

Scaling Applications with Microservices, MassTransit, and RabbitMQ

UNDERSTANDING MESSAGING ARCHITECTURE



Roland Guijt

INDEPENDENT SOFTWARE DEVELOPER AND TRAINER

@rolandguijt www.rmgsolutions.nl



About This Course



5 modules



Prerequisites:
.NET/C#



Fire On Wheels
demos



Module Overview



Distributed Systems

Microservices

Message Fundamentals

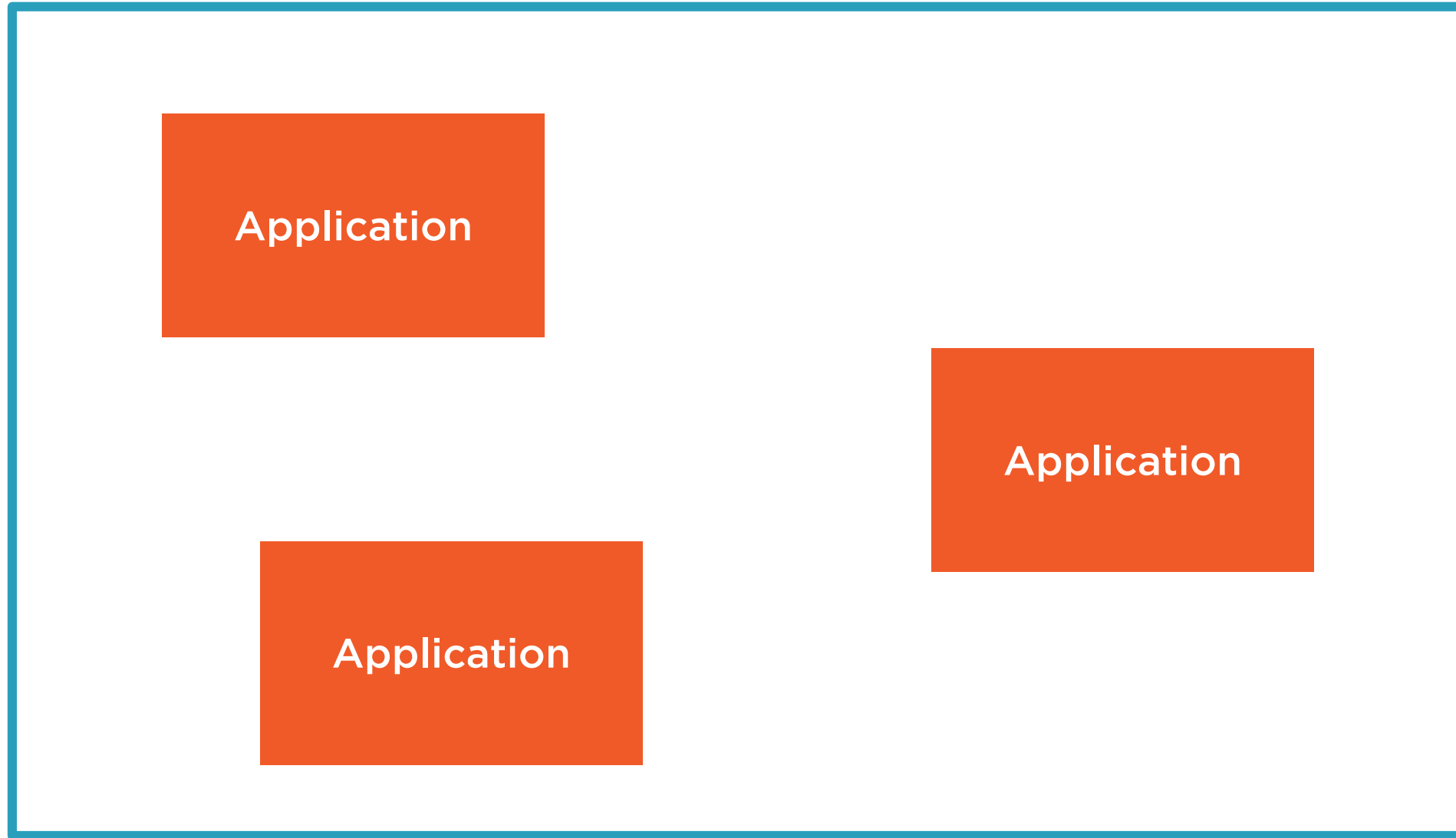
Scenarios



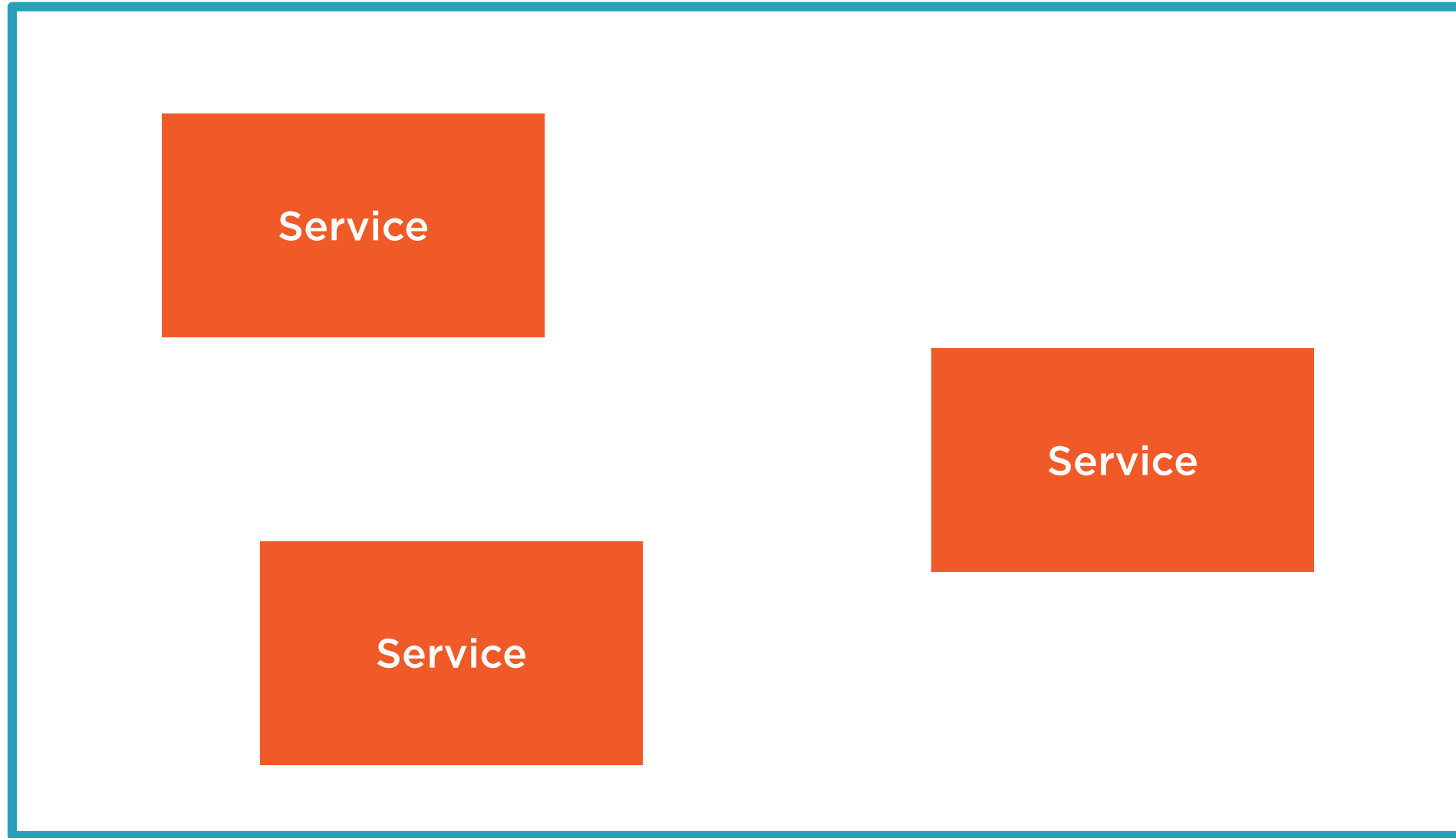
This course is
about building
distributed systems.



A Distributed System



A Distributed System



Why a Distributed System?

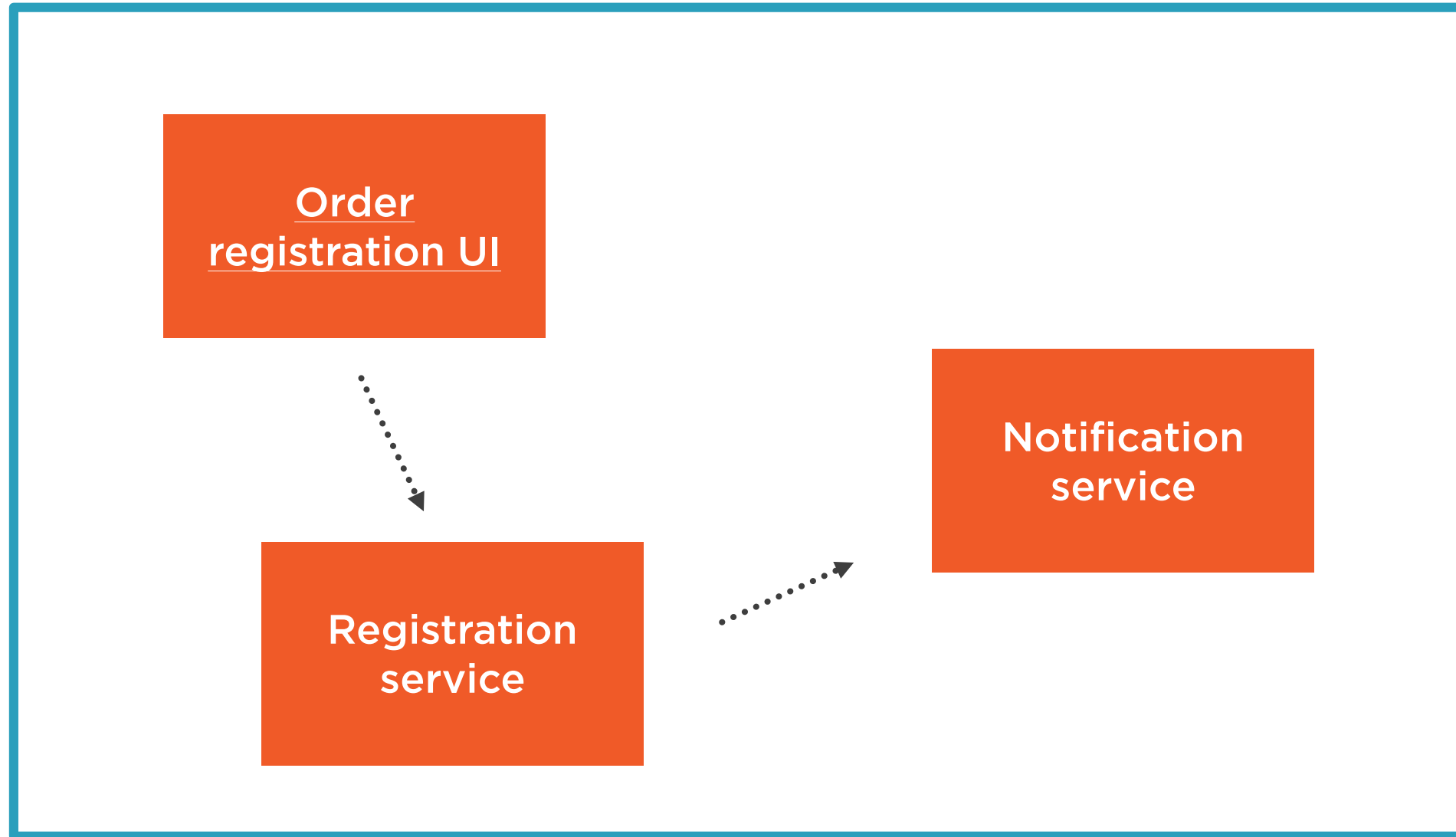
One big application is not an option

Separation of concerns

Integration by third-parties



A Distributed System



Communication Between Services

RPC

Remote Procedure Call

SOAP

Simple Object Access Protocol

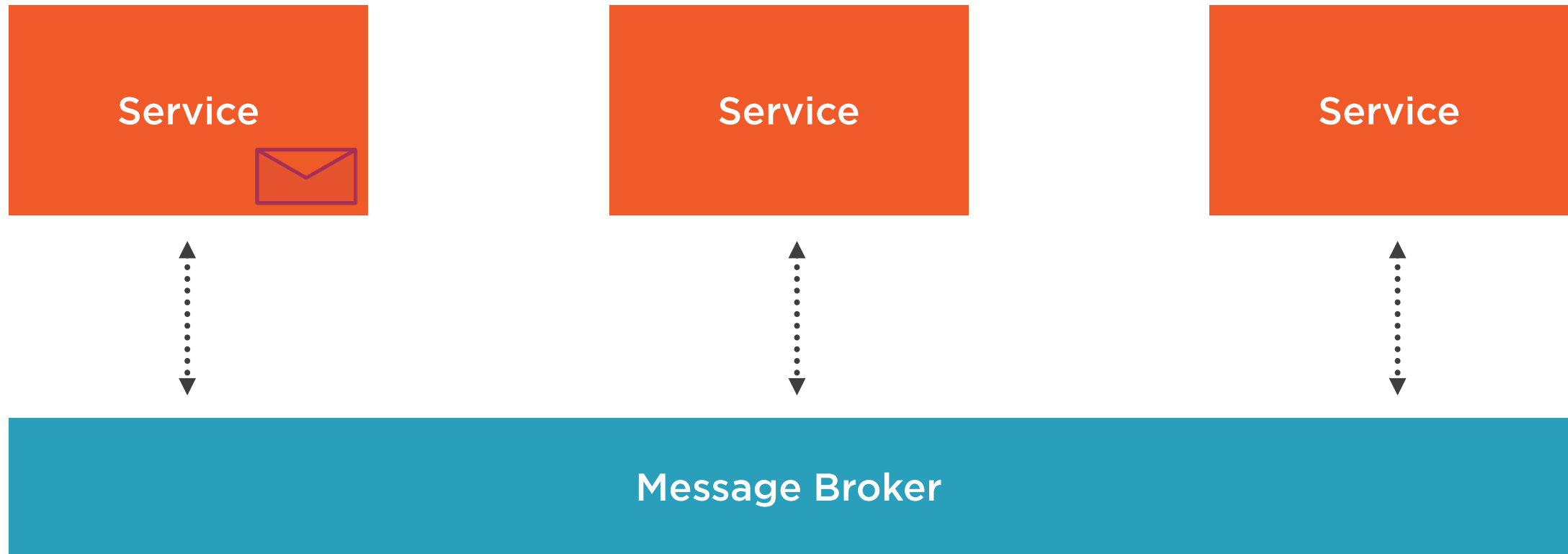
REST

Representational State Transfer

Messaging

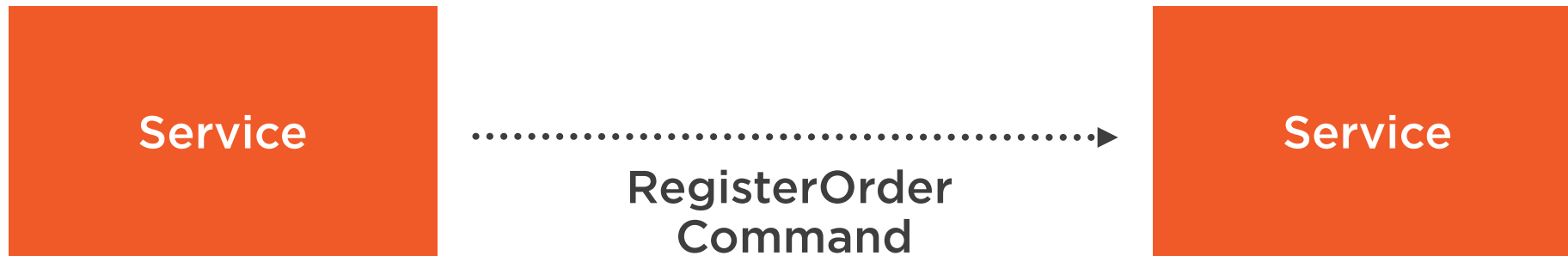


Messaging





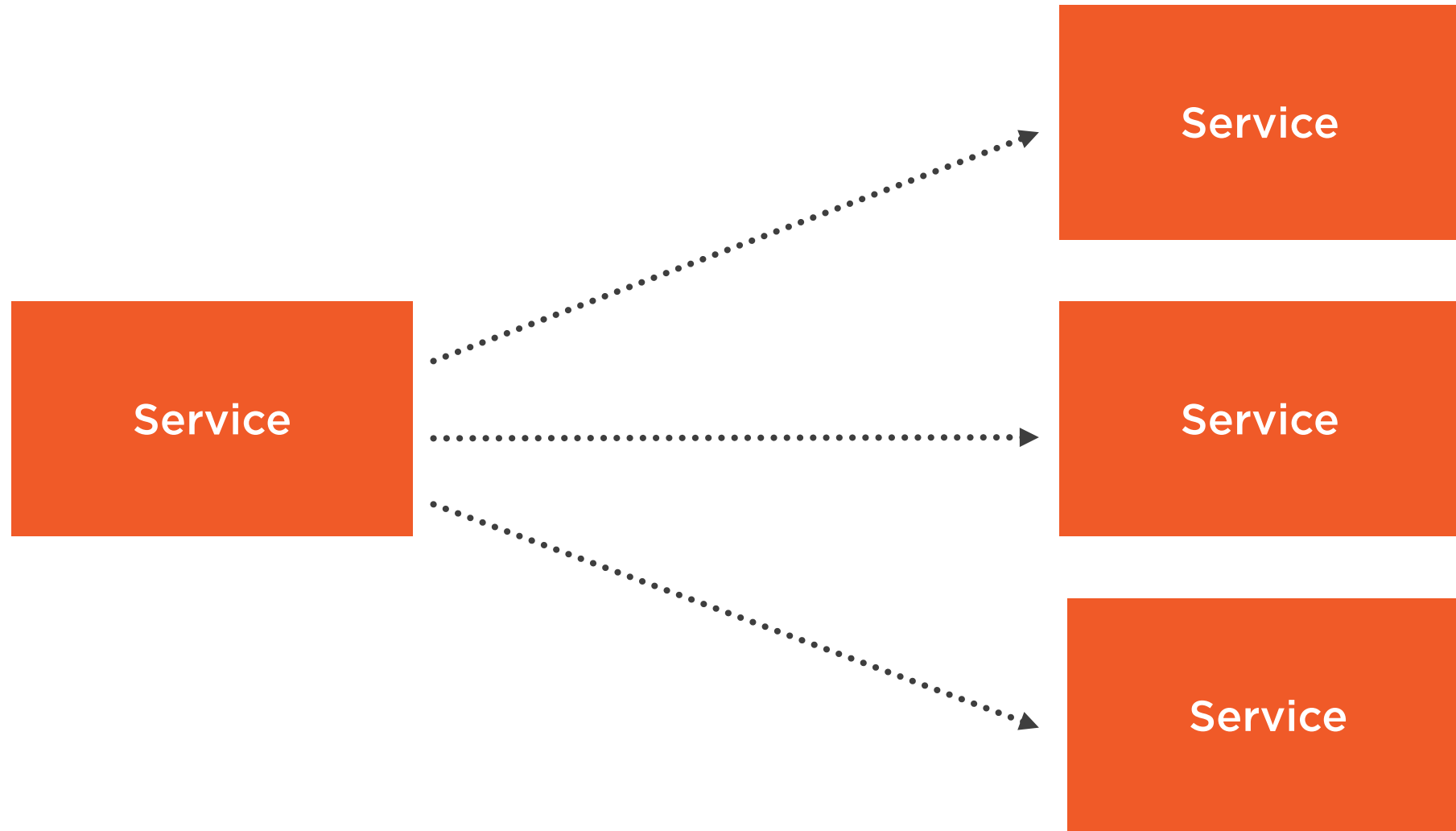
Messaging Patterns: Point-To-Point



Messaging Patterns: RPC



Messaging Patterns: Event Broadcasting



Message Brokers and MSMQ

Message broker

Centralized

Multi platform

Standardized

Scale with clustering

Supported by MassTransit

MSMQ

Decentralized

Windows only

No standard

Scales automatically

Supported by NServiceBus



Microservices

A software architectural style

Composed of small, highly decoupled independent processes

Communicate with each other using language-agnostic APIs

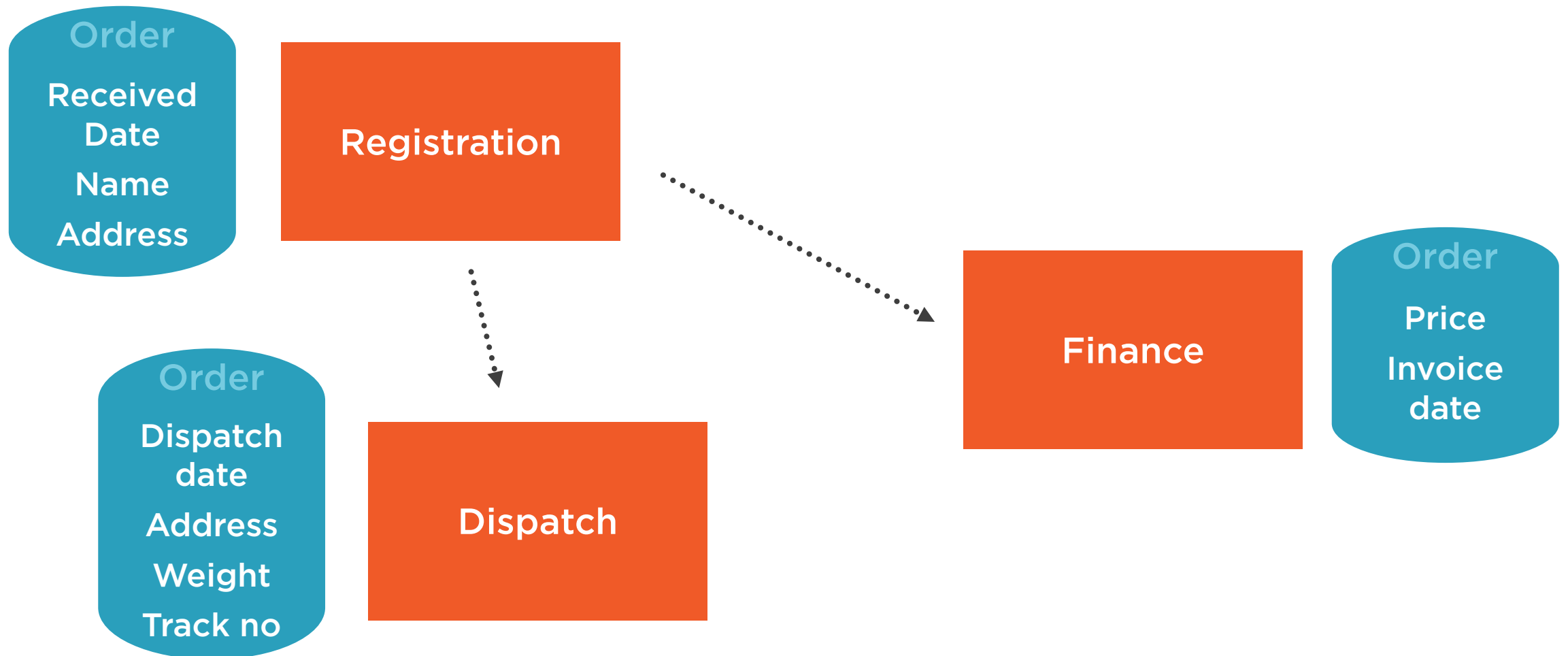
Each process focusses on doing one small task



Each individual
microservice
is **autonomous**



Microservices and Data



Scaling Applications with Microservices and NServiceBus

Module 1



FIRE ON WHEELS

Package delivery

Scaling

Continuous deployment

Multiple teams



FIRE ON WHEELS

Reliable



Messages as Interfaces

Shared assembly

IRegisterOrder

IOrderRegistered

Order registration service

RegisterOrder: IRegisterOrder

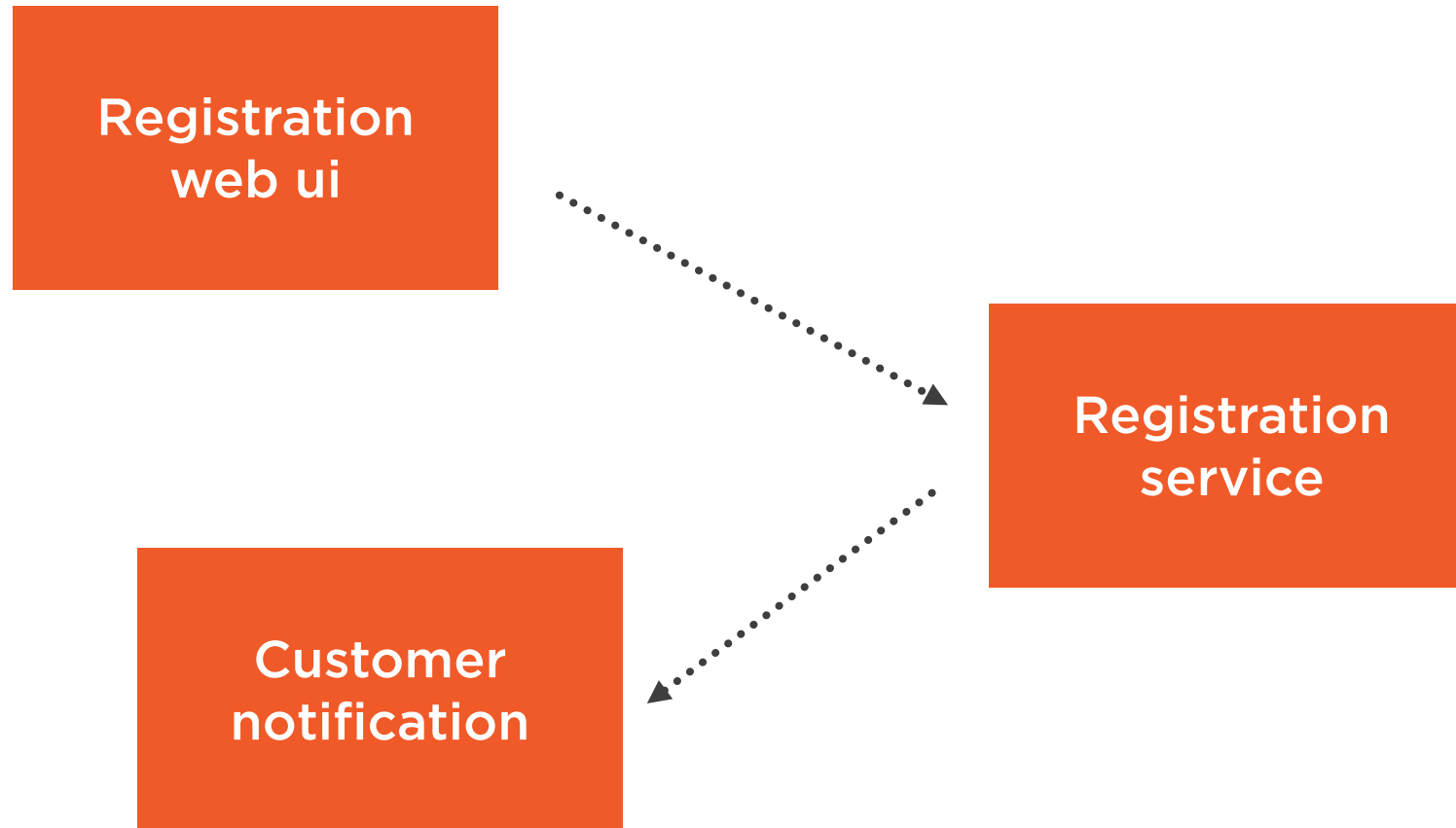
OrderRegistered: IOrderRegistered

Notification service

OrderRegistered: IOrderRegistered



Fire On Wheels POC



Messaging Scenarios

**Integrating legacy
applications**

**Powering
dashboard apps**

**Simplifying
complex
monoliths**



Summary



Distributed systems

Message broker

Microservices

Setting up

Messaging scenarios

