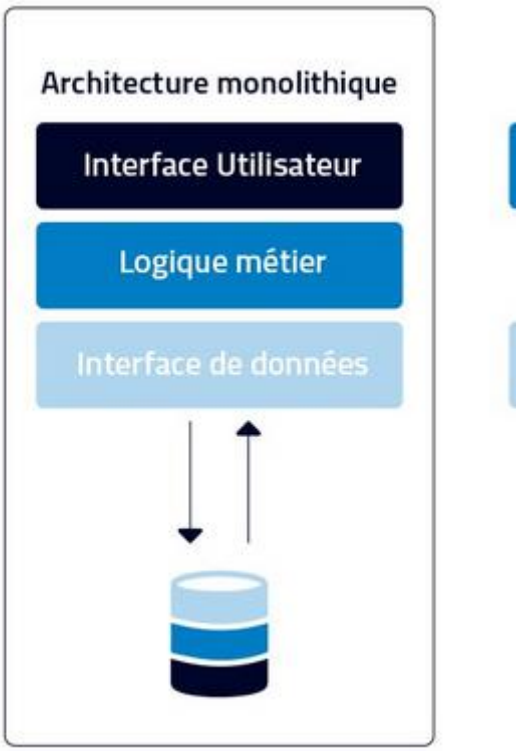
Introduction aux styles architecturaux des applications distribuées

I - Architectures monolithiques

1-1 Définition d'une architecture monolithique



Dans une architecture logicielle monolithique, les couches logicielles et leurs composants logiciels sont développés dans un seul programme et sur une seule plateforme matérielle (exemple : applications web classiques). Ainsi tous les composants sont liés et leurs fonctions sont gérées dans un environnement unique. Cette architecture est idéale pour les petites équipes de développement , c'est pourquoi beaucoup de startups l'utilisent.

Cette architecture est l'architecture traditionnelle de constructions des applications que certains développeurs trouvent dépassée. Cependant dans certains cas, cette architecture est la solution idéale.

1 –2 Avantages

- **Développement simple**
- **Déploiement simple**
- **Audit, sécurité, optimisation et journalisation (logs) simples puisque le code est unique et à un seul endroit.**
- **Si l'application monolithique est bien construite , les temps de réponse sont meilleurs que ceux des microservices car le code et la mémoire nécessaire aux composants sont partagés.**
- **Solution idéale si l'application doit être construite, testée et déployée rapidement**

1 – 3 Inconvénients

- **Dépendance technologique (langage de programmation et frameworks)**
- **Code très important difficile à maintenir et à comprendre**
- **Difficile d'adopter de nouveaux choix technologiques**
- **Agilité limitée : la moindre modification entraîne un redéploiement de toute l'application donc freine les équipes de développement**
- **Passage à l'échelle (scalability) difficile**

II – SOA

2-1 Définition

Service - Oriented

SOA est un style architectural dans lequel une application est constituée d'agents logiciels appelés des services. Chaque fonction réalise un ensemble traitements bien précis dans le système d'information de l'entreprise. 2 types de services : service producteur et service consommateur. Un agent logiciel peut-être les 2 à la fois. Une application construite selon SOA est constituée de services qui communiquent de manière transparente et qui peuvent être réutilisés. L'ESB (Bus Logiciel d'entreprise) sert d'interface de communication entre les services. Les services peuvent se connecter au bus via n'importe quel protocole (http, protocoles de messagerie, etc…). L'ESB assure les fonctions de transport, routage et conversion de protocoles des messages échangés entre services. Les services peuvent être développés dans différents langages de programmation. SOA est la base de l'interopérabilité entre plateformes logicielles différentes et il est à l'origine du concept B2B(Business To Business). Les Web Services de type SOAP basés sur XML et http ainsi que les services REST basés uniquement sur le protocole http sont des

implémentations possibles de la norme SOA. SOA est utilisé dans des systèmes complexes comme les systèmes bancaires qui sont difficilement décomposables en microservices.

2-2 Avantages

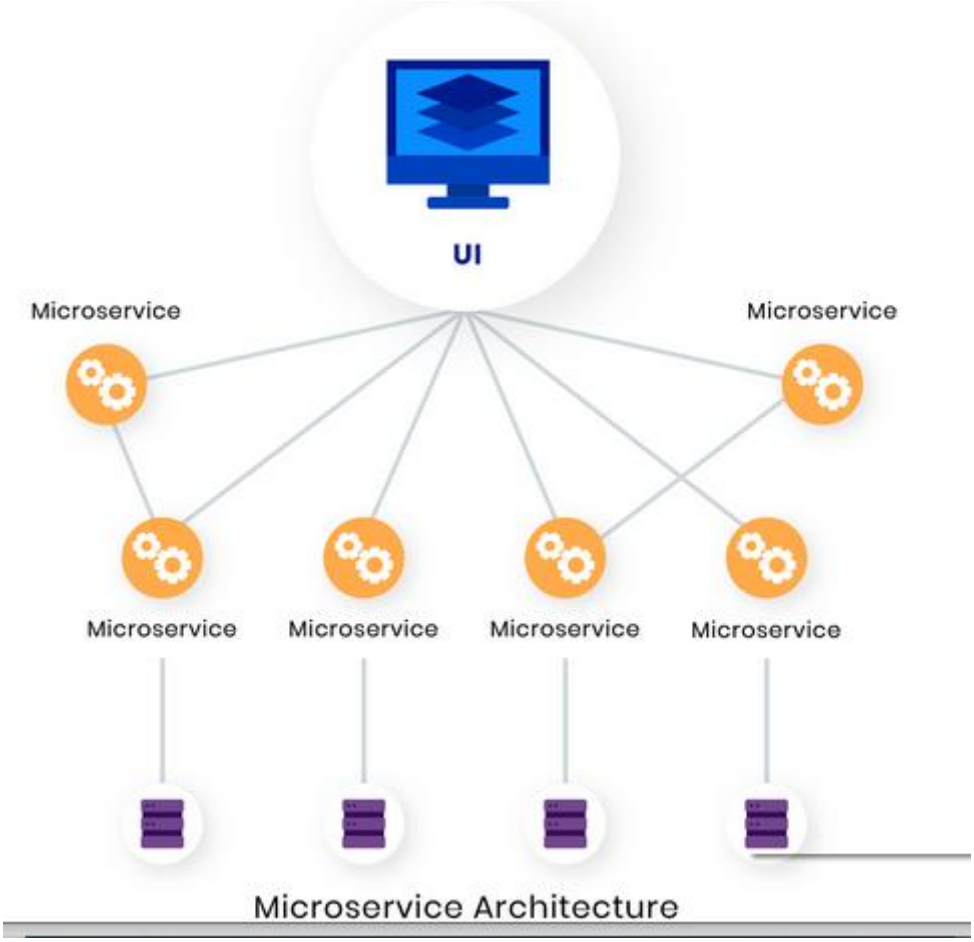
- **Réutilisabilité des services**
- **Haute fiabilité**
- **Meilleure Agilité (développement Agile)**
- **Développement parallèle des services**

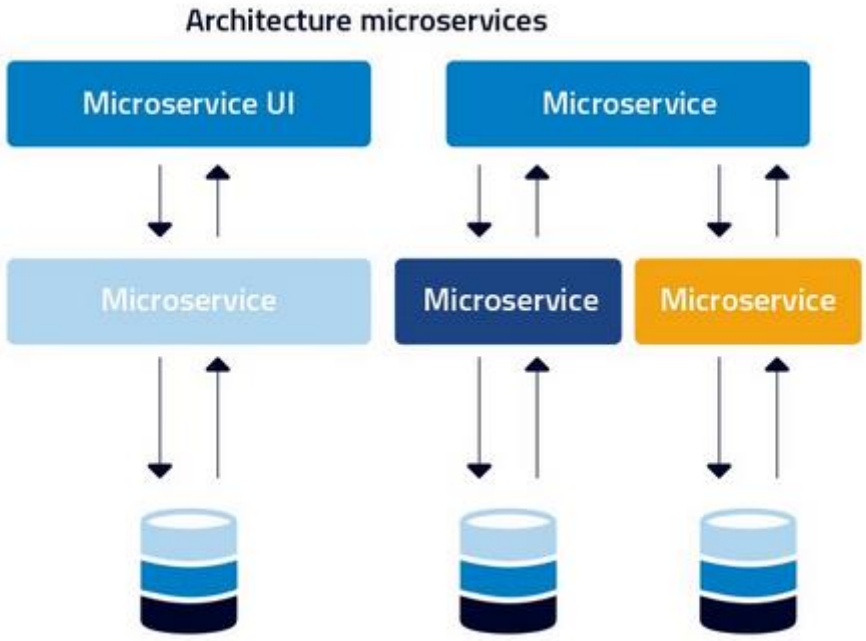
2-3 Inconvénients

- **L'ESB constitue éventuellement un goulot d'étranglement en cas de panne**
- **Management du système complexe**
- **Temps de réponse pas toujours optimal**
- **Coût d'investissement élevé**

III - Microservices

3-1 Définition





Les microservices sont apparus pour simplifier la complexité des applications SOA .

Un microservice est un type d'architecture orientée service qui se focalise sur un ensemble de composants autonomes qui constituent une application. Ces composants autonomes sont rattachés à des API REST. Les microservices se focalisent sur les fonctions métier contrairement aux applications monolithiques qui se focalisent sur les couches logicielles.

Les microservices sont adaptés à leur mise en œuvre sur le cloud. Ils sont donc cloud-native. Beaucoup de compagnies les utilisent comme Netflix,Amazon,ebay ,PayPal ,….

Les microservices peuvent être développés dans des langages différents.

3-2 Avantages

- **Productivité** : Les petites équipes spécialisés par microservice travaillent en parallèle et cette spécialisatio leur permet alors une plus grande productivité que les équipes plus nombreuses. Elles permettent d'accélére les prises de décision et les implémentations au sein du composant.
- **Isolement** : Si l'un des composants tombe en panne, cela n'impacte pas l'ensemble de l'application. Ce avantage donne aux développeurs la liberté de développer et de déployer des services en fonction des besoin sans forte interdépendance entre les services.
- **Évolutivité et modularité :** La bonne séparation du rôle de chaque microservice permet aux développeurs d gérer facilement l'évolution des microservices.
- **Scalabilité**
- **Facilité de déploiement :** Les microservices individuels et isolés sont beaucoup plus faciles et rapides déployer dans les pipelines d'intégration et déploiement continu. Seul le service concerné est modifié e redéployé lorsqu'une mise à jour est nécessaire.

- **Agilité meilleure**
- **Equipes de développements pluridisciplinaires**
- **Facile à développer, tester et déployer**

3-3 Inconvénients

- **Complexité de la communication entre microservices**
- **Gestion des transactions complexes car chaque microservice possède sa propre base de données.**
- **Problèmes de sécurité**

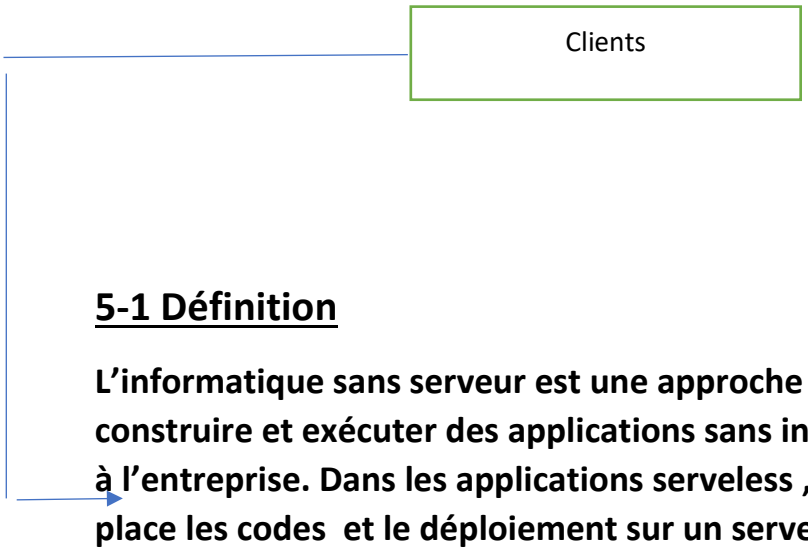
IV- Différences entre SOA et Microservices

SOA est une architecture qui permet aux divers systèmes d'information de l'entreprise, également appelés services, de communiquer entre eux, souvent par l'intermédiaire d'un bus de message. Ils sont implémentés sur différentes plateformes et utilisent multiples langages en mettant en œuvre un système dit de « couplage faible ».

L'architecture orientée services est similaire aux microservices en termes de modularité et de finalité : chaque service a effectivement un rôle et une responsabilité à gérer qui lui sont propres. Ils partagent plus ou moins les mêmes avantages. Par contre, ils diffèrent sur plusieurs critères :

- La première différence est la taille des services. La taille des services dans **SOA** est relativement grande vis-à-vis de celle des microservices.
- Dans une architecture orientée services, les services peuvent partager une source de données, contrairement aux microservices où chaque service est totalement indépendant.
- Des points précédents découlent une communication entre les services **SOA** plus simple que pour les microservices. Cela a pour conséquence une modélisation et une conception plus difficile du côté des microservices.
- Les microservices sont souvent containérisés avec des outils comme Docker qui permettent d'optimiser le temps de démarrage et le temps de déploiement, et de mieux gérer les ressources.
- Dans SOA, on risque de perdre l'agilité si la taille des services est importante, c'est-à-dire dans le cas d'équipes de plus de 10 personnes.

V – L'informatique sans serveur d'entreprise(Serveless Architecture)



5-1 Définition

L'informatique sans serveur est une approche cloud computing (nuage) pour construire et exécuter des applications sans infrastructure propre de serveur à l'entreprise. Dans les applications serveless , les développeurs mettre en place les codes et le déploiement sur un serveur sans se préoccuper de la maintenance des serveurs , ces derniers étant fournis par un prestataire de Cloud comme AWS par exemple.

Une architecture serveless évite à l'entreprise de maintenir des serveurs , de rendre à l'échelle les applications ainsi que de gérer et administrer ses propres bases de données. Toutes ces ressources sont ainsi délocalisées.

Les 3 modèles de cloud

Il existe trois modèles de cloud computing :

- **IaaS : L'infrastructure as a service est le premier modèle de cloud,où :**

– l'entreprise maintient les applications, les runtimes, l'intégration SOA (architecture orientée services : Service Oriented Architecture), les bases de données, le logiciel serveur

– le fournisseur Cloud maintient : la virtualisation, le matériel serveur, le stockage, les réseaux.

- **PaaS : La Platform as a service est le second modèle de cloud, où :**

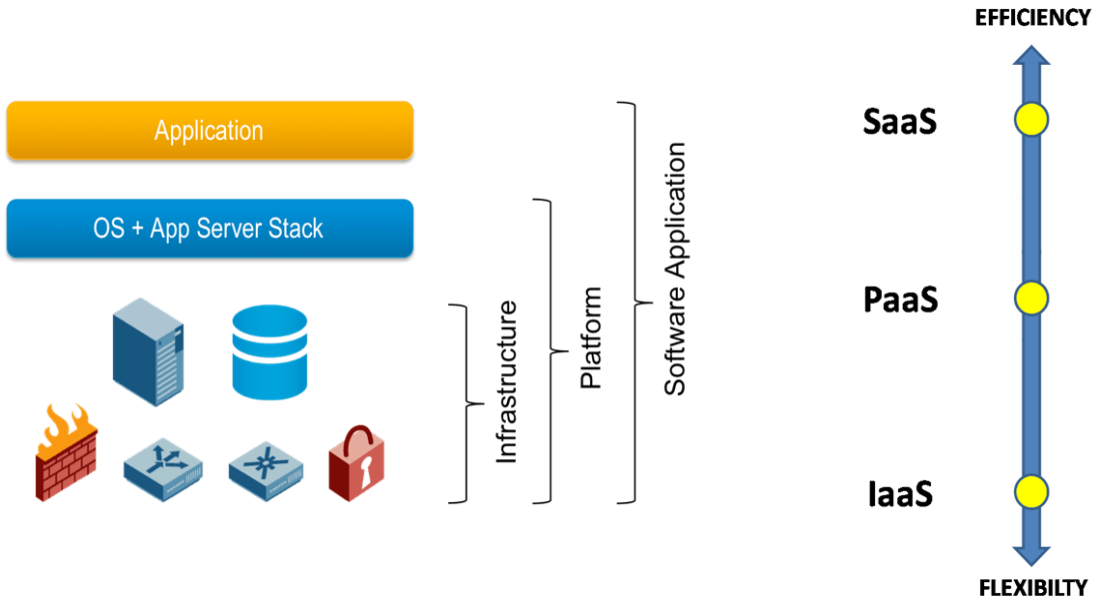
– l'entreprise maintient uniquement les applications ;

– le fournisseur Cloud maintient : les runtimes, l'intégration SOA, les bases de données, le logiciel serveur, la virtualisation, le matériel serveur, le stockage, les réseaux.

- **SaaS : Le Software as a service est l'ultime modèle de cloud, où :**

le fournisseur Cloud maintient les applications, les runtimes, l'intégration SOA, les bases de données, le logiciel serveur, la virtualisation, le matériel serveur, le stockage, les réseaux.

Le SaaS, souvent associé au « cloud computing » peut être vu comme un modèle économique de consommation des applications : celles-ci sont consommées et payées à la demande (par utilisateur et par minute d'utilisation par exemple) et non plus acquises par l'achat de licences. Le SaaS peut donc à ce titre reposer sur une infrastructure informatique dans le nuage.



5-2 Avantages

- **Facile à déployer**
- **Réduction des coûts**

- **Excellent passage à l'échelle**

5-3 Inconvénients

- **Dépendance vis-à-vis du vendeur de Cloud**
- **Non recommandé pour des tâches à long terme**