IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 3, March 2015.
www.ijiset.com
ISSN 2348 – 7968

Centralized Access of User Data Channel with Push Notification

Abhishek Priyadarshi[#1], Ritu Karamchandani[#2] , Nikhil Gupta[#3], Arsalan Gundroo[#4],

Department of computer Engineering, D.Y. Patil College of Engineering, Akurdi, Pune 411033
Savitribai Phule Pune Univerity, Maharashtra, India

Abstract-The subscription mechanism provided by various websites has helped in channelizing the user's data sources. But the problem now faced is how to keep the user updated for the availability of new data on that channel, once the user has subscribed for it. Under the solution that we are trying to provide, the user need not visit the site for checking the updates. The app will automatically store the updates by using push notification. Updated information will be collected from different sites and sorting will be done according to date and time. User can directly access updated page. These update notification will mostly originate from the user subscribed contents from a data-source like Face book, twitter, blogs, YouTube and pdf sources such as google drive, whichever provides us the free access. Mobile applications mostly require multimedia data from a server through the mobile network. To update multimedia data in the application with new data in the server, push mechanism is usually used. Push frameworks provide mechanism that a server can notify mobile applications to contact the server to fetch new data on a server. Developers who want to use these frameworks should implement a push gateway which interconnects with the push framework. By integrating the gateways, the server side developers do not have to implement many kinds of gateway functions.

Keywords- GCM, Push Notification, Centralized Access, Youtube Subscription, Polling, Server to server polling.

Introduction

Push notification provide a mechanism to automate the process of keeping the users updated about the recently uploaded data. Once subscribed, the push-notification keeps the user constantly in tandem with the data-source by providing various information about the recent activities on a particular channel. In our app, we will be using push notification mechanism to update the user about the new data that has been uploaded on a subscribed channel. The problem which now a days users are facing is that they have to go and check again and again what are the updates, but this application will solve the problem of users, they need not go again and again on the sites which the user has been using or is the subscriber of the sites. Push notification is type of technique which will send notifications of the updates. Previously polling was the technique which was used but now a days polling is not used since it has major drawbacks such as it consumes a lot of battery power. In this method this app will provide an ease to the user, as only facebook provides push notifications but for other sites which the subscriber has been visiting, the user has to manually check for updates. To update multimedia data in the application with new data in the server, push mechanism is used. For linking website, we have to create our own server which will be linked to the sites which the users are visiting and our server is connected to the GCM (Google Cloud Messenger) and through that GCM it sends push notifications to the user

.

ISSN 2348 – 7968

Push framework provides mechanism that a server can notify mobile application to contact the server to fetch new data on the server. There are many technologies for implementing so we are using Cloud to Device Messaging (C2DM).

This is an android application that enables the users to collect all their subscriptions at one place and get notified by the availability of new data. Current problems faced with subscriptions: 1) Un Centralized Access to web Content 2) Polling Mechanism3) Manual Follow Up.

Example Scenario Suppose a user has subscribed a channel on YouTube by the name ”NDTV”. Now to use the application, the user first logs in to his/her YouTube account to obtain the Author token. Now whenever a new video is uploaded on this YouTube channel, the user will get notified about it.

Literature Survey

To push multimedia information to mobile devices, feature phones have used SMS and MMS [2]. Mobile network operators have offered these services to their subscribers so that they can exchange text messages, images, videos, and photos. However, smart phones need more extended push functions than SMS and MMS. They cannot send messages to a specific mobile application in a smart phone. To solve this problem, Apple introduced APNS and Google introduced C2DM [1].

C2DM

C2DM is a push notification service framework made by Google for Android mobile applications. It is implemented in Android 2.2 (Froyo) or above. It can send messages to the mobile application which is registered at the Google Play. For using the service, the users should login to their Android phones with their Google accounts. The Flow of the C2DM message is very similar with the flow of the APNS message. Service provider generates a notification request and sends it to the C2DM gateway [2]. The C2DM gateway receives the request and forwards it to the C2DM server. Since the C2DM server has active sessions with the C2DM clients, it can send its request to the Android devices. Then Android can wake up the specified application and the application can be active and process the request. The detail operation mechanism of the C2DM.Since there are many kinds of push notification frameworks including APNS, C2DM, and 3rd party push notification frameworks, service provider should have implemented each notification gateway respectively to send multimedia messages to their mobile applications. Also, since mobile devices can lose its connection with a mobile network when they enter the shadow area, the notification messages can also be dropped. Even though some notification frameworks provide retransmission function when the messages were dropped, their retransmission policies might not meet service provider’s requirements. In this paper, an integrated multimedia push framework is introduced. The proposed framework supports APNS, C2DM, and 3rd party push notification framework as well as always-on-based mobile applications. It also supports the store-and-forward function which stores notification messages when the connection with the client is lost and forwards the message when the connection is recovered of the proposed multimedia push framework. If a service provider uses this framework, it does not have to implement all push notification gateways since the framework offers interfaces for sending the messages. The framework has the APNS gateway, the C2DM gateway, and 3rd party push notification gateway so that it can transform the messages from the service provider interface to the messages for each gateway. Also, the framework has storage for storing the lost messages and forwards them when the clients recover their connection with the framework.

Proposed work

Our proposed work is to combine the two concepts- Centralized access which enables the user to access his\her subscription from a single place. Push notification using GCM which notifies the user about some new data on the subscribed information sources and avoids the use of polling.

ISSN 2348 – 7968

Centralized access

A user subscribes various data sources on internet such as YouTube, facebook, an open blog, etc. where he can surf the specific information channels. But for retrieving or surfing such data the user needs to go that particular website to access his\her subscriptions. Also the user has to manually revisit these data channels to check for the upload of some new data. Our idea is to enable the user to be logged in to these websites through our app. User thus, will be able to keep track of all the subscribed channels from a single spot that is our app. This feature keeps the user updated about the availability of some new data, if available on the user subscribed channel. The notification can also provide some metadata about the subscribed channels, such as, number of videos, last update date, etc. This saves the user from memorizing the channels in order to manually check for updates from time to time. Push notification will be implemented using a google messaging service called “Google Cloud Messaging”.

Fig.1 working of push notification

Fig.1 describes stepwise flow of generation of push notification. It shows the working of GCM server. The client will send the request via application, the GCM will send the response with unique registration id. Backend server acts as middleware.

Methods and Algorithms

1) MVC using Spring libraries
2) Server to server polling
3) API consumption

MVC using Spring

The bulk of implementation lies in the server side programming of our app. This has been handled via MVC structure

Controller: The controller is implemented by Action Servlets which map the requests to appropriate methods.

Model: The Model is implemented by servlet methods which are kept distinct for each type of concerned data source.

View: View is the front-end display of the app which is displayed through Android UI itself.

Server to Server Polling

Server to server polling is done to regularly keep check on the source servers about the availability of some new data. This method saves the user device from the overhead of polling. Polling frequency is set to the most optimal value regarding the consumption of resources.

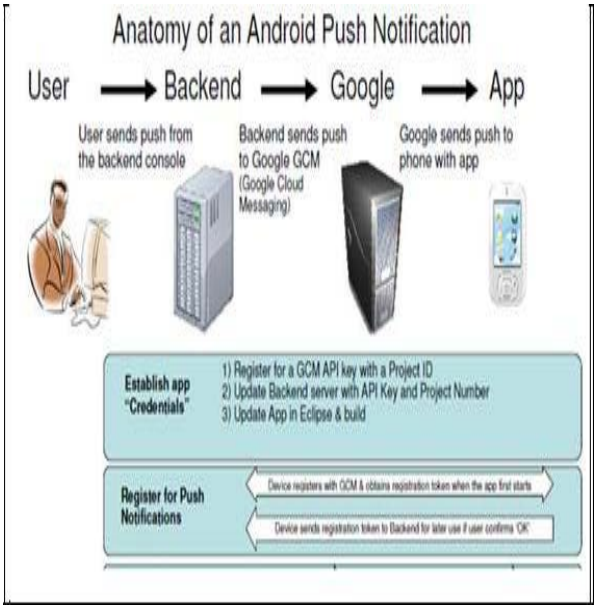

Fig.2 System Architecture

This is system architecture diagram of centralized access implementing push notification system. User in fig.2 is replaced by Youtube server whenever the backend server finds some new data on Youtube channel, it posts a message to the google cloud server which indicates it to generate push notification to the respective client.

ISSN 2348 – 7968

Mathematical Modelling

Let S = {I, O, F, Fc, Sc, Ss, Fs} be set of tuples.
Where s= system,
I = Input which contain API registration key, GCM client registration and user subscription, category titles.
O = updated notification (output)
F (Function) = Polling,Add & Delete Subscription Add new category, display categories.
Fc (Failure cases) = Failure of updated notification, delay of notification, failure of API registration unable to access particular subscription.
Sc (success case) = Timely updated of notification, page redirection for updated notification, access of subscription through link maintain through categories.
Ss (System start) = Login (user name, password) page starting state.
Es(End state) = Redirected page.

Result

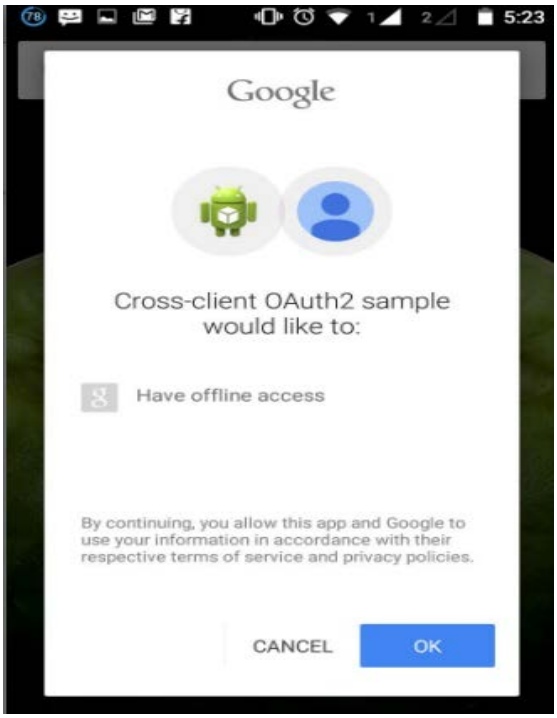

Fig. 3 client authentication

It is most secure authentication mechanism to access user's private data.

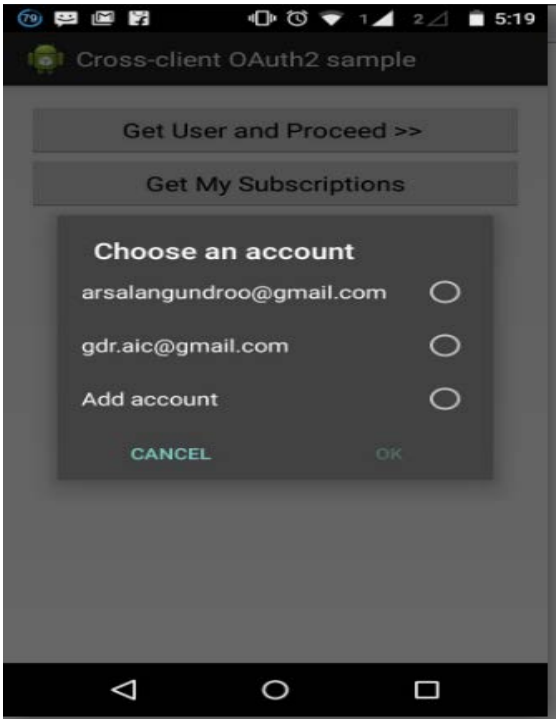

Fig. 4 creating an account

It synchronizes with user's accounts on device and saves from entering passwords.

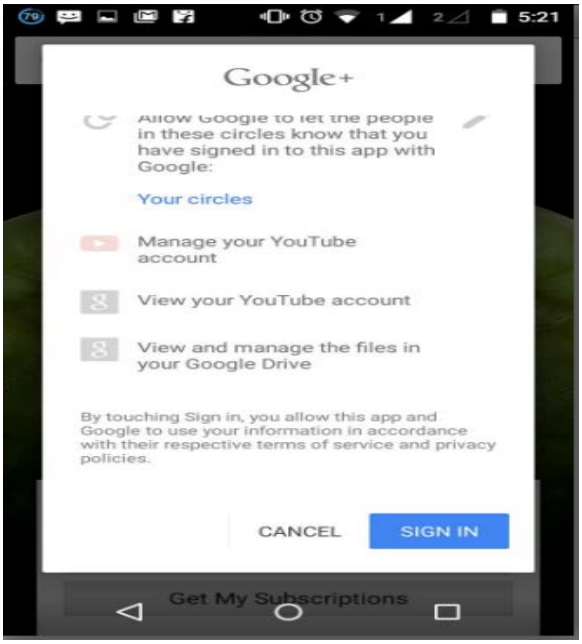

Fig. 5 signing in

It allows to get people in circles know that you have signed in to this app with google.

ISSN 2348 – 7968

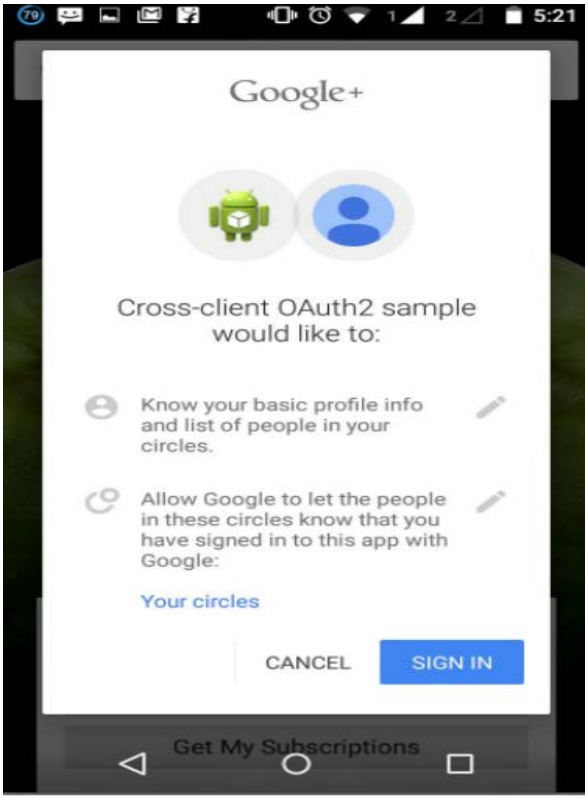

Fig. 6 provides basic profile info

Fig. 7 updated notification

CONCLUSION

This Android application implements new multimedia push framework which integrated existing push framework gateways and provided a store-and-forward function. By using this framework, mobile application developers do not have to implement diverse push framework gateways, which can reduce complexity of interworking function. Also, the developers who use the proposed framework do not have to concern about losing a push notification message in the air since the framework stores the lost message and forward it when the mobile device can connect with a mobile network.Our benchmarking test consisted of a client installed on the mobile devices and a server application running on the Google App Engine. The client had fixed time intervals where it would request a push message from the server.

REFERENCES

[1] H. Y. Ho and L. Y. Syu, “Uses and gratifications of mobile application users”, Proceedings of the International Conference on Electronics and Information Engineering, (2010).

[2] Q. H. Mahmoud and P. Popowicz, “Toward a Framework for the Discovery and Acquisition of Mobile Applications”, Proceedings of the Ninth International Conference on Mobile Business and Ninth Global Mobility Roundtable, (2010).

[3] I. Podnar, M. Hauswirth and M. Jazayeri, “Mobile push: delivering content to mobile users”, Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops, (2002).

[4] Apple Inc., “Apple Push Notification Service”, http://developer.apple.com/library/mac/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/ApplePushService/ApplePushService.html, (2009).