IoTSpaceReduce: Owners Social Network Association Usage in Reducing IoT Search Space

Chirabrata Bhaumik, Amit Kumar Agrawal, Priyanka Sinha, Avik Ghose
CTO Innovation Lab - Systems
Tata Consultancy Services
Ecospace-1B, Rajarhat,Kolkata,India
Email:{c.bhaumik,amitk.agrawal,priyanka27.s,avik.ghose}@tcs.com

***Abstract*—With the proliferation of sensor loaded smartphones, sensors are becoming a common entity and penetrated to every corner of human civilization. A smartphone today comes with sensors like accelerometer, GPS (Global Positioning System), gyroscope, proximity sensors, magnetometer etc. besides the ever-present microphone and camera sensors. In Internet of things (IoT) space, as these sensors grow in numbers, data generated from them also has started increasing many folds putting a pressure on the infrastructure. However, if we look closely at this huge data, we will find that all data is not meaningful for all intended applications. Moreover, the utility of data generated by these sensors is usually by some aspect of common interest in these sensors. In this paper, we have proposed IoTSpaceReduce, a novel way of reducing the search space for huge sensor data by using the sensor ownership associations in the social network based on the common interest or intent of the owners. An example of ride share application is used to illustrate this concept, and we have discussed how the same concept will be useful while applying on use cases requiring human centric association like medical devices in personal healthcare vertical and smartmeters in home energy vertical.**

I. Introduction

Social networks have changed the way people interact using communication technology. These networks exhibit exact social behavior patterns and associations. There tend to be more cohesion between the persons sharing common interest and intent. Social network can be used as a tool to quickly explore and form cohesive groups based on parameters such as location, age, education, affiliations, interest, entertainment, hobbies, etc. A piece of information related to the common parameter shared by a member of a group will be more useful for other members of the same group. This property of social networks is very useful and it can be applied to any scenario where human centric associations can be formed.

Internet of Things (IoT) deals with the sensor data and provides a means to manage them. [15] shows how low cost sensor platforms are changing the traditional way of energy management by utility companies. Similarly [3], [8], [9] shows the change that can be brought in personal healthcare segment using IoT. Modern smartphone sensors communicate with external world in a different fashion as compared to stand-alone mechanical sensors. With growing number of smart device users, there is a growing amount of data generated by these sensors. If IoT needs to handle and analyze such a huge amount of data, it will affect the real time performance and need more complex algorithms.

Smartphone based sensors are highly personal and human centric. The social behavior, properties and associations formed by the sensor owners as part of their social network can be linked with these sensors. By linking the sensors with the owner's social graph, it becomes capable of sharing all properties of owner's graph. As per application requirement, appropriate graph parameters can be picked to find associations and create corresponding meaningful group. The sensor data will be meaningful to the members of this group. It results in significant reduction in search space for sensor data used by an IoT implementation.

IoTSpaceReduce emphasizes the horizontal and generic aspect of search space reduction in IoT using graph properties. Sensor data search space reduction is clearly pointed out using Friend of a Friend (FOAF) relationship in Facebook® graph. Here FOAF relationship aptly maps to trust and significantly reduces search space. In the wellness vertical, several applications such as detecting similar health conditions within a closed knit group, identifying hereditary traits within a family or workspace imposed health conditions among a group of co-workers can benefit from Social Network Analysis (SNA) based search space reduction. In personal healthcare domain, similar health conditions can be a good additional filtering criteria whereas in home energy domain, timeline and location can serve as a criteria to reduce the search space. An application can be devised for grid level load balancing by considering the timeline and location graph of all the appliances lying within a grid zone. Such applications have to be authorized by consumers as well as utility providers for privacy reasons.

Rest of the paper is organized as follows. The next section describes work related to IoTSpaceReduce, followed by a section providing system overview and some implementation details. The 'Application Details' section next describes application design, algorithms and experiments. We finally conclude and explore future work.

II. Related Work

Representing the social network data in FOAF vocabulary for trust relationship is prevalent in semantic web as seen from [11] but it is not used to reduce search space. [16] explores formation of social network of devices, but this does not consider owner or provider related use cases. [22], [18] propose multi-tier architectures in order to tackle huge sensor data generated in IoT space, but do not suggest sensor associations based on existing human or utility centric relationships.

Ride-sharing is a nice and smart way to reduce cost, fuel emissions and traffic congestion. It is an active area of research with lot of relevant prior work [6], [7], [12], [19], [14], [10]. Trust is one of the basic requirements for a ride-share system. In IoTSpaceReduce, we have used trust as a basic filter to illustrate the concept of search space reduction, and it is defined as FOAF association in social networks. Few previous works suggest trust based on social network associations including [6], [23], [13], [17], [21], [19], [11] close to IoTSpaceReduce. [6] explains association of trust factor with social network and its variation with depth of social network hierarchy. Based on the sampling survey, it suggests that 82% would give a ride to the friend of a friend, where as 69% would accept ride from the friend of a friend. [23] proposes social network based trust establishment (SN-TE) for User Assisted Communication (UAC) implementation. [13] proposes six trust based filters that use three social-based estimators of trust including common interests, common friends, and the distance in a social graph. [17] identifies cyber social status based on various activities in cyber social space and suggest its use as a token of trust in open collaboration. [21] proposes a model for trust establishment, which is based on social computing and tries to avoid interactions with non-desirable participants. [19] suggests the use of well-known techniques from Web 2.0 and social networks to mitigate the threats and social discomfort emanated by ride-sharing. [11] suggests growing importance of trust in social networks and FOAF as a basis for trust in social network. [2] explores trust in both offline and online mode for extraction of components and principles of trust definition in true sense. [5] proposes an approach for supporting and expanding mobile social ride-sharing networks in particular localities using an iterative designing methodology. [16] provides similar concept to IoTSpaceReduce, where devices themselves can form an intelligent social network of their own to reduce search space. [4] first introduces this novel concept of search space reduction using social network association and we have developed IoTSpaceReduce based on this.

III. System Overview and Implementation

The platform on which 'IoTSpaceReduce' framework has been developed and deployed is an elastic cloud for sensor networks which allows application deployment across components and sub-systems. The platform consists of three broad sub-systems, the application sub-system, the edge sub-system and the backend sub-system. The architecture block diagram as shown in Figure 1 gives a high level view of the interactions between different sub-systems. The layered approach is depicted in Figure 2.

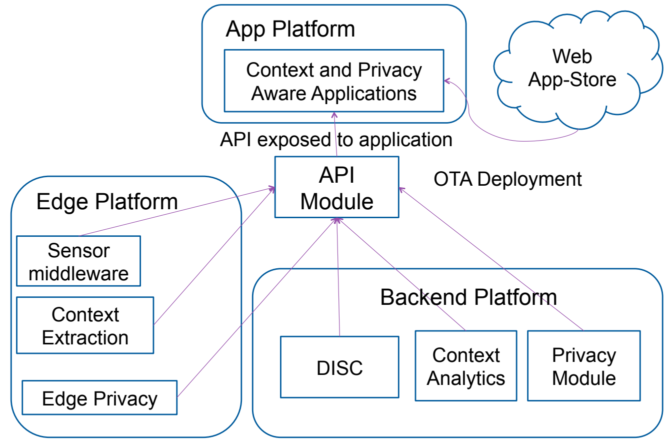

Fig. 1. Overall Platform Architecture

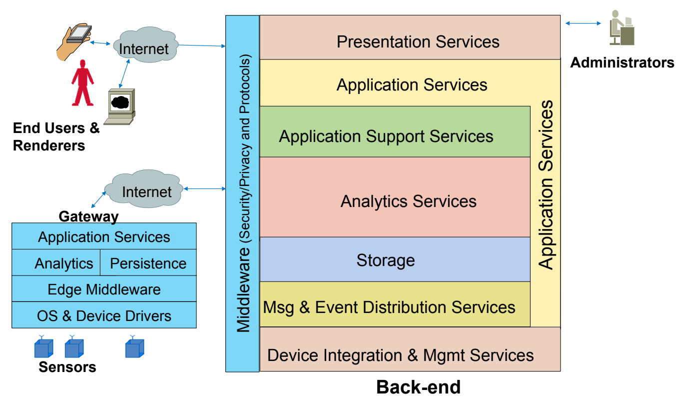

Fig. 2. Layers in Platform Architecture

A. Backend Platform

The backend platform is the sensor sink where all raw or processed sensor data terminates and are stored. Aggregated analytics are run on the sensor data in this sub-system. It has been built using 52 North based implementation of OGC (Open Geospatial Consortium) standards, an apache tomcat based web container and Play based lightweight web applications. Along with this Neo4J is being used as a graph database for caching social network data.

B. Edge Platform

The edge platform can be a Smartphone or other consumer device (like tablet, notebook) or dedicated embedded sensor gateway devices, based on the application use case. In context of the current application, the edge platform is the user's mobile phone which is carrying various applications.

C. Application Platform

The application platform is a logical platform which resides partially on the edge and partially on the backend sub-system. A part of the application that gathers data from individual sensors and runs some context extraction algorithm to extract meaningful feature vectors from raw sensor generated data,

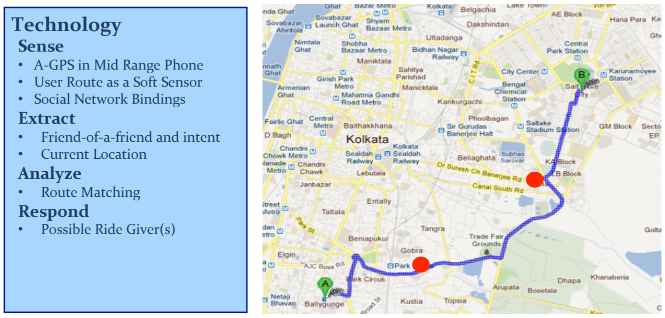

Fig. 3. Share A Ride Application Mapped on Platform

runs on the edge sub-system. However, the portion of the application that does aggregation or runs analytical algorithms like fusion, statistical modeling on the sensor or the extracted context information, typically runs on the backend sub-system. Since the platform is built for IoT and M2M(machine-to-machine) communications, applications that run on the platform typically follow the workflow state of sense-extract-analyze-respond. Figure 3 shows how the 'Share a Ride Application' has been mapped to the platform workflow.

IV. Application Details

This section focuses on the example application 'Share a Ride', its design, algorithms and experiments. The design subsection describes information flow of the application. The algorithm subsection discusses various approaches for Ride-Sharing application and their comparative advantages and shortcomings. In experiment subsection, details of experiments along with result screenshots are provided.

A. Application Design

To comply with the basic requirements for a ride share application as mentioned in [6], we have designed a Facebook application that allows ride givers to register their intent. While matching a ride share buddy, the application takes ride taker preferences and firstly identifies friend of friend for ride taker. This list is used to establish trust and it results in huge reduction of search space as well. Ride giver vehicle keeps on posting its location and timing details at regular intervals to sensor data repository as depicted in flowchart (Figure 4) labeled as B1 and B2. This repository provides route and timing details for the ride giver.

The starting point of the application is a web page which asks the ride giver to login using Facebook credentials labeled as A1. The application caches ride giver preferences along with friend details from Facebook in a local graph database labeled as A2 and A3. Facebook id is used to index the data. As new ride givers login to register their intent, the database grows and ride giver automatically connects with the friends having an entry in the existing database.

To find a ride giver, ride taker also needs to login to a Facebook application and provide preference for time and route. The application uses ride takers Facebook id and local graph database to identify their friend of friend list, which is

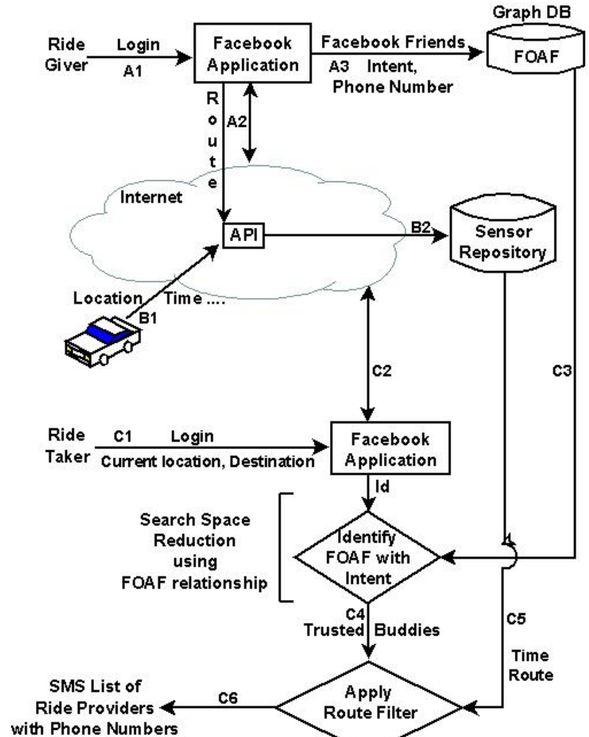

Fig. 4. Application Flowchart

their trusted potential buddy list. On this list, time filter and route filters are applied to get a final list of ride share buddies. These sequences of steps are labeled in flowchart (Figure 4) as C1 to C6.

B. Algorithms

In this subsection, we discuss three different approaches to find a ride share buddy along with their merits and shortcomings.

The Simple Approach is described in Algorithm 1.This approach matches route and time to suggest potential rideshare buddies, but it does not take care of the basic needs for a ride-share application viz. trust, intent and convenience as mentioned in [6]. Following are the major shortcomings of this approach:

- The growing number of application users will result in a huge search space.
- As, the intent or willingness to share the ride is not available, the list of potential ride-share buddies will be indicative.
- In this approach, trust is missing [6].

Algorithm 2 improves over the previous one by adding intent component. Ride givers declare their intent to share the ride by installing the application. Addition of intent has following implications:

- Explicit intent can be used as a first level filter to reduce the search space. However, this reduction in search space is limited by the number of users who have declared their intent.

Algorithm 1 Simple Approach

INPUT:
PR: Potential Ride Providers List
Rt(PR): Travelling Route
Tr(PR): Travelling Time Range
RS: Ridetaker Source
RD: Ridetaker Destination
RTr: Ridetaker's Timerange
OUTPUT:
PB:Potential Buddy List
PSEUDOCODE:

```
                                        ▷ For each ride provider
∀i ∈ PR
  loop
                          ▷ Apply time range & route match filters
    if RTr ⊂Tr(i) && RS ∈Rt(i) && RD ∈Rt(i) then
                              ▷ Append ride provider to buddy list
      PUSH [Rt(i), Tr(i)] to PB
    end if
  end loop
                                    ▷ Return potential buddy list
  return PB
```

Algorithm 2 Intent Based Approach

INPUT:
PR: Potential Ride Providers List
Rt(PR): Travelling Route
Tr(PR): Travelling Time Range
In(PR): Provider's Explicit Intent (Boolean)
RS: Ride Taker Source
RD: Ride Taker Destination
RTr: Ride Taker's Time Range
OUTPUT:
PBint: Potential Buddy List with Intent
PSEUDOCODE:

```
                            ▷ First level filter for explicit intent
∀i ∈ PR&&In(i) ≡ true
  loop
                          ▷ Apply time range & route match filters
    if RTr ⊂Tr(i) && RS ∈Rt(i) && RD ∈Rt(i) then
                              ▷ Append ride provider to buddy list
      PUSH [In(i), Rt(i), Tr(i)] to PBint
    end if
  end loop
                          ▷ Return potential buddy list with intent
  return PBint
```

- Even though, ride giver's willingness is known, but in absence of trust, one may not agree to share the ride. Trust component is missing and needs manual intervention.

Algorithm 3 adds trust component to Algorithm 2 to address the shortcomings in previous approach. It uses social network based friend of a friend (FOAF) relationship to establish trust. Following are the major improvements:

- FOAF is applied as a first filter to ensure that only trusted buddies are picked.
- Application of FOAF relationship as a first filter results in huge reduction of search space. [20] conducted Facebook social graph study on Nov. 2011, and provided following statistics related to Facebook friends.
  - An average Facebook user has around 190 reciprocal friends (excluding subscriptions).
  - A user with 100 friends has 27, 500 unique friends-of-friends.
  - The number of unique friends-of-friends grows almost linear. As per [20], a linear fit here will produce a slope of 355 unique friends-of-friends per additional friend.
- Using this data, we can project average number of unique friends-of-friends for an average Facebook user.
  - Friend count for average user = 190
  - FOAF count for a user with 100 friends = 27,500
  - Increase in FOAF count for an increment of 90 friends = 90 * 355 = 31950
  - FOAF count for an average Facebook user with 190 friends = 27,500 + 31,950 = 59,450
- We have also gathered popular Facebook application usage statistics. As per [1], Facebook applications are ranked based on monthly active user (MAU) count. MAU count for rank 1 application exceeds 41 million, whereas it exceeds 17 million for an application ranked 25. The above statistics suggest that number of users for a typical application will be much larger than the number of FOAF of a typical user on Facebook.
- In this approach, intent is explicitly declared and trust is also established. It successfully addresses the issues in the first two approaches.

C. Experiment

1) Setup: The application is implemented using java servlet, Facebook SDK(Software Development Kit), Google Map API(Application Programming Interfaces), Neo4j graph database, IoT platform services and Bulk SMS (Short Message Service) API. IoTSpaceReduce implementation spans across three different modules viz. Ride Giver module, Ride Taker module and Vehicle Sensor Data module.

Ride Giver Module provides a web interface to register intent by explicitly logging to this module using Facebook credentials. The module accesses and caches ride provider's basic information and friends list using Facebook javascript SDK. The ride provider is presented with another page (using Google maps JavaScript API) to mark their preferred route on Google map. It also accepts ride provider inputs regarding their time range for probable ride and phone number. The Facebook information gathered about the ride provider along with the friend list is forwarded to servlet which in turn stores it in

Algorithm 3 Intent augmented with Trust (FOAF)

INPUT:
PR: Potential Ride Providers List
Rt(PR): Travelling Route
Tr(PR): Travelling Time Range
In(PR): Provider's Explicit Intent (Boolean)
RS: Ride Taker Source
RD: Ride Taker Destination
RTr: Ride Taker's Time Range
RFl: Ride Taker's FOAF List (Trust Component)
OUTPUT:
PBit: Potential Buddy List with Intent and Trust
PSEUDOCODE:

▷ First level filter using FOAF for Search Space Reduction
$\forall i \in RFl$
  loop
    ▷ Apply Intent Filter
    if $In(i)$ ≡true **then**
      ▷ Apply time range & route match filters
      if RTr ⊂Tr(i) && RS ∈Rt(i) && RD ∈Rt(i) **then**
        ▷ Append ride provider to list
        PUSH $[RFl(i), In(i), Rt(i), Tr(i)]$ *to* PBit
      end if
    end if
  end loop
  ▷ Return potential buddy list with intent and trust
  return PBit

a Neo4j graph database. The ride provider's phone number and intent are stored as attributes of the ride provider in Neo4j graph database. The route and time range information is forwarded to sensor repository using IoT platform services.

Ride Taker Module provides a web interface, where one needs to login using Facebook credentials and provide inputs like time range and geographical locations for source and destination to avail the ride. The module fetches the user's information and friend list from Facebook account and forwards it along with user inputs (time range and source or destination points) to the servlet. Servlet stores Facebook information in Neo4j database and search for ride taker's FOAF with intent attribute as true. Application of this first level of filter leaves us with a list of potential trusted buddies with explicit intent. Now for each and every potential ride provider, time range filtering and route matching is applied using IoT platform services which results in a potential ride providers list with intent, trust, route and time range matching. This list augmented with phone number attribute is provided to ride taker as output. It is also sent to the ride taker as an SMS using Bulk SMS API services from telecom provider.

Vehicle Sensor Data Module is implemented as an application housed within OBD (On-board Diagnostic) sensor, installed in the vehicle. Whenever a vehicle is on the move,

Fig. 5. Snapshot of Full Social Network adapted from [4]

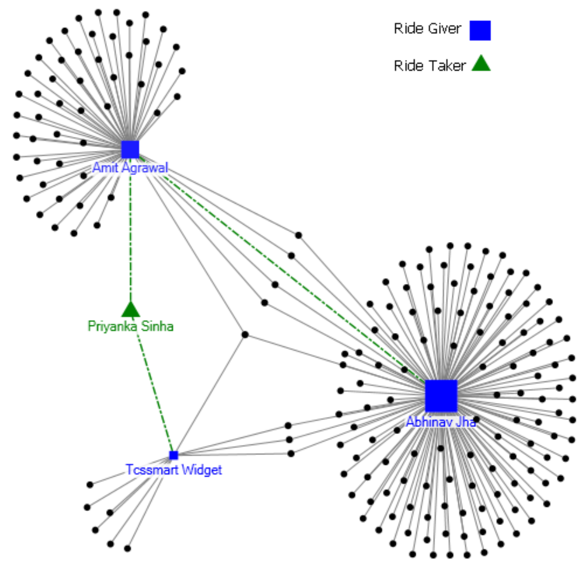

Fig. 6. Snapshot of Reduced Search Space adapted from [4]

this module tracks time, GPS location, speed and various other vehicle parameters at regular intervals and post them to the sensor repository using IoT platform services. Alternatively a smartphone in the car could also have been used as a sensor for the same purpose. This data is used for dynamic route matching.

2) Results: Figure 5 shows the network complexity when all the nodes are involved without any filtering mechanism in place. This depicts out an open space where any kind of combination is possible for matching the query result. Figure 6 shows the reduction in data points and eventual lessening of complexity when a FOAF based filtering approach is applied.

V. Conclusion

We have presented our results to show that IoTSpaceReduce uses FOAF associations and intent filters to effectively reduce search space for sensor data analytics. We have also explored how this approach can be extended to other use-cases. In near future, we plan to bring the same concept in the ubiquitous network of sensors themselves and prove that reduction in search space is possible whenever sensors have some kind of relationship between themselves other than existing social relationship of humans. We also plan to use social graph analysis in these networks to explore new possibilities.

References

[1] Monthly active user (MAU) estimate for Facebook apps from Appdata.

[2] Z. Aljazzaf, M. Perry, and M. Capretz. Online trust: Definition and principles. In *Computing in the Global Information Technology (ICCGI), 2010 Fifth International Multi-Conference on*, pages 163 –168, sept. 2010.

[3] C. Bhaumik, A. Agrawal, S. Adak, A. Ghose, and D. Das. Sensor observation service based medical instrument integration. In *SMART 2012, The First International Conference on Smart Systems, Devices and Technologies*, pages 48–54, 2012.

[4] C. Bhaumik, A. K. Agrawal, and P. Sinha. Using social network graphs for search space reduction in internet of things. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 602–603. ACM, 2012.

[5] M. Brereton, P. Roe, M. Foth, J. M. Bunker, and L. Buys. Designing participation in agile ridesharing with mobile social software. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*, OZCHI '09, pages 257–260, New York, NY, USA, 2009. ACM.

[6] V. Chaube, A. Kavanaugh, and M. Perez-Quinones. Leveraging social networks to embed trust in rideshare programs. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1 –8, jan. 2010.

[7] Y. Fu, Y. Fang, C. Jiang, and J. Cheng. Dynamic ride sharing community service on traffic information grid. In *Intelligent Computation Technology and Automation (ICICTA), 2008 International Conference on*, volume 2, pages 348 –352, oct. 2008.

[8] A. Ghose, C. Bhaumik, D. Das, and A. K. Agrawal. Mobile healthcare infrastructure for home and small clinic. In *Proceedings of the 2nd ACM international workshop on Pervasive Wireless Healthcare*, pages 15–20. ACM, 2012.

[9] A. Ghose, P. Sinha, C. Bhaumik, A. Sinha, A. Agrawal, and A. Dutta Choudhury. Ubiheld: ubiquitous healthcare monitoring system for elderly and chronic patients. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 1255–1264. ACM, 2013.

[10] K. Ghoseiri, A. Haghani, and M. Hamedi. Real-Time Rideshare Matching Problem, jan. 2011.

[11] J. Golbeck and J. Hendler. Filmtrust: movie recommendations using trust in web-based social networks. In *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, volume 1, pages 282 – 286, jan. 2006.

[12] S. H. Mirisaee. A human-centred context-aware approach to develop open-standard agile ridesharing using mobile social networks. In *Proceedings of the 22nd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction*, OZCHI '10, pages 445–448, New York, NY, USA, 2010. ACM.

[13] A. Mtibaa and K. Harras. Social-based trust in mobile opportunistic networks. In *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, pages 1 –6, 31 2011-aug. 4 2011.

[14] A. Niels, E. Alan, S. Martin, and W. Xing. Sustainable passenger transportation: Dynamic ride-sharing. *ERIM Report Series Reference No. ERS-2010-010-LIS*, Available at SSRN: http://ssrn.com/abstract=1568676, 2 2010.

[15] A. Pal, C. Bhaumik, J. Shukla, and S. Kolay. Energy information gateway for home. In *Intelligent Systems, Modelling and Simulation (ISMS), 2011 Second International Conference on*, pages 235–240. IEEE, 2011.

[16] A. Pal, C. Bhaumik, P. Sinha, and A. Ghose. Intelligent social network of devices. *Computational Social Networks*, pages 329–348, 2012.

[17] J. Park, Y. Cheng, and R. Sandhu. Towards a framework for cyber social status based trusted open collaboration. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010 6th International Conference on*, pages 1 –8, oct. 2010.

[18] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks*, 1(2):215–233, 2003.

[19] C. Stach. Saving time, money and the environment - vhike a dynamic ride-sharing service for mobile devices. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 352 –355, march 2011.

[20] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the facebook social graph. *Arxiv preprint arXiv:1111.4503*, 2011.

[21] X. Wang, F. Liu, R. Yang, and F. Xie. Social computing-based trust establishment in e-commerce. In *Internet Computing in Science and Engineering, 2008. ICICSE '08. International Conference on*, pages 246 –249, jan. 2008.

[22] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 148–159. ACM, 2002.

[23] T. Zhou and L. Harn. Security in user-assisted communications. In *Wireless Telecommunications Symposium, 2007. WTS 2007*, pages 1 –10, april 2007.