

### **Unit-5: Future Trends Activity**

Based on your reading this week, could you write a section that might be appended to this paper, Salah et al, 2016, which would present the next phase of evolution history, from microservices to the technologies which are commonly in use today?

#### **Answer: IoT and Fog**

The client-server architecture is one of the most used concepts in distributed systems. The client and the server are two communicating programmes, and this is a method for illustrating their interaction. Typically, the client initiates communication by sending a request, and the server answers. Client and server may reside on the same system or on distinct systems. The server typically stores resource representations for databases or applications given by the requester (client). This model's request-response mechanism is referred to as remote procedure calls (RPC). The Client-Server paradigm was most widely used in Java settings that employed the Remote Object invocation mechanism. Due to technological advancements, the Client-Server paradigm shifted from a two-tier design to a three-tier and n-tier architecture.

Service-oriented architecture, or SOA, is the most effective extension of the client-server model. SOA enables the dynamic assembly, reuse, and loose coupling of services, hence enabling the developing business environment. SOA enables clear boundaries between independent services hosted on different hosts in order to fulfil application requirements. SOA was established in an effort to increase the reusability of a service by a variety of end-user applications in response to the problems posed by large monolithic systems. (Salah et al, 2016)

Choosing which architecture to implement is never simple. According to the author, the researched architectures were created to serve a range of purposes based on the requirements of the services for which they are used. Each architectural was described so as to propose the next one. In truth, it is vital to assess if the programme will run completely and effectively employing the selected architecture, as opposed to focusing on what is most trendy and recent. The benefits and downsides of microservices were presented in order to illustrate if the built application would need that degree of scalability, integrity, resilience, and agility with known and unknown consequences. The four shown constructions were compared in terms of the most coveted characteristics of the contemporary period. Despite the fact that microservices can handle the bulk of work, we've determined that there are still several obstacles to overcome. Despite this, further study on microservices is required, taking into consideration the communication costs, security threats, and overall application performance.