

# Atelier 3

---

## Membre du Groupe

- Baldisserri Enzo
- Gamain Antoine
- Guillotin Antoine
- Valade Dorian

## Eléments réalisés du cahier des charges

- NA

## Eléments non-réalisés du cahier des charges

- NA

## Source vers le projet

<https://github.com/Gladear/JEECardGame/tree/msa>

## Questions de l'atelier

Quelle est la différence entre un test fonctionnel et un test unitaire ? A quoi sert la couverture de code ?

Un test fonctionnel vérifie le comportement d'une application ou d'un service dans un contexte. Un test unitaire s'assure qu'une fonction ou retourne le résultat souhaité sans prendre en compte l'environnement (par exemple, la base de données). Les tests de couverture de code permettent de vérifier que tout le code de l'application est bien utilisé, qu'il n'y a pas de code mort.

Qu'est ce qu'un test de non régression ? à quoi sert-t-il ?

Un test de non régression permet de vérifier que l'on n'a pas "cassé" une fonctionnalité lors du développement d'une autre fonctionnalité ou d'une correction.

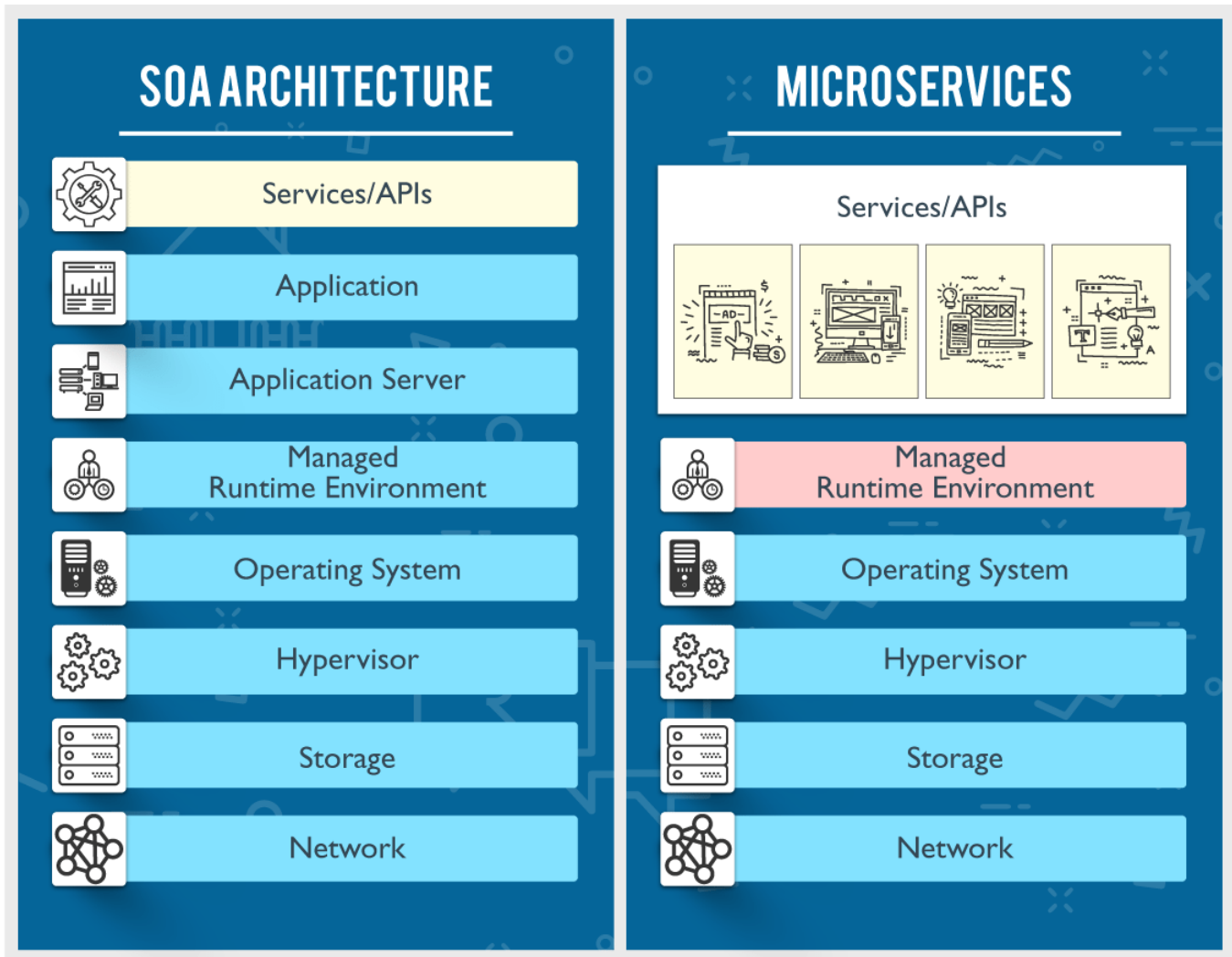
Expliquer le principe de développement « test driven » ?

Le principe est d'écrire les tests avant de coder l'application.

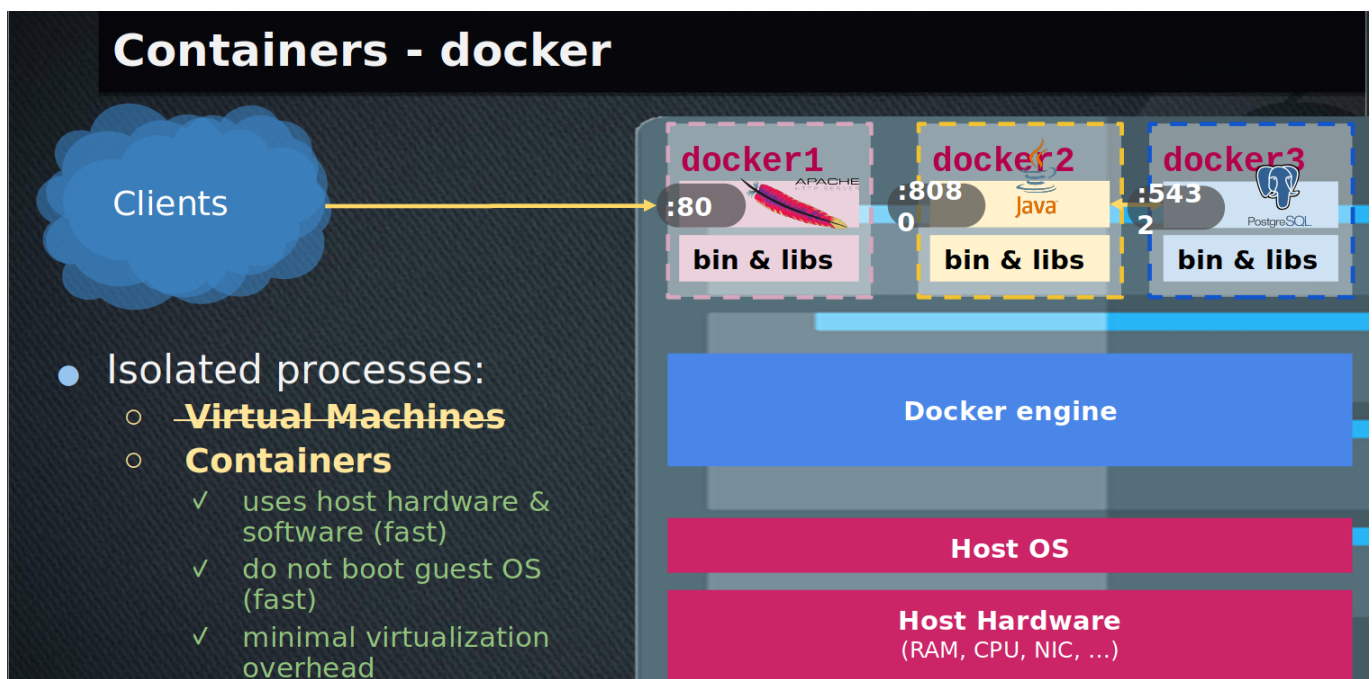
Quels intérêts présentent les micros services comparés aux architecture SOA ?

Le découplage entre tous les microservices permet de faire en sorte que toute l'application ne plante pas lors d'un problème, mais seulement la fonctionnalité liée à ce service. Cela permet aussi un déploiement beaucoup plus facile, puisque tous les services n'ont pas besoin d'être déployés en même temps.

Quelles sont les différences entre les micros services et le SOA ? Quel intérêt présente l'usage de docker et des micro-services ?



Qu'est-ce que docker ? En quoi diffère-t-il des méthodes de virtualisation dites classiques (vmware, virtualbox) ?



Quelle organisation en équipe permet la mise en œuvre de micro services ?

## AGILE

Que permet de faire l'outil Sonar ?

Sonar est un outil permettant de détecter les "code smells". Ce sont des bouts de code qui sont prompts aux erreurs, ou qui seront potentiellement difficiles à maintenir.

Qu'est ce que l'intégration continue ? Quels avantages/contraintes présentent cette organisation ?

## CI/CD Pipeline

- Continuous Integration (CI)
  - Regular integration => PR and merge
  - Automated build/tests
- Continuous Delivery/Deployment (CD)
  - Release artefact on repository
  - Automatic deployment in production

