Microservices: Getting Started

UNDERSTANDING THE BENEFIT OF MICROSERVICES

Overview

What are “microservices”?

Limitations of “monoliths”

Challenges of microservices

"The microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms."

James Lewis and Martin Fowler, Thoughtworks

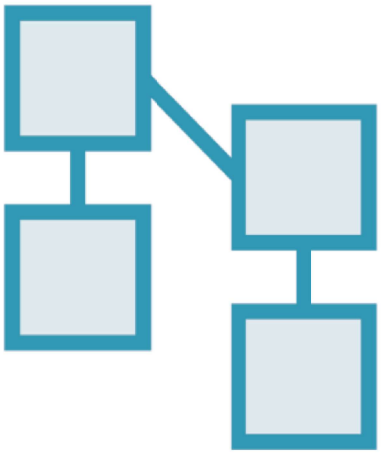
"Microservices are small, autonomous services that work together."

Sam Newman, Thoughtworks

What is a monolith?

A Monolith Is Typically...

Single Integrated Application



Adding features increases...

- Size
- Complexity
- Effort of next change

"Monoliths"

Single codebase

Single process

Single host

Single database

Consistent technology

Monolith Benefits

Simplicity

One codebase

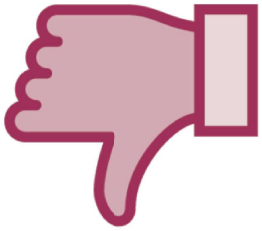
- Easy to find things

Deployment

- One application to replace

Monoliths are not “wrong”

Monolith Problems



Difficult to deploy

- Risky
- Requires downtime

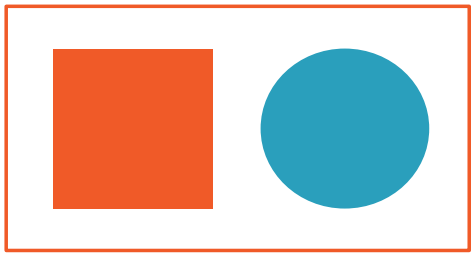
Difficult to scale

- Horizontal scaling often not possible
- Vertical scaling is expensive
- Whole application must be scaled

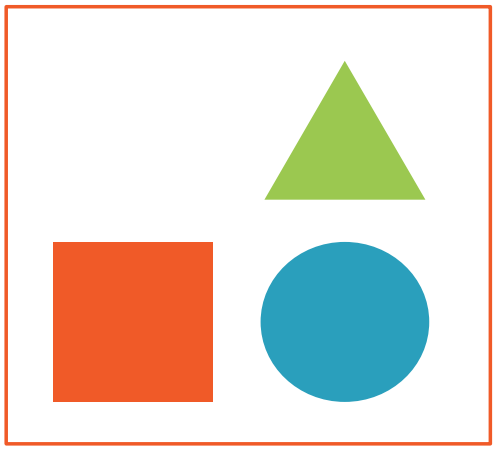
Wedded to legacy technology

- Reduces agility

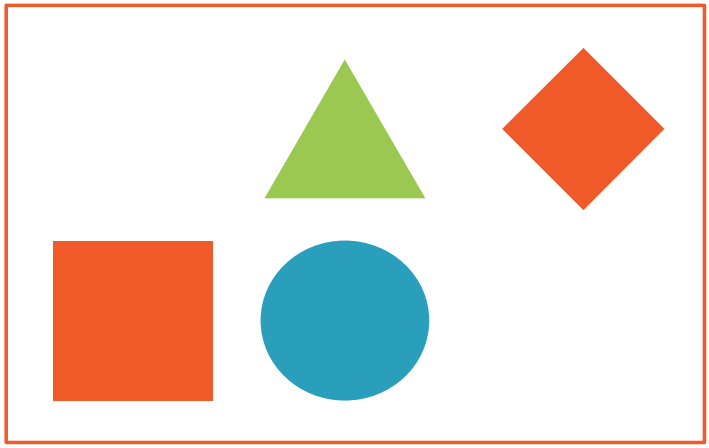
Monolithic Architecture



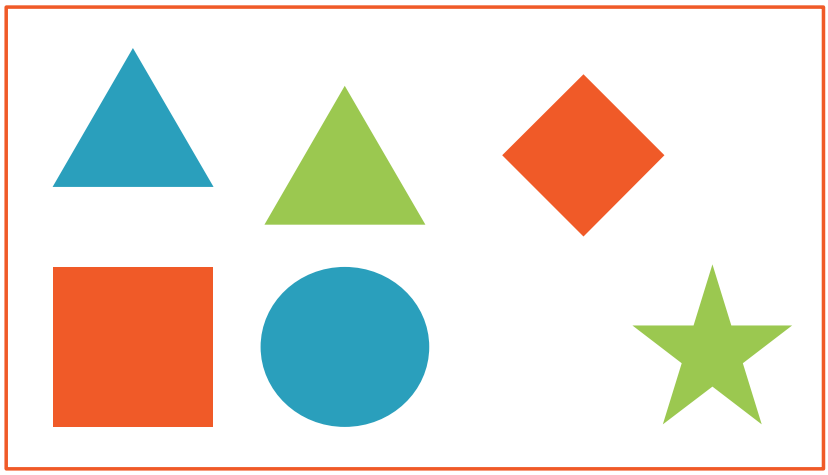
Monolithic Architecture



Monolithic Architecture

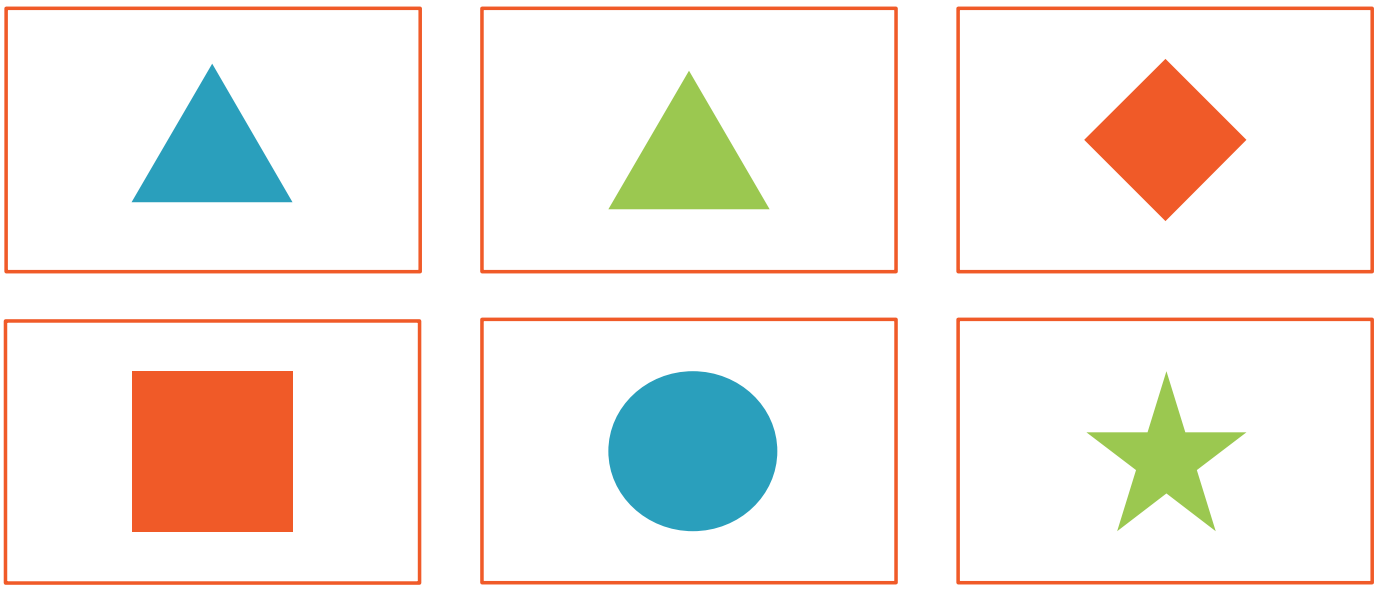


Monolithic Architecture



What are microservices?

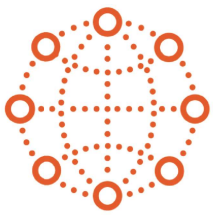
Microservices Architecture



Microservices are **small.**

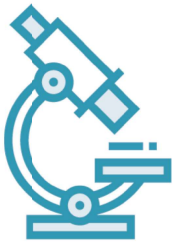
Microservices

autonomous, independently deployable services...



...collaborate to form an application

How small should they be?



Why do we need microservices?

The Problem of Scale

Monoliths can work well for small applications

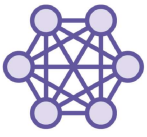
Many developers

Many users

Big data

Difficult to maintain

Technical debt

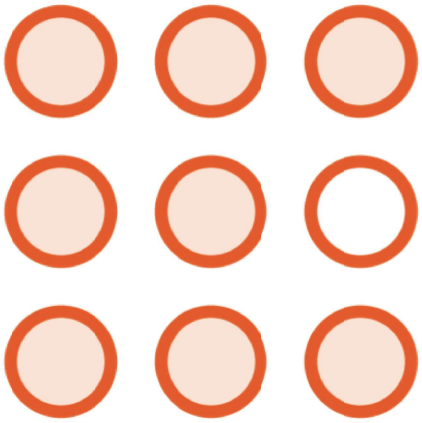


Entangled modules

Microservices Give You Choice

Microservices do not dictate technology choices

Use your favorite programming language and tools

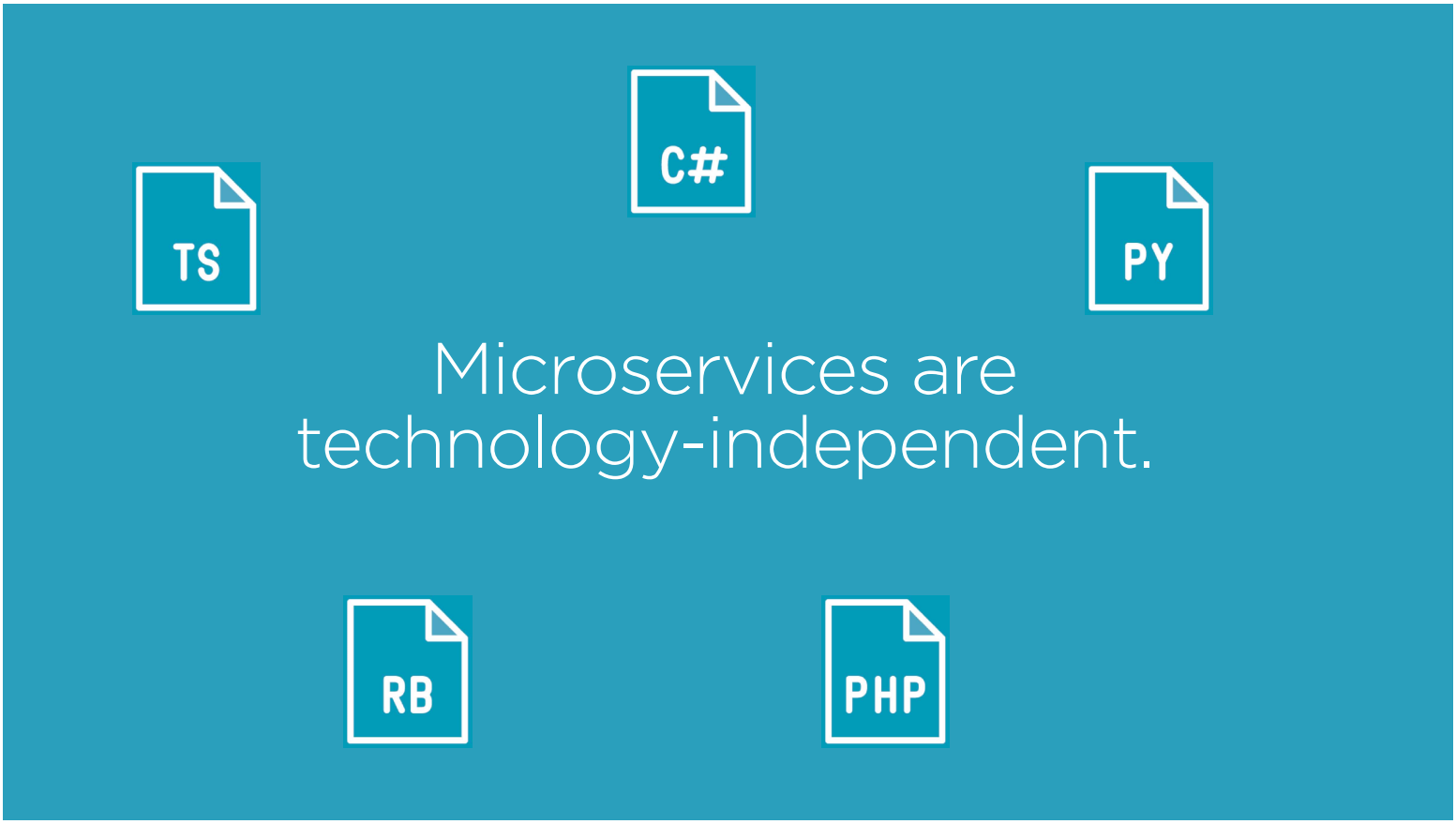


Microservices are independent

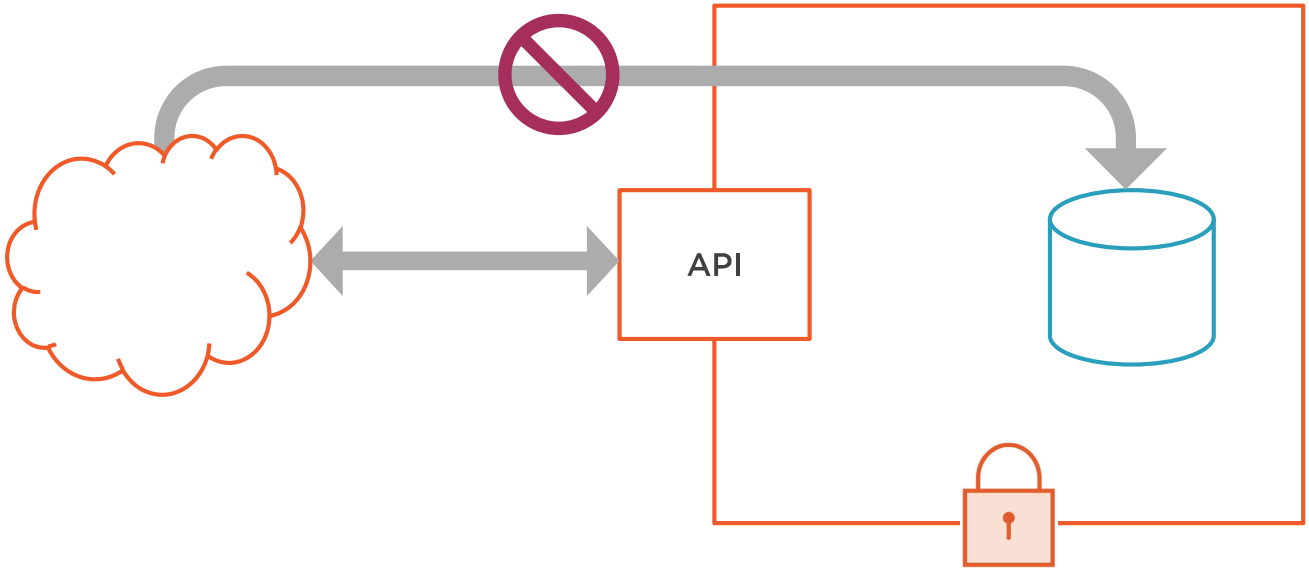
No inducing downtime in other services

Different processes

- Different servers?
- Different clouds?



Communication Only Via the API



Human Advantages of Microservices

Efficiently managing large teams and large applications

Maintenance Advantages of Microservices

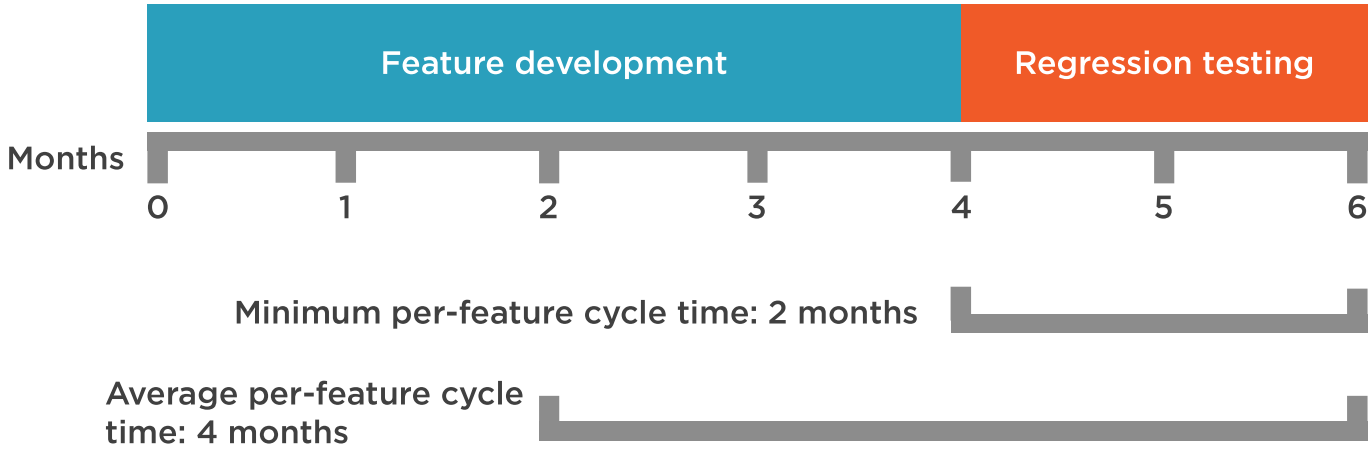
Clean break

Deliver new functionality faster

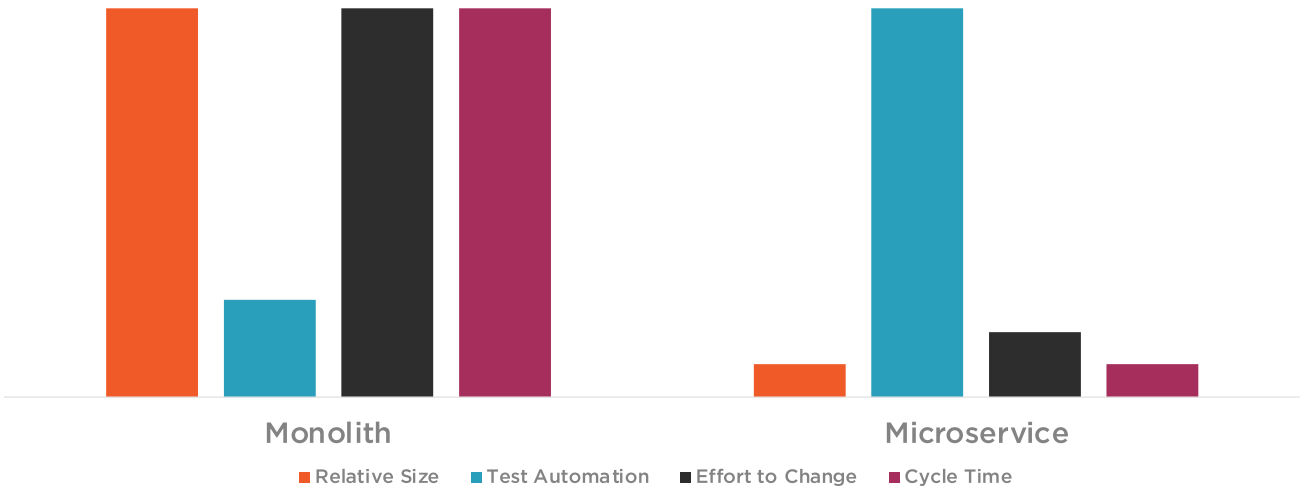
- Higher quality

The “Micro” in Microservice Means “Small”

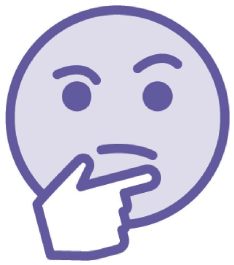
Release Cycles in a Poor-quality Monolith

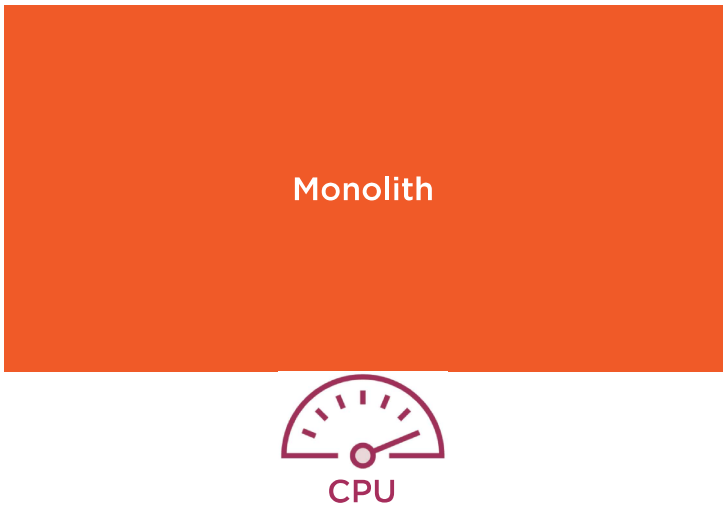


Cycle Time and Effort

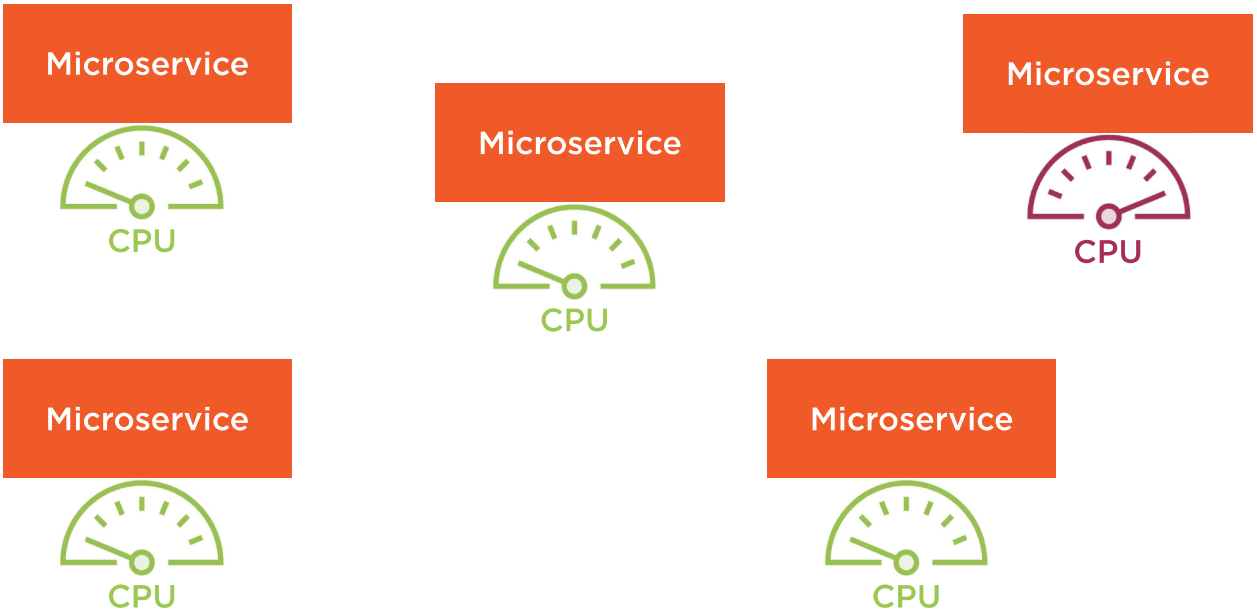


Performance and Scalability

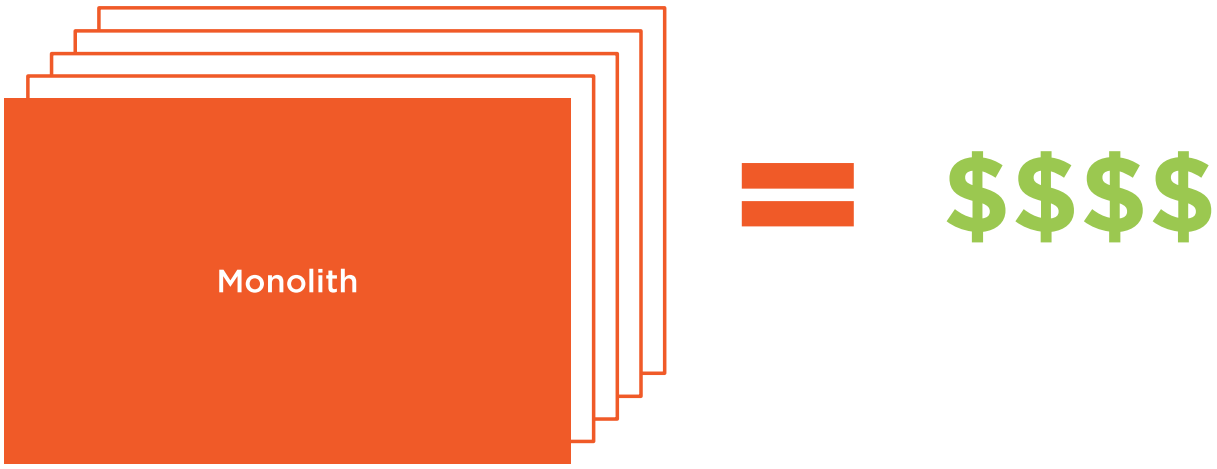




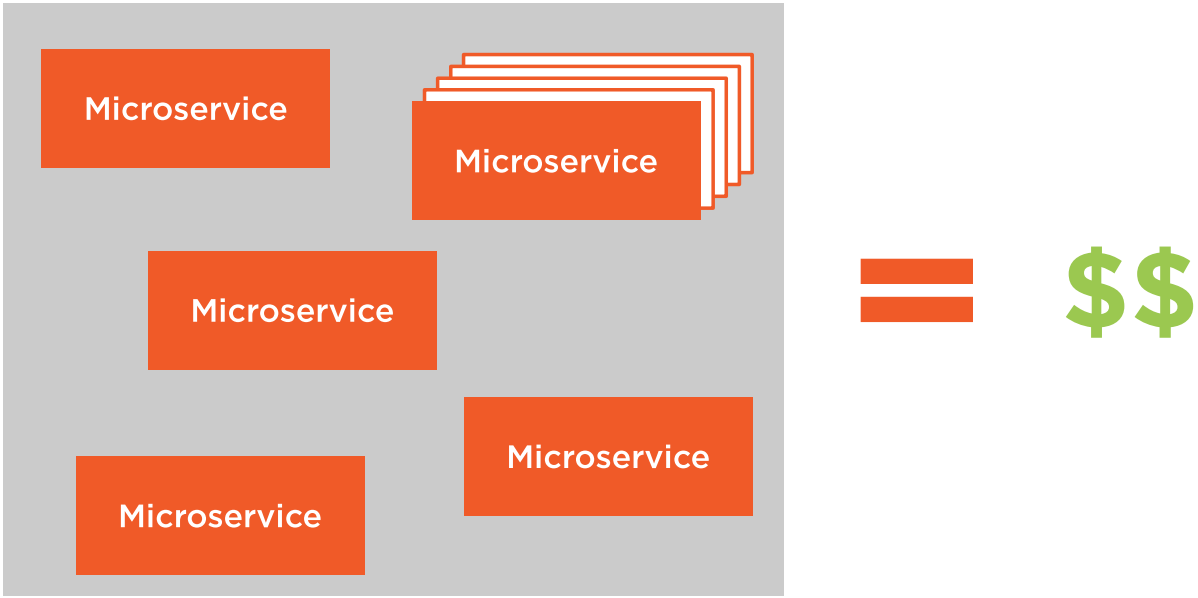
Performance and Scalability



Scaling Monoliths Can Be Expensive



Performance and Scalability



Scaling monolith is too expensive

- Scaling is all-or-nothing

Auto-scaling a microservice can limit costs

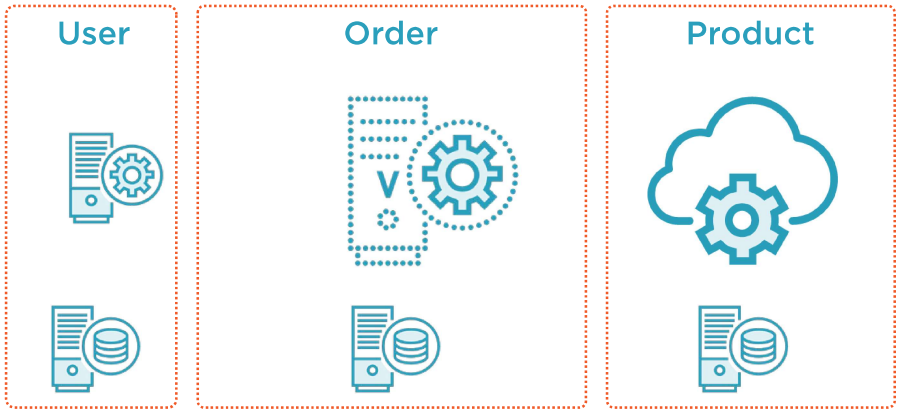
Host

Physical server

Virtual server

On-premise

In the cloud

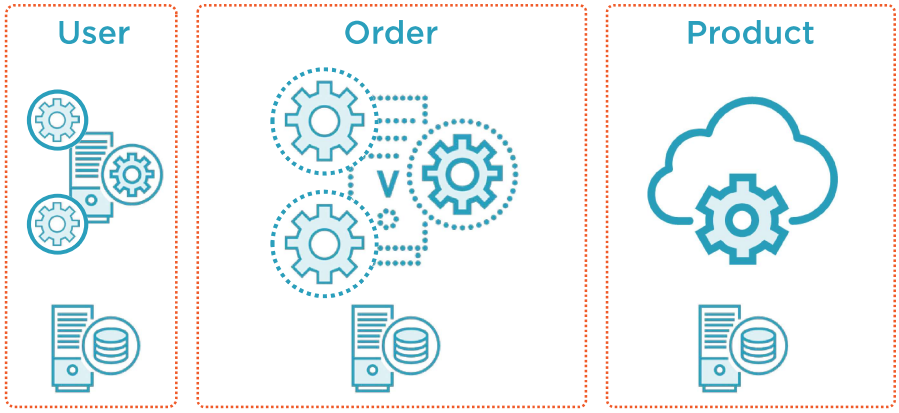


Multiple Services per Host

Microservice per host

Several services per host

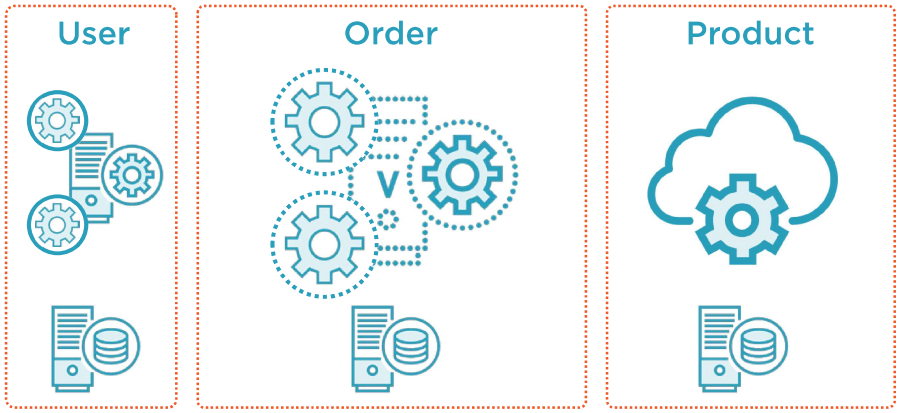
Several services per virtual host



Containers

Packaging microservice

With dependencies



Containers

Packaging microservice

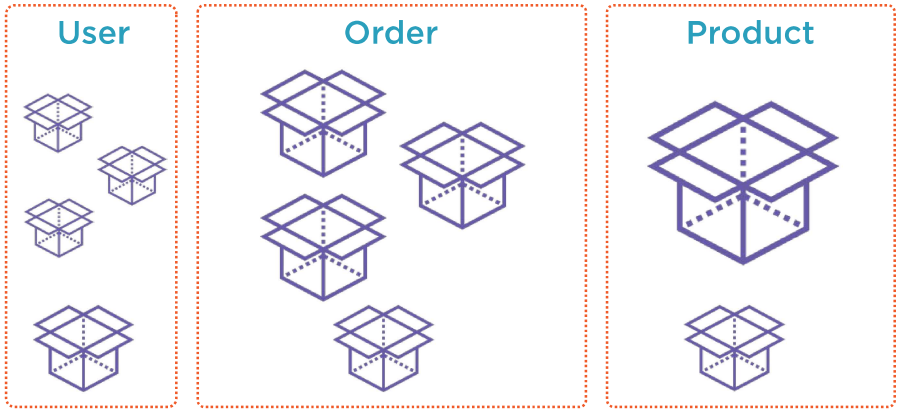
With dependencies

Container image

Easy to move from environment

Scale up and down

Docker, rkt



Orchestrator

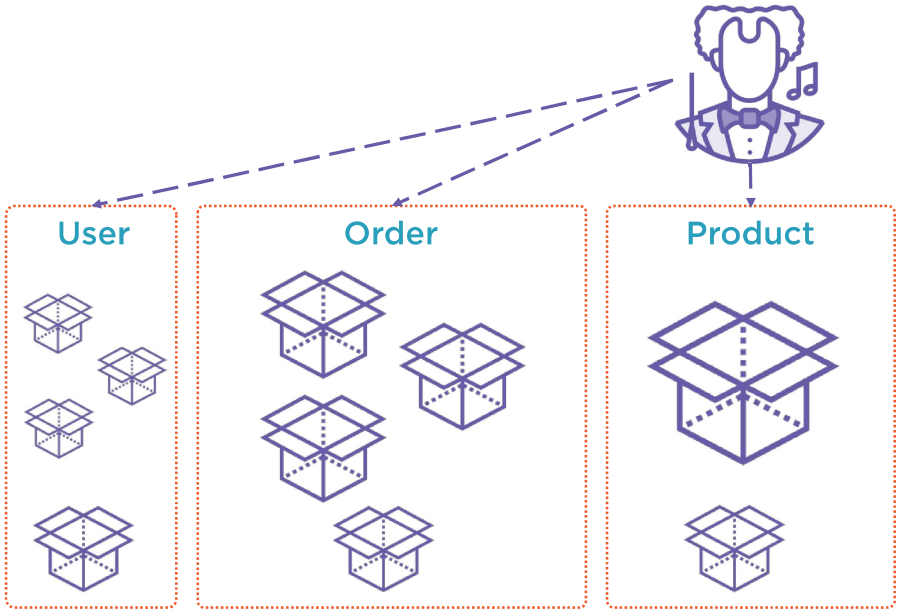
Multiple containers

Multiple machines

Start at the right time

Failed containers

Kubernetes, Mesos, Docker Swarm



Summary

Microservice architecture

- Comparison with monoliths
- Benefits over monoliths
- Challenges of microservices