Architecture of the Green Move System

Andrea G. Bianchessi, Gianpaolo Cugola, Simone Formentin, Angelo Morzenti, Carlo Ongini, Emanuele Panigati, Matteo Rossi, Fabio A. Schreiber, Sergio Matteo Savaresi, Letizia Tanca and Edoardo G. Vannutelli Depoli

Abstract In this chapter, we describe the architecture of the prototype of the Green Move system. The system includes three main components: the Green Move Center, which is the server side of the application; the mobile application that is installed on the smartphone of each user of the system; and the Green e-Box, which is a device installed on each Green Move vehicle that allows the system (including the user) to interact with the vehicle. In this chapter, we describe the key functions of each component and the main interactions that occur with one another. In particular, we focus on three aspects of the architecture: (i) the mechanisms through which the Green e-Box provides a uniform interface to interact with possibly very heterogeneous vehicles; (ii) the keyless approach through which users access vehicles; and (iii) the flexible real-time monitoring of the fleet achieved through Complex Event Processing technology.

This chapter includes material previously published in (Bianchessi et al. 2013b, 2014).

This work was carried out while these authors were at Department of Electronics Information and Bioengineering, Politecnico di Milano.

A.G. Bianchessi · G. Cugola · S. Formentin · A. Morzenti · C. Ongini · E. Panigati ·
M. Rossi (✉) · F.A. Schreiber · S.M. Savaresi · L. Tanca · E.G. Vannutelli Depoli
Department of Electronics Information and Bioengineering, Politecnico Di Milano,
Piazza Leonardo Da Vinci 32, 20133 Milan, Italy
e-mail: matteo.rossi@polimi.it

A.G. Bianchessi
e-mail: andrea.bianchessi@polimi.it

G. Cugola
e-mail: gianpaolo.cugola@polimi.it

S. Formentin
e-mail: simone.formentin@polimi.it

A. Morzenti
e-mail: angelo.morzenti@polimi.it

C. Ongini
e-mail: carlo.ongini@polimi.it

D.F. Bignami et al. (eds.), *Electric Vehicle Sharing Services for Smarter Cities*,
Research for Development, DOI 10.1007/978-3-319-61964-4_8

1 Introduction

As described in Chap. 3, new models of sustainable mobility would greatly benefit from the integration and sharing of vehicles that differ in type (cars, scooters, bicycles), technology (electric, hybrid or with classical combustion engines), and ownership (they can be publicly or privately owned, fully or partially shared) within the same system. Such systems should provide common functions, services, and interfaces both to end users and to system administrators, independently of the types of vehicles managed. In addition, the vehicle sharing service offered to users and built upon a heterogeneous fleet of vehicles should be easily accessible, and it should address the multiform needs of a wide range of customers (e.g., people reaching their workplace, company employees, families with children). Hence, the service should be highly configurable, to allow for the possibility of closely matching the user preferences with the services offered during the drive. This calls for innovative solutions in the design of a hardware/software architecture that manages the embedded control units that are onboard the vehicles and connects them to a back-end system that provides the mobility services to the final users, preferably through personal mobile devices, such as smartphones, tablets. In this chapter, we present the technological solutions adopted in the Green Move project to address the above issues.

The prototype platform developed in the project for the realization of electric vehicle sharing systems addresses the issues above in the following ways:

- It provides a hardware/software interface, the Green e-Box, which allows the system to interact with a heterogeneous fleet of vehicles in a uniform way, independently of the specificity of each vehicle.
- It relies on mobile devices to let users access and interact with the system—take possession/release a reserved vehicle, open/close its doors, enable/disable the drive.
- It uses standard protocols to allow external partner services to access information and operations (e.g., the reservation of vehicles) offered by the coordination center.

E. Panigati
e-mail: emanuele.panigati@polimi.it

F.A. Schreiber
e-mail: fabio.schreiber@polimi.it

S.M. Savaresi
e-mail: sergio.savaresi@polimi.it

L. Tanca
e-mail: letizia.tanca@polimi.it

E.G. Vannutelli Depoli
e-mail: edoardo.vannutelli@polimi.it

- It offers an infrastructure to customize the software configuration of vehicles by pushing new services on them and by removing existing ones, for example, depending on the user preferences.

In this chapter, we first give an overview of the overall architecture of the Green Move system prototype in Sect. 2. Then, in Sect. 3, we provide some details of the Green e-Box, the device interfacing the vehicles with the rest of the system, and we describe how this facilitates the interactions between vehicles and users on one side (Sect. 4), and between vehicles and management system on the other (Sect. 5). To highlight the benefits of the approach followed in the Green Move prototype system, in Sect. 6, we introduce some related services and platforms; finally, Sect. 7 concludes. Chapter 9 presents the mechanisms developed to obtain a high level of dynamism and customization in the software configuration of the vehicles.

2 Overall Architecture

The main elements of the Green Move platform are shown in Fig. 1. They are the *Green Move Center*, which coordinates the system, the *Green e-Boxes*, which allow vehicles to communicate and interact with the rest of the system, and the users' smartphones, on which the *Green Move client app* is installed.

The Green Move Center (GMC) coordinates the activities of the Green Move system and offers services such as user and vehicle registration, vehicle reservation/acquisition/release/monitoring. To retrieve and distribute information among the managed vehicles, the GMC includes the T-Rex Complex Event Processing (CEP) engine (Cugola and Margara 2012a; Cugola and Margara 2012b), a system explicitly conceived to efficiently collect, process, and analyze events.

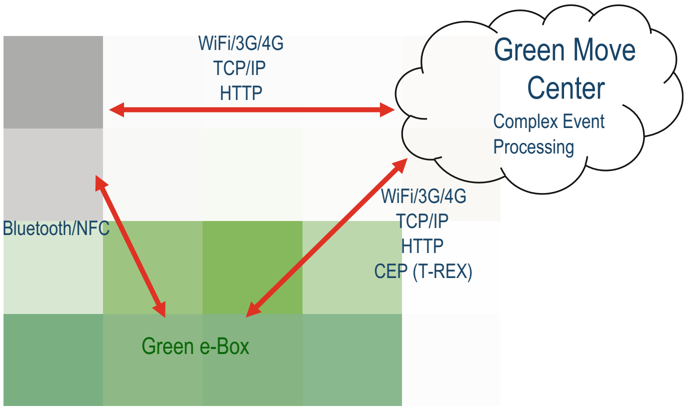

Fig. 1 Overview of the architecture of the prototype of the *Green Move* system

The Green Move platform uses CEP technology to both monitor the vehicles and send them configuration commands. The adoption of T-Rex provides several benefits: *efficiency*, as T-Rex is designed to perform the processing of events and to generate results with a low delay (typically, in the sub-millisecond range); *scalability*, since T-Rex is capable of providing low processing delay and hardware resource consumption even in the presence of a large number of sources, rules, and sinks; *modularity*, as T-Rex incorporates the management of events into a single software component, whose behavior is defined in a simple way through the set of deployed rules defining composite events.

Each Green Move vehicle is equipped with a device, the Green e-Box (GEB), through which it interacts with the overall system. From a technical point of view, the only requirement for a vehicle to be introduced in the Green Move system is to have a GEB; this ensures that the interaction between vehicle and system occurs according to standardized protocols. This allows for the possibility of adding to the system vehicles that are heterogeneous not only in their types, but also in their ownership (Alli et al. 2012). The GMC communicates with GEBs through mobile broadband channels to manage the fleet. The same channel is used by GEBs to send to the GMC vehicle data—which are distributed at regular intervals as T-Rex events—such as diagnostic information, usage statistics, and trip data (e.g., current GPS position, speed, state of charge). GEBs run the Android operating system, which executes the Green Move vehicle app. This app is the software interface between the vehicle and the rest of the system; in addition to executing core functions such as data retrieval and actuation of commands such as open/close doors, it provides hosting facilities for dynamic applications.

The interaction of Green Move users with the system occurs through their smartphones, on which the Green Move client app must be installed. The app communicates with the GMC through a WiFi or mobile broadband channel to reserve vehicles and to retrieve the electronic key that is necessary to access the vehicle. This key, whose contents are outlined in Sect. 4, is exchanged between the Green Move client app and the GEB through a Bluetooth or NFC channel; it is used to send commands from the user's smartphone to the vehicle, such as open/close the doors (if present), and enable/disable the drive. By using a direct (Bluetooth or NFC) link between the user's smartphone and the GEB, this communication can occur at any time, even when there is no data connection available between the GEB and the GMC (e.g., in an underground parking lot). The same connection can be used by the GEB to send data to the users' smartphones, such as customized advice of various kinds (e.g., commercial).

Every component of the Green Move system performs, to varying degrees, data management functions, which are pervasive throughout the platform. The description of these functions is the focus of Chap. 10.

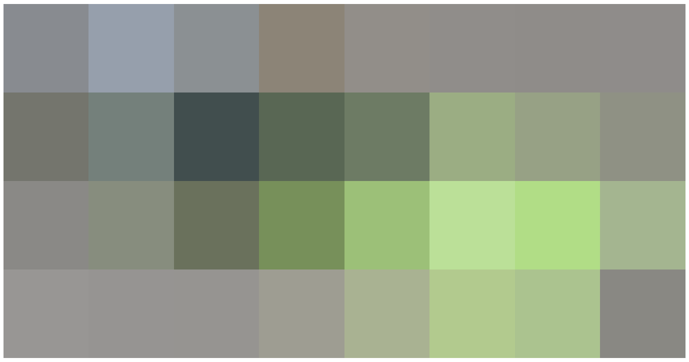

Fig. 2 Vehicles integrated into the prototype *Green Move* system, from left to right: Tazzari Zero Evo, Estrima Birò, Piaggio Liberty e-Mail

3 The Green E-Box

To be integrated into the Green Move system, a vehicle only needs to be equipped with a GEB, which allows it to communicate with other system elements (GMC, users' smartphones) in a manner that is independent of the specific type of vehicle. In fact, for demonstration purposes, the prototype of Green Move system has been realized for a small, but heterogeneous fleet of three vehicles with different characteristics. More precisely, the vehicles integrated into the prototype, shown in Fig. 2, are the following:

- *Tazzari Zero Evo*,[1] a two-seat electric car with a driving range of 140 km and a maximum speed of 100 km/h. The lithium-ion battery pack requires about 9 h for a full charge (0–100%). This vehicle is suitable for urban mobility and/or short-range interurban trips.
- *Estrima Birò*,[2] a two-seat electric vehicle with a maximum speed of 45 km/h and a range of about 50 km. The Pb-Gel battery pack takes about 9 h to be fully charged. Its extremely compact size makes it suitable for urban (or low-speed suburban) streets.
- *Piaggio Liberty e-Mail*,[3] an electric scooter with a top speed of 45 km/h and a range of 70 km. The lithium battery pack requires about 4 h for a full charge. The scooter is suitable for city driving, but, unlike the other two vehicles, it is not for all weather conditions.

[1] www.tazzari-zero.com.

[2] www.estrima.com.

[3] www.piaggio.com.

Fig. 3 Green e-Box architecture

The GEB is composed of a low-level embedded board and a high-level Android board (see Fig. 3). In order to have a constant monitoring of each vehicle, even when turned off and not in use, the GEB is directly connected to the permanent 12 V line of the vehicle. The GEB is also wired to the vehicle electronics, and it communicates with the GMC via a mobile broadband channel and with the users' smartphone via Bluetooth or NFC (Near Field Communication) links.

Figure 3 details the architecture of the GEB device, which was designed following the classic software engineering principles of modularity (separate components are responsible for different functions, e.g., retrieving data from the vehicle) and extensibility (e.g., new functions can be easily added that can exploit the data collected from the vehicle).

The embedded board acts as a hardware abstraction layer (HAL), and it is designed to abstract the vehicle-specific details; thus, it creates a virtual layer between software applications and the actual hardware, providing a general communication protocol to the high-level layers built on top of it. To achieve this, the embedded board has a CAN-bus to retrieve data directly from the vehicle ECU (engine control unit) and several analogic and digital input/output channels so that the GEB can be installed on a large variety of heterogeneous vehicles, even those without an ECU (e.g., Tazzari and Birò). A microcontroller handles each signal, acquiring the vehicle data at a constant rate and, since the set of available signals is strongly vehicle-dependent, it collects them into well-defined packets so that they can be easily transmitted to the high-level Android board. The vehicle signals are clustered into six categories: *battery*, *doors*, *speed*, *faults*, *commands*, and *others*. Each signal available on the vehicle must belong to one of the previous categories. For instance, the provided current, the state of charge of the battery, and the battery's state (charging or not) belong to the *battery* group. Figure 4 shows an example of the signals available on the Tazzari vehicle and how they are grouped by the embedded board. The signals of Birò and Liberty are not reported here for the sake of conciseness.

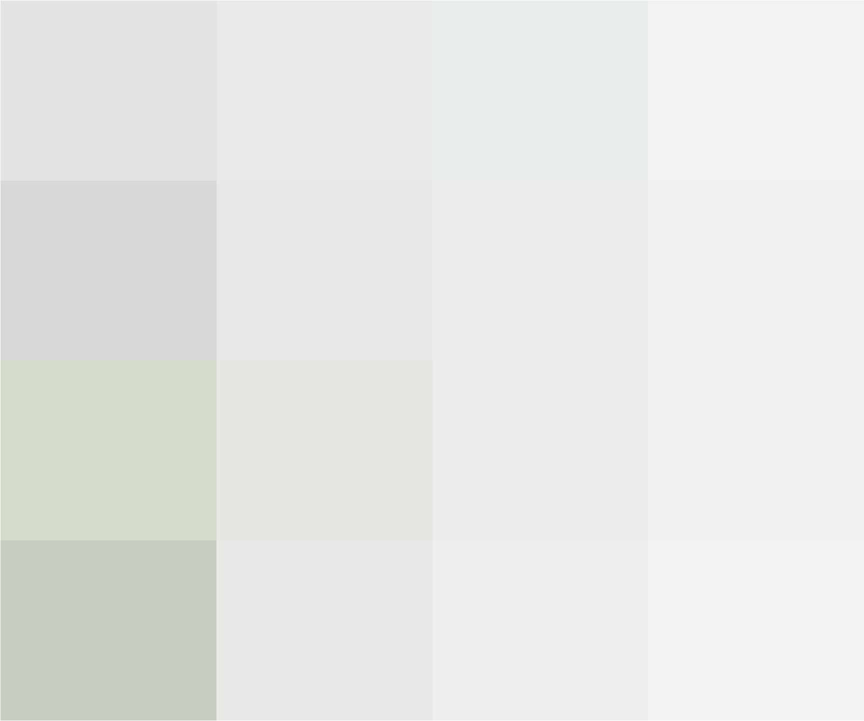

Fig. 4 Green e-Box abstraction layers and measured signals for the Tazzari Zero Evo

The Android board provides the software abstraction layer (SAL) which receives (in a vehicle-independent way) the data from the low-level board; the SAL uses a singleton object to store, for each monitored quantity, the last value received; this object also offers, for each quantity, a getter method that allows other Green Move applications residing on the GEB (see Chap. 9), to easily access the vehicle information. The GEB decouples the high-level fleet management functions from those, implemented in the vehicle ECU, related to the control of the vehicle motion; this isolates the latter from the former, thus establishing their separation and non-interference with one another, which guarantees the necessary safety requirements.

The functions realized by the GEB can be divided into two categories: core operations of the Green Move system (e.g., user authentication, vehicle monitoring), and optional functions that, though not essential, provide added value to GM users (e.g., commercial and traffic information). Whereas core operations are known from the design phase of the Green Move system and change infrequently, optional functions could be added or removed after the system deployment (e.g., because of new commercial agreements between the company running the vehicle sharing system and its partners).

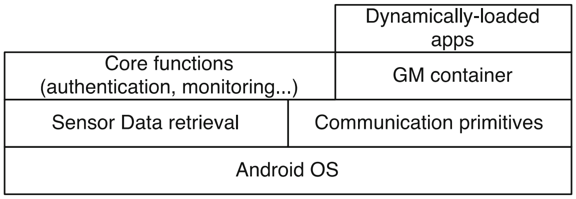

Fig. 5 Layered structure of the functions of the Green e-Box

The software realizing the functions of the GEB is modular, where each module oversees a cohesive set of GEB functions (e.g., vehicle data retrieval, communication with the GMC). Figure 5 shows the layered structure of the GEB software built on top of the Android OS. At the base lay the modules providing the mechanisms for the retrieval of data from the vehicle sensors and for the communication with the GMC. These modules are used to realize the core functions of the GEB, such as, for example, vehicle monitoring, and they are made available to third-party applications through a component called GMContainer.

The GMContainer is a part of the middleware infrastructure that allows system administrators to load and remove applications from GEBs after deployment, thus providing capabilities for the dynamic reconfiguration of the services offered to users at run-time. Chapter 9 explains these features of the Green Move system in more detail.

The GEB is central in the Green Move system, as it takes part in the most significant interactions that occur in it. In the next sections, we describe some of these, which occur between GEB and users' smartphones and between GEB and GMC, and which serve to illustrate some relevant functions performed by the other main components of the Green Move system. Other functions and interactions are described in Chap. 9 and in Chap. 10.

4 Vehicle–User Interaction

In the Green Move system, the user directly interacts with the vehicle—through the Green Move app installed on her mobile device—to get access to the vehicle.

More precisely, every time a new reservation is made by the user, a "virtual key" is generated by the GMC, containing the necessary information to securely access the vehicle at the right time, such as the user, the vehicle, and the date and time of the reservation. The virtual key also contains a cryptographic key, which is used to guarantee the security of the communication when the virtual key is used to access the vehicle. Each virtual key contains its own symmetric cryptographic key, which is freshly generated when the reservation is finalized.

The interaction among GEB, GMC, and the Green Move app installed on the user's mobile device occurs according to the following steps:

- the user, after having created a new reservation, retrieves the corresponding virtual key from the GMC;
- the user asks to get access to the vehicle by sending to the GEB the virtual key received from the GMC;
- the GEB checks with the GMC the validity of the received virtual key;
- if the virtual key is valid, the GEB grants to the user the permission to access the vehicle.

As shown in Fig. 1, the communication between the user's mobile device and the vehicle's GEB can occur through a Bluetooth or NFC channel, although the current prototype focuses on the Bluetooth case. Hence, it does not require an active Internet connection. The security of the connection is guaranteed through the symmetric cryptographic key that is included in the virtual key that both GEB and user's mobile device receive from the GMC. Communications between user's mobile device and GMC and between GMC and GEB occur through the Internet; their security is guaranteed through asymmetric cryptographic keys that are exchanged when the vehicle is added to the fleet, and when the user installs the app on her mobile device, respectively.

For accessing vehicles, the Green Move app offers the user the following functions:

- *retrieval of the virtual key associated with a reservation*, which allows the user to subsequently get access to the reserved vehicle;
- *taking charge of the vehicle*, through which the user gets permission to access the vehicle, which is now under her responsibility;
- *unlock the vehicle*, which allows the user to actually access the vehicle;
- *lock the vehicle, without yet releasing it*; this function prevents others from using the vehicle while the current user is not actively driving it;
- *releasing the vehicle*, which ends the reservation, making the vehicle available to other users.

The locking and unlocking of the vehicle in practice can be carried out in different ways, depending on the features of the vehicle. For example, for the Tazzari Zero car, this is achieved by locking and unlocking the doors of the vehicle; for the Piaggio Liberty scooter, instead, this corresponds to activating and deactivating the ignition. Locking/unlocking the vehicle—and activating/deactivating the ignition—is carried out by exchanging the virtual key that is loaded on both the GEB and the user's mobile device when the user takes charge of the vehicle. This entails that no Internet connection is necessary for either GEB or user's mobile device when this operation is carried out, thus making it available everywhere, including, for example, underground parking garages. Retrieval of the virtual key (by the user's mobile app, or by the vehicle's GEB), instead, requires communication with the GMC, hence Internet connection.

Figures 6 and 7 show the graphical user interface—in the case the reserved vehicle is the Tazzari Zero—and the process through which the user retrieves the virtual key from the GMC and then uses it on the vehicle. When the user requests

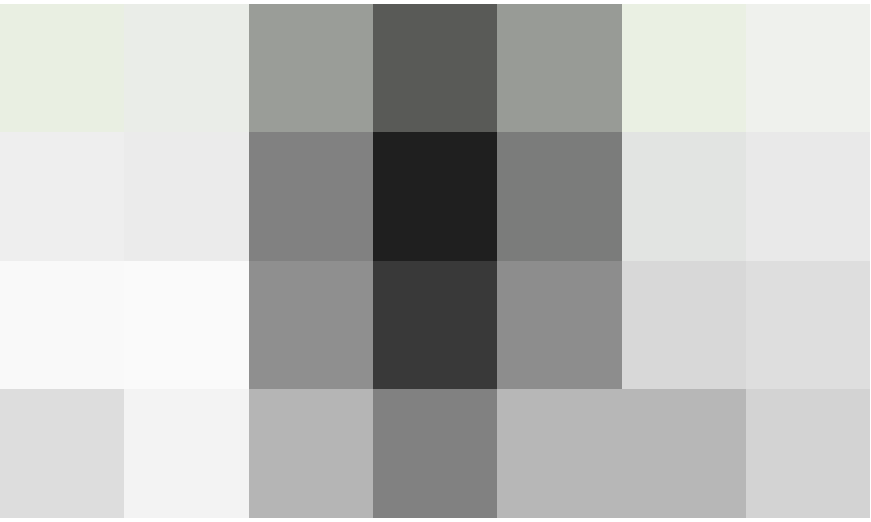

Fig. 6 Retrieval of the virtual key through the user's mobile device: **a** users ask to retrieve the virtual key ("recupera il tuo biglietto" = "retrieve your ticket"); **b** the Green Move app on the mobile device contacts the GMC to retrieve the requested information ("recupero biglietti" = "ticket retrieval"); **c** the virtual key is successfully retrieved ("biglietto ricevuto" = "ticket received")

the virtual key through the mobile app (Fig. 6a), the latter contacts the GMC (Fig. 8.6b), which in turn sends back the key (Fig. 6c). This step can occur well before the user actually needs to access the vehicle. After the virtual key is successfully retrieved, the Green Move app allows the user to take charge of the vehicle (the corresponding button is active in Fig. 6c). At this point, the user approaches the vehicle to get into Bluetooth range and then takes charge of it by pressing the corresponding button on the app (Fig. 7a). As a result, the button on the bottom right corner of the interface, which is used for locking/unlocking the vehicle, becomes active, so the user can press it (Fig. 7b) to actually access the vehicle. Finally, the user releases the vehicle through the corresponding button (Fig. 7c).

5 Vehicle Monitoring

One of the key functions performed by the GEB is the gathering of data concerning the current status of the vehicle (e.g., battery charge, position, speed). Collected data are sent to the GMC to perform a continuous and real-time monitoring of the fleet. To achieve a high level of flexibility in the design of the monitoring mechanisms, the Green Move system relies on the T-Rex CEP engine. Figure 8.8 gives a high-level view of T-Rex.

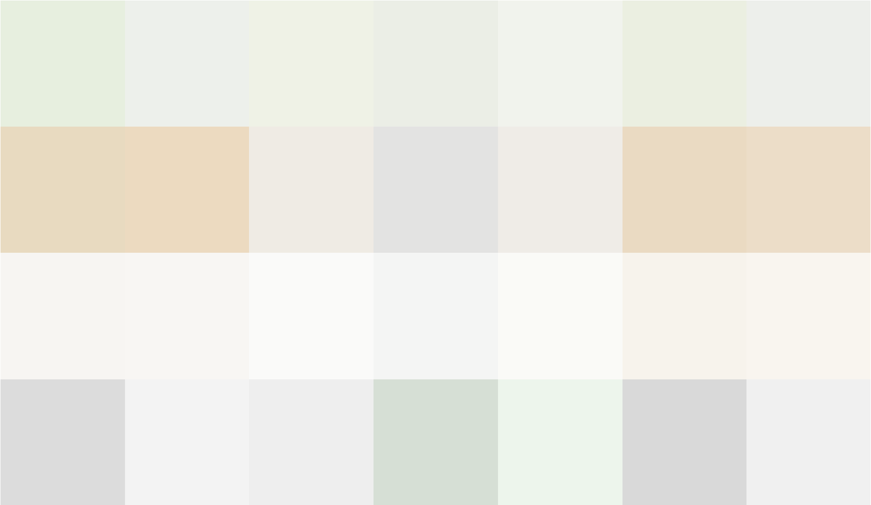

Fig. 7 Use of the virtual key to: **a** take charge of the vehicle ("prendi possesso del veicolo" = "take hold of the vehicle"); **b** lock/unlock the vehicle (through the key icon); **c** release the vehicle ("abbandona definitivamente il veicolo" = "release the vehicle for good")

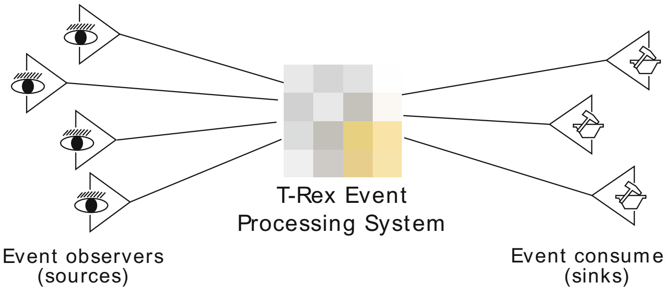

Fig. 8 Overview of the T-Rex system

In CEP infrastructures (Cugola and Margara 2012c), we distinguish between event generators (sources) and event consumers (sinks). The former observe primitive events and report about them, while the latter receive event notifications and react to them. Using the nomenclature typical of publish–subscribe systems, we say that sources *publish* event notifications (or simply events) and sinks *subscribe* to events. The CEP engine sits in the middle with the task of detecting so-called composite events from primitive ones through a set of rules—expressed in an ad hoc language called TESLA (Cugola and Margara 2010)—that are conceived and deployed by rule managers. An example of composite event is "if the vehicle is moving and there is no ongoing rental, then the vehicle is being stolen." The T-Rex engine is loaded with a set of TESLA rules (which can be dynamically added and removed at run-time), which are used to process events and to determine when new —complex—events occur and must be notified to subscribers.

In this framework, the GEB of each vehicle acts both as a source of primitive events, which are processed by the GMC to build an accurate figure of the status of the fleet, and as a sink of complex ones, as explained in Chap. 9.

More precisely, the following is a list of the most relevant primitive events generated by each vehicle's GEB:

- *VehicleData*, a periodic event, which is generated every 60 s, containing the information retrieved by the GEB about the status of the vehicle, such as battery charge, speed, status of the doors (if they are present); these are essentially the data depicted in Fig. 4.
- *Position*, a periodic event, generated every 2 s, containing the current geo-coordinates of the vehicle.
- *Taken*, which is generated every time the vehicle is taken by a user, that is, when the operation depicted in Fig. 7a is completed successfully.
- *Released*, which is the dual of *Taken* and is generated when the user releases the vehicle (corresponding to completion of the operation of Fig. 7c).

The GEBs of the vehicles are not the only sources of primitive events in the Green Move system. The GMC, which stores the information about vehicles' reservations, can also generate them. For example, it generates an event *ReservExpired* whenever the time interval in which a vehicle was reserved by a user terminates.

From the primitive events listed above, the T-Rex CEP engine integrated into the GMC generates complex events according to a set of TESLA rules. In the rest of this section, we illustrate the most significant ones.

Complex event *Theft* is generated whenever the system detects a situation that could correspond to an ongoing theft of a vehicle. The event is defined by the following TESLA rule:

```
define Theft(GreenBox_id: String)
from VehicleData(GreenBox_id = $a and speed > 0.0) and
last Released(GreenBox_id = $a) within 30 days
from VehicleData and
not Taken(GreenBox_id = $a) between VehicleData and Released
where Theft.GreenBox_id = VehicleData.GreenBox_id
```

The rule states that the *Theft* event is generated for a vehicle if: (i) the current status of the vehicle shows that it is moving (`VehicleData.speed > 0.0`) and (ii) since the last release of the vehicle (`last Released within 30 days from VehicleData`) (iii) that same vehicle (`VehicleData.GreenBox_id = $a` and also `Released.GreenBox_id = $a` and `Taken.GreenBox_id = $a`) was not taken again (`not Taken between VehicleData and Released`). The rule also defines that the complex event *Theft* has an attribute corresponding to the id of the GEB (and, by extension, of the vehicle) that triggered the rule (`Theft.GreenBox_id = VehicleData.GreenBox_id`).

The occurrence of a theft is detected also when the vehicle is moving, but it was never released; since the absence of a release would prevent the triggering of the previous rule, which requires the presence of a *Release* event, the next rule, providing a second, complementary definition of event *Theft*, is introduced in the system.

```
define Theft(GreenBox_id: String)
from VehicleData(GreenBox_id = $a and speed > 0.0) and
not Taken(GreenBox_id = $a) within 30 days from VehicleData
where Theft.GreenBox_id = VehicleData.GreenBox_id
```

The next TESLA rule defines a complex event *OpenDoors*, which is generated every time a vehicle is detected to be moving while its doors are open (condition `door_status = 3` corresponds to the doors being open).

```
define OpenDoors(GreenBox_id: String)
from VehicleData(speed > 0.0 and door_status = 3)
where OpenDoors.GreenBox_id = VehicleData.GreenBox_id
```

Notice that if the vehicle does not have doors—or if it has doors which are not electronically controlled—primitive event *VehicleData* will never satisfy constraint condition `door_status = 3`, so the *OpenDoors* event will never be triggered for it.

Finally, the next rule defines a complex event *NotGivenBack*, which is triggered when a vehicle is not released at the end of a reservation period indicated by a user (i.e., when there is not *Release* event between the *Taken* event and the expiration of the corresponding reservation).

```
define NotGivenBack(GreenBox_id: String, reservation_id: String)
from ReservExpired(GreenBox_id = $a) and
last Taken(GreenBox_id = $a) within 30days
from ReservExpired and
not Released(GreenBox_id = $a) between ReservExpired and
Taken
where NotGivenBack.GreenBox_id = ReservExpired.GreenBox_id and
NotGivenBack.reservation_id = ReservExpired.id
```

In the case of all rules presented above, the natural subscriber to the generated events is the console that managers can use to monitor the status of the system. However, the flexibility of the publish–subscribe paradigm supported by T-Rex allows for other actors (e.g., a unit of security guards on alert against thefts) to seamlessly receive generated events. In addition, thanks to the flexibility and the expressiveness offered by the TESLA language for defining rules governing the generation of complex events, new monitoring rules (hence complex events) can be added—possibly dynamically—to the system after its deployment.

6 Related Work

The Green Move project explored innovative solutions both in the area of the *interaction between user and vehicle*—which in the Green Move approach is keyless and smartphone-based, without requiring a membership card—and in the area of the *configurability of the services offered*—which can be dynamically modified on each vehicle after system deployment. As discussed in the rest of this section, more and more commercial car sharing services are pursuing a keyless approach similar, though not as powerful, to the one pioneered by the Green Move project; conversely, none of them has yet reached the level of configurability that the Green Move platform allows.

Using smartphones to de-materialize the interaction between user and vehicle and to dynamically add and modify available services is, especially in the field of electric vehicles (Alli et al. 2012; Dardanelli et al. 2012; conti et al. 2011), a growing area of interest. *INVERS,*[4] *Convadis,*[5] and *Eileo*[6] are the global market leaders in the domain of technologies for car sharing kits. The vehicle sharing systems built through these kits—which are the majority—have a number of drawbacks and limitations compared to the Green Move platform:

their configurability is limited;

- they do not support Bluetooth, nor NFC communication for replacing car keys;
- the ownership of the user database and the rights to the user profiles belong to Convadis/Eileo/INVERS rather than to the vehicle sharing company;
- they do not offer the possibility to add/remove functions depending on the user, location or period.

Some vehicle sharing systems have recently included concepts similar to those introduced in the Green Move project. For example, *Enjoy*[7] is one of the first systems which does not require a membership card for entering the vehicle. The opening/closing of the vehicle doors is performed through a smartphone app, but the communication between vehicle and user smartphone is always mediated by a central server, where the commands are processed and then forwarded to the

[4] www.invers.com.

[5] www.convadis.ch.

[6] www.eileo.com.

[7] enjoy.eni.com.

vehicle. This procedure requires time; thus, the vehicle doors do not open for many seconds (around 30) after the smartphone sends the command. On the contrary, the Green Move platform relies on NFC and Bluetooth communication channels between the user smartphone and the vehicle to directly send open/close commands to the latter. Hence, the interaction is much faster—vehicle doors open immediately after the user sends the command through the smartphone; in addition, it does not require a connection to the server, so it can occur also when this is absent, as in underground parking garages. Interested readers can refer to Bianchessi et al. (2013a) for a review of available car sharing technologies.

The topic of service configurability is even less explored than smartphone-based interaction: no vehicle sharing system offers the possibility to configure the services of vehicles depending on the user preferences or the state of the vehicle. Some of the most technologically advanced services offer, through the computers onboard the vehicles, applications that enrich the driving experience. For instance, *Car2go*[8] offers two onboard services, the navigator and the Driving Style Assistance System. However, no customization is available; the services behave the same regardless of the user's past habits or vehicle type.

7 Conclusions

In this chapter, we have introduced the architecture of the prototype of Green Move system. We have described the main functions of the system components (Green Move Center, Green e-Boxes, and the app installed on users' mobile devices) and how they interact with one another to provide the core mechanisms on which the vehicle sharing system is built, such as vehicle taking and releasing, fleet monitoring. The next chapters introduce some advanced, value-added services offered by the Green Move prototype, which are based on the architecture described above.

References

Alli G, Baresi L, Bianchessi A, Cugola G, Margara A, Morzenti A, Ongini C, Panigati E, Rossi M, Rotondi S (2012) "Green move: towards next generation sustainable smartphone-based vehicle sharing," sustainable internet and ICT for sustainability (SustainIT). IEEE, pp 1–5

Bianchessi A, Ongini C, Alli G, Panigati E, Savaresi SM (2013a) "Vehicle-sharing: Technological infrastructure, vehicles, and user-side devices -technological review," International conference on intelligent transportation systems (ITSC), pp 1599–1604

Bianchessi AG, Cugola G, Formentin S, Morzenti A, Ongini C, Panigati E, Rossi M, Savaresi SM, Schreiber FA, Tanca L, Vannutelli Depoli EG (2014) Green Move: A platform for highly vonfigurable, heterogeneous electric vehicle sharing. IEEE Intell Transp Syst Mag 6(3):96–108

[8] www.car2go.com.

Bianchessi A, Ongini C, Rotondi S, Tanelli M, Rossi M, Cugola G, Savaresi SM (2013b) A flexible architecture for managing vehicle sharing systems. IEEE Embed Syst Lett 5(3):30–33

Conti M, Fedeli D, Virgulti M (2011) Bluetooth for electric vehicle to smart grid connection. IEEE Intel Solutions Embed Syst 1:13–18

Cugola G, Margara A (2010) "TESLA: a Formally Defined Event Specification Language". Proceedings of the ACM international conference on distributed event-based systems (DEBS)

Cugola G, Margara A (2012a) Complex event processing with T-REX. J Syst Softw 85(8): 1709–1728

Cugola G, Margara A (2012b) Low latency complex event processing on parallel hardware. J Parallel Distrib Comput 72(2):205–218

Cugola G, Margara A (2012c) "Processing flows of information: From data stream to complex event processing," ACM Comput Surveys 44(3):1–15

Dardanelli A, Tanelli M, Picasso B, Savaresi S, di Tanna O, Santucci M (2012) A smartphone-in-the-loop active state-of-charge manager for electric vehicles. IEEE ASME Trans Mechatron 17(3):454–463