TABLEAU COMPARATIF DES ARCHITECTURES LOGICIELLES

Logiciel	Avantages	Inconvénients
MVC (Modèle-Vue-Contrôleur)	• **Meilleure organisation et lisibilité du code.** Conception claire et efficace (séparation donnée vue / contrôleur). • **Diminution de la complexité** lors de la conception. • **Réutilisation du code** accessible. • **Maintenance et évolution rapide**. • **Souplesse** sur la répartition des taches de développement • **Facilité de mise en place de tests unitaires.**	• **Implémentation plus complexe** (ne pas confondre avec compliqué). • Répartition des tâches met aussi en jeu un **cloisonnement des développeurs** dans leurs taches respectives. • Pour les projets simples et petits, on a une **architecture complexe**.. • **Mise en place de beaucoup de fichiers.** Ce qui représente une charge non-négligeable.
SOA (Architecture Orientée Services)	• **Facilement évolutif**. • **Maintenance simple** car code et maintenance sont la responsabilité du tiers. • **Sauvegardes fiables et configuration rapide**. Les fournisseurs de services tiers peuvent fournir des informations de sauvegarde en cas de problème • **Pièces indépendante** : Un service qui tombe en panne n'impacte pas les autres car fournis par différentes parties.	• **Coût de maintenance élevé**, main-d'oeuvre élevé pour configurer et gérer chaque code de service. • **Surcharge du trafic de données**. Pour une sécurité optimale, les entrées et commandes sont vérifiées avant d'être transférées au tiers. Donc plus le nombre de services augmente plus le nombre de vérifications augmente. • **Nécessite plus de ressources,** pour assurer la vitesse du réseau.
Microservices	• **Développement facile**. Les services sont petits et faciles à comprendre. • **Déploiement indépendant**. Toute modification d'un service n'impacte pas les autres. • La structure de l'architecture s'accorde avec la répartition de travail des équipes de développement. • **Modulation de chaque service**. On peut ajuster le service pour qu'il corresponde aux besoins du projet.	• **Multiples bases de données.** Une commande va devoir mettre à jour les données dans plusieurs services. • **Test complexe.** Chaque test demande à un service et ses dépendances à démarrer ou se configurer. • Déploiement complexe et long.Chaque instance d'une application de mircoservice passe par la configuration, le déploiement, la mise a l'échelle et la surveillance.

Conclusion sur le MVC : une des architectures logicielles les plus utilisées pour les applications Web à grande échelle. Elle est toutefois lente et coûteuse sans Framework, ce qui en fait une architecture optionnelle pour les plus petits projets.

Conclusion sur le SOA: Une architecture dans laquelle on intègre des outils de services d'un tiers déjà fonctionnel. Cela assure une mise en place rapide ainsi qu'une qualité et sécurité sur le système. Le système est facilement évolutif mais entraine un coût important en main d'oeuvre et en ressources.

Conclusion sur les microservices : Les microservices décomposent l'application monolithique en petits services à forte cohésion et faible couplage selon les domaines d'activité (Dependency Inversion Principle). Ces processus sont ensuites développés et maintenus par des équipes indépendantes. Toutefois ils mettent en jeu une forte charge sur la mémoire et apportent des problèmes de latence au réseau, un grand nombre de microservices augmente également la complexité de la gestion de ces derniers.

Sources :
- https://medium.com/@belcaid.mehdi/larchitecture-logicielle-mvc-1a8bbb5cf6dc
- https://cynoteck.com/fr/blog-post/what-is-service-oriented-architecture/
- https://cryptoweek.fr/avantages-et-inconvenients-des-microservices#:~:text=Avantages%20des%20microservices%20%3A%20d%C3%A9ploiement%20ind%C3%A9pendant,et%20d%C3%A9ploiement%20complexes%2C%20forte%20automatisation.