University of Essex

Masters in Cyber Security

Module-4: Secure Software Development November 2022

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

One of the most often used paradigms used in distributed systems is the client-server approach. The client and the server are two programmes that communicate with one another, and this is a technique to depict their interaction. Typically, the client sends the request to start the communication, and the server responds. It is possible for the client and server to be housed on the same system or on separate ones. The resource representations for the databases or apps supplied by the requester are frequently on the server (client). Remote procedure calls are the name for this model's request-response mechanism (RPC). Java environments that used the Remote Object invocation technique were where the Client-Server paradigm was most commonly used. The Client-Server paradigm underwent significant change as a result of technology, moving from a two-tier design to a three-tier and n-tier architecture.

The most successful extension of the client-server approach is called service-oriented architecture, or SOA. Services can now be dynamically assembled, reused, and loosely coupled thanks to SOA, supporting the evolving business environment. To meet application needs, SOA allows clear boundaries between independent services that are hosted on various hosts. Aiming to improve a service's reusability by various end-user applications, SOA was developed to address the difficulties associated with having huge monolithic systems. (Salah et al, 2016)

It's never easy to decide which architecture to utilize. Depending on the need for the services they are utilized for, the studied architectures were designed to serve a variety of functions, according to the author. Each architecture was described in a manner that led to the following one that was suggested. In actuality, it is not about doing what is most modern and fashionable; rather, it is crucial to determine whether the application would fully and efficiently operate utilizing the chosen architecture. To demonstrate if the developed application would require that level of scalability, integrity, resilience, and agility with the payoffs of known and unknowable repercussions, the pros and disadvantages of microservices were shown. The four shown structures were contrasted in terms of the most sought-after elements present in this modern era. We came to the conclusion that while microservices can handle the majority of functionality, there are still many difficulties. More research on microservices should be done, nevertheless, taking into account the communication costs, security risks, and general application performance.

References:

Salah, T., Zemerly, M. J., et al. (2016) The Evolution of Distributed Systems towards Microservices Architectures, in Proc. of the 11th International Conference for Internet Technology and Secured Transactions.