# Introduction

In this lesson, we'll walk through what this chapter holds for us.

WE'LL COVER THE FOLLOWING



## Motivation

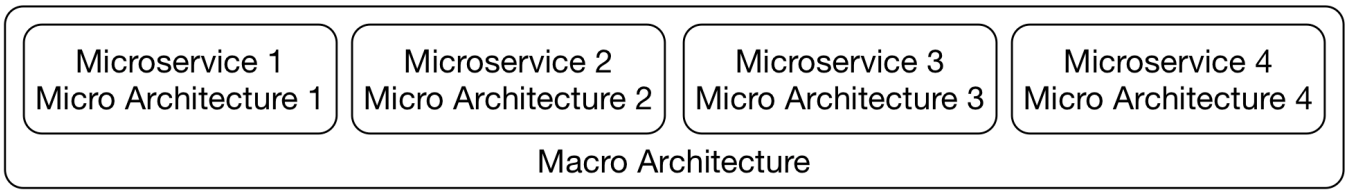
Microservices provide much better decoupling. Therefore, they help to modularize and isolate software modules (see Advantages). However, microservices are modules of a larger system. Therefore, they must be integrated. This poses a challenge for the architecture:

- On the one hand, the architecture has to **ensure that the microservices can work together** to form the overall system.
- On the other hand, the freedom of the **microservices should not be too restricted since this would compromise their isolation** and independence which are required for most of the benefits of a microservice architecture.

## Definition

For this reason, it is advisable to divide the architecture into a micro and a macro architecture.

- The **micro architecture** comprises all decisions that can be made individually for each microservice.
- The **macro architecture** consists of all decisions that can be made at a global level and apply to all microservices.




Micro and Macro Architecture

The drawing above illustrates this idea. The overarching **macro** architecture applies to **all microservices**, whereas the **micro** architecture deals with **individual microservices** so that each microservice has its own microarchitecture.

## Chapter walkthrough #

This chapter illustrates the following:

- The **division of domain logic** into microservices. *Domain-driven design* and *bounded context* are great approaches for such a division.
- The decisions that are part of the *technical micro and macro architecture* and how a **DevOps model** affects these decisions.
- The question of **who** divides the decisions into micro and macro architecture and creates the macro architecture.

## QUIZ

1

The e-commerce system discussed in the last chapter, can be divided into microservices like so:

- ordering
- registration
- product search

Suppose the product search team decides to optimize search with a new algorithm. Is this a micro or macro architecture decision?

In the next lesson, we'll look at domain-driven design and an introduction to bounded contexts and strategic design.