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. D'ailleurs, dans un service de l'architecture SOA, vous pouvez trouver des éléments typiques d'un Microservice. Pour autant, cela n'en fait pas un Microservice en bonne et due forme.

Les services dans l'architecture Microservices sont très spécialisés. Ils s'occupent d'une, et d'une seule, fonctionnalité, alors que dans la SOA, la pratique est de créer des services qui gèrent un domaine, par exemple un service de gestion des utilisateurs. Dans la MSA vous aurez plutôt un Microservice de gestion des rôles des utilisateurs , un autre pour l'ajout/suppression/suspension des utilisateurs, etc. La taille des services dans une MSA est souvent beaucoup plus petite que dans une SOA.

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 sensé 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. Néanmoins, ne soyez pas surpris si vous rencontrez une MSA où il y a une base de données partagée. Une des pratiques consiste à avoir une base de données partagée mais avec, par exemple, des "*Schémas*" différents qui garantissent une indépendance totale. Le résultats est le même, il y a une relation d'exclusivité entre le Microservice et ses données, car le seul moyen d'accéder à celles-ci et de passer par ce Microservice.

La SOA vient souvent avec un grand ESB central qui s'occupe des messages et de leur transformation. Vous devinez bien que si [l'ESB](https://fr.wikipedia.org/wiki/Enterprise_service_bus), qui relie tous ces services, est en panne, c'est tout le SI qui l'est aussi. Dans une MSA, on utilise un simple système très léger de messaging, qui n'est d'ailleurs pas toujours nécessaire. De plus, il est tout à fait possible d'appeler un Microservice directement si ce système de messaging ne fonctionne pas, car il n'a pas de rôle de modification ou d'adaptation des messages, mais un simple rôle de gestion de leur queue et de dispatching.

SOAP et  REST : bien que la SOA soit tout à fait capable d'utiliser *REST* pour la communication, c'est *SOAP* qui prédomine. Ce protocole est bien plus lourd et compliqué à mettre en place qu'une approche *REST*, beaucoup plus simple et flexible.

La SOA accepte que ses composants (services et autres) communiquent avec des protocoles différents, l'ESB s'occupant ensuite d'adapter et de transformer. La MSA tend à obliger à utiliser un seul protocole et de s'y tenir. On a bien sûr moins de liberté mais l'architecture s'en trouve simplifiée tout en se débarrassant de l'ESB.