Strangler Pattern at Blackboard Learn 2011

The Strangler Pattern is an architectural approach used during the migration from a monolithic application to microservices-based architecture. It derives its name from the way a vine slowly strangles a tree, gradually replacing its growth. The Strangler Pattern involves incrementally replacing parts of the existing monolithic system with new microservices. Instead of a big-bang rewrite, it allows gradual evolution, reducing risk and enabling continuous delivery.

In order to implement it we must first identify critical components or modules in the monolith that need improvement or replacement. Then we can create new microservices to handle these specific functionalities. After the microservices are completed, we then can route requests to the microservices while still maintaining the monolith for other parts. Over time you will then shift more functionality to microservices until the monolith becomes obsolete.

The Strangler Pattern provides many benefits. One is that the gradual approach minimizes disruption and allows for testing at each step. But thankfully, microservices are easier to manage and update independently, and they can scale individually based on demand. Although with benefits it does come with a few drawbacks. One being that handling data across both monolith and microservices can be complex. You have to make sure that there is proper communication between components otherwise it is doomed to fail. But you can avoid this by setting up monitoring tools to track performance and issues. There were many lessons learned from the case study. One is that you have to start small and with low-risk components to build confidence to avoid being stuck. We must also make sure to automate deployment, testing, and monitoring and keep stakeholders informed about the migration progress.

In summary, the Strangler Pattern facilitates a gradual transition from monolithic systems to microservices. By incrementally replacing components, it minimizes disruption, enables continuous delivery, and allows for independent management of microservices. While challenges exist, proper communication and monitoring mitigate risks. Lessons learned emphasize starting small, automating processes, and keeping stakeholders informed.

Sources

Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps Handbook, Second Edition*. IT Revolution.

Unna, B. (2024, March 5). *The Strangler Fig Pattern in Microservices | Baeldung on Computer Science*. Www.baeldung.com. https://www.baeldung.com/cs/microservices-strangler-pattern

‌