Understanding Queueing Architecture

Stephen Haunts

DEVELOPER, LEADER, AUTHOR AND TRAINER

@stephenhaunts   www.stephenhaunts.com

Overview

Message queueing architecture

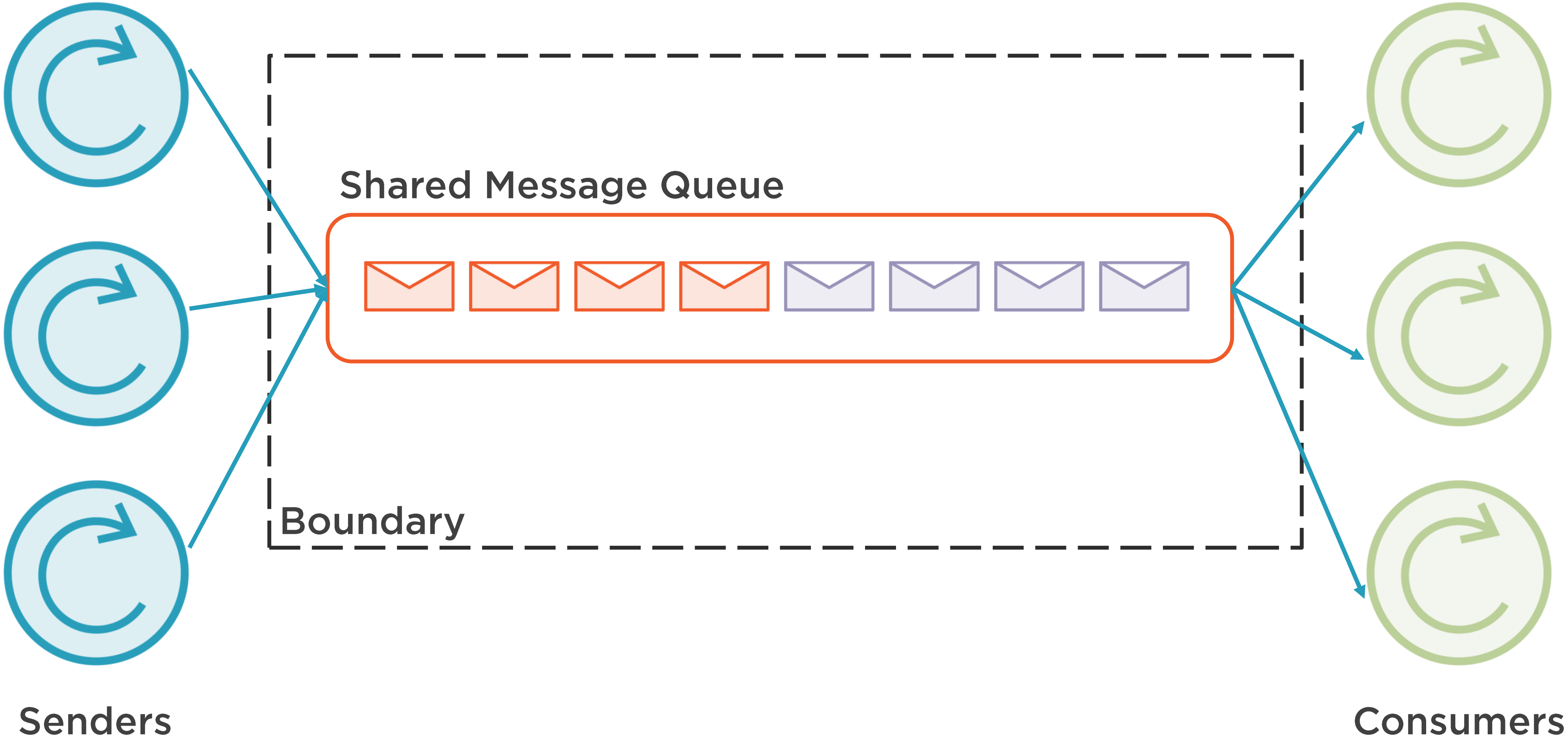
Microservices overview

Service autonomy

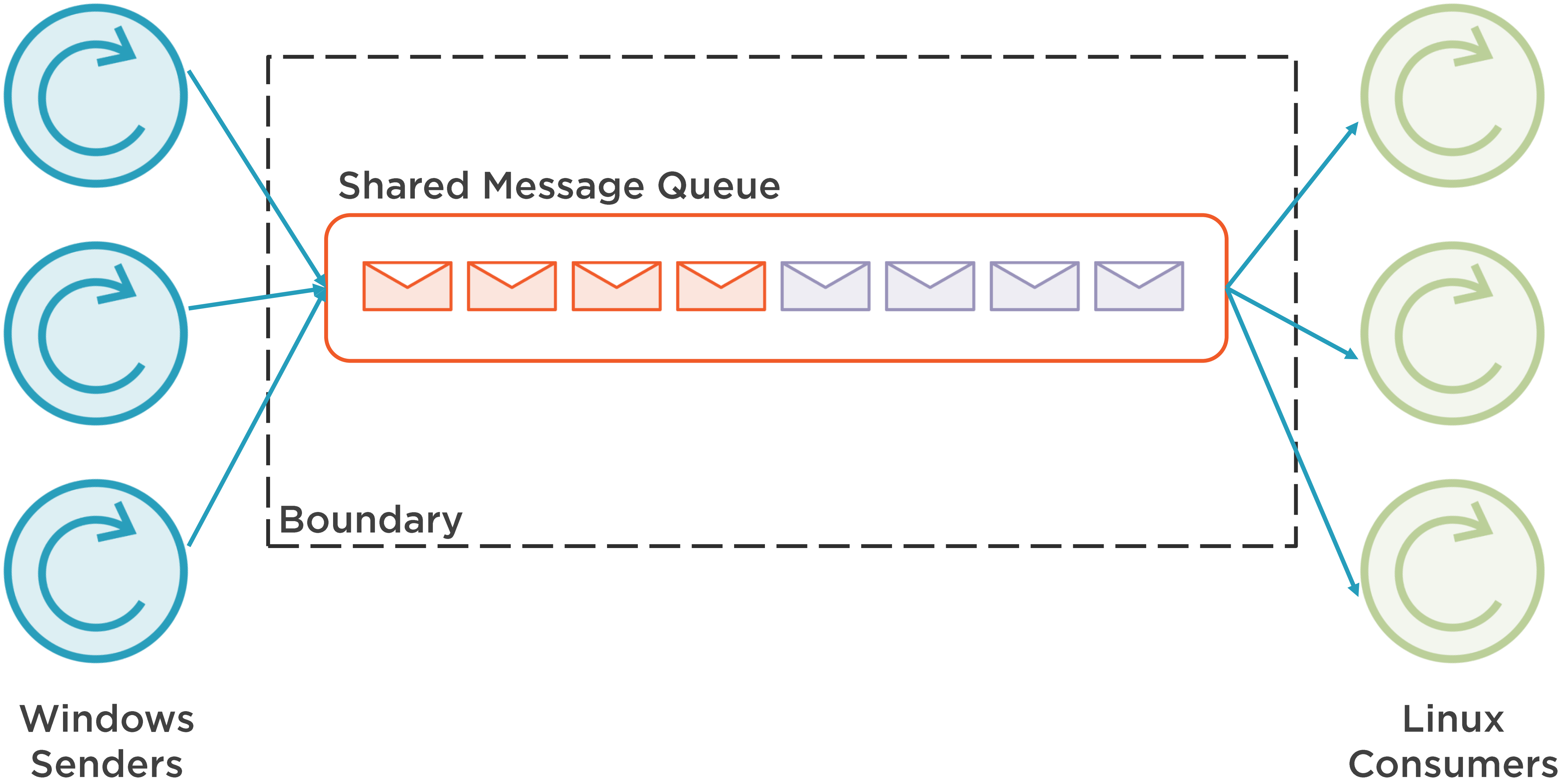
Microservice benefits

Asynchronous Microservices

Message Queueing



Message Queueing



Message Queueing

Commercial

- IBM WebSphere MQ
- Oracle Advanced Queueing
- SonicMQ
- SwiftMQ

Open Source

- Apache ActiveMQ
- RabbitMQ
- ZeroMQ

Message Queueing

Durability

Security Policies

Message Purging

Message Filtering

Delivery Policies

Routing Policies

Message Queueing

Batching Policies

Queuing Criteria

Receipt Notification

Uses for Message Queuing

Decoupling

Redundancy

Scalability

Resiliency

Uses for Message Queuing

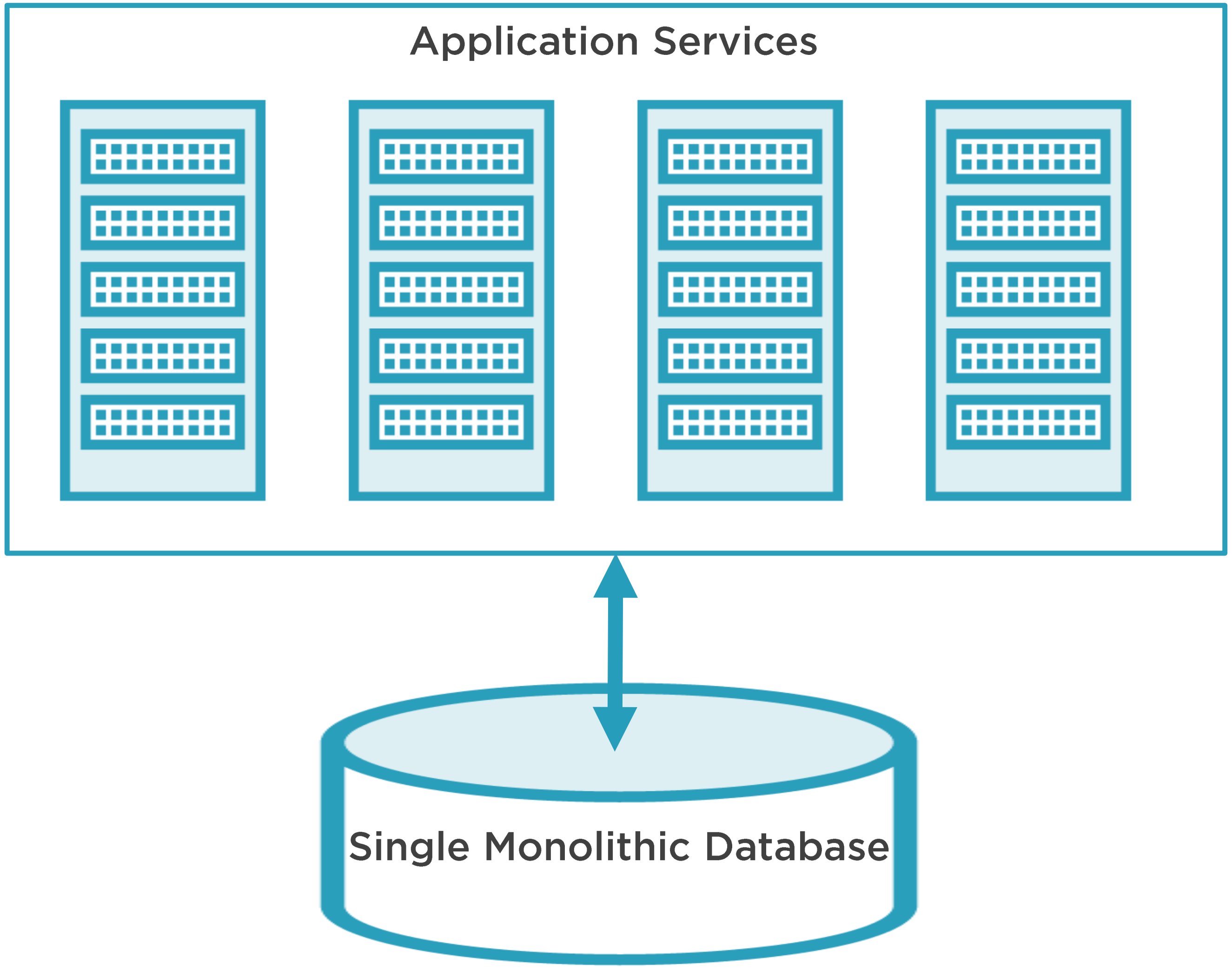
Delivery Guarantees

Ordering Guarantees

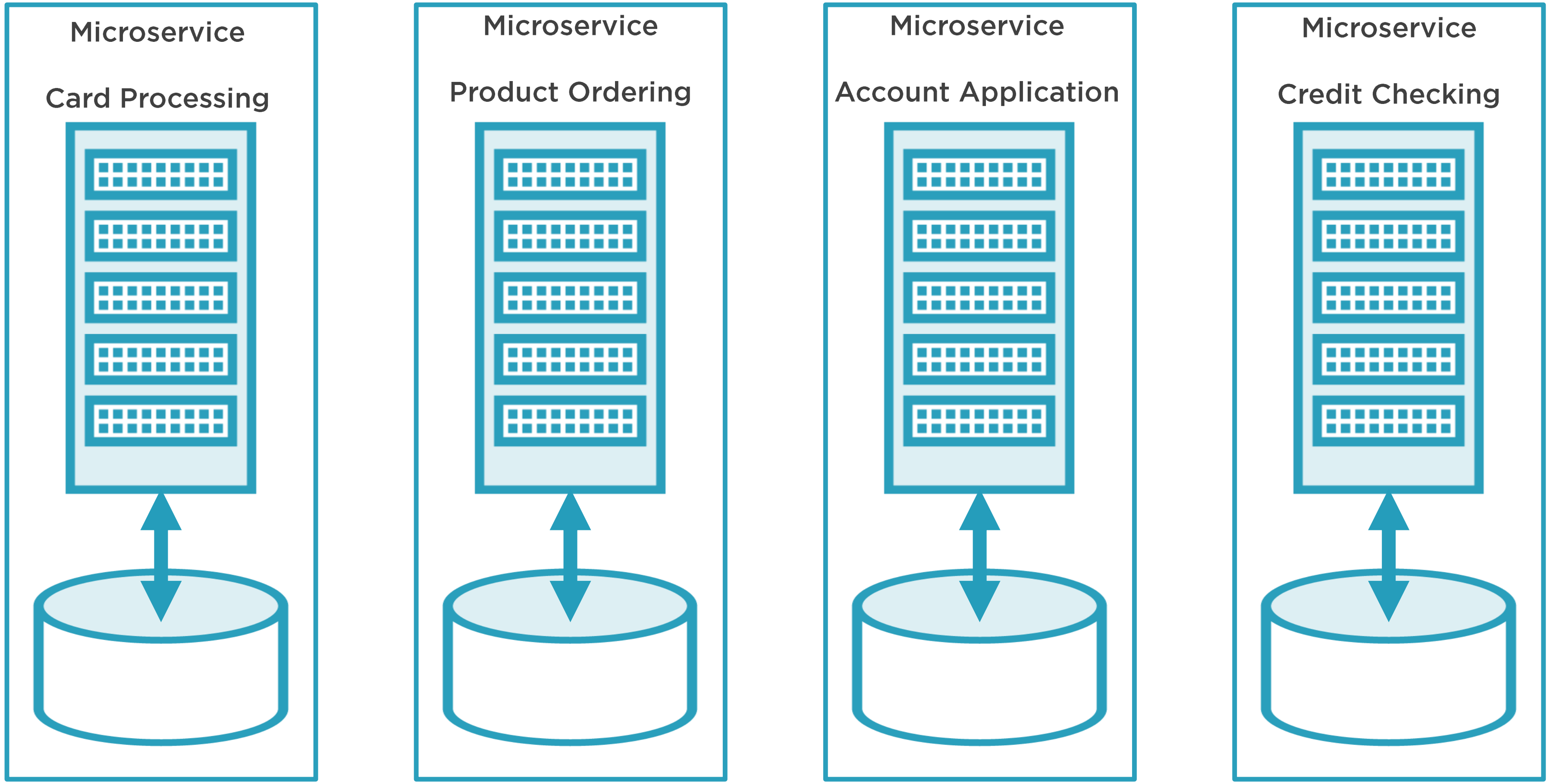
Buffering

Asynchronous Communication

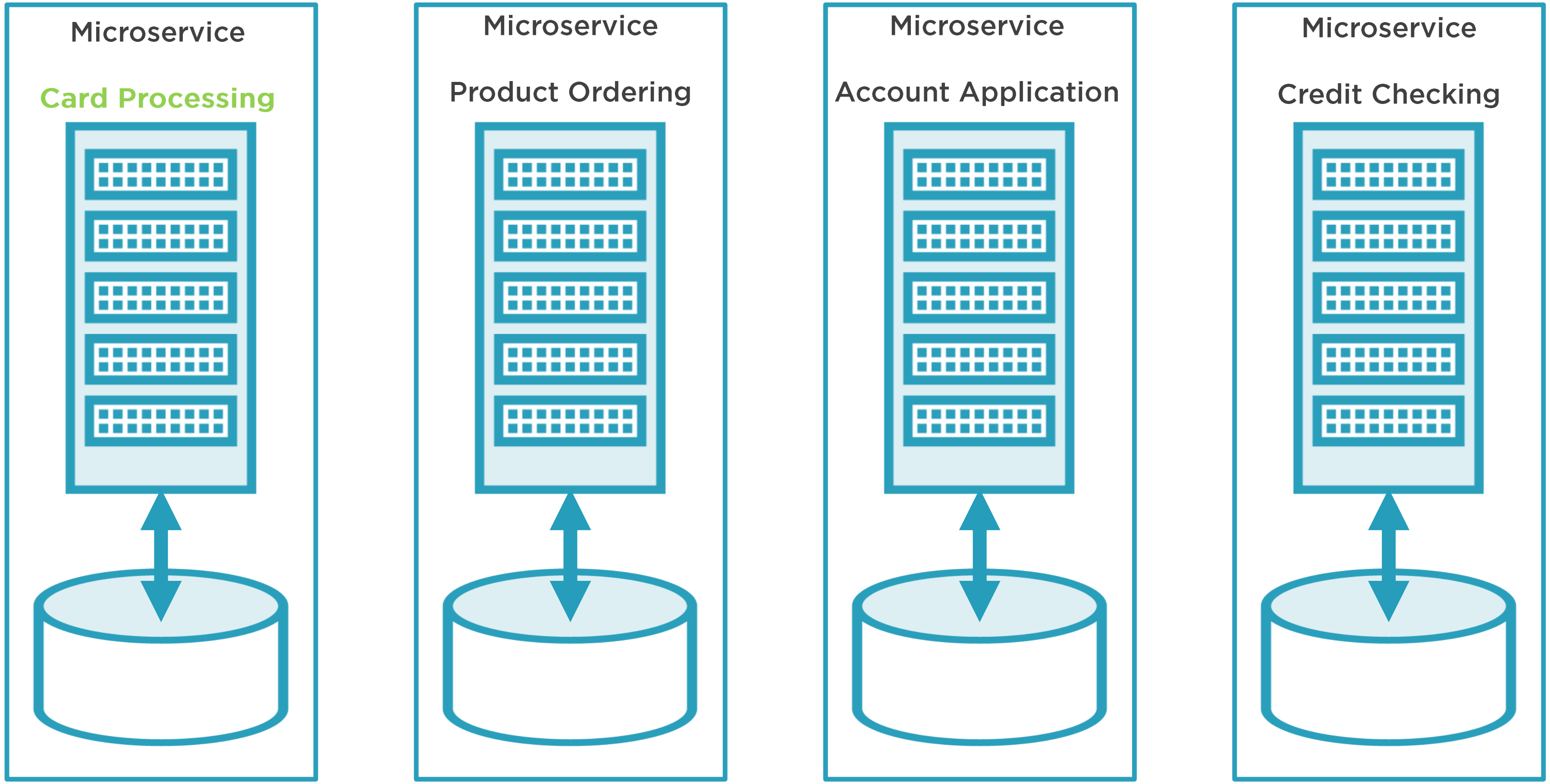
Microservice Architecture



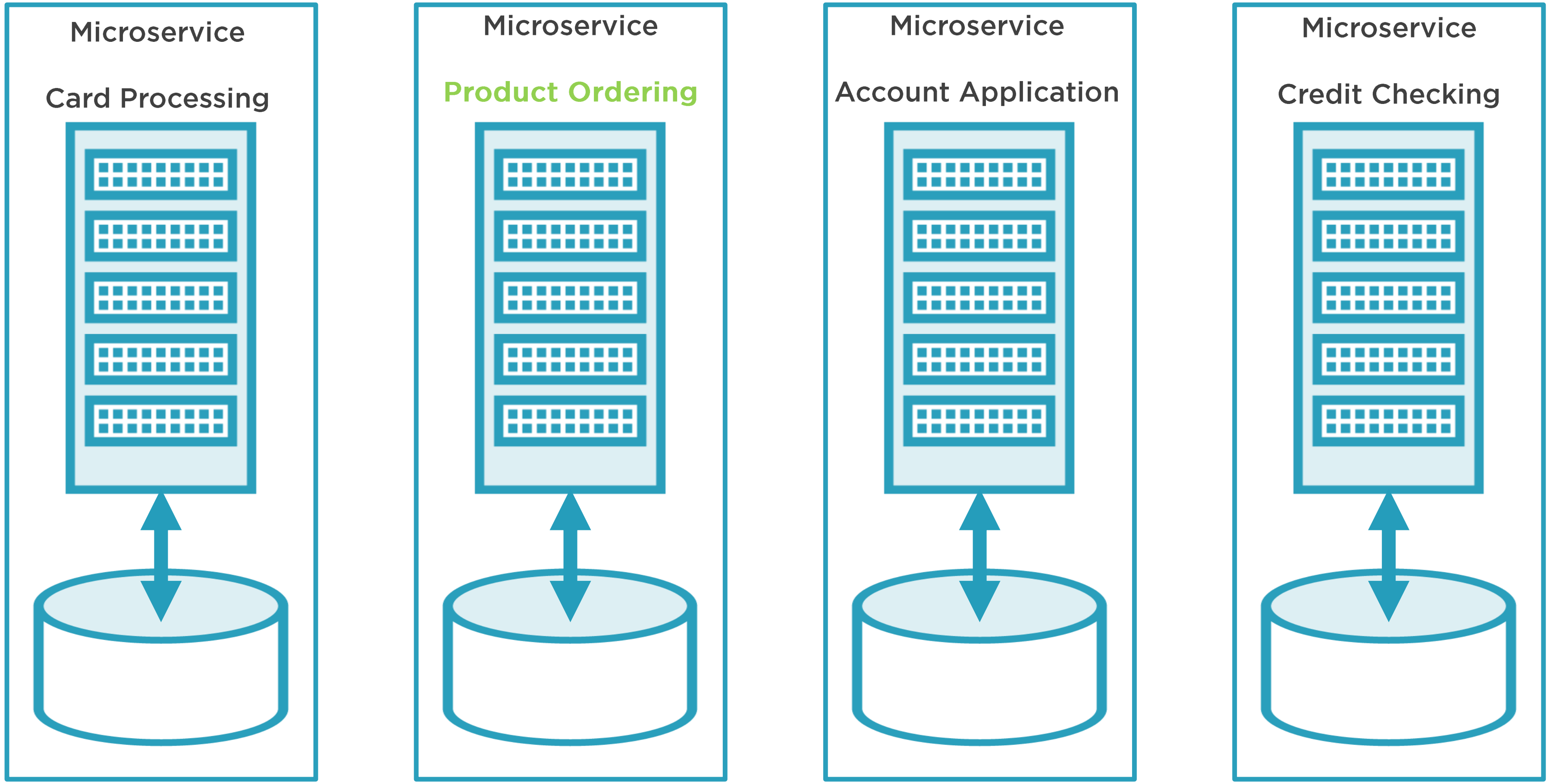
Microservice Architecture



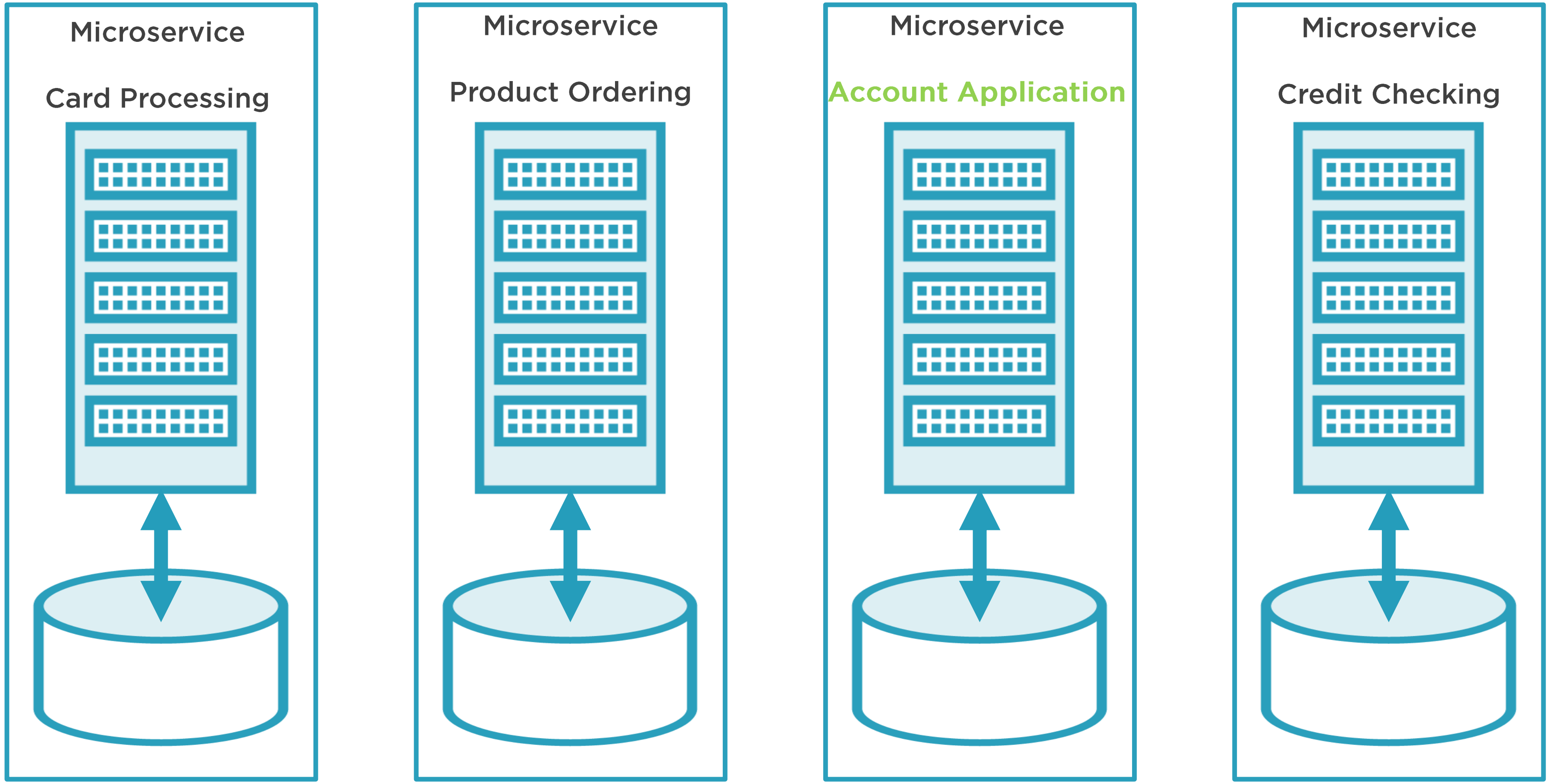
Microservice Architecture



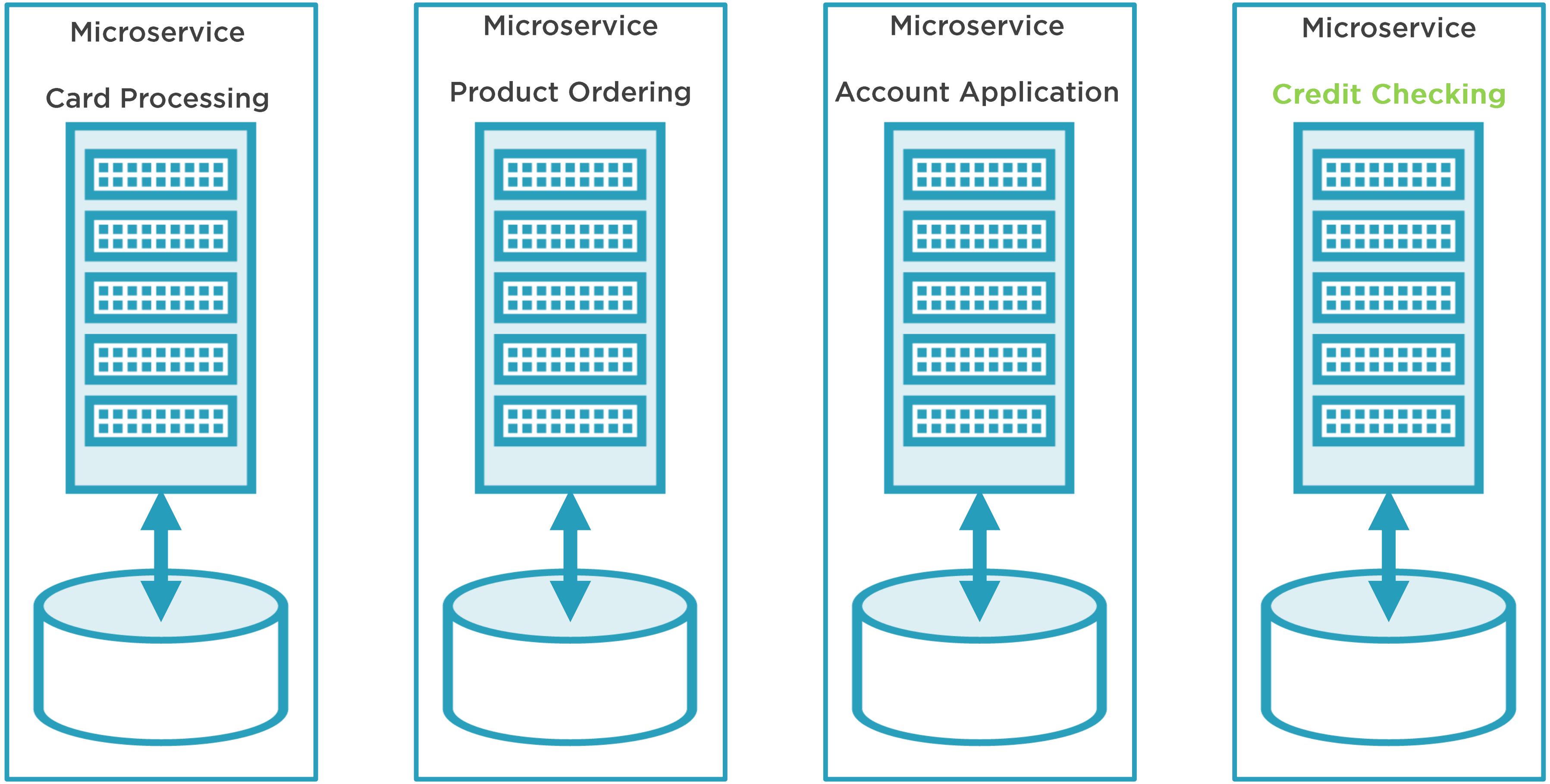
Microservice Architecture



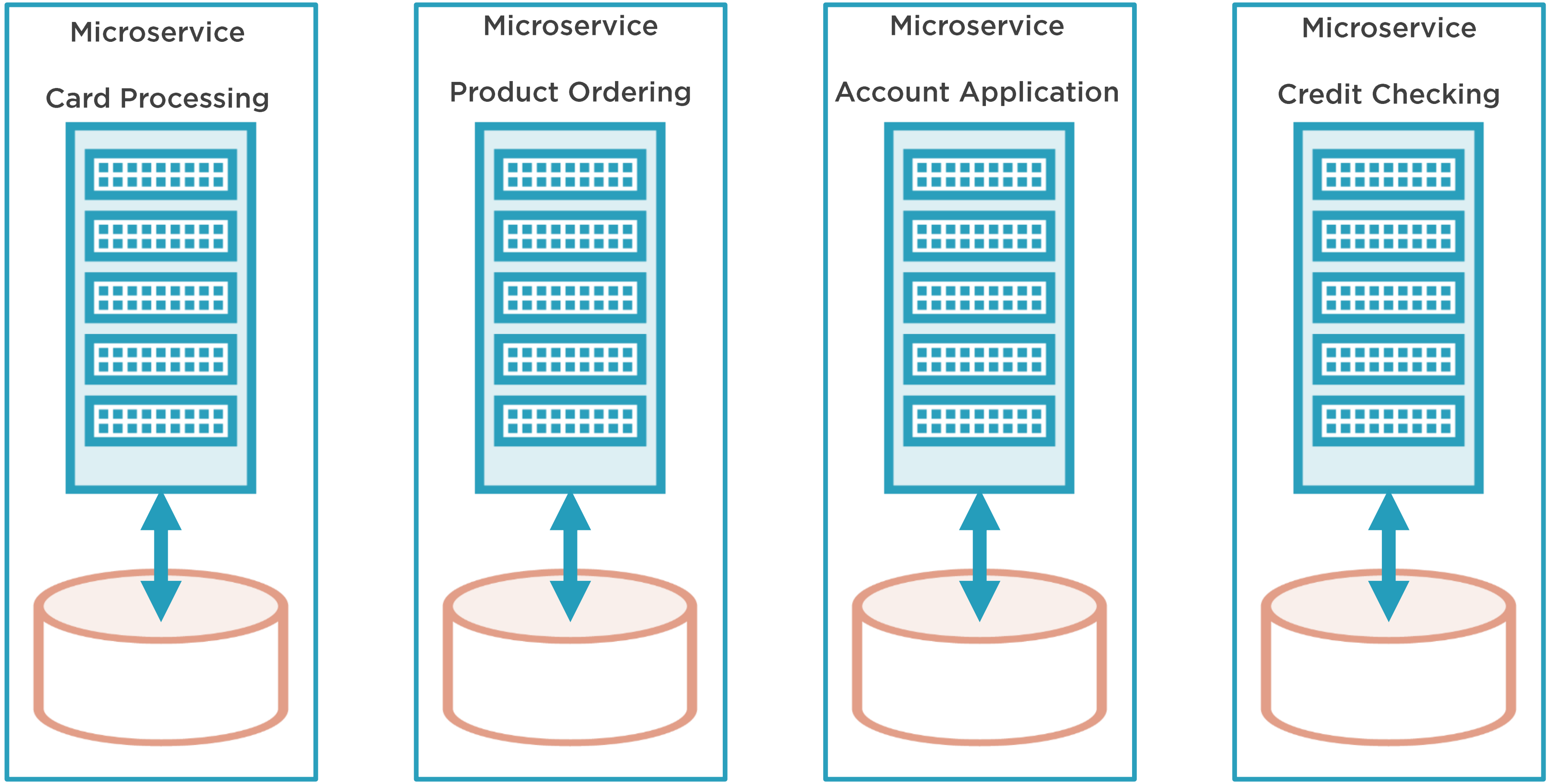
Microservice Architecture



Microservice Architecture



Microservice Architecture



Microservice Architecture

Single Responsibility Principle

Do one thing and do it well

Service Autonomy

Identical Server Deployments

Each Service Has Its Own Server

Platform as a Service (PaaS)

Containers

Microservice Benefits

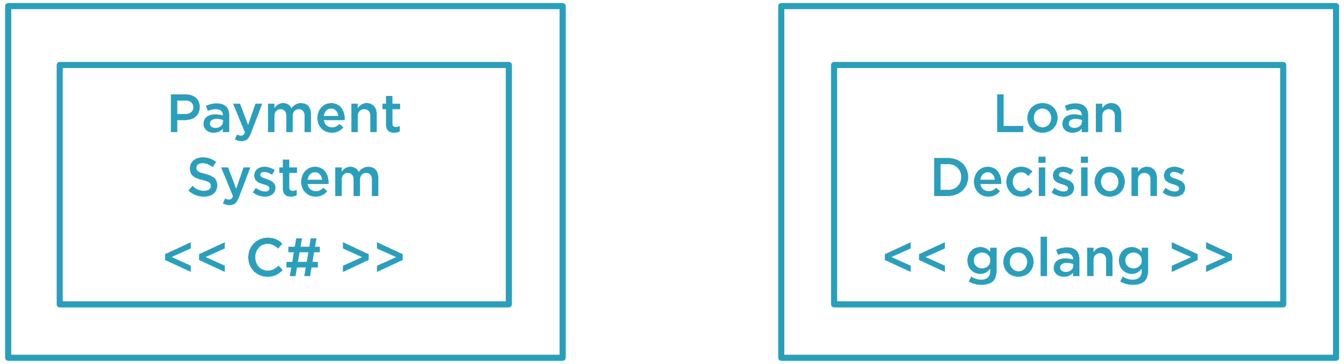
Technology Diversity

System Resilience

System Scaling

Ease of Deployment

Technology Diversity



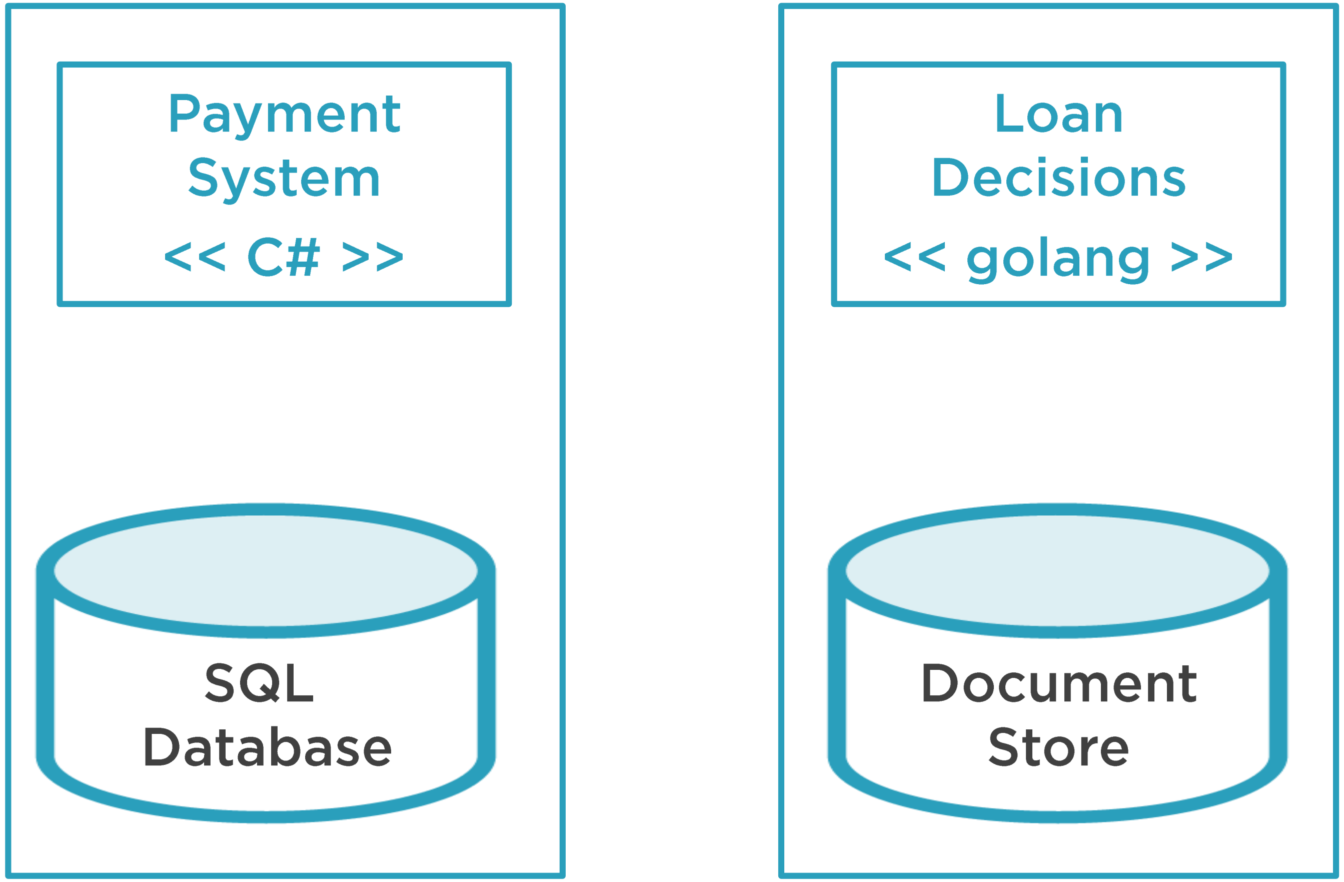
Payment System

<< C# >>

Loan Decisions

<< golang >>

Technology Diversity



System Resilience

Microservices can handle total failure of services and degrade functionality accordingly

System Resilience

The power or ability to return to the original form or position

The capacity to recover quickly from difficulties

The ability of a system to cope with change

System Resilience

The power or ability to return to the original form or position

The capacity to recover quickly from difficulties

The ability of a system to cope with change

System Resilience

The power or ability to return to the original form or position

The capacity to recover quickly from difficulties

The ability of a system to cope with change

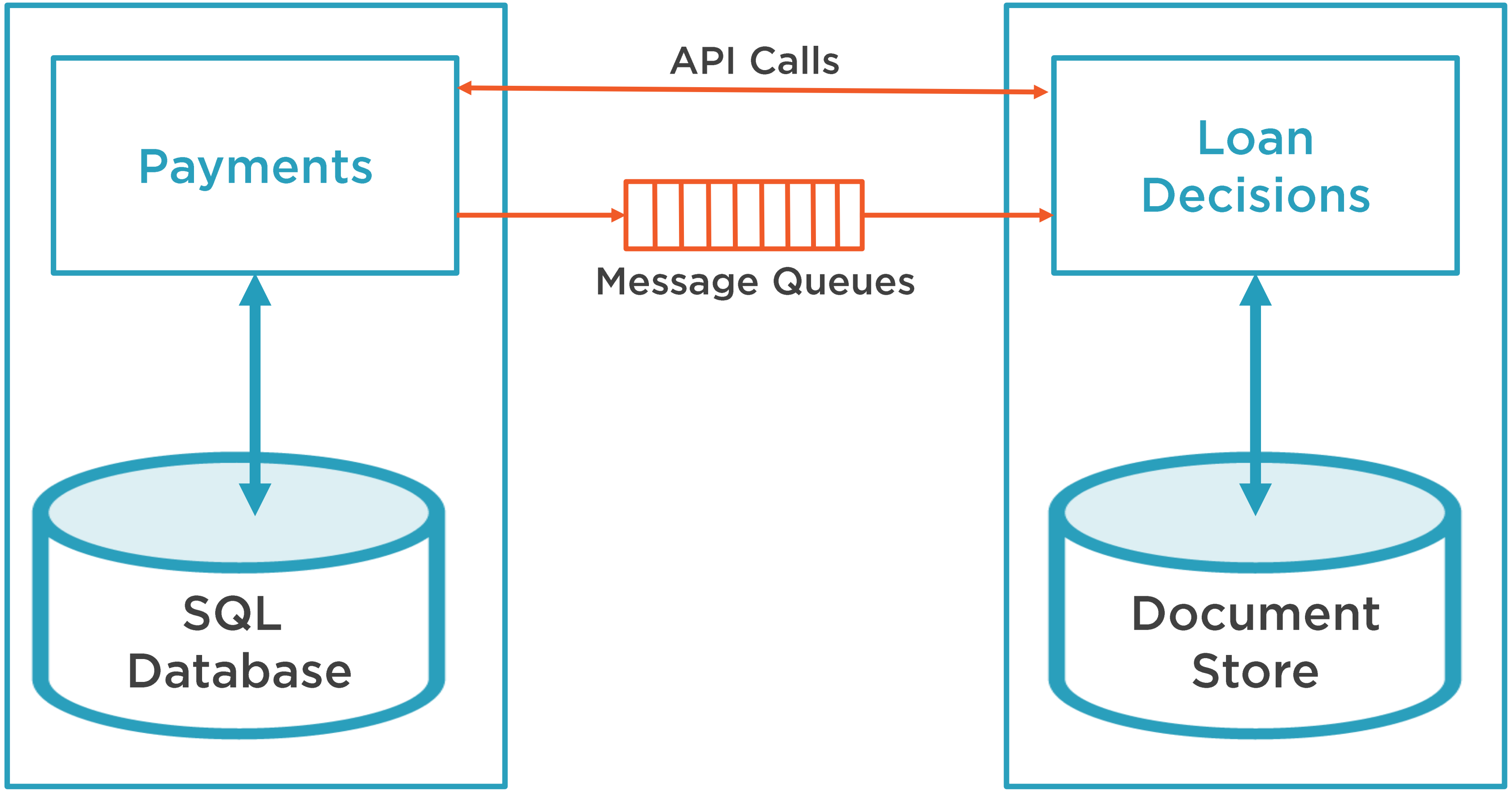
System Resilience

The power or ability to return to the original form or position

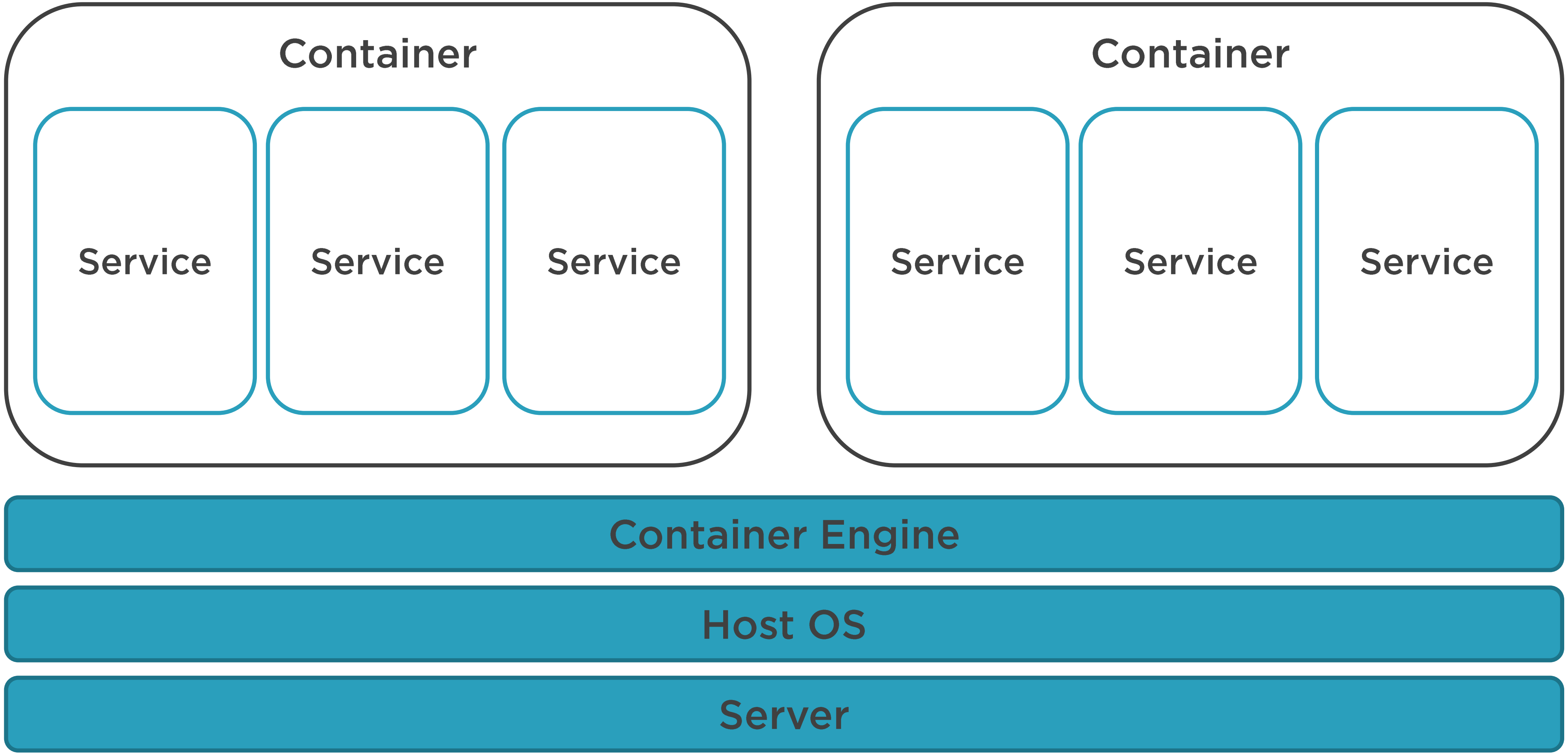
The capacity to recover quickly from difficulties

The ability of a system to cope with change

System Resilience



System Resilience



Container

Service

Service

Service

Container

Service

Service

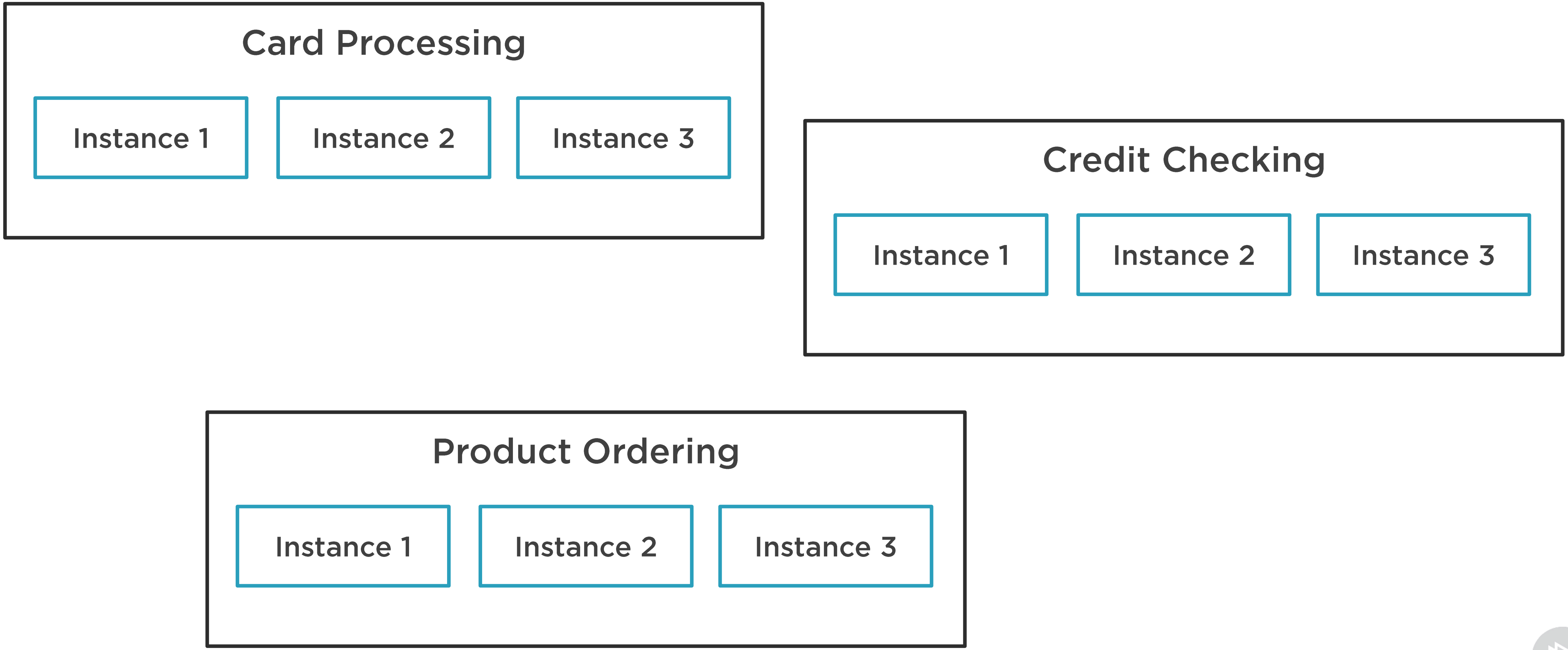
Service

Container Engine

Host OS

Server

System Scaling



Card Processing

Instance 1 Instance 2 Instance 3

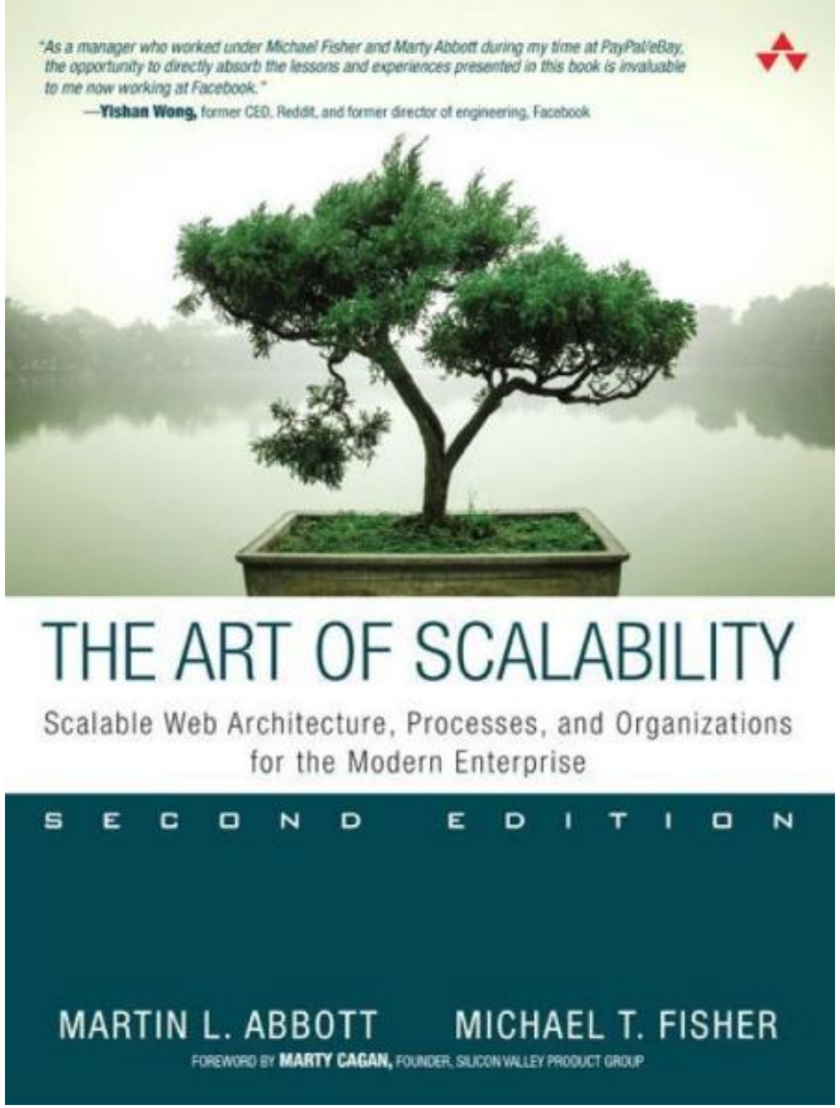
Credit Checking

Instance 1 Instance 2 Instance 3

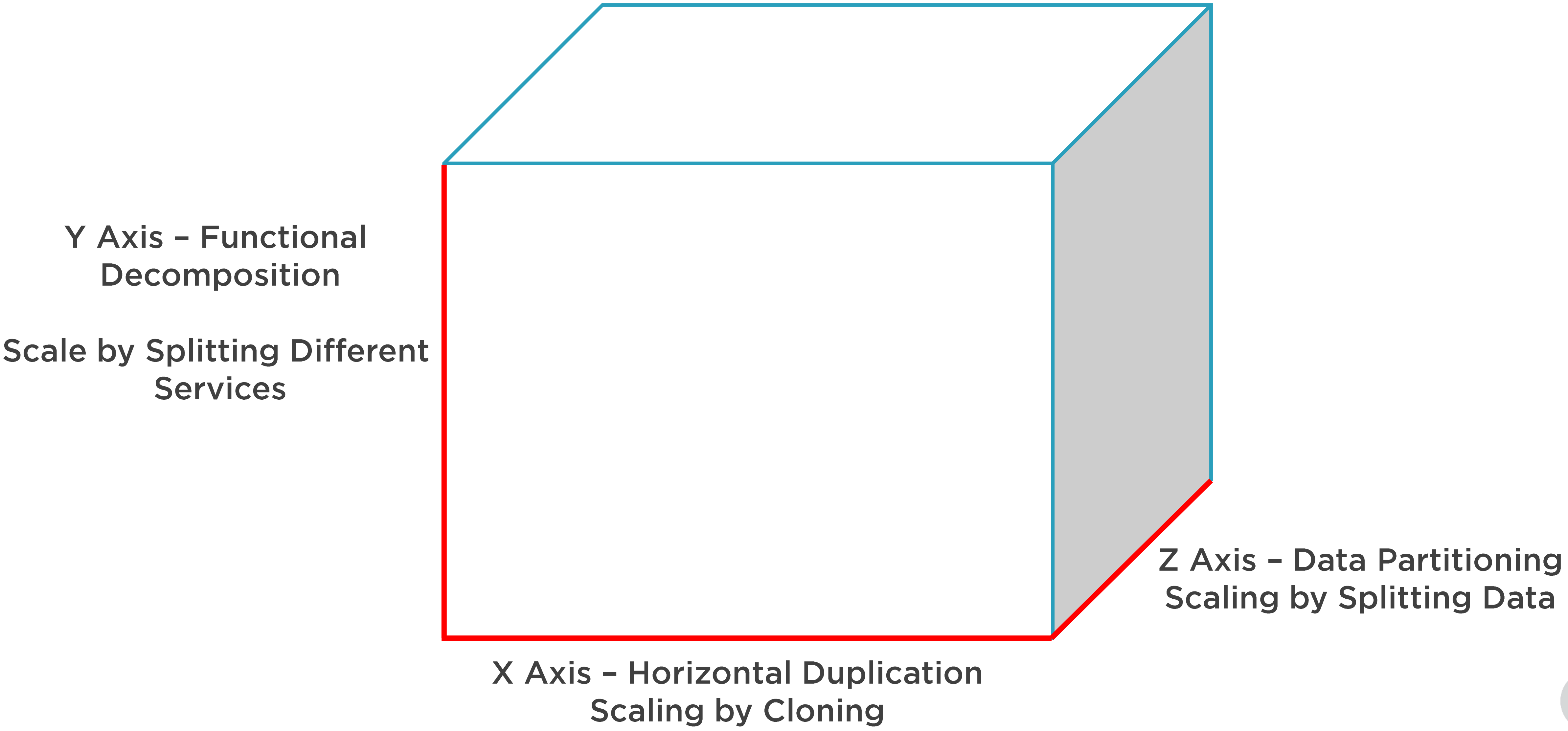
Product Ordering

Instance 1 Instance 2 Instance 3

System Scaling


"As a manager who worked under Michael Fisher and Marty Abbott during my time at PayPal/eBay, the opportunity to directly absorb the lessons and experiences presented in this book is invaluable to me now working at Facebook."
—Yishan Wong, former CEO, Reddit, and former director of engineering, Facebook
THE ART OF SCALABILITY
Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise
SECOND EDITION
MARTIN L. ABBOTT
MICHAEL T. FISHER
FOREWORD BY MARTY CAGAN, FOUNDER, SILICON VALLEY PRODUCT GROUP

System Scaling



Y Axis – Functional Decomposition

Scale by Splitting Different Services

Z Axis – Data Partitioning
Scaling by Splitting Data

X Axis – Horizontal Duplication
Scaling by Cloning

Ease of Deployment

Monolithic Applications

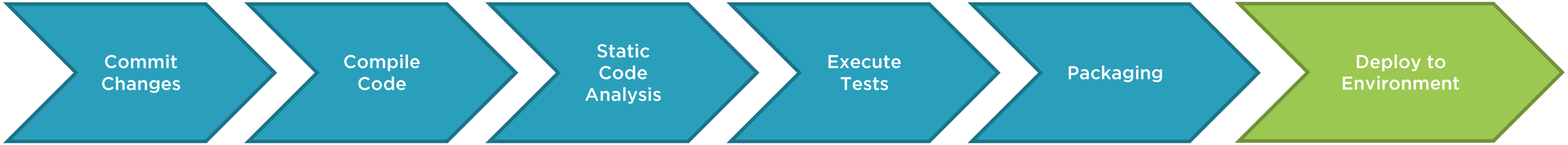
- Typically deployed all together
- Deployed less frequently
- Higher risk of deployment failure

Ease of Deployment

Microservice Applications

- Services deployed independently
- Deployed more frequently
- Lower deployment risk
- Deliver value sooner to the customer

Ease of Deployment



Asynchronous Microservices

Synchronous

Call made to the server that blocks until its ready to reply

Asynchronous

The caller doesn’t wait for the operation to complete before returning

Asynchronous Microservices

Request / Response

A client initiates a request and waits for the response

Event Based

A client initiates a request asking for something to happen