MjoY – Common Platform for Mobile Ticketing

A. S. K. Fernando, K. A. P. H. Gunatathna, K. A. Lakmini, M. M. D. Perera and D.E.W.V. Nanayakkara
Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka.

Abstract - **The project "MjoY - Common Platform for m-ticketing" is targeted to provide a general solution for M-ticketing systems in which the common functionalities of a typical M-ticketing system are embedded. The Common Platform is intended to be used by any merchant domain wishing to implement an m-ticketing solution by implementing only the domain logic related to ticket generation in a separate module and plugging it in to the common platform or by entering necessary parameter specific information through the simple APIs provided. This research paper is comprised of some details about the product and the main research areas which were focused during the formation of the Common Platform concept.**

I. Introduction

Mobile ticketing (M-Ticketing) has become one of the latest ticket reservation and delivering technologies which has emerged to address the issues related to traditional paper ticketing systems. Traditional ticketing has evolved from pre-printed tickets and the tickets printed at the point of purchase where purchasers buy the ticket from the ticket holder physically, then to tickets printed by the purchaser where ticket is purchased remotely and a unique ticket code is delivered to the purchaser in an electronic way. Next stage of evolution is the mobile ticketing where the ticket code is delivered to a mobile device of the purchaser in a form of text message using SMS or a multimedia message using MMS [1].

Currently most of the industries use M-Ticketing in replacement for their existing traditional paper based or web based ticketing systems. All such M-ticketing services which built independently have a set of common features in the process of ticket reservation from the ticket requesting of the purchaser up to the delivery of the ticket.

MjoY - Common Platform for Mobile Ticketing is a system which encapsulates all the common functionalities of the existing M-Ticketing solutions which can be used as a platform where new M-Ticketing solutions can be built upon. It facilitates the communication between three parties involved in mobile ticketing process namely ticket issuer, ticket purchaser and the mobile service provider.

Platform provides clearly defined API s to obtain those common services for the M-Ticketing application developers for the development of ticket purchaser (client) applications as well as the ticket issuer (merchant) applications. Mobile Ticketing platform consists of four main components.

1) Registration Module
2) Ticket reservation module
3) Ticket validation module
4) Reporting module

II. Background

Background research is done in order to identify the functions of the existing M-Ticketing solutions and identify the common utilities of those individual systems.

M-Ticketing is a process which facilitates ticket ordering, ticket delivery and the ticket validation [2]. Different M-Ticketing solutions use different techniques in the stages of this process. Whole idea of M-Ticketing is delivering the ticket to a mobile device. This can also be done in variety of ways.

Following are the variety of technologies that are used in current systems.

Ticket Ordering: Through web, text messaging, voice calls from a telephone, WAP sites or secure mobile applications

Ticket Delivery: Text messages, text messages through WAP push and picture messages through SMS or MMS.

Ticket Validation: Manual validation, scanning based validation using OCR or laser scanner.

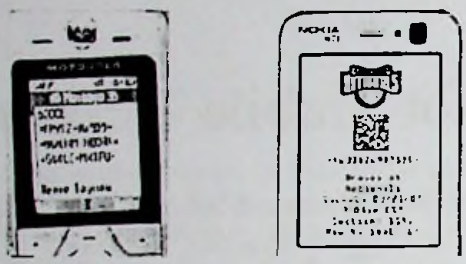

Fig. 1. Methods of delivering ticket.

Selection of techniques is highly dependent on the particular application domain and cannot be restricted. When analyzing the different techniques used in M-Ticketing, it could be identified that the techniques of obtaining inputs and delivering the output to and from the system are different from one system to the other. Internal processing of obtained input and process of creating the output may be common for two different applications.

Identified common functions of mobile ticketing systems are as follows;

1) Receiving ticket purchaser's request
2) Analyze request parameters
3) Decide the response parameters
4) Create a unique ticket code based on the response parameters
5) Send the response to the request originator

From the above actions, analyzing the request and deciding the response parameters can be done only in domain specific manner.

Problem: When a new industry or an event is interested on providing M-Ticketing facilities for their clients, they need to develop the M-Ticketing solution from the scratch. Most of the time is wasted for implementing utility services. Users have to follow different procedures for ordering M-Tickets in different domains.

Proposed Solution: Implementing a "Common Platform" which encapsulates the common and basic functions of M-ticketing. Providing clearly defined APIs for the external application developers for M-Ticketing application development on top of the platform by directing calling the API functionalities.

III. MJOY FUNCTIONALITIES

MjoY introduces a generalized message formats for external parties to be used in communication with the platform. This helps the platform to be common to many merchants.

A. Ticket Requests

MjoY uses a parameterized request format for ticket requests. Number of parameters and the parameter values are totally dependent on the merchant and are configurable through MjoY. These configurable parameters are used to provide facilities in ticket booking such as bulk reservations and ticket classes (first class, balcony etc). MjoY requires the first parameter to be the unique merchant ID and rest of the parameters are configurable by the merchant.

B. Merchant Types

Two categories of merchants have been identified based on the complexity of domain and behavior of request parameters.

Standard Category: The standard category is for the merchants with short term projects with less complex domain logic. Since the projects are short term and less complex the values that need to be incorporated with each parameter can be pre defined.

In the case of a drama festival, the format could be,
<Merchant ID> <Show time> <Seat class> <No of seats>.

In the above case all parameter values can be predefined for the whole project.

Standard category module is already built into the system. Therefore there is no need to alter the source code of the system when adding these merchants. They can be easily added through simple user interfaces provided by the system.

Standard category merchants which can support up to three parameters (excluding merchant id) are already implemented in the common platform. This could be easily extended to support a higher number of parameters.

Custom Category: Custom category is for the merchants having long term more complex projects where their requirements are subject to constant change. In this scenario, these type of merchants cannot easily pre define the values for each parameter easily.

A cinema can be considered as a custom category merchant as their requirements in order to reserve a ticket through text messages may vary occasionally. This may vary depending on the movies shown and different upgrades done to the cinema theatre from time to time.

Therefore a separate domain specific module should be separately plugged on to the common platform in order to handle custom type merchants in the platform.

C. Testing

Testing of the platform is done for both standard type and custom type merchant categories. Standard type merchants are currently supported and tested up to three parameters.

Custom type merchant handling and ticket reservations were tested using a sample application for a cinema hall. Tickets could be reserved through text messages and through the web interface provided by the system. The SMS syntax for ticket reservation is,

<Merchant ID> <Movie name> <Date> <Time> <Seat Type> <Adults/Children>

e.g. CineCinema Tansformers 10-07-2009 10.30a.m. balcony a2c2

D. Value added Services

End of the day (EOD) reports and customized reports are provided to the merchants by the system. EOD reports are produced at the end of each day and the customized reports are produced on merchant request.

E. Validation Techniques

Validating a long queue of ticket holders is identified as a major problem in any ticketing industry. Reducing the time of acquiring the ticket data from the ticket holders' mobile device (scanning the ticket) is the common solution is being used.

Common Platform takes another approach called batch validation which addresses the same problem in a different angle. Validator is provided with a separate mobile application where all tickets issued for an event can be downloaded. Initial validation is done at the mobile application level and a batch of such validated tickets is sent to the server afterwards. This reduces the time taken to communicate with the remote server [3].

F. Sending and Receiving SMS Messages using GSM Modem

Common platform is capable of receiving messages through SMS (ticket requests) as well as sending SMS (ticket responses) for clients. System uses a GSM modem which is capable of sending messages using SMS protocol. Modem is connected to the system through a logical COM port.

SMS message receiving is handled by a separate thread where it continuously polling the modem to retrieve new incoming messages. When the system attempts to send an SMS message from the system using a separate thread it always fails since the COM port is always allocated for the receiver thread.

Using a single thread (sender_receiver) for both message sending and receiving via modem solved this issue. Messages to be sent are initially stored in separate place and are in pending state. When sender_receiver thread get the execution time it sends the pending messages to the destination mobile device.

IV. Security

A. Mobile Client Application

The customized SMS application is being used to facilitate secure communication between the client's mobile device and the common platform via SMS. The SMSs are sent through the existing GSM networks which can be considered insecure.

The security is achieved through implementing public key encryption mechanism. Public key algorithms are used to implement the one-way trapdoor nature of a system in which a message encrypted using a particular public key can only be decrypted using the corresponding private key. Public key is made available to public, but the private key is intended to be unshared [5].

During the first run of the client mobile application, the key pair for that particular mobile device is being generated and the public key is made available to the common platform. The public key of the common platform is shared among all the mobile clients. When a client wishes to send a message to the common platform, the message is being first encrypted using the public key of the common platform. Then the message is directed to its destination through the general GSM network.

When the message is being delivered to the platform, it decrypts the message using the private key of the platform and the original message is being obtained. During the decryption process it can be tracked whether the integrity of data is being violated during transmission. The common platform encrypts the messages which are sent to the mobile clients using the corresponding public key of each client. At the reception of the message at the client's mobile device, the encrypted message is first decrypted

using the client specific private key and obtains the original plain text message.

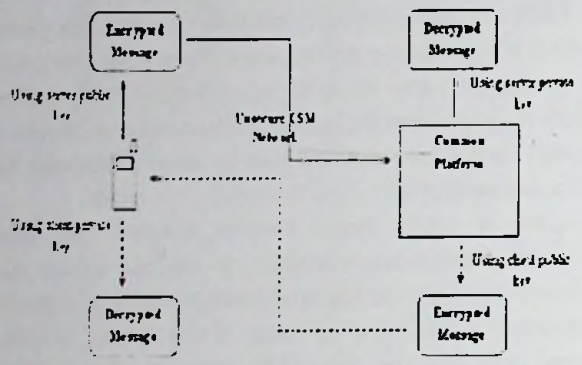

Fig. 2. Public Key Encryption between client mobile application and common platform

B. *Secure File Transfer*

The requirement of secure file transfer exists when transferring files containing transactional details between the common platform and the external systems, typically the merchant systems and operator systems.

The common platform XML file security is handled using Apache XML Security Framework [4]. Initially, the common platform and the external system agree upon a symmetric key which is then used for the encryption and decryption purposes. The key is not intended to be shared among external parties other than the two systems involved in file transfer. The XML file is encrypted at the common platform using the agreed key and directed to the external system using a file transfer web service. When the external system receives the encrypted XML file, it decrypts the file using the symmetric key. In case of a violation of integrity of data in the XML file during transmission through the unsecure network, it can be tracked.

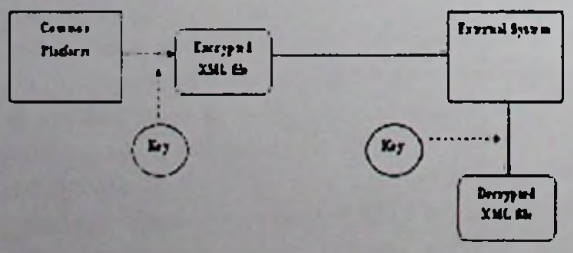

Fig. 3. XML File encryption and decryption.

V. DEVELOPMENT TECHNOLOGIES

A. *Client Server Architecture for Platform*

MjoY can be identified as a system which implements significant amount of client server communication as the system is used by merchants and general public in order to get their requested services done.

The project involves extensive communication among merchant applications, checker's mobile application, user's web clients and mobile clients and the common platform. In order to achieve the success in such communications, principals of java remote method invocation has been used. These principals implement the client server architecture in MjoY.

A stub program exists in the client side of the client-server relationship and a skeleton at the server end. The stub acts as a proxy to the client application when calling the remote object.

Similarly, the skeleton acts as a proxy to the remote object at the server which is being called. Therefore the complicated communication is purely handled by the stub and the skeleton objects.

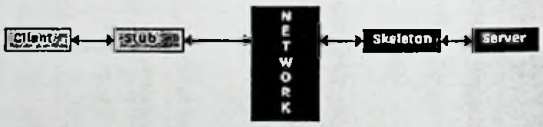

Fig. 4. Implementation of Remote Method invocation using stub and skeleton objects.

One objective of the system was to provide easily accessible APIs to all remote entities that communicate with the MjoY common platform. Such APIs include,

1) The interface between the user who reserves tickets and the common platform
2) The interface between the merchant and the common platform
3) The interface between the ticket checker and the common platform
4) The interface between the mobile operator and the common platform

The interfaces had been implemented with the help of EJBs as they provide remote and local interfaces to the client as implemented according to the above mentioned model.

VI. CONCLUSION

It can be concluded that a common platform that can be used for mobile ticketing can be a commercially feasible product with effective returns to the merchants who are willing to use its services. Client-server architecture and the security implementations discussed in the paper can be used effectively in order to develop the concept of the common platform.

ACKNOWLEDGMENT

The authors wish to acknowledge the support given by Mr. Bimal Gunapala and Mr. Mithila Somasiri from Millenium IT Pvt. Ltd. Sri Lanka.

REFERENCES

[1] Hand Held Products (2008, May) [Online] Avaliable: http://www.handheld.com/download.aspx

[2] Wikipedia - Mobile Ticketing (2008, May) [Online] Available: http://en.wikipedia.org/wiki/Mobile_ticketing

[3] Mobi Solutions: SMS-gateway & M-marketing in Estonia, Latvia and Lithuania, M-government & Mobile Apps development (2008, May) [Online] Available: http://www.mobisolutions.com/docs/mtickets.pdf

[4] Apache XML Project (2008, June) [Online] Available: http://xml.apache.org/

[5] Michael Juntao Yuan, "Enterprise J2ME: Developing Mobile Java Applications". In *Prentice Hall PTR*, October 2003