Increasing Privacy and Trust in Cooperative Social Platforms for Vehicular Applications

F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil and J. Molina-Gil

Department of Statistics, Operations Research and Computing,
University of La Laguna, San Cristóbal de La Laguna, Spain

Abstract. New research challenges have arisen from the increasing number of vehicles and consequent permanent traffic jams in urban environments. The main problem is that just one passenger occupies most cars, so a natural solution should involve a more optimized use of resources, which may be carried out through carpooling. However, carpooling is not seen as an acceptable solution by many users mainly due to the lack of trust in strangers. Nowadays, mobile technology can be combined with social media to create a trust-based system that allows sharing cars in a comfortable, fast and safe way, taking advantage of the comfort of travelling in a car. The main aim of this work is the design of an improvement for existing carpooling systems that introduces a security layer to provide privacy and trust through the measurement of user reputation. The proposal has been implemented as an Android application whose results are promising.

1 Introduction

The increase in the number of vehicles has raised the price of fuel and the pollution in big cities. These facts have led governments to take steps to try to decrease pollution mainly in urban centres. The use of public transport might significantly reduce these problems, but public transport is not always convenient and/or affordable for all users.

A convenient solution involves sharing cars so that empty seats could be used in most trips. This modality is known with the term carpooling and has been proposed as an effective way to reduce both pollution and spending. Another related but different proposal, known as carsharing, is based on collective fleets of cars with multiple users, but such a solution does not solve as many problems as carpooling. Ridesharing is the general term used to refer to solutions for sharing the use of a car with other people in order to travel to a given destination. Apart from carsharing and carpooling (also known as real-time or instant or dynamic or ad-hoc ridesharing), ridesharing also includes other versions known as slugging, lift sharing and covoiturage. However, ridesharing proposals that are different from carpooling are out of the scope of this work.

The use of both types of collaborative solutions has increased since the onset of the economic crisis thanks to technology 2.0. They are applicable in almost any environment, but are especially useful in places like universities, holidays, long journeys and even urban centres. This is because in these situations both vehicle owners and passengers agree on the same motivations to consider carpooling. Usually their main goal is

Martín-Fernández F., Caballero-Gil P., Caballero-Gil C. and Molina-Gil J..
Increasing Privacy and Trust in Cooperative Social Platforms for Vehicular Applications.
DOI: 10.5220/0004976800030013
In *Proceedings of the 11th International Workshop on Security in Information Systems* (WOSIS-2014), pages 3-13
ISBN: 978-989-758-031-4

to share fuel cost, but there may be other reasons such as solving the parking problem, wanting to talk with others or to take care of the environment, etc.

This work proposes an improvement for existing carpooling solutions that make use of recent technological advances of smartphones and social networks. The described solution allows the establishment of trust and reputation accountability between drivers and passengers, while taking care of their privacy at the same time.

This paper is organized as follows. Section 2 mentions several related works. Section 3 introduces the general design of the proposal, which is mainly based on the reputation algorithm sketched in Section 4. The security of the proposal is briefly analyzed in Section 5. Section 6 describes the developed Android application. Finally, some conclusions and open problems can be found in Section 7.

2 Related Work

The first carpooling projects emerged in the late 1980s [22], but in those days, without the technology available today, many difficult obstacles such as the need to develop a user network and of a convenient communication medium had to be faced.

Gradually, the media used to organize the trips was changing from telephone to more flexible means such as the Internet, email and smartphones. Nowadays, many different carpooling platforms and services exist, but even today, they may be considered in their early stages because none has reached a critical mass of users.

Table I shows several features of different existing carpooling systems, including the most relevant security-related ones. In particular, we have chosen for this comparative analysis the representative systems: CarPooling [7], Blablacar [5], Amovens [4], ZimRide [28], compartir.org [8].

The main trust enforcing system in all these platforms is based on points given by users. However, bypassing this security system is quite easy because users who obtain a negative score, can create a new profile with new credentials and no points.

Apart from these practical platforms already in operation, there are several papers that propose different solutions. The work [23] shows an integrated system for the organization of carpooling service by using different technologies such as web, GIS and SMS. The authors of [3] propose a web platform to carpool. The paper [12] presents a carpooling architecture that uses a credit mechanism to encourage cooperation between users.

Table 1. Carpooling Platforms.

Platform	Social network	Privacy	Points system	Phone cert	Trust algorithm
CarPooling [7]	yes	no	yes	no	no
Blablacar [5]	yes	yes	yes	yes	no
Amovens [4]	yes	yes	yes	no	no
ZimRide [28]	yes	yes	no	no	no
Compartir.org [8]	no	yes	no	no	no
Our system	**yes**	**yes**	**yes**	**no**	**yes**

A more recent work is [17], where an algorithm to encourage carpooling is proposed based on assigning priority to users with positive feedbacks through a fuzzy logic scheme. Another paper, [1], defines a push service to promote carpooling through instant processing. Finally, another interesting proposal is [6], based on a secure multi-agent platform that focuses on the security services allowing both the mutual authentication between the users and the application components with the system.

The main aspect of our work is different from the aforementioned because it deals with the trust aspect of carpooling services through a combination of reputation measurement with privacy protection.

3 Carpooling Platform

The main objective of the proposed design is the increase of both usability and security. Thus, its key factors are user-friendliness and privacy. One of the main features is that users who publish their trips have their privacy fully protected. Unlike other carpooling platforms, in the described system, no user is allowed to access data such as email, phone or full name of others, unless he/she is authenticated on the platform and the algorithm for checking mutual trust returns a valid permission for him/her. In this case, the interested user can see all the data in detail. Otherwise, he/she can only send a request so that the receiver can decide whether the applicant is to be trusted or not.

The algorithm is based on trust relationships so that people who want to use the platform first need to authenticate in the platform through social networks such as Facebook, Twitter or Google+. In this way, the algorithm checks the existence of some chain of trust between the applicant and other users, based on the so-called rule of six degrees of separation [9]. Besides, the reputation gained through the use of the application is an influent factor, which is considered in the decision on whether carpooling with another person. To do this, at the end of every shared travel, the application asks both drivers and passengers to score the other users. Such scores are used in future trips so that seats offered by car drivers with good scores appear in better positions than others with lower scores. Also well-scored passengers have higher probability to have access to more details of drivers.

Our overall system architecture used as an application model known as client-server. The client-server architecture of our system is showed in Figure 1. Its different elements are the following:

- **Client.** Mobile device used for the system.
- **Server.** Hosted in the cloud, and divided into two parts. On the one hand, the GCM server is the Google Cloud Messaging server that handles all the notifications and is responsible for sending the notification when the receiver clients are alive. On the other hand, the DB dedicated server is the server that stores in its DataBase all the data related to the users and system. It also serves as a gateway for sending notifications between the client and the GCM server.

The life cycle of the application is shown in Figure 2, and explained below step by step:

- **Step 1.** The user logs on the application through any of its social networks (Face-

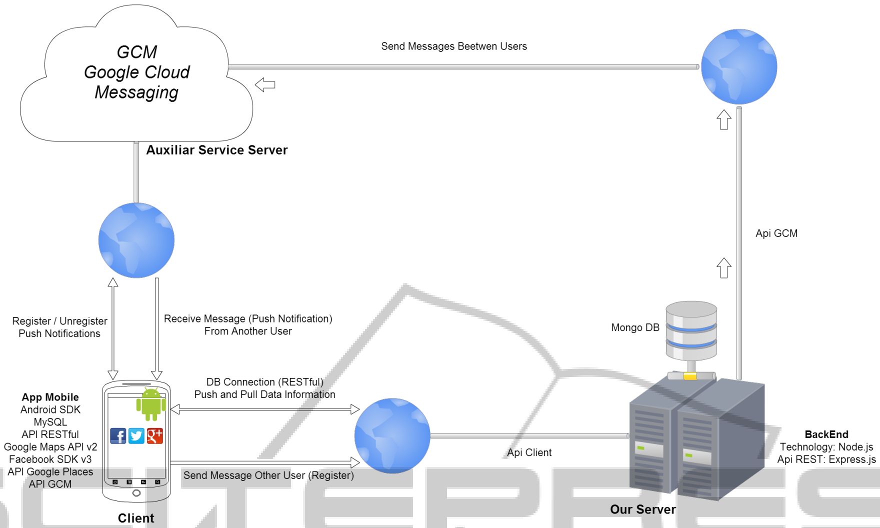


Fig. 1. Carpooling System Architecture.

book, Twitter, Google+). Then, its data are saved in the DB server and the device is registered in the GCM server for notifications.

- **Step 2.** Once logged in, the user has two possibilities:
  1. The user can play the role of driver, and create new routes or manage existing routes. On the one hand, if a new route is created, all related information is stored in real time on the DB server. On the other hand, if the user is managing some old routes, first, the client informs the dedicated server to store the shares in the DB. Then, it also indicates to notify possible affected users about route changes in order to ask for acceptance/ rejection to participate in the route, etc. The DB server sends the necessary information to the GCM server, so that this sends such notification to the involved clients when they are turned on and connected to the network.
  2. The user can play the role of a passenger who tries to find some trip. Once a trip is found, the user sends a request to be a passenger on that route. This request must be answered positively or negatively by the creator of the route, through a notification sent within a period of time. Such a notification will go from the dedicated server to the GCM server because this is the responsible server for sending notifications to the clients.

The proposed scheme protects user privacy through limited and controlled access to user data, according to the trust level stated for the relationship between each pair of users. This trust level is got through the combination of direct scores and trust networks so that it provides the system with enough data to deduce whether people can trust each other or not. In this way, privacy is dealt with as one of the most important aspects of the proposed carpooling system.

A first approach to the development of a trust measurement algorithm that provides a value to each pair of users is based on the use of the PageRank algorithm to predict

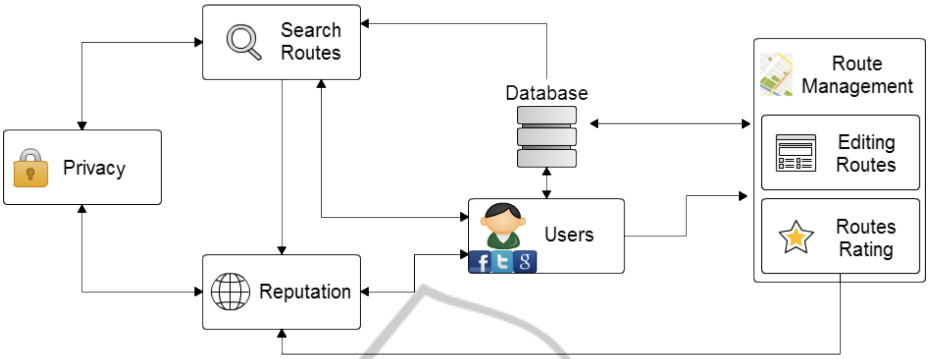


Fig. 2. Use Model.

whether two people can rely each other . However, since this algorithm does not conform totally to the morphology of the specific problem, a second approach is also being used to complement it, based on Bayesian networks to know whether people can trust others. The refined trust measurement algorithm is available in the Android Application.

4 The Reputation Algorithm

No existing carpooling proposal offers the user a quantitative method, based on the theory of six degrees of separation, that can be used to decide whether another user is trustful to share a vehicle. Some of them even do not allow users to decide who may or may not apply your route. There are some proposed that the quantitative method based on the similarities among users [24] [20], having to collect this information about the attributes and characteristics of each user. Our proposal aims to be simple for the user, without the user fills out some information about its attributes. With a simple click (that enables social login on the network that is previously registered) may be entered in our system and start using the platform. The main difference with our proposal is the proposed reputation algorithm. By using it, the user can use a quantitative measure to decide whether trusting another user. The algorithm is based on the theory of six degrees of separation and individual scores within our platform. Six degrees of separation is the theory that everyone is six or fewer steps away from each other in the world, so that a chain of 'a friend of a friend' statements can be made to connect any two people in a maximum of six steps. This number of steps may be reduced significantly by introducing the concept of social networks. Our application uses social networks when logging into the application to create network users to be used to interconnect with each other and provide a reliable measure of confidence. Through the use of social networks we can ensure that the six degrees of separation are reduced to only four. In particular, according to several researches on Facebook [10] [14] [16], the obtained average distance was 3.9 ,corresponding to intermediaries or 'degrees of separation', what shows that the world is even smaller than expected.

The reputation measurement is expressed as a number between 0 and 10, and is computed from the values given by each pair of users to inform about the reliability in each other. This measure is calculated by taking into account the two parameters mentioned above: degree of friendship and appreciation of other users on our platform. If the Facebook social network is used, the score given by the system with the degrees of friendship is broken down as follows:

- **7 points** if a user has only one degree of separation (i.e. it is direct friend).
- **6 points** if both users have a mutual friend.
- **4 points** if among those users there is a chain of friendship of more than two friends.
- **0 points** for the case that there is no chain friendliness.

Other users obtain the remaining scores from the assessments after sharing routes. When a route is completed, users who participated in it, can vote between **1 and 5** stars. Each passenger individually assesses the driver, and the driver individually assesses passengers. The weight is higher from driver to passengers than from passengers to driver, as the driver puts his/her vehicle available to the users. In order to account for the different ratings on a user, we use a simple arithmetic average. The metric taken into account for these ratings is as follows:

- **1 star** will impact the score in **-3 points** if that assessment was given to the driver by the passenger. **-1 point** if it was the driver who voted to the passenger.
- **2 stars** will impact the score in **-1 points** if that assessment was given to the driver by the passenger. **0.1 point** f it was the driver who voted to the passenger.
- **3 stars** will impact the score in **1 points** if that assessment was given to the driver by the passenger. **0.2 point** if it was the driver who voted to the passenger.
- **4 stars** will impact the score in **2 points** if that assessment was given to the driver by the passenger. **1 point** if it was the driver who voted to the passenger.
- **5 stars** will impact the score in **3 points** if that assessment was given to the driver by the passenger. **2 point** if it was the driver who voted to the passenger.

The maximum reputation score that a user can get is **10 points**, which corresponds to the situation when the user's direct friends have rated him/her with the highest scores. A user can have total null valuation if he/she is starting to be known, does not have any degree of friendship, and/or has negative reviews. In this case, the system does not use a score below 0, so that 0 is the minimum score for any user. This valuation is dynamically calculated as a function of the friendship degree that a user has. It helps users to have a reliability measure about whether to trust another user of the platform. Besides, only users who have a valuation higher than **7.5 points** and/or users who have been accepted by the driver to make the route can see certain route data, such as the phone number or any other confidential data.

So the mathematical expression applied to the calculation of the reputation score in the algorithm is:

$$\left(lvFs + \frac{\sum_{i=1}^{n} rat[i]}{n} \right)$$

where:

- **lvFs.** Friendship level measured in points between 0 and 7, depending on the level of friendship each user has with other user, as explained above.

- **rat[].** Each of the ratings a user has received both as driver and passenger, on the routes that has used. The points at which the rating of 1-5 stars are mapped.
- **n.** Total number of ratings that the user has received.

5 Security of the Scheme

Regarding the safety of the platform, Sybil attack is one of notorious attacks in traditional carpooling systems. This type of attacks are hacking attacks on peer-to-peer networks where a malicious device illegitimately takes multiple identities by forging them. Due to the privacy-preserving environment of carpooling schemes, sybil vulnerability is generally hard to defend against.

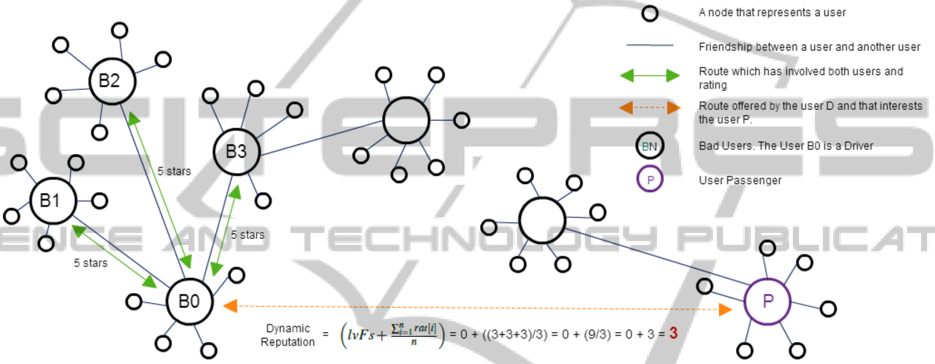

A node that represents a user
Friendship between a user and another user
Route which has involved both users and rating
Route offered by the user D and that interests the user P.
BN
Bad Users. The User B0 is a Driver
P
User Passenger
B2
B3
B1
B0
P
5 stars
5 stars
5 stars
Dynamic Reputation = $\left(lvFs+\frac{\sum_{i=1}^{n} rat[i]}{n}\right)$ = 0 + ((3+3+3)/3) = 0 + (9/3) = 0 + 3 = 3

Fig. 3. Insufficient dynamic reputation rating between an bad user and a possible passenger.

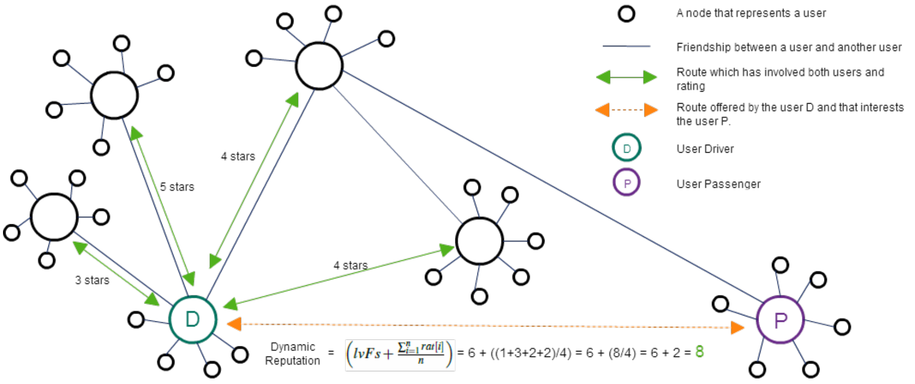

A node that represents a user
Friendship between a user and another user
Route which has involved both users and rating
Route offered by the user D and that interests the user P.
D
User Driver
P
User Passenger
4 stars
5 stars
3 stars
4 stars
D
P
Dynamic Reputation = $\left(lvFs+\frac{\sum_{i=1}^{n} rat[i]}{n}\right)$ = 6 + ((1+3+2+2)/4) = 6 + (8/4) = 6 + 2 = 8

Fig. 4. Reliable dynamic reputation rating between a possible passenger and a driver.

In a Sybil attack the attacker subverts the reputation system of a peer-to-peer network by creating a large number of pseudonymous identities, using them to gain a disproportionately large influence. A reputation system's vulnerability to a Sybil attack depends on how cheaply identities can be generated, the degree to which the reputation system accepts inputs from entities that do not have a chain of trust linking them to a trusted entity, and whether the reputation system treats all entities identically.

An entity on the analyzed peer-to-peer network is a piece of software which has access to local resources. It advertises itself on the peer-to-peer network by presenting an identity. More than one identity can correspond to a single entity. In other words the mapping of identities to entities is many to one. Entities in peer-to-peer networks use multiple identities for purposes of redundancy, resource sharing, reliability and integrity. In peer-to-peer networks the identity is used as an abstraction so that a remote entity can be aware of identities without necessarily knowing the correspondence of identities to local entities. By default, each distinct identity is usually assumed to correspond to a distinct local entity. In reality many identities may correspond to the same local entity.

A faulty node or an adversary may present multiple identities to a peer-to-peer network in order to appear and function as multiple distinct nodes. After becoming part of the peer-to-peer network, the adversary may then overhear communications or act maliciously. By masquerading and presenting multiple identities, the adversary can control the network substantially.

In our case, for example, considering the following scenario:

A 'bad' user ($B0$) sets up several bogus accounts in social media and the proposed system ($B1$, $B2$, $B3$, etc). He/she then advertises a possible trip from $X \mapsto Y \mapsto Z$. For the first leg ($X \mapsto Y$) he/she uses the bogus accounts to claim that his/her vehicle is full and all these 'passengers' state that they get off at Y. He/she can then get excellent scores and increase his/her reputation. At some point, this will be so high that other normal users will be able trust him/her from leg $Y \mapsto Z$.

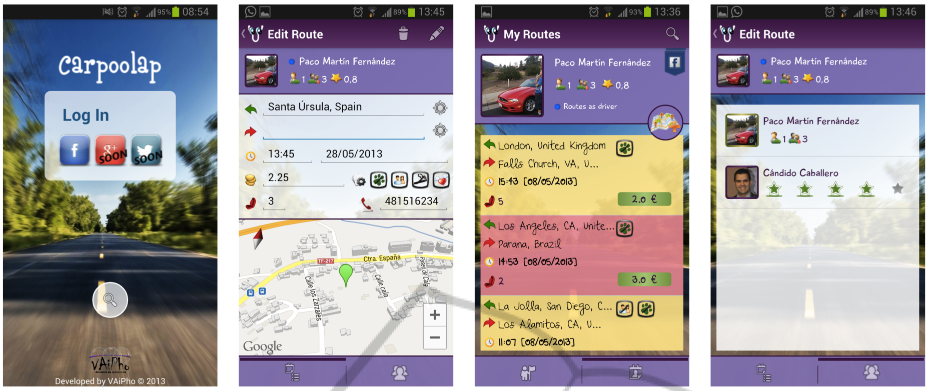
The proposed algorithm is protected against the attack described above because most of the score of our algorithm is preceded by confidence in degrees of friendship that binds each user to another user. Thus, if a user does not know (at all) another user, very high ratings of the latter in the system are not reliable enough for the fomer. It is remarkable that in our system the minimum reliability to trust another user is 7.5 points. At most, a user can have up to 3 points in relation to ratings for the routes that has done (see Figure 3) , far from the 7.5 points that are needed at least for that user to be reliable (see Figure 4).

6 The Android Application

The proposed design has been embodied in an Android application that is already published in the Android Play Store under the name 'Carpoolap' (see Figure 5).

The Android application is developed for the versions 3.0 or higher of the operating system. APIs like Google Maps v3.0, Google Places, Google Cloud Messaging, etc., and Facebook SDKs 3.0 and libraries like Action Bar Sherlock used to use the functionality of the new versions of Android on older versions, were used. Autocomplete in address searches, Google Maps 3D Technology (see second image of Figure 5), design based on the latest versions of Android, push notifications with requests or responses of passengers or drivers, etc. are among the features of the Android Application.

Each user can see the routes he/she proposes as driver (see third image of Figure 5), and whether potential passengers exist for those routes. Besides, with colour codes, he/she can know the routes that each user has already made and the routes that have



Santa Úrsula, Spain
13:45 28/05/2013
2.25
3 481516234

London, United Kingdom
Falls Church, VA, U...
15:43 [08/05/2013]
5 2.0 €

Los Angeles, CA, Unite...
Parana, Brazil
14:53 [08/05/2013]
2 3.0 €

La Jolla, San Diego, C...
Los Alamitos, CA, U...
11:07 [08/05/2013]

Fig. 5. Carpoolap Screens: Login / Edition Route / Routes List / Rating Passengers.

been confirmed by users. For the assessment of users participating in a route, after finishing it, each one can give a score through the screen shown in the fourth image of Figure 5. In order to deploy the carpooling platform, a server is also needed, so we developed one using javascript technologies by frameworks like 'node.js' and 'express.js'. As a database for all the data centralized on this server, we decided to adopt a No SQL database, such as MongoDB [18]. We deployed our server on a micro instance of Amazon Web Services, specifically under Ubuntu machine with Amazon EC2 account.

7 Conclusions

This work proposes an improvement scheme for existing carpooling solutions, whose objective is to increase their safety and reliability. In order to reach this aim, a few existing platforms and some related research work have been reviewed. Also, research on how to assess the social component of a carpooling system in order to build trust between users has been done. From the obtained conclusions, a new proposal has been developed based on gradual access to data according to trust levels got both from direct scoring and social networks. This work includes the implementation of a published Android application that makes use of this research to create a platform for safe carpooling. Many open questions exist, such as the improvement of the application, the development for other platforms, and the merging with existing carpooling solutions.

Acknowledgement

Research supported by the Spanish MINECO and the European FEDER Funds under projects TIN2011-25452 and IPT-2012-0585-370000, and the FPI scholarship BES-2012-051817.

References

1. A. J. Fougeres, P. Canalda, T. Ecarot, A. Samaali, L. Guglielmetti, A Push Service for Carpooling. IEEE International Conference on Green Computing and Communications, pp. 685-691, 2012.
2. A. Oram, Peer-to-peer: harnessing the benefits of a disruptive technology, O'Reilly, 2001.
3. A. Roberts, H. Pimentel, S. Karayevm, CabFriendly: A Cloud-based Mobile Web App. Amazon's EC2, 2011.
4. Amovens: https://www.amovens.com, January 2014.
5. Blablacar: https://www.blablacar.es, January 2014.
6. C. Bonhomme, G. Arnould, D. Khadraoui, Dynamic carpooling mobility services based on secure multi-agent platform, IEEE International Conference on Global Information Infrastructure and Networking Symposium, pp. 1-6, 2012.
7. Carpooling: www.carpooling.es, January 2014.
8. Compartir.org: http://compartir.org/, January 2014.
9. D. J. Watts, S. H. Strogatz, Collective dynamics of small-world networks, Nature, v. 393 pp. 440-442, 1998.
10. E. Y. Daraghmi, S. M. Yuan, We are so close, less than 4 degrees separating you and me!. Computers in Human Behavior, pp. 273-285, 2014.
11. F.E. Prettenthaler, K.W. Steininger, From ownership to service use lifestyle: the potential of car sharing, Ecological Economics 28, pp. 443-453, 1999.
12. J. Ferreira, P. Trigo, P. Filipe, Collaborative Car Pooling System, International Conference on Sustainable Urban Transport and Environment, 2009.
13. J. R. Douceu, The Sybil Attack, Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems, March 2002.
14. J. Ugander, B. Karrer, L. Backstrom, C. Marlow, The Anatomy of the Facebook Social Graph. Cornell University Library, November 2011.
15. K. Steininger, C. Vogl, R. Zettr, The size of the market segment and revealed change in mobility behavior. Transport Policy 3, pp. 177-185, 1996.
16. L. Backstrom, P. Boldi, M. Rosa, J. Ugander, S. Vigna, Four Degrees of Separation, Cornell University Library, November 2011.
17. M. Collotta, G. Pau, V.M. Salerno, G. Scata, A novel trust based algorithm for carpooling transportation systems. IEEE International Conference on Energy Conference and Exhibition, pp. 1077-1082, 2012.
18. MongoDB: www.mongodb.org, January 2014.
19. N. T. Fellows, D. E. Pitfield, An economic and operational evaluation of urban car-sharing, Transportation Research Part D: Transport and Environment 5, pp. 1-10, 2000.
20. N.V. Pukhovskiy, R.E. Lepshokov, Real-Time Carpooling System. International Conference on Collaboration Technologies and Systems, pp. 648-649, 2011.
21. R. Katzev, Car Sharing: A New Approach to Urban Transportation Problems. Analyses of Social Issues and Public Policy 3, pp. 65-86, 2003.
22. R. Teal, Carpooling: Who, how and why, Transportation Research, v. 21A pp. 203-214, 1987.
23. R. W. Calvo, F. De Luigi, P. Haastrup, V. Maniezzoi, A distributed geographic system for the daily car pooling problem, Computers and Operations Research 31, v. 13, 2004.
24. S. Cho, A. Yasar, L. Knapen, T. Bellemans, D. Janssens, G. Wets, A Conceptual Design of an Agent-based Interaction Model for the Carpooling Application. The 1st International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications, pp. 801-807, 2011.
25. T. Talele, G. Pandi, P. Deshmukh, Dynamic Ridesharing Using Social Media, International Journal on AdHoc Networking Systems 2, pp. 29, 2012.

26. Transport For London: www.tfl.gov.uk/tfl/roadusers/congestioncharge, January 2014.
27. U. Kuter, J. Golbeck, SUNNY: A New Algorithm for Trust Inference in Social Networks Using Probabilistic Confidence Models. Proceedings of the Conference on Artificial intelligence AAAI, pp. 1377-1382, 2007.
28. Zimride: http://www.zimride.com/, January 2014.