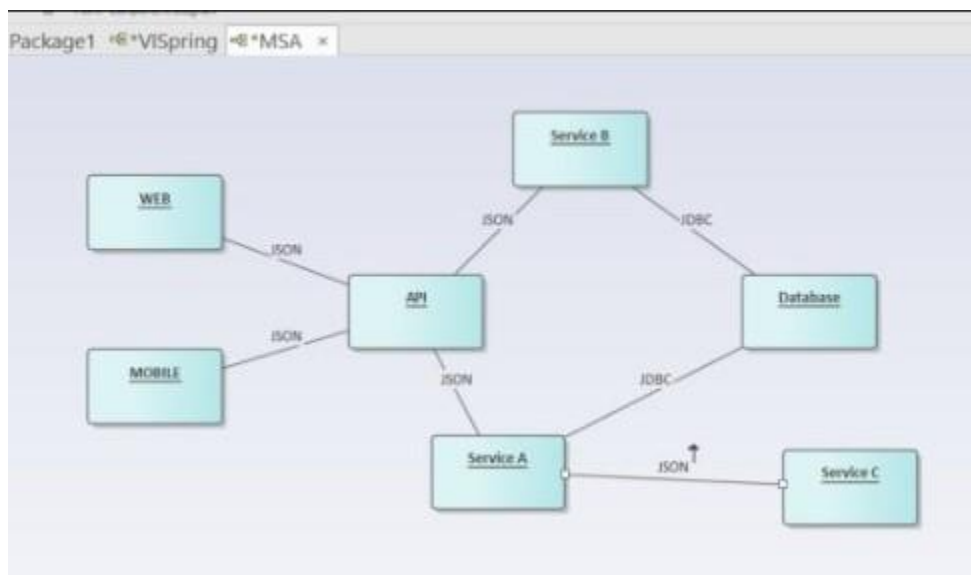
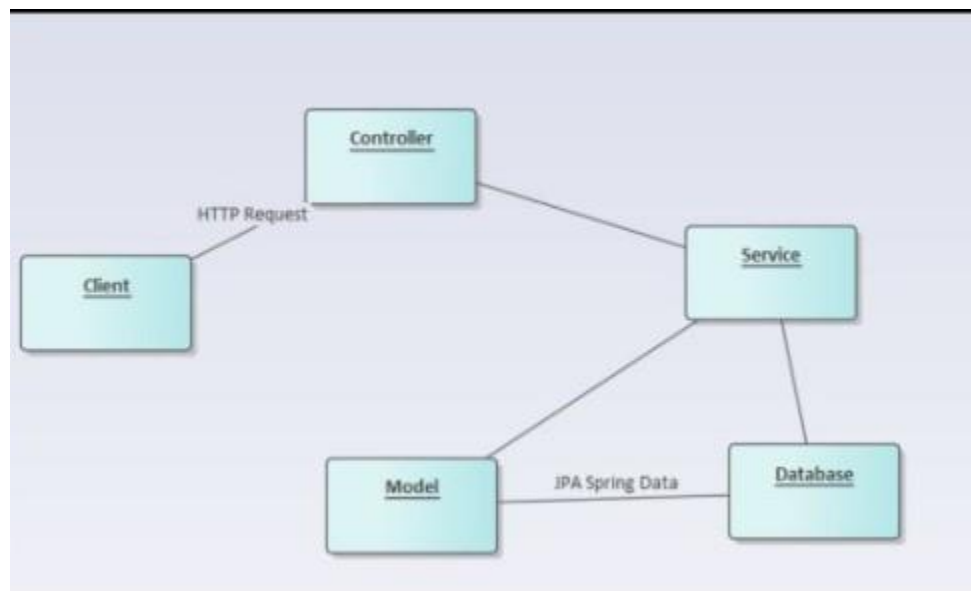


Nell David MOUKOKO

Master 2 Génie Logiciel

ISI

*Vue d'implémentation*



## **Architecture MSA et Architecture SOA**

L'architecture Microservices (MSA) est une évolution de la SOA dans laquelle les concepts de la SOA ont été repris et poussés à l'extrême car la plupart des principes de conception des Microservices étaient déjà disponibles dans le monde de la SOA. D'un point de vue conception. Alors la nouveauté réside principalement dans la description fine des design patterns à utiliser. La SOA nous offrait une boîte à outils vaste et d'excellents principes d'Architecture tandis que les Microservices nous apportent une série de bonnes pratiques et de patrons de conception plus réduits et plus adaptés aux contraintes du moment (déploiement quotidien, time to market réduit à quelques jours, interconnexion entre les applications internes et externes, relation client multi-canal).

Par ailleurs, les applications d'une Architecture SOA tournent généralement sous des serveurs d'applications (Websphere, JBoss, Tomcat, etc.) sensés garantir une homogénéité des pratiques d'administration et d'exploitation (configuration des Datasources, des logs, des URLs de ressources externes, monitoring de l'application, administration). En Architecture Microservices, on va privilégier la construction d'applications autonomes (exécutables sans conteneur d'application), basée sur des frameworks tels que Spring Boot, pour bénéficier au maximum des capacités des orchestrateurs de conteneurs comme Kubernetes telles que l'autoscaling (l'adaptation du nombre d'instances disponibles d'un service en fonction des ressources consommées à un instant T). Ces applications autonomes apportent la promesse d'arrêts/relances en un temps réduit et d'une réduction des ressources physiques consommées.

Dans la MSA, le couplage faible est primordial, à savoir que chaque Microservice doit être très indépendant à tous les niveaux. Chaque Microservice possède sa propre base de données, alors que dans une SOA, il est courant de trouver une base de données commune à plusieurs services. Théoriquement ce n'est pas censé être le cas, mais la pratique est différente. Cela s'explique par le fait que la SOA a été utilisée comme une approche pour moderniser les anciens systèmes. Or, manipuler ou re-designer les bases de données est tellement délicat que, souvent, on se contente de les optimiser.