



# microservice et docker



**beggar chaimaa**  
**bahri ilhem**



introduction

difinition microservice

les piliers fondamentaux

comparaison

Définition de Docker

**les Composants Structurels de Docker**

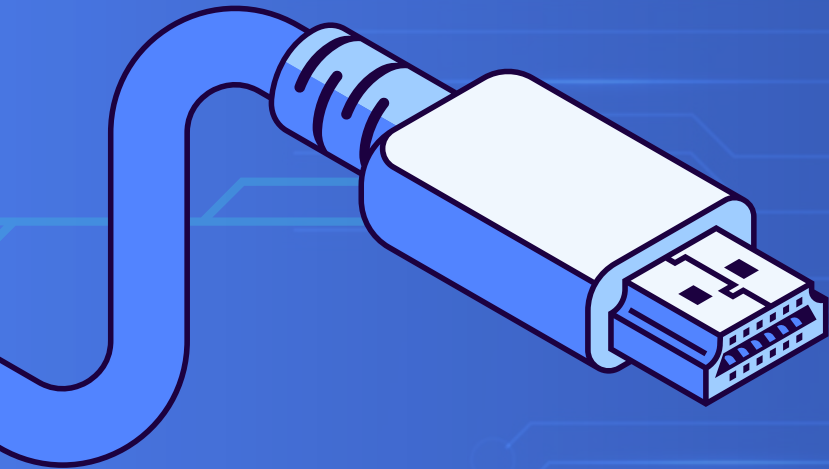
Docker vs Machines Virtuelles

La Relation entre Docker et les Microservices





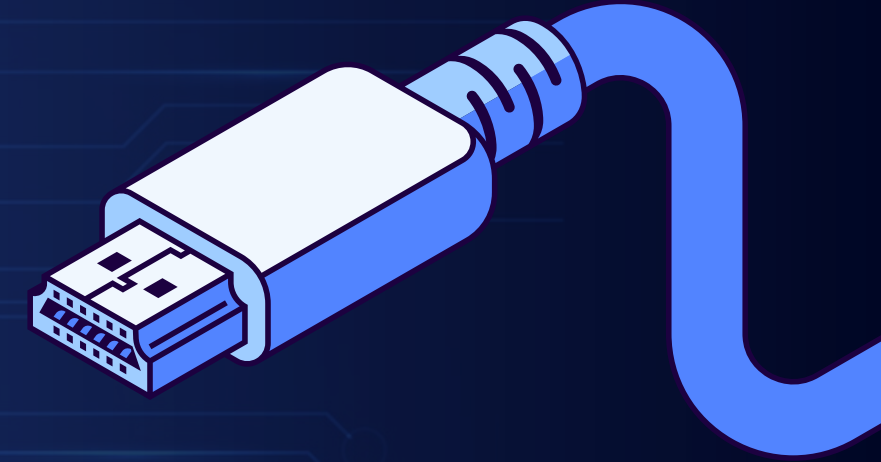
# INTRODUCTION



Le passage de l'architecture monolithique vers l'architecture microservices représente une transformation majeure dans l'ingénierie logicielle moderne. Ce changement ne concerne pas la manière de développer, mais répond aussi aux besoins actuels du marché tels que la rapidité, la flexibilité et la scalabilité.

Dans ce contexte, les technologies de conteneurisation, notamment Docker, jouent un rôle essentiel en offrant un environnement d'exécution isolé et portable, garantissant la cohérence des applications.

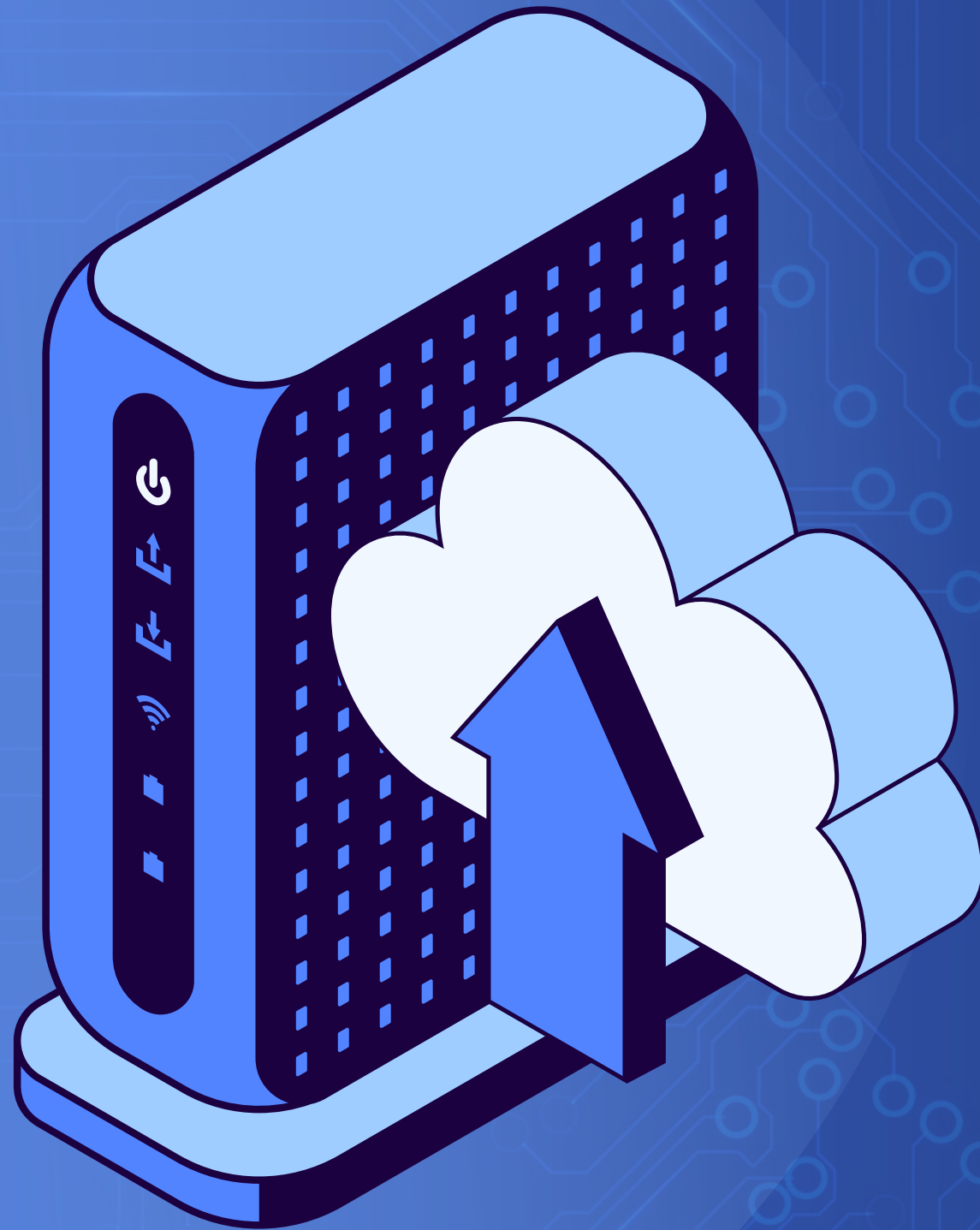
Ce travail présente les principes fondamentaux de ces technologies ainsi que leurs aspects pratiques et organisationnels.







# DÉFINITION DE L'ARCHITECTURE MICROSERVICES



L'architecture microservices est une approche de conception logicielle qui consiste à décomposer une application unique en un ensemble de services petits, autonomes et indépendants. Contrairement à l'architecture monolithique traditionnelle, chaque service est conçu pour remplir une fonction métier spécifique.



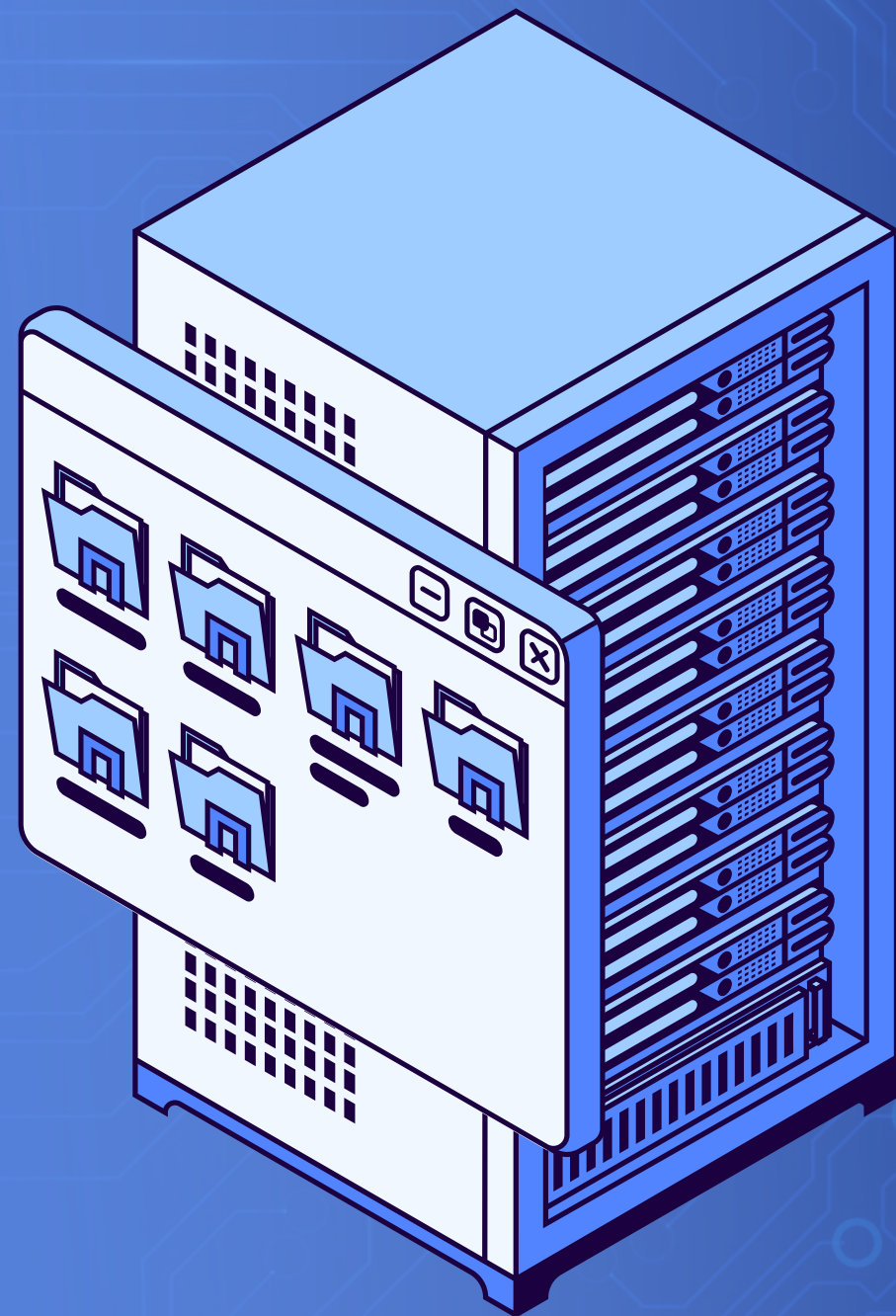
microservice







# LES PILIERS FONDAMENTAUX



Indépendance et Autonomie

01.

Communication via APIs

02

Database per Service

03.

Scalabilité Granulaire

04.

Polyglottisme Technologique

05

Résilience et Isolation des pannes

06



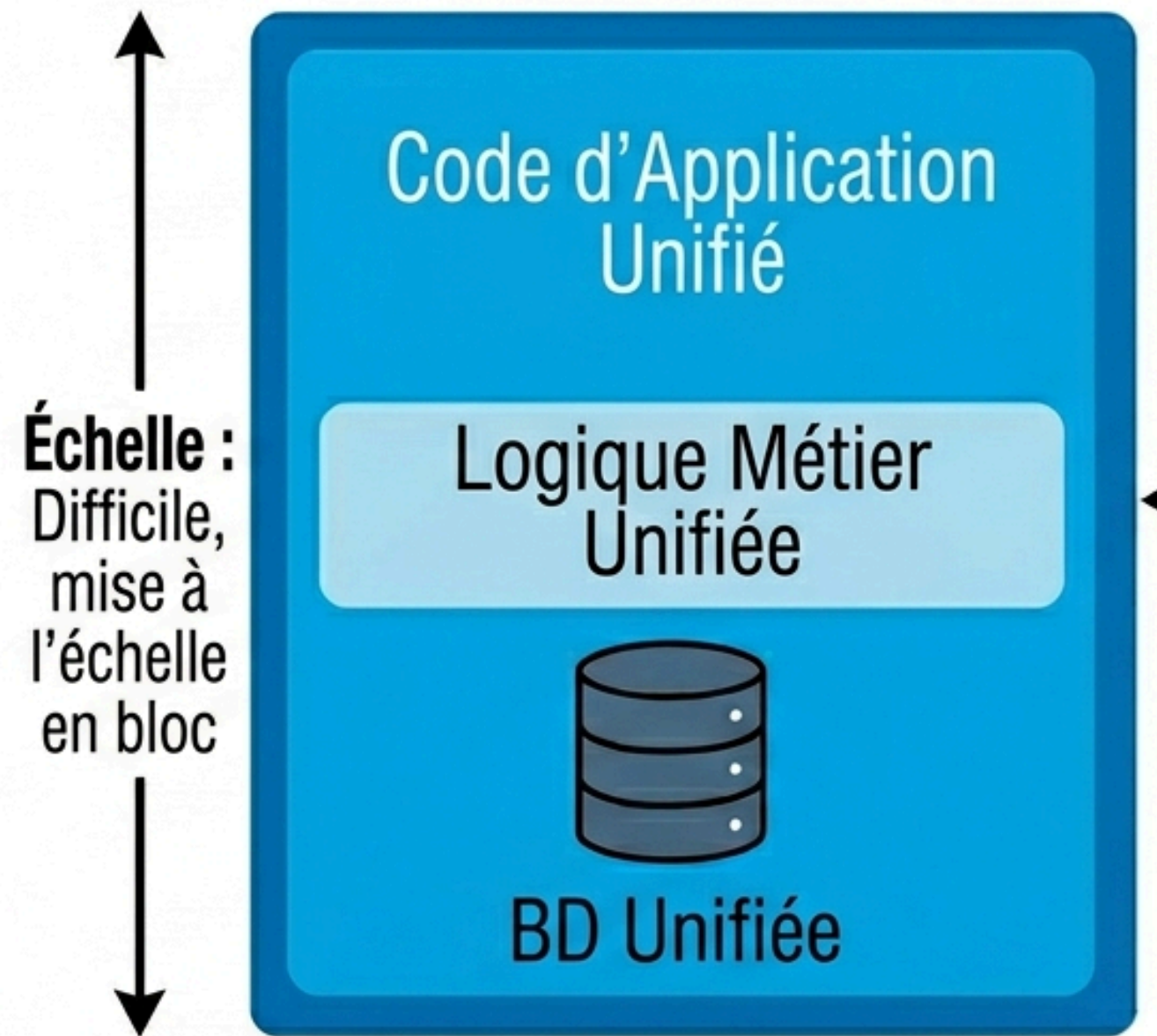
microservice





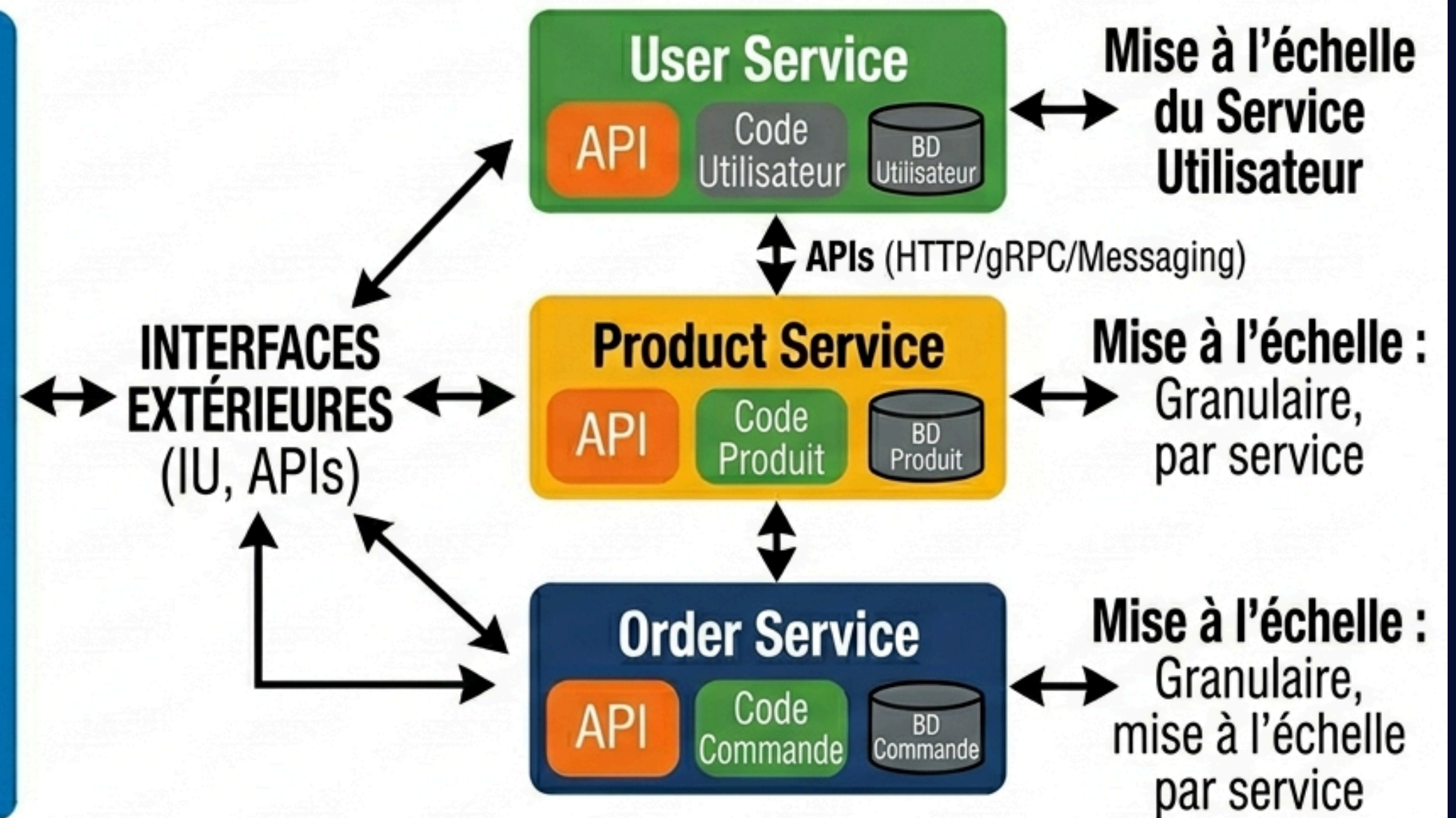
# COMPARAISON DE L'ARCHITECTURE : MONOLITHIQUE vs. MICROSERVICES

## ARCHITECTURE MONOLITHIQUE



Base de code unique, base de données partagée, couplage fort.

## ARCHITECTURE MICROSERVICES



Développement indépendant, bases de données indépendantes, couplage lâche.



# 🌀 Définition de Docker



Docker est une plateforme open-source qui automatise le déploiement d'applications dans des conteneurs logiciels. Un conteneur Docker encapsule une application avec toutes ses dépendances (bibliothèques, fichiers de configuration) pour garantir qu'elle fonctionne de manière identique dans n'importe quel environnement.



# les Composants Structurels de Docker



Docker repose sur une architecture Client-Serveur où les tâches sont réparties entre les éléments suivants .....

01. Le Démon Docker

02. Le Client Docker

03. Les Images Docker

04. Les Conteneurs Docker

05. Le Registre Docker

06. Les Volumes et Réseaux

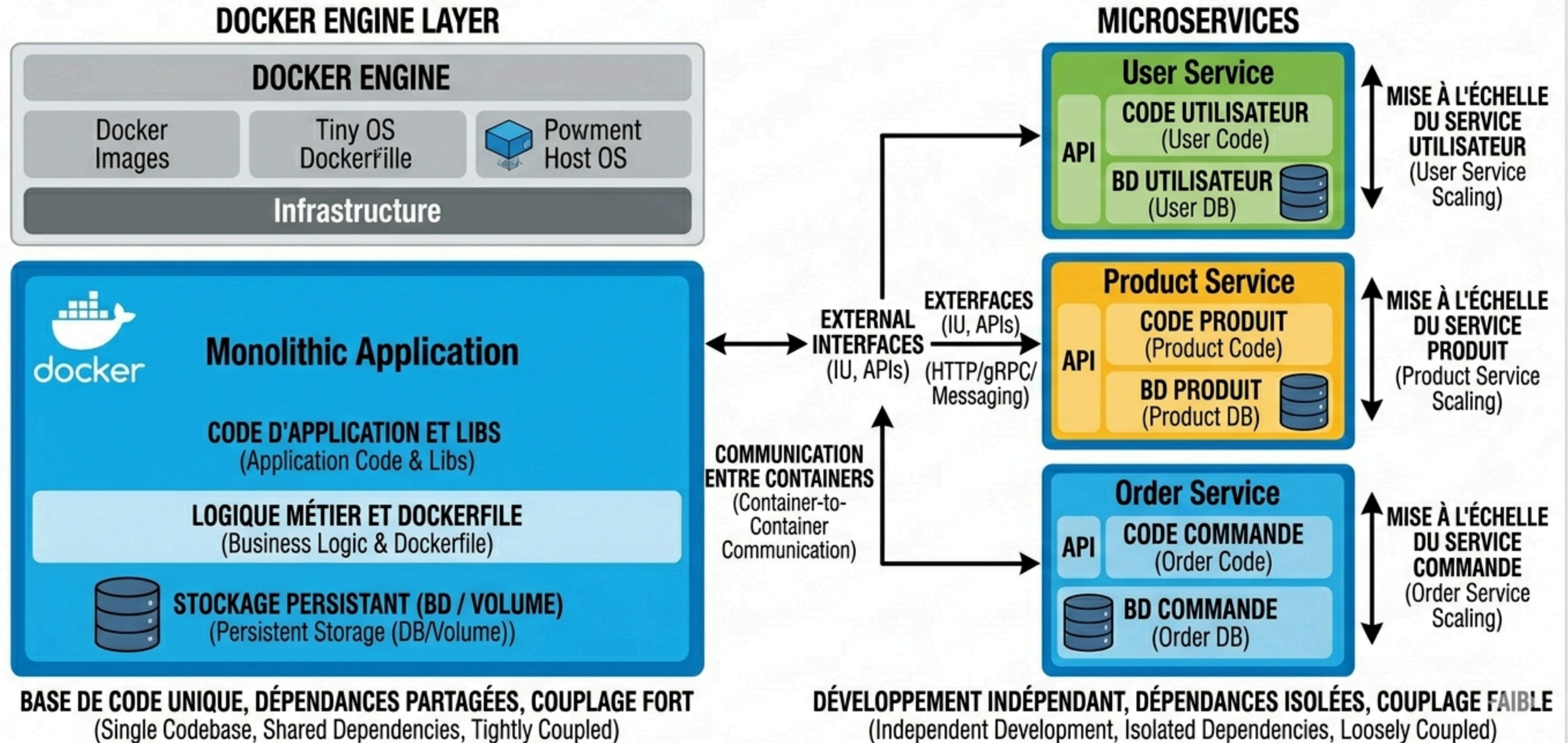


DOCKER





# ARCHITECTURES DES COMPOSANTS STRUCTURELS DOCKER (DOCKER ARCHITECTURE COMPONENT STRUCTURES)





# Docker vs Machines Virtuelles

## Docker (Conteneurs)

Utilise la virtualisation au niveau du système d'exploitation. Les conteneurs partagent le noyau (Kernel) de l'hôte, ce qui les rend extrêmement légers.

## Machines Virtuelles

- Utilisent la virtualisation au niveau du matériel via un Hyperviseur. Chaque VM inclut un système d'exploitation invité (Guest OS) complet, ce qui consomme beaucoup de ressources.





# La Relation entre Docker et les Microservices

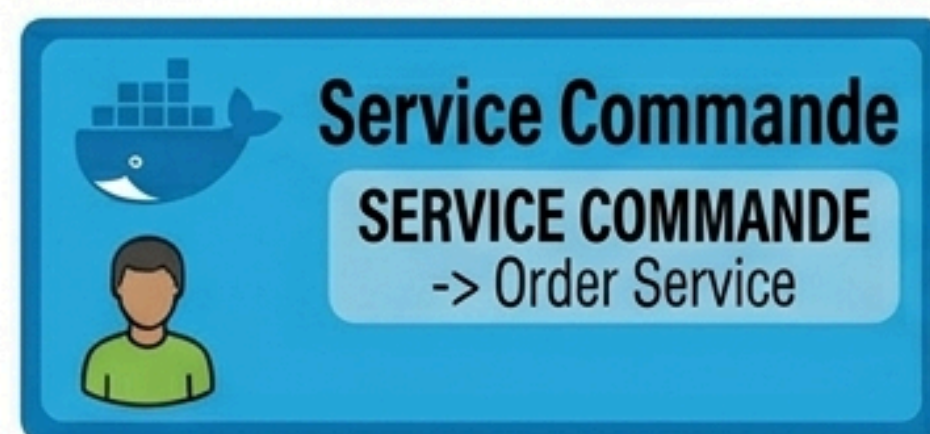
L'architecture microservices divise l'application en unités autonomes, et Docker fournit les conteneurs idéaux pour les encapsuler avec leurs dépendances.

Ce duo garantit une cohérence totale entre les environnements et un démarrage ultra-rapide par rapport aux machines virtuelles. Docker permet également de scaler chaque service indépendamment et d'isoler les pannes pour assurer la résilience globale du système. En résumé : les microservices représentent la "stratégie" et Docker est "l'outil" de mise en œuvre.



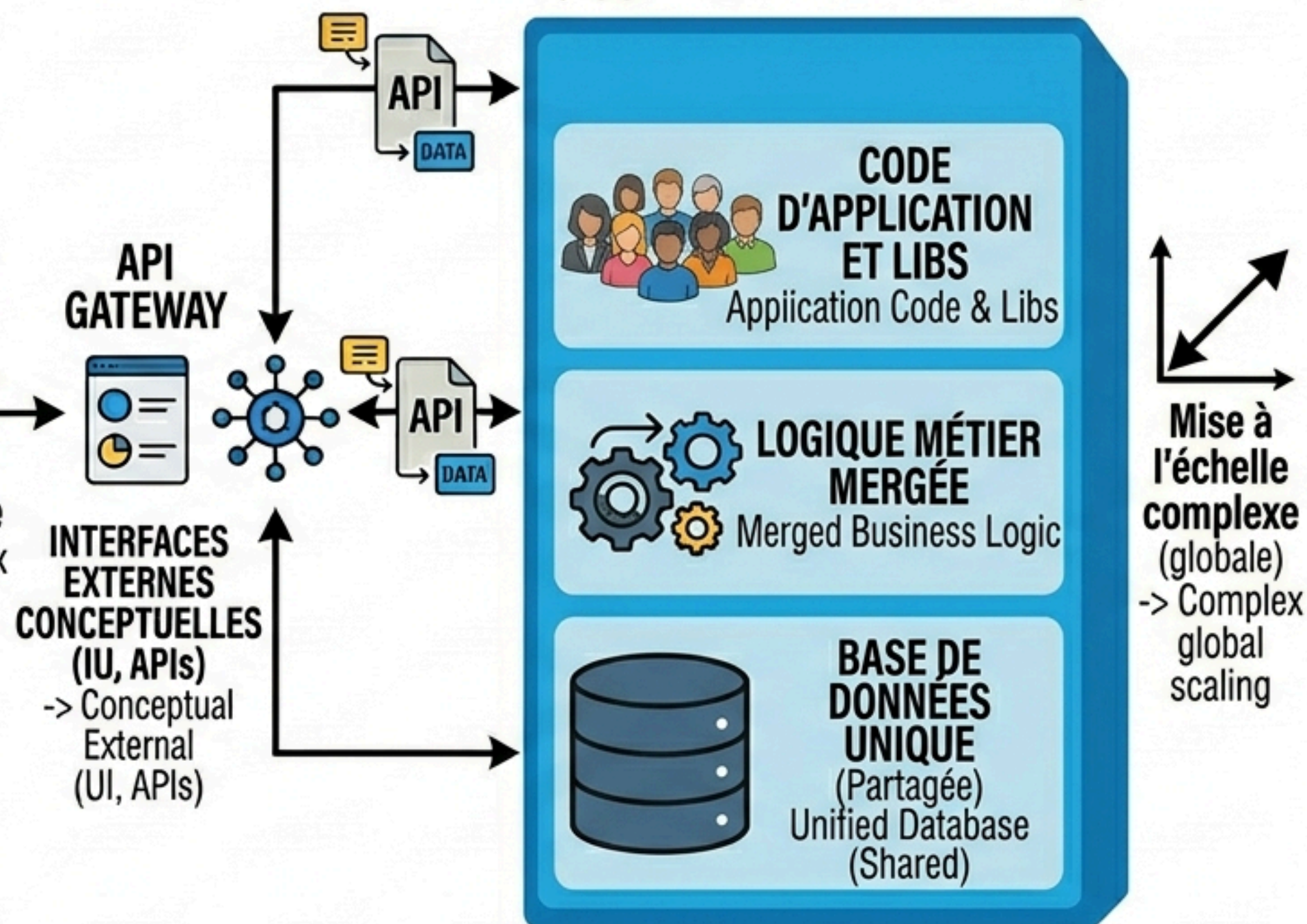
# COMPARATIVE DE L'ARCHITECTURE DES MICROSERVICES : AVEC vs SANS DOCKER

## MICROSERVICES AVEC DOCKER (Containerization)



**Mise à l'échelle granulaire, indépendante**  
-> Granular, independent scaling

## MICROSERVICES SANS DOCKER (Application Monolithique)



**Scale as one, dependancies shared, hard to decouple**  
-> Mise à l'échelle en bloc, dépendances partagées, difficile à découpler



The background is a deep blue gradient. It features faint, light blue circuit board patterns (traces) that crisscross the entire image. Overlaid on these are several stylized, layered wave or ribbon-like shapes in various shades of blue, some appearing to flow from the left and right towards the center. There are also small, dark blue dots and thin lines forming a network-like structure on the far left and right sides.

MERCI