

# Microservices

# What is a Microservice?

**Microservices** are a [software development](#) technique—a variant of the [service-oriented architecture](#) (SOA) architectural style that structures an [application](#) as a collection of [loosely coupled](#) services...

In a microservices architecture, services are [fine-grained](#)...

The benefit of decomposing an application into different smaller services is that it improves [modularity](#). This makes the application easier to understand, develop, test, and become more resilient to architecture erosion.<sup>[1]</sup>

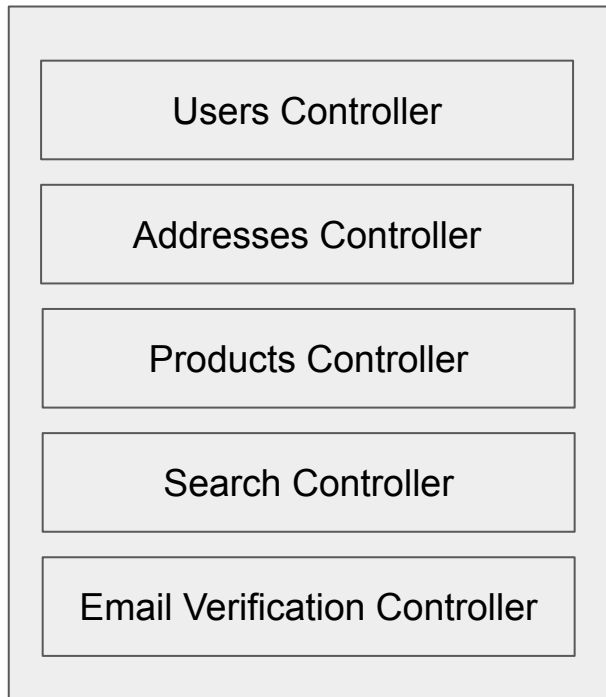
It parallelizes [development](#) by enabling small autonomous teams to develop, [deploy](#) and scale their respective services independently.<sup>[2]</sup>

# What is a Microservice

- A Web Service
- Small and Responsible for one thing (Search, Password Reset, Email Verification)
- Configured to work in the Cloud and is easily scalable

# Microservice vs Monolithic Application

## Monolithic Application



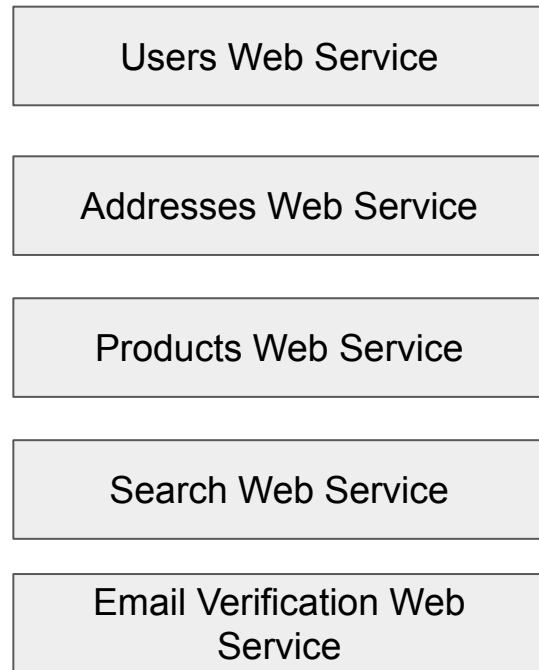
Spring Boot, Spring MVC

Spring Boot, Spring MVC

Jersey 2, JAX-RS

PHP

## Microservices



Cloud

Mobile Device

Web Browser

HTTP Request

API Gateway

Load Balancer

Config Server

Service Discovery

Users Microservice

Products Microservice

Products Microservice

Search Microservice

Search Microservice

Checkout Microservice

Checkout Microservice

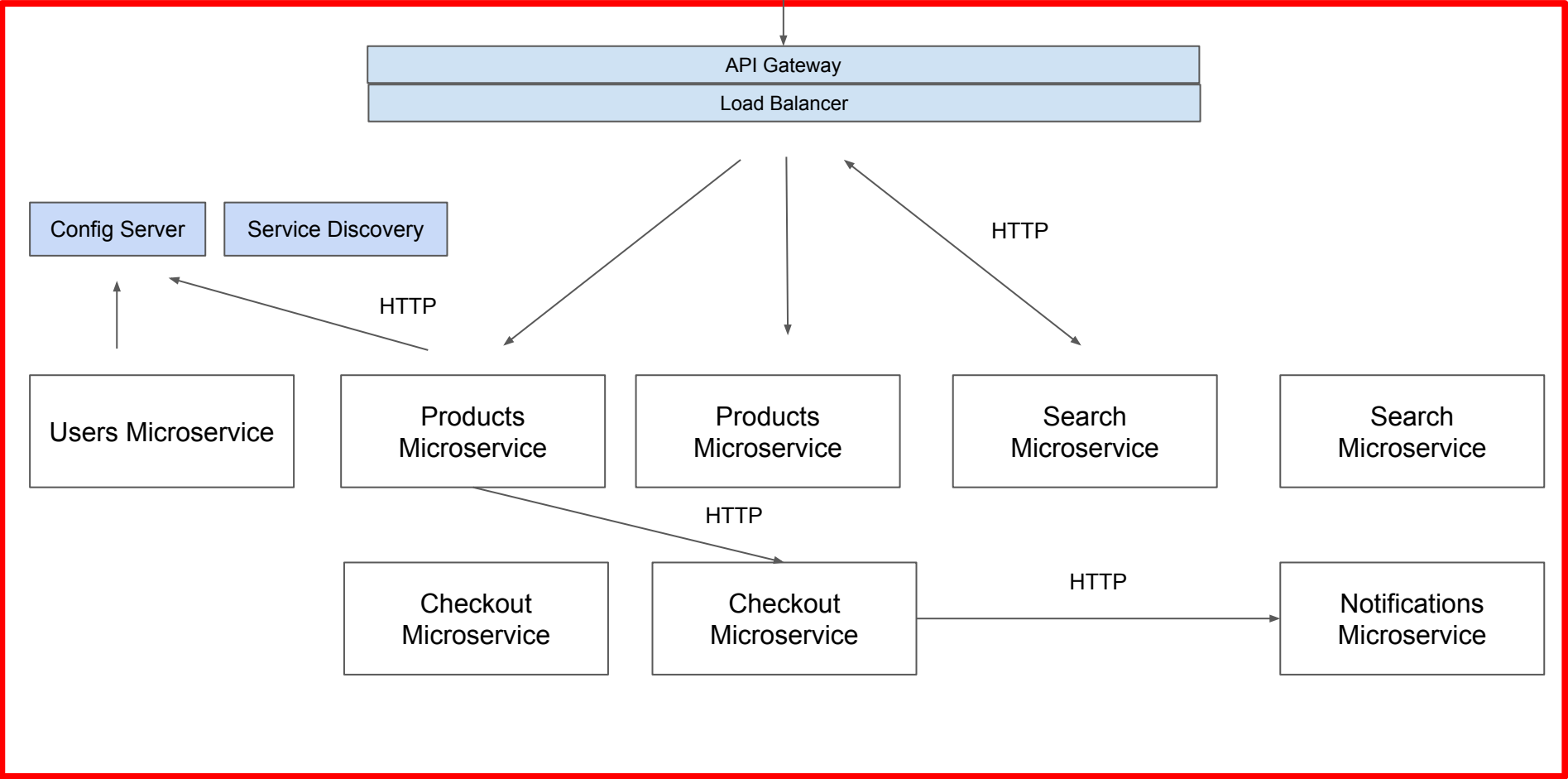
Notifications Microservice

HTTP

HTTP

HTTP

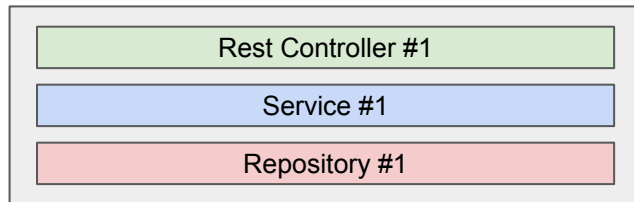
HTTP



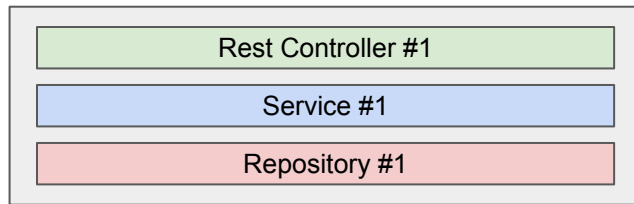
# What is a Microservice

- RESTful Web Service
- Small and Responsible for one thing (Search, Password Reset, Email Verification)
- Configured to work in the cloud

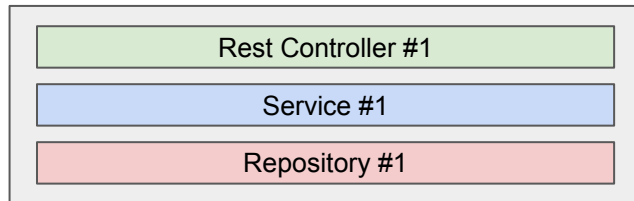
## Users



## Addresses



## Search



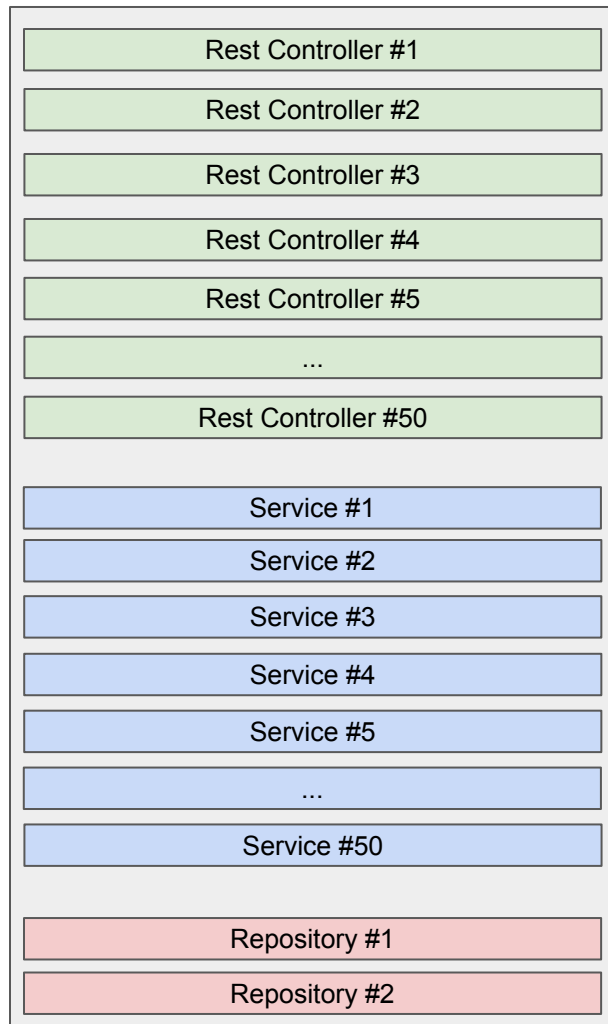
# Monolithic Application

## Advantages:

- It is easier to begin with,
- Faster initial development time,
- Easier to deploy 1 app, than 20 independent apps.

## Disadvantages:

- Harder to maintain, harder to release changes,
- Difficult for new developers to understand,
- All done with one technology.



# Microservices

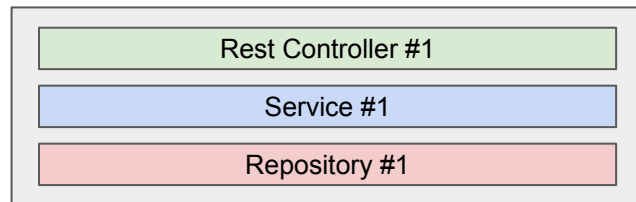
## Advantages:

- Small and focused on one domain or functionality,
- Independent,
- Easier to refactor, rework and use new technology.

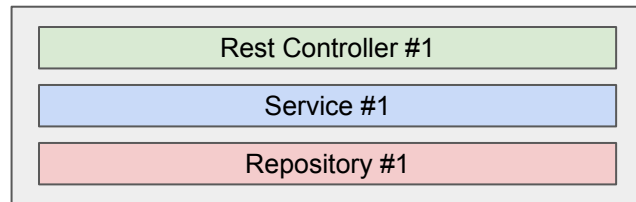
## Disadvantages:

- Slower to begin with,
- Slower to develop entire application if doing it alone,
- More complex architecture,

### Users



### Addresses



### Search

