Middleware for Mobile Social Networks: A Survey

Ahmed Karam and Nader Mohamed
Faculty of Information Technology
UAEU, Al Ain, P.O. Box 17551, UAE
{ahmed.karam, nader.m}@uaeu.ac.ae

Abstract
In recent years there have been two major but independent trends: the popularity of social networking applications and the adoption of smartphones. As a result, a new trend of networks labeled Mobile Social Networks (MSN) has emerged and attracted considerable attention from the academic and industrial communities. Middleware for MSN facilitates the design and development of new applications on top of the middleware. But due to the resource constraints of smartphones, middleware for MSN presents a number of new challenges. The middleware for MSN should address the constraints of mobile devices, i.e., limited power sources, low memory capabilities, limited processing power, and heterogeneity. In this paper, we provide a detailed review of the existing middleware projects for MSN and identify the major approaches and challenges for designing them. Furthermore, we present an exhaustive comparative study of different middleware projects for MSN. We also discuss some open research issues in this important area of research.

1. Introduction
The advent of Web 2.0 and social networking applications caused a social phenomenon on the Internet, offering an unprecedented opportunity for a wide variety of real time applications. In recent years, many social networking applications such as Facebook [8], MySpace [30] and Twitter [35] have gained tremendous popularity among users worldwide. These applications have been very successful because they attracted millions of users some of who have little or no prior experience using online applications. Social networking applications allow users to keep in touch with families and friends. In addition, social networking applications can also be used for business purposes. One example of social networking applications being used for business purposes such as finding new career opportunity is LinkedIn [23].

In parallel with the emergence of social networking, there has been an increase in mobile devices adoption, in particular “smart phones”. Smart phones allow third-party developers to create different applications by providing free SDKs. Examples of the most well-known smart phone operating systems include Apple iOS, Google’s Android, BlackBerry OS, Windows Mobile, and Symbian OS. Interest in social networking and smart phones has tremendously increased in recent years. As a result, a new trend of networks labeled MSN has emerged. MSN has attracted a lot of attentions from researchers in both academic and industrial communities. Examples of some of the famous commercial MSN applications are Google Latitude [11], iPhone BreaCrumbs [15], and Here I Am [2]. Smartphones have limited resources compared to their desktop counterparts.

The middleware for MSN should address the constraints of mobile devices, i.e., limited energy, low memory capabilities, limited processing power, scalability, and heterogeneity [12]. Middleware for MSN should provide a layer that provides common services needed by different MSN applications and separate social-network management concerns from application requirements [4]. The middleware should enable efficient operations of MSN applications by being self-configuring, self-adapting, self-optimizing and self-protecting [24].

In recent years, many researchers have dedicated substantial effort in designing and developing different approaches for middleware for MSN. As a result of this, several middleware platforms have been developed for mobile social networks. MAgNet [3], MobiClique [31], MobiSoft [20], MyNet [19], MobiSoc [12], SAMOA [4], Semi-distributed Pervasive Social Networking Middleware [27], High-Level Abstraction Social Middleware [5], and Semantic and Reasoning Social Middleware [26] are some of the state of the art middleware projects for mobile social networks.

The purpose of this paper is to survey middleware platforms for MSN. We will cover three objectives: first, to identify the middleware challenges for MSN; second, to explore different relevant and state of the art middleware platforms for MSN and discuss their advantages and disadvantages; and third, to provide an exhaustive comparative study of these middleware platforms.
A number of surveys have been conducted for identifying different middleware challenges and approaches for different networked systems such as mobile computing [10], ad hoc networks [13], wireless sensor networks [14] and networked robots [28]. In addition, some surveys were done on different issues of online social networks such as security [1] and usages [29]. However, none of the existing work surveyed the challenges and approaches of middleware for MSN.

The remainder of the paper is structured as follows. In Section 2, we provide a brief overview of MSN applications and discuss the most important challenges faced by middleware platforms for MSN. Section 3 describes several research projects in middleware for MSN. In Section 4, we highlight the existing MSN middleware projects with a comparison between all these middleware platforms and also discuss some open research issues. Finally, we conclude this paper in Section 5.

2. Challenges for Middleware for Mobile Social Networks

The development of MSN applications can be greatly simplified by the existence of the appropriate middleware support. Middleware must be responsible for collecting, organizing, processing and disseminating social data. Furthermore, middleware should also provide a flexible interface to access social data. This allows different application to access these social data. The design and development of a successful middleware layer for mobile social applications, however, is not trivial and must address a number of issues. In spite of their many appealing features and characteristics, MSN applications still face many challenges. In this section, we present a list of major challenges that should be addressed by middleware for social networking. In order to be successfully usable in the real world, the middleware for MSN must address the following challenges:

- **Simplifying the development process**: developing social networking applications for mobile phones is not an easy task [34]. Middleware platforms should simplify the development process of distributed MSN applications. In order to achieve this goal, the middleware should provide high-level abstractions with lightweight interfaces to mobile phone application developers. High-level abstractions will significantly simplify the developers’ task. Furthermore, the middleware should also facilitate application integration and reuse.

- **Energy Efficiency**: battery power represents the most important limitation that mobile phone application developers face in developing social networking applications [25]. MSN applications usually need to execute some resource-intensive tasks in real-time that can quickly drain the battery, leaving the mobile phone useless. Therefore, it is important to design energy-efficient social networking applications for mobile devices. MSN application developers should consider the energy consumption characteristic of each component of the mobile device while designing a social networking application. Hence, the middleware should provide mechanisms for efficient use of energy resource and perform well on resource-scarce mobile devices [12].

- **Privacy**: MSN applications deal with sensitive data, such as social relationships, activities and user preferences, which could be used to infer other sensitive information about the user over time [21]. Therefore, sensitive information about a user must be treated carefully and securely. The middleware platform must ensure that users cannot track each other and prevent indirect inferences [12]. The challenge, however, is to provide a simple yet powerful privacy technique while consuming relatively little resources. In addition, the middleware should define and enforce appropriate access control policies over social data exchange in order to ensure user’s privacy. These issues must be considered while developing new MSN services.

- **Scalability**: scalability is concerned with the ability of a system to handle increases in users and system resources. The middleware for social networking applications should be flexible enough to manage increases in the number of joining nodes without compromising system performance [32]. The limited computing power and battery energy of mobile devices need to be taken into account when considering future scalability requirements.

- **Fully distributed architecture**: current approaches for MSN applications are based on centralized architectures whereas the general architecture of mobile environments is naturally distributed. On a mobile environment, users must be able to access data anytime and anywhere even in the absence of a fixed network infrastructure. Centralized architectures for social networking applications are not appropriate for the mobile environment [32]. Therefore, middleware should be designed in a fully distributed fashion with no centralized control. Specifically, the middleware should be designed for
use in mobile ad-hoc networks without assuming centralized servers.

- **Heterogeneity and dynamic nature of mobile devices:** unlike traditional computing environments, mobile computing environments are more dynamic and heterogeneous. The mobile environment consists of a set of mobile devices, which have different processing powers, battery energy levels, network interfaces, operating systems and communication mediums. This makes building a social context, searching and accessing information and keeping social context up to date even more challenging [33]. In addition, the dynamicity of mobile environments makes the management of MSN very challenging [22]. Furthermore, enforcing security and privacy in dynamic and heterogamous environment is not an easy task as traditional security and privacy solutions are not designed to face mobility issues. Therefore, middleware for MSN should be designed in a fully distributed fashion while taking into account the heterogeneity and the dynamic nature of mobile devices, as well as users’ privacy concerns. The middleware should hide heterogeneity from applications and allow for adaptation in dynamic environments.

### 3. Different Middleware for Mobile Social Networks

As mentioned earlier, there are many researchers working on middleware solutions for MSN. As a result, a number of software design principles and research projects for MSN have been proposed in recent years. In this section, several middleware platforms that provide different approaches will be presented in detail and also evaluated to explore their advantages and disadvantages. A comparison of these approaches will be provided in the following section to help determine the best features for MSN middleware platforms.

#### 3.1. MAgNet

MAgNet [3] is an agent-based middleware developed at the University of Zagreb in Croatia to enable mobile users easily and efficiently to use Social Networking Services (SNSs). The aim of using an agent-based middleware is to minimize the interaction between the software agent and user, meaning that the software agent can act without direct intervention from users. This simplifies the use of SNSs on mobile phones. MAgNet was implemented using Java Agent DEvelepment framework (JADE). JADE is a software framework specifically designed for devices with limited resources, such as mobile phones. The middleware provides two types of services: Group Management and Planning Group Event. MAgNet enables users to create and manage their social relationships with other users and utilize these relationships to plan and manage group events. MAgNet middleware contains three types of agents: Graphical User Interface Agent (GA), User Agent (UA) and Social Agent (SA). Each software agent performs a number of tasks. The UA agent represents the user within the MAgNet middleware. The GA agent is an interface between the user and his/her UA. GA agent allows the user to communicate with other users through graphical interface (i.e. UA). In addition, GA agent also allows the user to access all the functionalities of the MAgNet middleware. The SA agent is responsible for providing social networking services within the MAgNet middleware. In order to ensure privacy of users, the middleware employs the concept of block lists, which allow mobile users to define a list of users who are forbidden to access users’ profile. However, since MAgNet design focused on issues such as minimizing interaction between user and software agent and user privacy, it did not take important issues such scalability and time efficiency of the proposed middleware into account. Another shortcoming of the MAgNet middleware is that it does not make use of users’ existing profiles on social networking sites such Facebook, Twitter, and MySpace.

#### 3.2. MobiClique

MobiClique [31] is a mobile social networking middleware that allows users to maintain and extend their connections in a virtual world through opportunistic encounters in physical world. The middleware leverages existing online social networks services (OSNs) and opportunistic encounters in real life to enable ad hoc interactions between mobile phones. MobiClique eliminates the need for any infrastructure connectivity by exploiting mobile ad hoc networks to disseminate content where Bluetooth enabled mobile phones communicates directly with each other and without the use of a wireless access point. MobiClique was implemented using C++ and C# to run on Windows Mobile based devices. The middleware does not depend on a central server to exchange information. Instead it relies on opportunistic contacts between mobiles users to disseminate content. MobiClique leverages existing social networks services to bootstrap the system. Currently the middleware only
retrieves user’s profile information from Facebook. MobiClique also provides a generic framework for developing new MSN applications on top of the middleware. However, due to the use of Bluetooth ad hoc networking, MobiClique will face scalability challenges. According to some studies, Bluetooth faces scalability problems [16][36]. Furthermore, issues such as privacy and processing overhead require further investigation and testing.

3.3. Mobisoft

The Mobisoft middleware [20] was developed in collaboration between Friedrich Schiller University Jena, the agent factory GmbH, and Godyo AG. Mobisoft is an agent-based middleware for managing peer-to-peer overlays in mobile ad-hoc networks. The middleware enables users to discover partners with common interest using peer-to-peer communication. The aim of the Mobisoft project is to facilitate, augment, and promote human social interaction by mobile devices during physical encounters. The proposed middleware is based on the idea of three different techniques like mobile agents, peer-to-peer networking and semantic descriptions of user profiles and preferences. Mobisoft does not rely on any central server to locate partners or exchange information with them. The middleware allows this because it utilizes ad hoc networks to disseminate information where Bluetooth enabled mobile phones communicates directly with each other. The MobiSoft middleware is constructed using three layers: the bottom layer, the middle layer, and the top layer. The bottom layer contains the micro kernel of the agent system. The middle layer consists of several plugins for agent system. The plugin provides an interface for agents as well as for other plugins to find platform services. Finally, the top layer contains the social mobile assistants. Social-mobile assistants are mobile agents that provide the business logic of the middleware. It is responsible for establishing groups based on common interests and exchanging user information with potential partners in social networks.

The notion behind an agent based approach is to let agents act on behalf of the user and search for potential partners in a virtual world. Mobile agents move autonomously from peer to peer to disseminate the information they carry. The project uses the Java-based Tracy2 mobile agent toolkit [6] to implement the middleware. The middleware employs JXTA [18] to publish and find information about services as well as available Tracy2 platforms. JXTA is an open source peer-to-peer protocol based on Java. Moreover, the middleware uses FOAF (Friend of a Friend Finder) profiles to describe user’s interests. This enable user agent to match the information received from roaming mobile agents with user’s interest on a semantic level. Mobisoft has many advantages over traditional middleware platforms; specifically, the middleware is easy to configure, decentralized and operates in personal areas network using wireless transmission techniques without the need for internet connectivity. Despite its advantage, Mobisoft has a number of drawbacks that may limit its use and adoption. Commonly, such limitations include restricting the wireless connectivity to only Bluetooth and lack of privacy control mechanisms.

3.4. MyNet

MyNet [19] is a secure P2P social networking platform of middleware and user interaction tools, which enables non-expert users to share their devices, services content and social contacts without requiring a central repository or infrastructure. The main goal of MyNet is to simply the deployment, management and use of distributed services. MyNet allows users’ social data and distributed services to be accessed and shared among friends and devices in real time without delay as they are created, directly from the users’ own personal devices. These devices can be several physical hops away from each other. MyNet is built on top of the Unmanaged Internet Architecture (UIA) P2P technology [9], which provides two main features: ubiquitous connectively and distributed device group management. UIA allows users to securely imprint their profiles and personal data to devices. Each device is uniquely identified by its Endpoint Identifier (EID), which acts as the permanent device address. In addition, EID is also used to encrypt communication. This enables users to share their social data in a secure fashion with their friends. Unlike other platforms which build authorization at the device level, MyNet builds authorization at the user level. MyNet provides authentication, authorization and fine-grained access control to protect user’s data. Furthermore, MyNet enables users to grant access to their data through Passlets which are best known as “passes” or “tickets”. This P2P social networking platform, however, has its own limitations. The availability of content and services could not be guaranteed as the devices may be turned off by users at anytime. Furthermore, the platform does not offer any connectivity guarantee due to the lack of infrastructure support by P2P networks.
3.5. MobiSoC

MobiSoC [12] is a middleware framework that supports rapid development and deployment of Mobile Social Computing Applications (MSCAs). The middleware offers a common platform for capturing, managing, and sharing the social state (people profiles, place profiles, people-to-people, and people-to-places affinities) of physical communities. Additionally, the middleware incorporates learning algorithms that infer previously unknown emergent geo-social patterns to augment the social state with newly discovered information. It also provides mechanisms to disseminate these social state data among mobile users while preserving privacy of users. MobiSoC is based on a centralized architecture where a trusted server collects and manages all social data and users communicate with the middleware through Wi-Fi connectivity. The middleware consist of a thin mobile client and an MSCA service where the client runs on mobile phones whereas the service runs on regular servers. The notion behind using a centralized architecture is to improve the responsiveness and energy efficiency of mobile devices. In a client-server architecture most of the processing happens on the server-side while very little processing occurs on the client side. As a result, the client consumes considerably less energy and gets fast response from the server when interacting with the middleware. However, due to the use of a centralized architecture, MobiSoC is therefore subject to single point of failure and scalability problems. Another limitation of MobiSoC is that it only models profiles of peoples and places and does not consider contents and events for social interactions.

3.6. SAMOA

Socially Aware and Mobile Architecture (SAMOA) [4] is a semantic context-aware middleware framework developed at the University of Bologna in Italy to enable mobile users create anytime, anywhere semantic context aware social networks. Specifically, the middleware provides logical abstractions that group together mobile users who are both socially and physically proximate and share common interests. The goal is to determine potential partners for social interaction based on user’s social profiles and physical proximity. The middleware integrates a set of common management facilities for managing location-aware social networks. SAMOA also provides application level visibility of social network activities. The middleware is based on two types of context visibility: Place Visibility and Profile Visibility. Place visibility is concerned with the visibility of the user’s physical place whereas profile visibility is concerned with the user’s requirements. More specifically, place visibility limits the discovery scope for social network extraction to users in the same place whereas the visibility of user profiles further refines the discovery scope to create personalized social networks. In addition, the middleware utilizes semantic matching algorithms for analyzing users’ profiles and inferring potential semantic compatibility. The middleware defines three management roles, namely Managers, Clients and Members. Managers are mobile user responsible for defining the discovery scope boundaries of their social networks. Clients are users located within the discovery scope boundaries and are eligible to become members of the manager’s social network. Finally, Members are users affiliated with a social network. Each mobile user can play all roles. SAMOA does not guarantee user’s privacy in untrusted ubiquitous environments.

3.7. Semi-distributed Pervasive Social Networking Middleware (Project 1)

This social networking middleware [27] is developed at the University College London to enable semi-distributed Pervasive Social Networking (PSN). PSN enables mobile users who are both socially and physically proximate to discover each other and perform activities of common interest (e.g., playing tennis). Bluetooth technology is used by each user’s device to discover other mobile devices. Each mobile device keeps track of regular encounters with other mobile devices in order to build its physical proximity network. The middleware consists of two types of nodes: Clients and Brokers. Clients are nodes that maintain updated information about users’ social preferences; while brokers are nodes responsible for gathering users’ social preferences and computing recommendations for the mobile users by employing social networking propagation algorithm. The middleware utilizes the Personalised PageRank (PPR) algorithm [7] as its intra-activity propagation mechanism to infer users’ relations within a single activity network. This is similar to ‘People You May Know’ service provided by the famous social networking website, Facebook. PPR algorithm uses a random walk over a graph of users to accurately infer missing social links in the social graph. The key features of the middleware are the availability and scalability of social networking services. However, this middleware has a couple of shortcomings. First, it does
not take the privacy of the users into consideration. Second, communication overhead increases as the number of brokers increase.

3.8. High-Level Abstraction Social Middleware (Project2)

This middleware [5] is developed at the University of Auckland in New Zealand to enable mobile devices to host mobile services that provides MSN services to different users of a social network. The main goal of this middleware is to allow rapid development of mobile service applications by higher level of abstraction. The middleware consist of two functional components: Communication and Service Hosting Infrastructure and Context-Aware Framework. The communication and service hosting infrastructure addresses problems confronted when hosting social networking services on 3G networks such as limited resources, limited power supply, low bandwidth and Intermittent connectivity. The context-aware framework allows gathering, processing and propagating context data such as user location (assisted GPS) and activity.

The middleware uses Jini Surrogate Architecture (JSA) [17] to provide mobile services. The JSA enables resource-limited mobile devices to participate in a Jini federation. Jini is an object-oriented service-based middleware architecture for service registration, discovery, and consumption. A Jini federation consists of clients, services and service locators in a distributed network. To participate in a Jini federation, mobile devices must have enough resources to run the Jini software. A host that meets all the requirements of the Jini federation is called Surrogate host. Without the JSA, mobile devices with limited resources are unable to participate in a Jini federation since they do not meet the requirements of the Jini federation. The middleware provides HTTP and Bluetooth-based interconnect to enable host devices to communicate with their surrogates. Some of the key features of the middleware are scalability, availability, responsiveness and minimum resource consumption. However, problems such as communication overhead, privacy and security require further investigation.

3.9. Semantic and Reasoning Social Middleware (Project3)

This middleware [26] is developed at University College London to provide semantic specification and reasoning on users’ tasks. More specifically, the authors propose a Pervasive Social Computing (PSC) model for the semantic specification of users’ tasks to assist users in realizing their tasks by reasoning on their social links and recommending them other users who share similar interests. In addition, the authors present a set of algorithms for matching these specifications on users’ social preferences. The proposed Pervasive Social Computing middleware is divided in three layers. The Semantic Middleware layer which provides two functionalities: Semantic Task Specification and Efficient Semantic Reasoning. The first functionality handles the heterogeneity of users’ social preferences and task specifications while the second functionality handles the scarcity of device resources. The Social Computing Middleware layer which includes two functionalities: Distributed Social Network Management and Task Matching. The first functionality supports the propagation of social preferences among users in order to guess missing links from existing ones while the second functionality performs matching of users’ tasks by considering their social tights. Finally, the Communication Middleware layer which consists of three functionalities, namely the Task Publication, Task Dissemination and Task Notification. The first functionality injects a user’s task in the network while the second functionality disseminate user’s task to other nodes and finally the third functionality communicates the result back to the user. The middleware extends FOAF (Friend of a Friend Finder) ontology to support multi-activity social networking, where users’ specify their interest in performing different activities with other users’. Despite its many advantages, this middleware has a couple of shortcomings. First, the middleware does not provide privacy of user sensitive tasks and preferences. Second, the middleware does not guarantee timely response to users’ requests. The user may receive a response to his/her request after hours which make the middleware ineffective.

4. Discussion and Open Issues

In the previous section we surveyed several existing middleware approaches for MSN. There are many types of middleware platforms for MSN, each proposing different solutions to the above listed problems. Each of these middleware approaches has its own strengths and weaknesses. However, these middleware approaches remain incomplete, with no single middleware infrastructure providing an integrated approach to solving all the problems of MSN. Before proceeding further, we need to specify the criteria we will use as the basis for evaluating these middleware approaches. To facilitate our discussion we
will evaluate these projects based on their main features which include solutions for the technical challenges presented in Section 2. We will add to these evaluation criteria some extra features that provide some social aspect support. The resulting list consists of nine main features which form the basis for comparing the different middleware approaches:

1. Enabling the rapid development of new MSN applications on top of the middleware by providing a generic application development framework, high level abstractions, and component-based development. Few projects have taken this goal into consideration, for example, MobiSoc and Project 3.

2. Utilizing decentralized architecture where mobile users should be able to communicate with each other anytime and anywhere without a need for centralized servers or central control. Many projects utilize a fully distributed architecture such as MAGNet, MobiClique, MobiSoft, MyNet, SAMOA and Projects 3 whereas few projects provide partially distributed architecture such as Project 1 and Project 2.

3. Integration with social networking sites such as Facebook where a middleware retrieves user’s social profile and uses this profile information to find other users. Out of nine projects discussed in this paper, only MobiClique provides this functionality.

4. Supporting heterogeneity where mobile devices with different processing powers, networks interfaces, operating systems and battery capacities can use the middleware to communicate with each other. Only two middleware projects solve the heterogeneity problem: SAMOA and Project 3.

5. Ensuring user privacy while disseminating user’s personal data and specially location based information by using flexible and easy to configure privacy mechanisms. Some projects implemented proper privacy mechanisms, for example, MAGNet, MyNet and MobiSoc.

6. Saving energy through better utilization of mobile devices resources and consumption, where the middleware should provide mechanisms for efficient use of energy resource and performs well on resource-scare mobile devices. Only a few middleware projects have addressed energy efficiency, for example, MobiSoc and Project 3.

7. Association between geographical and social based information where the middleware allows mobile users to discover other users who share same interests within physical proximity of him/her so they can easily communicate with each other. Many middleware projects have implemented this functionality, for example, MobiSoft, MobiClique, MyNet, MobiSoc, SAMOA, Project 1, Project 2 and Project 3.

8. Ability to infer previously unknown social patterns where the middleware utilizes some mechanisms to infer user’s missing social links in the social graph and also updates the user’s social state with newly discovered information. Several projects have implemented this mechanism, for example, MobiSoc, SAMOA, Project 1, Project 2 and Project 3.

9. Providing enough scalability for serving the increasing number of joining mobile devices without effecting the overall performance and energy consumption of the mobile devices. Many middleware projects provide enough scalability such as MobiSoft, MyNet, SAMOA and Projects 3 whereas few projects provide some scalability such as Project 1 and Project 2.

Although we listed examples for each feature, many middleware projects implemented several of these features with varying degrees. Of the nine middleware projects presented in this paper, Project 3 is the one with more features followed by both MobiSoc and SAMOA. However, as mentioned earlier, no single middleware platform solves all the problems of MSN. Table 1 presents a side-by-side comparison of features of all the middleware projects surveyed in the paper.

Furthermore, we have also identified several open issues that were either not addressed by the existing middleware projects or require further improvements. One major issue is to provide effective security mechanisms tailored for MSN. As traditional security mechanisms are not designed to face mobility issues. Security issues were not taken into consideration by many middleware projects. Therefore, further research to come up with effective security mechanisms are of crucial importance. Another main issue is to provide simplified and rapid development framework to facilitate the development of MSN applications on top of the middleware. Furthermore, middleware approaches should work well in heterogeneous mobile environments where mobile devices have different processing powers, battery energy levels, network interfaces, operating systems and communication mediums. In addition, the middleware approaches should also work well in dynamic environments where mobile devices join and leave the systems frequently. This means that the middleware should provide efficient mechanisms to cope with this scenario and also with different kinds of failures and faults within mobile device’s software and hardware.
5. Conclusion

In this paper, we presented a comprehensive review of the existing work on MSN middleware. We described and discussed the characteristics of existing MSN middleware projects as well as the differences between different approaches in detail. Furthermore, we also discussed the main middleware challenges for MSN. As a result, we identified several open issues that require to be fully addressed before we can design a comprehensive middleware solution for MSN. MSN is a new and rapidly evolving field and have attracted the attention of both academic and industrial researchers. Therefore, additional capabilities are constantly being introduced into the design of MSN middleware. Different middleware platforms have different features and functions such as ensuring user’s privacy, ability to infer previously unknown social patterns, providing application development on top of the middleware and integration with social networking sites. However, these middleware approaches remain incomplete, with no single middleware infrastructure providing an integrated approach to addressing all the issues of MSN. Finally we argue that there is still a lot that needs to be done before creating a comprehensive middleware solution for MSN. The full benefits of MSN middleware platforms are more likely to be realized when both academic and industrial researchers start working together.

TABLE I: A SUMMARY OF THE FEATURES OF THE MIDDLEWARE FOR MOBILE SOCIAL NETWORKS

<table>
<thead>
<tr>
<th>Feature</th>
<th>Projects</th>
<th>MAgNet</th>
<th>MobiClique</th>
<th>Mobisoft</th>
<th>MyNet</th>
<th>MobiSoc</th>
<th>SAMOA</th>
<th>Project 1</th>
<th>Project 2</th>
<th>Project 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit privacy features</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Decentralized architecture</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Some</td>
<td>Some</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Energy Efficiency</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Provide Scalability</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Some</td>
<td>Some</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Offer Heterogeneity</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Provide application development</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Integration with social networking sites</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Association between geographical and social based information</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Ability to infer previously unknown social patterns</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

REFERENCES


[18] JXTA (http://www.jxta.org)


