1. Let’s see how the microservice architecture evolved over the last few decades before jumping onto the formal definition of microservices.
2. In industry, we have three kinds of architectures.
   1. Monolithic Architecture: Back in the days of 2000, monolithic was the mostly used architecture framework.
      1. In this architecture, you club all your UI, Business Logic, Data Access Layer into a single server (Apache, WebLogic Servers) and the entire server and the code present inside it can communicate with a single DB.  
         That is why we call it monolithic architecture.
      2. The entire app would be deployed as a single WAR or EAR file.
      3. But later on, organizations and teams started feeling that there is no modularity so no loose coupling so due to that issue, industry started following **SOA architecture**.  
         Let’s see what we started doing in SOA Architecture.
   2. SAO Architecture:
      1. In monolithic architecture, no modularity or loose coupling, so later on industry started following SOA architecture.
      2. In this architecture, we take certain common logic that you want to expose to outside vendors and at the same time you can also use that common logic from your application, like
   3. Microservice Architecture