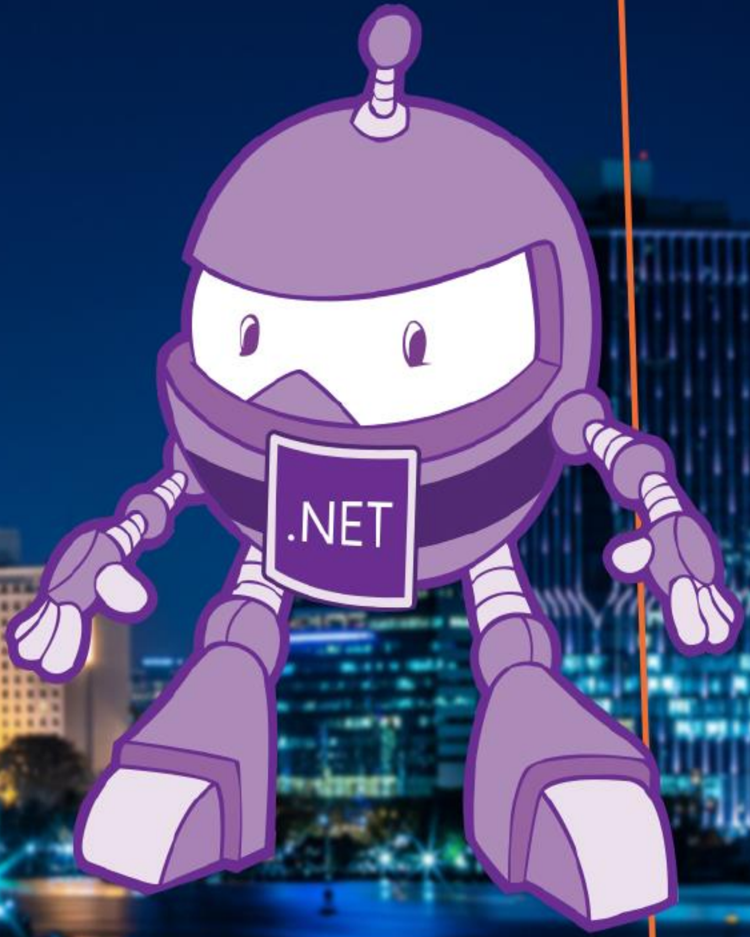
ORGANIZER DEV CAFE Connect and share

SPONSOR NASH TECH The Power to Innovate Microsoft

.NET Conf

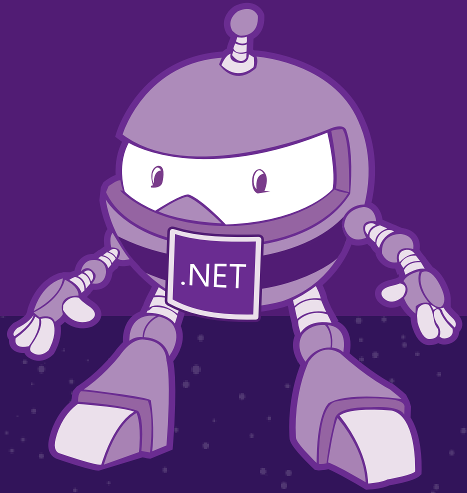
05 OCT 2019 | Ho Chi Minh City

COMMUNITY PARTNER Xamarin Developers Vietnam .NET



Building Cloud-native Apps with .NET Core 3.0

Thang Chung – DevArch
https://github.com/thangchung



Agenda

- Introduction
- Microservice Templates
- Microservices Modelling
- REST vs gRPC Protocol
- Docker Images
- Kubernetes
- Service Mesh

Introduction

Moving FAST

https://pixabay.com/photos/blue-angels-jets-navy-military-sky-946754/

Cloud-native Apps Characteristics

- Smaller
- Lightweight
- Good communication and connection
- Secure
- Deployment frequently and periodically
- Spin up faster on the managed-orchestrator

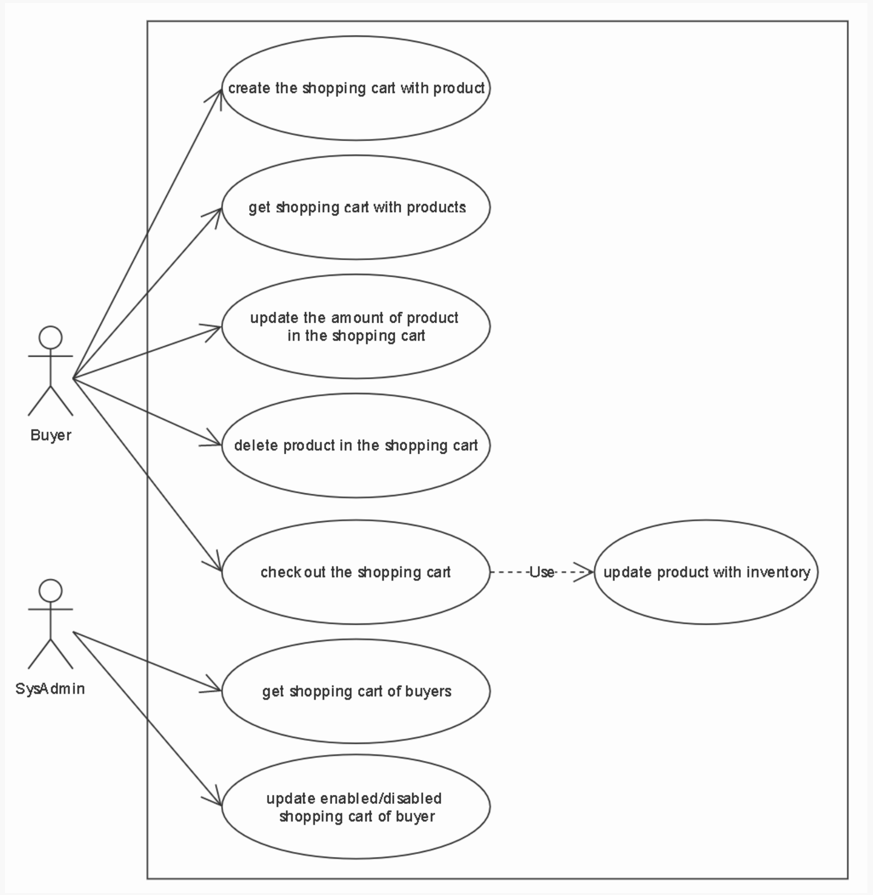
https://pixabay.com/illustrations/matrix-network-data-exchange-1027571/

Microservices Modelling

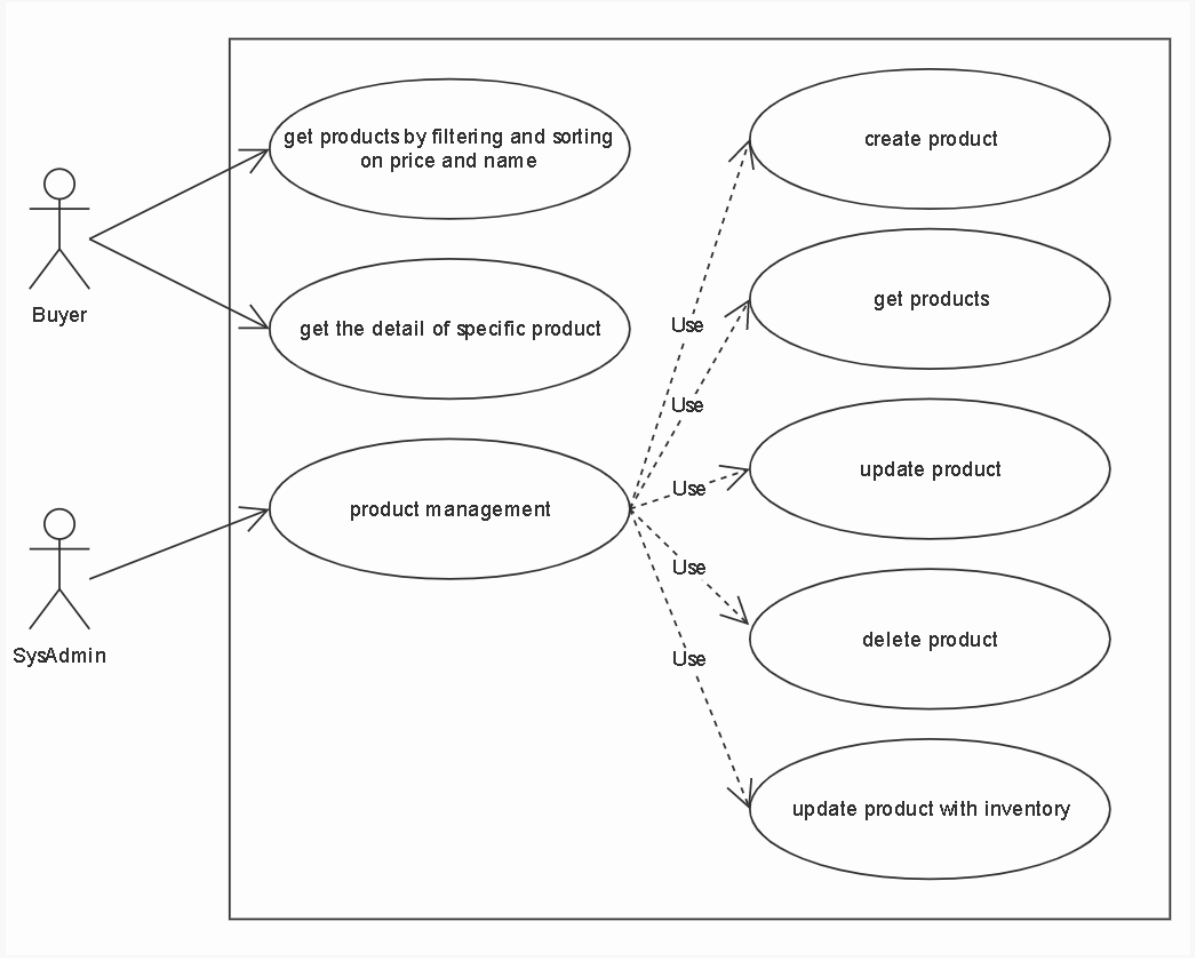
https://vietnam-devs.github.io/coolstore-microservices/model-microservices/#business-context

.NET

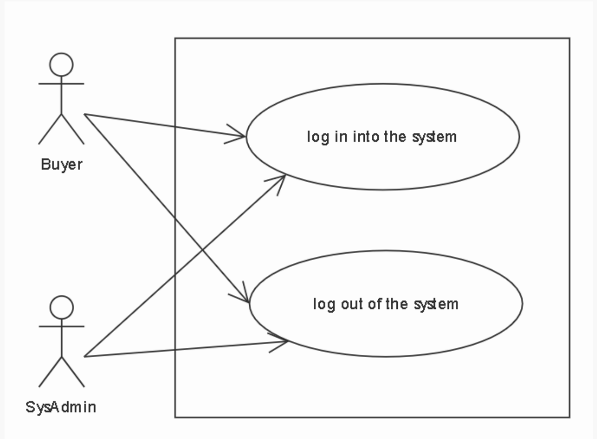
Business Context



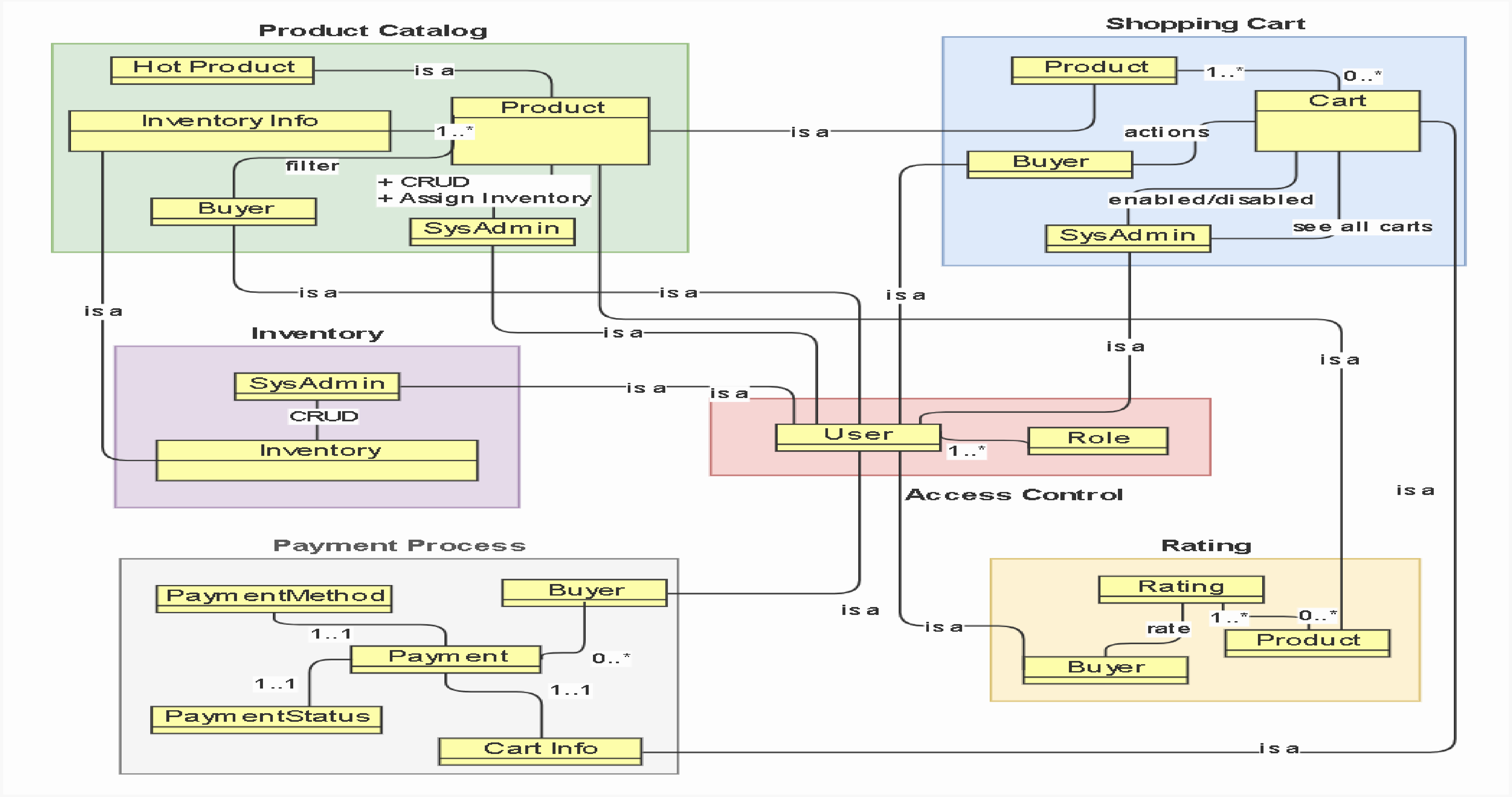
Buyer
SysAdmin
create the shopping cart with product
get shopping cart with products
update the amount of product in the shopping cart
delete product in the shopping cart
check out the shopping cart
Use
update product with inventory
get shopping cart of buyers
update enabled/disabled shopping cart of buyer



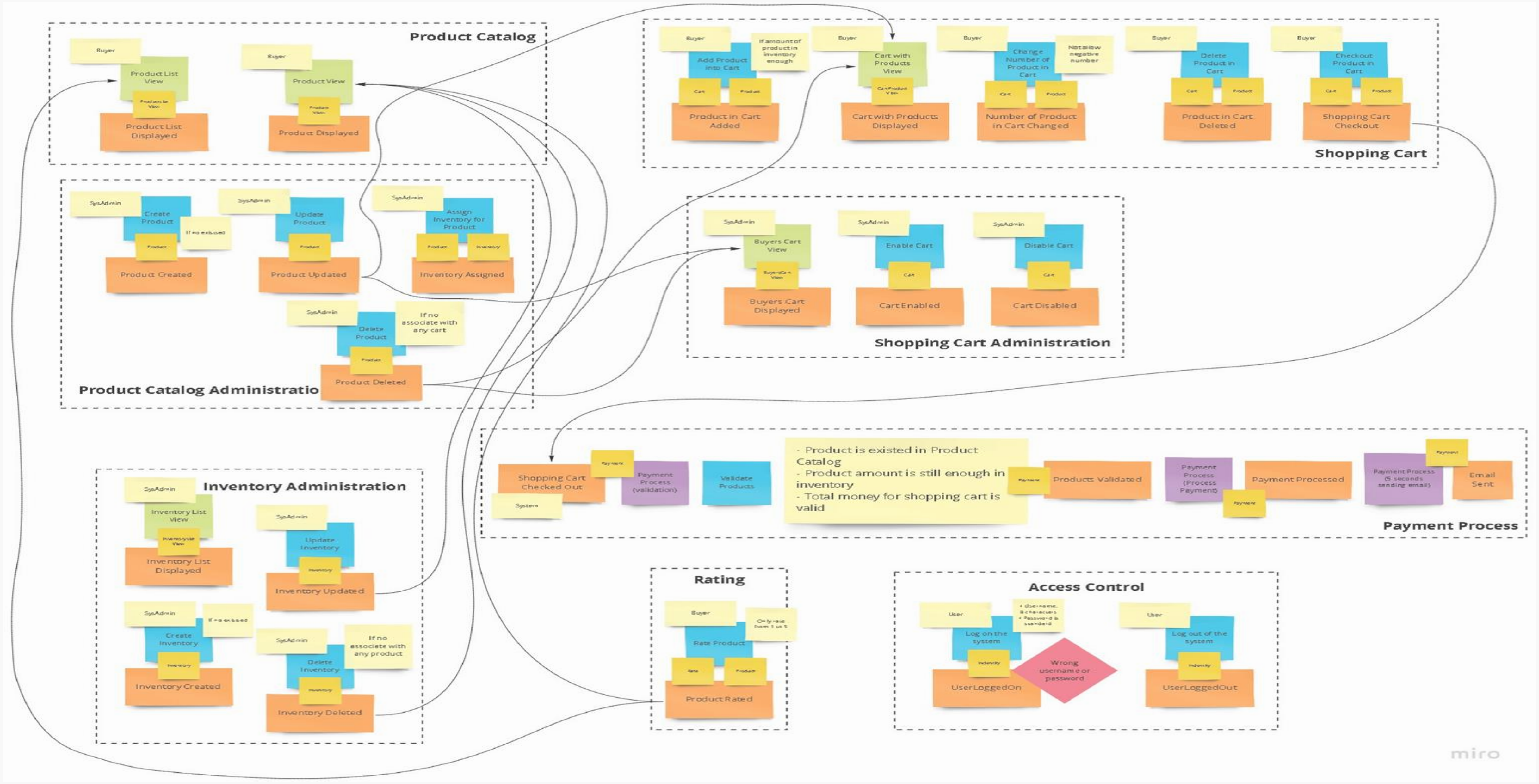
Buyer
SysAdmin
get products by filtering and sorting on price and name
get the detail of specific product
product management
Use
Use
Use
Use
Use
create product
get products
update product
delete product
update product with inventory



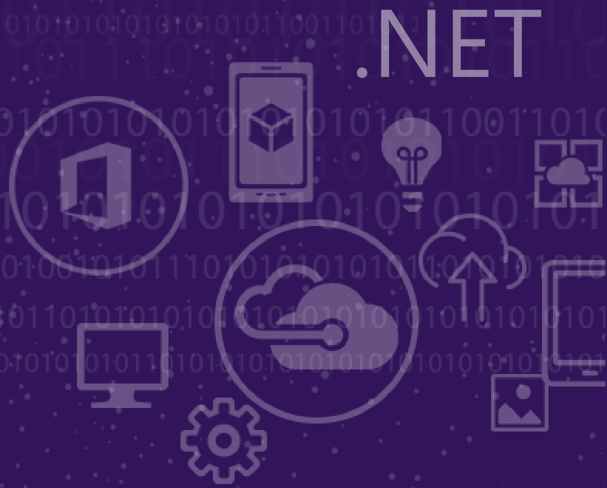
Conceptual Model



Event Storming



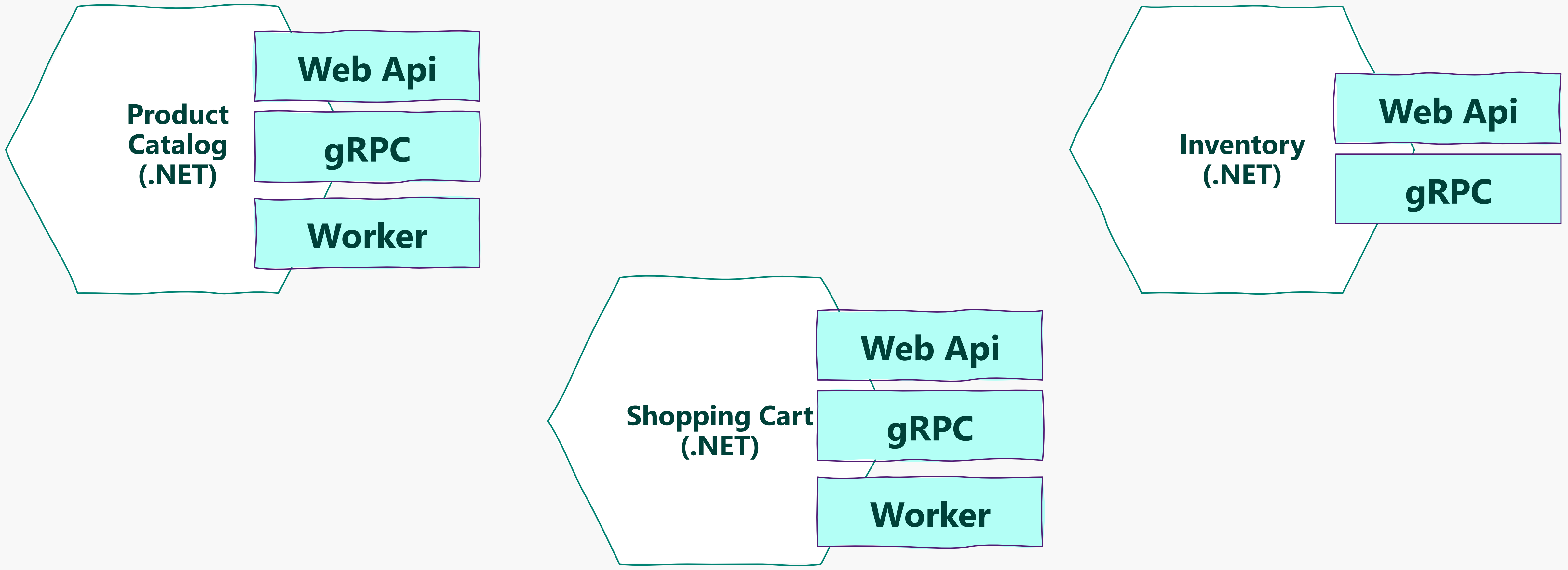
.NET Core 3.0 Microservice Templates



.NET Core 3.0 Microservice Template

- Web API template
- gRPC template (*)
- Worker template (*)

DEMO – Multiple Protocols on Host



REST vs gRPC protocol

.NET

JSON

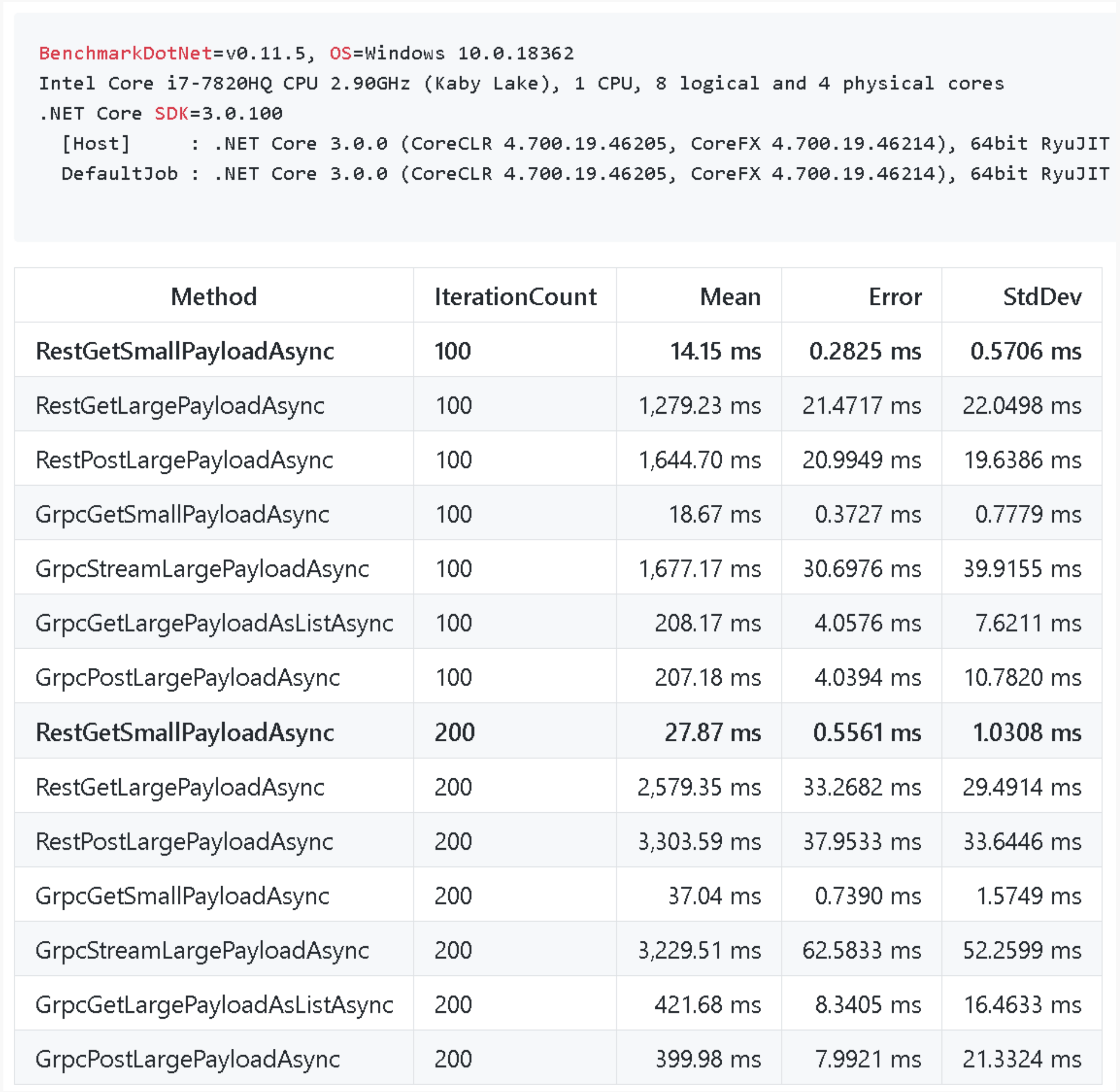
```json
{
  "swagger": "2.0",
  "info": {
    "title": "Coolstore services",
    "version": "1.0",
    "contact": {
      "name": "coolstore-microservices project",
      "url": "https://github.com/vietnam-devs/coolstore-microservices",
      "email": "thangchung.onthenet@gmail.com"
    }
  },
  "schemes": [
    "http",
    "https"
  ],
  "consumes": [
    "application/json"
  ],
  "produces": [
    "application/json"
  ],
  "paths": {
    "/cart/api/carts": {
      "post": {
        "operationId": "InsertItemToNewCart",
        "responses": {
          "200": {
            "description": "A successful response.",
            "schema": {
              "$ref": "#/definitions/coolstoreInsertItemToNewCartResponse"
            }
```

Protobuf

```protobuf
syntax = "proto3";
package coolstore;
option csharp_namespace = "VND.CoolStore.ShoppingCart.DataContracts.Api.V1";

import "cart_dto.proto";

service ShoppingCartApi {
  rpc GetCart(GetCartRequest) returns (GetCartResponse) {};
  rpc InsertItemToNewCart(InsertItemToNewCartRequest) returns (InsertItemToNewCartResponse) {};
  rpc UpdateItemInCart(UpdateItemInCartRequest) returns (UpdateItemInCartResponse) {};
  rpc Checkout(CheckoutRequest) returns (CheckoutResponse) {};
  rpc DeleteItem(DeleteItemRequest) returns (DeleteItemResponse) {};
}

message GetCartRequest {
  string cart_id = 1;
}

message GetCartResponse {
  repeated CartWithProductsRow rows = 1;
}
```

REST and gRPC Benchmark

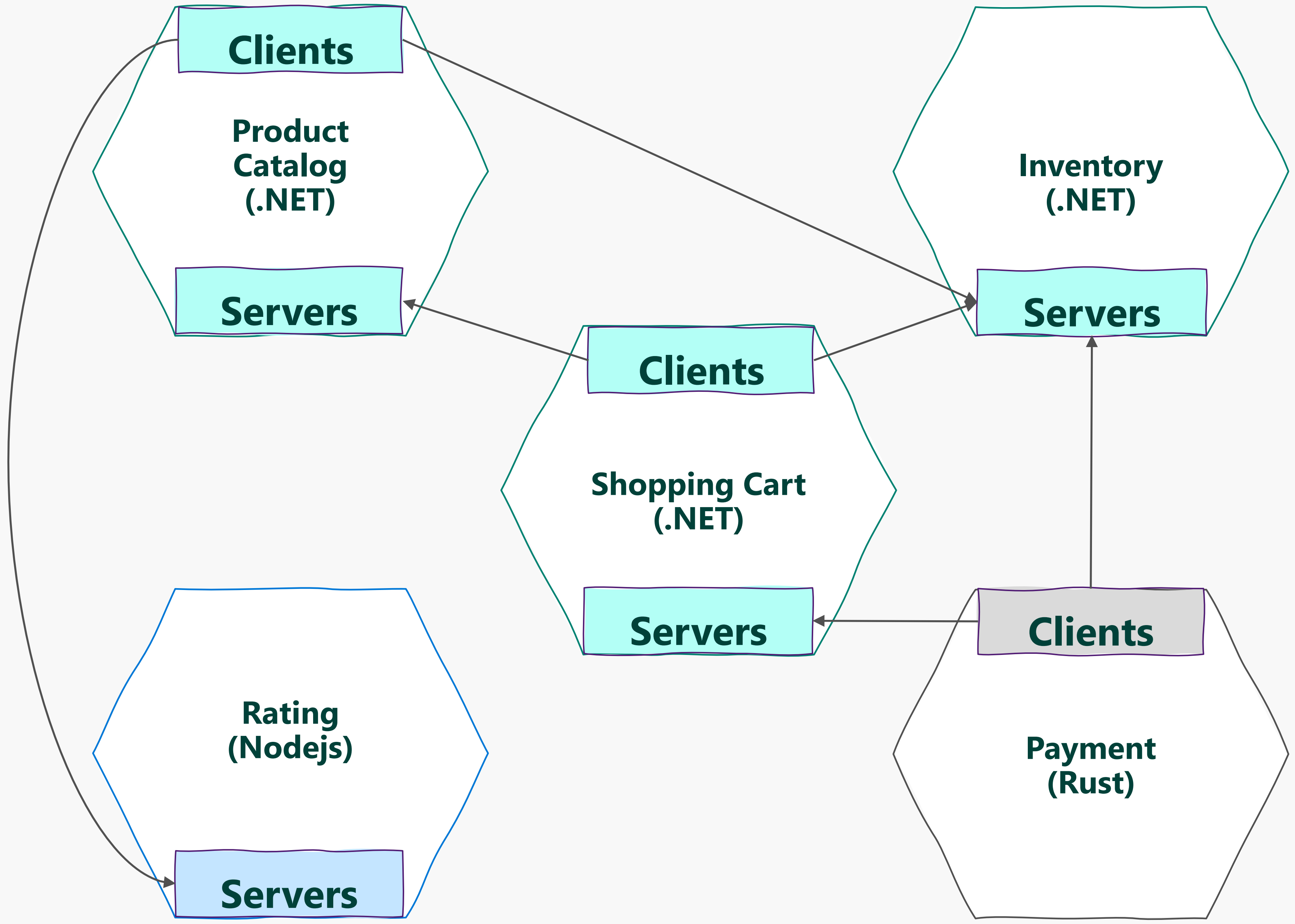
https://github.com/thangchung/RESTvsGRPC



```
BenchmarkDotNet=v0.11.5, OS=Windows 10.0.18362
Intel Core i7-7820HQ CPU 2.90GHz (Kaby Lake), 1 CPU, 8 logical and 4 physical cores
.NET Core SDK=3.0.100
  [Host]     : .NET Core 3.0.0 (CoreCLR 4.700.19.46205, CoreFX 4.700.19.46214), 64bit RyuJIT
  DefaultJob : .NET Core 3.0.0 (CoreCLR 4.700.19.46205, CoreFX 4.700.19.46214), 64bit RyuJIT
```

Method	IterationCount	Mean	Error	StdDev
RestGetSmallPayloadAsync	**100**	**14.15 ms**	**0.2825 ms**	**0.5706 ms**
RestGetLargePayloadAsync	100	1,279.23 ms	21.4717 ms	22.0498 ms
RestPostLargePayloadAsync	100	1,644.70 ms	20.9949 ms	19.6386 ms
GrpcGetSmallPayloadAsync	100	18.67 ms	0.3727 ms	0.7779 ms
GrpcStreamLargePayloadAsync	100	1,677.17 ms	30.6976 ms	39.9155 ms
GrpcGetLargePayloadAsListAsync	100	208.17 ms	4.0576 ms	7.6211 ms
GrpcPostLargePayloadAsync	100	207.18 ms	4.0394 ms	10.7820 ms
RestGetSmallPayloadAsync	**200**	**27.87 ms**	**0.5561 ms**	**1.0308 ms**
RestGetLargePayloadAsync	200	2,579.35 ms	33.2682 ms	29.4914 ms
RestPostLargePayloadAsync	200	3,303.59 ms	37.9533 ms	33.6446 ms
GrpcGetSmallPayloadAsync	200	37.04 ms	0.7390 ms	1.5749 ms
GrpcStreamLargePayloadAsync	200	3,229.51 ms	62.5833 ms	52.2599 ms
GrpcGetLargePayloadAsListAsync	200	421.68 ms	8.3405 ms	16.4633 ms
GrpcPostLargePayloadAsync	200	399.98 ms	7.9921 ms	21.3324 ms

DEMO gRPC Protocol



```
$ docker-compose -f docker-compose.yml -f docker-compose.dev.yml up -d
```

More information is at https://github.com/vietnam-devs/coolstore-microservices

gRPC Pros and Cons

- Human read
- Contract based
- Effective binary serialization (low CPU overhead) using Protobuf
- Smaller payload
- HTTP/2 is default
- Code-gen in many languages/frameworks

- grpc-dotnet is only working with .NET language
- Kestrel doesn’t support HTTP/2 with TLS on Mac
- Default is not working with Load Balancer in Kubernetes
- Client tooling is not popular such as Postman

.NET Core 3.0 Docker Images

.NET

Docker Images

.NET SDK

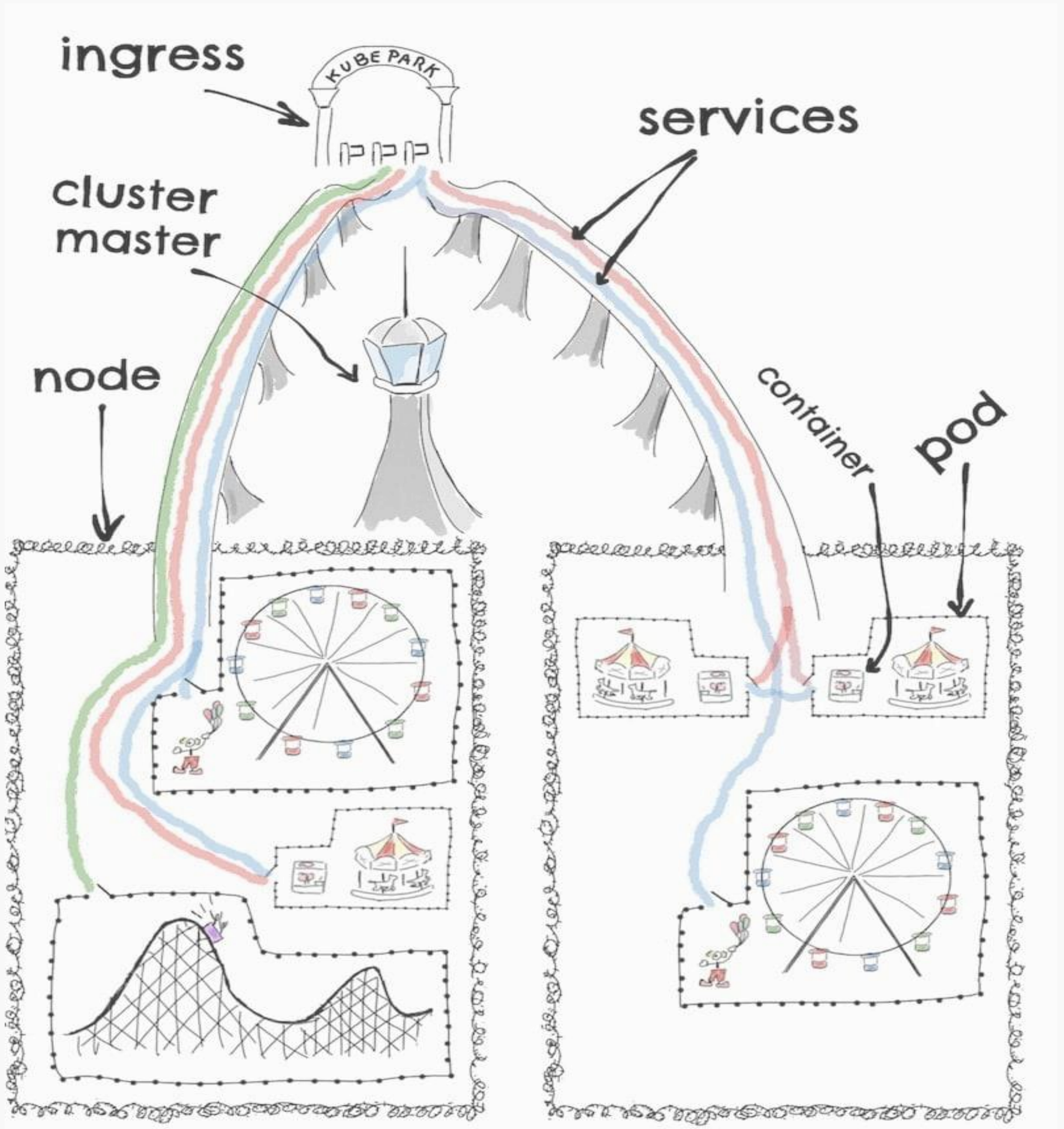
- /dotnet/core/aspnet:2.2-stretch-slim – 261MB
- /dotnet/core/aspnet:3.0-buster-slim – 207MB
- /dotnet/core/aspnet:2.2-alpine – 166MB
- /dotnet/core/aspnet:3.0-alpine – 106MB

.NET Runtime

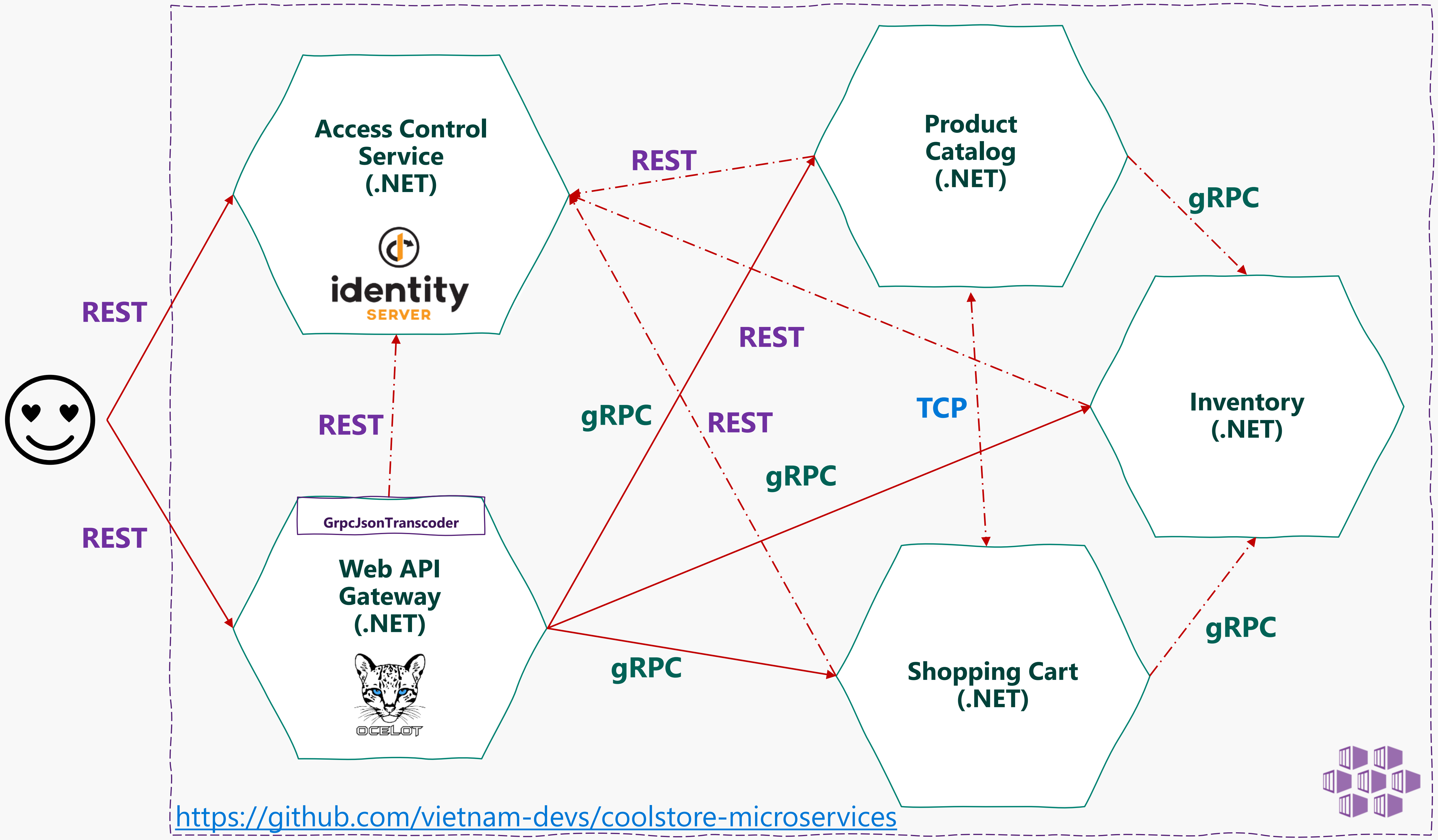
- /dotnet/core/runtime:3.0-buster-slim – 189MB
- /dotnet/core/runtime:3.0-alpine – 88MB

.NET Core 3.0 on Kubernetes

What is Kubernetes?



https://dev.to/danlebrero/kubernetes-explained-in-pictures-the-theme-park-analogy-3d99



https://github.com/vietnam-devs/coolstore-microservices

Demo

CoolStore-Microservices on Kubernetes

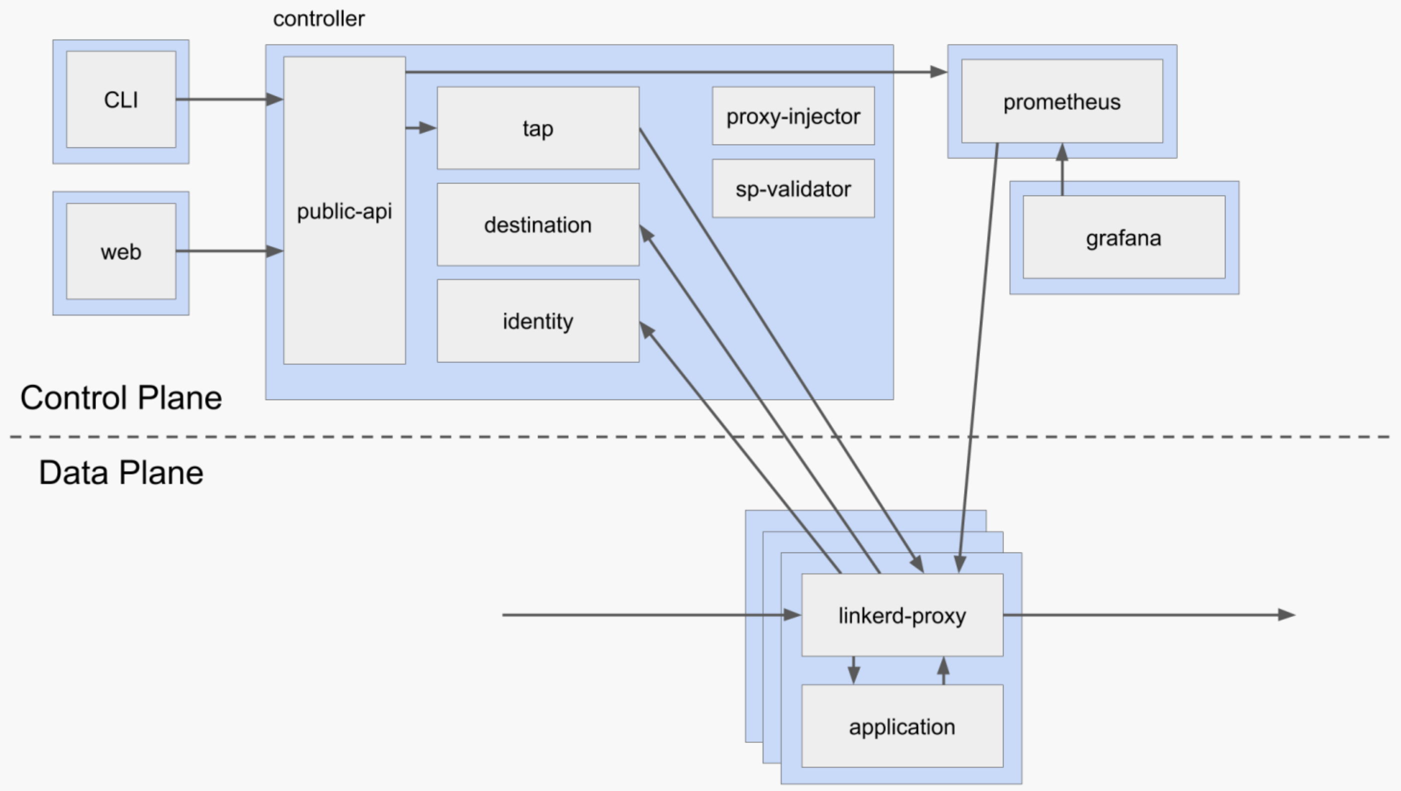
```
$ kubectl apply -f coolstore-infrastructure.yaml
$ kubectl apply -f coolstore-migration.yaml
$ kubectl apply -f coolstore.yaml
$ octant
```

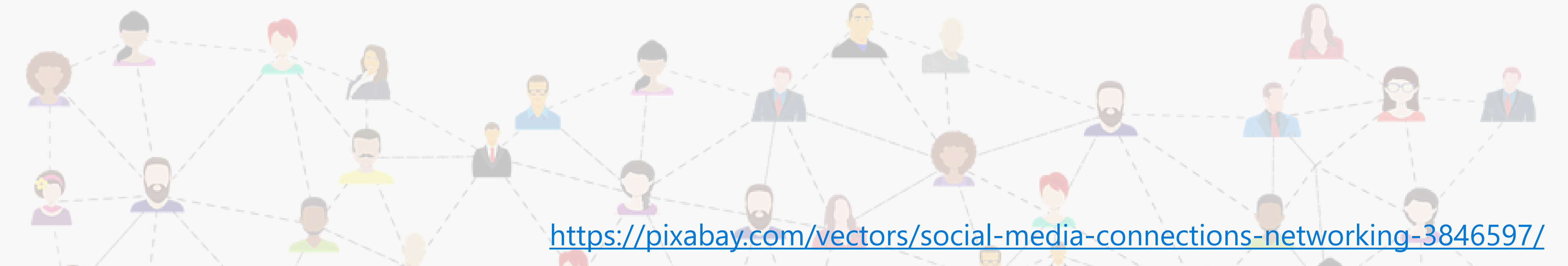
More information is at https://github.com/vietnam-devs/coolstore-microservices

Manage communication with Service Mesh – Linkerd2

What is Service Mesh?

- Traffic management
- Observability
- Policy enforcement
- Security





https://pixabay.com/vectors/social-media-connections-networking-3846597/

Demo

CoolStore-Microservices on Linkerd2

.NET

```
$ kubectl get -n coolstore deploy -
o yaml | linkerd inject - | kubectl apply -f -
$ linkerd dashboard --port 9999
```

Now you can see gRPC Services can be load-balancer correctly ☺

More information is at https://github.com/vietnam-devs/coolstore-microservices

“Sometimes your circle decreases in size but increases in value” – Unknown

COMMUNICATION is CRUCIAL

https://pixabay.com/illustrations/lego-frame-lego-building-blocks-3774083/

References

- https://dot.net
- https://docs.microsoft.com/en-us/aspnet/core/grpc/?view=aspnetcore-3.0
- https://github.com/dotnet/docs/blob/master/docs/architecture/grpc-for-wcf-developers/index.md
- https://www.docker.com
- https://grpc.io
- https://kubernetes.io
- https://linkerd.io/2/reference/architecture
- https://hbr.org/2019/01/the-era-of-move-fast-and-break-things-is-over

Q&A

.NET