Modèles de services

M346 — Concevoir et réaliser des solutions cloud

Jérôme Frossard

EPAI

15 août 2024

Introduction

L'émergence du cloud computing a transformé la manière dont les entreprises gèrent et exploitent leurs infrastructures IT.

Cette présentation aborde les points suivants :

- Les fondements des modèles de services cloud : IaaS, PaaS, et SaaS.
- Le partage des responsabilités entre le fournisseur et le consommateur.
- Des cas d'utilisation courants pour chaque modèle.
- Les principaux inconvénients de ces modèles.
- Un bref aperçu des nouveaux modèles de service XaaS (*Anything as a Service*)
- Un bref aperçu de l'informatique sans serveur (*serverless computing*)

Plan

Plan

Qu'est-ce qu'un modèle de service ?

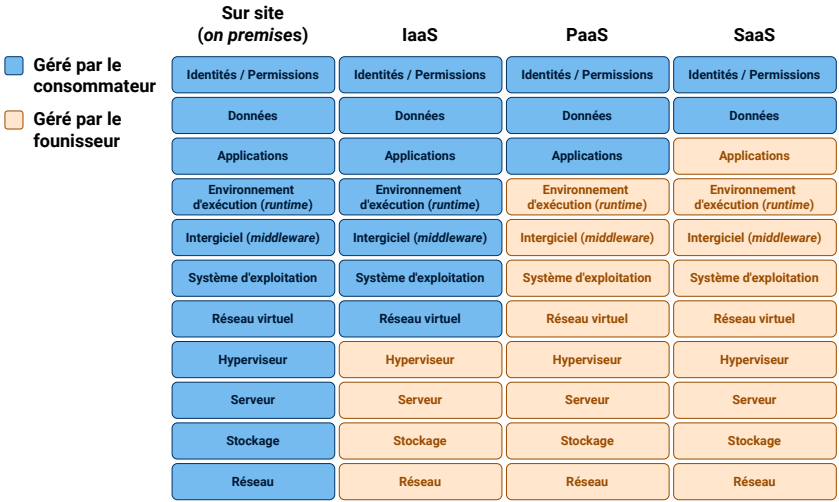
Un **modèle de service** définit :

- La répartition des responsabilités et des tâches de gestion,
- Entre un **fournisseur** de service cloud et le **consommateur** de service cloud,
- Par rapport à une infrastructure informatique sur site (*on premises*)

Dans son modèle de référence, le NIST définit les trois modèles suivants :

- SaaS (*Software as a Service*)
- PaaS (*Platform as a Service*)
- IaaS (*Infrastructure as a Service*)

Partage des responsabilités selon le modèle de service



Responsabilités partagées

Quel que soit le modèle de service, certaines responsabilités sont toujours partagées et sont, à différents niveaux, à la fois celles du fournisseur et celles du consommateur.

C'est notamment le cas de :

- La sécurité
- La protection des données
- La sauvegarde et la restauration

Responsabilités partagées — Sécurité et protection des données

Sécurité :

- Le fournisseur gère la sécurité de l'infrastructure physique, des serveurs, du réseau, et des logiciels fournis (y compris les mises à jour et les correctifs de sécurité).
- Le consommateur gère les comptes d'utilisateurs, le contrôle d'accès aux données et aux applications, ainsi que la sécurité des logiciels et des données sous son contrôle.

Protection des données :

- Le fournisseur assure le respect des normes de protection des données au niveau de l'infrastructure et pour les données personnelles ou sensibles sous son contrôle.
- Le consommateur s'assure que les mesures prises par le fournisseur sont adéquates et respectent les normes en vigueur dans sa région et dans son domaine d'activité, et assure le respect des normes de protection des données pour les données personnelles ou sensibles sous son contrôle.

Responsabilités partagées — Sauvegarde et restauration des données

Sauvegarde et restauration des données :

- Le fournisseur assure la continuité des opérations et la reprise après sinistre de son infrastructure, par exemple en cas de panne majeure ou de catastrophe naturelle.
- Les solutions de stockage sont généralement redondantes et la probabilité d'une perte de données à cause d'une défaillance est faible. Mais cela ne protège pas contre les ransomwares ni contre les autres types de cyberattaques, ou les erreurs humaines.
- Les fournisseurs offrent diverses solutions de sauvegarde et de restauration en SaaS, adaptées à différentes charges de travail, et des options de stockage appropriées.
- La responsabilité de la mise en place des procédures de sauvegarde, de leur vérification et de leur restauration incombe généralement au consommateur (dans le cas du SaaS, le consommateur doit s'assurer que de telles procédures sont en place).

Plan

SaaS — Software as a Service

Le SaaS (Software as a Service) est un modèle de distribution de logiciel où les applications sont hébergées dans le cloud et exploitées en dehors de l'organisation ou de l'entreprise par un fournisseur de service cloud.

Une application SaaS est accessible à la demande via une connexion Internet. Le programme client peut être :

- Un navigateur web (*thin client*),
- Une application mobile,
- Une application de bureau (*thick client*).

L'accès via un navigateur web (le plus courant) ne nécessite ni installation ni maintenance de la part de l'utilisateur ; l'application est immédiatement prête à l'emploi.

Responsabilités du fournisseur et du consommateur

Le fournisseur gère :

- Le centre de données (locaux, électricité, refroidissement) et l'infrastructure réseau.
- Les serveurs physiques, le stockage, et les hyperviseurs.
- Les serveurs virtuels et leur système d'exploitation.
- Les logiciels intermédiaires (*middleware*) et les applications.
- La sécurité et la mise à jour du logiciel intermédiaire et des applications.

Le consommateur gère :

- L'intégration avec d'autres services et applications utilisées par l'entreprise.
- L'administration des données et leur sécurité.

Cas d'utilisation du SaaS

Parmi les principaux cas d'utilisation du SaaS, on peut mentionner :

- **Productivité et collaboration.** Utilisation de suites bureautiques en ligne (ex. Google Workspace, Microsoft 365) pour faciliter la collaboration entre les équipes.
- **ERP (Enterprise Resource Planning).** Utilisation de solutions ERP (ex. SAP, Oracle) pour gérer les processus métiers essentiels, tels que la comptabilité, la gestion des stocks, et les ressources humaines.
- **Gestion de la relation client (CRM).** Utilisation de systèmes CRM (ex. Salesforce) pour gérer les interactions avec les clients et améliorer le service client.
- **Plateformes d'analyse de données**. Permet de collecter et traiter des données pour en extraire les informations pertinentes et faciliter la prise de décision.
- **Marketing digital.** Utilisation de plateformes de marketing pour gérer des campagnes publicitaires, l'analyse de données et la gestion des réseaux sociaux.

Inconvénients du SaaS

Parmi les éléments à prendre en considération lors de la mise en œuvre d'une solution SaaS, on peut mentionner :

- **Dépendance à Internet.** Une connexion Internet instable ou momentanément interrompue peut affecter l'accès aux applications SaaS.
- **Personnalisation limitée.** Les options de personnalisation peuvent être limitées, car les applications sont conçues pour être standardisées.
- **Sécurité des données.** Les données sont stockées chez le fournisseur, ce qui peut poser des problèmes de confidentialité et de conformité réglementaire, selon les politiques de sécurité du fournisseur.
- **Enfermement propriétaire.** Le transfert de données et d'applications vers un autre fournisseur de SaaS peut être complexe et coûteux.

Plan

PaaS — Platform as a Service

Le PaaS (Platform as a Service) est un modèle où le fournisseur gère l'infrastructure matérielle et logicielle nécessaire pour développer, tester, et déployer des applications.

Les équipes de développement peuvent ainsi se concentrer sur leur cœur de métier.

Quelques exemples de composants PaaS :

- Environnements d'exécution (Java, Spring Boot, .NET, Node.js, Python, etc.),
- Plateforme d'orchestration de conteneurs,
- Stockage objet,
- Agents de messages (*message broker*),
- Bases de données SQL et NoSQL,
- Répartiteurs de charge (*load balancer*),
- Passerelles d'API (*API gateway*),
- Chaînes d'intégration continue et de déploiement continu (*CI/CD pipeline*).

Responsabilités du fournisseur et du consommateur

Le fournisseur gère :

- Le centre de données (locaux, électricité, refroidissement) et l'infrastructure réseau.
- Les serveurs physiques, le stockage, et les hyperviseurs.
- Les serveurs virtuels et leur système d'exploitation.
- La sécurité et la mise à jour du logiciel intermédiaire.

Le consommateur gère :

- Le choix et la configuration des options de la plateforme.
- Le développement, le déploiement, et la gestion des performances des applications.
- La sécurité des applications.
- La conformité aux normes de protection des données.

Cas d'Utilisation du PaaS

Parmi les principaux cas d'utilisation du PaaS, on peut mentionner :

- **Test et développement.** Facilite et accélère le développement et le déploiement d'applications web et mobiles.
- **Intégration continue et déploiement continu.** Mise en œuvre de chaînes d'intégration continue et de déploiement continu (*CI/CD pipelines*).
- **Microservices.** Facilite le déploiement et la mise à l'échelle d'applications basées sur des conteneurs, notamment des microservices.
- **Services backend.** Facilite et accélère le déploiement d'API REST.
- **Traitement en temps réel.** Facilite la réalisation d'applications nécessitant le traitement de données en temps réel, comme des systèmes d'IA ou des jeux.

Inconvénients du PaaS

Parmi les éléments à prendre en considération lors de la mise en œuvre d'une solution PaaS, on peut mentionner :

- **Compatibilité.** Les applications peuvent nécessiter des ajustements pour être compatibles avec les services et les contraintes spécifiques de la plateforme PaaS.
- **Enfermement propriétaire.** La dépendance aux outils et aux services spécifiques à une plateforme peut compliquer la migration vers un autre fournisseur.
- **Gestion des coûts.** Les coûts peuvent être difficiles à prévoir en fonction de l'utilisation des ressources.
- **Protection des données.** Bien que le fournisseur gère une partie de la sécurité, le consommateur doit s'assurer que ses applications respectent les normes de sécurité et de protection des données, et que les pratiques de sécurité du fournisseur sont alignées avec ses besoins.

Plan

IaaS — Infrastructure as a Service

L'Infrastructure as a Service (IaaS) désigne la disponibilité à la demande de ressources informatiques hautement évolutives en tant que services sur Internet.

Les entreprises ne doivent pas acheter, configurer et gérer les infrastructures elles-mêmes, et ne payent que ce qu'elles utilisent.

Quelques exemples de composants IaaS :

- Machine virtuelle (*virtual machine* ou VM),
- Disque virtuel (disques durs ou SSD),
- Réseau virtuel,
- Serveur physique dédié, etc.

Responsabilités du fournisseur et du consommateur

Le fournisseur gère :

- Le centre de données (locaux, électricité, refroidissement) et l'infrastructure réseau.
- Les serveurs physiques, le stockage, et les hyperviseurs.
- Les serveurs virtuels et leur système d'exploitation.

Le consommateur gère :

- La capacité des machines virtuelles et des disques.
- La configuration et la mise à jour des systèmes d'exploitation.
- L'installation, la configuration et la mise à jour de l'ensemble du logiciel : environnement d'exécution (runtime), logiciel intermédiaire, applications, etc.
- Le routage pour les réseaux virtuels.
- La sécurité des données et la conformité aux normes de protection des données.

Cas d'utilisation de l'IaaS

Parmi les principaux cas d'utilisation de l'IaaS, on peut mentionner :

- **Test et développement.** Facilite la création d'un environnement de test pour le développement d'une application, et accélère sa mise sur le marché.
- **Pic de trafic épisodique.** Permet d'absorber des pics de trafics épisodiques sur une application, par exemple durant la période des fêtes.
- **Reprise après sinistre (*disaster recovery*).** Facilite la continuité des opérations et la reprise des activités après une panne ou une catastrophe naturelle, par exemple.
- **Calcul à haute performance (HPC).** Permet d'effectuer des calculs complexes par exemple sur des modèles climatiques, météorologiques ou encore financiers.
- **Analyser des données massives (*big data*).** Permet d'analyser des volumes de données extrêmement grands pour en extraire des informations pertinentes.

Inconvénients du IaaS

Parmi les éléments à prendre en considération lors de la mise en œuvre d'une solution IaaS, on peut mentionner :

- **Connectivité.** Une mauvaise connectivité ou une panne d'Internet peuvent impacter le fonctionnement des processus qui dépendent de l'infrastructure IaaS.
- **Ressources partagées.** En raison de la mutualisation des ressources, la disponibilité d'une ressource peut être affectée par l'activité d'un autre consommateur.
- **Manque de transparence.** Il peut être difficile d'obtenir une vue détaillée sur les performances et la sécurité de l'infrastructure gérée par le fournisseur.
- **Sécurité.** Le consommateur n'a pas de contrôle sur la sécurité de l'infrastructure gérée par le fournisseur et doit donc s'en remettre à lui.

Plan

XaaS — Anything as a Service

Avec la multiplication et la diversification des offres de service dans le cloud, les modèles standard (SaaS, PaaS, et IaaS) manquent de précision et des services très différents finissent par se retrouver dans une même catégorie.

C'est pourquoi l'industrie du cloud parle beaucoup de XaaS (Anything as a Service).

Le XaaS englobe les modèles de services standard, mais permet de décrire des modèles de services plus spécifiques.

Quelques exemples de modèles XaaS

Parmi les nombreux modèles de service XaaS devenus courants, on peut mentionner :

- **DBaaS** (*Database as a Service*) pour les bases de données relationnelles et NoSQL
- **CaaS** (*Container as a Service*) pour les plateformes d'orchestration de conteneurs.
- **DaaS** (*Desktop as a Service*) pour des bureaux virtuels à usage général ou spécialisé (CAD, développement, etc.)
- **FaaS** (*Function as a Service*) et le **BaaS** (*Backend as a Service*) qui constitue les fondements de l'informatique sans serveur (*Serverless*), que nous aborderons plus en détail.

Plan

Informatique sans serveur (*serverless computing*)

Depuis quelques années, on parle de plus en plus d'informatique sans serveur.

Cela ne signifie pas que les serveurs disparaissent des centres de données, mais qu'ils disparaissent des préoccupations des équipements de développement qui peuvent coder, tester, et déployer leurs applications sans jamais avoir à se préoccuper de l'infrastructure.

Le but de cette présentation est d'examiner rapidement les caractéristiques, les cas d'utilisation, et les avantages et inconvénients de l'informatique sans serveur.

Qu'est-ce que serverless computing ?

Selon le glossaire de la Cloud Native Computing Foundation, l'informatique sans serveur (*serverless computing*) a les caractéristiques suivantes :

- **Abstraction des serveurs.** Le fournisseur assure la gestion opérationnelle des machines virtuelles et physiques (approvisionnement, maintenance, gestion, etc.).
- **Interfaces utilisateurs.** Les services sont fournis à travers des interfaces telles que SDK et API, CLI, et environnements d'exécution conformes au standard OCI (*open container initiative*), pour faciliter l'automatisation et le développement d'applications.
- **Paiement à l'utilisation.** Le consommateur ne paye que pour les ressources effectivement utilisées (temps de calcul, bande passante, espace de stockage, etc.).
- **Mise à l'échelle automatique.** Les ressources (calcul, stockage, réseau, etc.) sont automatiquement ajustées à la hausse ou à la baisse selon les besoins de l'application, sans intervention de l'utilisateur.

Serverless computing et PaaS

Bien qu'il y ait indéniablement des similitudes entre Serverless et PaaS, ils présentent un certain nombre de différences. Notamment :

- **Portée.** Le serverless englobe divers services avec les caractéristiques mentionnées, tandis que le PaaS se réfère spécifiquement à un modèle de service cloud.
- **Mise à l'échelle.** Le PaaS offre une élasticité rapide, mais pas toujours une mise à l'échelle automatique ni un ajustement aussi fin de la capacité des ressources.
- **Libération des ressources.** En serverless, les ressources inutilisées sont libérées automatiquement et réapprovisionnées lorsque c'est nécessaire, ce qui n'est pas typique du PaaS.
- **Tarification.** En général, la tarification du serverless est plus dynamique et a une granularité plus fine que le PaaS (p. ex., temps d'exécution mesuré en millisecondes plutôt qu'en minutes).

Cas d'utilisation

Le serverless est bien adapté pour :

- **Fonctions.** La réalisation de fonctions pour le traitement d'événements ou de flux de message avec un service FaaS (*function as a service*)
- **Automatisation de flux de travail.** L'automatisation de flux de travail (*workflow*) permet de réaliser des tâches qui intègrent différents services et applications, et dont l'exécution peut être déclenchée par des événements.
- **Backend d'API.** La réalisation de backend d'API pour des applications web ou mobiles avec un service BaaS (*backend as a service*).
- **Microservices.** Prise en charge des architectures de microservices et, en particulier, la réalisation de microservices sans état (*stateless*).

Avantages

En plus des avantages du cloud computing et de la tarification à l'utilisation, on peut encore mentionner au moins deux avantages du serverless :

- **Amélioration de la productivité.** : Les équipes de développement peuvent se concentrer sur l'écriture du code, l'innovation et l'optimisation, plutôt que sur la gestion de l'infrastructure.
- **Cycles de développement optimisés.** Facilite la mise en œuvre du DevOps en fournissant l'infrastructure nécessaire pour intégrer, tester, et déployer du code en production.

Inconvénients

Le serverless peut également avoir un certain nombre d'inconvénients dont il faut tenir compte :

- **Enfermement propriétaire.** Les offres serverless des différents fournisseurs sont similaires, mais généralement incompatibles entre elles.
- **Démarrage.** Si une ressource n'a pas été utilisée depuis un moment, le démarrage peut être lent (jusqu'à plusieurs minutes) et affecter les performances et la réactivité d'une application sans serveur.
- **Processus de longue durée.** Le serverless computing n'est pas conçu pour exécuter du code pendant de longues périodes. Pour des processus de longue durée, une solution serverless peut être plus onéreuse qu'une solution IaaS ou PaaS.
- **Manque de visibilité.** En optant pour le serverless computing, une entreprise perd tout contrôle sur le matériel et les environnements d'exécution. Le manque de visibilité sur les processus backend peut également compliquer les tests et le débogage.