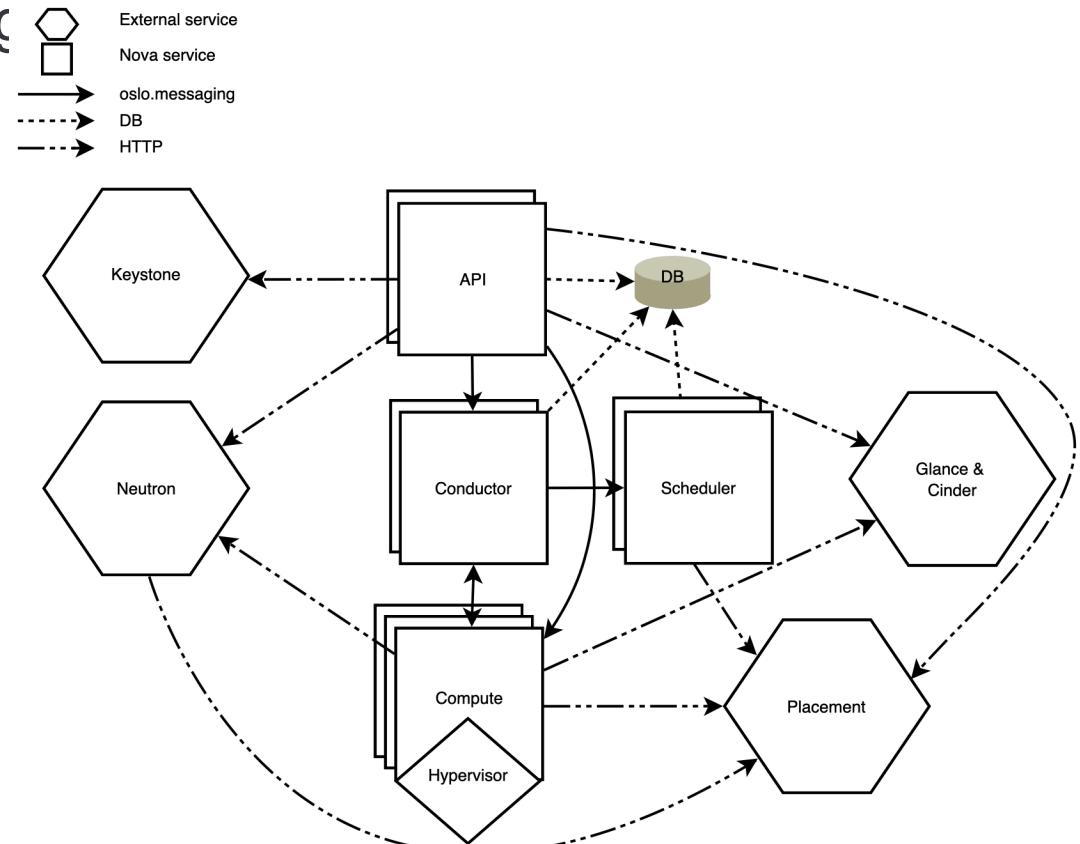
[docs.openstack.org](http://docs.openstack.org)

# Les composants OpenStack

## Nova (calcul)

gère l'allocation des noeuds ressources (machines virtuelles)  
 peut utiliser différents modèles de virtualisation et d'hyperviseurs  
 ensemble de démons à installer sur une machine Linux  
 conçu avec passage à l'échelle comme obsession

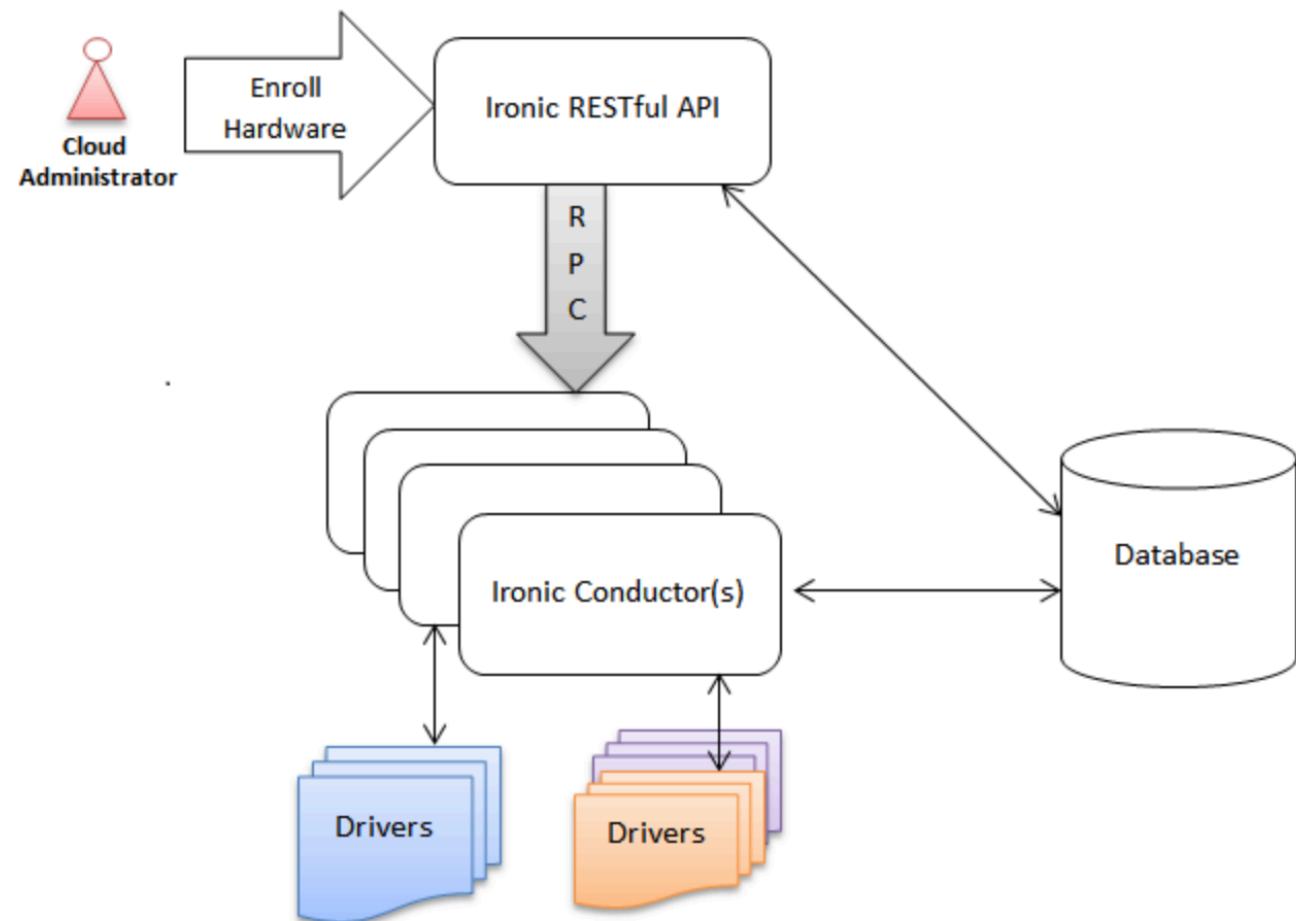
en particulier pour le monitoring



# Les composants OpenStack

## Ironic

gère l'allocation des machines physiques ( $\neq$ virtuelles)



# Les composants OpenStack

## Glance (images)

gestion des images disques permettant le démarrage des VMs  
peut aussi contenir des fonctionnalités de migration dynamique des VMs

## Neutron (réseau)

gestion des adresses réseau et machine  
avec modification dynamique des adresses et du routage  
(équilibrage de charge, tolérance aux fautes)  
mise en place de réseaux virtuels (VLAN et/ou VPN)

# Les composants OpenStack

## Cinder (stockage)

stockage par blocs (au-dessus d'une ressource existante)  
système intégré de prise d'images avec différents grains

cinder-api : routage des requêtes,  
cinder-volume : lecture et écriture sur le support sous-jacent  
cinder-scheduler : choisit où placer les ressources (allocation et écriture)  
cinder-backup : coordonne les snap-shots  
bus de message pour la communication

# Les composants OpenStack

## Swift (stockage)

stockage redondant d'objets

assure la distribution du stockage et la cohérence entre les versions  
fournit des fonctionnalités de plus haut niveau que Cinder

proxy server : point d'entrée assure le routage

ring : maintient la correspondance entre les noms des objets et leur  
localisation

replication : en charge de s'assurer que les données sont toujours  
disponibles (push based)

auditors : ausculte en permanence les données pour s'assurer qu'il  
n'y a pas eu de corruption

## Keystone (sécurité -> on en reparlera)

gestion des identités, single sign on

mapping sur les systèmes d'authentification des noeuds du système

# OpenStack

## Horizon (dashboard)

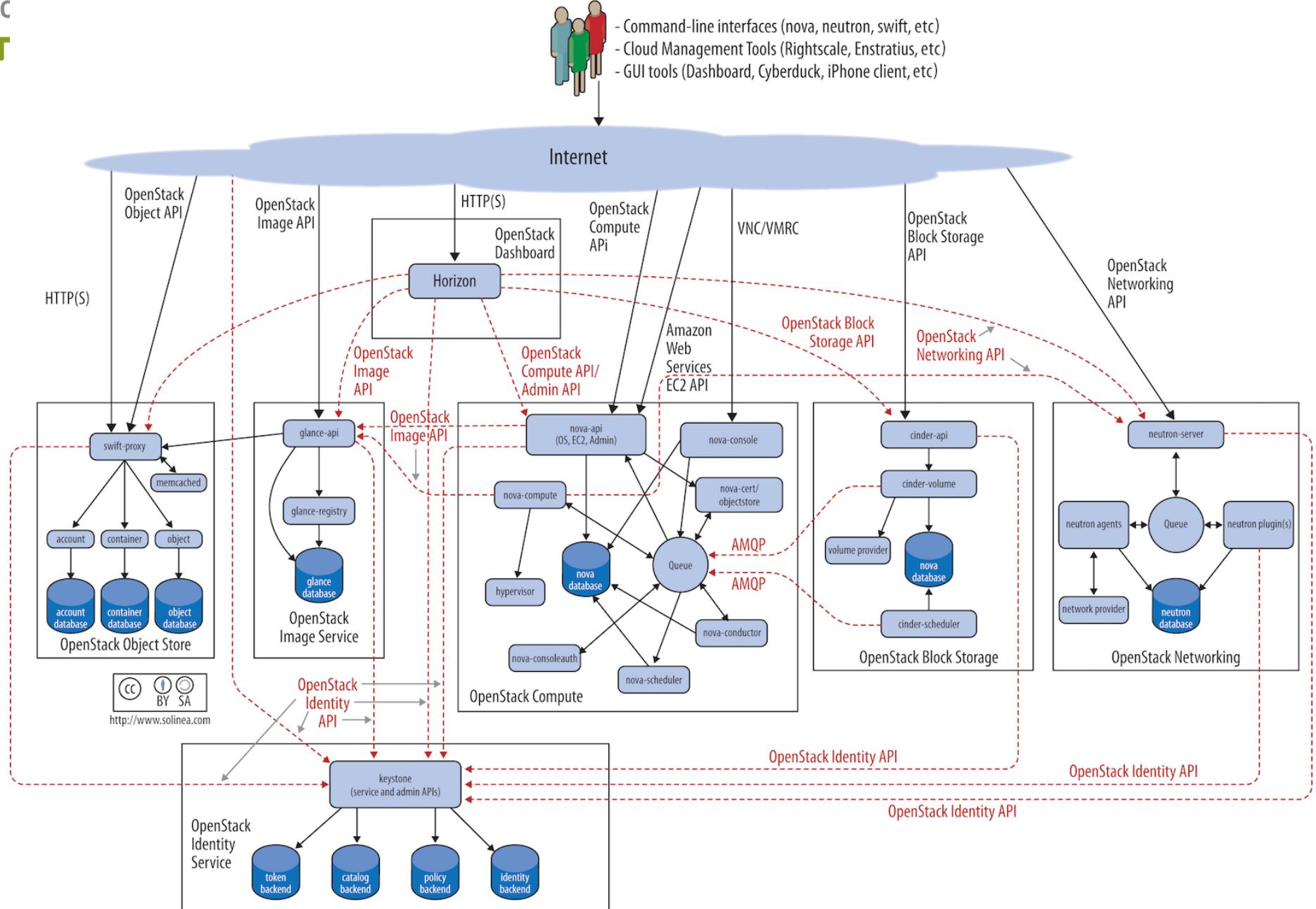
surveillance et reporting de l'état du système  
administration possible

## Heat (orchestration)

administration simultanée de plusieurs services  
utilise des scénarios à exécuter  
super langage de scripts  
ex : déploiement d'un wordpress

Intégration avec les autres solutions de cloud  
plus dans les outils et protocoles d'accès que dans les services proprement dit

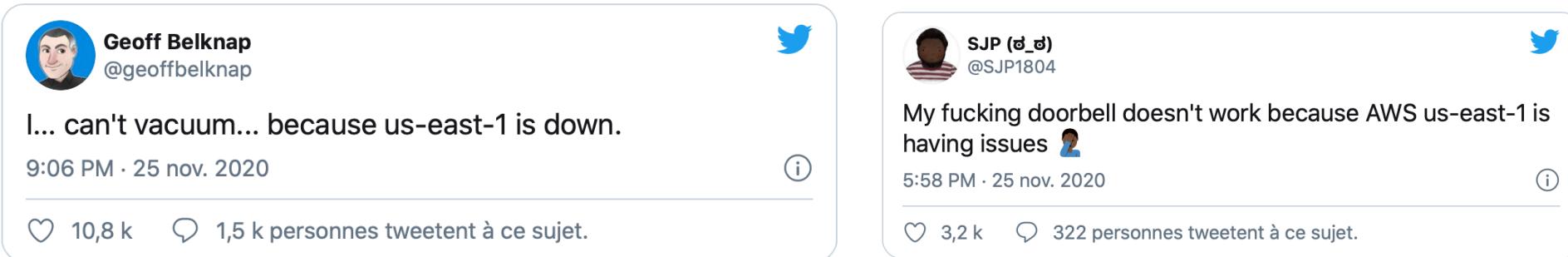
# OpenStack



[docs.openstack.org](http://docs.openstack.org)

# Panne de AWS (il y a 1 an...)

Problème sur Kinesis (stream de données)  
Affecte us-east-1



The image shows two tweets from the Twitter interface. The first tweet is from Geoff Belknap (@geoffbelknap) at 9:06 PM · 25 nov. 2020. It reads: "I... can't vacuum... because us-east-1 is down." The second tweet is from SJP (相助) (@SJP1804) at 5:58 PM · 25 nov. 2020. It reads: "My fucking doorbell doesn't work because AWS us-east-1 is having issues 🤦". Both tweets have engagement counts below them: 10,8 k likes and 1,5 k people tweeting about the first; 3,2 k likes and 322 people tweeting about the second.

1password, Adobe Spark, Coinbase, Flickr...

Fog computing

OpenFog

<https://aws.amazon.com/fr/message/11201/>