Definition

In this lesson, we'll introduce asynchronous microservices.

Asynchronous microservices are different from synchronous microservices, which are covered in depth in Chapter 9.

> A microservice is **synchronous** if it makes a request to other microservices while processing requests and waits for the result.

The logic to handle a request in the microservice might therefore not depend on the result of a request to a different microservice.

So, a definition of **asynchronous microservices** would be:

> A microservice is **asynchronous** if:
> **(a)** It does not make a request to other microservices while processing requests. OR
> **(b)** It makes a request to other microservices while processing requests and does not wait for the result.

There are **two cases here**, let's discuss each.

No communication

The microservice **does not communicate at all with other systems** while

processing a request. In that case, the microservice will typically

communicate with the other systems at a different time, see the drawing below.

For example, the microservice can replicate data that is used when processing a request. In this way, customer data can be replicated so that when processing an order, the microservice can access the locally available customer data instead of having to load the necessary customer data for each request via a request to another system.

Communication Only Outside of Requests
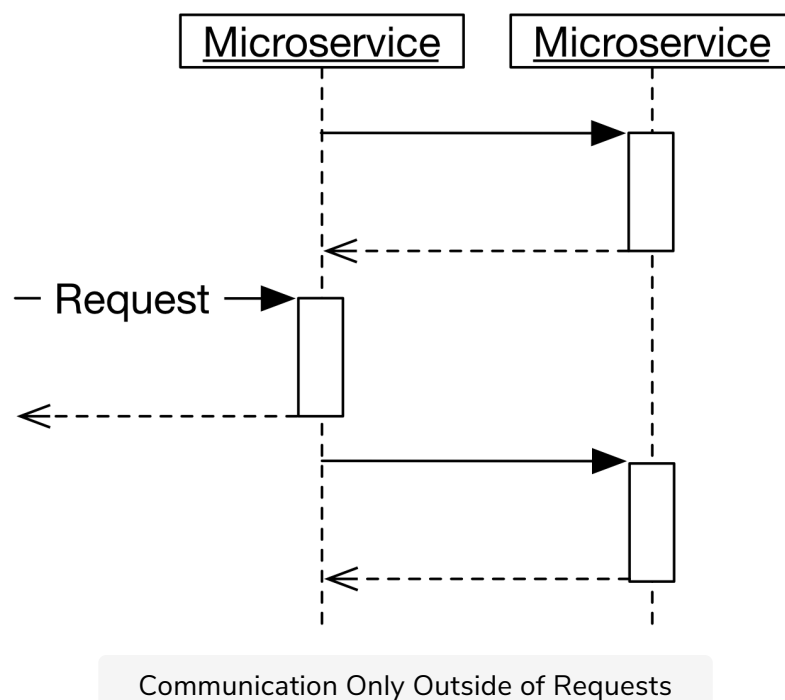
Does not wait for a response

The microservice sends a request to another microservice but **does not wait for a response**, see the drawing below.

A microservice responsible for processing an order can send a request to another microservice which generates the invoice. A response to this request is not necessary for processing the order so there is no need to wait for it.
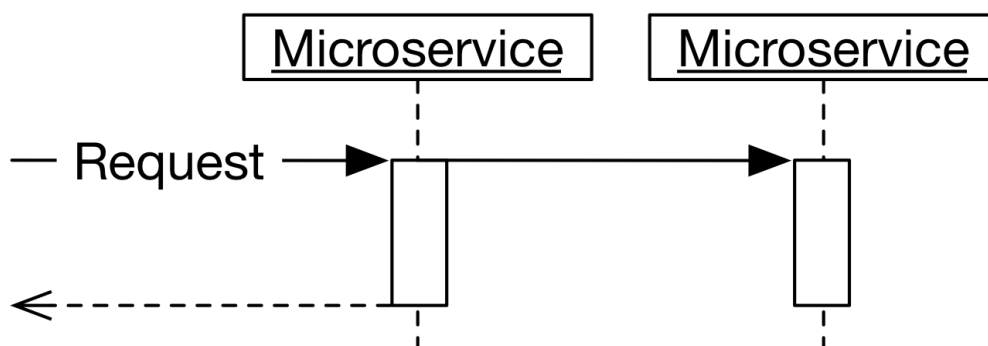
The drawing below shows an example of a more complex asynchronous architecture.

In this e-commerce system, orders are processed in the following way. It starts when customers can choose goods for an order through the catalog.

- The *order process* **generates the orders**.

- An *invoice* and a *shipping* **data record is produced** for the order.

- The *registration* microservice **adds new customers to the system**.

- The *listing* microservice is **responsible for new goods**.
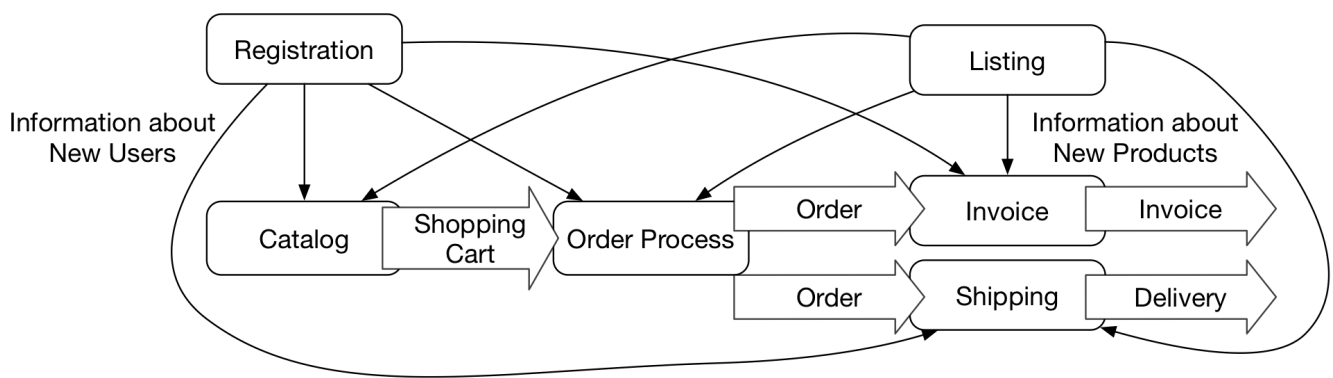
Architecture for an Asynchronous System

Asynchronous communication with no response
#

The four systems, *catalog*, *order process*, *invoice*, and *shipping*, send asynchronous notifications for processing the orders.

- The **catalog** collects goods in the shopping cart. If the user orders the shopping cart, the *catalog* transfers the cart to the *order process*.

- The **order process** turns the shopping cart into an order.

- The order then becomes an invoice and a delivery.

Such requests **can be executed asynchronously**. No data has to flow back. The responsibility for the order is transferred to the next step in the process.

QUIZ

1

Suppose a microservice sends off a request for a resource but resumes processing. What kind of microservice is this?

In the next lesson, we'll learn about data replication, bounded contexts, and communication protocols.