Cloud: accéder à des ressources (apps ou ses propres données) via internet sans avoir à gérer l’infrastructure et les services sous-jacents -> grande flexibilité et réduction des coûts.

apparu début 2000 car les entreprises avaient besoin d'une infrastructure informatique plus flexible et plus économique.

solutions traditionnelles -> coûteuses et difficiles à adapter à l’évolution des besoins.

cloud offre une infrastructure élastique et donc facilement adaptable aux besoins de l’entreprise.

Le modèle de cloud a été popularisé par Amazon, Google et Microsoft. Ces services ont permis aux entreprises de louer des ressources informatiques en fonction de leurs besoins et de ne payer que ce qu'elles utilisent.

**Principaux services**: calcul, stockage, sécurité des données.

Comparaison des différents cloud providers:

**AWS**: plus large collection de produits et services, tourné vers le cloud public

**faiblesses**: orientation quasi unique vers le **cloud public** (ressources partagées entre les différents utilisateurs, contrairement au **cloud privé** où les ressources sont gérées par l’organisation qui la demande) et **complexité** de la tarification.

**Azure**: Le principal avantage de Microsoft est que de nombreuses entreprises utilisent déjà leurs services (Office, SharePoint, Windows / SQL Server, etc.).

**forces**: écosystème windows et un cloud hybride

**faiblesses**: support technique et la documentation disponible

**GCP**: dernier arrivé, investit massivement pour séduire les entreprises.

**forces**: ML, DL, IA, data analyse et le support de l’open source

**faiblesse**: infrastructure bcp moins dév que celles d’aws et d’azure.

Différentes gammes de services (IaaS, PaaS, SaaS)

IaaS:

fournit les serveurs, stockage, réseaux, systèmes d’exploitation sur le cloud public.

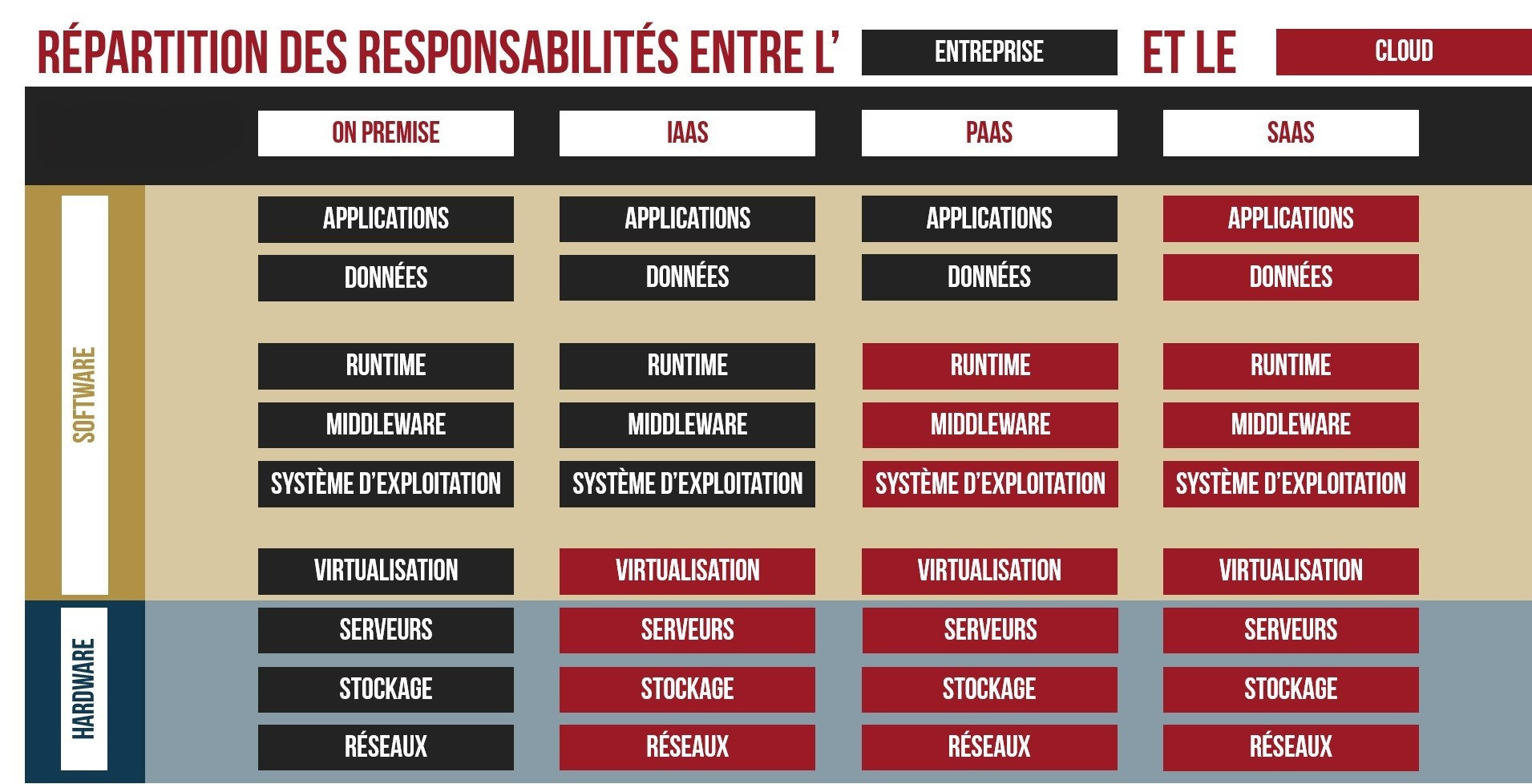
les clients peuvent créer et gérer leurs propres machines virtuelles, installer leurs propres apps et gérer leur propre stockage.

PaaS:

fournit plate-forme de dév et d’exécution d’apps, clients peuvent développer et déployer leurs propres apps.

SaaS:

fournit apps hébergées sur le cloud public, clients n’ont pas besoin d’installer ou de gérer l’app eux-mêmes.



**pay as you go: on ne paye que ce qu’on utilise (on paye au Go consommé, à l'utilisateur, à l'heure, etc**

TCO: Total Cost of Ownership, utilisé pour comparer les coûts d'exploitation de différents services de cloud computing.

**Comparaison coûts DC vs Cloud:**

Cloud: utilisateurs n’ont aucune influence sur la structure tarifaire des fournisseurs.

Cloud -> OpEx (dépenses opérationnelles)

DC -> CapEx (coûts d’acquisition et de mise en service) (capital de départ + imptt)

avantage → avec son propre DC, les coûts mensuels ainsi que les capacités requises peuvent être planifiés.

Qu'est-ce que le ML Ops ?

pratique de dév → automatiser et optimiser les processus de création, formation, déploiement, surveillance et maintenance des modèles de ML.

Comment est-il apparu ?

apparu en réponse aux challenges de déploiement d’apps ML à grande échelle

Sur quoi se base-t-il ?

→ pratiques DevOps (automatisation des tests, intégration continue de nvlles fonctionnalités dans le code existant, livraison continue (automatisation process de déploiement), collaboration)

→ automatisation

→ sur infrastructure cloud

→ gestion de version (pour suivre les modifications apportées aux modèles)

Quelles sont les solutions techniques populaires pour les grandes étapes du MLOps ?

**dév**: jupyter, pycharm

**formation**: tensorflow, pytorch

**déploiement**: docker, kubernetes

**surveillance**: prometheus, grafana

**gestion de version**: git, github

Trucs en plus:

octoml: voir hardware recommandé pour un modèle donné (entre autre)

cloud automl: permet aux développeurs avec peu d’exp. en ML d'entraîner des modèles de haute qualité. (Vertex AI)