

Architecture Recommendation

Monolithic or Microservices?

Developing information systems today can take two approaches, one where all code is maintained in a single collection of code (“monolithic”) or one whose functionalities are maintained independently within its own process. Which architecture to choose depends on the which benefits development teams wish to extract.

What are microservices?

Microservices are an evolution of service-oriented architectures (SOA) in that they maintain the distributed architecture nature of SOA where services are accessed remotely typically using Representational State Transfer (REST) protocol. However, microservices are designed, in contrast to SOA, to “share-as-little-as-possible” (Richards, 2016).

Advantages of microservices

1. Since each “service” is self-contained, it is far easier for each service to utilise technologies appropriate to fulfilling their requirements and performance goals.
2. The nature of separating services from each other enables greater resilience since the failure of one service does not affect the whole system.
3. Following from the separation, scalability concerns are more easily addressed.
4. Together with scalability is the idea that microservices can be deployed independently of each other, thereby enabling unique deployment scenarios that work with an organisation’s requirements.
5. Map well onto the idea of a *bounded context*—having minimal dependencies—that allow organisations to build focused domain services. (Richards, 2016)

Disadvantages of microservices

1. Cross-domain transactions may prove difficult to achieve.

2. Service discovery may be required because a microservice may be dynamically deployed to locations.

What is monolithic architecture?

Monolithic architectures generally consist of three application layers.

1. Presentation Layer. This layer concerns itself with the presentation of information.
2. Business (Logic) Layer. Here all business-related concerns about data are handled.
3. Database Access Layer. This layer typically handles CRUD (Create, Read, Update, Delete) operations to and from a database on behalf of the “upper” layers.

Advantages of monolithic architecture

1. In this approach to building information systems, all service interaction is contained within a single, tightly coupled codebase. This allows for a single location of change management.
2. Generally, monolithic architectures have higher performance metrics since very few requests require context switch or network traversals (Al-Debagy & Martinek, 2018).
3. Ease of testing and development given the encapsulated nature of this architecture (De Lauretis, 2019).

Disadvantages of monolithic architecture

1. All code is developed in a single programming language.
2. Scalability is difficult as the entire application must be scaled and replicated across hardware, making scalability an expensive option.
3. Evolution of information systems built using monolithic architectures complicates the codebase, leading to poor scalability and poor continuous deployment (De Lauretis, 2019).

Recommendation

Recommending microservices should be an easy win for organisations. However, this is not the case when developing small systems that may only be used by a handful of users. In this scenario, the cost-risk reward to building a system using microservice architecture is not beneficial: too much effort for little reward. In this scenario—few users—a **monolithic approach** is best suited.

On the other hand, if an organisation has the required budget, and their system is to be used by several users at once, then *scalability* and *maintainability* are key concerns. In this scenario, adopting a **microservice approach** is beneficial as the system is built for workload.

Lastly, if an organisation's systems are growing from their original intended designs, the work by De Lauretis (2019) presents a migration method for organisations to move from a monolithic architecture to a microservices architect.

References

- Al-Debagy, O. & Martinek, P. (2018). A comparative review of microservices and monolithic architectures. *2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI)*:149-154).
- De Lauretis, L. (2019). From monolithic architecture to microservices architecture. *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*:93-96.
- Richards, M. (2016). Microservices vs. Service-Oriented Architecture. Available from https://cloud.redhat.com/hubfs/pdfs/Microservices_vs_SOA_OpenShift.pdf?hsLang=en-us [Accessed 31 Oct. 2021]