

Background and introduction

Microservices

Microservices are an instance of Service-Oriented Architecture, which emphasize the decomposition of software systems into smaller, loosely coupled services. These services can communicate using various protocols, such as HTTP[1]. Some key characteristics of microservices can be found in Figure 1.

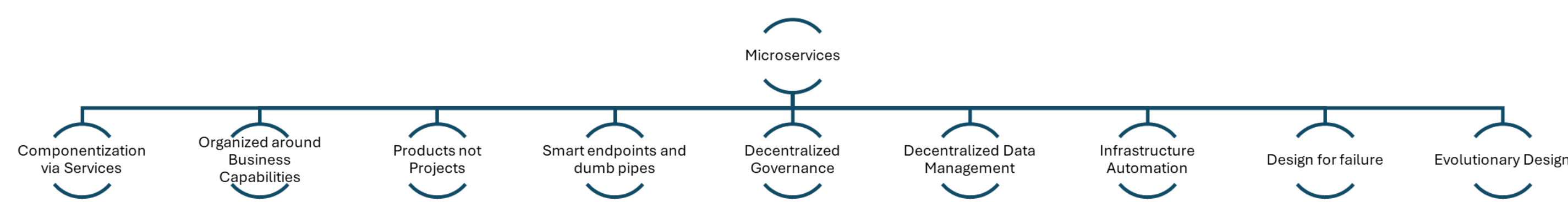


Fig. 1: Characteristics of microservices[2].

Research gap and motivation

In literature review, among first-hand data collection on management of microservices, none of the analyzed 13 key articles identifies the importance of service ownership, while the same review notes that 43% interview participants discussed the need for identifying and enforcing strict service ownership[3].

As number of microservices grow in system, so does their dependencies and overall complexity of the system. This is well presented in Figure 2, which represent complex microservice architecture. Lower level example can be seen in Figure 3, where the repositories B and C no longer have clear-cut owner. Any changes to these repositories would be deployed to both microservice 1 and 2. This introduces multiple issues, for example:

- Quality of the code
- Accountability and operational issues
- Development velocity
- Security

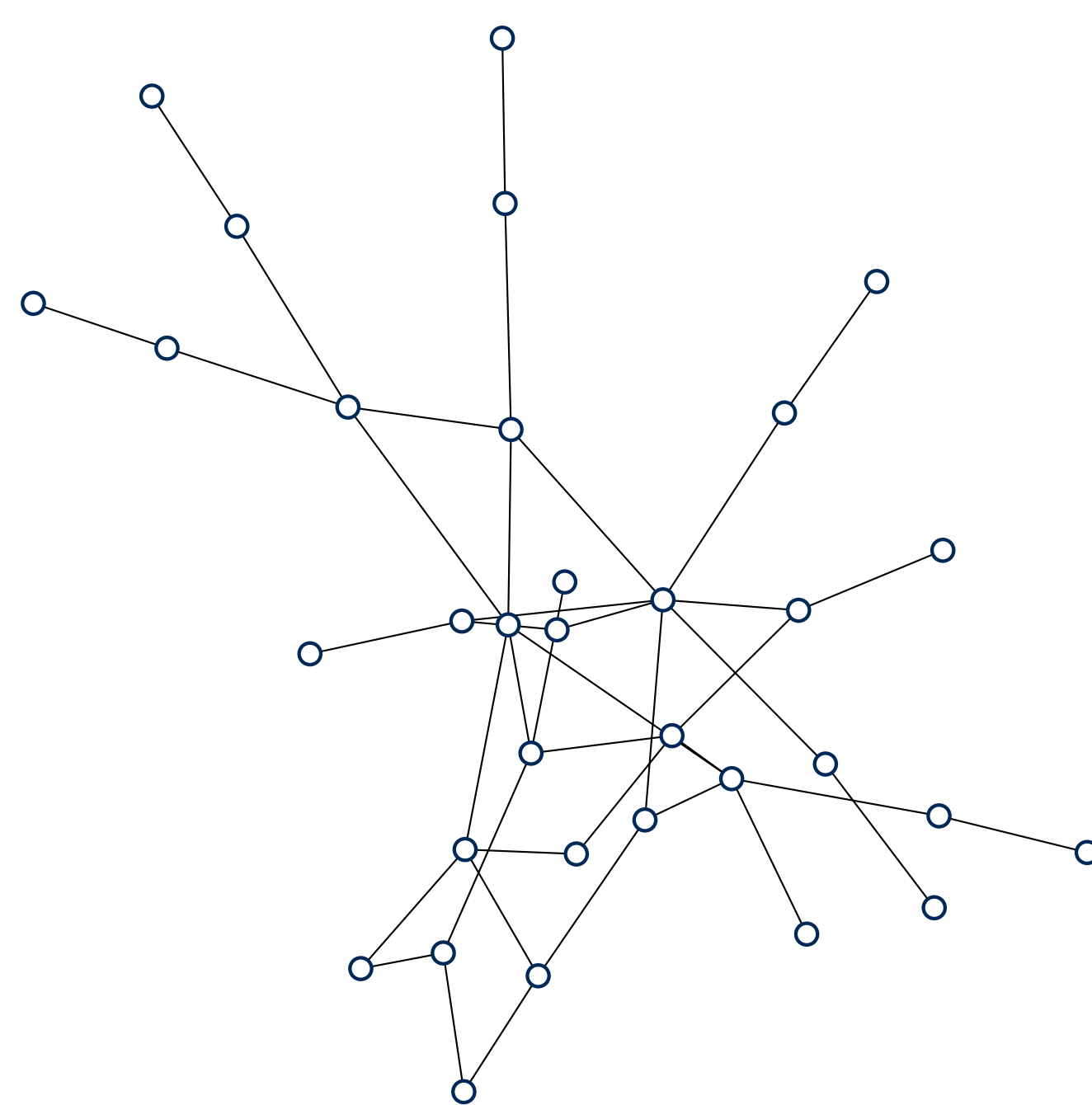


Fig. 2: Generated microservice architecture inspired by Uber microservice architecture circa mid-2018[4]. Nodes represent microservices and lines represent connections between them. How should we divide the ownership in scenario where microservices are tightly coupled?

Well researched and trialed solution will enable companies to optimize the management of microservice ownership and associated artifacts, thereby ensuring high-quality, high-velocity, and secure development by removing pain points created by unclear ownership and accountability boundaries.

Research questions

1. What are the best practices of microservice ownership?
2. What is the state of the art of microservice ownership in practice?
3. Which are the most effective ways to improve microservices through better microservice owner management?
4. Can Artificial Intelligence (AI) be used to assist in ownership management in microservices?

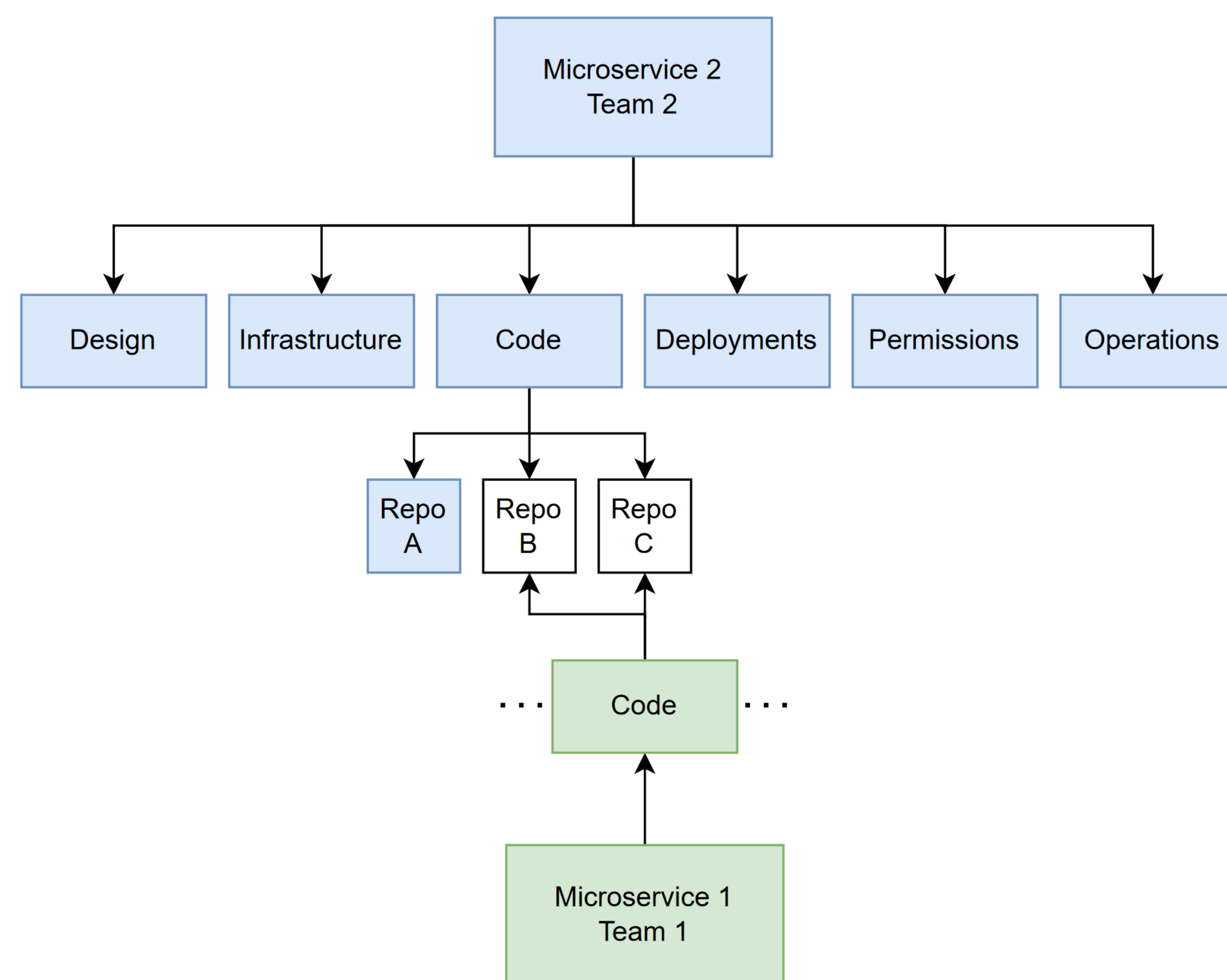


Fig. 3: Lower level view of two microservices with shared resources: who owns the code, deployments, permissions, etc.? In many situations these repositories could be used by dozens of microservices, increasing the blast radius of any issue caused by unclear ownership.

Research methods

The research should begin by investigating the current knowledge and best practices related to microservice ownership, both in academic literature and in industry. Based on the knowledge acquired in the first step, prototyping and trialling possible solutions can be started. For example what data (Git, documentation history, etc.) can be used to determine ownership? As a last step, evaluation of the solution and its resulting benefits should be conducted.

The preliminary understanding of the research methods to be used to study each of the research questions (RQ) are presented in the Figure 4.

RQ1 and RQ2

Literature review

Interviews, audits and surveys to software houses

RQ3 and RQ4

Solution prototyping, feedback and iterative development

Final solution and evaluation

Assess the impact and effectiveness of the solution

Fig. 4: Suggested research methods to deep-dive into microservice ownership issue and their relation to the research questions mentioned earlier.

Conclusion

While microservice architecture shows great promise in terms of flexibility, scalability and resilience, the overall increase in system complexity creates challenges related to ownership of microservices and associated artifacts.

Successful completion of the research should provide clear ownership solution for microservice architecture that can be utilized by companies using microservices. This should address most of the common pain points discovered during the research.

References

- [1] Eberhard Wolff. *Microservices: flexible software architecture*. Addison-Wesley Professional, 2016.
- [2] James Lewis and Martin Fowler. A definition of this new architectural term, 2014.
- [3] Yingying Wang, Harshavardhan Kadiyala, and Julia Rubin. Promises and challenges of microservices: an exploratory study. *Empirical Software Engineering*, 26(4):63, 2021.
- [4] Introducing domain-oriented microservice architecture. <https://www.uber.com/en-IE/blog/microservice-architecture/>. Accessed: 2024-08-11.