


0100

Architecture logiciel

[M347] Mettre à disposition des services avec
des conteneurs



- Architecture monolithique
 - Regroupe toutes les fonctionnalités d'une application en un seul et même système
 - Plus simple à développer et à déployer
 - Peut rendre le système difficile à maintenir et à faire évoluer
- Architecture de microservices
 - Divise une application en plusieurs services indépendants qui communiquent entre eux via des API 
 - Offre une plus grande flexibilité et évolutivité
 - Peut rendre l'application plus complexe à développer et à déployer

En résumé, l'architecture monolithique est simple mais difficile à faire évoluer, tandis que l'architecture de microservices est plus complexe mais permet une plus grande évolutivité.

Le choix entre les 2 architectures dépend des besoins et des priorités de chaque projet

Architecture monolithique

Architecture monolithique

Avantages / Inconvénients

Avantages	Inconvénients
Plus simple à développer, déployer et tester car tout est dans un seul système	Plus difficile à maintenir et faire évoluer à mesure que l'application devient plus grande
Moins de complexité pour les petits projets	Risque de dépendance élevé entre les différentes parties de l'application, ce qui peut entraîner des conflits de version ou des temps d'arrêt

- Les premières versions de Photoshop étaient des applications monolithiques où toutes les fonctionnalités étaient regroupées en un seul programme
- Les jeux vidéo “traditionnels”, tels que “Mario Bros” ou “Sonic”, sont souvent conçus comme des applications monolithiques où toutes les fonctionnalités du jeu, comme les graphismes, la musique, les contrôles et la logique du jeu, sont regroupées dans un seul programme
- Un système de gestion de contenu (CMS) tel que WordPress ou Drupal peut être considéré comme une application monolithique, car toutes les fonctionnalités sont regroupées en un seul système.

Architecture de microservices

Avantages	Inconvénients
Évolutivité horizontale, car les services peuvent être ajoutés ou supprimés en fonction des besoins de l'application	Complexité accrue en raison de la nécessité de gérer de nombreux services
Indépendance entre les différents services, ce qui permet une plus grande flexibilité en matière de développement et de déploiement	Risque de surcharge du réseau en raison de la communication entre les services
Permet une meilleure tolérance aux pannes	Plus difficile à tester, déployer et surveiller en raison

- Netflix et Spotify utilisent une architecture de microservices pour gérer leurs fonctionnalités de streaming et de gestion de compte utilisateur. Les différents services, tels que la recherche, les recommandations ou les notifications, sont tous conçus pour être indépendants et évolutifs
- Uber utilise des microservices pour gérer les différentes parties de son application, telles que les systèmes de géolocalisation, de facturation et de notification

Célèbre débat Linux vs Minix



Linus Torvalds est un informaticien finlandais, célèbre pour être le créateur du noyau Linux, un système d'exploitation open source basé sur Unix qui est largement utilisé dans les serveurs, les ordinateurs personnels et les appareils mobiles.



Tanenbaum est un célèbre scientifique en informatique, connu pour être l'auteur de MINIX, un système d'exploitation de type Unix conçu pour l'éducation et la recherche.

- Le débat Tanenbaum-Torvalds est une célèbre discussion qui a eu lieu en 1992 sur la qualité et la sécurité des systèmes d'exploitation, en particulier MINIX et Linux.
- Andrew Tanenbaum, créateur de MINIX, a critiqué Linux en affirmant que son noyau monolithique était moins sûr et moins fiable que son propre système d'exploitation, qui était basé sur une architecture micro-services.
- Linus Torvalds, créateur de Linux, a répondu en défendant son approche de conception du noyau Linux, qui privilégiait la performance et la simplicité par rapport à la modularité.
- Le débat a suscité de nombreuses réactions dans la communauté informatique et a contribué à mettre en lumière les différences de philosophie et de conception entre les partisans du micro-noyau et ceux du noyau monolithique.
- À ce jour, Linux est devenu l'un des systèmes d'exploitation les plus populaires au monde, utilisé dans de nombreux domaines, tandis que MINIX est principalement utilisé à des fins académiques et de recherche.

- Tanenbaum attaque Linux en disant que Linux est « a giant step back into the 1970s » et que son noyau est ultra-obsolète.
- Il critique aussi le lien entre Linux et l'architecture x86 en disant que dans 5 ans, elle serait complètement dépassée et plus personne ne parlera ni de Linux, ni de x86.
- Linus répond que Minix a de nombreux défauts, comme le manque de multithreading.
- Il admet quand même « that he finds the microkernel kernel design to be superior "from a theoretical and aesthetical" point of view. »
- Aujourd'hui Linux est un des OS les plus populaire

Technologies et outils

Il existe plusieurs technologies et outils qui sont couramment utilisés dans le cadre des architectures de microservices. Quelques exemples :

- Les technologies de “conteneurisation” telles que Docker et Kubernetes permettent de créer et de gérer des conteneurs isolés pour chaque service, facilitant ainsi le déploiement, l'orchestration et la gestion des applications de microservices
- Les “API RESTful” (Representational State Transfer) sont souvent utilisées pour permettre la communication entre les différents services de microservices, car elles permettent de transférer des données entre les services de manière simple et cohérente

- Les langages de programmation tels que Java, Node.js, Python et Go sont couramment utilisés pour développer des microservices. Toutefois, il n'y a pas de langage de programmation "unique" pour ce type de développement
- Les bases de données NoSQL (Not Only SQL) telles que MongoDB, Firebase, Cassandra et Couchbase sont souvent utilisées pour les microservices, car elles offrent une évolutivité horizontale et une meilleure flexibilité que les bases de données relationnelles

- Les outils de surveillance tels que Zabbix, Prometheus et Grafana sont utilisés pour surveiller la santé et les performances des services mais également et pour faciliter la détection et le dépannage des erreurs