

# Journée Cloud

Yves Denneulin

# Objectifs de cette journée

Savoir ce qu'est le cloud  
aspects techniques et organisationnelles  
connaître le paysage industriel et institutionnel  
avoir du recul sur les solutions

DevOps

Sécurité et Souveraineté

Consommation énergétique

Après-midi : Introduction au cloud Google et programme de certification

Cours optionnel : Fondamentaux du cloud

# Définition

Cloud computing is a delivery model for technology-enabled services that provides on-demand access via a network to an elastic pool of shared computing assets (e.g. services, applications, servers, storage, and networks) that can be rapidly provisioned and released with minimal service provider interaction. The entire value can be bi-directionally scaled as needed to enable pay-per-use.

Le cloud c'est juste l'ordinateur de quelqu'un d'autre

# Historique

Idée aussi vieille que l'informatique moderne (time-sharing dans les 70s)

Essor du cloud commercial étroitement lié au développement d'Internet

Différentes phases

90s: Location de places dans un datacenter

1999 : Salesforce utilisable dans un navigateur (2004, IPO)

début des années 2000 : Architecture de grilles (recherche)

2002 : création de AWS

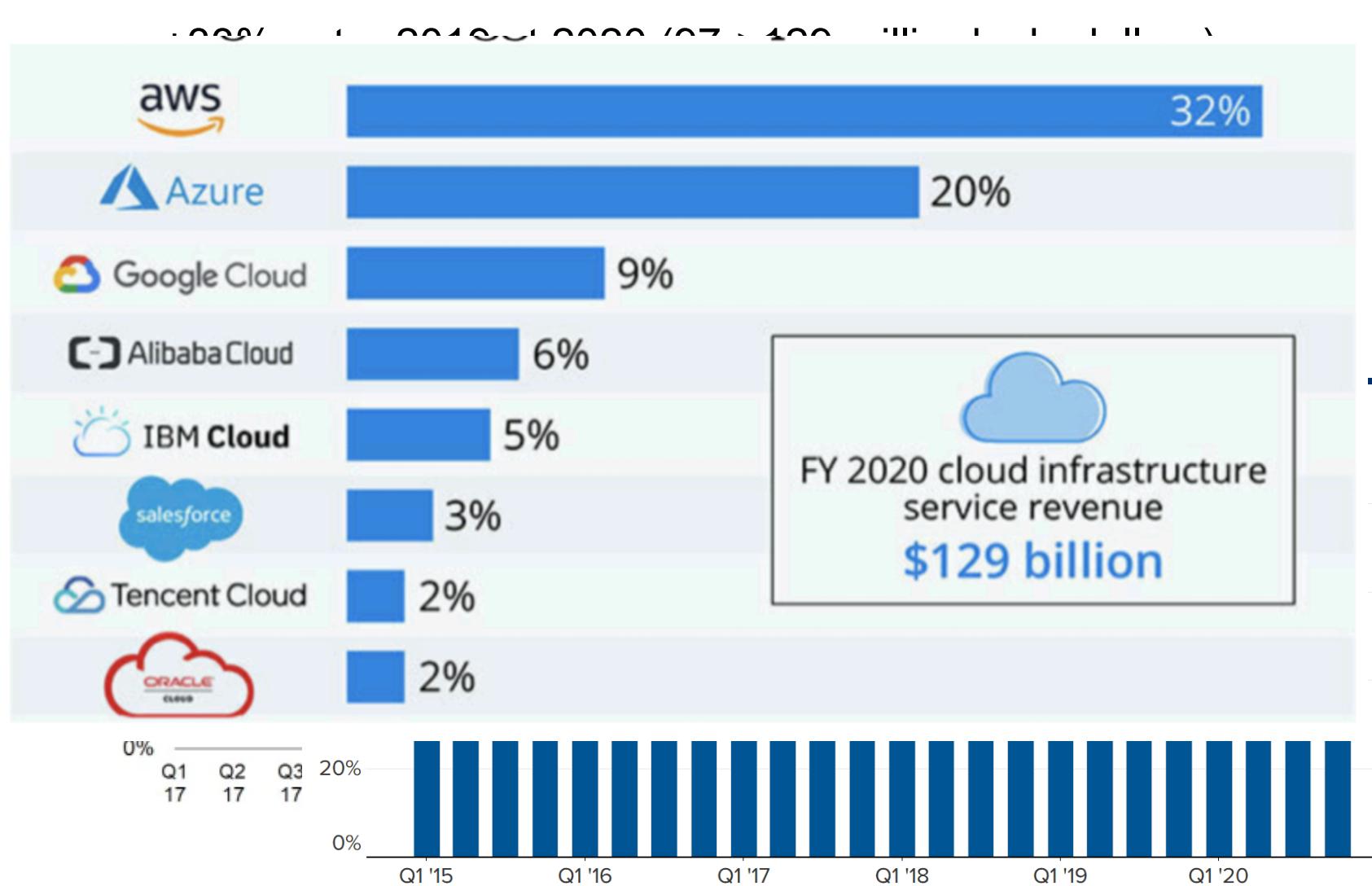
2006 : location de machines virtuelles sur AWS

2009 : services google aux entreprises par le web (gmail, goffice)

Maintenant solutions disponibles pour toutes les dimensions d'un système informatique

au départ à destination des start-ups, maintenant grand groupe aussi

# Marché majeur



SOURCE: Company reports



# Au delà de la technique...

- Toute activité informatique est devenue un service
  - y compris l'achat d'une machine
  - importance des contrats liant clients et fournisseurs
- Les devices ne sont plus qu'une interface
- Le rendement énergétique augmente
  - car utilisation au plus juste
  - vrai pour les infrastructures de base, pas pour les piles!
- La complexité des piles logicielles est cachée (+/-)
  - baisse des compétences nécessaires pour s'en servir
- Le passage à l'échelle est techniquement très facile
  - nivelle le paysage
    - gros facteur de croissance des start-ups
  - méthode de développement et d'intégration continue

# Impact sur les organisations

- Restructuration des organisations
  - cloud d'entreprise ou d'administration
  - réflexions au niveau des gouvernements
    - quel degré de mutualisation ?
    - quels facteurs de sécurité et de disponibilité ?
- Enjeu de la souveraineté
  - europe/USA/Chine
- Attention au Vendor lock-in
- Évolution très rapide, coût de mise à niveau à ne pas négliger

# Les promesses du cloud

- Agilité : passage à l'échelle transparent et rapide
- Protection des données
  - externalisation de la responsabilité
- Réduction des coûts (matériels, logiciels, humains, énergie)
- Recentrage sur le cœur de métier de l'entreprise
- Rapidité de déploiement
  - cloud accompagne le développement agile : devops
  - évolution vers le Serverless
- Pour les informaticiens, là où beaucoup de choses se passent (construction et utilisation)

# Exemples de fonctionnalités fournies par un cloud

- Services transparents aux utilisateurs
  - Siri, Alexa, Google Home
- Utilisation de logiciels
  - Office365
- Utilisation de services logiciels
  - bibliothèque de Machine Learning, chatGPT
- Espace de stockage
  - DropBox, onecloud, gitlab
- Hébergement de ressources physiques explicitées
  - Machine virtuelle, machine physique

rôle central de la notion de service

□ Machine Learning
Amazon SageMaker
Amazon Augmented AI
☆ Amazon CodeGuru
Amazon Comprehend Recommandations et l'exécution d'apprentissage
Amazon Forecast
Amazon Fraud Detector
Amazon Kendra
Amazon Lex
Amazon Personalize
Amazon Polly
Amazon Rekognition
Amazon Textract
Amazon Transcribe
Amazon Translate
AWS DeepComposer
AWS DeepLens
AWS DeepRacer

Quelle pourcentage d'activités faites vous en local vs cloud ?

Et vos proches ?

Quelle évolution en 10 ans ?

Paiement à l'usage ou au forfait

# Impacts du cloud

Organisationnel : *commoditisation* des ressources matérielles et logicielles

flexibilité accrue des configurations et des solutions prototypage plus rapide

évolution des compétences nécessaires pour développer et faire tourner un système complet

Financier : passage du modèle de l'investissement amortissable au modèle du service

côté clients : lissage, montée en puissance automatique

côté vendeurs : pour chaque client, prévisionnel plus facile à faire

# Pourquoi le cloud est possible ?

- Standardisation des services et de leur invocation
  - rôle majeur de l'open source
  - généralisation des APIs dans Amazon
- Amélioration des performances et ubiquité du réseau
- Micro-paiement
- Développement des techniques de virtualisation
  - machine, pile logicielle
  - réseau

# Catégories des clouds

Privé : tout est géré par l'entité  
déploiement et services fournis au reste de l'entreprise

Communautaire : application métier privée partagée entre plusieurs tiers

Public : ressources fournies par un tiers (*hyper-scalers*)  
dans un environnement partagé  
éventuellement avec des ressources dédiées  
avec une politique de sécurité propre, des données propres

Hybride : combinaison des deux  
exemple : une infrastructure en propre faisant appel à des services externes

*De facto*, tout devient hybride maintenant

# Différentes catégories de services

IaaS : infrastructure (matériel : machines + stockage + réseau)

CaaS : Container as a Service (~machine virtuelle)

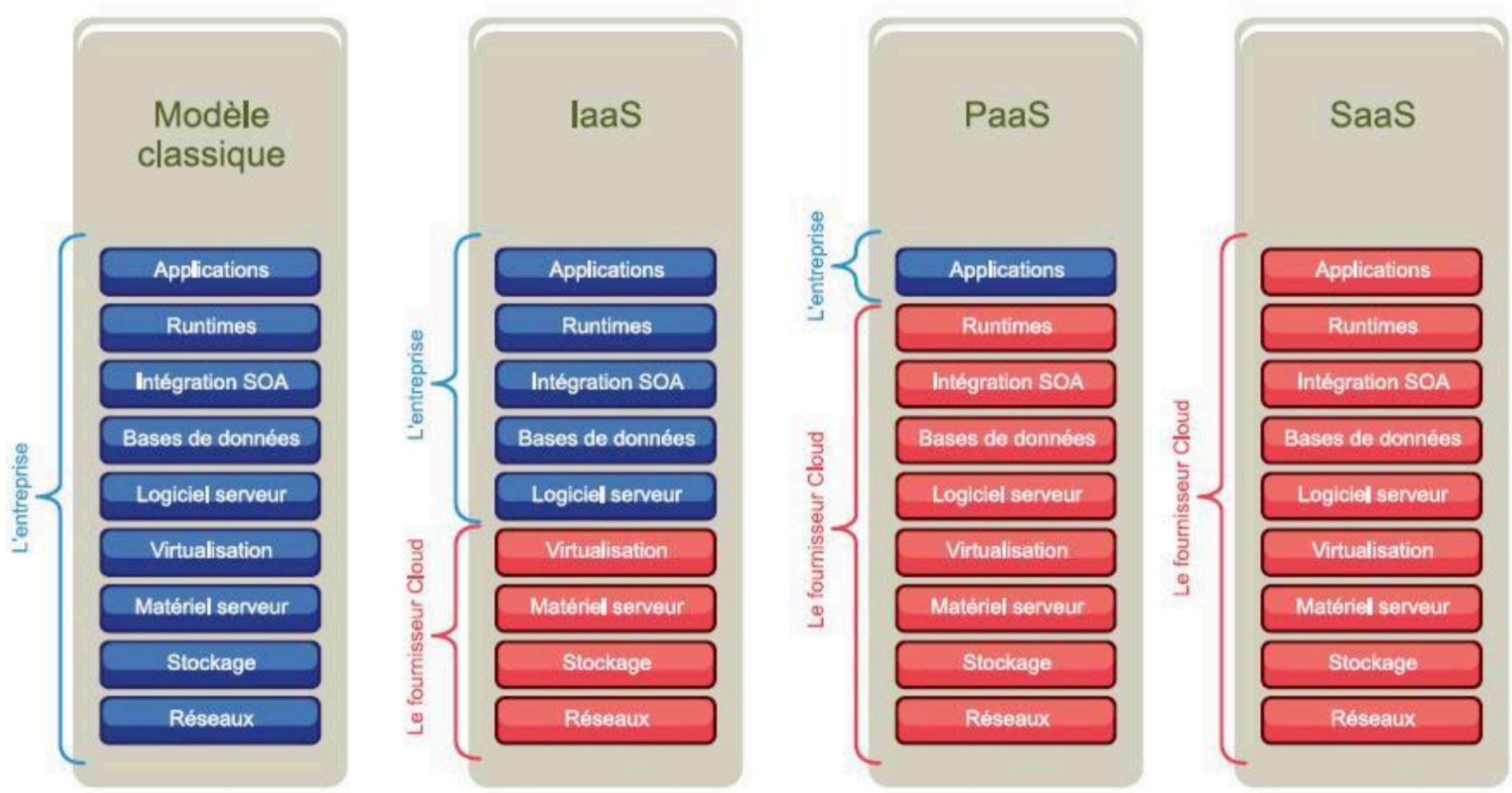
PaaS : plateforme (piles logicielles complexes)

SaaS : services (aux utilisateurs ou briques logicielles)

Point clé : standardisation des protocoles de communication et des piles logicielles (open source)

# Vision schématique

Source : SYNTÉC



# Catégories d'applications

## Historique

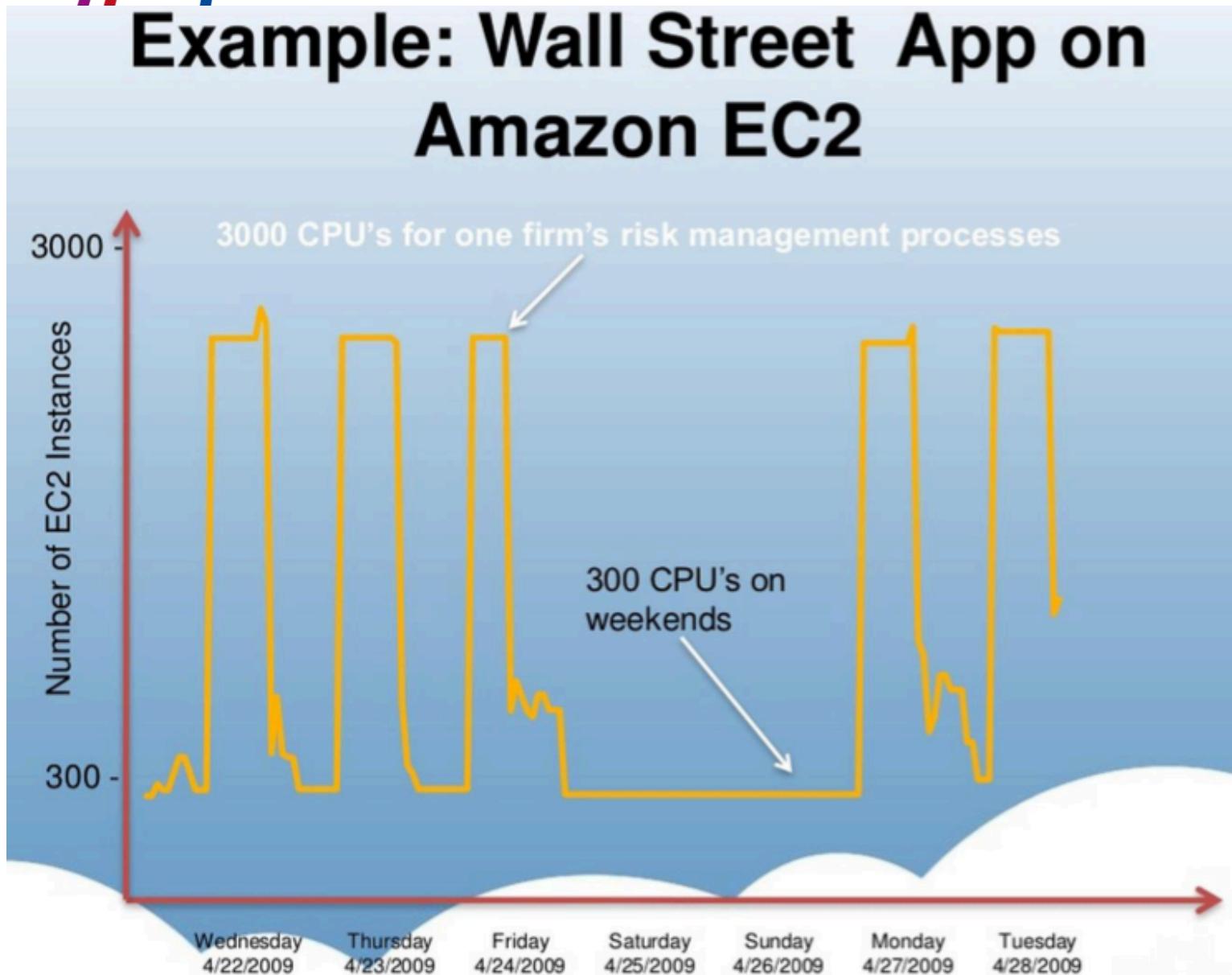
application bien connue stable dans son comportement  
tourne pendant des années sans modification  
dimensionnement effectué lors du déploiement  
stable dans le temps

## Moderne

distribué  
s'appuie sur des frameworks évolutifs  
cycle de vie à la semaine ou au mois  
peut avoir à supporter des workloads évolutifs

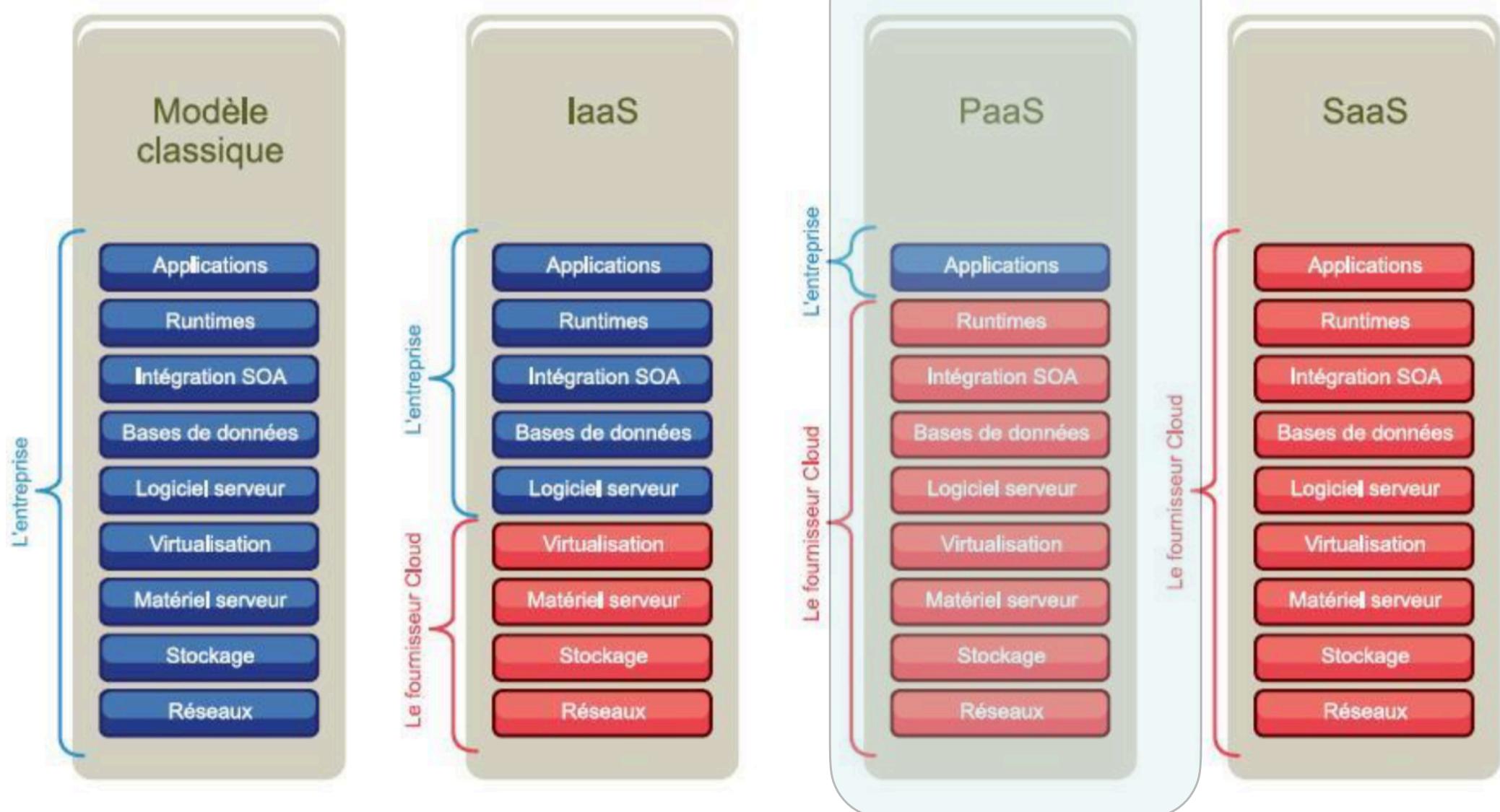
## Big Data & IA : passage à l'échelle rapide implique cloud

# Intérêt pour le fournisseur



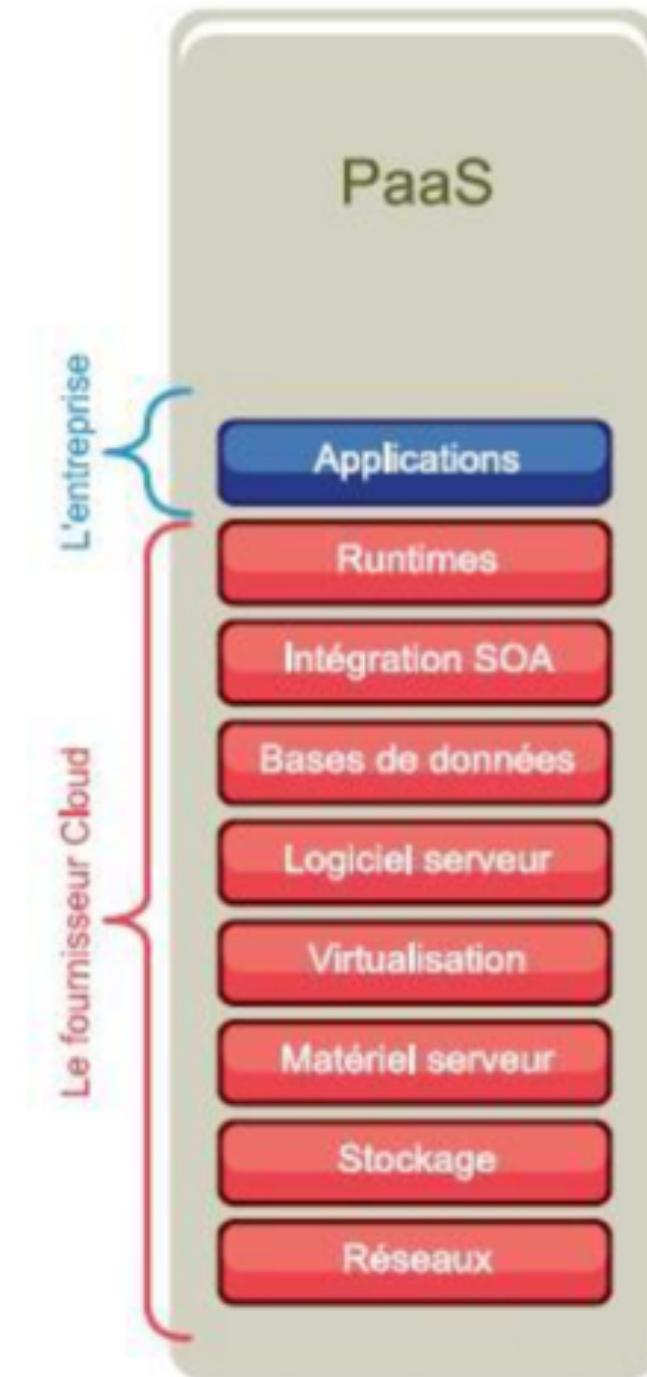
# Vision schématique

Source : SYNTEC



# PaaS

- Platform as a Service
- Fournit une pile complète
  - Préinstallée
  - configurée
- Services : déploiement et exécution d'une application

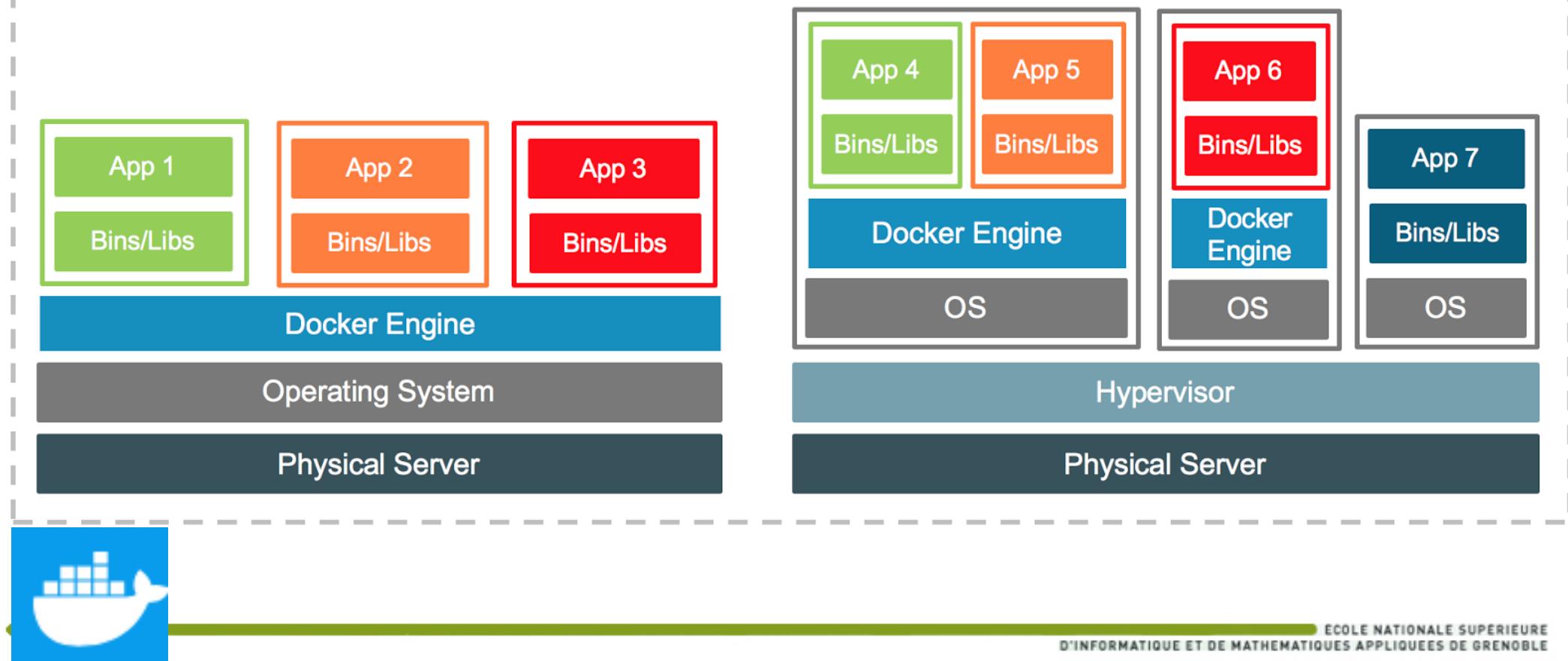


# Les containers

Idée : distribuer une application avec son environnement

Passage de l'IaaS au PaaS

Your Datacenter or VPC



# Docker

Solution de Container la plus répandue

Composée de  
un constructeur d'image de containers

un gestionnaire d'exécution des containers

une interface REST  
permet une automatisation de la gestion des containers

un repository de containers  
officiel mais possibilité d'en créer des locaux

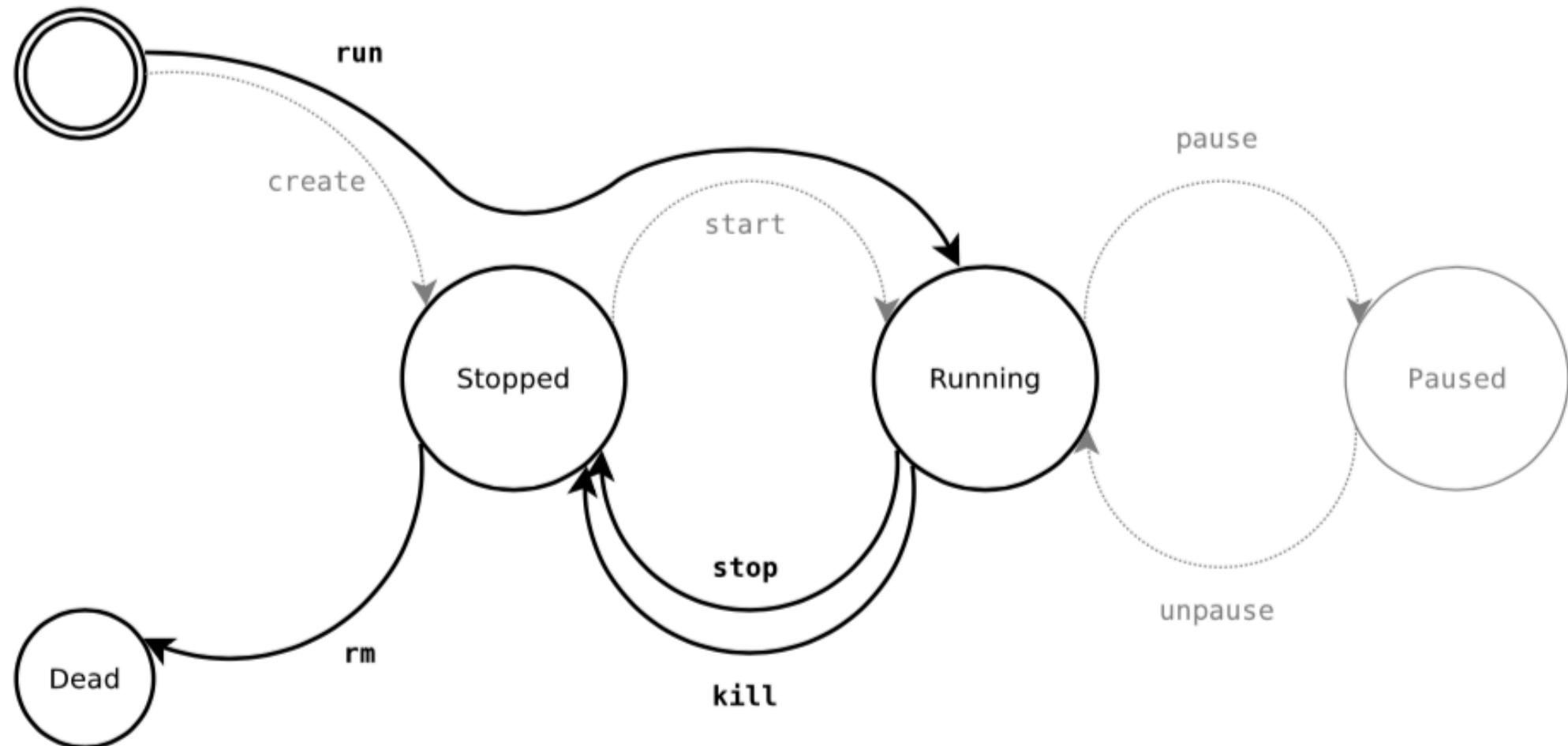
# Docker (2)

```
$ cat Dockerfile
#This is a sample Image
FROM ubuntu
MAINTAINER poftut@gmail.com

RUN apt-get update
RUN apt-get install -y apache2
```

<https://www.poftut.com/how-to-use-dockerfile-to-build-images-for-docker-containers/>

# Cycle de vie



Permet de normaliser et figer un environnement de développement et d'exécution

on ne conserve un container que quand toutes les versions des composants qu'il contient fonctionnent ensemble

Efficacité de gestion

déploiement en moins d'une seconde  
à partir d'une image (souvent en cache) qui peut venir d'un repository

S'appuie sur le noyau (et les librairies systèmes) de l'hôte et sur le support de container qui doit y être installé

# Container et cloud

Exécutable en IAAS  
il existe des instances avec Docker

Soumission directe de Containers  
Amazon Elastic Container Service  
envois de containers à exécuter  
facturation à la seconde + CPU et mémoire  
idéal pour : application longue et ponctuelle, build  
applications self contained

Azure Container Instances  
« serverless containers »  
MS Flow pour construire une application en utilisant des containers  
comme blocs

# Orchestration

Avantages des containers

- entité légère (configuration et déploiement)
- permet des déploiements rapides et adaptatifs

Un système d'informations peut être constitué de centaines de containers qui s'exécutent simultanément

- impossibilité d'administrer cela manuellement
- notamment les mises à jour
- et la tolérance aux fautes ?

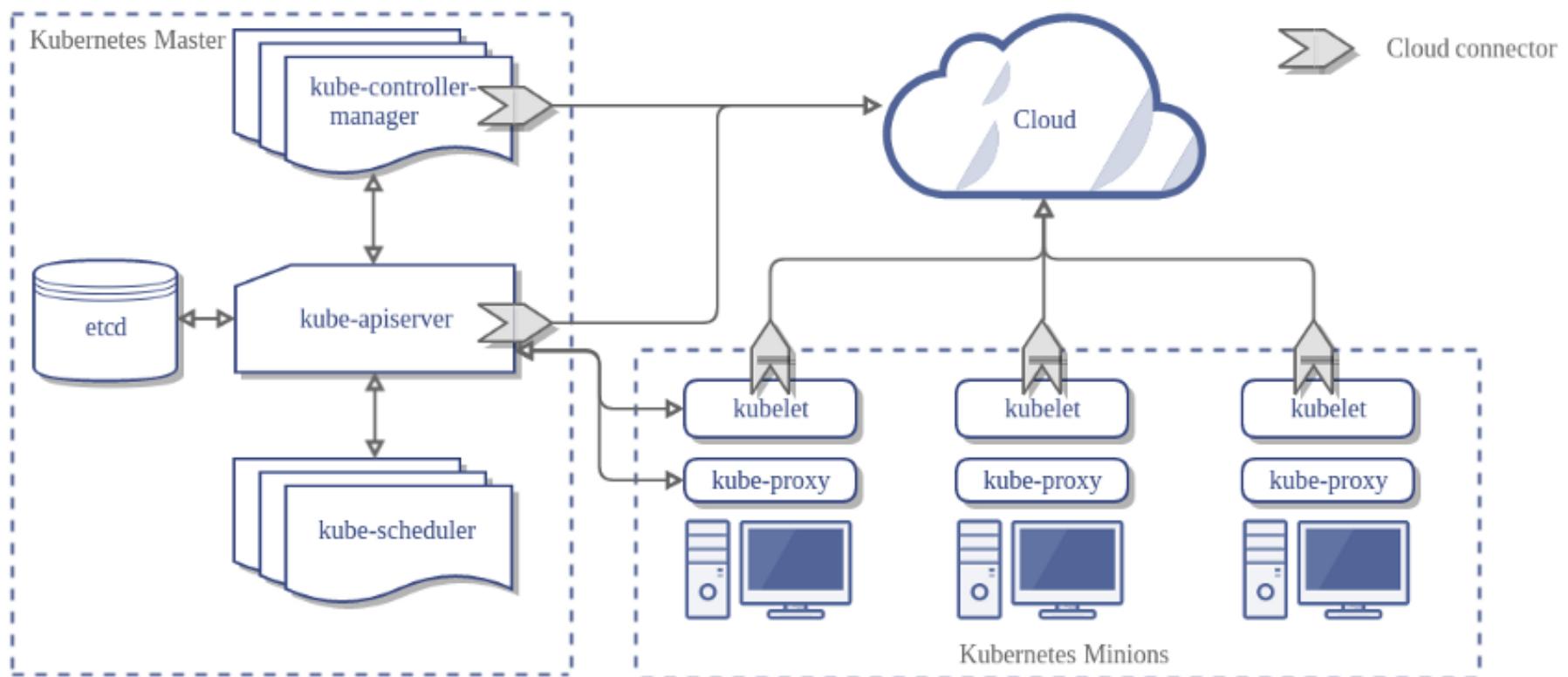
**L'orchestration** est essentielle

- différentes solutions logicielles (docker compose)

- un standard : Kubernetes

- Permet d'automatiser le déploiement et l'administration des applications structurées en containers
- open source par Google

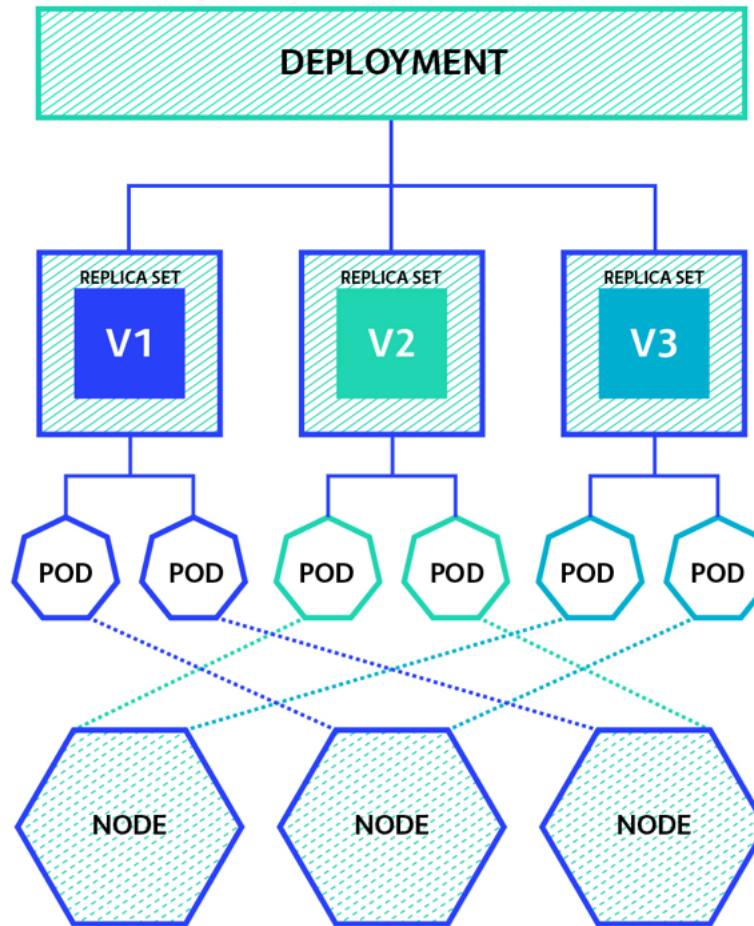
# Architecture globale



# Importance du versioning

Les pods permettent d'avoir des versions différentes mais cohérentes

Durée de vie d'un pod variable  
très court : micro services  
long : application persistente



# Description des applications

## Déploiement

décrit comment une application doit s'exécuter  
ensemble des containers la composant et comment les démarrer  
nombre de copies de chaque container  
K8S s'assure que le déploiement voulu est toujours respecté  
`kubectl run` -> crée un déploiement qui est ensuite respecté

K8S crée un objet replicaset en charge de s'assurer qu'il y en a toujours assez  
en charge d'un ensemble de pods tous identiques  
typiquement un pod pour chaque instance d'application (replication)

Voir les déploiements en cours dans un cluster K8S  
`kubectl get deployments`

# Cloud et Kubernetes

Création et administration d'un cluster Kubernetes

pile logicielle complexe

environnement très évolutif

de très nombreux outils pour aider le développeur/opérateur

IaC (Infrastructure as Code)

Utilisation d'un cluster clé en main (facturation à la ressource)

Google Kubernetes Engine (GKE)

par les créateurs de Kubernetes, la référence  
cluster autoscaling

Amazon Elastic Kubernetes Service (EKS)

sur plusieurs zones de disponibilité

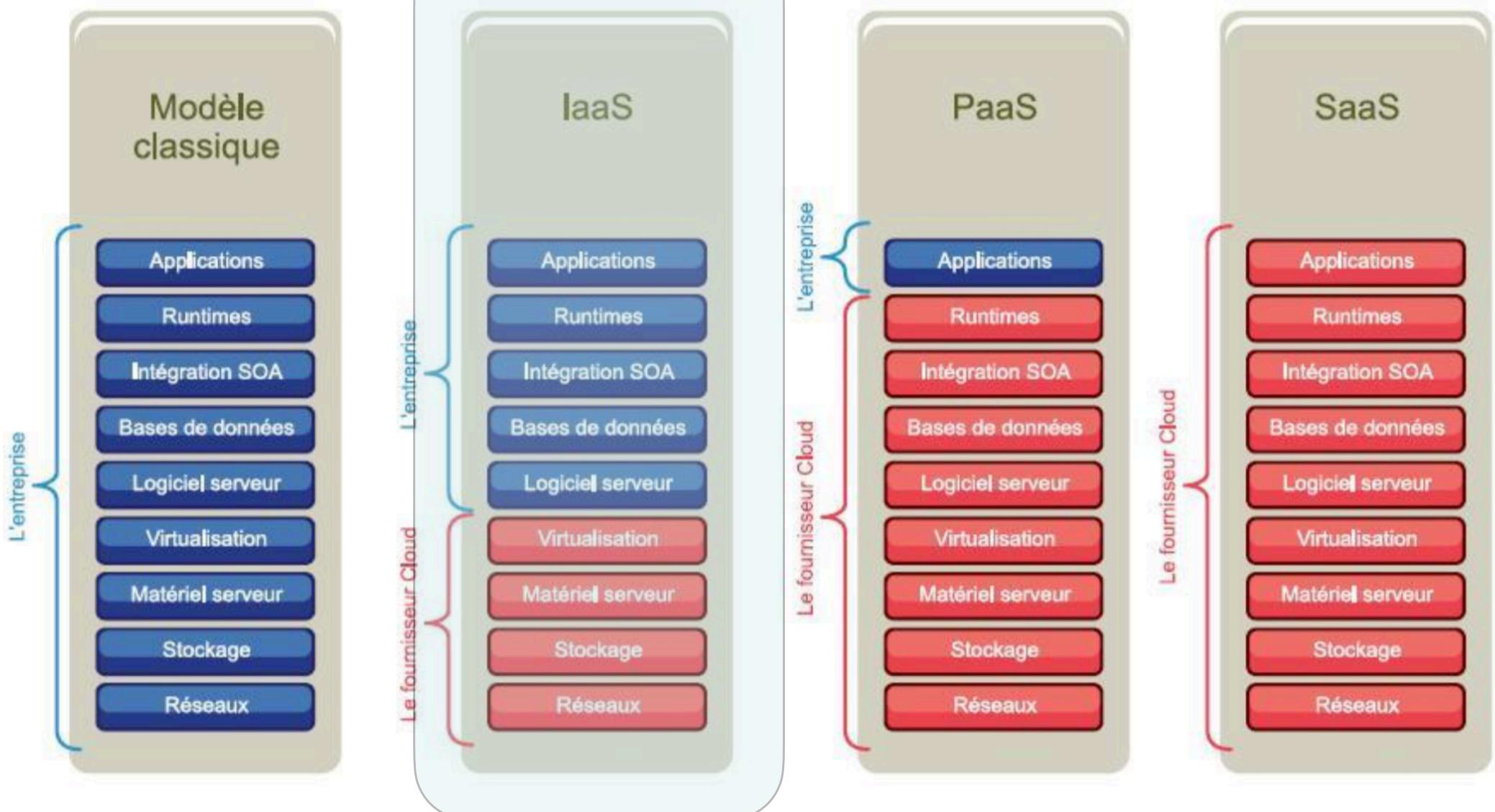
peut utiliser les instances spot (\$\$\$)

Azure Kubernetes Service

support de windows

# Vision schématique

Source : SYNTEC

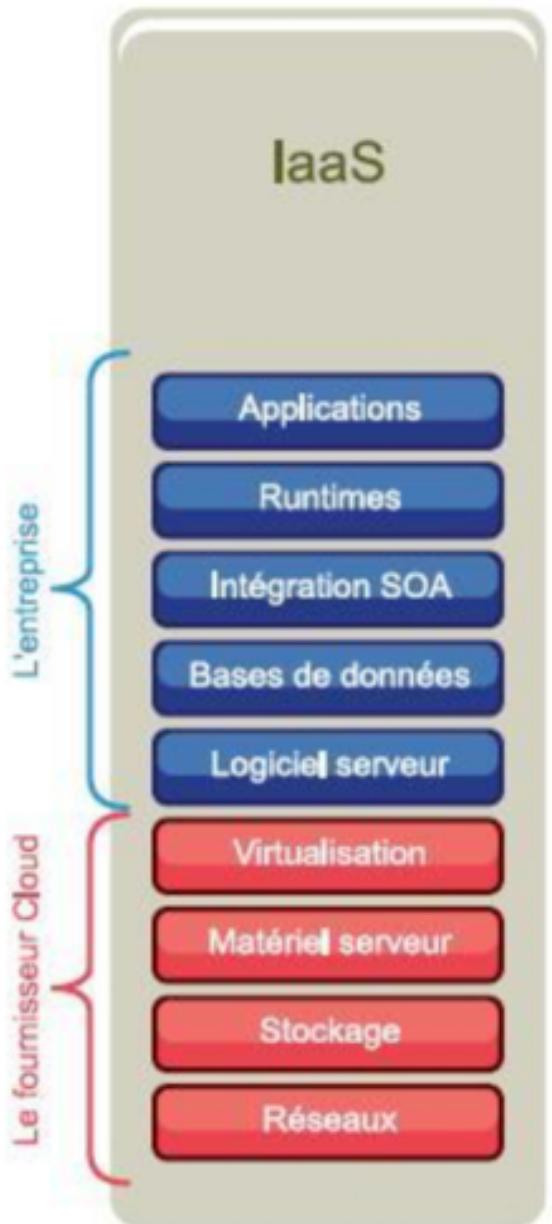


# Infrastructure as a Service

- Historiquement le premier type fourni
  - d'abord pour le HPC, calcul puis stockage
  - ISP avec hébergement, milieu des 90s
- Service fourni : stockage, couche de virtualisation, matériel serveur, réseau
- permet de ne pas avoir à acheter et gérer de matériels
  - la notion de ressource virtuelle est centrale
    - permet le partage au niveau du datacenter

# IaaS : implémentation

- Pas de standards dans le déploiement
  - vision « bas niveau » des ressources
- Notion de machines
  - virtuelles ou physiques
- Isolation (théorique?) entre les clients
  - disque
  - réseau
- Solution pour cloud privé : OpenStack, Xen, ...



# IaaS : services de base

- machines virtuelles (ou physiques)
- espace de stockage
  - vue fichier ou clé, valeur
  - stockage longue durée
  - ( R )DBMS
- Services réseau de base : DNS, IP, routage

# SaaS

- Exécution de services logiciels sur un site distant
- interface d'accès
  - applications dédiées
  - navigateur
- Importance de la standardisation des protocoles
  - REST, JSON, HTTP(S)
  - clé pour la standardisation et donc le déploiement de solutions
  - avec des implémentations de référence

# SaaS

- utilisation quotidienne : Office 365, Gmail, ...
- Implique qu'on ne maîtrise technique (presque) plus rien en terme d'implantation et de confidentialité
- OpenSaaS : Wordpress, Wiki, ...
  - code de l'application en Open Source
  - hébergement possible ou déploiement autonome
    - avec les risques de sécurité afférents

# Tendance à la migration vers le SaaS

- Serverless illustre bien cela
- Dans l'intérêt des fournisseurs de cloud
  - meilleure maîtrise de leur infrastructure
  - hébergement mutualisé
- modèle de coût plus opaque
  - meilleure marge

# Microservices

Serverless basé sur 2 concepts

FaaS : Functions as a Service

BaaS : Backend as a Service

Principe

plus aucune gestion de l'infrastructure, entièrement délégué au cloud  
transmission du code et/ou des applications  
exécution entièrement prise en charge par le prestataire cloud

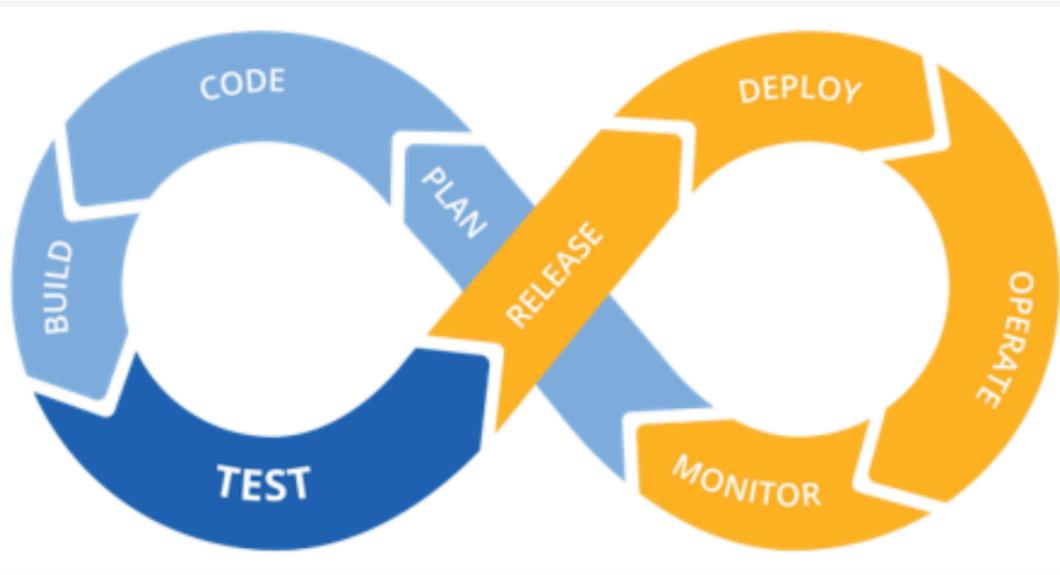
Adhère au concept devops

les développeurs sont aussi les administrateurs  
permet une grande agilité

à condition d'avoir les infrastructures adaptées (container,  
orchestration)

et la bonne culture aussi!

# Devops



- casser la séparation entre développement et déploiement
- du Devops au Devsecops

# Devops et cloud

- permet d'externaliser complètement la gestion de l'IT
- encapsulation d'une application avec son environnement
  - container
- facilité de déploiement
  - y compris en contexte hétérogène et géographiquement distribué
- Orchestration facilite
  - la coexistence des versions
  - le déploiement d'applications complexes
- développement du serverless

# Les pratiques Devops

- CI/CD : intégration continue, livraison/déploiement continu
  - tests permanents
- gestion des versions
  - de l'ensemble des éléments simultanément
- développement agile : cycle court
- infrastructure as code : description des besoins systèmes et réseaux des applications comme du code (K8s)
  - permet de passer du Dev à l'Ops
- gestion des configurations
- supervision continue

# Sécurité, souveraineté

- Confiance dans le fournisseur
- aspects techniques
  - authentification : organisation + mise en oeuvre
- Contexte législatif
  - secnumcloud

# SecNumCloud

- référentiel établi par l'ANSSI depuis 2016
  - permet de qualifier les acteurs (visa)
  - en cohérence avec les lois européennes (RGPD)
  - inspiré de ISO27001
- analyse approfondie de l'organisation, des procédures, de l'architecture et des configurations techniques
- peu de plateformes qualifiées
  - <https://www.ssi.gouv.fr/uploads/liste-produits-et-services-qualifies.pdf>
  - OVH, wordline : IaaS

- Association issue de la volonté de la commission européenne
  - un espace/marché européen commun pour les données
  - libre circulation entre les pays et les secteurs
  - respect des règles et valeurs européennes : claires et mises en oeuvre
- Focus sur les données : stockage, échange et accès
  - point clé pour la souveraineté

# GAIA-X (2)



- Gaia-X
  - définit une architecture et des règles associées
  - développe une implémentation de référence open-source
  - certifie des offres
- Gaia-X n'est pas
  - un organisme de standardisation
  - un développeur de plate-formes
- Le projet est dirigé par une association et l'architecture développée par ses membres
  - mais ne fournit pas de services

# GAIA-X (3)



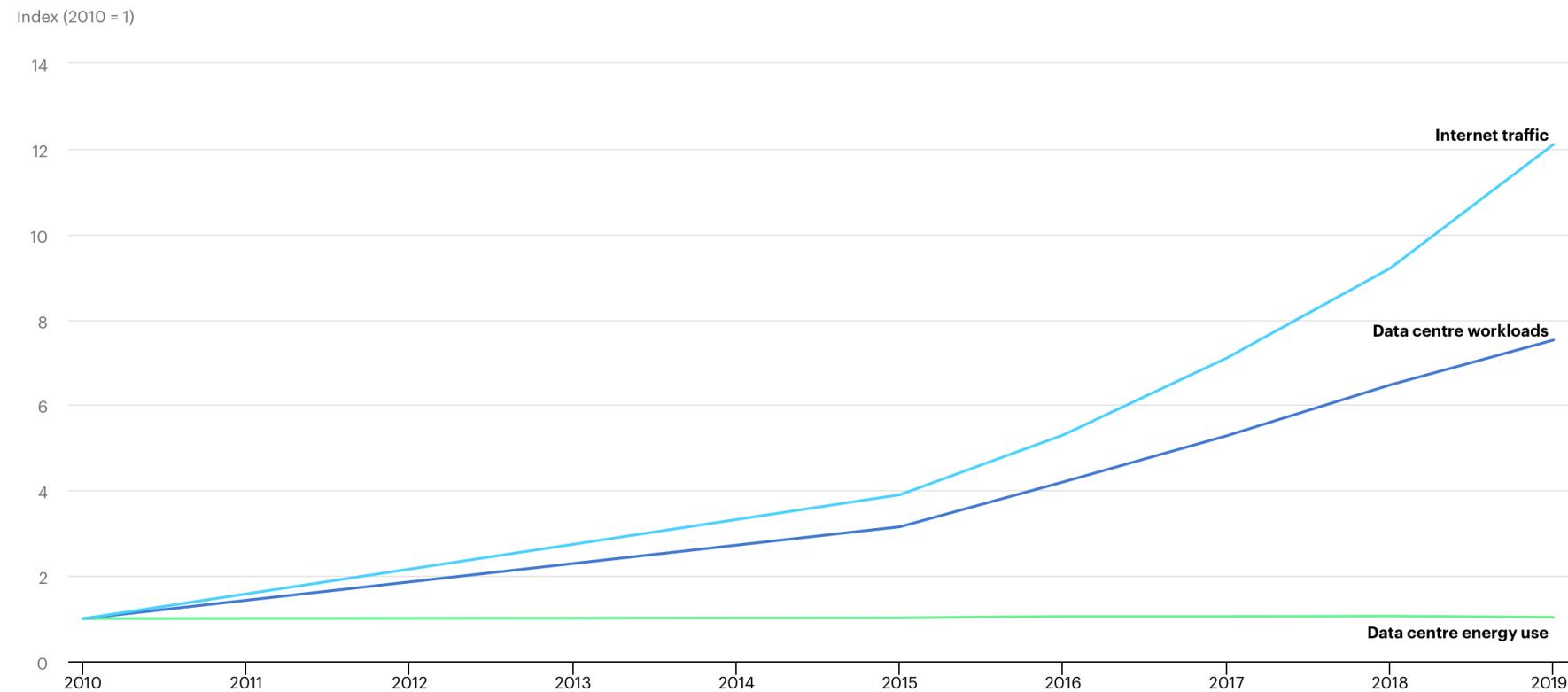
- Spécifie
  - des documents d'architecture
  - des spécifications de services
  - des documents de référence sur confiance et sécurité
- Développe
  - des frameworks open source
  - l'implémentation de référence des services fédératifs
- Certifie
  - des processus
  - des niveaux de qualité de service

# Performances énergétiques

# Quelques repères...

2020 : 400 TWh de consommation cumulée pour les data centers et les réseaux mondiaux

Global trends in internet traffic, data centre workloads and data centre energy use, 2010-2019



<https://www.iea.org/reports/data-centres-and-data-transmission-networks>

# Le Green500

Classement des machines les plus puissantes et les moins gourmandes

Green500 Data

TOP500		System	Cores	Rmax (PFlop/s)	Power (kW)	Energy Efficiency (GFlops/watts)
Rank	Rank					
1	29	<b>Frontier TDS</b> - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	120,832	19.20	309	62.684
2	1	<b>Frontier</b> - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	21,100	52.227
3	3	<b>LUMI</b> - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	1,110,144	151.90	2,942	51.629

# Progression vs 2021

TOP500			Cores	Rmax (TFlop/s)	Power (kW)	Power Efficiency (GFlops/watts)	
Rank	Rank	System					
1	301	MN-3 - MN-Core Server, Xeon Platinum 8260M 24C 2.4GHz, Preferred Networks MN-Core, MN-Core DirectConnect, Preferred Networks Preferred Networks Japan	1,664	2,181.2	55	39.379	
2	291	SSC-21 Scalable Module - Apollo 6500 Gen10 plus, AMD EPYC 7543 32C 2.8GHz, NVIDIA A100 80GB, Infiniband HDR200, HPE Samsung Electronics South Korea	16,704	2,274.1	103	33.983	
3	295	Tethys - NVIDIA DGX A100 Liquid Cooled Prototype, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100 80GB, Infiniband HDR, Nvidia NVIDIA Corporation United States	19,840	2,255.0	72	31.538	
4	280	Wilkes-3 - PowerEdge XE8545, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 80GB, Infiniband HDR200 dual rail, DELL EMC University of Cambridge United Kingdom	26,880	2,287.0	74	30.797	
5	30	HiPerGator AI - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Infiniband HDR, Nvidia University of Florida United States	138,880	17,200.0	583	29.521	

# Progression vs 2020

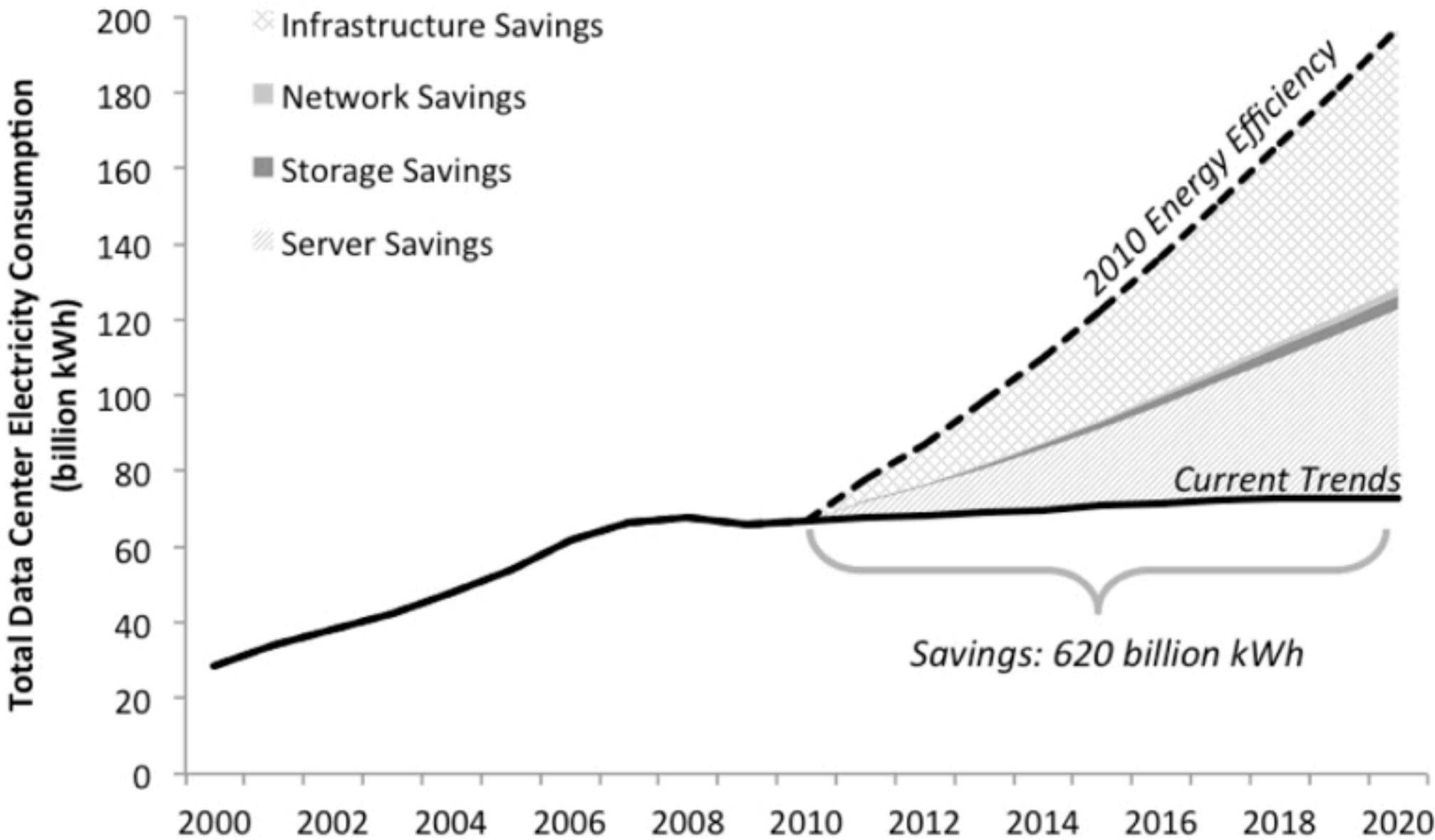
Classement de 2020 : 26 -> 39 : +50%

TOP500			Cores	Rmax (TFlop/s)	Power (kW)	Power Efficiency (GFlops/watts)
Rank	Rank	System				
1	170	<b>NVIDIA DGX SuperPOD</b> - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	19,840	2,356.0	90	26.195
2	330	<b>MN-3</b> - MN-Core Server, Xeon Platinum 8260M 24C 2.4GHz, Preferred Networks MN-Core, MN-Core DirectConnect, Preferred Networks Preferred Networks Japan	1,664	1,652.9	65	26.039
3	7	<b>JUWELS Booster Module</b> - Bull Sequana XH2000 , AMD EPYC 7402 24C 2.8GHz, NVIDIA A100, Mellanox HDR InfiniBand/ParTec ParaStation ClusterSuite, Atos Forschungszentrum Juelich (FZJ) Germany	449,280	44,120.0	1,764	25.008
4	146	<b>Spartan2</b> - Bull Sequana XH2000 , AMD EPYC 7402 24C 2.8GHz, NVIDIA A100, Mellanox HDR Infiniband, Atos Atos France	23,040	2,566.0	106	24.262

# Leçon du Green500

Clés pour la baisse de la consommation  
processeurs économies : les GPUs ont un très bon  
rendement énergétique  
mais ne sont pas des processeurs généralistes

“I am very much in favor of renaming it ‘high-efficiency computing’ instead of ‘high-performance computing since we really are the community that knows how to design fast algorithms, we know how to make really efficient implementations, how to efficiently parallelize algorithms – which is always necessary – and how to choose the right hardware to do it.”



# Leviers utilisés

Développement des machines virtuelles  
diminution du nombre total de serveurs

Développement de très grands data centers  
plus d'investissements sur l'infrastructure externe

Hausse de la température  
développement du free cooling

Amélioration de la gestion des ressources

# Bilan et leçons

Big is beautiful

faire le maximum de calcul dans des datacentre  
pour les fournisseurs être frugal est aussi être économe  
SaaS plus optimisable que IaaS

Poids des transferts  
difficile à évaluer de manière individuelle

Restriction des usages  
thème très délicat  
qui décide et comment ?  
des réflexions internes aux communautés qui commencent à prendre  
(IA)

Quid des externalités positives ?  
Enjeu : prolonger la durée de vie des équipements (rôle du logiciel)

# Cloud : problèmes et limites

- concentration -> faible tolérance à la défaillance
- risques de sécurité
- lieu d'hébergement (et législation associée)
- dépendances à un fournisseur
  - compétition pour une source de revenus réguliers
  - potentiellement prisonnier d'une solution
    - qui peut être supprimé du jour au lendemain!
- complexité de la facturation, prévisionnel difficile
- coût si haute utilisation des ressources
  - exemple de Netflix



Geoff Belknap  
@geoffbelknap



I... can't vacuum... because us-east-1 is down.

9:06 PM · 25 nov. 2020

10,8 k 1,5 k personnes tweetent à ce sujet.

