Cloud to Device Messaging with Voice Notification Using GCM

C. Tamilselvi, B. Vijaya Kumar

Abstract: **Push Messaging for Android devices are going to be implemented with Voice Notification using technology provided by Google called "GCM (Google Cloud Messaging) .The GCM is Cloud based push messaging service developed for android devices. It acts like proxy server between Third party application server and Android client terminal. Getting latest news from server is very important when notification corresponds to emergency news like earth quake notification news. Nowadays smart phone plays very important role in accessing information from web because of faster growth of mobile internet. In order to push some important notification messages from third party server to all android terminals, GCM has been used as proxy server. It allows the third party server to send message to android terminals. So the communication can be happened through GCM. GCM Connection servers give the support for handling of messages like queuing and delivery of messages to particular target android terminals in which the messaging application running. The current work deals with implementation of android messaging application with voice notification using GCM service, so that the visually impaired person can easily come to know about the notification messages arrived at android terminals.**

Index Terms: **C2DM, GCM, Cloud Computing, Push Messaging for Android, Voice Notification**

I. INTRODUCTION

GCM (Google Cloud Messaging) is main paradigm for project going to be implemented. . It allows to send data or message from third party application server to client's or user's Android –powered devices. Services provided by the GCM server includes message handling like queuing of messages received from the third party application server and delivery of messages from its queue to client application available on android phone. GCM acts like proxy server in between the application server and android terminals. GCM support s olny light weigt message like notification message or messages having upto 4kb of payload data.[1]

Manuscript received April 22, 2015; revised May 18, 2015.

C. Tamilselvi, Student, CS Department, Bits-Pilani, Dubai Campus, UAE, Mail id: h2012034@dubai.bits-pilani.ac.in, tamil.vetri@gmail.com

B. Vijaya Kumar, Associate Professor, CS Department, Bits-Pilani, Dubai Campus, UAE, Mail id: vijay@dubai.bits-pilani.ac.in

A.MOTIVATION

The communication between two electronic devices in network is commomnly through some protocols. Normally two methods can be used in network communication especially between client and server called PULL AND PUSH messaging technology. with the traditional pull technogy the client has to poll the server regularly for updates. This increases unnecessary network traffic and reduces the android terminal battery when used as a client device. If the frequency of polling is in-appropriate then it will lead to information delay which is very bad in case of latest emergeny news like earth quake messages getting form the server.

In order to ensure the social stability this kind of emergency notification messages and latest server news needs to be arrived at time. So using GCM service provided by google for android terminal, there is a possiblity to enhance the realiability of notification messages. Using store and forward technique, the GCM server will store the notification messages arriving from the server and pushes those information to all android terminals registerd with this GCM server.

B.PROBLEM DEFINITION

Instead of using normal pull messaging mechanism, a new technology has been examined called GCM (Google Cloud Messaging) provided by Google for Android terminals. Using this GCM server the Android client can get notification messages or latest messages from the third party server. Here, GCM acts as proxy server in between the android client and server. GCM handles some functions like queuing and delivery of notification messages to target android application running on android terminal. The current work deals with implementation of android messaging application with voice notification using GCM service, so that the visually impaired person can easily come to know about the notification messages arrived at android terminals.

Using GCM service, the android terminals battery power is going to be saved and unnecessary network traffic is also going to be reduced. It increases the timeliness and effectiveness of information received from the third party application server.

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

II. RELATED WORK

On demand, dynamic-provisioning of resources like computing services, hardware and platforms can be possible for the end-user with the advent of new technology called Cloud Computing. These computing resources will be provided to customer as in three form namely Sotware as a service, Platform as a service and Infrastructure as a service and abbreviated as SAAS,PAAS, and IAAS respectively.[2]

Actually GCM is a result of messaging service iteration developed by Google and discovered at I/O conference. Significant performance improvements can be realized by this new invention made by google. The first messaging service invention called as Android Cloud to Device Messaging (C2DM). GCM invented as a final version, and the core functions of C2DM had been retained by the GCM.

By tweaking the way of communication done by C2DM, the GCM service has been expected to get more power efficiently. The GCM push messaging service supports notification message or messages up to 4kb of payload data.

The added capability of GCM is "Multicasting". It means pushing the data to multiple devices at time (simultaneously). So seamlessly the data is going to be synchronized among devices and the server will be able to push live updates to multiple users more efficiently and easily. For example, cricket score updates [3].

There is messaging service iteration developed by Google. The first version of messaging service for Android device is called as C2DM (Android Cloud to Device Messaging). By adding new features to C2DM, it is deprecated, and new version has been released in June26, 2012. The added new feature may include like the way of communication happening between the server and client, and multicasting. So the existing users and developers of C2DM are encouraged to use the new version called "GCM" [4].

Google Cloud Messaging for Android is newest and latest messaging service provided by Google, which allows users and application developers to send message data from their application server to their user's android powered devices. For things to be happened, both the server and client need to register with Google's GCM server first. The GCM service provided by Google uses store and forward technique to push messages to all android terminals when these devices come to online. It will handle some aspects of messaging service like queuing and delivery of these messages to particular android devices in which the application is running.

As mentioned before, the messages supported by Google are like kind of light weight messages say, notification messages or messages having size up to 4 kb of payload data.

The GCM could not support to create application like Instant messaging application, because it would not provide any guarantee for message delivery and order. But it can support application like giving notification messages to that instant messaging application like the users have received new messages from servers. So the application has to synchronize with server and need to fetch the data from server. It is up to the client for information processing. The GCM service would not provide any built in user interface to handle messages.

Key Concepts of GCM: 2 terms are used in GCM namely,

- Components – Real Physical entity invloved in GCM service.
- Credentials - Credentials are authentication related information like id and token which are used to ensure that only the authenticated persons are sending and receiving messages.

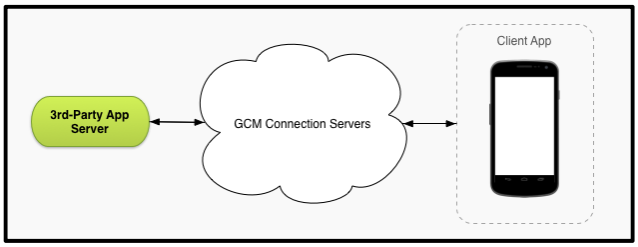

Fig: 1 GCM Component Interaction

- The responsibility of Google's GCM is to store and take the messages from application server and to deliver them to all Android device in which the GCM enabled application is running. There are two Google connection server provided for GCM namely HTTP and XMPP.
- The second component is **Third-party application server** which has been implemented to work with selected connection server provided by google. The third party application server pushes whatever the newest and updated data or notification messages to GCM Connection servers. The GCM server uses store and forward technique to store the notification messages arrived from the server and pushes the information to all android devices when the devices come to online.
- The Client App is android application running on android device for the GCM services should be enabled in Google API. In order to receive the message from GCM this android client application must register with GCM and get registration ID. This registration id should be shared with application server. So the server comes to know about the registered and authenticated client to receive server's messages.[1]

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

III. SYSTEM DESIGN

A. *USE CASE DIAGRAM*

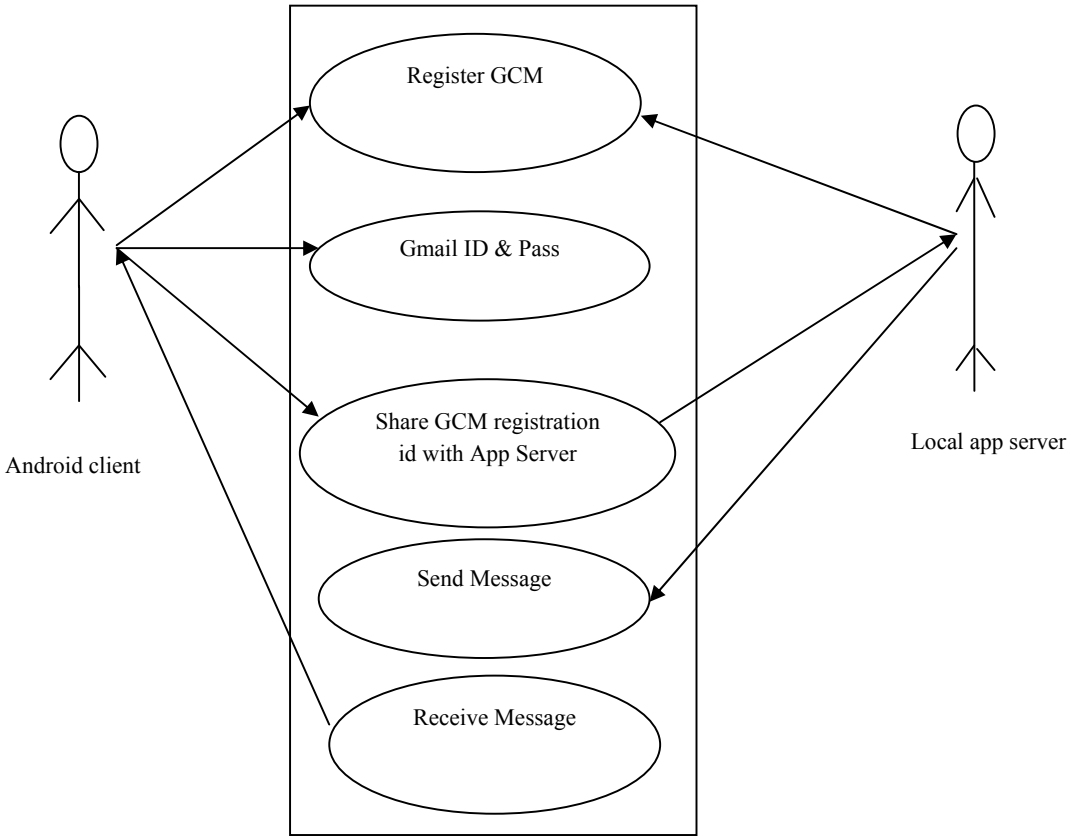

Fig: 2, GCM Use Case

Analysis and design the framework for GCM service to push notification messages to android client terminal from application server with voice reading the notification messages.

In fig: 2, Use Case diagram, there are two entities, called third-party application server and android user perform important activities about the GCM registration.

Activities:

- Enable GCM. Server and An Android application should register first with GCM server in order to send and receive messages accordingly. Server needs to get API key as authentication token from GCM server. Android application needs to get GCM Registration id from GCM and sent that id to Application server.
- Send a message. Sending the latest message is done by the application server to connection server called GCM. Here the server GCM will act as proxy server.
- Receive a message. Receiving a message is nothing but the GCM server will queue the message arrived from the application server and it will send those messages to android application.

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

B.SYSTEM ARCHITECTURE:

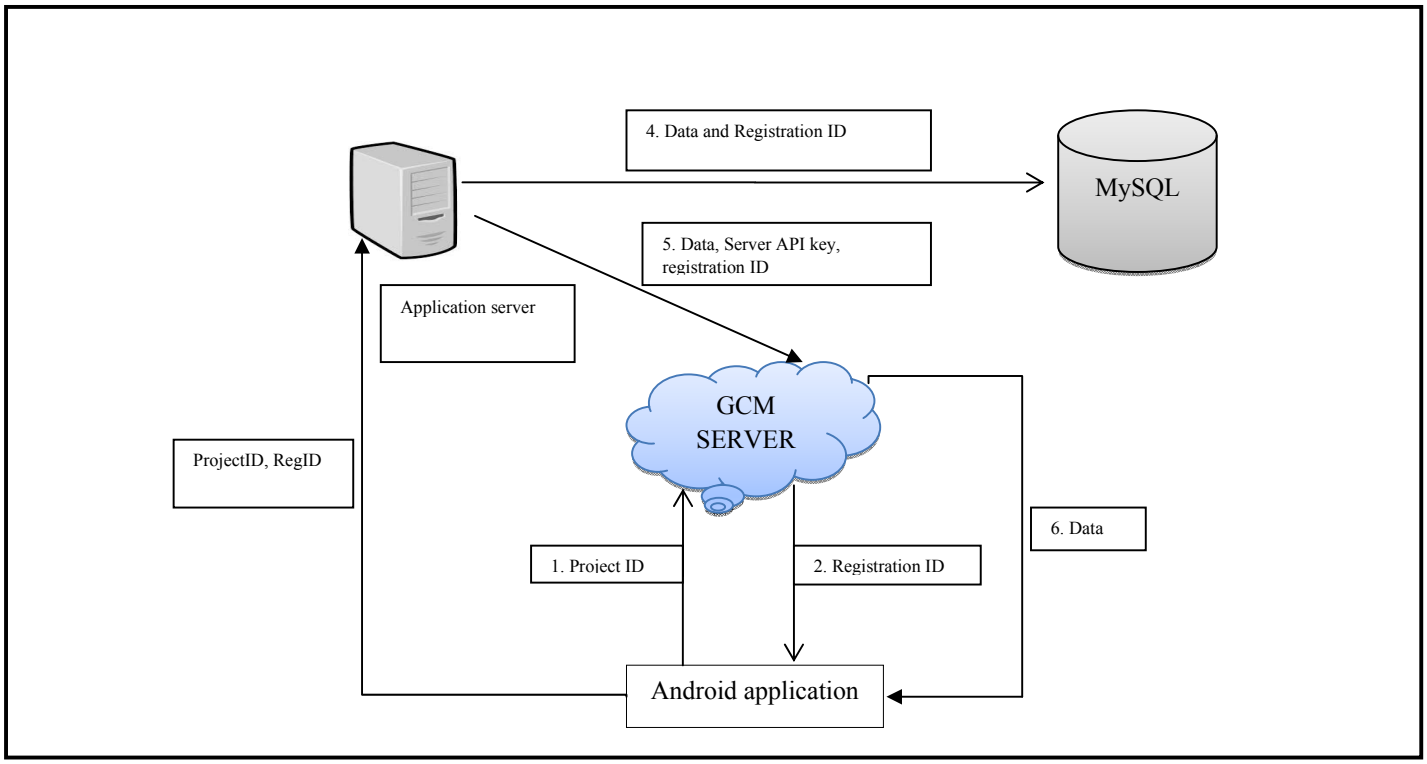

Fig: 3, GCM System Architecture (Schematic)

The sender ID is the project id or number of the third party application server which has acquired from API console. So the application server is permitted to send the messages to GCM. Application Id is the android application package name mentioned in manifest.xml file of android coding. So these ids ensure that the messages are targeted to the correct Android application.

Procedure:

Project ID and Application ID will be given to GCM server in order to get GCM Registration ID for the android client application. This ID is unique and it should be shared and stored with app server.

GCM Service for Android should be enabled in API Console and App server needs to get API key as authentication token from the API Console.

The app server sends the actual notification messages, Server API key and registration ID to GCM server.

The GCM server will push the notification message to android application when the device comes to online.

IV. IMPLEMENTATION

In this project, Android Virtual Device (AVD) emulator can be used as target android device. Instead of using multiple physical android phones, multiple AVD can be used as android device. This will reduce the cost of buying multiple phones.

A. Module Implementation:

The three important life flow activities had been done in GCM which would be

1. Enable GCM,

2. Send a message,

3. Receive a message.

Enable GCM:

The GCM service for android device can be enabled by the function called register () in gcm class. This method returns unique registration id (GCM-Reg ID) for the android device. This id should be shared with app server for android application authentication.

Input: Application package name in the manifest file of application and Project ID

Process: GCM server stores this application Id and sends GCM-RegID to android application

Output: GCM-RegID will be the output to android client application

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The following figure 4 describes the UI of this project which enables the initial authentication information sharing between entities.

When the user clicks on the first button called "register with google GCM server", the Register Activity will be called. GCM server gets the input as application ID, Project ID and return unique GCM-Reg ID for the application.

When the user clicks on the second button called" Share GCM-RegID with Appserver", the RegID will be shared with app server.

Send a Message:

The application server sends message to GCM server and the GCM server will queue the messages while the android device is offline. When the device connects to online, the GCM server will push those messages to android device.

Input: Appserver will make http request to GCM server and post the messages and API key value to header of post message

Process: The GCM server will get the server API key and once it is authenticated, it will receive the messages from app server.

Output: The GCM will store and forward the message to android application.

Receives a message:

The GCM server will push the message to android device. Here the application needs not to run in the background in order to receive the message until it is setup with proper intent broadcast receiver permission. Whenever the message arrives at the system, it will broadcast the message to particular android application.

Process: The android device will receive the message sent from application server through GCM server with proper valid registration id.

Output: In order to help for the visually impaired person, the voice is implemented with notification. So the voice will read the text notifications arrived at android application.

V. RESULT

Fig: 4, Register with GCM

Fig: 5, Share GCM-RegID with Application server

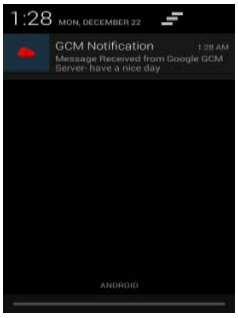

Fig: 6, Message arrived

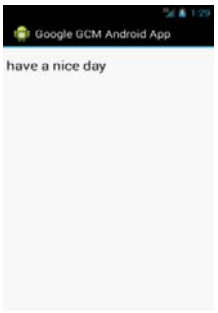

Fig: 7, played with voice

VI. CONCLUSION AND FUTURE WORK

A meta-model for cloud application is presented in this paper. Elastic and flexible application is developed and marketed within the less time using this new technology called Cloud computing. The promise need to be made to reduce the overhead of developing, configuring, deploying, and maintaining cloud applications. Currently, there is no common strategy of having vocabulary, development methodologies, or best practices that distinguish the cloud development technology from the existing ones. Because of lack of standardization and common terminologies, it challenges portability and migration between different cloud platforms. So the cloud consumer will be affected. On the other hand, the lack of software architectural models and design patterns makes cloud application development an ad-hock approach.

The meta-model to address all mentioned problem is defined in this paper for cloud application. So this will have the capability of capturing the syntax and some of the semantics of cloud applications. So the application developer can use this model to better understand the cloud application independent of any specific cloud development environment. This meta-model can be served as a first step toward a cloud modeling language.

Developing two-way communication as from android terminal to application server will be considered as future work of this project.

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

REFERENCES

[1] Google Inc, "Google Cloud Messaging for Android", available: http://developer.android.com/google/gcm/index.html

[2] Wikipedia, "Cloud Computing", available: http://en.wikipedia.org/wiki/Cloud_computing

[3] GIZMODO, "Google's new cloud messaging system does more for less", available: http://gizmodo.com/5921766/android-messaging-is-headed-to-the-cloud

[4] Google, "Android Cloud to Device Messaging Framework", available: http://code.google.com/android/c2dm/

[5] Penghui Li ; Yan Chen ; Taoying Li '; Renyuan Wang ; Junxiong Sun," Implementation of cloud Messaging system Based on GCM service", In proceedings of the 5th International Conference on Computational and Information services(ICCIS), Shiyang, 2013,pp:1509 -1512.

[6] Hansen, J; Gronli, T-M. ;Ghinea, G., "Cloud to device push Messaging on Android: A Case Study", In proceedings of the 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA),Fukuoka, 2012,pp:1298-1303.

[7] Wei- Tek Tsai; Xin Sun; Balasooriya, J, "Service-oriented Cloud Computing Architecture", In proceedings of the 7th international conference on Information Technology: New Generations(ITNG), LAS-Vegas,NV,2010,pp:684-689.

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)