RESEARCH AND APPLICATION OF CARRIER-GRADE MASHUP PLATFORM ON ONEAPI

Jie Di, Haihong E. and Lin Ma
PCN & CAD Center, Beijing University of Posts and Telecommunications, Xitucheng Street, Beijing, China

Keyword: Mashup platform, GSMA OneAPI, RESTful web service.

Abstract: The emergence of Mashup enables its development more convenient, with more and more service providers open their own API. Although users join the Mashup developing group using the service provided on internet, there are hardly any telecommunication services supplied. Based on the characteristics of Mashup technology, in this paper we provide a new Mashup application system framework based on the GSMA OneAPI interface and REST Web Service, which access telecommunication services to Mashup. The application we present demonstrate the feasibility of carrier-grade Mashup application, and indicate that OneAPI interface and Mashup application in the information field covering Internet and telecommunications network has great potential.

1 INTRODUCTION

Mashup application is a web application which integrates content retrieved from multiple external data sources (Jackson C, 2007). Having been widely used recently, the type of it includes maps Mashup, video and image Mashup (Merrill D, 2006), and other types of search and shopping Mashup. Currently, most web service providers such as Yahoo, eBay, Amazon, Google has been providing Web services API, which allows developers to use the Mashup API to build the corresponding applications, making this new type of Web application model become reality. Mashup attracts academic and industrial with its characteristic of more reusability, based on Web, lightweight and end users oriented (Liu Xuanzhe, 2007).

In this paper, we adopt the OneAPI provided by GSMA and Google map API to construct a Mashup application.

And we introduce the feasibility and potential of Mashup application which based on the REST web Service performing in information sharing and application development.

In this article we first introduce some related technologies, then as an application for instance, we provide the system structure and implementation process of Mashup application.

2 CONCEPTION AND FEATURES OF GSMA ONEAPI

2.1 Conception of the GSMA OneAPI

The GSMA OneAPI (GSMA OneAPI Reference, 2008) is an initiative to define a commonly supported API to allow mobile (and other network) operators to expose useful network information and capabilities to a Web application developers. It aims to reduce the effort and time needed to create applications and content that is portable across mobile operators. The project aims to reuse existing standards (or proper subset thereof) as well as advise standards bodies as to what Web developers expect from network operator APIs, so that such standards can evolve accordingly.

The project is a work in progress we appreciate your feedback via the forum to help us improve the APIs.

2.2 Work Mode of OneAPI

The GSMA OneAPI is intended to complement existing client-side and server-side APIs by providing a missing piece: access to network capabilities and information, regardless of operator. As such it can be accessed by client and server side technologies when required; although for reasons of

Di J., E. H. and Ma L..
RESEARCH AND APPLICATION OF CARRIER-GRADE MASHUP PLATFORM ON ONEAPI.
DOI: 10.5220/0003585505040509
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (SSE-2011), pages 504-509
ISBN: 978-989-8425-53-9

authentication and security it is likely that client APIs will not call the GSMA OneAPIs directly, but instead by invoking a server-side API to make the call. For example the OMTP BONDI framework includes a 'wrapper' API that can be used to encapsulate a OneAPI call if needed.

2.3 Location&SMS API

The OneAPI location function can acquire location information of one phone through GPRS networks anytime and anywhere, in conjunction with the relevant other message it may develop a good application. We can develop a emergency rescue apply by combining the position information, electronic map, traffic information and other information. If the tourist attractions combined, we also can carry out many tour guide's information, and of course, there are many other good business, such as advertising and children's custody. if thinking openly and deeply, there would be many other apply.

The OneAPI SMS interface allows a Web application to send and receive SMS messages. To receive SMS sent to your Web application by end users, you will need to obtain a registration Id (e.g. short code or similar identifier) to identify your application to the network for correct routing.

Mashup Based on mobile network has a very wide range of application (LI Yan, 2008) The application of the Internet combined with mobile network can offer more abundant application. The application of Mashup based on location is specific characteristic of network, which can acquire location information at anytime and anywhere through mobile network.

3 MASHUP PLATFORM

3.1 Introduction of Mashup

As a new type of data integration application based on Web on the internet Mashup is gradually arisen. Mashup technique was originated from pop music, which is a new song constituted by audio tracks mixed by singing and instruments two different songs. According to the explanation of Wikimedia (http://en.wikipedia.org/wiki/Mashup,2010), Mashup is a new Web service formed by stacking a number of different application which support Web API. It uses the content retrieved from external data source to create new innovation service, which will combine contents come from more than one data source, creating more value-added services. As more Web site open its API, Mashup applications constructed by using eBay, amazon.com, Google and Yahoo APIs appear ceaselessly, which make this kind of new Web application mode become a reality.

In IBM Developer Works of China's description (IBM QEDWiki, 2010), a Mashup application can be divided into three different components: the API/content providers, Mashup sites and Mashup client (i.e. user's browser).

The main advantages of Mashup development modes are as follows:

- Development form simplified, which is a light and convenient development mode.
- Advocate to use again, which can reduce duplication of work.
- Data sources are rich.
- Based on open standards, easy to realize sharing and interoperability of data and service.
- Development cost is reduced greatly.

3.2 The Differentiation between Mashup Technique and Traditional Integration System

With the rapid development of information technology, integrated information get more and more attention, such as enterprise informationization, EAI (enterprise information integration) etc. On the information integration, there are many differences between integration scheme adopting the traditional data and the technology of Mashup based on Web2.0:

- The requirements of platform and system. Based on the traditional system integration program needs the support of the traditional platform API layer, such as the J2EE platform and the. NET platform, and integration process is very complicated. Using Mashup technology to build applications, the integrate process is relatively simple. It can easily aggregate and restructuring the content from different places, including the services and data, to conduct agile business development, fast to meet their changing needs due to different scenes.
- Supporting for new features and functionality extensions. Data integration based on the traditional way required programming to complete the system extensions, involving the presentation layer, business logic and data storage layer or multi-layer structure, with poor scalability; Mashup-based technology uses loosely coupled approach to integration information to create new applications and functions, because it is loosely coupled manner, it could greatly improve system compatibility and scalability.
- Presentation layer support. The presentation layer of traditional system integration program in the traditional integration mode mostly use a

graphical user interface GUI, when the client sends a request, you need to load the entire page; Mashup technologies based on Web2.0 uses a technology called Ajax to call a web applications, browser clients do not need to reload the entire page each time it communicating with the server, which has a great dynamicity.

- Bottom integrated technical support. Based on the traditional data integration solutions use multilayer enterprise application integration technology, involving different layers of integration, such as business process integration and application integration and data integration, etc.; Based on the Web2.0 Mashup technology uses the Service oriented architecture (SOA) ideological to integrate underlying system, such integrated system has a modularization, loose coupling feature, and also a good characteristic of extensibility.
- The physical location of the content aggregation. Traditional web application is a mode based on HTML page and server-side data transmission, and HTML is suitable for text. With the increasing complexity of web applications, the traditional web application can not meet the gradually higher, all-round experience requirements of web browser. At present, many Mashup site choose to aggregate content on the client machine, data can be cached on the client, in order to achieve a user interface with a faster response than that based on HTML and a fewer number of times the data transporting between the fewer. The interface provides fast response time without refreshing the page.

3.3 The Framework of Mashup Application

The framework of Mashup applications is composed of three different parts, they are logically and physically disconnected from each other (Duane Merrill, 2006) (possibly separated by the network and organizational boundaries): service providers (including the API interface and data sources), Mashup Web site and client browser. Framework is shown in Figure 1:

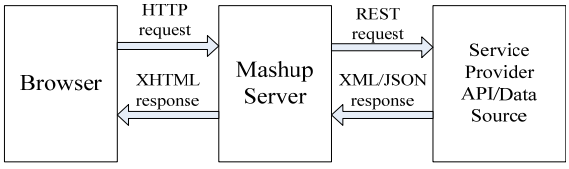

Figure 1: Framework of Mashup application.

- API interface or data source. They are generally independent providers of content or interface of each other. In order to facilitate data retrieval, the provider will usually provide outside their own content through the Web protocol outside. However, there are many potentially valuable data source did not easily provide API out. However, developers can also use screen scraping technology to achieve data acquisition.
- Mashup site. That is where the Mashup exists. It is very interesting that this simply because this is the place where Mashup logic being, rather than the implementation of these logical. On the one hand, Mashup can use server-side dynamic content generation technologies (such as Java Servlets, CGI, PHP or ASP) to implement Web applications similar to conventional ones.The client-side logic is often a combination of directly the code embedded in the Mashup Web pages with these scripting API reference library or applet (provided by the content provider) in Web pages.
- Client Web browser. This is where the application presented graphically, and where the user interaction takes place. Mashup often use client-side logic to build integrated content.

4 APPLICATION OF THE MASHUP PLATFORM BASED ON ONEAPI

4.1 System Architecture

In this paper we use a kind of Mashup architecture, which can create combination service. It consists of Mashup API access components, Mashup server and Mashup generator, corresponding SOA (Luo Xiaoxiang, 2008) of service providers, service agent and service users executing function expansion.

Mashup API access components will access all the Meta-services various format provides including API, RSS or screen grab to system; Mashup server functions include service directory, Mashup component storage; Mashup generator completing the user in the browser Mashup component selection and complete service combine function.

In this case, we will use the Location API of OneAPI which GSMA provides and Google Map API to create an application, then transform the response of an XML format from the request URL to JSON format, finally display the result in the output of system, thus, the user can query the location information of the cell phone number user inputs, and marked in the Google Map, realizing the telecommunication level Mashup service.
System structure chart is shown in Figure2:

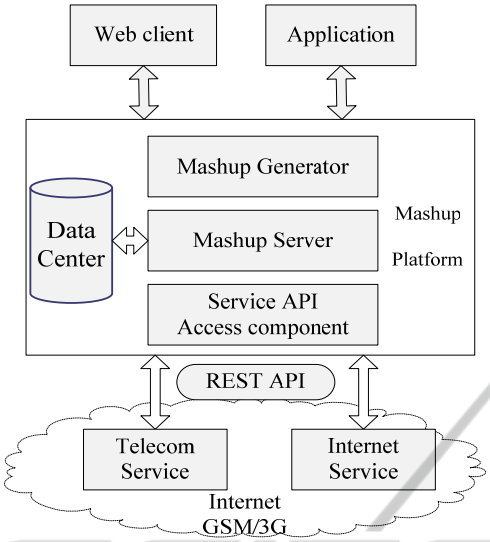

Figure 2: System Architecture.

4.2 The Key Technology

4.2.1 Ajax Building API Engine

Ajax is a Web application model, rather than a specific technology. It includes several asynchronous loading and present technologies of content attention:

- The XHTML and CSS used to determine the present style;
- Browser provided document object model (DOM) API for dynamic display and interactive;
- Asynchronous data exchange, usually XML data;
- The browser script, mainly JavaScript.

With these techniques used together, they aim to create smooth, good Web user experience for users through small amounts of data with content server, not having to perform reloading and restart presents the entire page after certain tasks. Google Maps API(Google Mashup Editor, 2010), (Google Maps, 2010) includes a special Ajax engine, it having a really strong influence on user experience: its work way similar to a real native application, of which no scrolling can operate, also did not have a move button with compulsory page reloading.

4.2.2 REST Web Services Access

Web Service is some Independent, modular application modules that can be described, published, located, and invoked through the Internet. Its purpose is to build a common platform-independent, language-independent technology layer on variety of heterogeneous platforms presently, to ensure the applications interoperable on different platforms.

REST is the acronym for Representational State Transfer (Fielding RT, 2000), this is a only use HTTP and XML, Web-based communication technologies. Its simplicity and lack of strict configuration files make it isolated with SOAP, and has attracted widespread attention. REST fundamentally supports only a few operations (i.e. POST, GET, PUT, and DELETE), these operations apply to all messages. REST emphasizes the information itself, referred to as resources (Eric van der Vlist, 2007)

4.2.3 Return Data in JSON

As to Ajax Web 2.0 site application, JSON is the most flexible lightweight solution(MILLER M S,2009), such as at present more popular Mashup Web services.

Same as XML, JSON is a data format based on pure text. JSON data format is very simple, we can use JSON transmit a simple String, Number, Boolean, also can transmit an array, or a complex Object. Through comparing the characteristics between XML and JSON, you can find JSON simplified data coding and visit, so as to increase the efficiency of data transmission.

4.3 Application Realization

4.3.1 User Interface

The front interface design uses Ext.

Register the required services API into the system services database with JavaScript (Nilanjan Banerjee, 2008), and generates the corresponding window in the front interface for users.

Input the form according to function. Form contains a String, Long and other types of input box (Inputs array), by editing the input box you can pass the required parameters. Click the Refresh function key to display the results in the output window.

Call Location Services, you can enter the phone number in the window to get its location information, as shown in Figure 3:

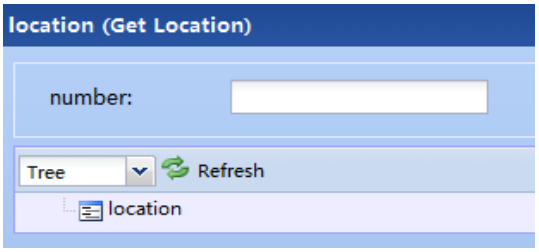

Figure 3: Service invoke window.

4.3.2 Servlet Calling Service API

Servlet is a server-side Java applications, with platform-independent and protocol-independent characteristics, which can generate dynamic Web pages. It bears a mid layer between customer request (Web browser or other HTTP client) and the server response (HTTP server database or application). Servlet is Java application located on server-side inside the Web server, loaded by Web server, which must include Java virtual machine supporting of Servlet.

Request the URL corresponding to the API, while validating some information parameters, then deliver the request by invoking of HTTP GET method, and get the response.

```
  String numb =
req.getParameter("number");//get the
phone number user inputs
  String getUrl = SANDBOX_ENDPOINT +
FORWARD_SLASH + sandbox + FORWARD_SLASH +
subscriber; //API URL
 GetMethod getMethod = new
GetMethod(getUrl);
 Int getStatusCode =
client.executeMethod(getMethod);
 System.out.println("HTTP status code: "
+ getStatusCode + " " +
getMethod.getStatusText());
 String
response=getMethod.getResponseBodyAsStr
ing();//get response
```

4.3.3 Parse the Data in JSON-format

The response of OneAPI Location interface is XML-format data, we need to transfer them to JSON-format in the backed system.

We import net.sf.json.jar in the Servlet which invoke the service API, using it to complete the JSON transform, then display them in certain format the Mashup output system required.

```
 XMLSerializer xmls = new
XMLSerializer();
       JSON json =
xmls.read(response);//transform to JSON
format
    String res = callback + "('" + index
+ "', "+ json +", 200, null, 200)";
 resp.getWriter().println(res);
 resp.getWriter().close();
```

The results are shown in Figure 4:

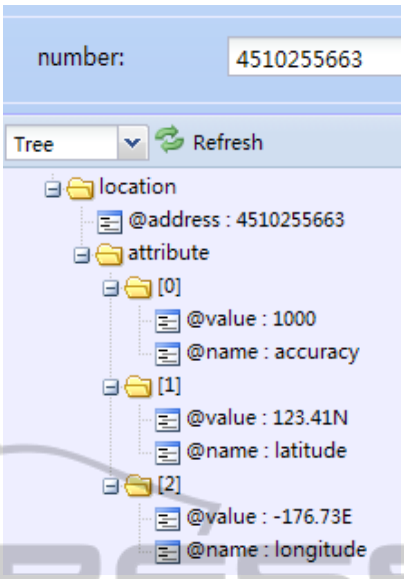

Figure 4: The location result.

4.3.4 Location API combined with Google Map API

The result of Location API as listed below:

```
{"@address":"4253658456","attribute":
[{"@value":"1000","@name":"accuracy"}
,
 {"@value":"45.28N","@name":"latitude"
},
 {"@value":"9.10E","@name":"longitude"
}]}
```

There include the latitude and longitude information data of the phone.

Combine the location information data with Google Map API, marking the location of the phone in Google Map by the latitude and longitude data we already got.

The results are shown in Figure 5:

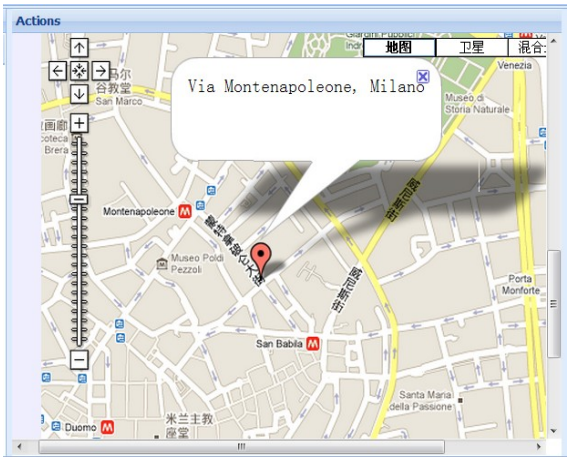

Figure 5: The result marked in Google Map.

5 CONCLUSIONS

In this paper we analyze the characteristics of Mashup technology, Mashup has many advantages, development in simplified forms, advocating the reuse, rich data source, based on open standards, the development cost reduced greatly, etc. It allows the user to develop Web application in a simple and interesting process, fully embodies simple, user participation and resource sharing characteristic of the Web2.0.

Based on this, we present a Mashup development platform based on a carrier-grade OneAPI service, and introduce the system architecture and its key technologies, finally we verify the effectiveness of this Mashup platform through an application.

We believed that Mashup applications could bring more abounding experience for users and promote telecommunications business development.

ACKNOWLEDGEMENTS

This work is supported by the National Key project of Scientific and Technical Supporting Programs of China (Grant Nos. 2008BAH24B04, 2008BAH21B03, 2009BAH39B03); The National Natural Science Foundation of China Grant No.61072060; The Program for New Century Excellent Talents in University No.NECET-08-0738; and the Innovation Technology Star Program of Beijing (Grant No.2007A045).

REFERENCES

Jackson C, Wang H. Subspace:Secure cross-domain communication for web Mashups [C]. PJ *Proceedings of the 16th International Conference on World Wide Web WWW'07*. New York:ACM,2007:611—620.

Merrill D. Mashups: New member in Web Application [EB/OL].2006-08.http://www.ibm.com/developerworks/cn/xml/x-mashups.html.

Liu Xuanzhe, Hui Yi, Sun Wei. et a1. Towards service Composition Based On Mashup [C]. 2007. *IEEE Congress on Services*. 2007:332—339.

GSMA OneAPI Reference. www.gsmworld.com/oneapi.

Li Yan. Telecommunication Mashup Service Architecture. http://labs.chinamobile.corn/community/my_blog/9339/ 9355.

Mashup (Web application hybrid). http://en.wikipedia.org/wiki/Mashup.

IBM QEDWiki. http://services.alphaworks.ibm.com/qedwiki.

Google Maps. http://en.wikipedia.org/wiki/Google_Map.

Duane Merrill. Mashups: the new breed of Web app IBM Developerworks, 2006, http://www.ibm.eom/developeworks/xml/Librmy/x-mashups.html?S_TACT=105AGX52&S_CMP=cn—a—x

Luo Xiaoxiang, Xu Huiyang, Song Meina, et a1. Research On SOA-based platform to enable mobile Mashups. In: *ICCT 2008*. Hangzhou, China, 2008.

Google Mashup Editor. http://editor.googlemashups.corn.

Fielding RT. Architectural Styles and the Design of Network-based Software Architecture [A]. *Doctorial Dissertation*, Dept. of Computer Science [C]. Univ. of California, Irvine, 2000.

Eric van der Vlist. *Professional Web 2.0 Programming* [M]. Wiley Publishing, 2007.

Miller M S, Samuel M, Laurie B, et al. Caja:Safe active content in sanitized JavaScript [EB/OL]. 2009, 09,20. http://code.google.com/p/google-caja.

Nilanjan Banerjee, Koustuv Dasgupta. Telecom Mashups: enabling Web2.0 for telecom services. In: *Proceedings of the 2nd lnternational Conference. Suwon, Korea*, 2008.