A Software Solution can be built independently in pieces and assembled to form one big complex application. The roadmap to this goal sites underneath.

Micro Services

Research Document

Ismail Chbiki

Table of Contents

Problem Statement **2**

Main Question **2**

Sub Questions **2**

Chosen Methods to Answer the Main Question **3**

Results **4**

Conclusion & recommendation **4**

References **5**

Works Cited **5**

Problem Statement

There are different approaches and alternatives to build a scalable and maintainable software architecture. One of these alternatives is Microservice architecture.

This document investigates this alternative and how it can help achieving scalability and maintainability.

Main Question

How to achieve scalability and maintainability in my software architecture using microservices?

Sub Questions

1. Why is it important to break down a software solution into independent components?

2. Why does Microservice Architecture Design help achieve software scalability and maintainability?

3. How does Microservice Architecture Design help achieve software scalability and maintainability?

Chosen Methods to Answer the Main Question

In order to answer the main question, the sub questions need to be answered.

Sub Question 1:

1. Why is it important to break down a software solution into independent components?

To answer this question, I will make use of Decomposition in the Workshop method because it describes and explains the importance of breaking down a complex IT system into smaller parts.

Sub Question 2:

2. Why does Microservice Architecture Design help achieve software scalability and maintainability?

The chosen methods to answer this question are Decomposition in Workshop and Design Pattern Research in Library

Decomposition describes the importance of breaking down an IT system into smaller parts.

And, Design Pattern Research shows that applying the design patterns help improving the quality of an IT system.

Sub Question 3:

3. How does Microservice Architecture Design help achieve software scalability and maintainability?

   For this question, the chosen methods are Decomposition in Workshop, Design Pattern Research and available Product Analysis in Library.

   The third technique (available Product Analysis) is chosen to read about the process adopted in the development of a complex IT system to learn about and make use of the previous experiences and outcomes for the built of a software solution.

Results

1. Why is it important to break down a software solution into independent components?

The importance of "Breaking a complex IT system or problem into smaller parts ensures its maintainability and robustness, and facilitates cooperation in large-scale software projects (Decomposition, 2022)."

2. Why does Microservice Architecture Design help achieve software scalability and maintainability?

Microservice Architecture Design helps applying the design principles which "improves the quality (and structure) of the designed software (Design pattern research, 2022)."

3. How does Microservice Architecture Design help achieve software scalability and maintainability?

As mentioned above Microservice Architecture Design help solve a lot of performance issues and achieve maintainability and scalability by splitting complex systems into smaller reusable and maintainable components by splitting the backend into more small components (layers) that communicate between each other and deliver resources after treatment to the frontend using API services.

The question now is the how and the process, which can be answered by identifying "existing solutions" and "embed their work" into mine (Available product analysis, 2022).

Conclusion & recommendation

The answer to the main question about how to apply Micro Services to my project is to make sure to apply all the design principles (SOLID) and DRY principle and by also making sure that all the components are independent and reusable.

This will help creating a maintainable and scalable application and also will help developing a performing software solution with replaceable components that can be plugged/unplugged easily without causing production break or time consumption.

References

Works Cited

Decomposition. (2022). From Ict Research Methods: https://ictresearchmethods.nl/Decomposition

Design pattern research. (2022). From ICT Research Methods: https://ictresearchmethods.nl/Design_pattern_research

Available product analysis. (2022). From ICT Research Methods: https://ictresearchmethods.nl/Available_product_analysis