Creating Synchronous Communication between ASP.NET Core Microservices

Gill Cleeren
CTO XPIRIT BELGIUM

@gillcleeren www.snowball.be

Overview

Synchronous microservice communication

Exploring a REST microservice architecture built with .NET Core

Communicating with other services

Working with gRPC

Disadvantages of synchronous communication

Synchronous Microservice Communication

No more method calls

Different applications

Invoking other service through network call

Synchronous Microservice Communication

No more method calls

Simple protocols

Logic inside the services

Types of Communication

Synchronous

Asynchronous

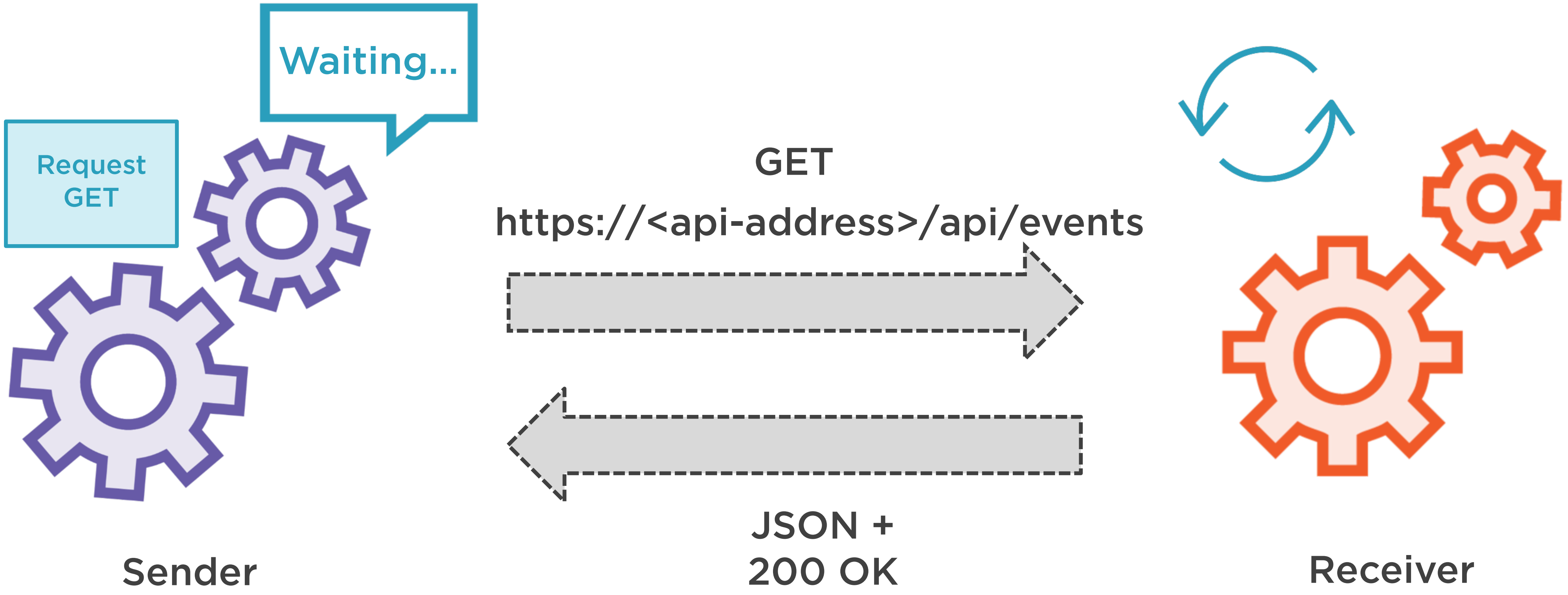
Synchronous communication

- Request and response
- Not related to async code

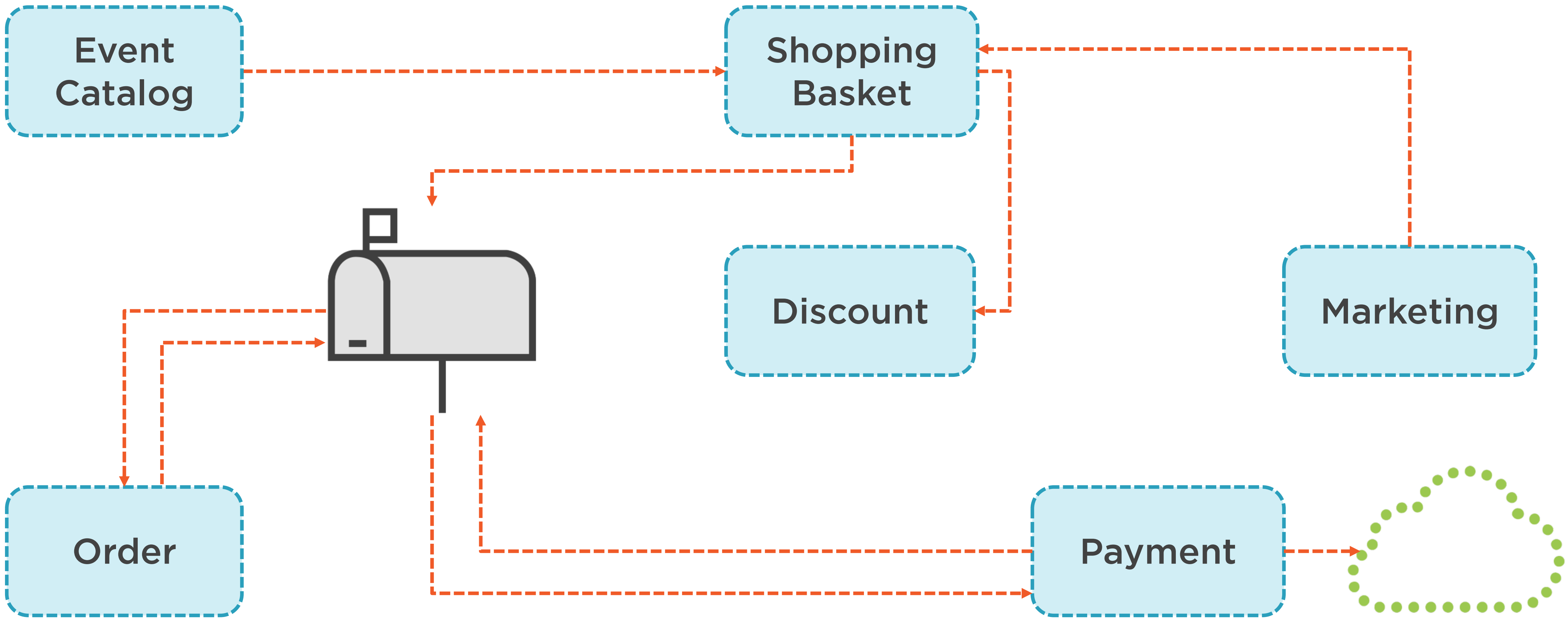
Types Used for Synchronous Communication

- REST
- gRPC
- SignalR
- TCP

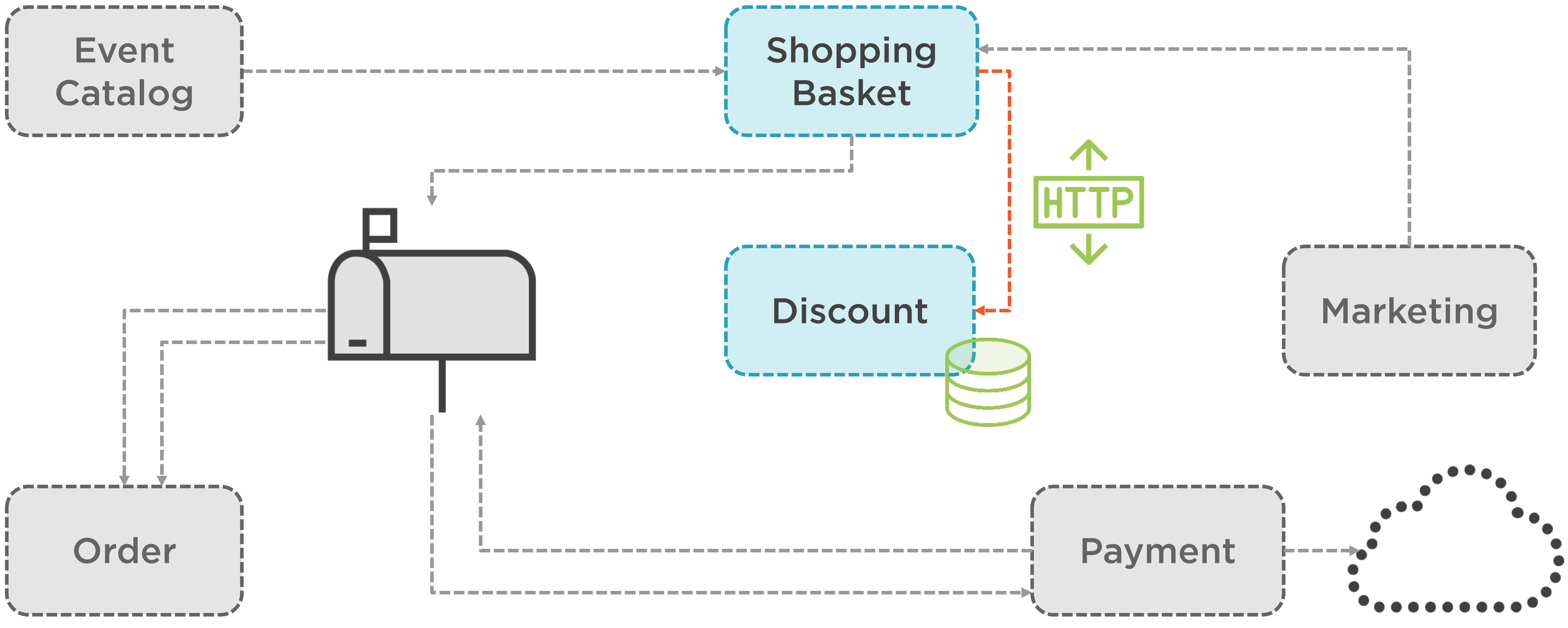
Using REST


Request GET
Waiting...
GET
https://<api-address>/api/events
JSON +
200 OK
Sender
Receiver

Synchronous Communication in GloboTicket



Synchronous Communication in GloboTicket

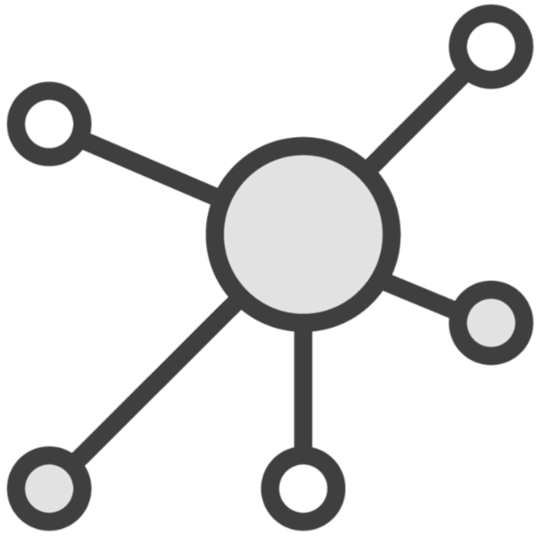


Demo

Exploring the synchronous communication between microservices

Running the application for this module

Exploring a REST Microservice Architecture Built with .NET Core



REST microservice is “just” an API

- ASP.NET Core API

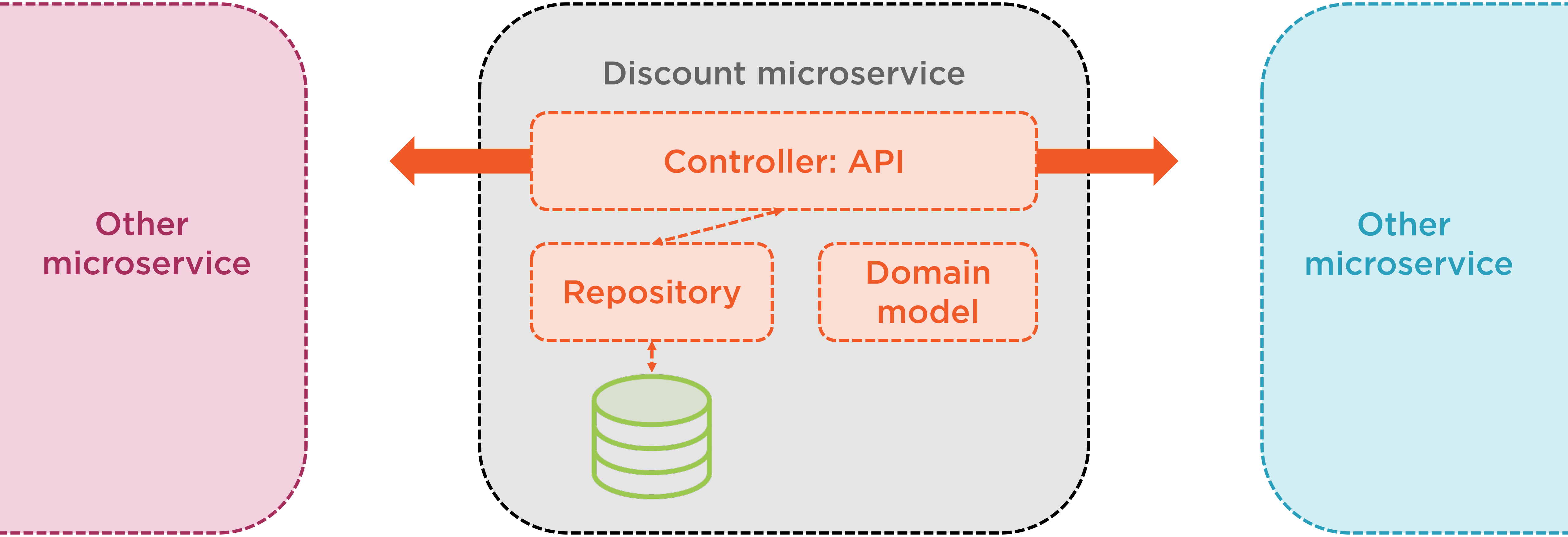
Can be created with every approach

- Internals
- Clean architecture, layered…

Own its data

- Sync of data between services through integration events

ASP.NET Core API



Demo

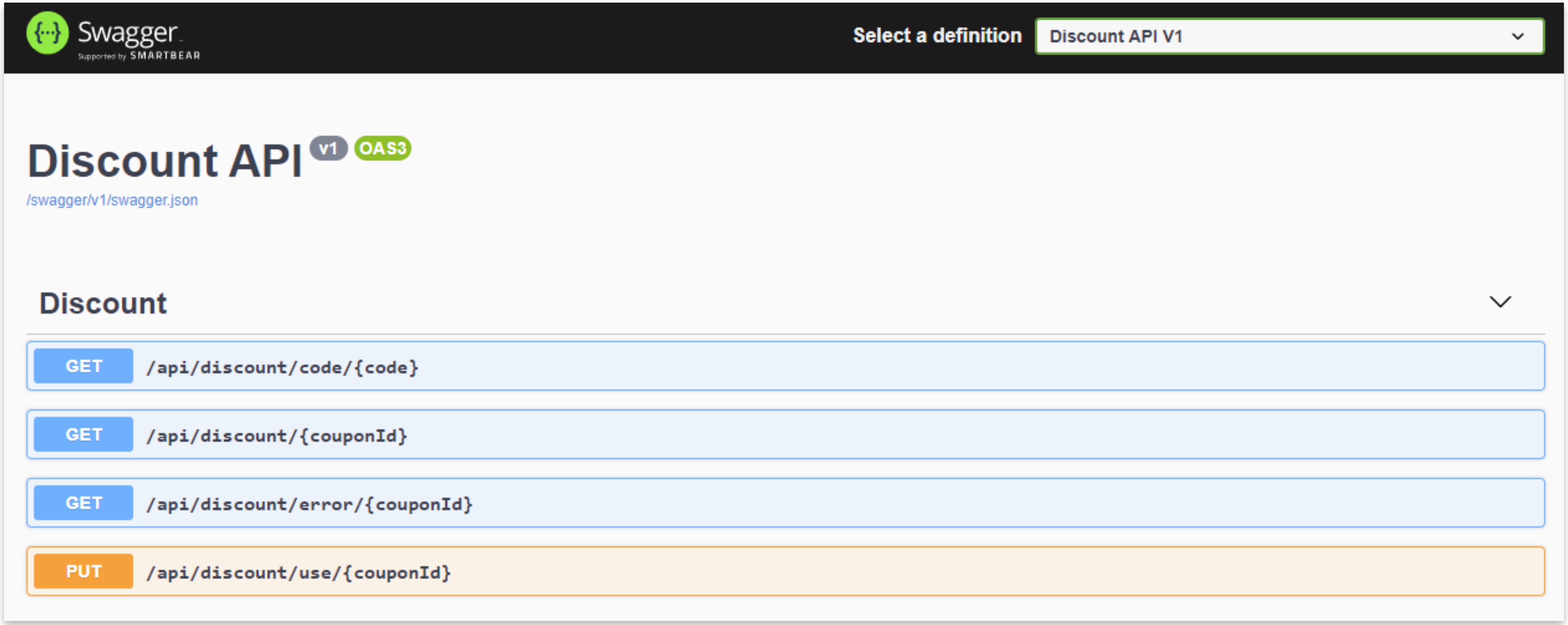
Exploring the Discount Service

"The DiscountService is ready.
How can we let other teams know about its functionalities?"

Adding Swagger

- API description
- Specification
  - JSON or YAML
- Tooling
  - Swashbuckle

Exposing the Swagger API Contract



Discount API v1 OAS3

/swagger/v1/swagger.json

Discount

```
GET /api/discount/code/{code}
GET /api/discount/{couponId}
GET /api/discount/error/{couponId}
PUT /api/discount/use/{couponId}
```

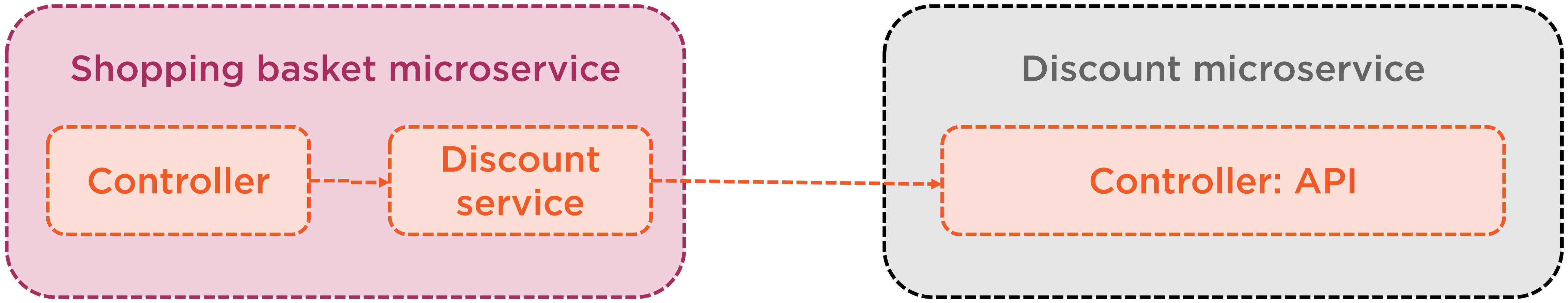
Demo

Adding Swagger to the discount microservice

Testing the microservice API

Inter-microservice Communication

Synchronous Microservice Communication



A Synchronous Await Call?

```
public async Task<Coupon> GetCoupon(Guid couponId)
{

    var response = await
        client.GetAsync($"/api/discount/{couponId}");

    return await response.ReadContentAs<Coupon>();

}
```

Demo

Connecting with the Discount Service from the Shopping Basket Service

Working with gRPC

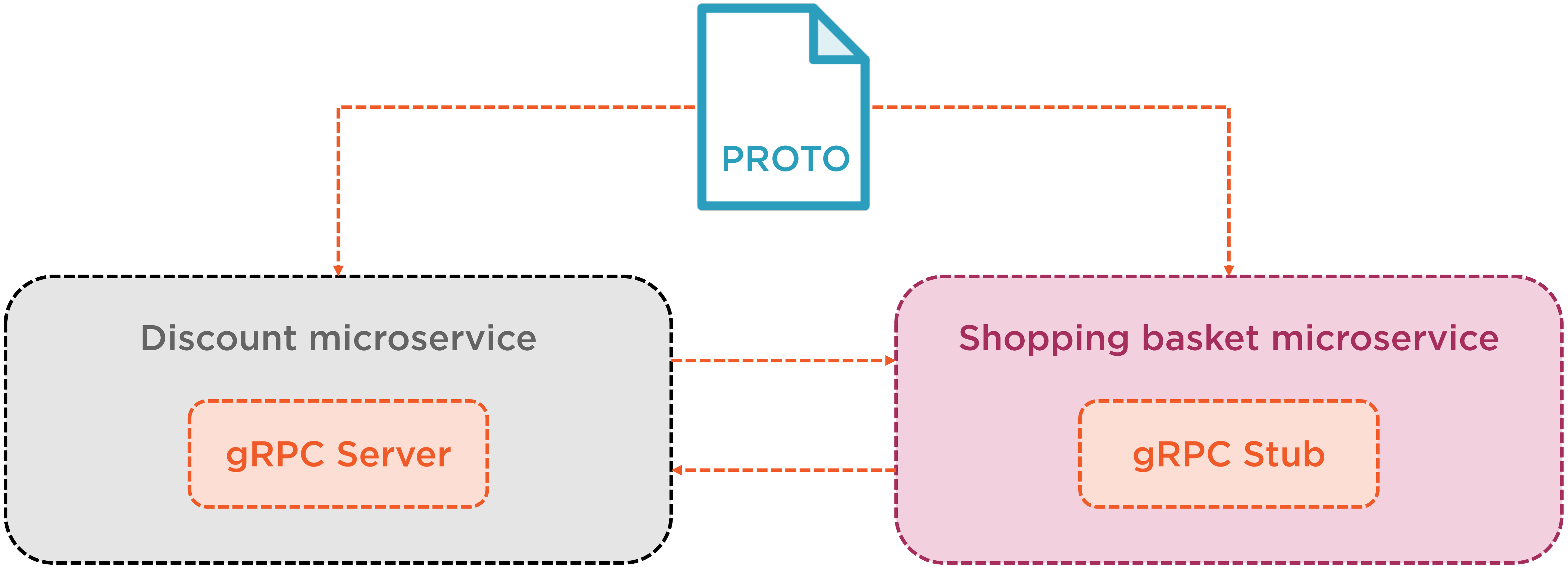
Using gRPC

RPC framework

High performance and lightweight

Supported from ASP.NET Core and many other languages

Using gRPC


PROTO
Discount microservice
gRPC Server
Shopping basket microservice
gRPC Stub

Based on .proto file

- Contract
- Client and server code generated
- Defines service and messages

Sample .proto File

```
service Discounts {
    rpc GetCoupon (GetCouponByIdRequest) returns (GetCouponByIdResponse) {}
}

message Coupon {
    string CouponId = 1;
    string Code = 2;
    int32 Amount = 3;
    bool AlreadyUsed = 4;
}
```

```
public class DiscountsService: Discounts.DiscountsBase
{

public override async Task<GetCouponByIdResponse>
    GetCoupon(GetCouponByIdRequest request, ServerCallContext context)
    {
        ...
    }

}
```

Exposing the Service Functionality

```
private readonly Discounts.DiscountsClient discountsService;

GetCouponByIdResponse getCouponByIdResponse =
    await discountsService.GetCouponAsync(getCouponByIdRequest);
```

Calling the gRPC Service

Demo

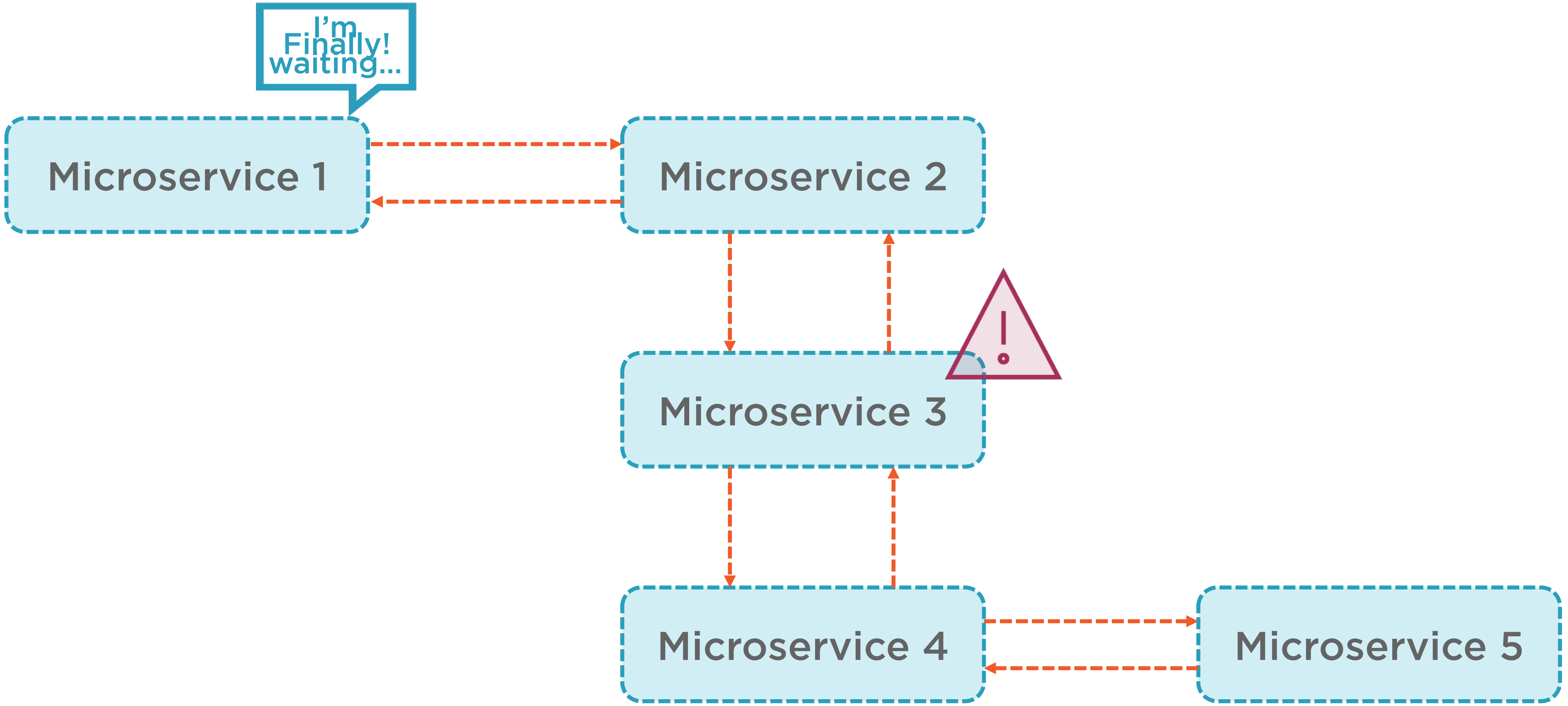
Exploring the gRPC approach for the Discount Service

Disadvantages of Synchronous Communication

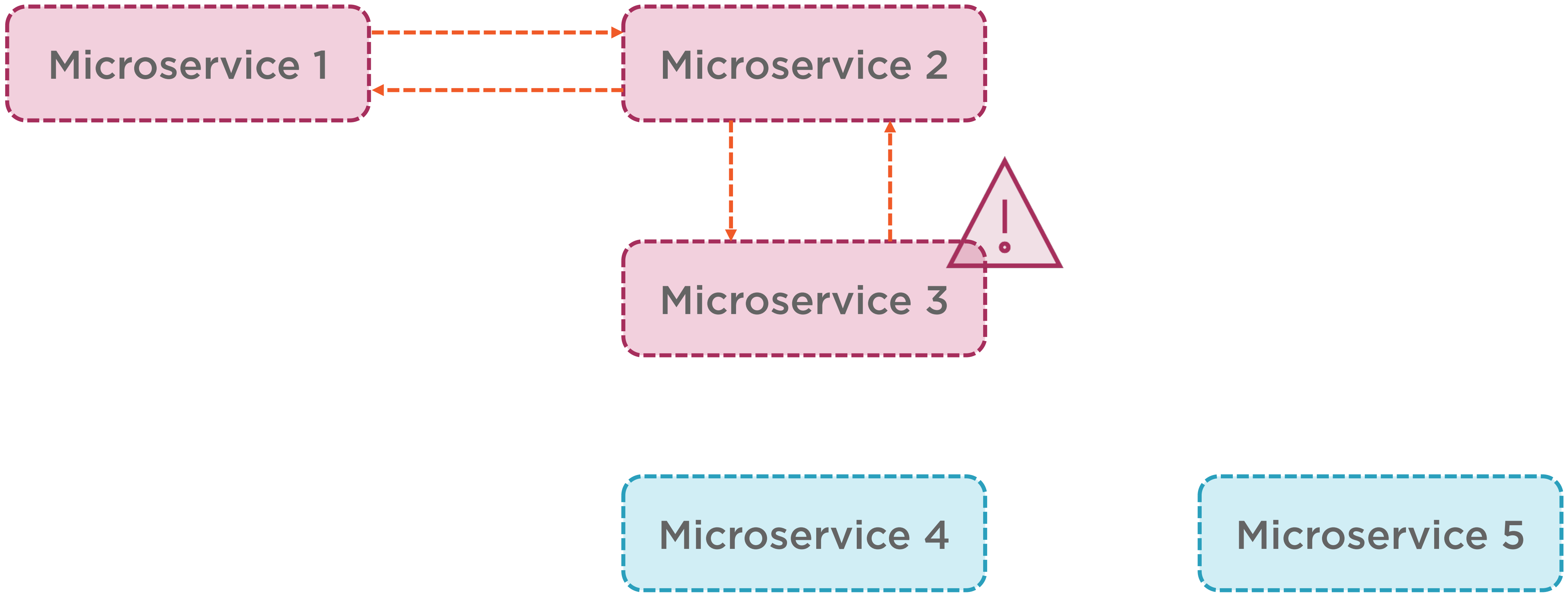
Disadvantages of Synchronous Communication

- Tight coupling
- Bottleneck in system
- Knowledge in one place
- One-to-many
- Changes are hard
- Errors difficult to catch

A Chain of Synchronous Calls



A Chain of Synchronous Calls



How can we solve this?

Asynchronous communication will solve a lot of the issues.

Summary

Synchronous communication uses standard service approach

REST, gRPC work well

Not a perfect fit for all communication in a microservice system

Up next:

Asynchronous service communication