

# Autoconfigurazione di un veicolo in ambito Car Sharing

**Candidato:** Risoli Antonio

**Relatori:** Fulvio Corno, Stefania Basciu

## Introduzione

Al giorno d'oggi l'Internet of Things (IoT) sta pervadendo ogni settore industriale e civile. A fianco agli ormai noti filoni della Smart Home, Smart City, Smart Grid, dell'IoT, si sta iniziando a sviluppare il concetto del **veicolo connesso**.

Integrando all'interno del veicolo una centralina elettronica adibita alla comunicazione via internet con il mondo esterno (Telematic Box), questi diventa in grado di interfacciarsi con altri veicoli (V2V), con l'infrastruttura stradale (V2I) e i pedoni (V2P).

I veicoli della flotta di una compagnia di Car Sharing hanno a bordo una centralina di questo tipo che consente di sbloccare la macchina da remoto senza l'utilizzo di una chiave (Keyless System), di tracciarne la posizione e le condizioni interne e permettono la consuntivazione dell'utilizzo da parte dell'utente.

Le funzionalità implementate in queste centraline (TBox) sono ad oggi limitate. Nulla vieta di pensare di poter migliorare l'esperienza di guida degli utenti del Car Sharing tracciandone ad esempio le abitudini e attraverso algoritmi di Machine Learning prevederne il comportamento in base a determinate condizioni quali temperatura, umidità, tono della voce, pioggia. Si potrebbe ad esempio attivare in automatico il condizionatore ad una data configurazione di temperatura e umidità, si potrebbero chiudere i finestrini in caso di pioggia, riprodurre una canzone in base all'umore dell'utente, ottimizzare i parametri del motore in base allo stile di guida e al percorso che l'utente sta percorrendo ecc... Si svilupperebbe così un **veicolo intelligente**.

Il lavoro di tesi è stato svolto presso l'azienda **Concept Reply**.

## 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. Si è reimplementato 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. Altresì 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.

## Analisi delle funzionalità richieste

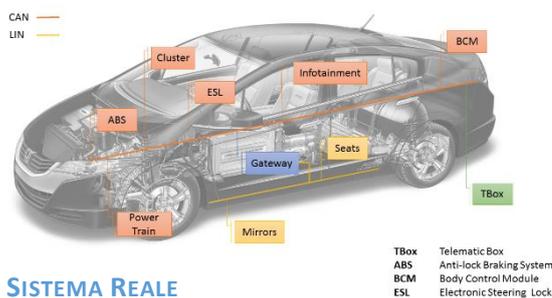
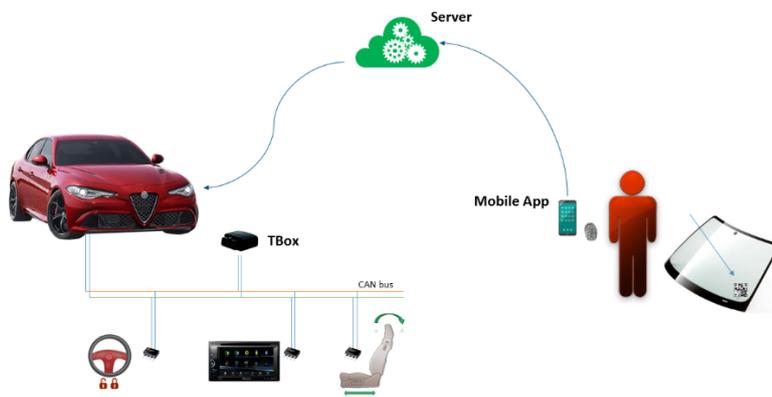
Una società di Car Sharing deve poter installare all'interno dei suoi veicoli una TBox capace di comunicare con le ECU presenti a bordo veicolo e in remoto sulla rete internet. Le ECU devono poter emettere su bus CAN il loro stato interno (es. frequenza FM, posizione del sedile, ecc...) e poter rispondere a comandi esterni per variare il loro stato. Nel momento in cui l'utente modifica la stazione radio o la posizione del sedile, il sistema deve rilevarlo e comunicare in remoto tale cambiamento. Si necessita di geolocalizzare il veicolo, sbloccare da remoto il veicolo (apertura porte e Electronic Steering Lock (ESL) unlock), bloccarlo in automatico (ESL lock) e **configurarlo** rispetto all'utente corrente.

La stessa società di Car Sharing ha già un'infrastruttura informatica che le consente da un lato di geolocalizzare i veicoli e comunicare con essi e dall'altro di gestire le richieste degli utenti. Lo sblocco deve essere possibile solo se l'utente è nelle immediate vicinanze del veicolo.

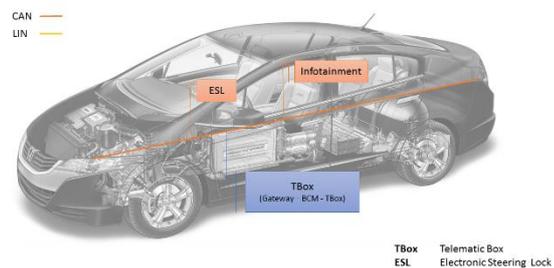
## L'architettura del sistema

Analizzando le funzionalità ho progettato un'architettura basata su tre elementi principali:

- Una **Mobile Application** che consenta all'utente di visualizzare i veicoli a lui vicini e gli permetta di sbloccare uno della flotta.
- Il **Server** che riceve le richieste dall'utente e che, nel caso in cui una di queste sia di sblocco di un veicolo, provvede a verificare le condizioni per le quali lo stesso può essere sbloccato (veicolo libero, posizioni GPS di utente e veicolo compatibili, distanza dell'utente dal veicolo inferiore ai 3 m). Se tali condizioni sono tutte soddisfatte, il server, procede allo sblocco del veicolo e all'invio delle configurazioni per l'utente corrente.
- La **TBox** che ricevendo le configurazioni (stati delle ECU), procede innanzitutto a sbloccare il veicolo e poi ad utilizzare le informazioni ricevute per modificare lo stato corrente delle centraline. Alla chiusura del veicolo (Key ON → Key OFF transition), essa provvede a bloccarlo.



SISTEMA REALE



SISTEMA EMULATO

Poiché non è stato possibile utilizzare un veicolo reale per effettuare l'autoconfigurazione, si è provveduto ad emularlo restringendone le componenti a TBox, ESL e Infotainment. In questo caso la TBox svolge anche le funzionalità del Body Control Module (BCM) e del Gateway.

## Implementazione

### Mobile Application

L'applicazione è stata realizzata per dispositivi Android (min. Sdk API 23). Consente all'utente di registrarsi sul server e di effettuare il login tramite autenticazione biometrica (Fingerprint). In particolare la password utilizzata per la registrazione è salvata sullo smartphone (nelle *SharedPreferences*) cifrata con AES-256\_CBC\_PKCS7. La chiave di cifratura utilizzata è generata in automatico dal *KeyStore* di Android e ivi salvata. La chiave di cifratura è altresì utilizzabile solo previa autenticazione biometrica. Le richieste al server sono effettuate utilizzando un *IntentService*. I veicoli vicini sono mostrati tramite un *SupportMapFragment* che è aggiornato in modo asincrono tramite push notifications. Una volta effettuato il login l'utente può scegliere il veicolo da sbloccare inquadrandone il QR code presente sul parabrezza, tramite l'utilizzo dei componenti *CameraSource* e *Detector*. Viene dunque inviata una richiesta al server contenente la posizione GPS dell'utente, il valore alfanumerico del QR code (che è la targa del veicolo) e l'identificativo del beacon più vicino con la relativa distanza. In caso di richiesta accettata mostra una schermata con l'esito positivo. Consente anche di modificare la password di registrazione.

### Server

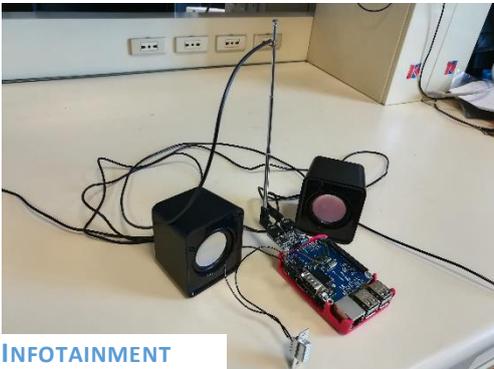
Il server è l'anello di congiunzione tra la Mobile Application e il veicolo. Il linguaggio utilizzato è stato Python e sfrutta *MariaDB* come database relazionale per la persistenza dei dati. Offre delle API REST HTTPS verso l'applicazione per le funzionalità di registrazione, login all'utente e richiesta di sblocco. Il server ha un'architettura multiprocess e multithread. Comunica con i veicoli (TBox) tramite socket stream SSL (client e server authentication), da cui riceve la loro posizione, lo stato delle centraline alla chiusura degli stessi veicoli e a cui invia la richiesta di sblocco. La richiesta di sblocco da parte dell'utente è accettata solo se l'identificativo del beacon inviato corrisponde a quello associato al veicolo. Utilizzando le API offerte da Google Firebase Cloud Messaging, il server, è in grado di inviare push notification alla Mobile Application.

## TBox

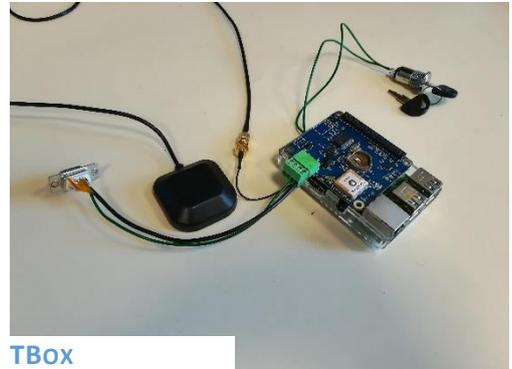
La TBox è stata realizzata tramite un Raspberry Pi 3 ed il modulo PiCAN GPS and Accelerometer. Il linguaggio utilizzato è stato Python. Grazie al PiCAN è stato possibile comunicare su bus CAN con le altre centraline. In particolare i moduli con cui comunica sono un ESL e un Infotainment. Il primo è un blocco sterzo elettronico che può essere in due posizioni lock o unlock. Alla richiesta di sblocco la TBox provvede ad inviare un messaggio CAN di unlock. L'Infotainment viene svegliato dalla TBox sempre tramite messaggi CAN e configurata rispetto all'utente corrente. Alla chiusura del veicolo, spegnimento del motore tramite chiave, la TBox invia la richiesta di lock all'ESL ed estraendo lo stato dell'infotainment (frequenza FM attuale, abilitazione canali destro e sinistro) lo invia al server.

## Infotainment

Poiché non è stato possibile avere un'infotainment reale l'ho realizzato mediante un'ulteriore Raspberry Pi 3 con i moduli PiCAN2 e TEA 5767 (radio FM). Il linguaggio utilizzato è sempre Python. Per l'interfaccia grafica ho utilizzato *html.py* che è basato su PySide (QT). Il modulo TEA 5767 è controllato tramite protocollo bus I<sup>2</sup>C. L'infotainment è acceso, configurato e spento attraverso il messaggio CAN RadioCmd. Attraverso il messaggio CAN RadioSts emette ciclicamente (periodo 1 s) il suo stato (frequenza FM, canale destro ON/OFF, canale sinistro ON/OFF).



INFOTAINMENT



TBOX

## Conclusioni

Al termine del lavoro sono stati effettuati dei test sulle funzionalità richieste in fase di analisi. Rispetto a tali test il POC realizzato soddisfa l'obiettivo iniziale della tesi. Si è dimostrato che è possibile interagire con un dispositivo reale a bordo veicolo (l'ESL) e autoconfigurare un Infotainment rispetto all'utente corrente, che seppur emulato, rispecchia le modalità di interazione di un Infotainment reale.

La TBox può essere inserita come ECU fisica a bordo veicolo (in questo caso bisognerebbe inserire un modulo GSM invece di sfruttare il WiFi del Raspberry, validarlo rispetto alle interferenze elettromagnetiche emesse e integrare un sistema di sleep/wakeup per ottenere uno stato di basso consumo) ma ciò richiederebbe molti accorgimenti.

Per una produzione in scala invece si dovrebbe sviluppare una board apposita e fare il porting del codice su quest'ultima. Il server e la Mobile Application sarebbero invece pienamente riutilizzabili e integrabili con i server preesistenti della compagnia di Car Sharing. A prescindere da come tale componente può essere integrata a bordo del veicolo, vi è una serie di possibili sviluppi futuri.

## Sviluppi futuri

Innanzitutto si può ampliare il numero di ECU che sono supportate. Tramite un convertitore CAN/LIN si potranno inserire specchietti e sedili in modo da poterli autoconfigurare. Nella previsione iniziale del progetto erano presenti anche tali dispositivi ma, per tempistiche derivanti dal fornimento degli stessi da altre aziende, non è stato possibile integrarli. Altresì la TBox potrebbe comunicare anche con il climatizzatore e automatizzarne l'accensione/spegnimento in base alle abitudini dell'utente. Conoscendo la destinazione dell'utente si potrebbe mitigare lo sbalzo termico in prossimità dell'arrivo.