Contents

[1 Theory 1](#_Toc94488604)

[2 Architecture 2](#_Toc94488605)

[3 Implementation 3](#_Toc94488606)

[3.1 First Implementation 3](#_Toc94488607)

# Why MS

1. **Heterogenous**: Different MSs in **different programming languages** based on business functionality and requirements.  
   And can be deployed to **different operating system**.
2. **Robustness**: If one of the MSs goes down, others continue to deliver.
3. **Stateless & Scalable**: So easily scalable.
4. **Easy to Deploy**: A change or new feature can be easily deployed whereas in monolithic app, the whole app is deployed.
5. **Reusable**: A ms can be reused in some other ms whenever required.
6. **Replaceable**: As they are loosely coupled, so can be replaced by other ms.

# REST vs Messaging

1. We can use REST or Messaging for communication among MSs.
2. REST is good for Sync Request-Response Scenario because it uses HTTP protocol which is good to implement Req-Res Scenario.  
   To develop External Facing API which is exposed outside of organization.
3. Messaging can be used for async communication for non-blocking kind of app which is more reliable as messages can be persisted and they will not be lost.
4. Messaging is used for the MSs that communicate within the organization.  
   