

Performance improvements by using message driven architecture for microservices

Mădălin Giurcă

March 18, 2023

Topic description

In the microservices architecture a common way of information exchange is over HTTP protocol and REST calls. This is a convenient way because of the reliability and simplicity of the protocol. It doesn't require a governor of the data exchange. Although it is reliable in terms of message integrity, under high load the classic TCP characteristics and blocking calls will be a bottleneck for the system scaling.

Looking on what's happening inside a **microservices architecture**, a worthy candidate to try out is an **event driven architecture** where events will be considered to be messages. The same purpose can be accomplished by the **message driven architecture** while performance should increase due to its asynchronous characteristics (lack of feedback to a request).

This kind of architecture implies **loose coupling between system components** because every of them interacts directly only with the message bus. So that each one works independently and reacts to a message consumed from the bus.

Technology stack

Main technologies to be used for building the "laboratory setup" can be found below. The setup will be used for load testing of the same system having different implementation.

- **Docker** - for creating the environment where each system component will be contained in a container
- **JVM-based programming languages** - Java/Kotlin/Scala - for building the system components
- **Apache Kafka** - (distributed) streaming platform to distribute the messages between system components
- **Gatling** - as the load testing framework to evaluate the difference in performance between the two architectures

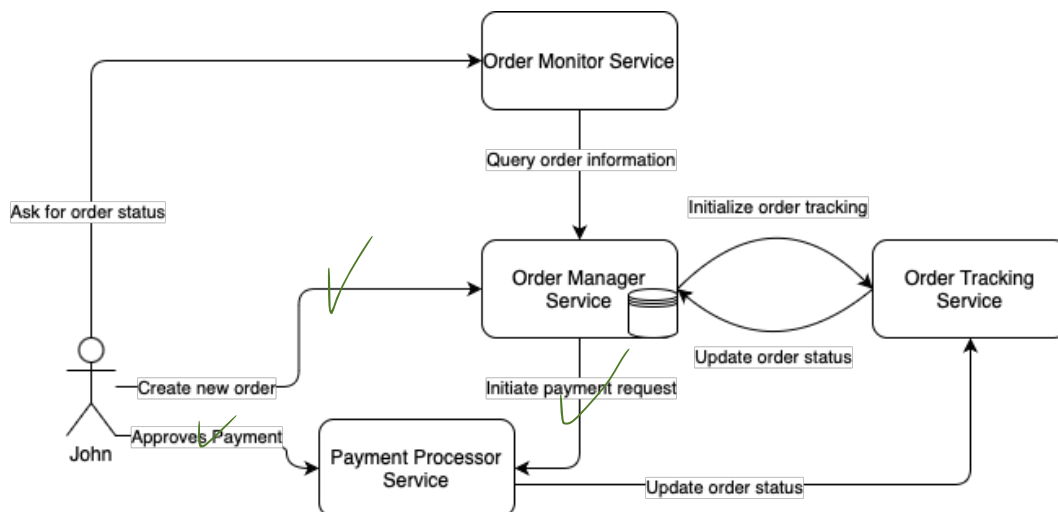


Figure 1: Diagram of the architecture overview for an order management system. This mock-up system will be implemented using the two architecture and each implementation will be load tested.