



**POLITECNICO  
DI TORINO**

Tesi di Laurea Magistrale in Ingegneria Informatica

# **Autoconfigurazione di un veicolo in ambito Car Sharing**



**Relatore** Fulvio Corno

**Tutor aziendale** Stefania Basciu

**Candidato** Antonio Risoli

# Introduzione

## Internet of Things

L'IoT è una rete di dispositivi connessi che si scambiano informazioni con l'obiettivo finale di semplificarci la vita quotidiana. Si prevede che nel 2020 ci saranno circa 80 miliardi di dispositivi connessi.



## Veicolo connesso

I produttori di automobili iniziano a collegare i veicoli alla rete internet, il veicolo diventa un nodo di questa enorme rete. I paradigmi del V2V, V2I, V2P stanno per essere integrati nella maggior parte dei veicoli. Per il 2021 si prevedono 122,6 miliardi di € di investimenti rispetto agli attuali 40,3 miliardi nel settore automobilistico per ciò che concerne il concetto di veicolo connesso.

2016 40,3 miliardi di €

2021 122,6 miliardi di €



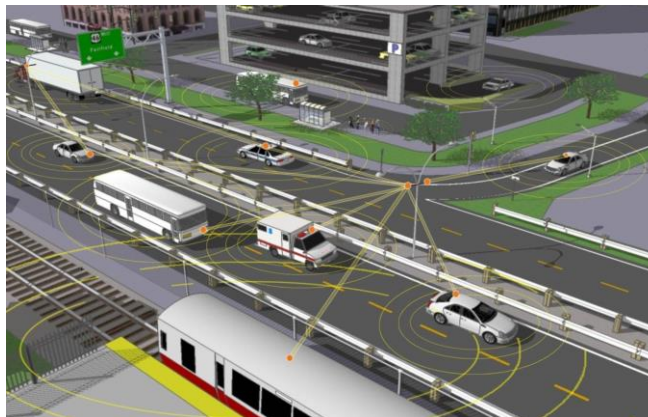
# Car Sharing

Nasce come noleggio breve delle autovetture. Oggi è possibile utilizzare tale servizio in modo semplice e immediato tramite smartphone.

La componente abilitante è una Telematic Box (TBox) che è il punto di giunzione tra la rete internet e la parte interna del veicolo. Tale centralina è anche deputata anche ad integrare le funzionalità richieste dal V2V, V2P e V2I.

La TBox è la centralina che conetterà il veicolo al resto del mondo IoT!

La TBox oggi viene sfruttata solo per la geolocalizzazione del veicolo e lo sbloccaggio da remoto dello stesso (Keyless System).



# Obiettivo della tesi

L'obiettivo della tesi è stato la realizzazione di un **Proof of Concept** (POC) per l'autoconfigurazione di un veicolo in ambito Car Sharing. Il POC deve

- ✓ Reimplementare lo sblocco di un veicolo da remoto senza l'utilizzo di una chiave fisica, ormai standard, ma attraverso un'*autenticazione biometrica* dell'utente.
- ✓ Nel contempo, all'atto dell'apertura del veicolo, il POC dovrà sperimentare la nuova funzionalità del configurare le varie componenti interne (per esempio infotainment, sedili, sterzo, specchietti ecc...), dotate di centralina elettronica, in base alle preferenze dell'utente corrente al fine di farlo sentire a suo agio.
- ✓ Alla chiusura del veicolo il POC deve salvare le configurazioni di ciascuna centralina in modo da poterle replicare su qualsiasi altro veicolo della flotta del servizio di Car Sharing.



# Il Sistema

# Analisi delle funzionalità

## L'infrastruttura informatica

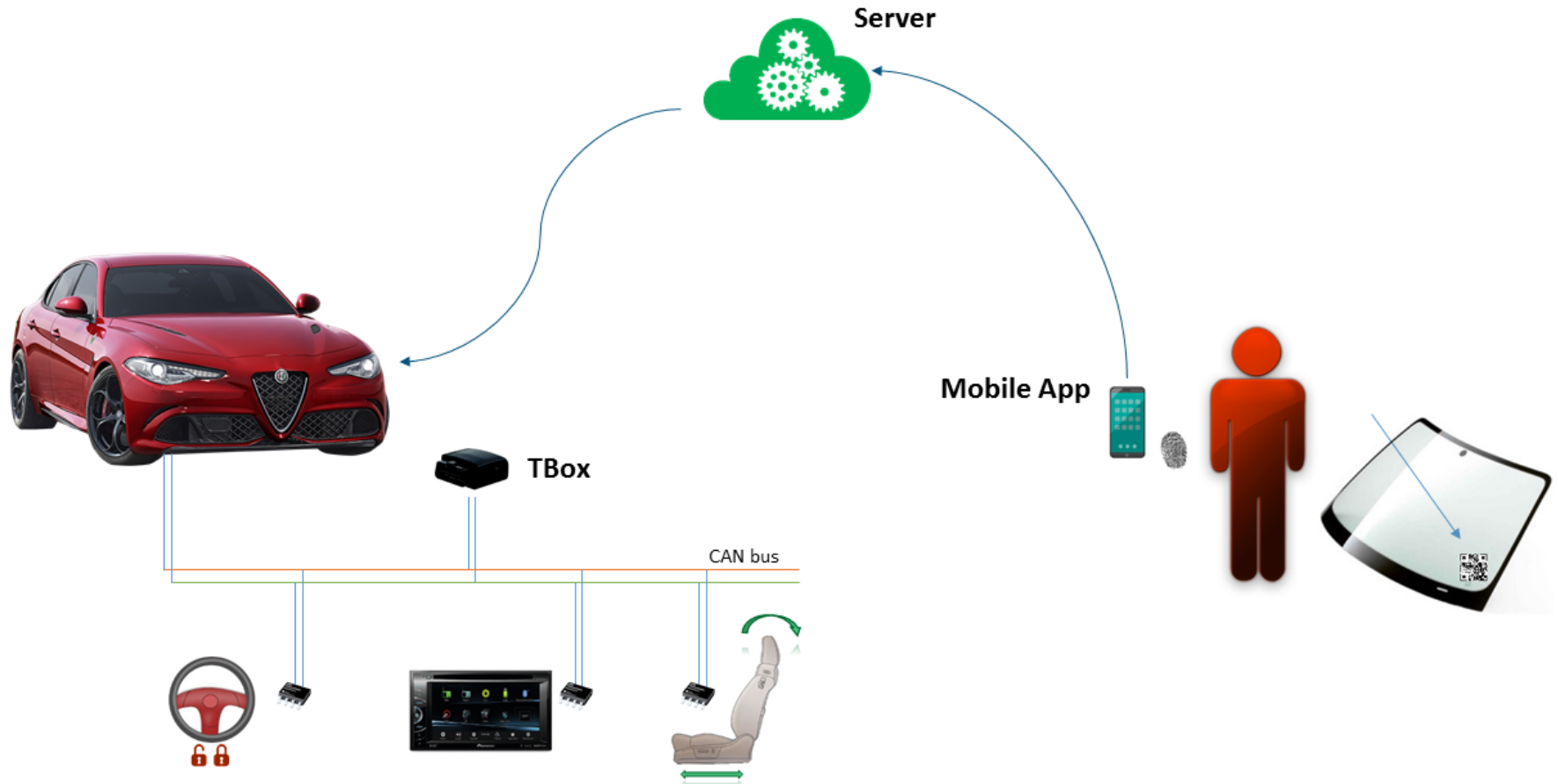
- ✓ Deve consentire la registrazione e il login di un utente
- ✓ Deve autenticare l'utente tramite biometria
- ✓ Deve consentire all'utente la visualizzazione dei veicoli a lui vicini
- ✓ Deve poter conoscere la posizione GPS degli utenti e dei veicoli
- ✓ Deve poter comunicare con il veicolo
- ✓ Deve poter conoscere lo stato del veicolo (libero, occupato)
- ✓ Deve poter sbloccare un veicolo
- ✓ Deve sbloccare il veicolo solo se l'utente è in prossimità dello stesso
- ✓ Deve poter salvare la configurazione di un veicolo relativa ad un utente

## Il veicolo

- ✓ Deve poter essere sbloccato da remoto
- ✓ Deve poter ricevere e applicare le configurazioni sulle ECU
- ✓ Deve poter leggere dalle ECU le configurazioni e salvarle
- ✓ Deve poter conoscere e inviare la sua posizione GPS
- ✓ Deve poter accorgersi della sua chiusura



# L'architettura



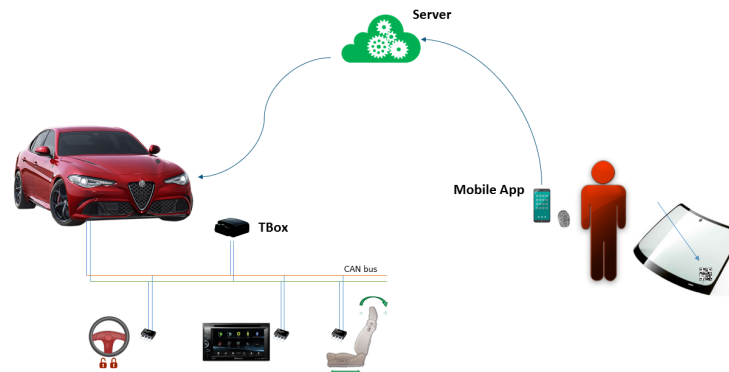


Attraverso la Mobile App l'utente può

- Registrarsi al sistema di Car Sharing
- Visualizzare i veicoli a lui vicini e chiederne lo sblocco al server

Il Server riceve la richiesta di sblocco, controlla che l'utente sia a meno di 3 m dal veicolo scelto e se è libero invia la richiesta di sblocco e le configurazioni.

La TBox ricevendo il comando di sblocco dal server sblocca il veicolo aprendo lo sportello lato conducente e inviando un comando di unlock alla ECU del bloccasterzo elettronico (ESL).

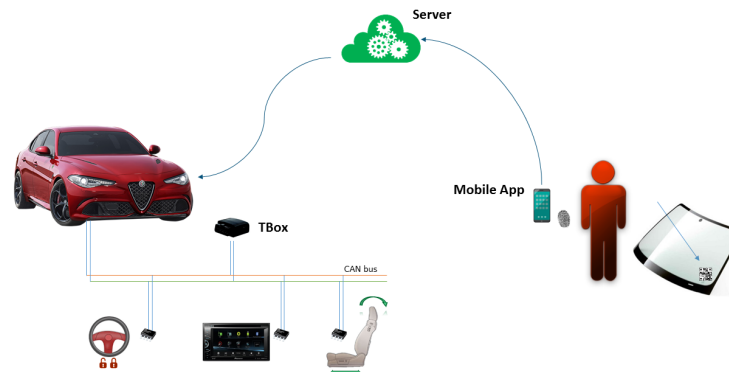


Configura le varie parti del veicolo rispetto all'utente corrente impostando i parametri di configurazione ricevuti dal server.

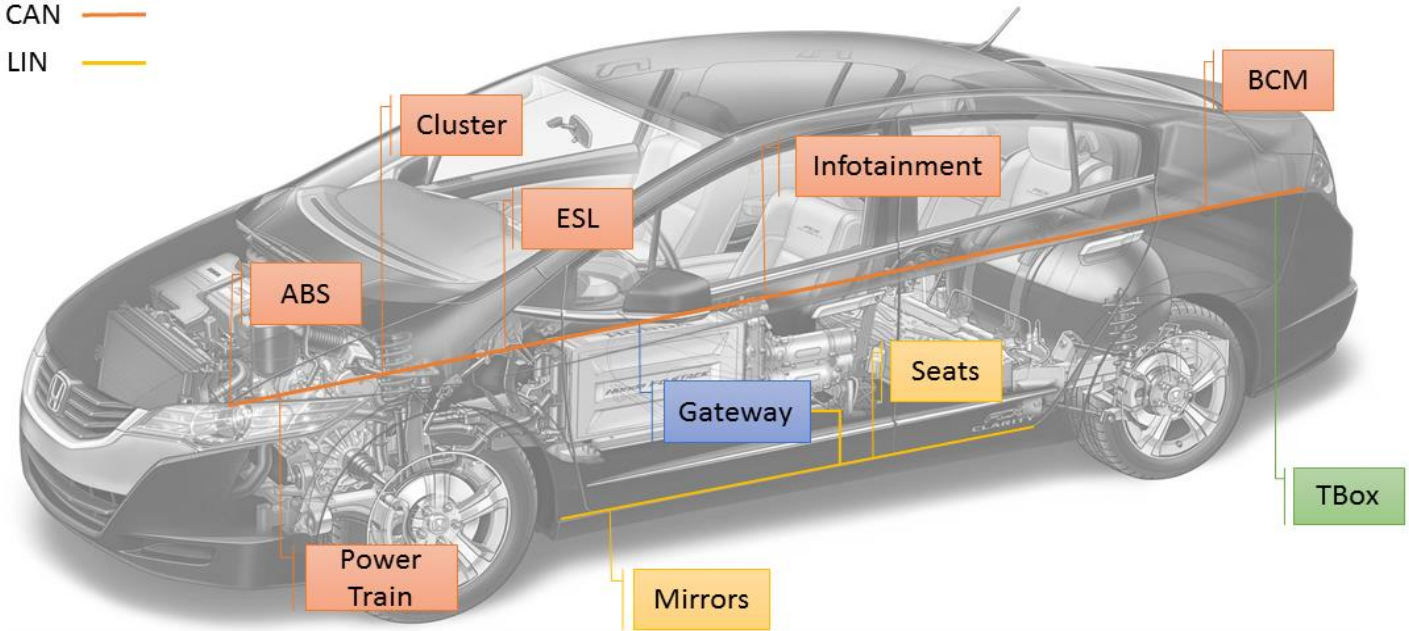
La TBox si accorge della chiusura del veicolo tramite la transizione del segnale fisico di chiave dalla posizione ON alla posizione OFF

A questo punto blocca il veicolo inviando un comando di lock all'ESL.

Estrae lo stato di ciascuna ECU configurata precedentemente e li invia al server



# Veicolo reale

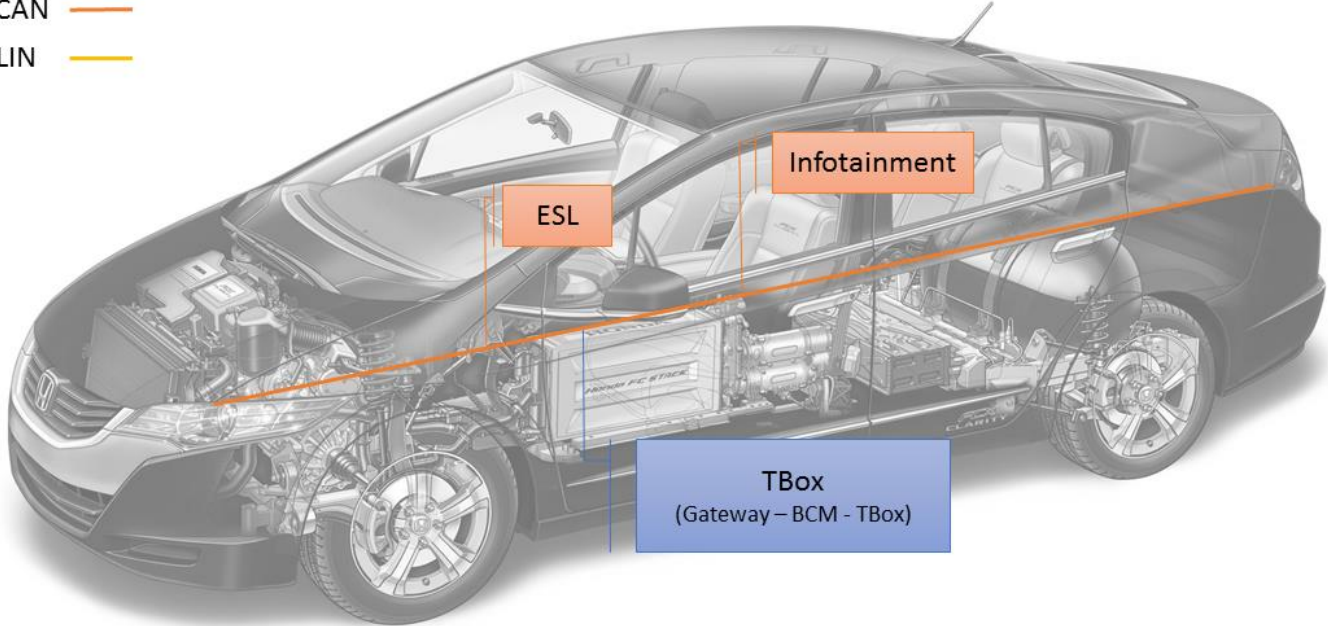


- TBox** Telematic Box
- ABS** Anti-lock Braking System
- BCM** Body Control Module
- ESL** Electronic Steering Lock



# Veicolo emulato

CAN —  
LIN —



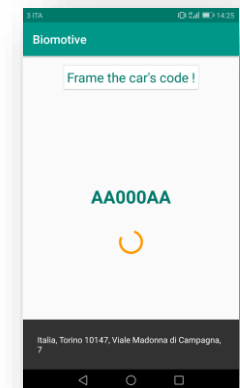
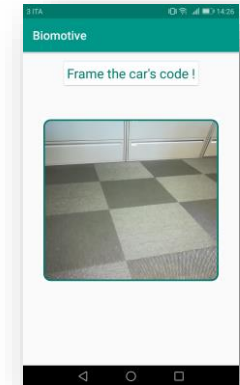
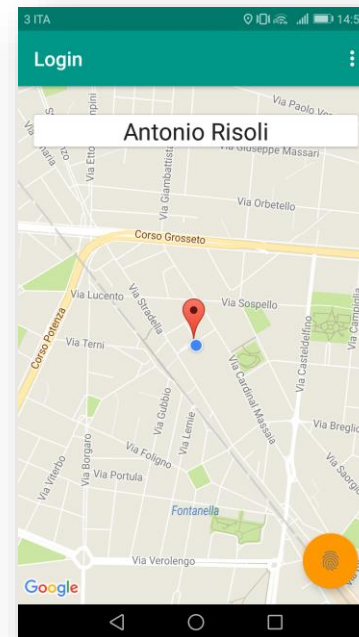
**TBox** Telematic Box  
**ESL** Electronic Steering Lock



# Il Prototipo

# Mobile Application

- ✓ Target 18-55 anni
- ✓ Android Sdk API  $\geq$  23
- ✓ Autenticazione biometrica *locale* tramite fingerprint
- ✓ KeyStore e AES-256\_CBC\_PKCS7
- ✓ SupportMapFragment
- ✓ Lettore QR code, Geolocalizzazione,
- ✓ rilevamento Bluetooth vicini
- ✓ Connessioni HTTPS con il server



# Server

Verso la Mobile App

- ✓ API REST HTTPS
- ✓ Multiprocess, Multithread
- ✓ Notifiche push



Verso la TBox

- ✓ Socket stream con wrapping in un contesto SSL
- ✓ Invia e riceve le configurazioni dalle TBox

Utilizza come database relazionale per la persistenza dei dati il software MariaDB.



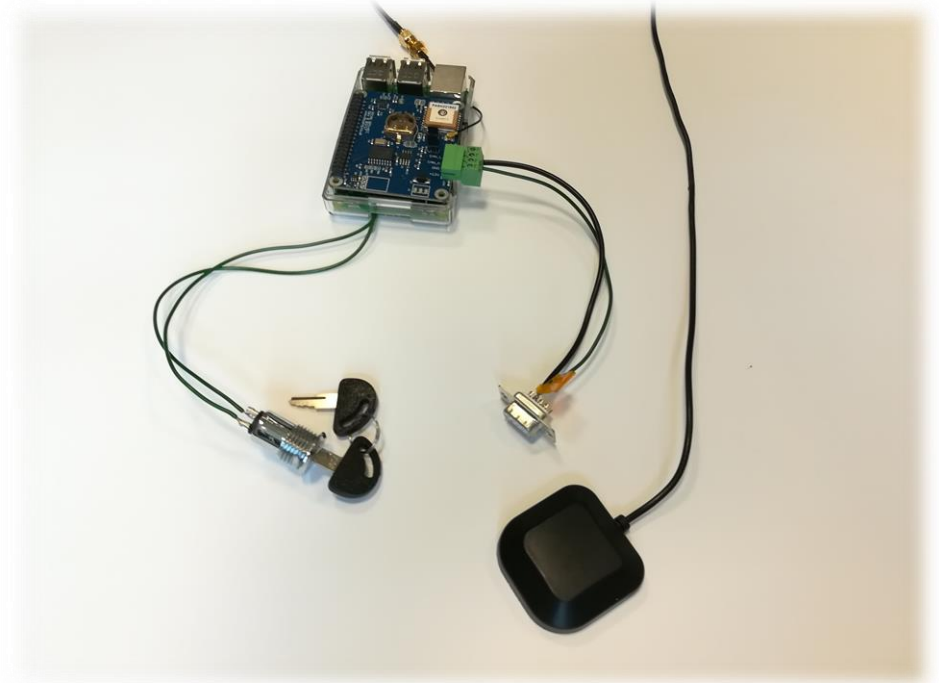
# TBox

## Hardware

- ✓ Raspberry Pi 3
- ✓ PiCAN GPS and Accelerometer
- ✓ Chiave fisica

## Software

- ✓ Raspbian
- ✓ Python, Python-can



Il modulo che implementa la comunicazione

con le altre ECU a bordo del veicolo attraverso il protocollo CAN.

Gestisce una moltitudine di thread ognuno deputato alla gestione di un messaggio CAN.

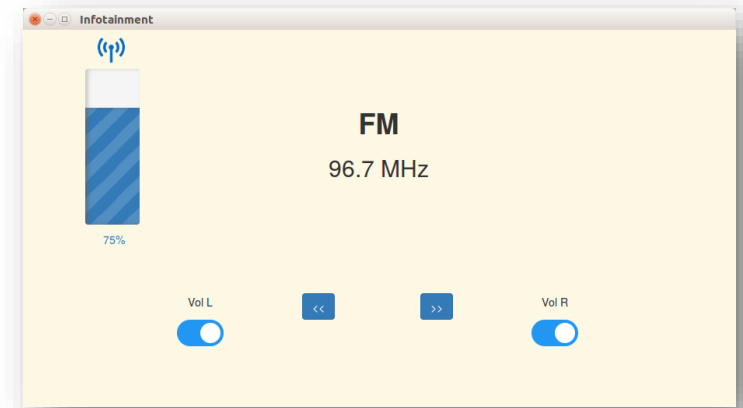
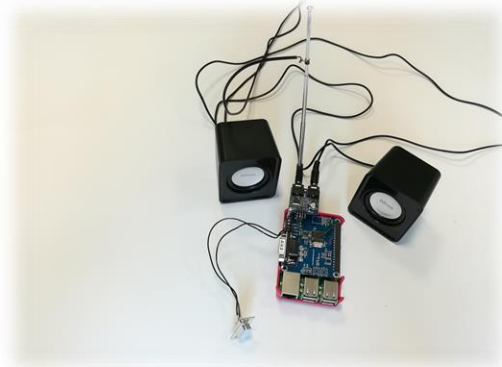




# Infotainment

## Hardware

- ✓ Raspberry Pi 3
- ✓ PiCAN 2
- ✓ TEA 5767



Il modulo che implementa l'Infotainment.

È pilotato tramite bus CAN. Tramite il messaggio CAN RadioCmd è acceso, configurato e spento da parte della TBox.

Ciclicamente emette il messaggio CAN RadioSts in cui comunica alle altre ECU la frequenza FM attuale, lo stato del canale audio destro (aperto/chiuso) e lo stato del canale audio sinistro.

Utilizza html.py per l'interfaccia grafica.



# Conclusioni

# Obiettivo raggiunto!

Il prototipo dimostra che l'obiettivo della tesi è stato raggiunto

- ✓ L'utente sblocca il veicolo tramite un'autenticazione biometrica
- ✓ Il veicolo è configurato rispetto alle preferenze dell'utente
- ✓ Le configurazioni sono salvate in remoto e replicate ad un accesso successivo



## Sviluppi futuri

- ✓ Ampliare il numero di ECU supportate
- ✓ Con più sensori a disposizione si potranno applicare algoritmi di Machine Learning per anticipare il conducente
- ✓ Tracciando i percorsi effettuati dal conducente si potrà agire sulla temperatura all'interno del veicolo al fine di ridurre lo sbalzo termico in prossimità della destinazione





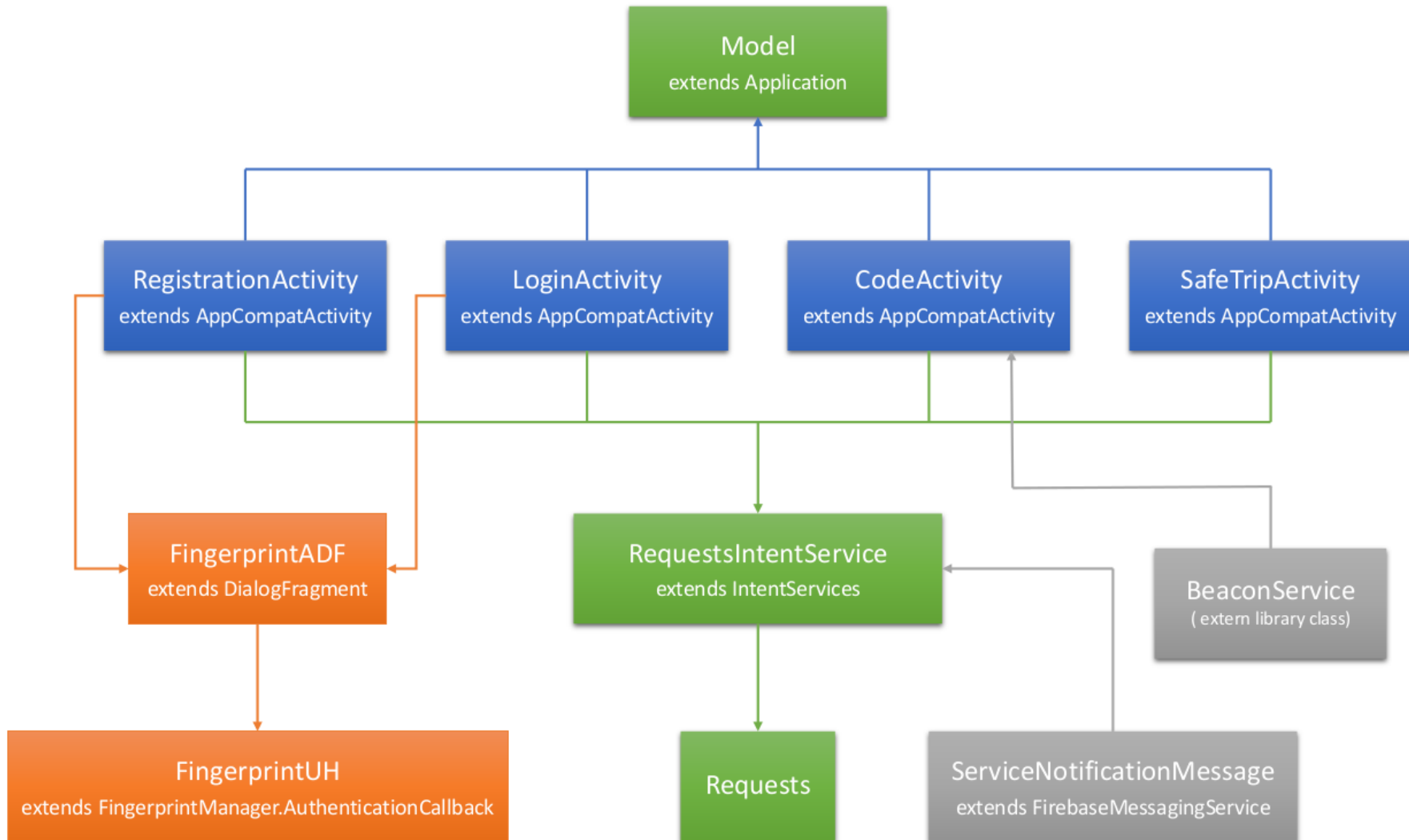
**POLITECNICO  
DI TORINO**

Grazie per l'attenzione !

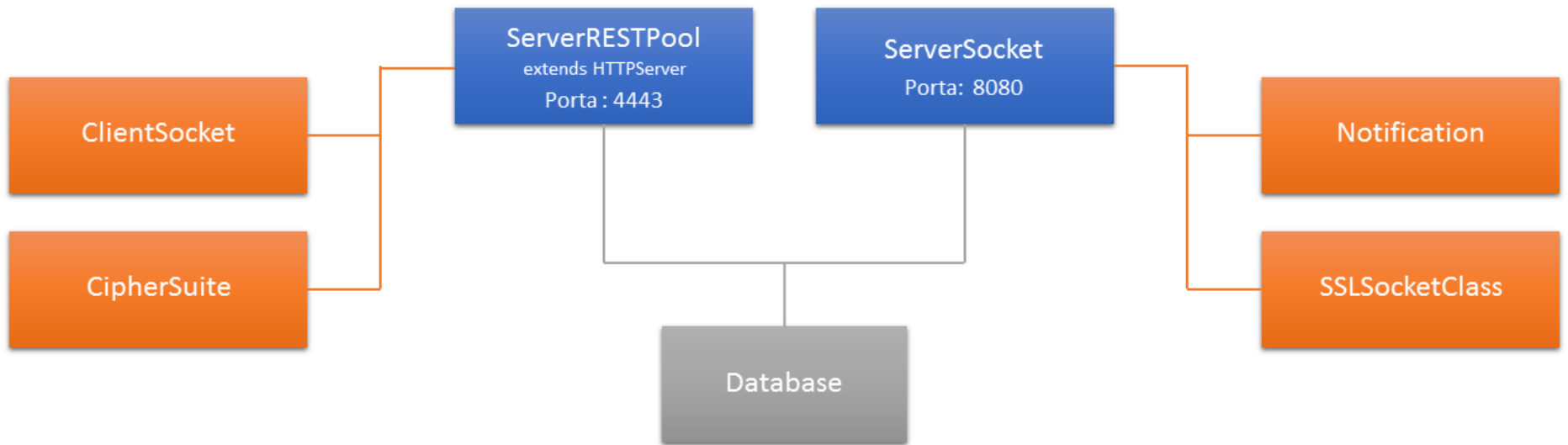
A decorative graphic at the bottom of the slide consists of several overlapping, wavy, ribbon-like shapes in shades of red and grey, creating a sense of motion and depth.

 **Reply**  
concept

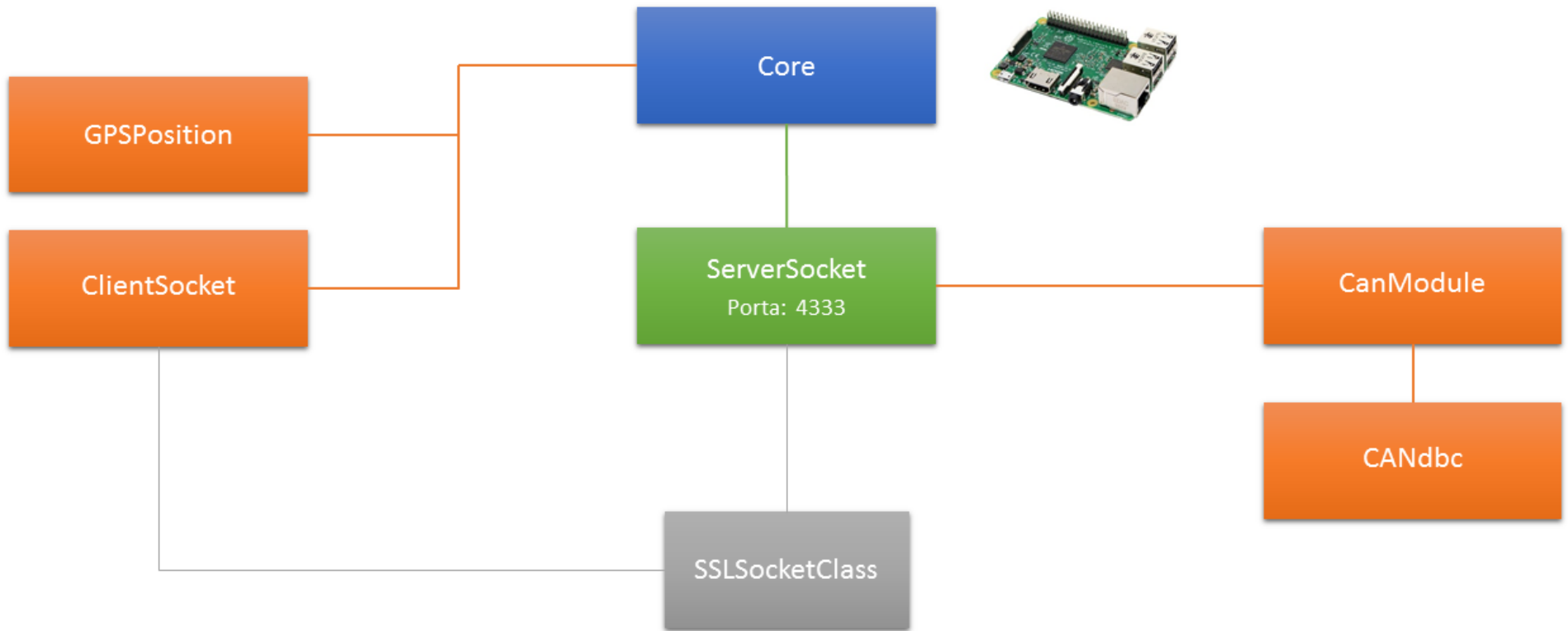
# Mobile App



# Server



# TBox





# Tabelle SQL

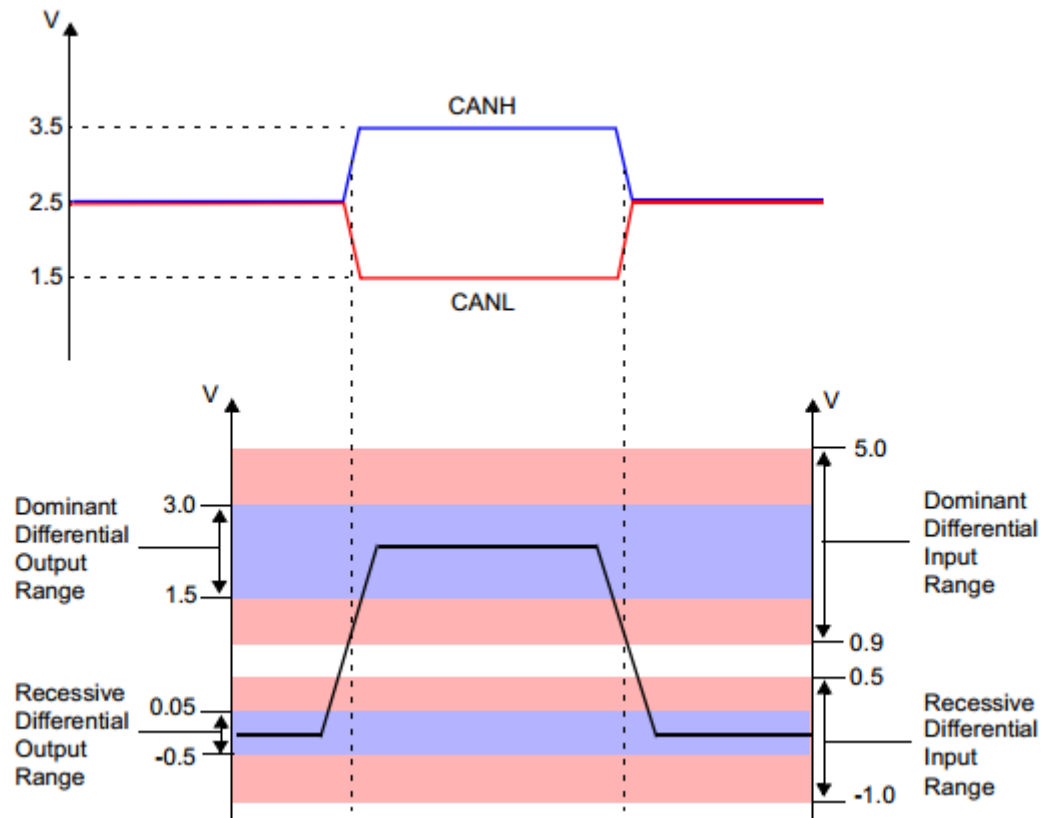
- ✓ Users(username, salt, secret, recovery);
- ✓ Cars(id, isUsedNow, IP, lat, lon, precision, blue\_id);
- ✓ Configurations(username, f, vr, vl);
- ✓ Token(username, token, time);

```
CREATE EVENT delete_token_event
  ON SCHEDULE EVERY 2 MINUTE DO
  DELETE FROM Token WHERE time < DATE_SUB(NOW(),INTERVAL 1 MINUTE);
```

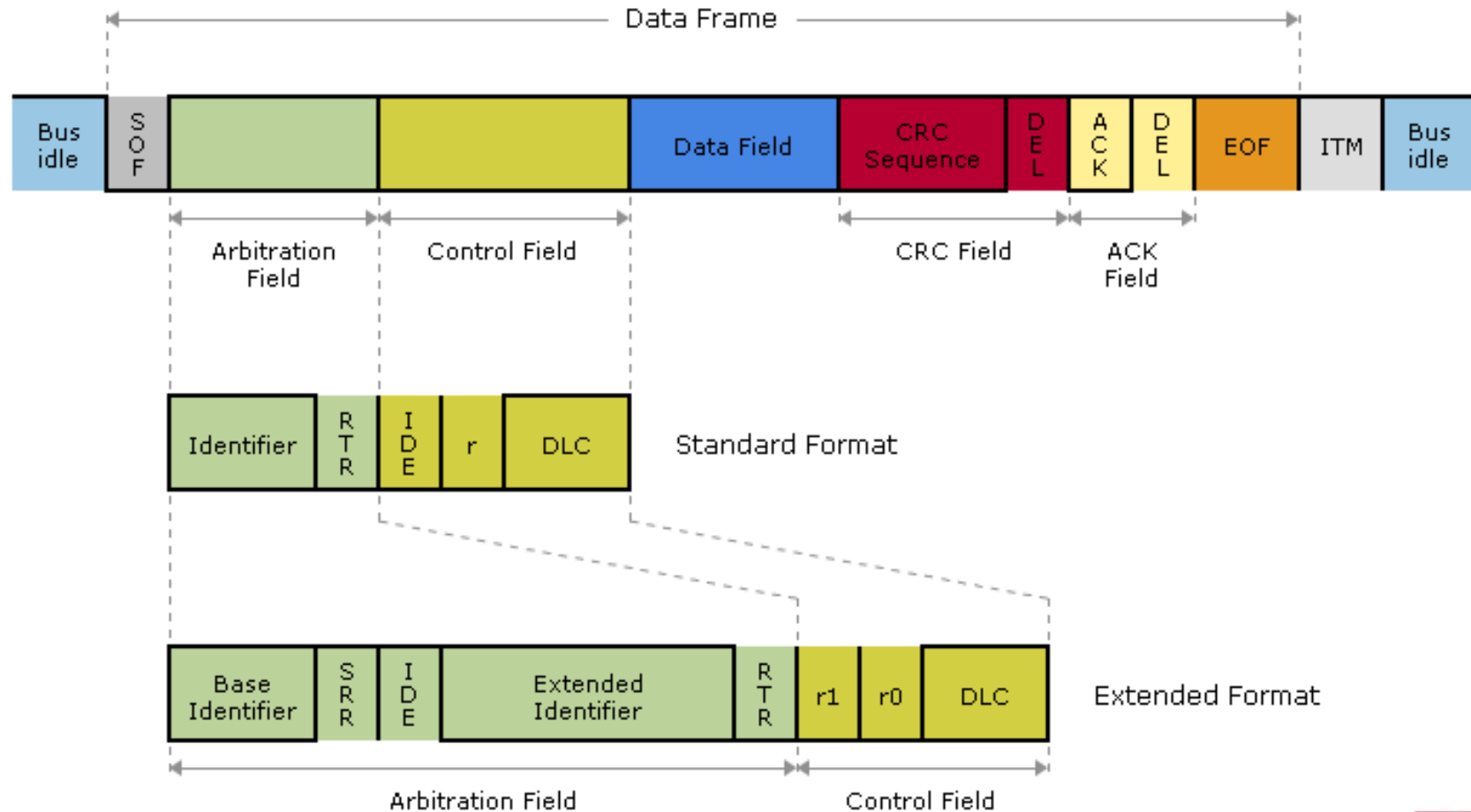


# CAN

- Broadcast, seriale, multi-master, due fili CAN-H e CAN-L
- Max 1 Mbit/s (  $l < 40$  m )
- Due livelli, Dominante ( 0 Dbit ) , Recessivo ( 1 Rbit )



# CAN



# LIN

- Broadcast, seriale, single-master multi-slave, un filo
- Max 20 Kbit/s
- Max 15 slaves
- Data Length 2,4,8 byte

