Organizational Aspects

In this lesson, we'll study some organizational aspects to architectural decisions.

WE'LL COVER THE FOLLOWING

There is a connection between decision and responsibility. Whoever makes the decision takes responsibility. Therefore, if the decision about the technology of metrics is made as part of the macro architecture, then the macro architecture group must take responsibility. For example, they would be responsible if this technology proves unsuitable in the end because it does not cope with the amount of data.

If the responsibility for monitoring microservices is completely transferred to the teams, then the teams must also be allowed to select a technology.

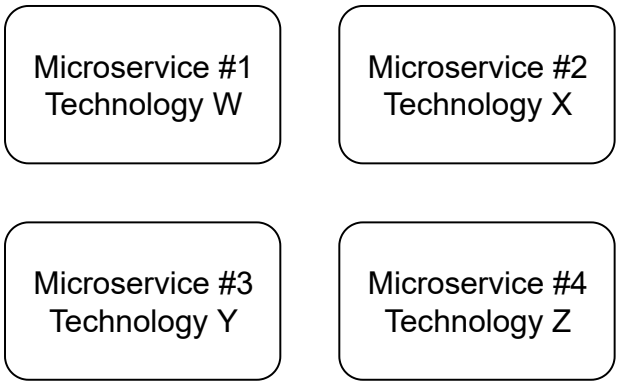
Uncontrolled growth?

The freedom regarding micro architecture can lead to **a huge number of technologies in use**. But this is not necessarily the case. If all teams have had good experiences with a particular monitoring technology, then a new microservice will most likely be monitored with the same tool. Using another tool would require a great deal of effort. Other options are only evaluated and used if the tool used so far is insufficient.

Even without a macro architecture rule, there is standardization if

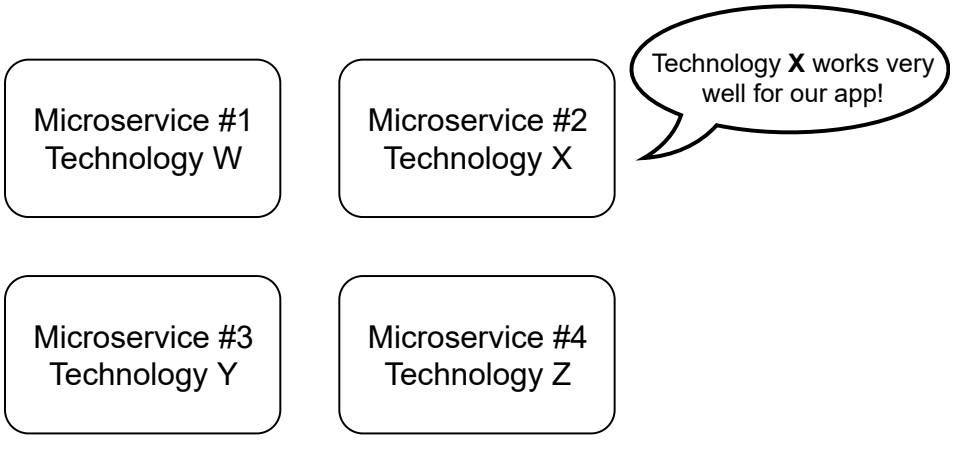
uniform decisions bring advantages for the teams. The prerequisite for this

is, of course, an exchange between the teams about best practices and about which technologies work and which do not.



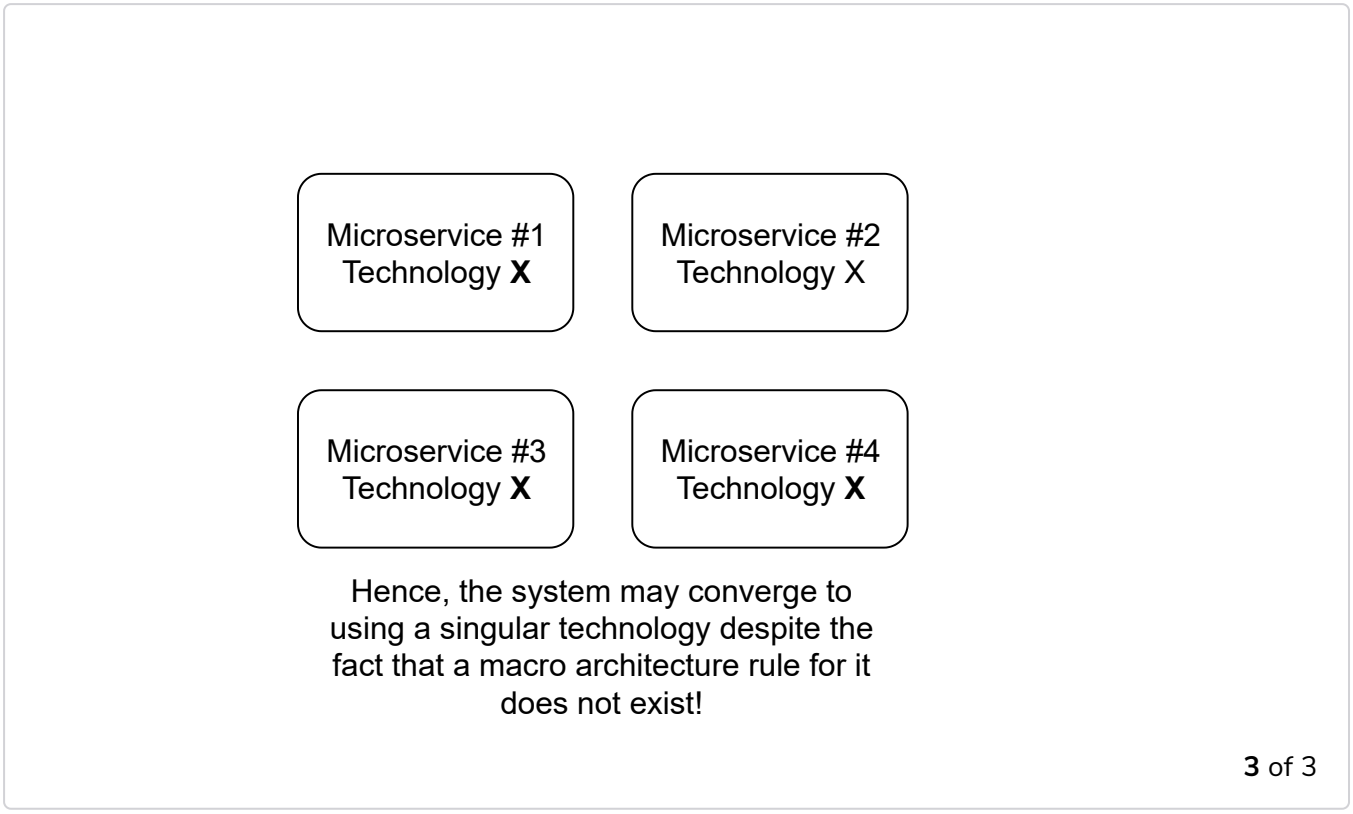
Initially, each microservice may be using a different technology

1 of 3



A team may communicate the effectiveness of a technology to the rest

2 of 3



Who defines macro architecture?

Committee of representative team members

Macro architecture restricts the freedom of the teams when it comes to implementing the microservices. This can be counteracted by having a **committee define the macro architecture**, which consists of one member of each team.

However, it is possible that the committee **may become too large to work effectively**. With ten teams, the team would have ten members and effective work is then hardly possible. You can reduce the number of members by **excluding teams** or sending **individual members as representatives for multiple teams**.

Unfortunately, the team members are often too focused on their own microservices to be interested in the overall picture of macro architecture.

Independent architecture committee

The alternative is to have an **independent architecture committee** decide on

macro architecture, which is staffed by architects who do not belong to the teams.

In such a scenario, it is important that:

- This **body's goal is to support the teams** in their development of microservices and to moderate decisions rather than enforce them. The most important work takes place in the teams. Therefore, the macro architecture should support the teams and not hinder them.
    - Collaboration between the architecture committee and the teams can also be improved by the members of the architecture committee working at least partly in teams.
- The members of the committee are **integrated and interested in the developed system**.
    - The specific domain and business requirements should never be forgotten.
    - An important part of the work on the architecture is to understand stakeholders and ensure that their goals are supported by the architecture.

How to enforce?

The need for macro architecture should be understandable because it ensures that the entire system can be developed and operated. **To enforce** the macro architecture, the team should **document reasons for each rule** to avoid unnecessary discussions behind their reasoning.

For example, certain macro architecture rules may be necessary to allow the operations team to bring the software into production, or to ensure that compliance rules are followed.

It's not so much about enforcing rules as it is about **promoting macro architecture and conveying the ideas/reasons** for macro architecture. If good reasons for changing the macro architecture exist, improving the architecture might be a better option than enforcing an obsolete one.

Testing conformance

In some cases, it is possible to test the conformance to the macro architecture by deploying a microservice and checking its log output and metrics. This

ensures that deployment, logging, and monitoring all conform to the defined macro architecture.

> Such a test is called a **black box test**, which checks the behavior of the microservice from the outside.

The **benefit** of this approach is that it:

- Does not limit the free choice of technology for implementing microservices,
- Does not enforce unnecessary standards for specific frameworks.

Therefore, testing for the conformance on the code level does not make a lot of sense.

QUIZ

1

Even if an operational technology is not standardized at the macro architecture level, certain technologies are naturally conformed to by teams that communicate well.

COMPLETED 0%

1 of 3

In the next lesson, we'll look at Independent Systems Architecture principles.