

Introduction au génie logiciel pour le Cloud



Année académique :2021-2022

Présentation du formateur



M. OUEDRAOGO W. A. Marc Christian

Ingénieur des travaux en réseaux et maintenance
informatique

Ingénieur de conception en analyse et programmation

Doctorant en Intelligence Artificielle

PLAN GENERAL DU COURS

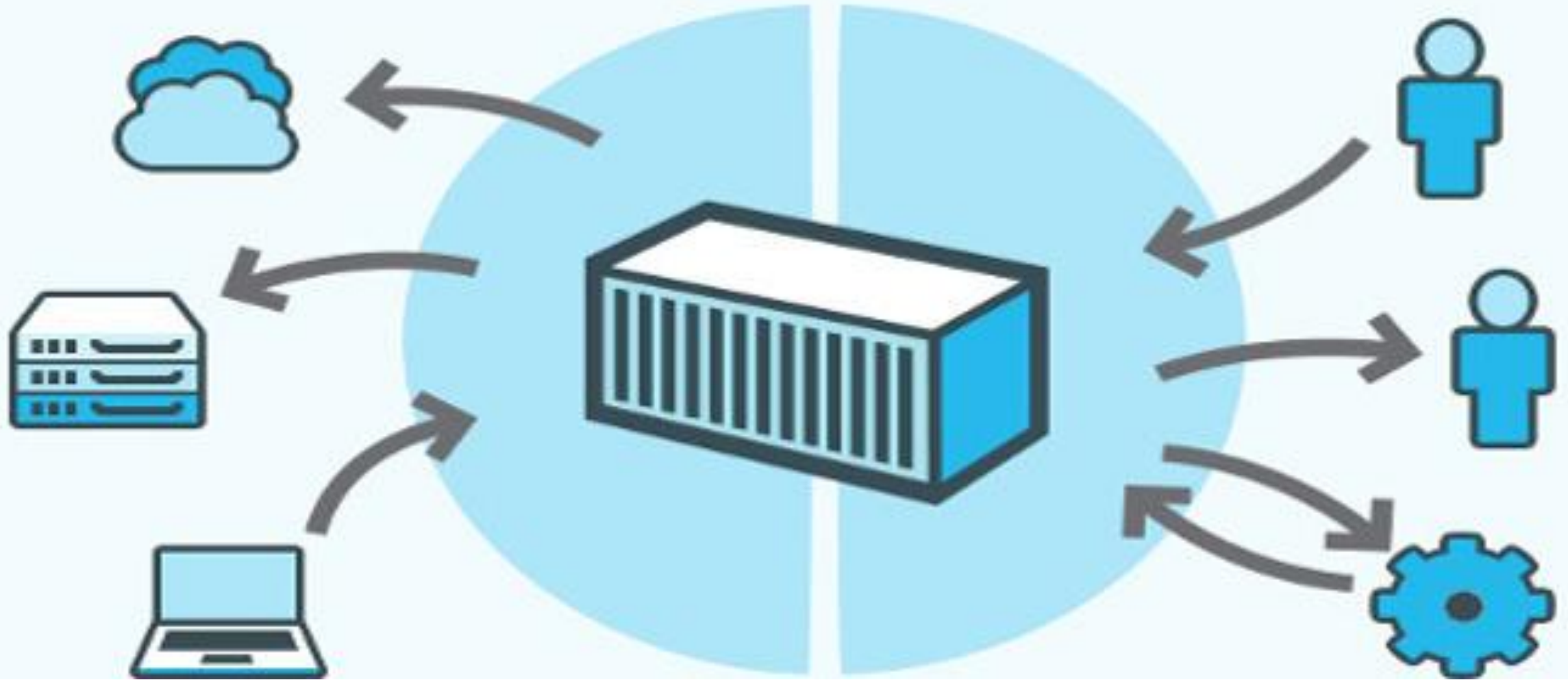


Séquence 1 : Les notions fondamentales de la virtualisation

Séquence 2 : Les notions fondamentales du cloud computing

Séquence 3 : La containerisation

Séquence 4 : La conception d'applications cloud



Séquence 3 : La conteneurisation

OG 3 : Comprendre la conteneurisation

- Décrire la conteneurisation
- Distinguer les conteneurs des machines virtuelles
- Distinguer les conteneurs des micro services
- Expliquer Orchestration des conteneurs

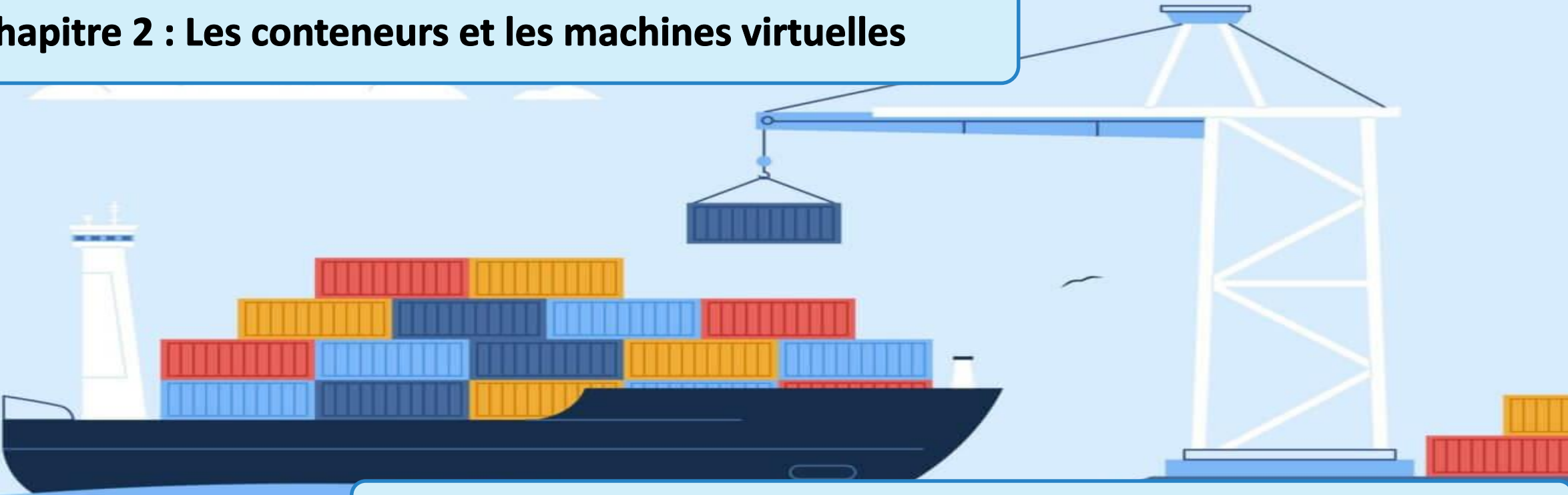
PLAN

Chapitre I : La conteneurisation

Chapitre 2 : Les conteneurs et les machines virtuelles

Chapitre 3 : Les conteneurs et les micro services

Chapitre 4 : Orchestration des conteneurs



PREAMBULES

Les conteneurs et les microservices sont des concepts connexes qui aident les entreprises à concevoir et à adopter des applications avec rapidité, efficacité et flexibilité.

Il est difficile de parler de l'un sans parler de l'autre. Les microservices et les conteneurs sont des technologies distinctes.

Elles ne doivent pas nécessairement être utilisées ensemble, mais elles le sont souvent.

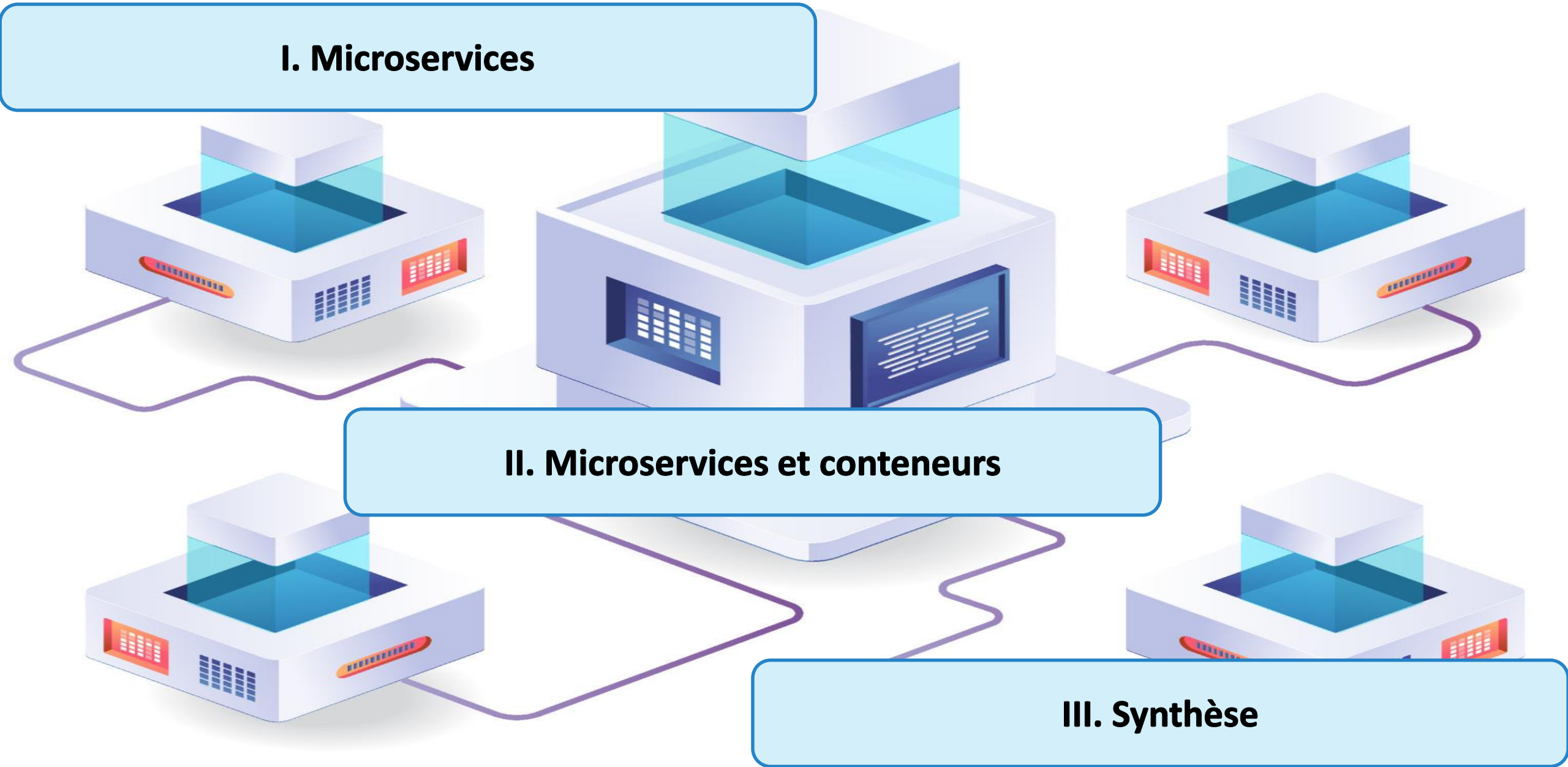
Voyons comment ils fonctionnent, quelles sont leurs principales similitudes et différences, et comment une entreprise peut les utiliser en tandem.

Chapitre 3 : Les conteneurs et les micro services

I. Microservices

II. Microservices et conteneurs

III. Synthèse



I. Microservices

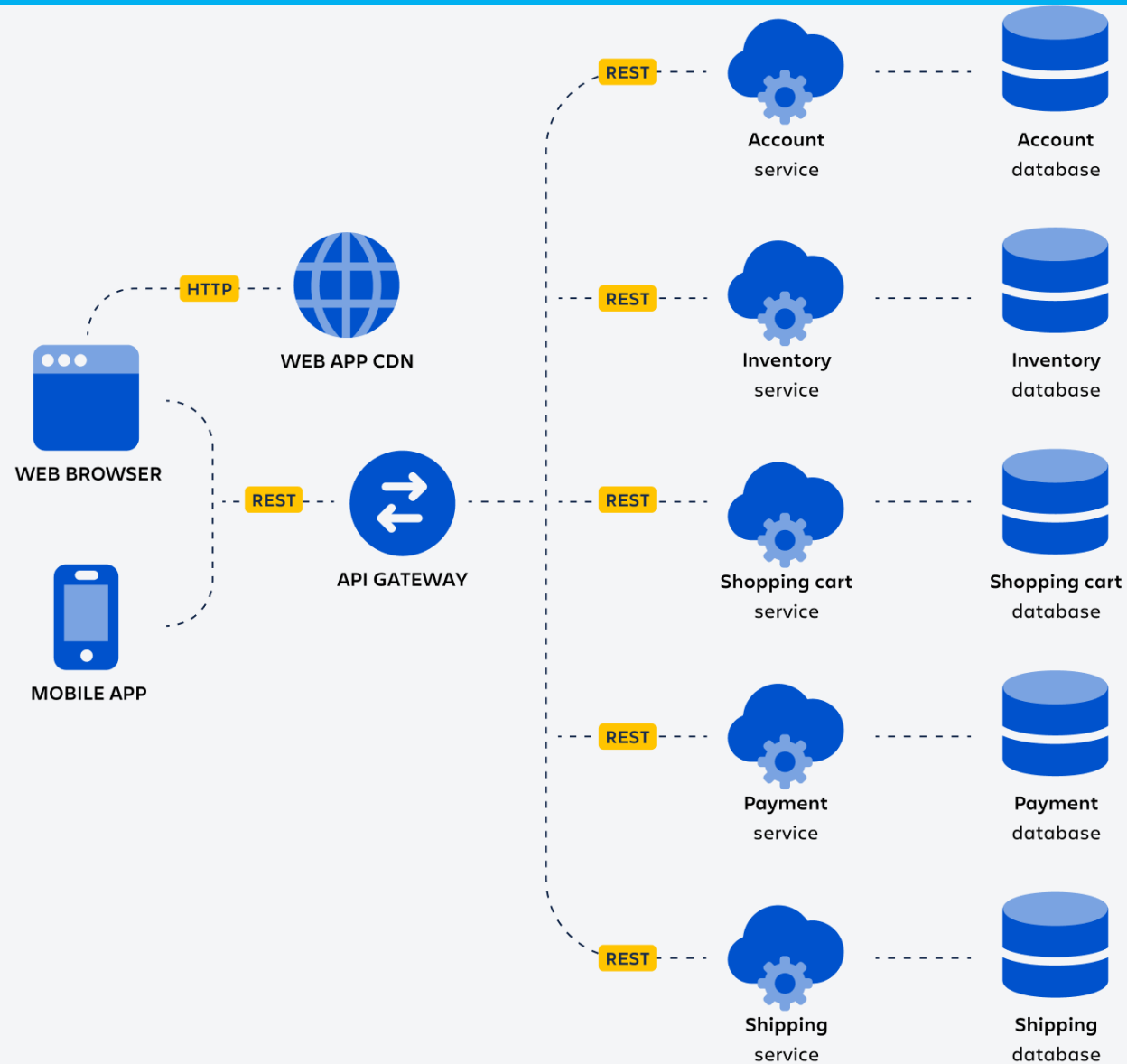
Les microservices sont des unités individuelles de logiciel qui se combinent pour fournir toutes les fonctions requises pour exécuter une application.

En général, chaque microservice gère un type discret de fonctionnalité au sein d'une application.

Par exemple,

- ✓ **un microservice gère les connexions,**
- ✓ **un autre génère l'interface utilisateur,**
- ✓ **un autre remplit l'interface avec un contenu spécifique à chaque session utilisateur**
- ✓ **et un autre encore s'interface avec la base de données qui stocke les données utilisateur.**

II. Microservices et conteneurs



I. Microservices

Il y a plusieurs raisons pour lesquelles les microservices sont devenus populaires au cours de la dernière décennie :

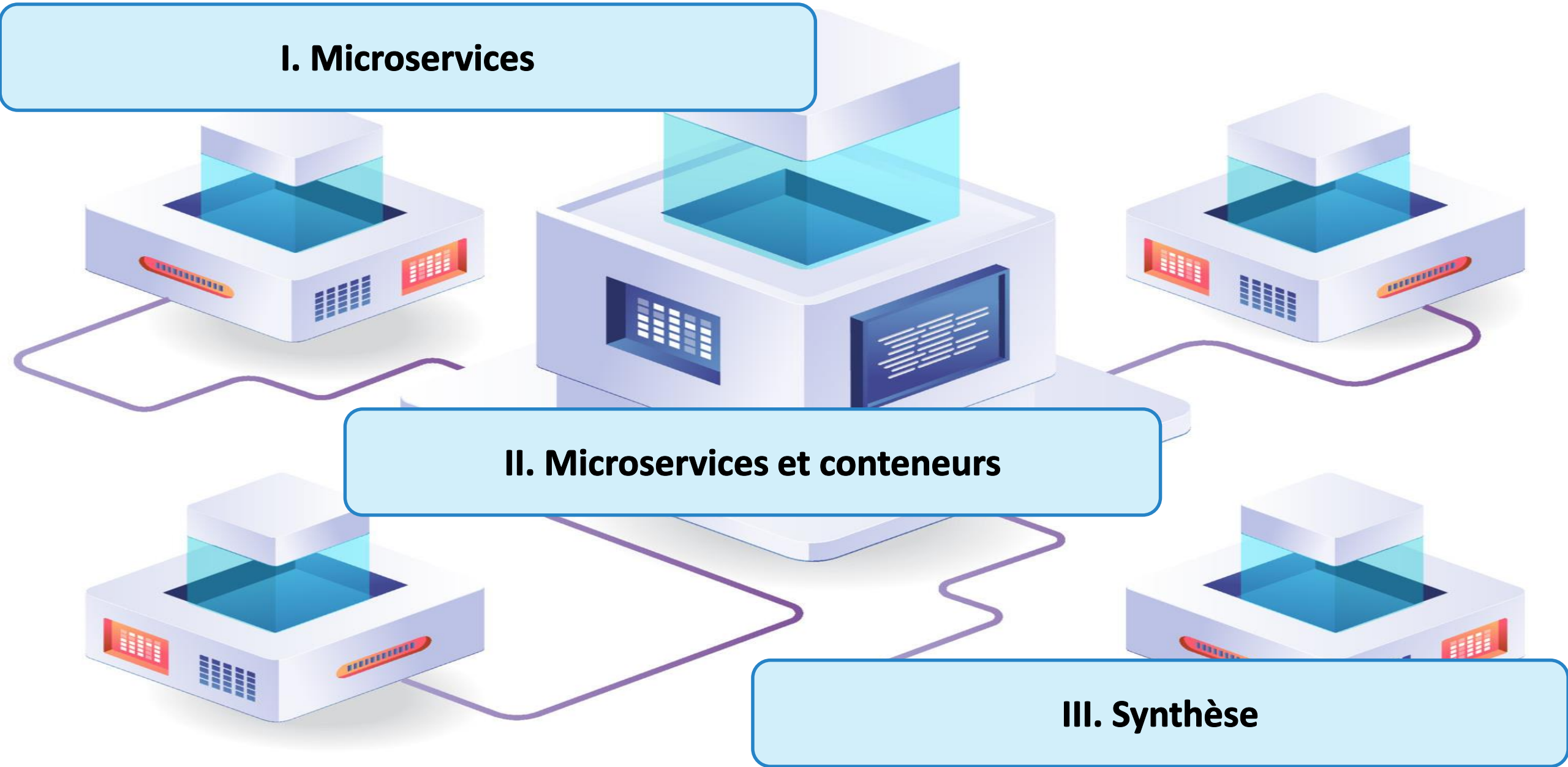
- ❑ **L'agilité.** En divisant les applications en petites parties, il est plus facile de faire évoluer ces différentes parties vers le haut ou vers le bas. Par exemple, si les demandes de connexion augmentent, l'application peut exécuter plus d'instances du microservice spécifique qui gère les connexions.
- ❑ **Résilience.** Si un microservice tombe en panne, le reste de l'application reste intact, bien qu'une partie de sa fonction puisse être indisponible. À cet égard, les architectures de microservices sont plus résilientes.

Chapitre 3 : Les conteneurs et les micro services

I. Microservices

II. Microservices et conteneurs

III. Synthèse



II. Microservices et conteneurs

La principale différence entre les microservices et les conteneurs est que les microservices sont un paradigme architectural, tandis que les conteneurs sont un moyen de mettre en œuvre ce paradigme.

Les conteneurs hébergent les microservices individuels qui forment une application microservices.

II. Microservices et conteneurs

Toutefois, une entreprise peut héberger et déployer des microservices de diverses autres manières :

- ❑ **Les fonctions sans serveur.** Celles-ci fournissent des environnements isolés qui exécutent du code préconfiguré pour répondre à des déclencheurs tels qu'une demande de connexion d'un utilisateur.
- ❑ **VMs.** Il est rare d'héberger des microservices dans des VM. Néanmoins, il est techniquement possible pour les développeurs de déployer un ensemble de microservices dans des VM individuelles, puis de les connecter ensemble pour former une application de microservices. Cela permet une isolation encore plus stricte entre les microservices que les conteneurs.

II. Microservices et conteneurs

Toutefois, une entreprise peut héberger et déployer des microservices de diverses autres manières :

- ❑ **Directement sur le système d'exploitation.** Il n'y a pas non plus de raison technique pour laquelle vous ne pouvez pas déployer un ensemble de microservices directement sur le même système d'exploitation et ne pas les isoler dans un conteneur ou une VM.
- ❑ **Unikernels.** Ces environnements légers et autodémarrés contiennent tout ce qui est nécessaire pour exécuter une application ou un service spécifique. Ils peuvent être utilisés pour déployer des microservices sur un serveur sans système d'exploitation conventionnel.

II. Microservices et conteneurs

Néanmoins, les conteneurs sont le moyen le plus populaire de mettre en œuvre une architecture de microservices, pour plusieurs raisons :

- ❑ **Des temps de démarrage rapides.** Les machines virtuelles prennent jusqu'à quelques minutes pour démarrer, mais les conteneurs peuvent généralement démarrer en quelques secondes seulement. Il est plus facile d'optimiser l'agilité des microservices lorsqu'ils sont hébergés dans des conteneurs.
- ❑ **Sécurité.** Les conteneurs assurent l'isolation de chaque application ou microservice conteneurisé, ce qui réduit le risque de propagation d'une faille de sécurité. Les microservices déployés directement sur un OS hôte sont moins sûrs à cet égard.

II. Microservices et conteneurs

Néanmoins, les conteneurs sont le moyen le plus populaire de mettre en œuvre une architecture de microservices, pour plusieurs raisons :

- ❑ **Découverte des services.** Il est plus simple pour les microservices de se localiser et de communiquer entre eux lorsqu'ils s'exécutent tous dans des conteneurs gérés sur la même plateforme. Si vous déployez des microservices dans des VM, des fonctions sans serveur ou des unikernels, chaque hôte peut avoir une configuration réseau différente. Il est donc plus difficile de concevoir une architecture réseau pour une découverte fiable des services.

II. Microservices et conteneurs

Néanmoins, les conteneurs sont le moyen le plus populaire de mettre en œuvre une architecture de microservices, pour plusieurs raisons :

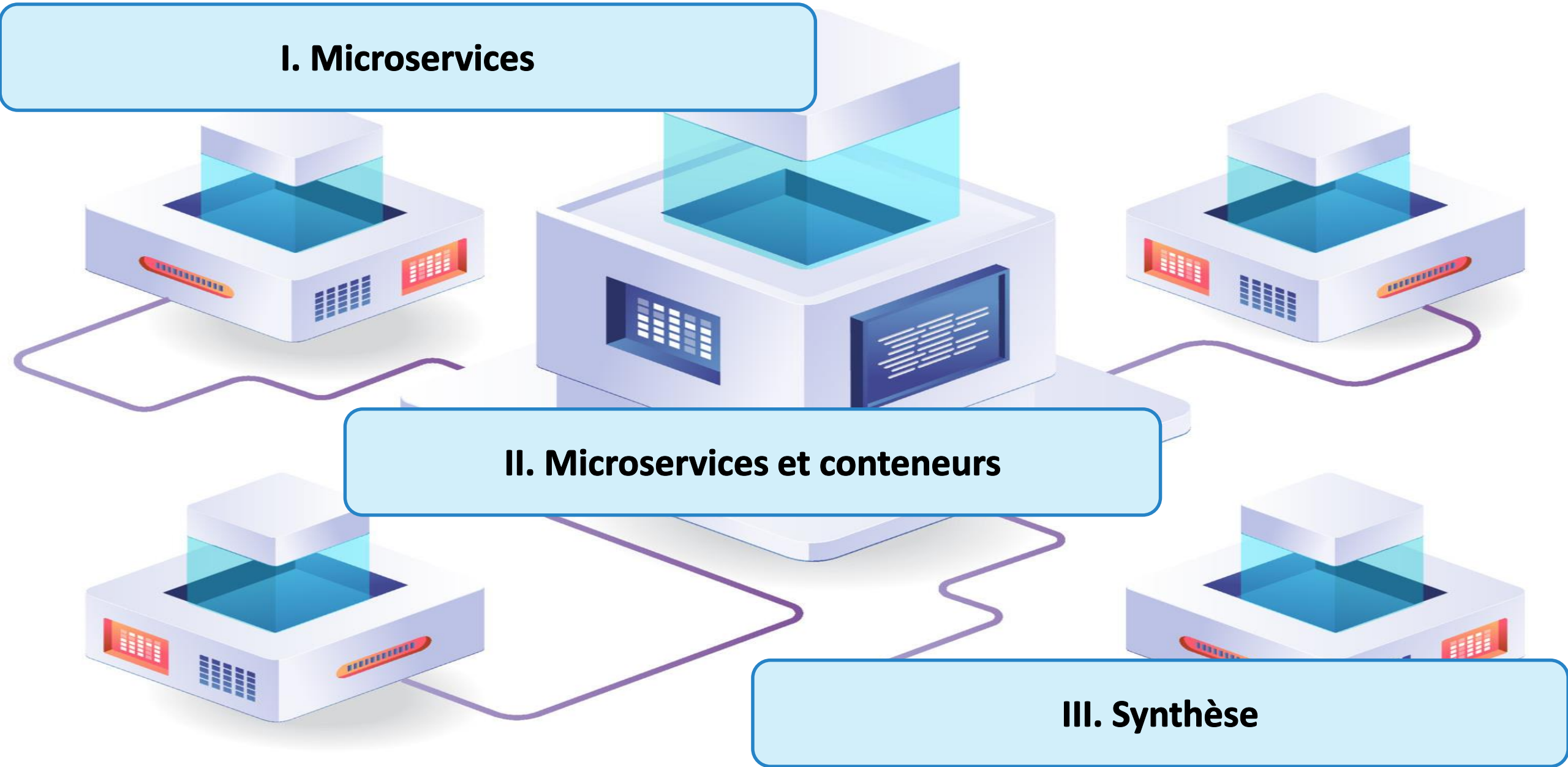
- ❑ **Orchestration.** Dans le même ordre d'idées, les microservices sont plus faciles à **orchestrer (planification, démarrage, arrêt et redémarrage)** lorsqu'ils sont exécutés dans des conteneurs sur une plateforme partagée.
- ❑ **Outils.** Les outils qui prennent en charge le déploiement de microservices avec des conteneurs ont considérablement évolué au cours des dernières années. Les plateformes d'orchestration pour les conteneurs, telles que **Kubernetes**, sont massivement populaires et bien supportées dans tous les principaux clouds publics actuels. En comparaison, il existe peu d'outils pour orchestrer les microservices hébergés dans des unikernels ou des VM.

Chapitre 3 : Les conteneurs et les micro services

I. Microservices

II. Microservices et conteneurs

III. Synthèse



III. Synthèse

En bref, pour une entreprise qui souhaite déployer des microservices, les conteneurs offrent le meilleur compromis entre sécurité, performances et gestion.

Dans certains scénarios, il est plus judicieux de déployer des microservices sans conteneurs - par exemple, une charge de travail peut exiger une isolation particulièrement stricte entre les microservices, ou plusieurs microservices nécessitent des environnements d'OS différents.

Les VMs peuvent être mieux adaptées à ces situations.

N'oubliez pas non plus que les conteneurs peuvent accueillir des applications monolithiques, et pas seulement des microservices.

III. Synthèse

Dans un avenir prévisible, il faut cependant s'attendre à ce que les conteneurs et les microservices s'intègrent d'une manière différente des autres technologies du paysage des microservices.

CONCLUSION

**MERCI DE VOTRE
AIMABLE ATTENTION!!!**