# ATELIER 3

## A diagram of a server Description automatically generatedArchitecture

Figure 1 - Diagramme d'architecture MicroService

## Microservice Vs SOA

**Architecture Microservices**

L’architecture microservices se caractérise par une approche modulaire où chaque service est indépendant et responsable d’une fonction spécifique. Chaque microservice peut être développé, déployé et mis à jour de manière autonome, ce qui permet une grande flexibilité et agilité dans le développement logiciel. Les microservices communiquent entre eux via des API légères, souvent HTTP/REST, ce qui simplifie les intégrations. Cette indépendance facilite également le déploiement continu et la scalabilité, car chaque service peut être mis à l’échelle indépendamment des autres.

**Architecture SOA (Service-Oriented Architecture)**

L’architecture orientée services (SOA) est une approche de conception où les services sont exposés via des interfaces bien définies, généralement en utilisant des protocoles de communication tels que REST. SOA se concentre sur l’intégration d’applications disparates, souvent héritées, et sur la réutilisation des services existants. Les services dans SOA sont plus gros et plus complets que les microservices, englobant souvent plusieurs fonctionnalités liées. SOA favorise l’interopérabilité et l’intégration entre différents systèmes d’entreprise, souvent à travers un bus de service (ESB) qui facilite la communication et la gestion des services.

**Avantages et Différences**

| **Critère** | **Microservices** | **SOA** |
| --- | --- | --- |
| **Granularité** | Services très fins et spécialisés. | Services plus larges, souvent couvrant plusieurs fonctionnalités liées. |
| **Indépendance des services** | Services totalement indépendants les uns des autres. | Services interconnectés mais souvent dépendants via un bus de service (ESB). |
| **Développement** | Permet le développement autonome de chaque service avec des équipes dédiées. | Favorise la réutilisation de services existants pour une intégration facile. |
| **Scalabilité** | Échelle fine, chaque service peut être mis à l’échelle indépendamment. | Échelle plus grossière, nécessite souvent de mettre à l’échelle des composants plus larges. |
| **Déploiement** | Déploiement continu facilité grâce à l’autonomie des services. | Déploiement plus complexe en raison des interconnexions des services. |
| **Interopérabilité** | API légères (souvent HTTP/REST), facilitant l’intégration avec d’autres services. | Utilise des protocoles standardisés pour assurer l’interopérabilité entre systèmes divers. |
| **Maintenance** | Facilité de mise à jour et de maintenance de chaque service individuellement. | Maintenance plus complexe due à l’interdépendance des services. |

En résumé, l’architecture microservices offre une grande flexibilité et scalabilité, particulièrement adaptée aux environnements de développement agiles et aux systèmes nécessitant des déploiements fréquents et indépendants. L’architecture SOA, en revanche, est plus adaptée aux grandes entreprises cherchant à intégrer divers systèmes hérités et à maximiser la réutilisation des services existants. Les choix entre ces architectures dépendent des besoins spécifiques de l’organisation en termes de scalabilité, flexibilité, interopérabilité et complexité de gestion.