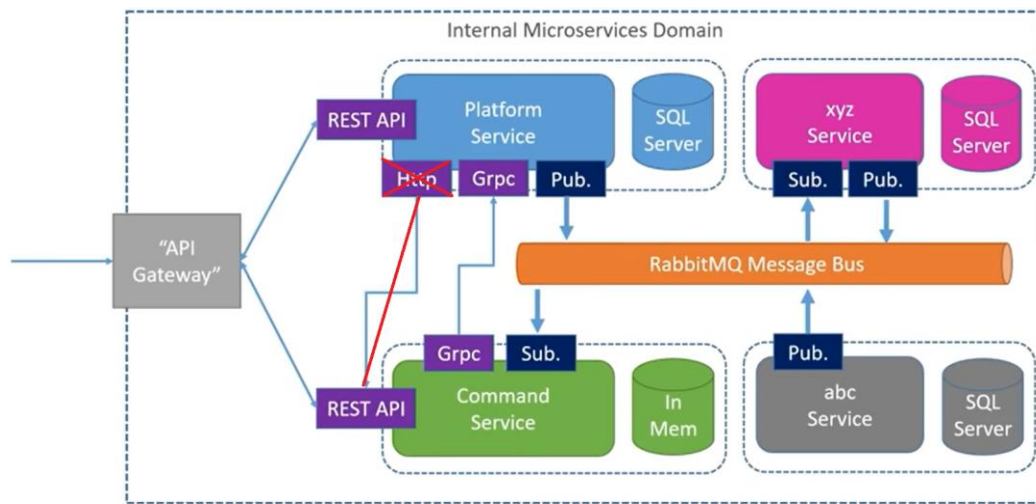
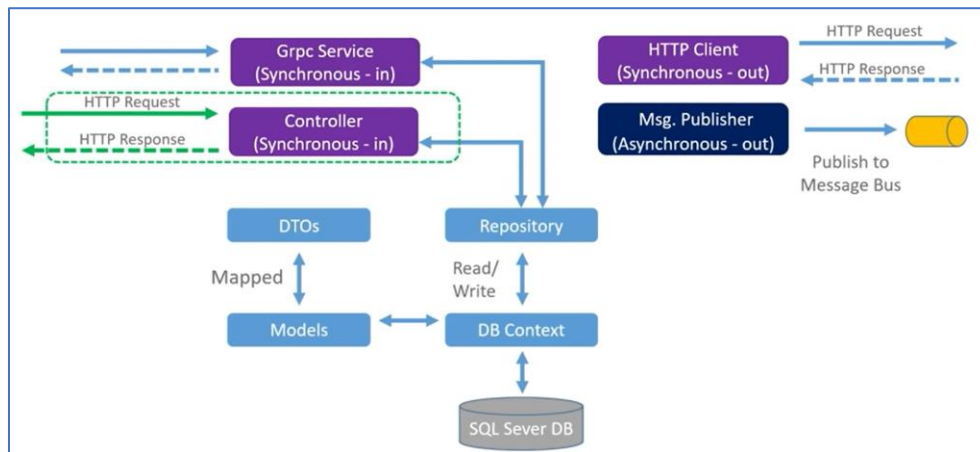


Microservices Best Practices Project

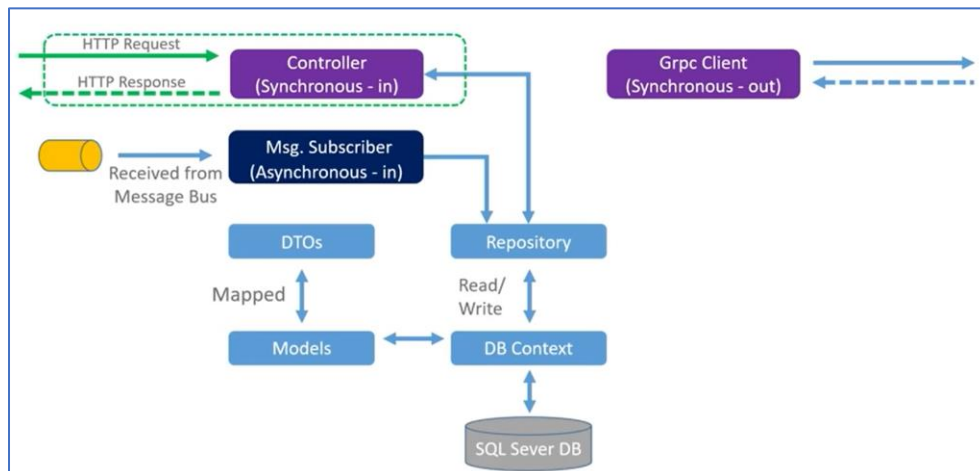
Stav Sofer 2021

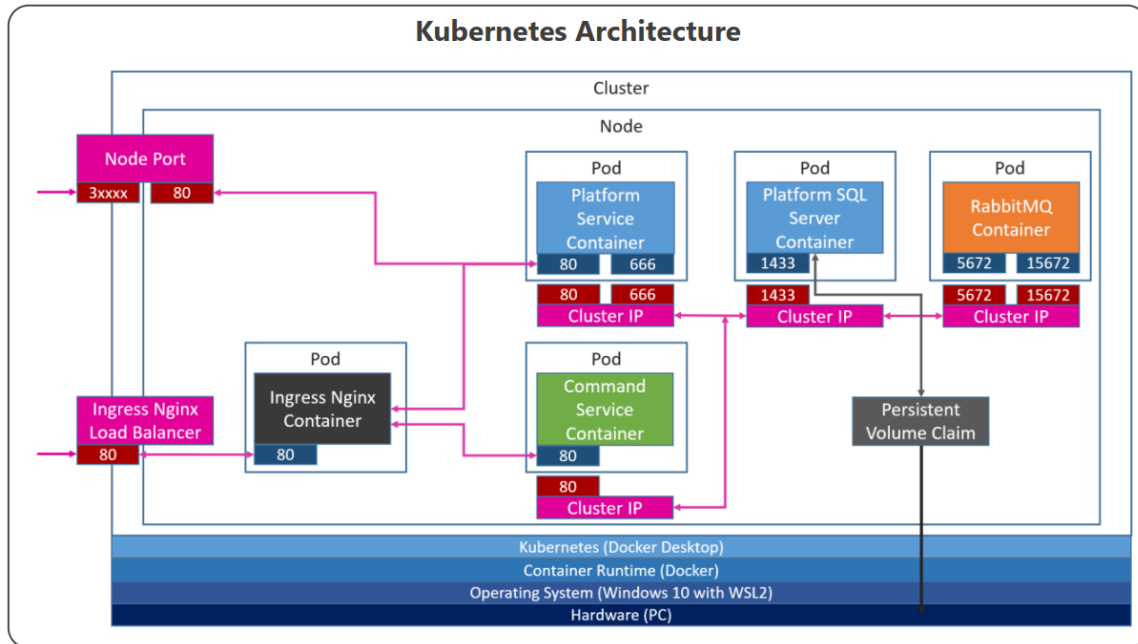


Platforms Service:



Commands Service:





Technologies In Use:

- Dotnet 5
- Microservices
- Docker
- Kubernetes
- Synchronous Http Restful API
- Synchronous gRPC & Protobufs
- Asynchronous Events Based Messaging using RabbitMQ
- Background Service
- API Gateway – Ingress Nginx
- Dependency Injection & Interface-Repository Approach
- Data Transfer Objects (DTOs)
- Auto Mapper
- Entity Framework Core

Endpoints & URIs:

Local base URL: <https://localhost:5001/> & <https://localhost:6001/>

Production base URL: <http://acme.com/>

Platforms

GET /api/Platforms

POST /api/Platforms

GET /api/Platforms/{id}

Commands

GET /api/c/platforms/{platformId}/Commands

POST /api/c/platforms/{platformId}/Commands

GET /api/c/platforms/{platformId}/Commands/{commandId}

Platforms

GET /api/c/Platforms

K8S Pods on Docker Desktop:



k8s_platformservice_platforms-depl-5868588f7-znmcv_default_63047bd9-4846-4127-9a71-2674b6d356bc_0 stavsofer/com...
RUNNING



k8s_commandservice_commands-depl-7cb7c8d88-h74x8_default_5dfb9d5d-e137-48d1-a8b7-d1b05e34be0f_4 stavsofer/com...
RUNNING



k8s_mssql_mssql-depl-856b8c48fd-fs6sh_default_012ae1bb-20da-4353-9f9e-2fb5b2dcfe88_5 sha256:5af364e...
RUNNING



k8s_controller_ingress-nginx-controller-fd7bb8d66-bds4s_ingress-nginx_e5a634b2-83f2-462d-b277-01341914ca7a_6 sha256:ef43679...
RUNNING



k8s_rabbitmq_rabbitmq-depl-76f9ff665c-w97fv_default_3cdf4c0d-bedc-47de-bbcc-3d442768bce4_3 sha256:3e83da...
RUNNING

Useful Commands:

Docker:

```
docker build -t <docker user id>/<image name>:<version> .  
docker push <docker user id>/<image name>:<version>  
docker ps  
docker run -p <external port>:<internal port> -d <docker user id>/<image name>  
docker stop <container Id>  
  
docker start <container Id>
```

K8S:

```
Kubectl apply -f <name of yaml file>  
  
Kubectl rollout restart deployment <name of deployment>  
  
Kubectl get namespace  
  
Kubectl get deployments  
Kubectl get pods --namespace=<name of namespace>  
  
Kubectl get <object type>  
Kubectl delete <object type> <object name>  
  
kubectl scale --replicas=<i> deployment/<name of deployment>  
  
(i=0 → Stop the pod, i=1 → start the pod)
```