ISSN:0975-9646

A Novel Approach for Cloud Integrated Gateways

G.V.M.Vasuki[1], A.Mithun[2], B.Radha[3], C.N.Chaitanya[4], L.Yashoda[5]

[1] Professor, [2,3,4,5]Student,
Department Of CSE, LENDI Institute Of Engineering & Technology,
Vizianagaram ,AP, India

Abstract-In this paper, we focus on mobile cloud computing newest applications of cloud computing which is recently seeing explosive adoption since the capabilities of mobile device are enhanced by moving the computing power and data storage away from mobile phones into the cloud. Smartphone enables a new, rich user experiences but their hardware is still very limited in terms of computation memory, and energy reserves, thus limiting the potential applications. Here we present a novel architecture that addresses these challenges by partially off-loading execution from a Smartphone's to a computational infrastructure hosting a cloud of Smartphone clones. Our paper mainly deals with the application in mobile environment with which the user can have multiple benefits as he can use the basic application and also he can access different gateways from our application through which the APIs don't require any browsers. The application itself gets the gateways api into the application. Even though the application needs the internet to be accessed but it doesn't uses any of the browser embedded in your device. The general application deals with an basic android message application in which the user can text a message and send it to the particular recipient using that individual GSM provider. The basic application generally doesn't require any mandatory internet facility for the client side environment. This application also provides security for the users.

Keywords: GSM provider, android system, integration of gateways, mobile cloud, web view.

INTRODUCTION

Mobile Cloud computing at its simplest refers to an infrastructure where both the data storage and the data processing happen outside of the mobile device. Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing applications and mobile computing to not just Smartphone users but a much broader range of mobile subscribers. General overview of how the applications from the mobile device getting integrated with the cloud and the cloud again getting communicated with the different servers which are embedded in the application.

The application deals with a general messenger application for a mobile device in android environment which in interior communicates with different API's gateways using the Oauth technology. OAuth is an open standard for authorization. OAuth provides a method for clients to access server resources on behalf of a resource owner (such as a different client or an end-user). It also provides a process for end-users to authorize third-party access to their server resources without sharing their credentials (typically, a username and password pair), using user-agent redirections. The integration between different components is done using web view concept

Web-View:

A Web-View is like a tab in a browser. You load pages into it, interact with it, and display it in some pre-defined manner.There are two different types of Web-Views:

Off-screen Views

Off-screen Views are rendered continuously to a pixel-buffer surface (see Surface). It is your responsibility to display this surface in your application and pass all mouse/keyboard events. This gives you the flexibility to display web-content in any environment (such as a 3D engine or headless renderer).

In Awesomium.NET, we provide predefined surfaces for every technology that take care of copying the native pixel buffer to managed equivalent bitmaps and help you render the buffer to technology specific surfaces.

Windowed Views

Windowed Views are rendered directly to a platform window (HWND, NSView, etc.) and capture all input themselves.When using a windowed WebView on MS Windows, you should call WebView. ParentWindow immediately after creating this type of view (the view cannot create a window until a parent is set). Technology specific WebControls handle this internally.

The IWebView Interface

The IWebView interface exposes the full API of a native Awesomium Web-View, as well as .NET specific API added to web-views. It is provided for advanced coding and in certain scenarios, it allows you to override much of the NET internal logic and call on the native web-view directly.

All managed Web-View components, implement this interface. It provides most of the common API of all Awesomium.NET Web-View components. It also exposes all the events that a web-view component can fire.

In Awesomium.NET documentation and articles, a Web-View instance is usually referred to as an IWebView instance and common members of this interface implemented by each Awesomium.NET Web-View component are referenced through this interface.

For more details, read the documentation of I WebView.

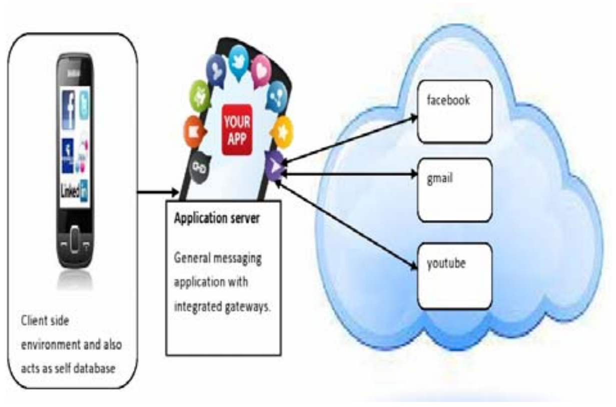
While WebViews give us the ability to leverage the power of the web in our apps, they do come at a cost. The WebView is one of the most expensive components to create in terms of resources and performance. Every WebView loaded requires its own rendering context and will take a moment to load, regardless of how simple its contents are.

For these reasons, it's critical that you are absolutely certain you need a Web View to create a specific functionality in your app. A good rule of thumb is that if you can recreate web content using Titanium native components, then you should do so. Minimal usage of Web Views will not likely be a problem, but their resource requirements can quickly become unmanageable if continually used. A prime example of this would be usage of WebViews in Ti.UI.TableViews. Performance is of utmost concern withTableViews and including WebViews in your TableViewRows is a recipe for sluggish performance.

About gateways:

Here we are integrating different gateways into our application in which each and everyone can easily access and useful to everyone in day-to-day life such as YouTube, facebook, Gmail e.t.c., the briefing about the gateways is discussed in the architecture.

ARCHITECTURE


YOUR APP
facebook
gmail
youtube
Application server
General messaging application with integrated gateways.
Client side environment and also acts as self database

From the concept of MCC, the general architecture of MCC can be shown in Fig. mobile device is connected to the mobile networks itself in it that connect to the mobile networks and mobile devices. Mobile user's requests and information can type a message in the basic application and also can able to send to the user who is about to receive the message using the GSM provider who is the middle ware between the application client and the recipient who is going to receive the message. Here mobile network operators can provide services to mobile users as the messaging application and browsing the integrated gateways such (facebook, Gmail, YouTube e.t.c) based on the home agent (HA) and subscribers' data stored in databases. After that, the subscribers' requests are delivered to a Cloud through the Internet. In the cloud, cloud controllers process the requests to provide mobile users with the corresponding cloud services. These services are accepted with the concepts of utility computing, virtualization, and service-oriented architecture (e.g., web, application, and database servers).

RELATED WORK

Generally there exists a message sending application which is a proposed system for us. In the existing system we are going to integrate few API's into that particular application. In our proposed system the user can access the proposed systems messenger application and also he can use the integrated applications using webview technology. Here we want to access the servers of Gmail, YouTube and facebook into our application using cloud concept. The key feature of our application is that we don't use any browser in our application to connect to the servers of any integrated gateways. Instead we will get the API of the different gateways into our application. The proposed system is a client-server paradigm for the gateways that are accessed using our application and the basic application is peer-peer application. Many applications now-a-days use this concept of integrating different servers into a unique platform but our application is maintaining a difference as we are integrating the servers into a basic messaging application through which we can access few social websites as mentioned earlier. And the most useful aspect in our application is that we can send few messages without using mobile data connection as it acts just like an interface for sending messages when we deployed this application in our mobile devices. For accessing the gateways which are provided in our application the mobile should have mandatory mobile data connection as we are utilizing the services from the server. Our application is a different kind of approach on the basis of implementation which involves a webview technology rather than a browser when we are using the gateways.

METHODOLOGIES FOR INTEGRATED GATEWAYS

The basic application is just a message application in android environment in which the user is able to send a message and get an acknowledgement for that particular message whether it is sent or not. It is done by getting the permissions from the GSM service provider such that the message is to be sent from the application provider in the perspective of any of the GSM provider. Those permissions are granted from the android interface by the following package

android.telephony.gsm.SmsManager;

And the basic application runs in following process

```
SmsManager s=SmsManager.getDefault();
s.sendTextMessage(phone No, , SMS, ,);
Toast.makeText(getApplicationContext(),"SMS
Sent!...",Toast.LENGTH_LONG).show();
```

And now the integrated applications works under the principle of the main concept called as webview in android environment. As our paper determines the integration of the gateways into our application here comes the main description of how the code for the different gateways be done .this integretation of gateways can be done by using the concept of web view. The webview class is an extension of Android's view class that allows you to display web pages as a part of your activity layout. It does *not* include any features of a fully developed web browser, such as navigation controls or an address bar. All that webview does, by default, is show a web page common scenario in which using webview is helpful is when you want to provide information in your application that you might need to update, such as an end-user agreement or a user guide. Within your Android application, you can create an activity that contains a webview, then use that to display your document that's hosted online.Another scenario in which webview can help is if your application provides data to the user that always requires an Internet connection to retrieve data, such as email. In this case, you might find that it's easier to build a webview in your Android application that shows a web page with all the user data, rather than performing a network request, then parsing the data and rendering it in an Android layout. Instead, you can design a web page that's tailored for Android devices and then implement a webview in your Android application that loads the web page. The user should have to get the permissions from the internet for accessing the gateways

```
<uses-permission android:name =
"android.permission.INTERNET" />
```

The main gateway that the particular user want to access is by default given in the following code in the place of example

```
Uri Uri = Uri.parse("http://www.example.com");
 Intent intent = new Intent(Intent.ACTION_VIEW, uri);
 startActivity(intent);
```

To provide a WebView in your own Activity, include a in your layout, or set the entire Activity window as a WebView during on create();

```
 WebView webview = new WebView(this);
 setContentView(webview);
```

Then load the desired gateway using:

```
webview.loadUrl("http://slashdot.org/");

 String summary = "<html><body>You scored
<b>192</b>points.</body></html>";
 webview.loadData(summary, "text/html", null);
```

the more complicated thing is that an error message is shown if the following is not done:

```
webview.getSettings().setJavaScriptEnabled(true);
```

The screen resolution is also one of the criteria for the application. The screen density of a device is based on the screen resolution. A screen with low density has fewer available pixels per inch, where a screen with high density has more — sometimes significantly more — pixels per inch. The density of a screen is important because, other things being equal, a UI element (such as a button) whose height and width are defined in terms of screen pixels will appear larger on the lower density screen and smaller on the higher density screen. For simplicity, Android collapses all actual screen densities into three generalized densities: high, medium, and low.

IMPLEMENTATION

This section of the document illustrates different functions provided by the application:

ENTERING BASIC APPLICATION IDE

Description:

This feature of the application shall provide android application which is used to get the IDE where the user can handle only one user interface which shows the place to be entered to show.

Rationale: Once the user starts the application, he/she has an option for entering the location he is searching for. He/she can also have another option of choosing other gateways included in the application.

Requirements: In order to use this application, the User should have a mobile phone which runs on Android Mobile Operating System. At any point of time only one program can be written on the screen.

Using the messenger application

Description:

This feature provides the user to enter desired location to be written and the basic application features were available with Google maps application.

Rationale:

Once the name is entered, the user can get the location and landmarks exactly where he had entered .he can zoom to get the exact position where he is searching for. And also he can search the location by city name or village or country.

Requirements:

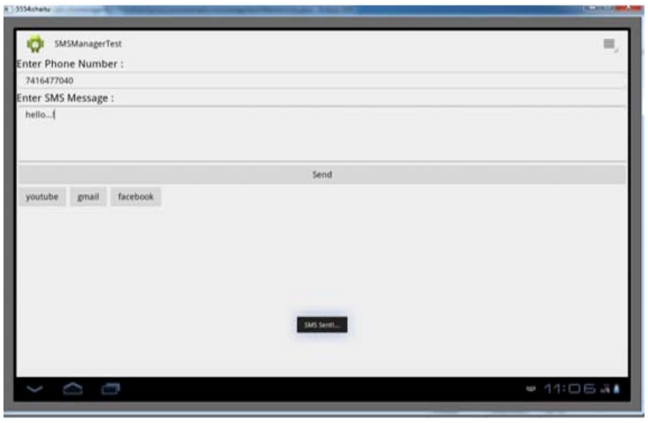
In order for the user to able to use this feature of the application, he/she should have a mobile phone which runs on Android Mobile Operating. At any point of time only one program can be compiled.

Using other API's in the application

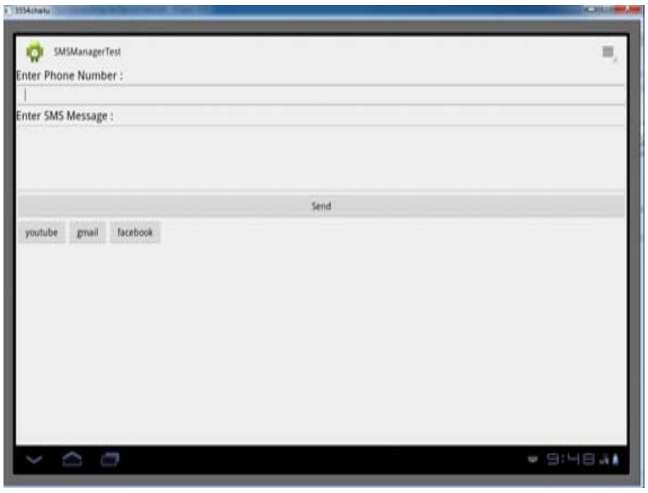
Description: This feature provides the user to select an application to be viewed and can access that applications server in the current application that we are using.

Rationale:

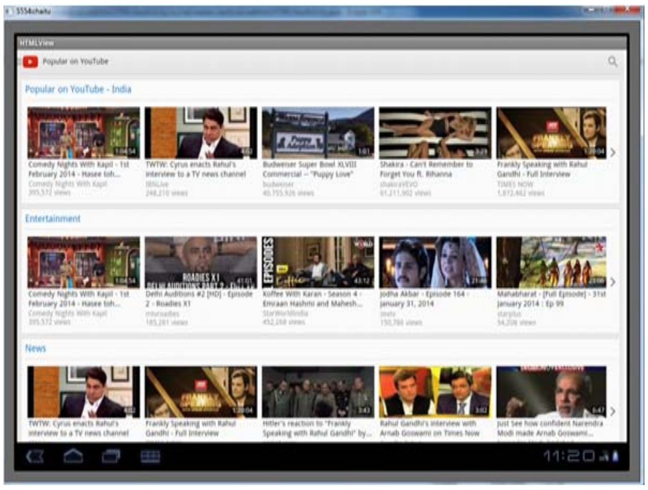
Once the particular API is selected the user can access that particular applications server in the current host i.e. application which is deployed in the device.



SMSManagerTest
Enter Phone Number :
7416477040
Enter SMS Message :
hello...



SMSManagerTest
Enter Phone Number :
Enter SMS Message :


5554chaitu
HTMLView
Popular on YouTube
Popular on YouTube - India
Comedy Nights With Kapil - 1st February 2014 - Hasee toh...
TWTW: Cyrus enacts Rahul's interview to a TV news channel
Budweiser Super Bowl XLVIII Commercial -- "Puppy Love"
Shakira - Can't Remember to Forget You ft. Rihanna
Frankly Speaking with Rahul Gandhi - Full Interview
Entertainment
Comedy Nights With Kapil - 1st February 2014 - Hasee toh...
Delhi Auditions #2 [HD] - Episode 2 - Roadies X1
Koffee With Karan - Season 4 - Emraan Hashmi and Mahesh...
Jodha Akbar - Episode 164 - January 31, 2014
Mahabharat - [Full Episode] - 31st January 2014 : Ep 99
News
TWTW: Cyrus enacts Rahul's interview to a TV news channel
Frankly Speaking with Rahul Gandhi - Full Interview
Hitler's reaction to "Frankly Speaking with Rahul Gandhi" by...
Rahul Gandhi's interview with Arnab Goswami on Times Now
Just See how confident Narendra Modi made Arnab Goswami...
11:20

Requirements: In order for the user to able to use this feature of the application, he/she should have a mobile phone which runs on Android Mobile Operating. At any point of time only one application can be run at once in the device.

CONCLUSION

This paper has shown what an android application is doing in the real time environment not only as an single application but also as an multiple servers using in a cloud in one application. it is described how to set an basic android application and integrating the newer applications into the different applications via gateways into the basic application..Also there is an issue of version of android as it doesnt work on the android basic version i.e., 8th version but the application only runs on the 10th and plus editions only.Therefore our application is very friendly with the client side environment as it includes different applications into one application. And the internet connection is mandatory for the different gateways that are to be accesed from our application but for the default general messenger only gsm service is sufficient but not the internet connection.

REFERENCES

1) Murphy M. L., Android Programming Tutorials, 2009. Commons Ware, ISBN 978-0-9816780-2-3

2)http://grepcode.com/file/repository.grepcode.com/java/ext/com.google.android/android-apps/2.0_r1/com/android/mms/ui/ComposeMessageActivity.java

3)https://github.com/Pirngruber/AndroidIM/commit/33a56ebf96d2c475a6313638395929cce36c4844

4)http://androidexample.com/Device_To_Device_Messaging_Using_Google_Cloud_Messaging_GCM_Android_Example/index.php?view=article_discription&aid=122&aaid=142

5) http://stackoverflow.com/questions/7461879/looking-for-android-viewflipper-example-with-multiple-webviews

6) http://stackoverflow.com/questions/19508436/browse-multiple-pages-in-webview-on-menu-selection

7) http://www.mkyong.com/android/android-webview-example/

8) http://www.techrepublic.com/blog/software-engineer/try-androids-useful-drag-and-drop-api/#.

9) Goel, Utkarsh, Kanika Shah, and Mohammed Abdul Qadeer. "The personal SMS gateway." *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*. IEEE, 2011.

10) Shah, Sumiran, et al. "Zip it up SMS "A path breaking model in the field of mobile messaging"." *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE Internation-al Conference on*. Vol. 4. IEEE, 2010.

11) Wang, Xibo, and Yanting Yang. "Method and Implementation of Sending and Receiving Mobile Phone Messag-es." *Computer Science-Technology and Applications, 2009. IFC-STA'09. International Forum on*. Vol. 1. IEEE, 2009.

13) Creating Android Applications: Develop and DesignChrisHaseman .