Prosiding
Seminar Nasional Informasi, Teknologi dan Komunikas

Innovate 'n Create

2007

"New Trends and Developments in Information Communication and Technology (II) towards Creative Economy"

Fakultas Ilmu Komputer
Jakarta, 8 Desember 2007
INTEROPERABLE MOBILE MESSAGING APPLICATION

Raymond Kosala¹; Tri A. Budiono²

School of Computer Science, BINUS International
JL Hang Lekir I no. 6, Kelayoran Baru, Jakarta 12120
¹rkosala@binus.ac.id, ²tbudiono@binus.ac.id

ABSTRACT

Instant messaging (IM) applications are now broadly used as a way of communicating, not only for individual use but also for corporation. By its ability to identify presence as well as its synchronous nature, IM offers a better communication channel than email. Despite these benefits, the use of IM for corporations is not without problems. The most crucial problems are security and and interoperability. In this paper, we have implemented an interoperable and secure mobile instant messaging solution to solve the messaging problems above. The tests and experiments show that our IM solution is able to improve security, mobility, interoperability and collaborative works among employees.

Keywords: Instant messaging, collaborative application, mobile application, XMPP protocol

INTRODUCTION

Instant Messaging

Instant Messaging (usually called IM) application is an application that provides online, real-time communication using typed text messages between two or more people who are connected to the Internet [3]. With the IM application, people can communicate with their colleagues who are online in their contact list. The contact list contains all contacts, usually by email address, that are online and also offline. Nowadays, instant messaging application is not only used for personal importance but also becoming a popular communication tool for business purposes. The ability to connect with colleagues and to have discussions with team members have somewhat pushed the telephone and email back from the forefront of communication tools. Millions of corporate users utilize instant messaging on their work. According to report from market analyst IDC, IM applications are now broadly used by people around the world as a way of communicating. In addition to the above fact, the corporate IM market grows from 5.5 million users worldwide in 2000 to 180 million in 2004, representing a growth rate of 140 percent [3].

The reason why many companies now start to adopt IM in its operation is because business communications have become more complex and companies still have difficulties collaborating with the key people in a timely manner. Companies need to respond more quickly to their employees and customers, and must avoid communication obstructions to maintain a competitive edge and grow profitably. The need to respond in a timely manner is crucial to businesses as delays often result in serious financial consequences. For instance in IT field, the consequences could be the delay of the whole project and will waste time, resource and money.

According to the survey done by META group, a market research company, 57% of respondents use IM at work, and 56% of those surveyed use IM at home for business purposes [8]. IM also gives worker the ability to get quick answers from people who are working together, but are in different geographical locations. The META Group survey also found that 78% agreed that messaging gives a quicker response than email, and 74% agreed that messaging gives a quicker problem solving. Most of the surveyed (65%) liked the ability to see whether someone was online and available, while 37% said IM lowered the cost of monthly phone bill due to less long-distance calling.

Interoperable Mobile Messaging Solution

Despite the above benefits, however, the use of IM for corporate use is not without problems. One problem is the security problem. This problem is due to the explosion of trojans, viruses, worms, IP address disclosure, and instant messaging spam towards the IM application. Another problem is that most current IM solutions are not interoperable, in the sense that they can only connect to the same IM applications, or they can only connect to another different IM application at one specific time. Most of these applications cannot connect to two or more different IM applications at the same time. For example, an IM application running on Windows operating system can connect to the same IM application running on Linux, but cannot connect to two or more different IM applications running on Windows. This is the case with popular mobile IM applications such as Pocket MSN, ICQ, etc.

To deal with these problems, we propose an alternative communication means which does not depend on the Internet connection. We propose an interoperable mobile messaging solution based on Wildfire, which is an enterprise instant messaging (EIM) server [2].
An internally installed Wildfire server, to be used for supporting the chat application, can connect all the employees even if they are not in the office. The IM application can be accessed through their mobile communication devices, such as cellular phones and PDA. These devices can connect to the Internet by using the GPRS feature which does not depend on the Internet connection, and it does not cost as much as phone calls. Through the GPRS Internet connection, they can connect to the mobile Instant Messenger to chat with their online colleagues with no geographical boundaries.

In this paper, two different IM applications are developed to demonstrate the interoperability. One application is for Pocket PC using .NET Compact Framework and C#. The other application is build using Java J2ME running on the Symbian-based operating systems. The company can improve the security of its communication channel using this internal IM application.

On top of that, the instant messenger application can also be used to connect with other public messenger servers, such as Yahoo and MSN servers. Thus it provides the interoperability of company’s communication. In the case of Internet down, the users can still communicate with each other using the Wildfire server. Moreover, since it will be implemented in the mobile device, mobility will be achieved.

The rest of the paper is structured as follows. Section 2 explains the proposed messaging solution. Section 3 describes the development methodology. Section 4 describes the result and tests, and Section 5 concludes the paper.

The Proposed System

The solution that is needed is a chat application that can be implemented in mobile devices from any platform. Although the solution can be easily developed for PC or non-mobile devices, mobile platforms are chosen to show the portability of the applications.

System Architecture

![System Architecture Diagram]

Imagine a company wanted to provide an internal server to ensure security of the IM application. Thus all the employees’ devices will be connected to the company’s server. In order to be connected to other IM applications from other server, the company’s server will be the one who manages the connection to the other server. Thus, the communication can be controlled completely by the company. The architecture of the system is depicted in Figure 1.

The client application is designed so that it can be connected to any other devices with different platform, which are PDA, cellular phone, desktop or laptop computers. In order to support multi-platform devices, the protocol that would be used in the IM application development is XMPP (Extensible Messaging and Presence Protocol). XMPP employs XML streaming that can be implemented in any platforms. This is suitable to work with the Wildfire server, which support XML streaming.

Wildfire

In order for the IM client applications to connect to other IM servers, a Jabber (or XMPP) server is needed. What we will use to support the implementation of the IM applications is a Wildfire Server. Wildfire server is an enterprise instant messaging (EIM) server written using Java language, licensed under the Open Source GPL (General Public License) and also licensed commercially (dual-licensed). It uses the leading open protocol for instant messaging, XMPP (also called Jabber) [2].

XMPP

XMPP stands for Extensible Messaging and Presence Protocol. It is an IETF’s (Internet Engineering Task Force) formalization of XML streaming protocol for near real-time information exchange and extended to be used for instant messaging and presence. It was developed within the Jabber open-source community in 1999 [11]. It provides a generalized framework for changing XML data from one entity to another but its main purpose is to help building instant messaging and presence applications [9]. XMPP protocol specifications that are used within this development are documented as RFC 3920, RFC 3921 and RFC 2831. Those three specifications contain the documentation of technical and organizational notes about instant messaging.

| RFC 3920 | XMPP Core | The core protocols for XML streaming, including strong authentication, channel encryption, and internationalized addressing. |
| RFC 3921 | XMPP: Instant Messaging and Presence | It is used for instant messaging, contact lists, presence, and privacy blocking. |
| RFC 2831 | Using Digest Authentication as a Simple Authentication and Security Layer (SASL) Mechanism |
Jabber

Jabber, created and developed by Jeremie Miller since 1998, is an open-source instant messaging platform which uses XML-based protocols (XMPP) to provide the standard functionality that is expected from an instant-messaging system, such as peer-to-peer chat, group chat, and the ability to subscribe to other user's presence [7]. Jabber enables developers to provide built-in or client-based services based on an open, asynchronous, extensible, decentralized, and secure XML protocol riding directly on TCP/IP to provide real-time exchange of messages and presence [4].

Jabber employs client-server architecture, instead of peer-to-peer. Thus, there is no direct communication between the clients and each client will be connected to one Jabber server, which listens on TCP port 5222. When two Jabber clients wish for delivering a message, the message would be passed through (minimum) one Jabber server. If both clients are on different domain, the message would be passed from one Jabber server to another Jabber server, afterwards the message will be sent to the desired client.

Instant messaging supports collaborative work. Collaborative work means work undertaken as part of a group activity towards some shared goals or has some common purpose. Through collaborative work the people are united to work together, share ideas, makes decision as a team. Unlike other communication means that only provide peer-to-peer communication, with the use of instant messaging a person can chat to several people at the same time, anywhere, anytime.

Development Methodology

In this work, we used the Waterfall methodology, which comprises of requirement specification phase, analysis phase, design phase and implementation and testing phase [1].

The initial phase in Waterfall methodology is to gather the requirement. The requirement is basically to create secure and interoperable mobile messaging applications that have basic functions, including connecting to the Wildfire server, sending and receiving real-time messages, changing status, adding a new contact, and deleting a contact.

The next phase is the analysis. In this phase, we structure and analyze the requirements obtained from the previous phase. The result of the analysis is that we would add additional functions that will be useful and can be implemented in the system. The additional features include: sending and receiving offline messages, connecting to MSN and Yahoo servers, viewing a contact's profile, editing user profile, sound effect, searching contact(s), and group capability (allows user to group the contacts).

In the design phase, we perform high-level and low-level design. High-level design includes use case diagram and use case narrative, whereas low-level design consists of dataflow diagram and activity diagram. Use case diagram is used to depict interaction between the system actor and the system, whereas use case narrative is created to give detail of textual description of how actor will interact to the system to accomplish the tasks. Dataflow diagram depicts the flow of data through a system and the work of processing performed by the system, whereas activity diagram is used to show the flow of each process in the system. Due to the space limitation, we cannot put the design diagrams here. For the detailed design diagrams, interested reader can consult these publications [6, 10].

Finally based on the design, we developed the IM applications. Afterwards, we performed alpha and beta testing. Alpha testing is done through testing each function to make sure that the application meets the expected result. Beta testing is conducted with some business users as the subject of the test. The business people, as the end user, will be given the opportunity to try the application. Afterwards, a questionnaire will be delivered to them. The questionnaire is used for measuring the usability of the application: whether the application can solve their corporate messaging problem.

Results and Tests

Some screenshots of the IM applications are given below.
In Table 2 below, we can see all the features that have been successfully implemented and tested.

<table>
<thead>
<tr>
<th>IM Application Features</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign In</td>
<td></td>
</tr>
<tr>
<td>Remember Me</td>
<td></td>
</tr>
<tr>
<td>Forget Me</td>
<td></td>
</tr>
<tr>
<td>Send Message</td>
<td></td>
</tr>
<tr>
<td>Receive Message</td>
<td></td>
</tr>
<tr>
<td>Send Offline Message</td>
<td></td>
</tr>
<tr>
<td>Receive Offline Message</td>
<td></td>
</tr>
<tr>
<td>Close Emotion</td>
<td></td>
</tr>
<tr>
<td>Add Contact</td>
<td></td>
</tr>
<tr>
<td>Accept Subscription Request</td>
<td></td>
</tr>
<tr>
<td>Refuse Subscription Request</td>
<td></td>
</tr>
<tr>
<td>Delete Contact</td>
<td></td>
</tr>
<tr>
<td>Search Contact</td>
<td></td>
</tr>
<tr>
<td>View Contact's Profile</td>
<td></td>
</tr>
<tr>
<td>Edit User Profile</td>
<td></td>
</tr>
<tr>
<td>Change Status</td>
<td></td>
</tr>
<tr>
<td>Create a Group</td>
<td></td>
</tr>
<tr>
<td>Change Contact's Group</td>
<td></td>
</tr>
<tr>
<td>Connect to MSN and Yahoo server</td>
<td></td>
</tr>
<tr>
<td>Disconnect from MSN and Yahoo server</td>
<td></td>
</tr>
<tr>
<td>Sound Effect</td>
<td></td>
</tr>
<tr>
<td>Sign Out</td>
<td></td>
</tr>
</tbody>
</table>

We also did some experiments in order to determine the time required to connect to the server, sending and receiving message from the two instant messaging applications from two different platforms that had been developed: Java and .Net.

The experiments are done as follows. A running Wildfire server is located in a building located at around 5 km from the testing location. We did the experiments with both mobile devices either inside or outside of the room. For sending and receiving message, two different messages are sent. The first one is the long message which consists of 50 characters (space included): “Hai Charlie... Did you get my message? How are you?” while another one is the short message which consists of 14 characters: “I’m fine thanks”. Each of this experiment is done ten times to obtain a time range to send a particular message. The result of the experiments can be seen in Table 3 and 4.

From the experiments, we can see that the time needed to connect, send and receive messages between each other is varying according to the GPRS connection. The time needed to send a message from .Net application to Java application is quite similar and does not show significant differences. Thus, the platform of the application seems irrelevant to the application’s performance. We also observe that the delay is not too long and is acceptable for the IM application to be used in practice.

In addition to the experiments, we were also asking for feedback from the potential end users. The subjects of the testing are the employees of a certain company who use Pocket PC/PDA rather than cellular phones as their mobile communication device. The number of business users that participated in the testing was 20.

This testing was started by giving a trial for each user to test the application. Afterwards, the user will be given a questionnaire containing questions about the IM application. The result of the questionnaire can be summarized below.

The users gave positive response to the functionality and GUI of the application.

It does not take much time for the users to learn about how to use the IM application.

The users support the company to complete the IM application.

The result of testing and questionnaire shows that our IM solution can provide alternative means of communication that support collaboration work between the business users, as well as mobility and interoperability.

Fakultas Ilmu Komputer, BINUS University
CONCLUSION

Mobile instant messaging application can be used as an alternative communication channel for users that currently employ cellular phone, cable phone, email and instant messenger (using public server). When business people were working outside the office, they can only use cellular phone calls and SMS, which may cause them to spend lots of money for cellular phone calls and SMS. On the other hand, communication inside the office through instant messaging application using public server causes security issues.

In this paper, we have implemented a interoperable and secure mobile instant messaging solution to solve the corporate messaging problems above. The experiment results show that our IM solution is interoperable in the sense that it can connect to different IM applications simultaneously. This is the case with popular mobile IM applications such as Pocket MSN, which is the built-in IM application installed in all Windows Mobile devices. Although we did not test it, the interoperability with other XMPP/Jabber based IM applications, such as Google Talk, should be possible. The security is based on Digest Authentication as a Simple Authentication and Security Layer (SASL) mechanism. Furthermore, the use of private server solved the security issues because the conversation is stored in private server instead of the public one. Possible further works include adding more features, such as file transfer, etc., and also adding connections to other IM applications, other than Yahoo and MSN, to further enhance its interoperability.

REFERENCES

B. Bruegge and A.H. Dutoit, 2003, Object-Oriented Software Engineering using UML, Patterns and Java, Prentice Hall, USA.


