**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?

The microservices architectural approach offered numerous advantages, including the ability to deploy and build software more quickly owing to reduced component coupling and the ability to break down the program into smaller, simpler to manage and maintain components (Salah et al, 2016).

(Kanjilal, 2021), The needed amount of coupling is not always reached by microservices design alone. For instance, if middleware and microservices rely on the same libraries, any library changes would be difficult since the middleware and microservice components will still be tightly connected

However, the multiruntime microservices architecture, a new developing paradigm in software design,avoids this intimate interaction between the middleware and microservices

The two-component multi-runtime microservices (MRM) paradigm is a bit like the client-server design, except in this case both components are based on the same host, allowing for dependable networking between them.

The first part, known as Micrologic, comprises the business logic without taking into account distributed systems, and it simplifies communication by utilizing a few chosen APIs and protocols. It doesn't connect with other systems directly.

It solely communicates with the Mecha, the second part of the MRM architecture.

Each Mecha part may be configured to operate with a single Micrologic or to be shared by many Micrologic parts.

The "Mecha" component provides pre-built distributed system primitives. It may be altered, shared, and used again. Declarative configuration and JSON/YAML are used to manage the essential application states, such as the features that the services activate and the communication channels (Ibryam & Bryan, 2020). The separation of the distributed primitives and the business logic is one of the most important advantages of this architecture, notwithstanding its other advantages. The business logic was developed in a setting where vendors often deal with distributed primitives and is distinct, flexible, and dynamic. As a result, any modifications are carried out in line with the needs of the vendor's business. Because these two systems are independent, there is limited chance of cross-pollination in the growth of their autonomy (Ibryam & Bryan, 2020).

**References**

Ibryam, B., Bryant, D. (2020) Multi-Runtime Microservices Architecture. Available from: https://www.infoq.com/articles/multi-runtime-microservice-architecture/ [Accessed 15 July 2022].

Kanjilal, J. (2021) A quick rundown of multi-runtime microservices architecture. Available from: https://www.techtarget.com/searchapparchitecture/tip/A-quick-rundown-of-multi-runtime-microservices-architecture [Accessed 15 July 2022].

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.