

Microservices Communication in ASP.NET Core

INTRODUCING MICROSERVICE COMMUNICATION
IN ASP.NET CORE



Gill Cleeren

CTO XPIRIT BELGIUM

@gillcleeren www.snowball.be



Overview



The Path to Microservices at GloboTicket

What you need to know before starting

Looking at the finished application

Understanding the different services



The Path to Microservices at GloboTicket





GLOBOTICKET

A Globomantics Company

Introducing GloboTicket





“Hello, I’m Bob T. Hickett,
founder of GloboTicket.”





“Our customers have been affected by our system not being able to handle the high demand.”

“We want to bring them a new site and mobile app which is always available.”

Bob T. Hickett
CEO GloboTicket



“Our old system is a typical monolith, it’s built as a single application that needs to be deployed as a whole.”

“We’re also in a business where traffic peaks occur and we sometimes have issues with scaling.”

“Some teams would also like to use different technology stacks.”

Bethany
Lead Architect at GloboTicket





“But... I’ve heard about this thing called microservices. I hear this will solve all of our problems.

Can we use this our new system?”

Bob T. Hickett
(Still CEO, but now a bit more technical...)



“A microservice exposes a single capability through an API.

A microservices architecture consists out of many services, each covering a specific functionality.”

Bethany
Lead Architect at GloboTicket



Why Microservices at GloboTicket?

**Different
technology stacks**

Scalable

**Independent and
frequent updates**



Challenges Faced When Building Microservices

Identifying the services

Distributed

Observability

Technical challenge



“Although Bob is not a technical person, microservices may be a good approach for the new system.”

Bethany
Lead Architect at GloboTicket





This course will show you
the full architecture
for the microservices
solution.





Communication between
different services is key for
success.



Different Communication Types

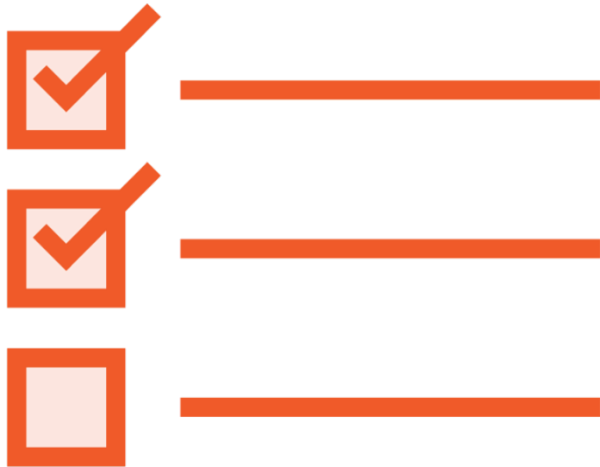
Synchronous

REST, gRPC

Asynchronous

Message-based





Other requirements

- Resiliency
- Different front-end applications



What You Need to Know before Starting



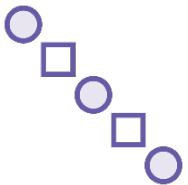
What I Assume You Know



C#

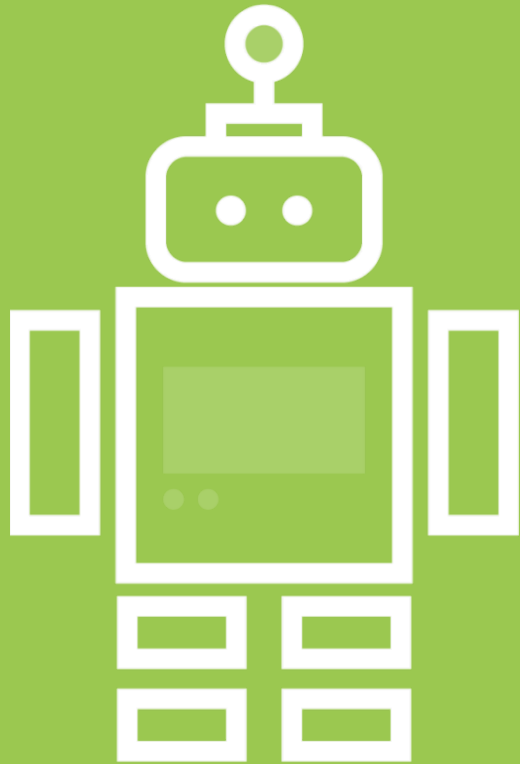


ASP.NET Core & EF Core



Typical patterns



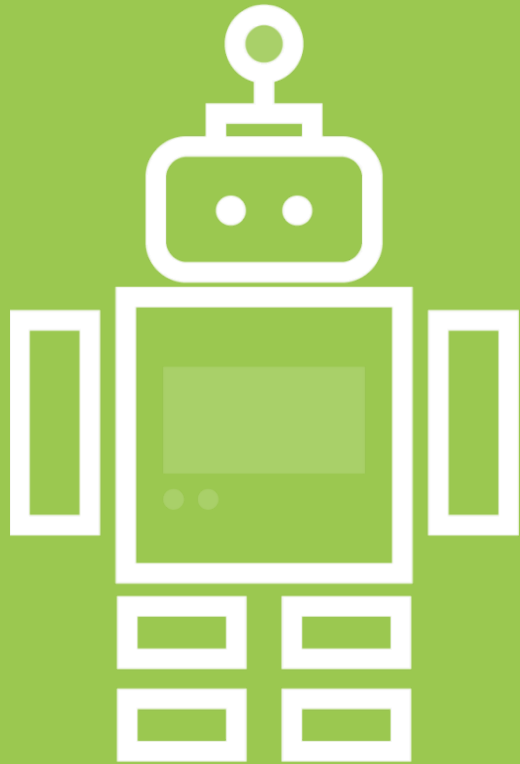


Microservices

Basic idea of microservices

Creating a basic REST services with
ASP.NET Core API





ASP.NET Core Microservices: Getting Started

[https://www.pluralsight.com/library/courses/
getting-started-asp-dot-net-core-microservices](https://www.pluralsight.com/library/courses/getting-started-asp-dot-net-core-microservices)

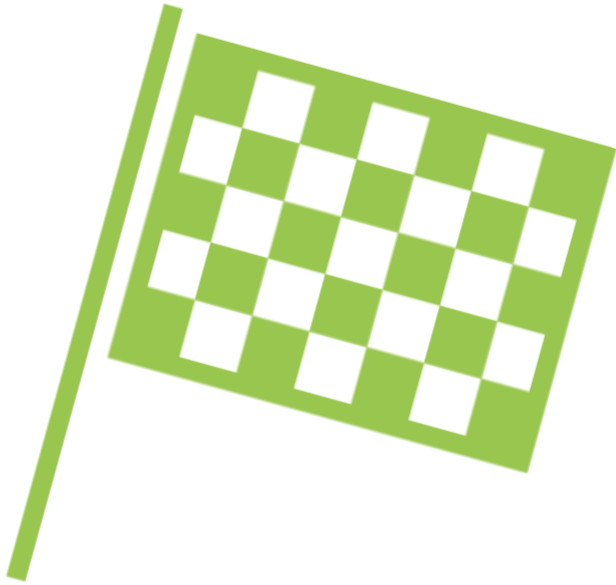


Downloads for This Course

| | | | | | | |
|-------------------|-------------|------------|-----------------------|------------|----------------|-----------------|
| Table of contents | Description | Transcript | Exercise files | Discussion | Learning Check | Related Courses |
|-------------------|-------------|------------|-----------------------|------------|----------------|-----------------|



What You Need to Have Installed



Visual Studio

.NET Core 3.1 SDK

Optional

- Xamarin
- Azure account



Demo



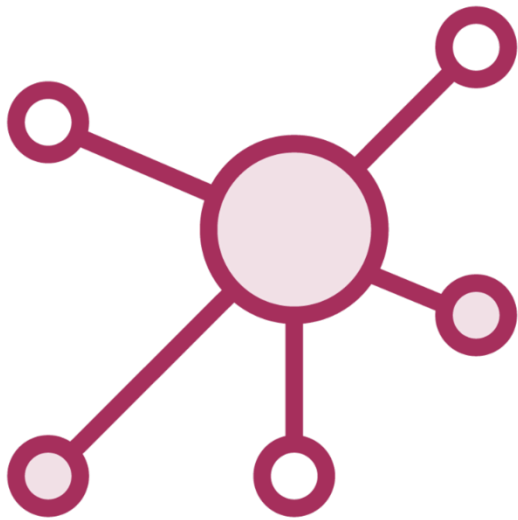
Exploring the web application

Looking at the mobile application



Understanding the Different Services





Identifying the different capabilities

Map to services in the solution



Identifying the Different Services

**Browsing the
event catalog**

**Adding items to
the shopping cart**

Placing order

Handling payment

**Recommending
other events**

**Calculating the
discount**



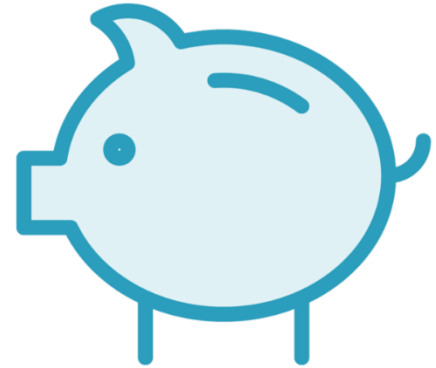
The Different Services



Catalog



Shopping Basket



Discount



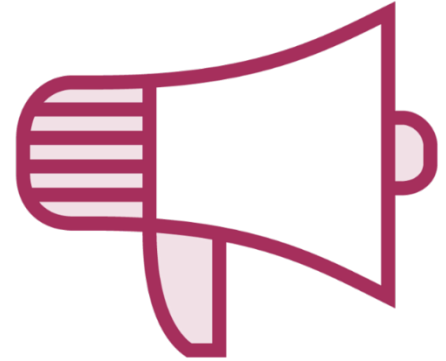
The Different Services



Order

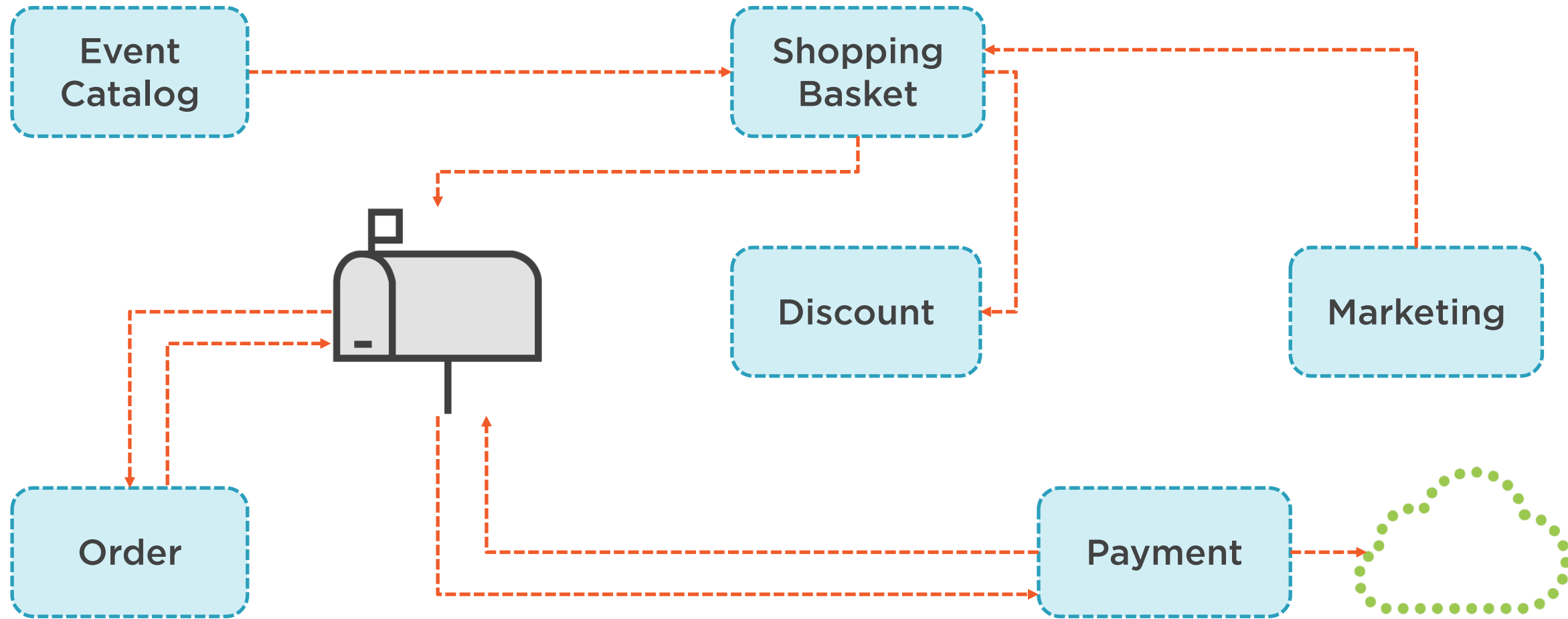


Payment

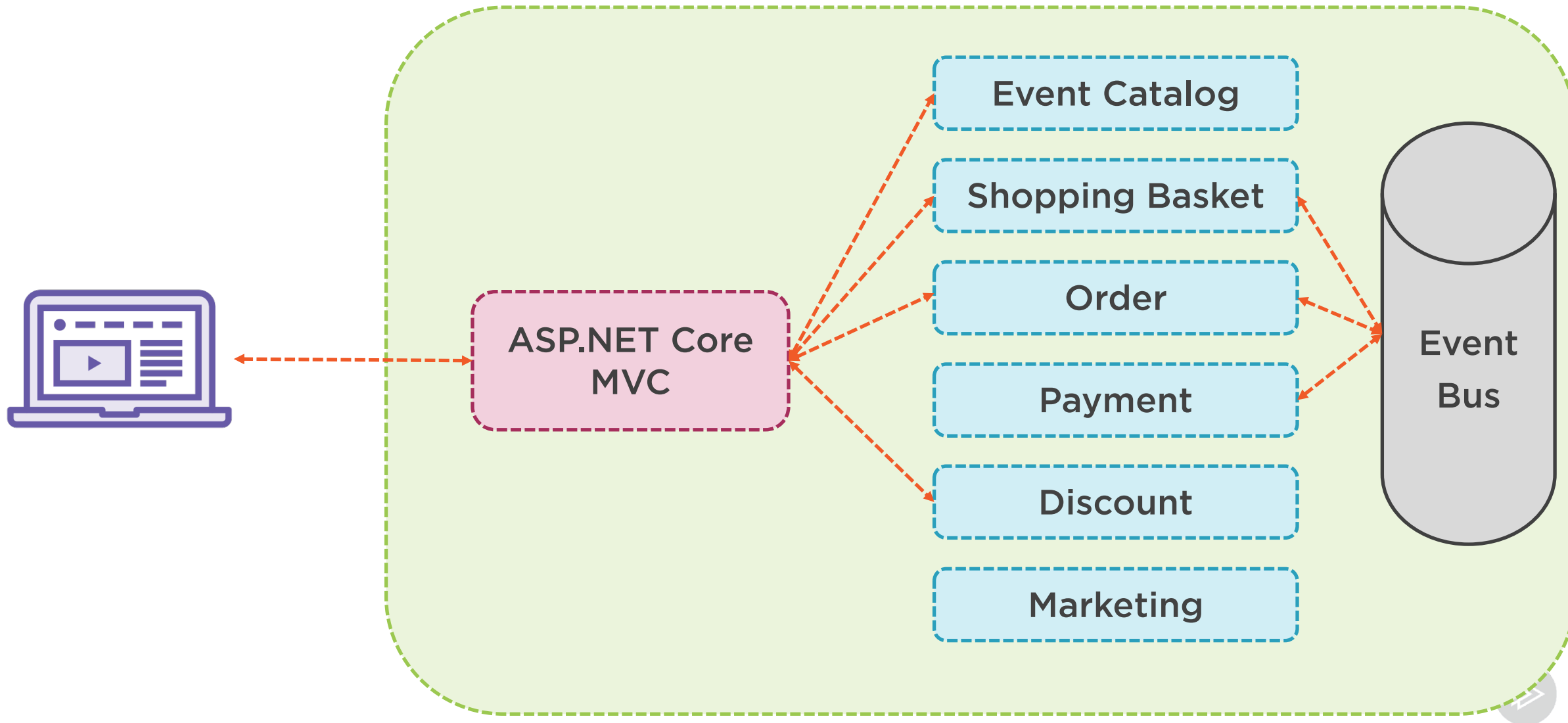


Marketing

The Overall Service Architecture



Exposing the Services to the Outside





Up next:
Synchronous service
communication

