Design Document

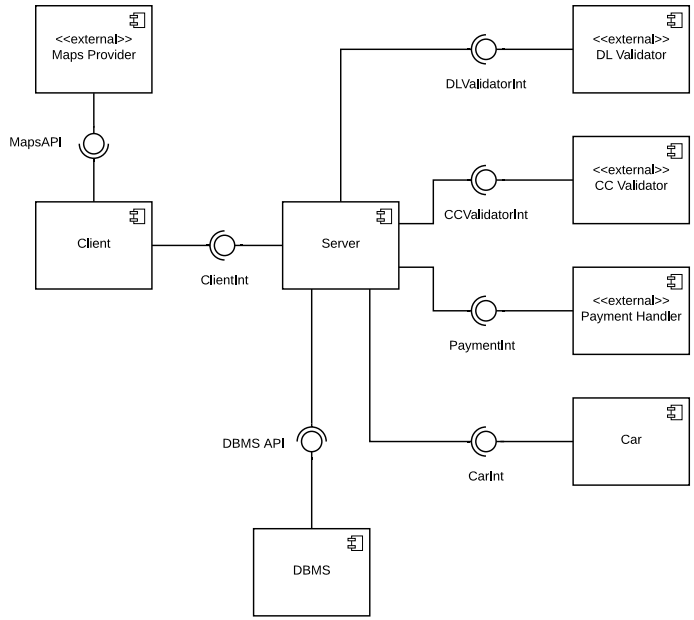
Philippe Scorsolini,
Lorenzo Semeria,
Gabriele Vanoni

Politecnico di Milano

December 14, 2016

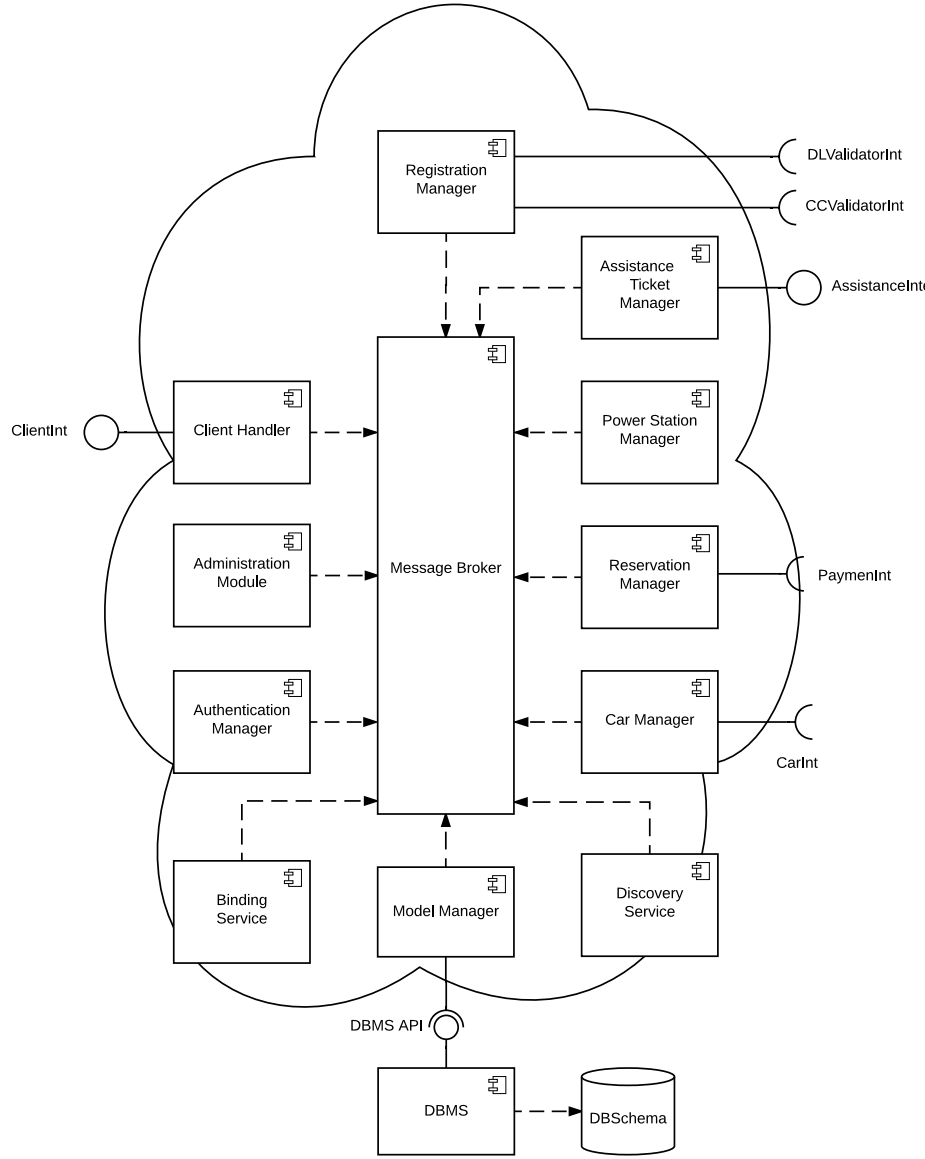
Overview

- Three tier
- Thin client: mobile app for both operators and users, web app only for users
- RESTful APIs over HTTPS in JSON between server and clients
- OpenVPN with the cars and MQTT with the power stations
- Microservice oriented architecture with shared database



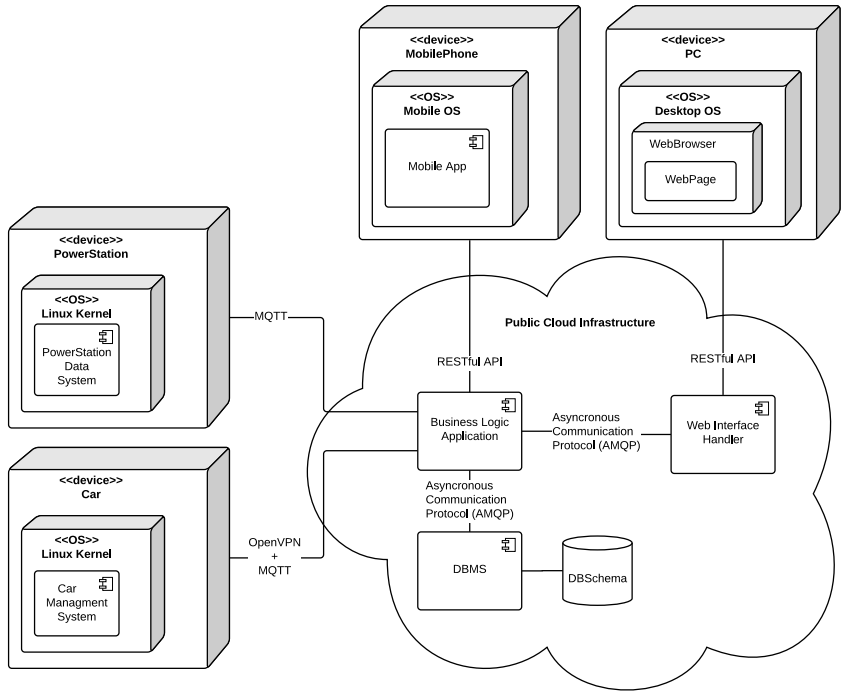
Serverside Component View

- Microservice architecture
- Stateless, indipendently deployable and business domain specific
- Message oriented (RabbitMQ)
- AMQP (Advanced Messaging Queue Protocol) Asynchronous communication
- Discovery and Binding Services to setup new services at instantiation
- Allows advanced traffic routing features and elastic load balancing



Deployment View

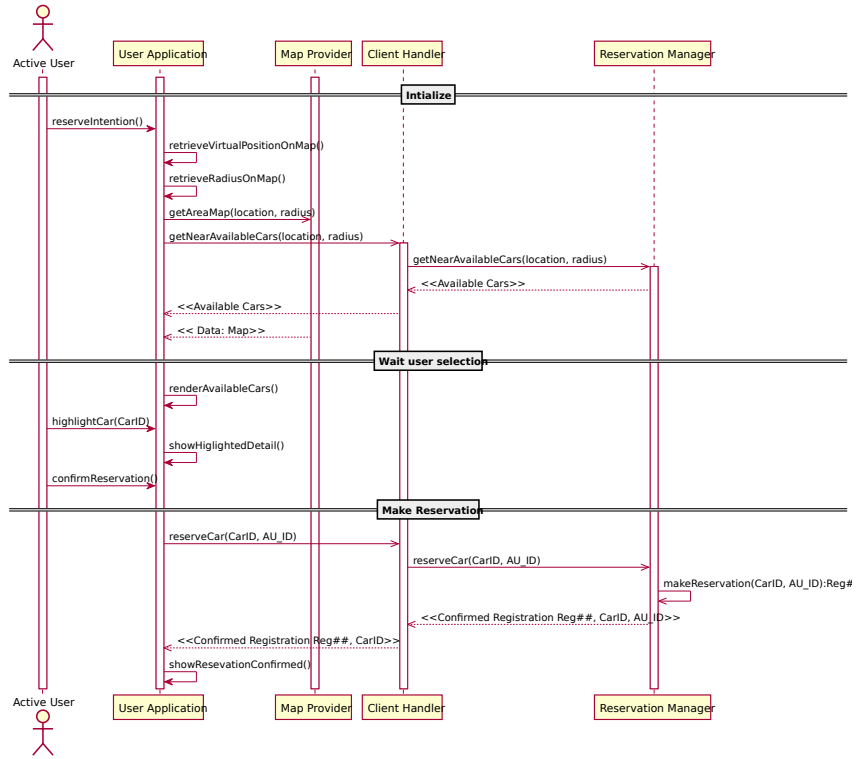
- Various Microservices : AMQP
- OnBoardCarManager : OpenVpn + MQTT
- PowerStationDataSystem : MQTT Protocol
- MobileApp : RESTful APIs
- WebPage : RESTful APIs



Sequence Diagram

This Diagram represents the intentions needed to complete the reservation of a car:

1. The Map and the current car's position is retrieved
2. The User chooses a car and tries to reserve it
3. The Car is reserved and a confirmation is sent to the user

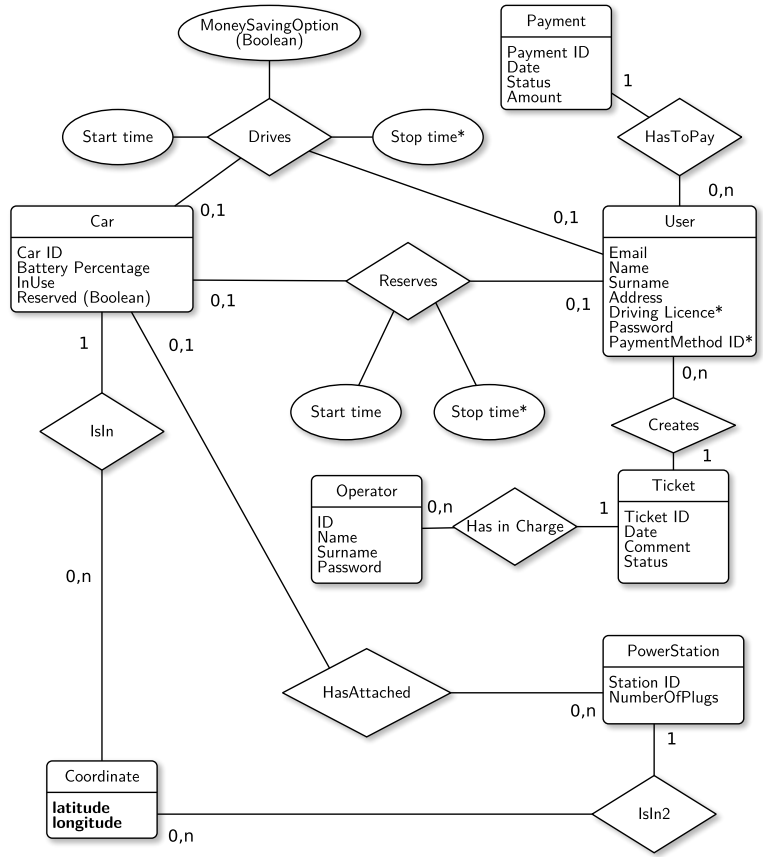


Selected Architectural Styles And Patterns

- Three-Tier Cloud Application
- Microservice Architectural Style
  - testability
  - scalability
  - deployability
  - isolation
  - extensibility
- Message Bus Architectural Style
- REST architectural style
- Data Access Component
- API gateway pattern
- Publish/Subscribe pattern
- Façade pattern

Data management

- We opted for a **centralized approach** since our **data model** is very small and interconnected and we do not need different types of data models (eg. SQL and noSQL).
- We chose a **SQL approach** because it offers an easy and **standardized language** for queries, and grants **ACID** properties.



Algorithmic decisions

- All **algorithms** needed in the project are trivial but the one dealing with uniform repartition of cars in the city.
- This problem has been studied a lot and there are in literature various algorithms that solve it.
- They are mainly based on **mixed integer linear programming** techniques and in particular [1] presented a complete **model**. In [2] is presented a **greedy algorithm** that achieves almost the same result.

The mixed integer linear programming formulation for the problem is:

$$\text{Min} \quad Z = c_{ij}\left(\sum_{(i_t, j_{t+t_{ij}})\in A_3}\sum_{k\in K} u^k_{i_t j_{t+t_{ij}}} + \sum_{(i_t, j_{t+t_{ij}})\in A_4}\sum_{k\in K} v^k_{i_t j_{t+t_{ij}}}\right) + c_x\sum_{k\in L} x^k + c_d\sum_{i_t\in V} d^r_{i_t} + c_s\sum_{i_t\in V} s^r_{i_t} \qquad (1)$$

subject to

$$\sum_{i\in N} y^k_{i_1 i_2} + \sum_{i\in N} z^k_{i_1 i_{1+t_m}} + \sum_{\substack{i,j\in N\\ i\neq j}} u^k_{i_1 j_{1+t_{ij}}} + \sum_{\substack{i,j\in N\\ i\neq j}} v^k_{i_1 j_{1+t_{ij}}} = x^k \qquad \forall k \in \boldsymbol{L} \qquad (2)$$

$$y^k_{i_{t-1} i_t} + \sum_{(i_{t-t_m}, i_t)\in A_2} z^k_{i_{t-t_m} i_t} + \sum_{(j_{t-t_{ji}}, i_t)\in A_3} u^k_{j_{t-t_{ji}} i_t} + \sum_{(j_{t-t_{ji}}, i_t)\in A_4} v^k_{j_{t-t_{ji}} i_t} - y^k_{i_t i_{t+1}} - z^k_{i_t i_{t+t_m}} - \sum_{(i_t, j_{t+t_{ij}})\in A_3} u^k_{i_t j_{t+t_{ij}}} - \sum_{(i_t, j_{t+t_{ij}})\in A_4} v^k_{i_t j_{t+t_{ij}}} = 0 \qquad \forall i_t \in \boldsymbol{V}, k \in \boldsymbol{L}, t > 1 \qquad (3)$$

$$r_{i_t} = r_{i_{t-1}} + \sum_{(j_{t-t_{ji}}, i_t)\in A_4}\sum_{k\in L} v^k_{j_{t-t_{ji}} i_t} - \sum_{(i_t, j_{t+t_{ij}})\in A_4}\sum_{k\in L} v^k_{i_t j_{t+t_{ij}}} + \sum_{\substack{(i_{t-t_m}, i_t)\in A_2\\ k\in L}} z^k_{i_{t-t_m} i_t} + (s_{i_t} - s^r_{i_t}) - (d_{i_t} - d^r_{i_t}) - m_{i_t} \qquad \forall i_t \in \boldsymbol{V} \qquad (4)$$

$$\bar{r}_{i_t} = \bar{r}_{i_{t-1}} - \sum_{k\in L} z^k_{i_t i_{t+t_m}} + m_{i_t} \qquad \forall i_t \in \boldsymbol{V} \qquad (5)$$

$$r_{i_t} + \bar{r}_{i_t} \leq p_i \qquad \forall i_t \in \boldsymbol{V} \qquad (6)$$

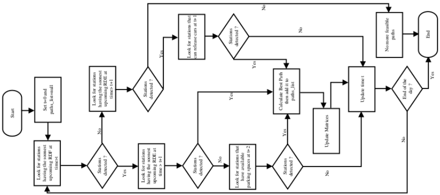
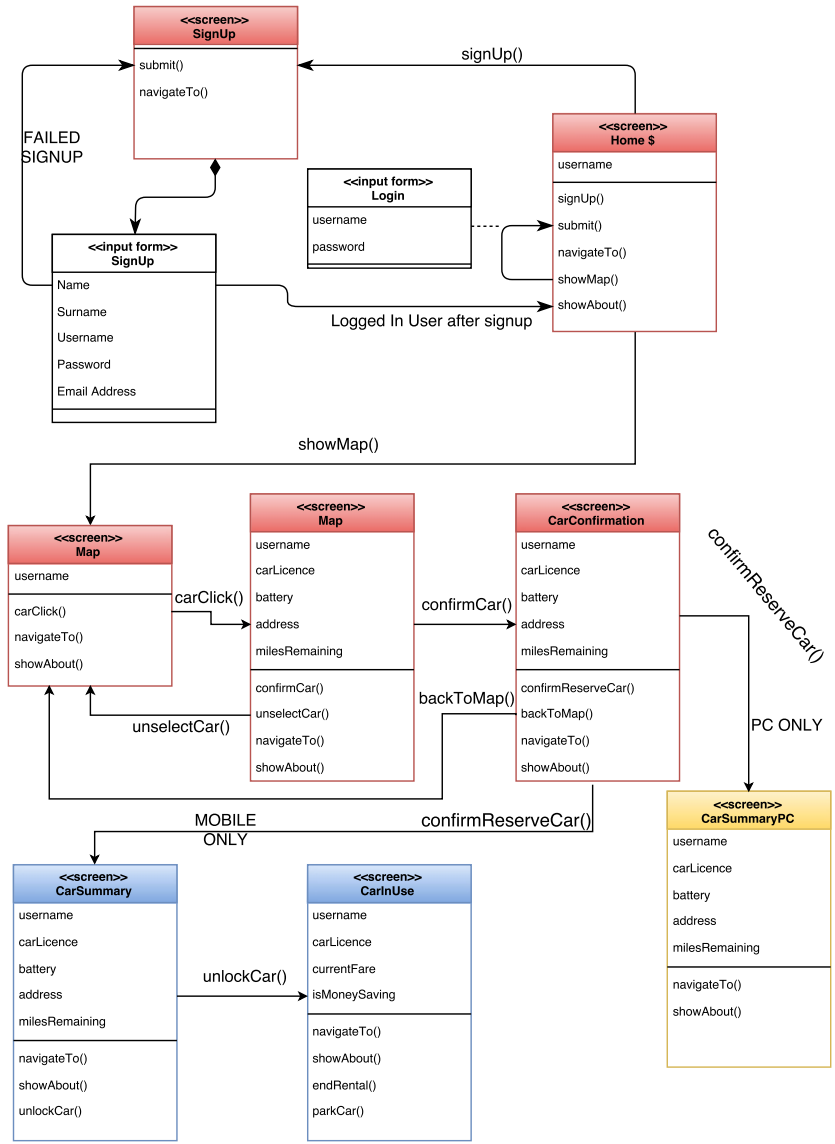


Figure 5 Flowchart of the Relocation Algorithm

User Experience Diagram

This Diagram shows the main interactions between the "Screens" that compose both the Web and the Mobile App.
The main actions are:

- signUp (User registration)
- showMap (shows available cars and allows to reserve one)
- ConfirmReserveCar (confirms the reservation and shows a summary with the car's position)



Sample Mockups

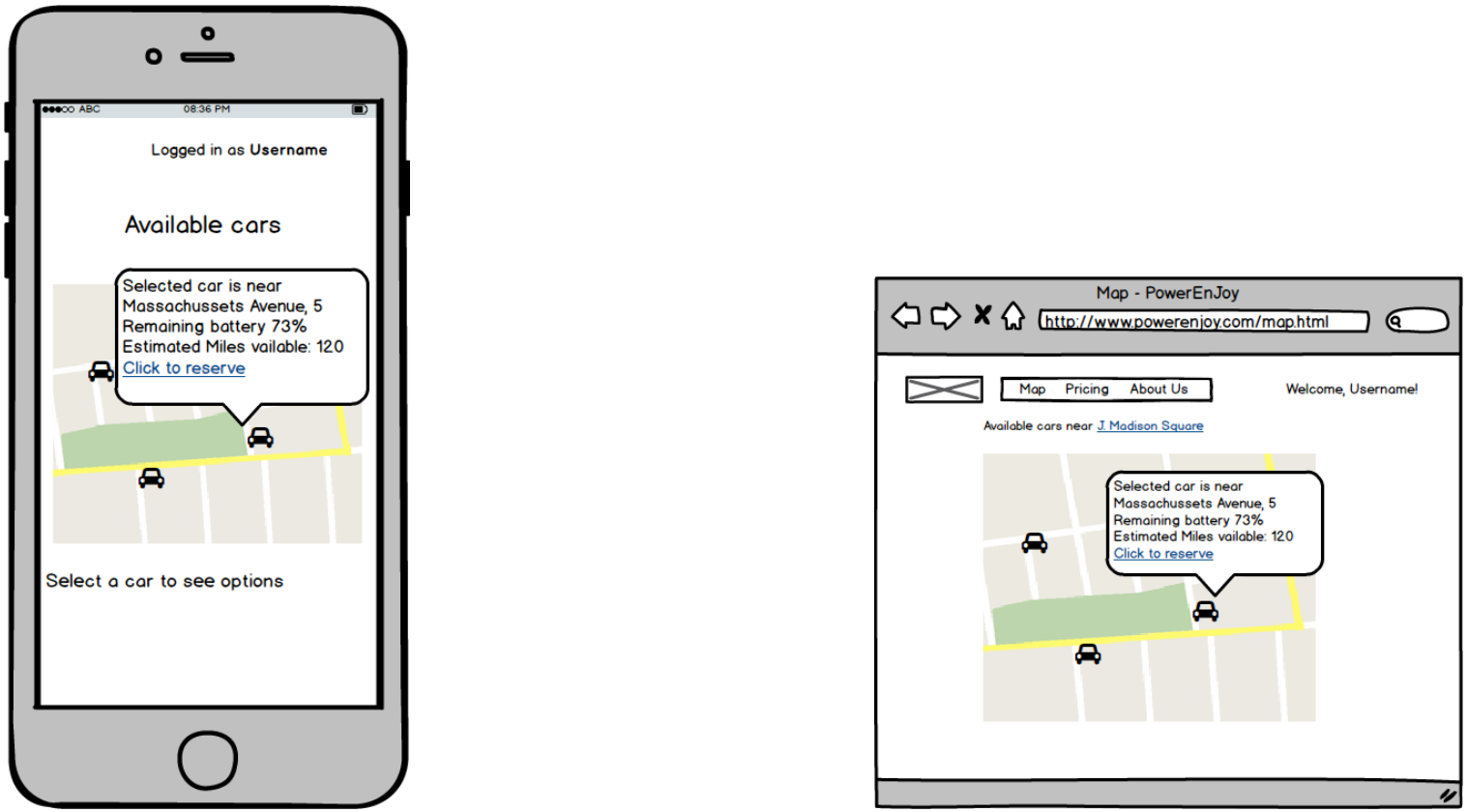


Figure: Selected Car - Mobile and Web mockup

Requirements traceability

1. Allow visitors to sign up.
2. Allow visitors to log in.
3. Allow Users and Active Users to update or modify their profile's information.
4. Show updated information on available cars.
5. Allow Active Users to reserve a car.
6. Allow Active Users to unlock the car reserved
7. Compute the fare.
8. Allow System Administrator(s) to update system's information.
9. Ensure that the fare is paid.
10. Allow the driver to choose the money saving option and get near their destionation.
11. Allow the user to park the rented car in safe zone.