About Myself

# My carrer & technologies

More than seventeen years working as a web developer and architecting .Net solutions, always making sure that SOLID principles are applied using C#, Entity Framework, Web API, SQL Server, JavaScript and others cutting-edge technologies. Considering quality and performance, I am used to apply DDD, Tests and being an Agile enthusiast.

# Companies Sumary

* FENABRAVE / FIESP
* HOSPITAL ALBERT EINSTEIN
* BANCO REAL / CAIXA ECONOMICA
* PEPSI
* SKY
* AZUL AIRLINES
* SERASA EXPERIAN
* MARISA
* AVON
* FARFETCH

The Challenge

There is a scooter rental legacy system that needs to be scalable to support the business growth.

### Challenge architectural statements:

* Implement micro-services
* Guarantee that all communication between the services be resilient
* Use container
* Must be scalable to support the business network initiatives

# Technical statements

### Problem Definition:

* Hard to maintain the code and new features.
* Too much effort and expensive costs to create and scale Windows Virtual Machine (VM)
* The database is not supporting the data volume without lost performance
* Bad user experience

### Proposal Solution:

The proposal includes to migrate the currently technology for:

|  |  |
| --- | --- |
| LEGACY | NEW |
| .net Framework 4.6.1 | .net Core 3.1 |
| Monolith MVC on IIS | Microservices using Docker & Kubertes |
| Razor | Angular 2+ |
| SqLite | SqlServer |
| Virtual Machine Host | Azure |

Why use these technologies?

* .net Core

Performance, Mult-Platform

* Docker

Container

* Kubernetes

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services. It has a large, rapidly growing ecosystem, turning the application widely available.

* Angular
* SQL Server
* Azure

Way to be followed in architecture:

* Domain Driven Design (DDD)

It is a group of techniques to make your code near from the business language, broken it in specific domains, making possible to identify boundaries and microservices

* Domain Events

Some action will have a reaction to do something related from it

* “Service Bus” as queue

In case of unavailable service, the queue will guarantee the communication, keeping the microservices working independently

The Azure Service Bus has a dead letter and retry schemas in case of exception on service before Pop.

* “Azure Functions” as Job

In this case will be executed in time to time

What is the benefits:

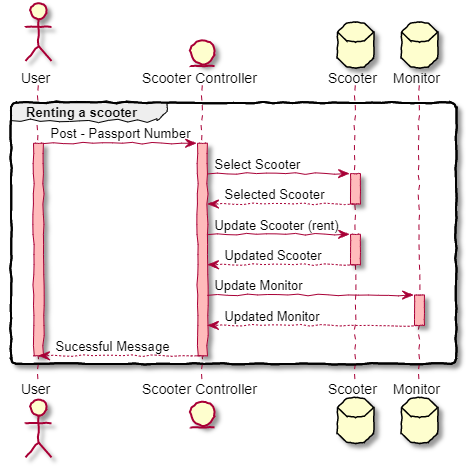
* Easy to scale to support an elevate web traffic and data
* Easier to increase new features on code
* Multiplatform and scalable system
* Increase of web performance and user experience
* Lower cost in pay-per-use on cloud

# Business Scenario

*\* The current business rules will be maintained*

### Rent scooters

Legacy

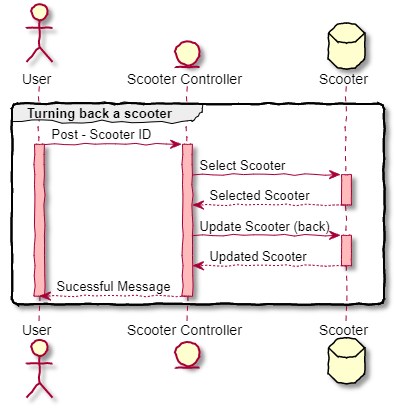


Proposal

<<Diagram>>

### Turn Back scooters

Legacy

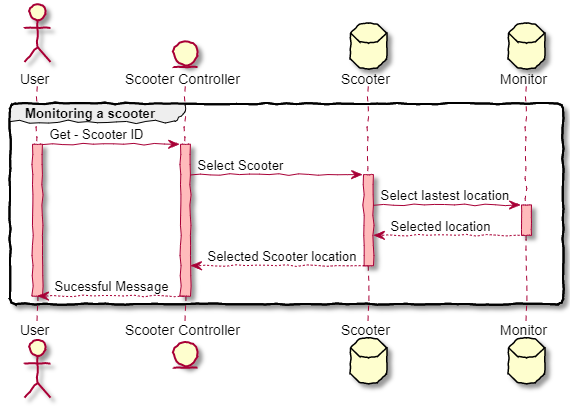


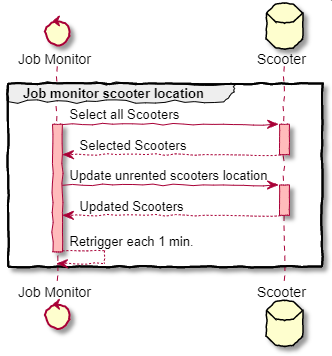
Proposal

<<Diagram>>

### Tracking scooters

Legacy





Proposal

<<Diagram>>

# Technical Scenario

### Identified Domains

* Scooter
* Tracking