

EVS27
Barcelona, Spain, November 17-20, 2013

A Smart Phone version of an urban e-transportation reservation service

Christos S. Ioakimidis¹, Aitor P. Zabala², Dragan Simic³, Dionisis Kehagias⁴, Alvaro S. Miralles⁵

¹*DeustoTech, Energy Unit, University of Deusto, Department of Industrial Technologies, Avda. de las Universidades, 24 - 48003 Bilbao, Spain, christos.ioakimidis@deusto.es*

²*University of Deusto, Department of Informatics and Industrial Organisation, Avda. de las Universidades, 24 - 48003 Bilbao, Spain, apalzab@gmail.com*

³*Mobility Department, Electric Drive Technologies Unit, Austrian Institute of Technology (AIT), Giefinggasse 2, 1210, Vienna, Austria, dragan.simic@ait.ac.at*

⁴*Centre for Research and Technology Hellas/Information Technologies Institute (CERTH/ITI), 6th Km. Charilaou - Themi Rd., 57001, Thessaloniki, Greece, diok@iti.gr*

⁵*Electronics Department, Instituto de Investigación Tecnológica, Universidad Pontificia Comillas (IIT/ICAI), Madrid, Spain, alvaro@upcomillas.es*

Abstract

The objective of this work is to present the use of a Smart Phone operation as part of a larger project at the University of Deusto on Smart Grids (UDSmartGridTM), Spain, as the way of being able to make a reservation of an electric vehicle service under a short-term leasing/sharing/pooling case study. The ReadyToCharge[®] is an intelligent system application protocol that has as a purpose to be able to be used by the customer of such a service in a very and simple efficient way. The technologies used are the Google API 7, the Eclipse software and the SQLite Manager.

Keywords: business model, communication, intelligent, mobility, optimization

1 Introduction

Due to the growing concern over the increasing pollution in our planet, electric vehicles (EVs) began to gain popularity, along with the use of services like carpooling, carsharing and short-term leasing [1,2]. The main problem with EVs has been the concern of the users being unable to have a sufficient autonomy for their vehicle, typically referred to as range anxiety. Along with this concern and the increasing use of

smartphones, the envisioned application offers a support platform for services like carpooling and carsharing in order to offer a solution to this problem. In this context, the idea of the ReadyToCharge[®] application was born.

The objective of this application is to provide a real-time image of the availability from the vehicles and the charging posts to the users of the system. This service is a part of a larger project (Fig.1) at the University of Deusto on Smart Grids (UDSmartGridTM), Spain.

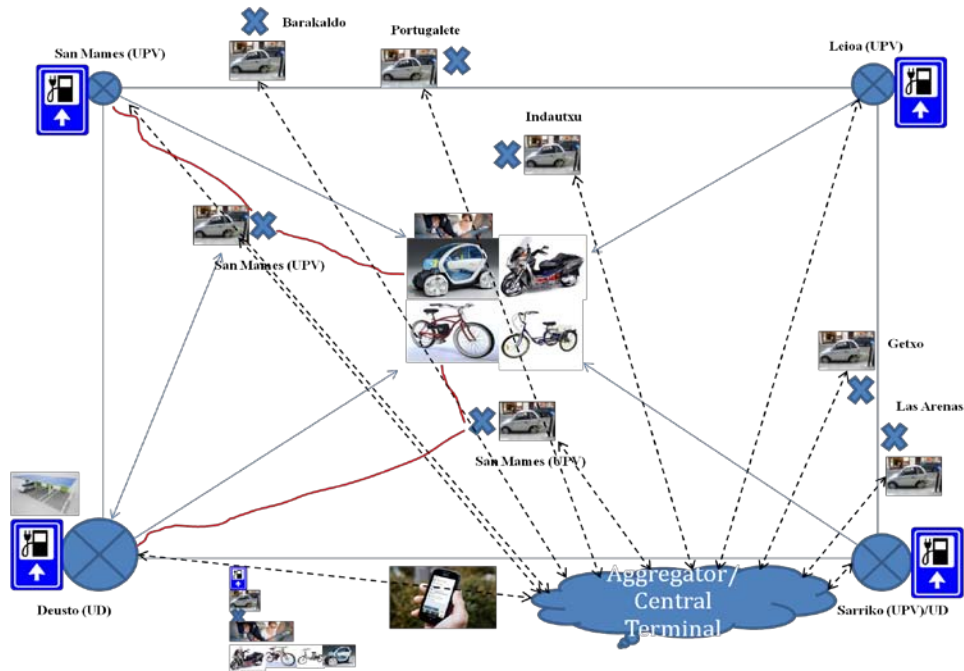


Figure1: Architecture of the Proposed Business Model and Grid (UDSmartGridTM) and the corresponding parts of an e-transportation service

2 Requirements

To use the ReadyToCharge application, the user must be registered in advance with the company that manages this service. Another requirement is that the user possesses a mobile phone running the Android Operating System and having an internet connection. At the moment of writing, the application supports all Android versions starting from 2.3 (Gingerbread) and up. In further development, support may be extended to versions starting from 2.1 (Eclair).

3 Application Specifications

3.1 Location and positioning

Let us examine further now how such a service takes place. The steps that are taken are the following: The customer of a service related with the use of a 'Short-Term Leasing' of EVs needs to be able to make an e-transportation reservation on real time either by the use of a PC using internet or by the use of a Smart Phone. Here is examined the second case. The customer considered here is one of the 8.067 persons that have as a declared residence located in the urban and suburban city of Bilbao and the possibility also of having a Smart Phone access. The number mentioned before includes the University students and the administration and faculty ones of the University of Deusto (that are part of this service). The latter is considered to be one of the seven (7) different charging points at

which the people will be able to charge the EVs when using the service under study through their Smart Phones.

The application presented is designed to make reservations for e-vehicles charging in any of the seven charging stations around the city of Bilbao: Deusto (central station), another two in the centre of the city, Las Arenas, Getxo, Barakaldo and Portugalete (Fig.2). In each of these service stations, there can be found different posts for the three available types of electric vehicles: cars, bicycles and motorbikes.

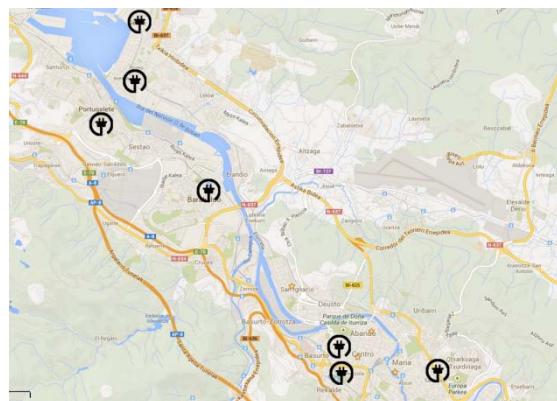


Figure2: Map of the available charge stations

As for the positioning, the ReadyToCharge application offers a great flexibility. Since it is running on the Android platform on a mobile phone, the current user's position can be retrieved

by normal antennas, i.e. the position is determined through the nearest antenna available. Since the ReadyToCharge application is running on a mobile phone, there is also access to the GPS functions: the GPS from the phone can therefore be easily used to locate the vehicle accurately and to calculate, with further data, if the user will get to the charge station with remaining battery charge.

3.2 System structure

The application uses a system structure based on a very simple design, consisting of a central server that controls the entire system (Fig.3).

The user asks from the server for available times, validation, current state... and after the processing if the request, the server sends back the requested data (information about users, current state of the charge stations, news, booking information...).

3.3 Deployment

The smart phone application asks information to a central server that knows if there are or not available times for an e-vehicle charging (car, motorbike or bike), thereafter referred to as the ReadyToCharge Server. The application is deployed in GAE (Google App Engine), coded in Java following the REST (Representational State Transfer) architectonic design patterns.

The term *representational state transfer* was introduced and defined in 2000 by Roy Fielding in his doctoral dissertation [3]. Fielding is one of the principal authors of the Hypertext Transfer Protocol (HTTP) specification versions 1.0 and 1.1.

For that purpose uses Jersey RESTful Web Services framework, which is open source, production quality, framework for developing RESTful Web Services in Java that provides support for JAX-RS APIs and serves as a JAX-RS (JSR 311 & JSR 339) Reference Implementation in [3-5].

The data are stored in the Datastore (GAE database) and they are retrieved upon client request. These requests are made using HTTP verbs (GET, POST, PUT and DELETE) to the resources (Fig.4) and the exchange of data between the server and the mobile application, such as the web page is made using JSON (Fig.5), which is becoming more and more popular nowadays.

The key goals of this kind of system architecture include: scalability of component interactions, generality of interfaces and independent deployment of components, among others. These

systems are stateless, so each HTTP request occurs in complete isolation. When a client makes a request, it includes all the information necessary for the server to fulfill that request. The server never relies on information from previous requests. If that information was important, the client would have to send again the request.

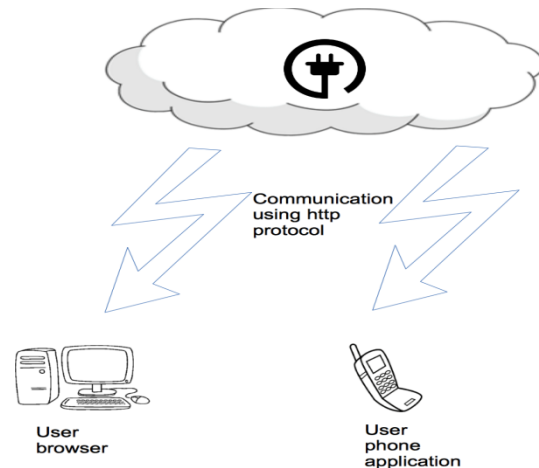


Figure3: System structure

```
Request URL: http://localhost:8888/app/rtc_available_times/Deusto/moto/2013-07-22
Request Method: GET
Status Code: 200 OK
Request Headers
Accept: */*
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
Connection: keep-alive
Host: localhost:8888
Referer: http://localhost:8888/ready_to_charge.html
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_0_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1
X-Requested-With: XMLHttpRequest
Response Headers
Cache-Control: no-cache
Content-Length: 2216
Content-Type: application/json
Date: Mon, 22 Jul 2013 17:00:03 GMT
Expires: Fri, 01 Jan 1990 00:00:00 GMT
Server: Development/1.0
```

Figure4: Making a HTTP GET request to the ReadyToCharge Server

```
[
  {
    "year": "2013",
    "month": "7",
    "day": "22",
    "hour": "19",
    "minutes": "0",
    "chargeStation": "Deusto",
    "chargeType": "coche"
  },
  {
    "year": "2013",
    "month": "7",
    "day": "22",
    "hour": "19",
    "minutes": "30",
    "chargeStation": "Deusto",
    "chargeType": "coche"
  },
  {
    "year": "2013",
    "month": "7",
    "day": "22",
    "hour": "20",
    "minutes": "0",
    "chargeStation": "Deusto",
    "chargeType": "coche"
  },
  {
    "year": "2013",
    "month": "7",
    "day": "22",
    "hour": "20",
    "minutes": "30",
    "chargeStation": "Deusto",
    "chargeType": "coche"
  }
]
```

Figure5: The server response: a JSON array containing all the available times for Deusto charge station

4 Using the Application

This part gives a general outline of the features of the smart phone application, by running through each screen of the application.

4.1 Login

The first screen that pops up when the user opens the application is the validation panel (Fig.6); this contains two textboxes, allowing the user to fill his username and password (registered beforehand with the company hosting this service) and using this information to login to the system.

Privacy is another of the aspects to take into account. REST resources that contain sensible information should not be sent in clear [6]. One important point is to know how to perform the authentication and authorization. HTTP provides mechanisms to do that: WWW-Authenticate header is one of them. The idea is when a request is made to a protected REST resource and that request does not have sufficient privileges, the server responds with a 401 that contains the header WWW-Authenticate. This header details the kind of credentials and the security domain that is needed. This is what is called authorization.

As already pointed out, REST systems we are stateless, so the client has to send in every request all the information that the server needs to carry out the response. Then, according to this, do we need to authenticate again and again? The HTTP protocol does not tell us much about this. The simplest designs sends in each request the Authorization header that we have talked above, but this usually has some implementation problems. The solution is to authenticate only the first time and then use a security token. When the server authenticates the user the first time sets a cookie with that token in the response header. Next requests must have that token and if it is valid the server would trust on it and serve the request.

How to generate that token? One of the techniques is to use a HMAC.

1. PUBLIC_PART = UUID() + ":" + SECURITY_LEVEL + ":" + TIMESTAMP
2. SIGNATURE = HMAC(SERVER_SECRET, PUBLIC_PART)
3. TOKEN = SIGNATURE + "_" + PUBLIC_PART

With the UUID we get a universal unique identifier, SECURITY_LEVEL indicates the security role and the TIMESTAMP allow detecting the old tokens. Some people think that

this kind of token is like a session token, but this approach has one difference: the security tokens are not stored on the server side (session tokens are stored, so they represent a memory space on the server side, stateful servers). In a RESTful API there is not a session, it is stateless. The good part of doing like that is that we can validate the tokens calculating the HMAC of the PUBLIC_PART, and if it coincides with the SIGNATURE the token is valid.

Regarding the application, if the sent user validation data is correct, the central server will send the user data registered in the database as a response that the login succeeded. The next requests will have the aforementioned token.

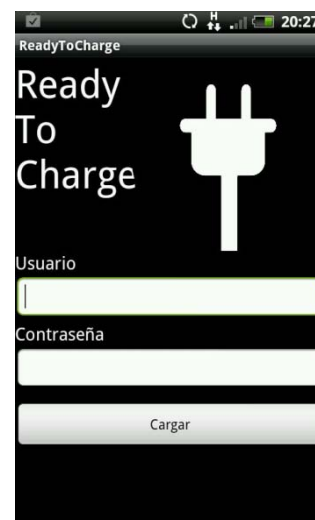


Figure6: Validation screen

4.2 Vehicle and pickup time selection

After the user has logged in to the system, the application shows the three available types of electric vehicles (car, motorbike and bicycle), and lets the user pick a vehicle, select the preferred date, and a button to confirm the selection. This is shown in Fig.7.

4.3 Map

If the preferred vehicle type and pickup time are selected, the application shows a map of the environment (Fig.8), showing the selectable charge stations. When touched, the user gets information about the distance between the selected station and his own location.

The map also shows the traffic overlay, indicating the fluid as either green (fast traffic) or red (slow traffic).

This map and the traffic state is provided by Google Maps API v3.



Figure7: Vehicle and pickup time selection screen

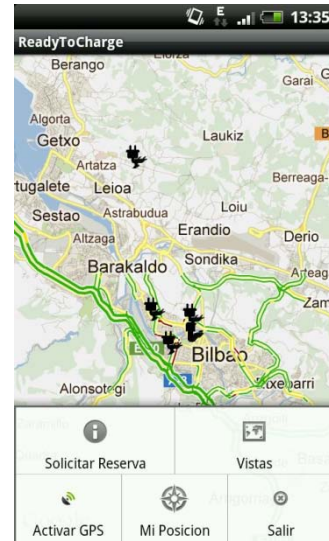


Figure9: Options panel

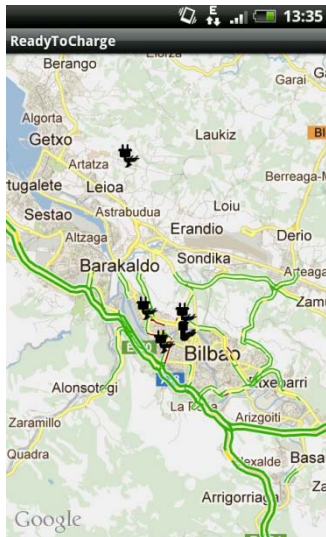


Figure8: Map showing host stations

4.4 Options panel

If the user then pushes the menu button (often looks like this), a panel (Fig.9) pops up, showing options to activate the GPS, make a booking etc. The available options are:

- Activate GPS: this enables the GPS setting on the mobile device.
- Change view: this toggles the view between normal view and satellite view.
- My position: this locates the user on the map.
- Make reservation: this proceeds to the reservation itself (this is explained further on).
- Close: this closes the panel and returns to the full screen map.

4.5 Station selection

If the user chooses to make a reservation, the application first shows a list of all the stations, with the respective distance shown (Fig.10). Here, the user can select the preferred station where the user wants to charge the vehicle.



Figure10: Station selection screen

4.6 Hour selection

When the user selects the preferred charging station, a request is made to the server asking for the availability of the data passed and sends the response with the available timetables (HTTP GET readytocharhe.appspot.com/available_times/deusto/car/04-08-2013), as shown in Fig.11. Now, all relevant information is known to the system to make the reservation: the location (from the station selection screen) and the preferred hour as well as the type of vehicle.



Figure11: Hour selection screen

4.7 Reservation confirmation

When a reservation hour is selected, the application shows a confirmation screen (Fig.12). Here, the selected station with its ID and the selected hour are shown.

From here, the user can either close the application by pushing the ‘Close’ button, or confirm the reservation for the time and place shown by pushing the ‘Confirm’ button. The confirm button fires another HTTP call to the server that completes the process. After that a navigation option is available that works like a usual GPS navigation system to guide the user to the charge station.



Figure12: Confirmation screen

4.8 Notification

Finally, the application puts a notification in the phone’s notification bar, stating the reservation hour and place (Fig.13). An email is also sent to the user containing all the booking details like date, charge type, charge station, price, booking

identification, charge post ID... The option of cancelling the reservation is available from the phone, but depends on the case some if extra charges will be applied on the user’s account or another kind of penalization.



Figure13: Confirmation screen

5 Under development

The ReadyToCharge&Go web page (Fig.14), which is currently under development, is a site that offers the same functionality as the phone application so that the users can for example make a booking from a PC (Fig.15). We are also working on e-car sharing-pooling & leasing because we believe that this is the future of commuting. You can have a look at readytocharge.appspot.com.

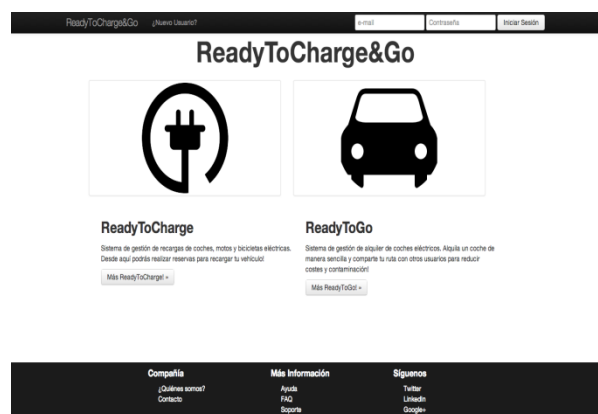


Figure14: ReadyToCharge main page

6 Discussion

There is general consensus that the increasing human mobility needs, combined with the high use of private vehicles are primary causes of serious environmental and social problems worldwide, having negative impact on the quality of life. The

fact that the transportation section is considered one of the primary fuel use sectors worldwide, while being responsible for a significant share of greenhouse gas (GHG) emissions, emphasizes the importance of an alternative in this area even more.

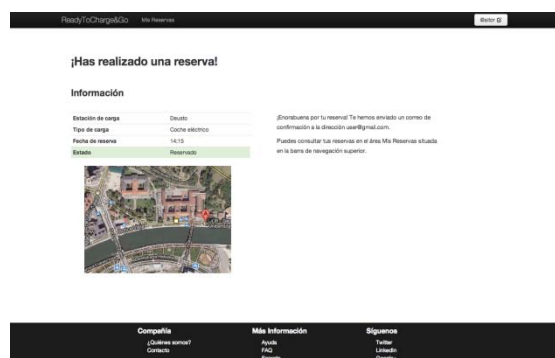


Figure 15: ReadyToCharge booking confirmation screen

This work is related with the creation of an ecological, technological and more economic service provided mainly in the urban area of Bilbao, to serve the mobility for students, academics and administration staff from their residential point to the University Campus of Deusto. Even more this is a Short-Term Leasing E-transportation service (i.e., electric car-scooter-bicycle) under the specific case of an UDSmart-GridTM concept. ReadyToCharge is a part of that project and the purpose is to provide a booking system for electric vehicles in the city of Bilbao, Spain [7].

7 Conclusions

High costs of acquisition and a poor autonomy are the two main factors advanced to explain the inertia in the electro-mobility sales. Others crippling criteria are the charging and the poor infrastructure (about 15,000 charging posts in Europe in November 2012). A system like ReadyToCharge that allows users to book a slot of time in a charge station solves the problem of the poor infrastructure because the client ensures the charge of the electric vehicle.

The actual autonomy of an electric car reaches 130 km, depending on the condition of use. It is far beyond the average of daily 33 km routes in France for example. Electric vehicles are also characterized by a lower cost by km and a better efficiency compared to ICEVs.

The use of such an application is very useful and its ease of use might have a positive effect on the participation from more users in such a system.

Exploiting the possibilities of this system is a good step towards a cleaner environment, reducing the emissions of our daily transport.

References

- [1] K.N. Genikomsakis, C.S. Ioakimidis, B. Bocquier, D. Savvidis, D. Simic, "Electromobility and carsharing/carpooling services at the University of Deusto: A preliminary exploratory survey", 16th International IEEE Conference on Intelligent Transportation Systems, October 6-9, 2013, The Hague, The Netherlands.
- [2] C.S. Ioakimidis, D.F. Rioja, A.O. Collado, D. Savvidis, "An optimization of Electric Vehicles parking spot locations and its economic analysis under a short term leasing case", Report, Univ. de Deusto, Bilbao, 8 p., 2012.
- [3] R. Thomas Fielding, *Architectural Styles and the Design of Network-based Software Architectures*, University of California, 2000
- [4] Project Kenai, <https://jax-rs-spec.java.net/>, accessed on 2013-03-06
- [5] O'Reilly RESTful Web Services (Leonard Richardson & Sam Ruby, May 2007), O'Reilly RESTful Web Services Cookbook (Subbu Allamaraju, March 2010), O'Reilly REST API Design Rulebook (Mark Massé, October 2010).
- [6] E. Amodeo, Principios de diseño de APIs.
- [7] Electro mobility, communication intervehicles and relation to the grid for Deusto students and workers. EcoEICar™

Authors

Christos S. Ioakimidis was born in Thessaloniki, Greece. He is currently a Senior Researcher/ Professor and Group Leader on Energy Modeling and Systems Integration at DeustoTech (Energy Unit), University of Deusto, Bilbao, Spain. He holds a BSc degree in Mechanical Engineering, (Greece, 1994), an MSc degree in Mechanical & Aerospace Engineering, (USA, 1996), a PhD in Mechanical & Chemical Engineering, (U.K., 2001), and an MBA, (Greece, 2007). His research interests are on transport/energy modelling, smart grids, energy systems integration (renewables, CCS, EVs), innovation and green economy and nanotechnology applied on automotion/energy.





Aitor Zabala is a final year undergraduate student at the University of Deusto, Department of Informatics and Industrial Organisation, Bilbao, Spain. His research interests are on communication protocols applied in the transportation electricity and energy market, web design and on the creation of TICS and innovation.



Dragan Simic is a senior researcher at AIT, Vienna, Austria. He received the Dipl. Ing. Degree in Mechanical engineering from the faculty of electrical, mechanical engineering and naval architecture, University of Split, Split, Croatia in 1999, his PhD from Vienna University of Technology, Austria in 2007 and an MBA in automotive industry from TUV and the Slovak University of Technology in 2013. His research activities are focused on the longitudinal simulation of the conventional and hybrid vehicles including the simulation of the auxiliaries. He is also a member of the Modelica Association.



Dionisis Kehagias is a Senior Research Associate at the Centre for Research and Technology (CERTH), Greece. He holds a Diploma (1999) and a PhD (2006) in Electrical and Computer Engineering from the Aristotle University of Thessaloniki (AUTH). His research activities lie in the areas of ICT-enabled Energy Efficiency, Semantic Web and Intelligent Agents. He has been also serving as an Adjunct Lecturer in two Universities in Greece (DUTH and AUTH – Department of Electrical and Computer Engineering) and a faculty member in the Master of Science Program in Web Science at AUTH.



Alvaro Sánchez Miralles obtained his Electronic Engineering Degree from Universidad Pontificia Comillas (ICAI), Madrid, Spain, in 1998. He obtained the Degree of Doctor Ingeniero Industrial from Universidad Pontificia Comillas, in 2003. His interest areas are: smartgrids, electrical vehicles, security systems and mobile robotics.