International Journal on Advanced Science Engineering Information Technology

Vol.6 (2016) No. 6
ISSN: 2088-5334

The Design and Implementation of Middleware for Application Development within Honeybee Computing Environment

Nur Husna binti Azizul*1, Abdullah bin Mohd Zin*2, Elankovan Sundararajan*3

*Research Center for Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600, Bangi, Selangor, Malaysia
E-mail: 1husna2005.nh@gmail.com , 2amzftsm@ukm.edu.my, 3elan@ukm.edu.my

Abstract— **Computing technology is now moving from ubiquitous computing into advanced ubiquitous computing environment. An advanced ubiquitous environment is an extension of ubiquitous environment that improve connectivity between devices. This computing environment has five major characteristics, namely: large number of heterogeneous devices; new communication technology; mobile ad hoc network (MANET); peer-to-peer communication; and Internet of Things. Honeybee computing is a concept based on advanced ubiquitous computing technology to support Smart City-Smart Village (SCSV) initiatives, which is a project initiated within Digital Malaysia. This paper describes the design and implementation of a middleware to support application development within Honeybee computing environment.**

Keywords— **Software Development Kit; Android; REST API; Honeybee Computing**

I. Introduction

Computing technology consists of a number of elements such as user devices, network, servers, and software. Changes in any of the elements introduce new phases of computing technology. Currently, computing technology is moving from ubiquitous computing into advanced ubiquitous computing environment. An advanced ubiquitous environment is an extension of ubiquitous environment that improves connectivity between devices. The major characteristics of this environment can be stated as follows: (i) Large number of heterogeneous devices; (ii) New communication technology; (iii) Mobile ad hoc network (MANET); (iv) A peer to peer communication; and (v) Internet of Things.

Heterogeneous devices include devices such as notebook computers, tablets, smartphones and wearable computers are common user devices [1]. Most of these devices operate under many different operating systems such as Windows, Linux, Android or iOS.

The new communication technology refers to an enhancement of communication technology such from 3G to 4G, IPv4 to IPv6 and Near Field Communication (NFC). The introduction of 4G is expected to provide a more stable communication connection between devices [2], [3]. NFC mainly based on Radio-frequency identification (RFID) technology enables a set of devices to establish radio communication with each other by touching or putting the device within the range provided that is not more than a few inches. Problems with IPv4 such as address shortages, routing scalability, and broken end-to-end property are expected to be overcome with the introduction of IPV6 [4].

Mobile ad hoc network (MANET) is different from the conventional network as it does not require a fixed infrastructure. Nodes in MANET are free to move and organize themselves in an arbitrary fashion [5]. In a peer to peer network concept, a device acts as a server and as a client at the same time. Another characteristic of advanced ubiquitous computing is Internet of Things (IoT). IoT extends Internet connectivity beyond traditional devices to a diverse range of devices and everyday things.

Honeybee computing is a concept based on advanced ubiquitous computing technology to support Smart City-Smart Village (SCSV) initiatives. SCSV is a project initiated within Digital Malaysia that is basically an extension of the Multimedia Super Corridor initiative that was introduced in the 1990s. The main objective of SCSV is to create a new benchmark in urban and rural development at strategic locations nationwide. It involves consolidation and enhancement of existing initiatives carried out by the Malaysian government [6]. The concept of Honeybee Computing is derived based on beehive model. A bee represents any computing devices such as mobile, PC, sensors, and device with embedded software. The bee can communicate to the internet via the wireless network. Some of the bees (from two to a thousand) can create ad hoc communication between them to implement a certain task.

Honeybee computing is an extension of ubiquitous environment by adding peer to peer communication and ad hoc network.

In this paper, we present the design and implementation of a middleware to support advanced ubiquitous computing environment. The remainder of the paper is organized as follows. Section II presents the proposed design for the middleware development . Next, in section III, the honeybee middleware is described. Finally, section IV summarizes the paper.

II. Materials And Methods

Middleware is a software that helps to integrate hardware and software. Each computing technology is supported by a specific middleware. For example, Globus is a middleware that support Grid Computing, while Alljoyn [7] provides middleware for ubiquitous computing.

In order to design and implement the middleware, we have considered a number of similar researches. One of them is concerning the design and implementation of RFID middleware since there are a lot of researches in this area that have been carried out. The design of a lightweight RFID middleware is presented by Lin et al [9]. Another middleware that uses RFID technology is WinRFID [10] that is specifically designed as a platform for .NET applications.

Many middleware's are designed based on message-oriented approach [11]. A message-oriented middleware (MOM) allows application modules to be distributed over heterogeneous platforms and reduces the complexity of developing applications that involve multiple devices, operating systems, and network protocols. Another approach for designing a middleware is by taking advantage of the virtualization and transparency of Cloud computing [12].

Middleware is located in the server, where the functionality involves task inside the server. To bridge the client side with the middleware, a software development kit (SDK) is normally provided. For example, Facebook SDK enables the programmer to access data inside the Facebook's server [13]. Spotify provides SDK and web service for a developer to use the music playlist [14]. The Qualcomm LTE Broadcast SDK gives developers the power to bring LTE Broadcast connectivity and content to their apps [15]. SDK normally provides APIs and interface to make it easier for application to be developed. Most SDKs also include tools, libraries, documentation and sample codes. SDK is normally designed for a specific server since the type of data is different between servers.

III. Results And Discussion

A. Middleware Design

As shown in Fig. 1, the general architecture of Honeybee Computing environment contains various devices that can communicate between one another through various apps. These apps can be supported by a number of tools such as predictive analytic and semantic knowledge tool. Data collected from various sources will be analysed by semantic knowledge tool and will then be stored in a knowledge base. Data in the knowledge base can then be used directly by the apps or can be further analysed by a predictive analytic tool.

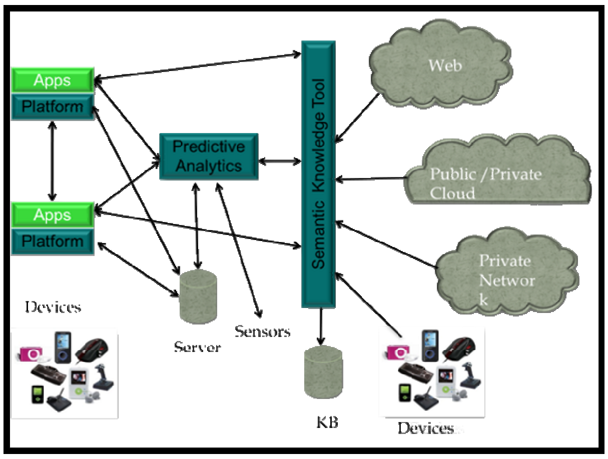


Fig. 1 Honeybee computing

In honeybee environment the user devices act as a node. The structure of the node is shown in Fig. 2.

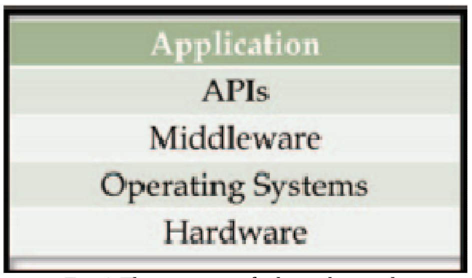


Fig. 2 The structure of a honeybee node

The architecture of Honeybee middleware is shown in Fig. 3. It consists of six managers to support six basic functionalities, namely Service Manager, Communication Manager, Security Manager, Semantic Manager, and News Manager.

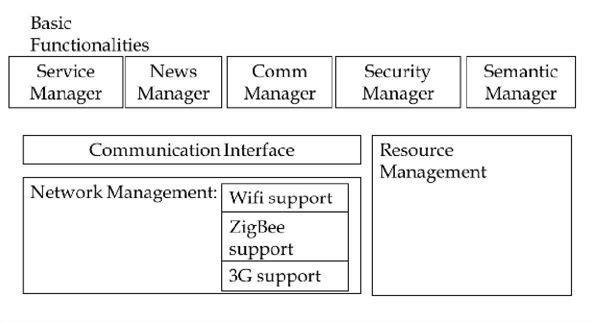


Fig. 3 The architecture of honeybee middleware

The design of these managers is described in the following subsections.

1) Service Manager: A honeybee device needs to communicate with other honeybee devices in a neighbourhood through an ad hoc network. It also needs to communicate with servers through the Internet. In order for a honeybee device to be recognized, it needs to be registered with the Honeybee server.

The general architecture of the middleware is shown in Fig. 4.

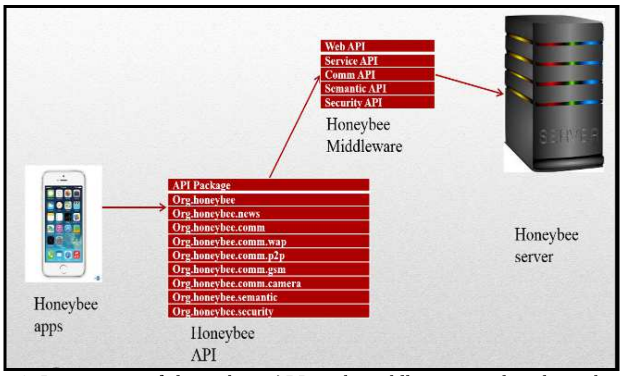


Fig. 4 Interaction of honeybee API and middleware within honeybee computing

Fig. 4 shows the interaction of honeybee middleware that located in the server and honeybee API that located in the mobile devices. The middleware enables data from the server to be accessed by the honeybee API, but to access the middleware, a correct request must be sent to the server, so that it will return an output. To send the correct request to the middleware, honeybee API will do the job. Honeybee API provides the request to be sent to the middleware. Honeybee API also will process the output so that it can be used by the application.

Fig. 5 shows the interaction of honeybee API and middleware inside service manager. The API is part of the SDK that will be used during honeybee application development.

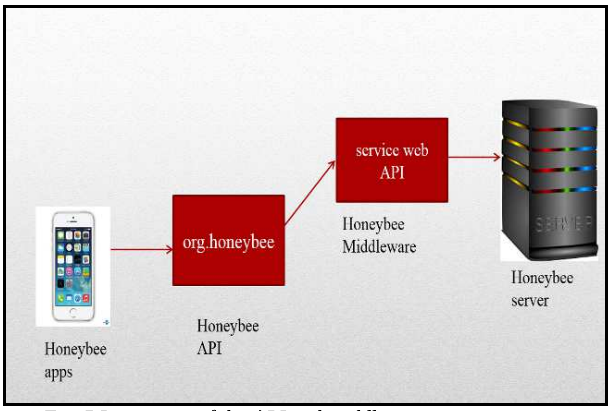


Fig. 5 Interaction of the API and middleware in service manager

The main role of the service manager is to manage device registration with the server. It also helps Honeybee device to recognize other honeybee devices in a neighbourhood. Fig. 6 shows the registration process for a mobile device.

2) *News Manager:* One of the important facilities provided within Honeybee computing is the ability to obtain latest information about certain issues. This facility can be implemented by using the News Manager. Fig. 7 shows two news servers that can be accessed through honeybee API. The news API provides bridge between API to the news server, for example, multiple local news such as Berita Harian (www.bharian.com.my), News Strait Times (www.nst.com.my) and Utusan Malaysia (www.utusan.com.my).

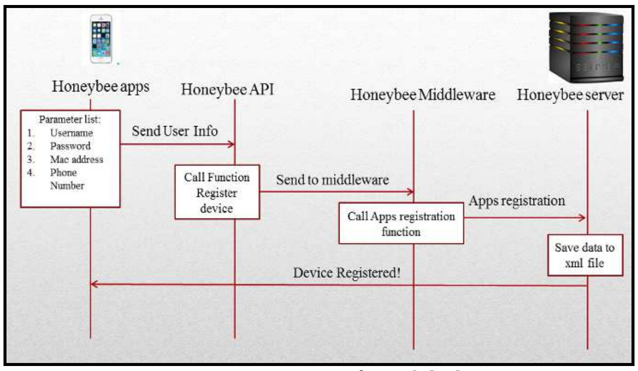


Fig. 6 Device registration for mobile device

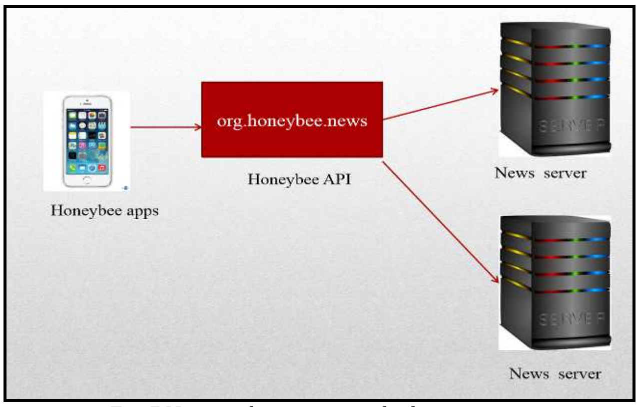
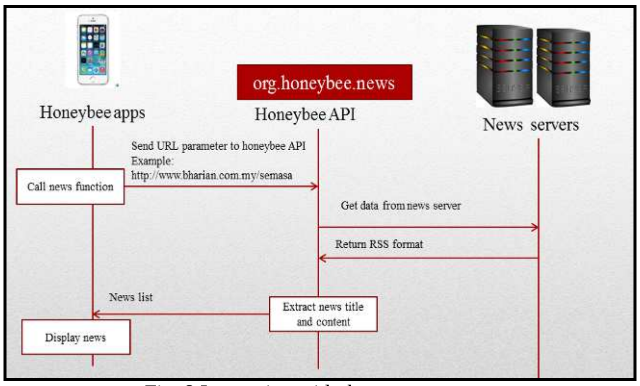


Fig. 7 News package access multiple news servers

Fig. 8 shows the application interaction with API and the server. The request sent by the API is a function that is used in honeybee apps.


org.honeybee.news
Honeybee apps
Honeybee API
News servers
Send URL parameter to honeybee API
Example:
http://www.bharian.com.my/semasa
Call news function
Get data from news server
Return RSS format
News list
Extract news title and content
Display news
Fig. 8 Interaction with the news manager

3) *Communication Manager:* Communication Manager provides communication for three type of network, which are W.A.P, p2p, and gsm. The application will first check the type of network needed and used suitable protocol for the communication. Fig. 9 shows the flow of the protocol processes under different types of network.

Different protocols will enable different data to be sent without causing any conflicts. Since different protocols support different types of data thus the package of the API classification are based on the network name. This package contains functions for sending and receiving messages through different network connections, which are: wireless network, GSM network and p2p network, as shown in Fig. 10. GSM network is provided since it is the most popular technology and widely used.

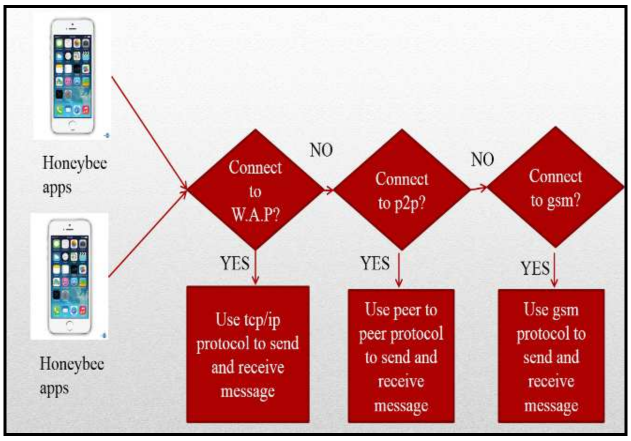

Honeybee apps
Honeybee apps
Connect to W.A.P?
NO
Connect to p2p?
NO
Connect to gsm?
YES
YES
YES
Use tcp/ip protocol to send and receive message
Use peer to peer protocol to send and receive message
Use gsm protocol to send and receive message

Fig. 9 List of network and protocol provided by honeybee API

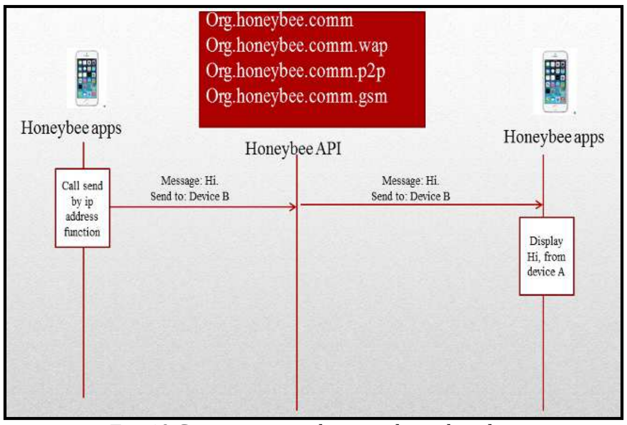


Fig. 10 Communication between honeybee devices

A function embedded in honeybee application send messages to another device that provides the message and the source of the device. The information sent is then received by another device and the embedded will then display messages from the sender. List of functions provided by the Communication manager is shown in Table 1.

TABLE I
LIST OF FUNCTIONS PROVIDED BY THE COMMUNICATION MANAGER

Functions	**Description**
Discover neighbour devices	To discover a list of the devices in current Wi-Fi access point. List of device is in XML format to save memory
Send message to IP address	To send a message to other device based on IP address.
Broadcast message	To broadcast message to group device in list
Get device list to broadcast	To provide list of device to broadcast

GSM only allows messages to be sent in a text format. In order for images to be sent through GSM network, an additional function for converting text to image is also provided. Communication manager also can be used as a communication between mobile devices and other devices such as an IP camera. List of functions provided for a camera is shown in Table 2.

TABLE II
LIST OF FUNCTIONS PROVIDED FOR CAMERA

Functions	**Description**
View current state	To view the state of the camera
Register device.	To enable device to access the camera

Normally, cameras are located in the different network, communication between the mobile device and camera can only be done through a server as shown in Fig. 11.

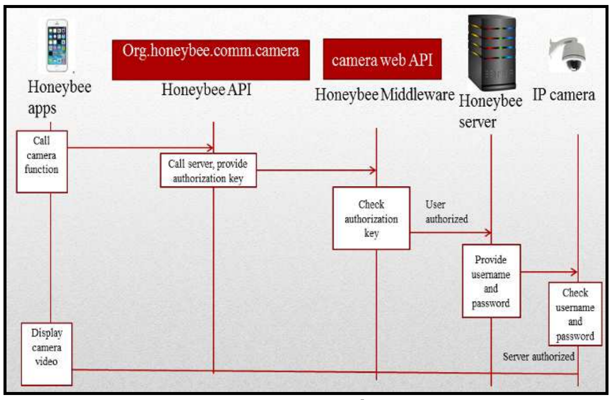


Fig. 11 Communication with an IP camera

Fig. 12 is the sequence diagram that shows the flow of data from multiple cameras.

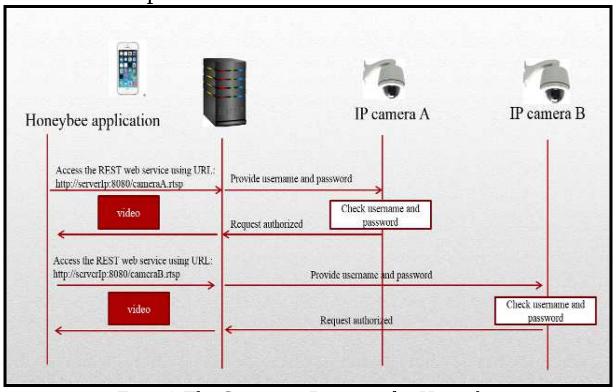

Honeybee application
IP camera A
IP camera B
Access the REST web service using URL: http://serverIp:8080/cameraA.rtsp
Provide username and password
video
Check username and password
Request authorized
Access the REST web service using URL: http://serverIp:8080/cameraB.rtsp
Provide username and password
video
Check username and password
Request authorized

Fig. 12 The Sequence Diagram for Honeybee

As shown in Fig. 12, each request provides the type of data needed, such as RTSP. The server will access the IP camera since it has username and password for each camera. Once the camera received the correct username and password, honeybee application can access the selected camera.

4) Semantic Manager: In honeybee computing, users can search information by using a semantic search engine. Semantic manager provides access to the semantic search engine. In order to obtain information, the app needs to send a query to the server which will then reply by giving a list of information that matched the query as shown in Fig. 13.

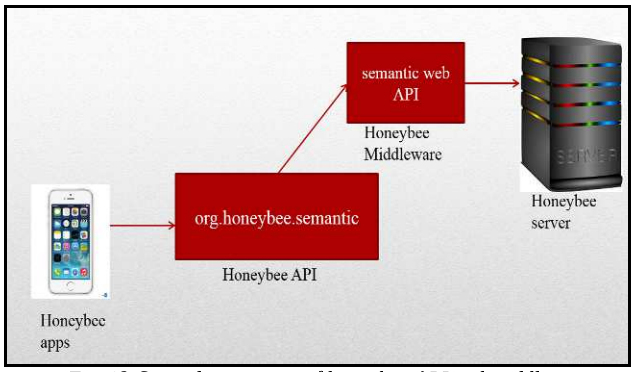


Fig. 13 General interaction of honeybee API and middleware

This diagram shows the interaction of the honeybee API using semantic package that communicates with the semantic middleware located at the server. The semantic middleware will then process the request and provide the output to the apps.

In order to obtain an information, the apps need to send a query to the server which will then reply by giving a list of information that matches the query as shown in Fig. 14.

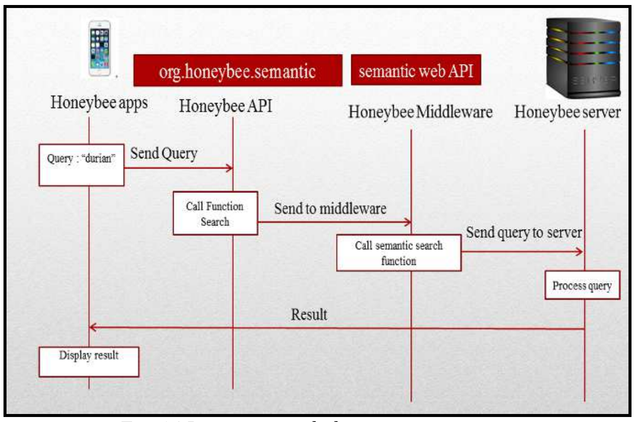


Fig. 14 Interaction with the semantic manager

B. *Implementation*

1) Client Side: In order to make it easier for developing an app, the SDK that support software development within the Honeybee computing is provided. The SDK provides interfaces as well as methods for manipulating the interface.

Thus to create a new app, a programmer needs to create a class and then initialize the SDK. To call a specific manager, an object representing that particular manager is created and the interface will also be provided. To do a specific task, a suitable method can be used. The classes provided by the SDK are identified by standard naming convention as shown in Table 3.

TABLE III
NAMING CONVENTION

Package	Description
Org.honeybee	Service Manager
Org.honeybee.news.	News manager
Org.honeybee.comm	Communication manager
Org.honeybee.comm.wap	Communication manager for wireless communication
Org.honeybee.comm.gsm	Communication manager for GSM network
Org.honeybee.comm.p2p	Communication manager for peer to peer communication
Org.honeybee.comm.camera	Communication manager for the camera
Org.honeybee.semantic	Semantic manager

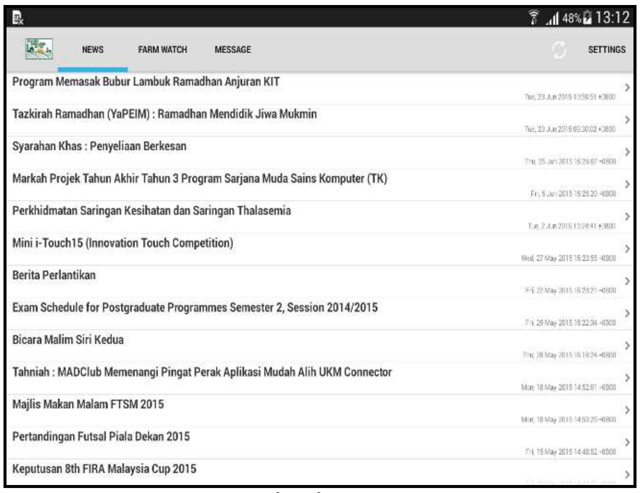
Some of the classes provided for each package are also shown in Table 4.

TABLE IV
LIST OF CLASS FOR EACH PACKAGE

Packages	List of class
Org.honeybee	Server.java
Org.honeybee.news.	AddURL.java ListSource.java newsSection.java
Org.honeybee.comm.wap	IncomingFragment.java OutFragment.java
Org.honeybee.comm.gsm	Convert.java
Org.honeybee.comm.p2p	IncomingPeerFragment.java OutPeerFragment.java
Org.honeybee.comm.camera	Camera.java
Org.honeybee.semantic	Semantic.java

In order to manipulate the News manager, the following steps are needed:

1. Select Org.honeybee.news package.
2. Create a new object for NewsSection and name it, for example, newsection1.
3. This particular class has three methods: refresh, addnewserver, and listserverURL.
4. An interface for a News manager is provided as shown in Fig. 15.



Program Memasak Bubur Lambuk Ramadhan Anjuran KIT
Tue, 23 Jun 2015 12:59:51 +0800

Tazkirah Ramadhan (YaPEIM) : Ramadhan Mendidik Jiwa Mukmin
Tue, 23 Jun 2015 03:30:02 +0800

Syarahan Khas : Penyeliaan Berkesan
Thu, 25 Jun 2015 15:26:07 +0800

Markah Projek Tahun Akhir Tahun 3 Program Sarjana Muda Sains Komputer (TK)
Fri, 5 Jun 2015 15:25:20 +0800

Perkhidmatan Saringan Kesihatan dan Saringan Thalasemia
Tue, 2 Jun 2015 15:24:41 +0800

Mini i-Touch15 (Innovation Touch Competition)
Wed, 27 May 2015 15:23:55 +0800

Berita Perlantikan
Fri, 22 May 2015 15:23:21 +0800

Exam Schedule for Postgraduate Programmes Semester 2, Session 2014/2015
Fri, 29 May 2015 15:22:34 +0800

Bicara Malim Siri Kedua
Thu, 28 May 2015 16:18:24 +0800

Tahniah : MADClub Memenangi Pingat Perak Aplikasi Mudah Alih UKM Connector
Mon, 18 May 2015 14:52:01 +0800

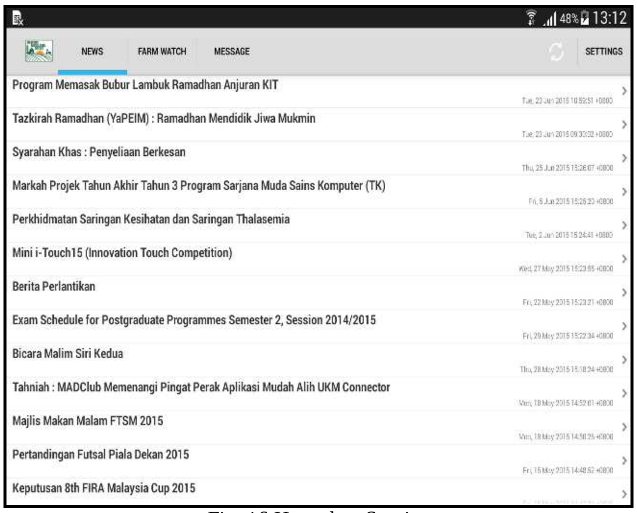
Majlis Makan Malam FTSM 2015
Mon, 18 May 2015 14:50:25 +0800

Pertandingan Futsal Piala Dekan 2015
Fri, 15 May 2015 14:48:52 +0800

Keputusan 8th FIRA Malaysia Cup 2015

Fig. 15 Interface for News Manager

2) Device to Device Communication: Honeybee service that involves communication between devices is implemented by using Android API as shown in Fig. 16.



Program Memasak Bubur Lambuk Ramadhan Anjuran KIT
Tue, 23 Jun 2015 10:52:51 +0800

Tazkirah Ramadhan (YaPEIM) : Ramadhan Mendidik Jiwa Mukmin
Tue, 23 Jun 2015 09:33:02 +0800

Syarahan Khas : Penyeliaan Berkesan
Thu, 25 Jun 2015 15:26:07 +0800

Markah Projek Tahun Akhir Tahun 3 Program Sarjana Muda Sains Komputer (TK)
Fri, 5 Jun 2015 15:25:23 +0800

Perkhidmatan Saringan Kesihatan dan Saringan Thalasemia
Tue, 2 Jun 2015 15:24:41 +0800

Mini i-Touch15 (Innovation Touch Competition)
Wed, 27 May 2015 15:23:55 +0800

Berita Perlantikan
Fri, 22 May 2015 15:23:21 +0800

Exam Schedule for Postgraduate Programmes Semester 2, Session 2014/2015
Fri, 29 May 2015 15:22:34 +0800

Bicara Malim Siri Kedua
Thu, 28 May 2015 15:18:24 +0800

Tahniah : MADClub Memenangi Pingat Perak Aplikasi Mudah Alih UKM Connector
Mon, 18 May 2015 14:52:01 +0800

Majlis Makan Malam FTSM 2015
Mon, 18 May 2015 14:50:25 +0800

Pertandingan Futsal Piala Dekan 2015
Fri, 15 May 2015 14:48:52 +0800

Keputusan 8th FIRA Malaysia Cup 2015

Fig. 16 Honeybee Service

3) Client Server Implementation: Services that involve servers are implemented by using the concept of web services. Currently, web services are becoming a simple service-oriented architecture (SOA) with lower development cost [16]. SOA combines Distributed Object Computing, Component Based and web based concepts [17]. According to Al-Rrawahi and Baghdadi [16], web services can (i) easily live with distributed object computing middleware; (ii) make legacy database; (iii) integrate with semantic web and (iv) implement the business transaction. SOA is a set of XML-based open standards that defines everything from service description (for example, WSDL-the Web Services Description Language), to communication with the service, to discovering and connecting to the services (for example, universal description, discovery, and integration-UDDI), to details on combining services to create composite services, transaction completion, and security [18].

There are a number of technologies that have been developed to enable communication with the services. Major software infrastructure providers provide tools support for SOAP–based web services. However, the parsing of SOAP (Simple Object Access Protocol) messages can be memory and computation intensive.

An alternative to SOAP is REST (Representational State Transfer). REST is easier to be used since it is based on URLs and HTTP's four methods, namely, POST, PUT, GET, and DELETE. Furthermore, REST is also understandable by non-programmers [20]. REST also accept XML and JSON format, while SOAP on the other hand only accepts XML format [19]. REST has the advantage because it is an architectural style that provides constraints for the design of networked applications.

Resources are the main entities of abstraction in RESTful architectures. A resource's state is defined by the values of its attributes and connected resources [21]. Access to the web services is done through a Web Service or a simple Web API. The HTTP level access requires many low-level operations and the knowledge of various protocols. High-level libraries in popular languages are often provided [22].

Garber [23] defines API implementations trends to four classifications: (i) API frameworks; (ii) Aggregation; (iii) Backend as service, and (iv) API service providers.

An example of API framework is Facebook API that is used by developers to access Facebook social network. Facebook API is divided to three general APIs, namely, Core API, Facebook SDK, and Advanced API. Core APIs example is Graph API that is a Web API for desktop applications. Facebook SDK is provided in different application language development such as iOS, android, PHP, and Python.

The server side implementation of the Honeybee middleware is done by using REST. For example, Service manager is supported by two web APIs: Web API for registration and Web API for download list. Web API for registration is provided by using the POST method as shown in Fig. 17.

```
POST /honeybee/manager/register
Host: <<host_name>>

Body: <<device_name>>, <<device_mac_address>>
```

Fig. 17 Post method

<<host_name>> is the name of the host, <<device_name>> is the name of device input by user, and <<device_mac_address>> is the mac address of device get from honeybee SDK. In order to download the list for the device, the GET method is used. The request method is shown in Fig. 18.

```
GET /honeybee/manager/download
Host: <<host_name>>
```

Fig. 18 Request method

IV. Conclusions

In this paper, we have described the architecture and implementation of the Honeybee middleware. In particular, we have described the implementation of the client side and server side. The availability of the middleware helps software developers to develop applications within the Honeybee computing environment.

There are three challenges faced during the implementation stage. The first challenge is to explore functions of a hardware component, for example how to connect to IP camera for a user device to access the camera. The node communicates with each other by using different network connection such as W.A.P, peer-to-peer, and GSM. Thus the second challenge is related to finding the suitable communication mechanism to be implemented. The third challenge is the transmission of video to the user device from the local server.

The next stage of the research is to focus on the security for data transmission. Since the middleware involves the use of web service and peer to peer technologies, different security method is needed. Since web service acts as a bridge between server and client, it is important to ensure the security of data transfer. The concept of security in the client-server application, involves multiple roles of client since some of the data are restricted to a particular user.

ACKNOWLEDGMENT

This research is one of the Malaysian Government funded projects under Ministry of Higher Education (MOHE) Long Term Research Grant (LRGS) with reference number LRGS/TD/2011/UKM/ICT/02.

REFERENCES

[1] O. Amft, and P. Lukowicz, “From backpacks to smartphones: Past, present, and future of wearable computers”. *IEEE Pervasive Computing*, vol. 3, pp. 8-13, 2009.

[2] P. Wuttidittachotti, and T. Daengsi, “QoE of social network applications: A study of VoIP quality from Skype vs LINE over 3G and 4G”, in *Seventh International Conference on Ubiquitous and Future Networks (ICUFN)*, 2015, pp. 462-464.

[3] M. Lin, H. Choi, T. Dawson, and T. La Porta. “Network integration in 3G and 4G wireless networks,” in *Proceedings of 19th International Conference on Computer Communications and Networks (ICCCN)*, 2010, 2000, pp. 1-8.

[4] Y. Cui, P. Wu, M. Xu, J. Wu, Y. L. Lee, A. Durand, and C. Metz “Network layer virtualization for IPv4-IPv6 coexistence,” *IEEE Network*, 26(5), pp. 44-48, 2012.

[5] C. K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Pearson Education. 2001.

[6] (2015) Smart City & Smart Village. [Online]. Available: http://www.gsiac.org/programmes/smart-city-smart-village-2/

[7] J. Kwak, J. H. Jin, and M. J. Lee, “A Mobile Application for Information Sharing and Collaboration among Co-located People,” in *Proceedings of the 7th International Interdisciplinary Workshop Series, 2015*, Vol.106, pp.17-21.

[8] Y. Wang, L. Wei, Q. Jin, and J. Ma, “AllJoyn based direct proximity service development: Overview and prototype,” in *17th International Conference on Computational Science and Engineering (CSE)*, 2014, pp. 634-641.

[9] F. Lin, B. Chen, C. Y. Chan, C. H. Wu, W. H. Ip, A. Mai, H. Wang, and W. Liu, “The design of a lightweight RFID middleware,” *International Journal of Engineering Business Management*, vol. 1(2), pp. 25-30, 2009.

[10] B. S. Prabhu, X. Su, C. Qiu, H. Ramamurthy, P. Chu, and R. Gadh, “WinRFID–middleware for distributed RFID infrastructure,” *International Workshop on Radio Frequency Identification (RFID) and Wireless Sensors*, 2015.

[11] P. Tran, P. Greenfield, and I. Gorton, “Behavior and performance of message-oriented middleware systems,” in 22nd *International Conference on Distributed Computing Systems Workshops*, 2002, pp. 645-650.

[12] W. Tian, R. Xue, X. Dong, and H. Wang, “An approach to design and implement RFID middleware system over cloud computing,” *International Journal of Distributed Sensor Networks*, vol. 2013, pp. 1-13, 2013.

[13] (2016) Facebook for developers. [Online]. Available: https://developers.facebook.com/docs/android/

[14] (2016) Spotify developer. [Online]. Available: https://developer.spotify.com/technologies/spotify-android-sdk/

[15] (2016) Qualcomm developer network. [Online]. Available: https://developer.qualcomm.com/software/lte-broadcast-sdk

[16] N. Al-Rawahi and Y. Baghdadi, “Approaches to identify and develop Web services as instance of SOA architecture,” in *International Conference Services Systems and Services Management, ICSSSM’05. 2005*, vol. 1, pp. 579-584.

[17] D. Controneo, C. D. Flora, and S. Russo, “Improving dependability of service oriented architectures for pervasive computing,” in *Eighth International Workshop on Object-Oriented Real-Time Dependable Systems, WORDS, 2003*, pp. 74-81.

[18] R. Welke, R. Hirschheim, and A. Schwarz (2011). Service oriented architecture maturity. [Online]. Available: https://www.infoq.com/articles/soa-maturity-model.

[19] S. Kumari, and S. K. Rath, “Performance comparison of SOAP and REST based Web Services for Enterprise Application Integration,” in *International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2015*, pp. 1656-1660.

[20] X. Shi,”Sharing service semantics using SOAP-based and REST web services.” *IT Profesional*, vol. 8(2), pp. 18-24, 2006.

[21] O. Liskin , L. Singer, and K. Schneider, “Welcome to the real world: A notation for modeling REST services,” *Internet Computing*, vol. 16(4), 36-44. 2012.

[22] J. Li, Y. Xiong, X. Liu, and L. Zhang, “How does web service API evolution affect clients?” in *20th IEEE International Conference on Web Services (ICWS), 2013*, pp. 300-307.

[23] L. Garber, “The lowly API is ready to step front and center”. *Computer*, vol.8, pp. 14-17. 2013.

[24] (2014) R. Horrigan. API Vs. SDK: What The What?. [Online] Available: http://robhorrigan.com/api-vs-sdk-what-the-what/

[25] (2009) G. Simon. Infrastructureless Wireless networks. [Online] Available: http://www.slideshare.net/gwendal-/infrastructureless-wireless-networks

[26] (2014) HTG Explains: What’s the Difference between Ad-Hoc and Infrastructure Mode?. [Online]. Available: http://www.howtogeek.com/180649/htg-explains-whats-the-difference-between-ad-hoc-and-infrastructure-mode/

[27] Introducing JSON [Online]. Available: http://www.json.org/

[28] J. Jeong, D. Shin, and D. Shin, “An XML-based single sign-on scheme supporting mobile and home network service environments,” *IEEE Transactions on Consumer Electronics*, vol. 50(4), pp. 1081-1086. 2004.

[29] S. Hodges, S. Taylor, N. Villar, J. Scott, D. Bial,, and P. T. Fischer, “Prototyping connected devices for the Internet of Things”. *Computer*, vol. 46(2), pp. 26-34. 2013.

[30] N. H Azizul, M. F. Nasruddin, M. R. Mokhtar and A. M. Zin, "Advanced ubiquitous computing to support smart city smart village applications," in *International Conference on Electrical Engineering and Informatics (ICEEI), 2015*, pp. 720-725.

[31] F. Lin, and B. Chen, “The design of a lightweight RFID middleware”, *International Journal of Engineering Business Management*, vol. 1(2), pp. 25-30, 2009.

[32] B. S. Prabhu, X. Su, H. Ramamurthy, C. Chu, R. Gadh, “WinRFID: a middleware for the enablement of radio frequency identification (RFID) based applications”, in *Proceedings of the Wireless Internet for the Mobile Enterprise Consortium (WINMEC '05), 2005*. In *Mobile, Wireless and Sensor Networks: Technology, Applications and Future*, pp. 331-336, John Wiley & Sons, 2005.

[33] P. Tran, P. Greenfield, I. Gorton, “Behavior and performance of message-oriented middleware systems,” in *Proceedings of the 22nd International Conference on Distributed Computing Systems, 2002*.

[34] W. Tian, R. Xue, X. Dong, and H. Wang, “An Approach to Design and Implement RFID Middleware System over Cloud Computing”. *International Journal of Distributed Sensor Network*, vol. 9(10), 2013.