# Architecture en micro-services

L'architecture microservices est une déclinaison de l'architecture orientée services (SOA), conçue pour structurer une application en un ensemble de services faiblement couplés. Cette approche de développement logiciel vise à minimiser la redondance et le couplage entre les différents composants. Les microservices interagissent entre eux via des API indépendantes du langage de programmation.

Les avantages de l’architecture en microservices sont les suivants :

* Scalabilité : permet de mettre à l’échelle chacun des microservices indépendamment les uns des autres en fonction des besoins
* Développement agile : Plusieurs équipes peuvent travailler sur différents microservices sans se gêner.
* Déploiement flexible : Il est possible de déployer une mise à jour ou un correctif sur un seul microservices sans perturber le fonctionnement du reste de l’application
* Résilience accrue : La panne d’un microservices n’affecte pas nécessairement les autres

Les inconvénients de l’architecture en microservices sont les suivants :

* Complexité accrue : L’existence de nombreux services et bases de données distinctes complexifie le schéma architectural et la gestion globale du système
* Coûts opérationnels : Les conteneurs et/ou machines virtuelles le plus souvent utilisés par l’architecture en micro-service entraine des surcoûts comparés à une architecture centralisée.
* Problèmes de communication : Les temps de latence et erreurs de communications pouvant intervenir entre les processus peuvent poser des problèmes de performance
* Gestion des données distribuées : Maintenir la cohérence des données à travers plusieurs bases de données distribuées peut s’avérer complexe

Dans le cadre de notre exemple d’architecture en micro-services, nous avons réalisé l’architecture suivante :   
Une image contenant texte, capture d’écran, diagramme, ligne

Description générée automatiquement

Les microservices communiquent entre eux en utilisant RabbitMQ.

Le processus « CommandSource » sert d’interface client. L’utilisateur peut insérer des commandes dans celui-ci.  
Les commandes envoyés via le CLI apparaissent sur le service Monitor, et est également envoyé au Validator. Celui-ci vérifie que la commande est valide, et redirige la requête vers TextProcessor ou MathProcessor en fonction de la nature de celle-ci. Le résultat des Processor est ensuite également affiché sur le service Monitor.