WEB SERVICES FOR MOBILE COMPUTING

Piyush M.Patil ,
Computer Department,University Of Mumbai,
Mumbai,India,Mob-9699398650

Kushal Gohil,
Computer Department,University Of Mumbai,
Mumbai,India,Mob-9323916806

Rohit Madhavi
Computer Department,University Of Mumbai,
Mumbai,India,Mob-7738457195

Abstract**-- In todays day to day life,mobile users are becoming popular clients to consume any type of web resources such as web service. However, there are problems in connecting mobile devices to existing WS. This paper focuses on three of the following challenge : time and speed,bandwidth/latency,limited resources. This paper implements and developes a cross-platform architecture for connecting mobile devices to the WS. The architecture includes a platform independent design of mobile client and a middleware for enhancing the communication between mobile clients and WS. Finally, the middleware can be deployed on Cloud Platforms, like Google App Engine and Amazon EC2, to enhance the scalability and reliability. The experiments evaluate the optimization/adaptation, overhead of the middleware.**

Keywords**— JavaScript Object Notation,Mobile Web Services,Web Services Description Language,Simple Object Access Protocol,Representational State Transfer,Gzip, Cashing,Mashup.**

I. INTRODUCTION

Use of mobile devices is increasing rapidly which will grow day by day as compare to the numbers of PCs (desktop, laptop, netbook).Network providers are working to improve the services to the customer which will be reliable,efficient and easy to use.These network providers are becoming popular clients to consume Web Services (WS).Mobile phones like iPhone, Blackberry, Android have developed applications that consume WS and they are quite user friendly.

However,there are problems in connecting mobile phones to webservice through network provider.Mobile devices have small processor compared to PCs,so the size of the WS messages needs to be reduced to fit the bandwidth of mobile clients and they should be processed in middleware to have optimised result for mobile users.Mobile devices consume various types of WS and they should be easily adaptable,for example, SOAP and RESTful WS.To have optimised result to mobile clients,WS messages should be compressed using Gzip or deflate.To reduce redundancy of data or request WS messages should be cached.So increased number of mobile devices and improved networks drives the need of optimised consumption of WS.In future,cloud computing will provide more efficient results.

II.LITERATURE SURVEY

Web Services :

Web services are application components.Web services communicate using open protocols.They are self-contained and self-describing.Web services[3] can be discovered using UDDI.They can be used by other applications.XML is the basis for Web services.

When all major platforms could access the Web using Web browsers, different platforms could interact. For these platforms to work together, Web-applications were developed.

Web-applications are simply applications that run on the web. These are built around the Web browser standards and can be used by any browser on any platform.

By using Web services, your application can publish its function or message to the rest of the world.

Web services use XML to code and to decode data, and SOAP to transport it (using open protocols).

With Web services, your accounting department's Win 2k server's billing system can connect with your IT supplier's UNIX server.

Web services extend the World Wide Web infrastructure to provide the means for software to connect to other software applications[4]. Applications access Web services via ubiquitous Web protocols and data formats such as HTTP, XML, and SOAP, with no need to worry about how each Web service is implemented. Web services combine the best aspects of component-based development and the Web, and are a cornerstone of the Microsoft .NET programming model.

Mobile Web Service :

For mobile devices[5], Web services is one of the best ways to use the powerful computing capabilities of workstations.

Mobile Web services is mainly designed so embedded devices can consume the service provided by the server; in other words, mobile Web services are designed from the perspective of the Web services consumer, to enable lightweight devices to share the computing capability and database with the server.

Mobile Web services seamlessly integrates two different applications running on different platforms and provides interoperability between them. In general, there are three kinds of integration techniques that can be applied when concerned with the participation of mobile devices:

1] Socket communication

2] Web services

3] Messaging techniques

Middleware For Mobile Devices :

A mobile device is a pocket-sized computing device[6], typically comprising a small visual display screen for user output and a miniature keyboard or touch screen for user input”.

There is an increase in the demand to deliver content to mobile devices, however these devices and the systems which manage content have limitations as to how they can interact,display and deliver new and existing information.

In addition to the device limitations, information has been created in formats that are not compatible with the growing number of devices in use. These formats must be converted in order to be effective on mobile devices such as XML to JSON conversion.

To have information in less memory size we need to compress it using Gzip/Deflate compression in middleware,which will save processing time of mobile, as mobile devices have small processors.

One of the most important factors in building high-performance, scalable Web applications is the ability to store items, whether data objects, pages, or parts of a page, in memory the initial time they are requested. You can store these items on the Web server or other software in the request stream, such as the proxy server or browser. This allows you to avoid recreating information that satisfied a previous request, particularly information that demands significant processor time or other resources. Known as caching, it allows you to use a number of techniques to store page output or application data across HTTP requests and reuse it. Thus, the server does not have to recreate information, saving time and resources.

ASP.NET provides two types of caching[7] that you can use to create high-performance Web applications. The first is called output caching, which allows you to store dynamic page and user control responses on any HTTP 1.1 cache-capable device in the output stream, from the originating server to the requesting browser. On subsequent requests, the page or user control code is not executed; the cached output is used to satisfy the request. The second type of caching is traditional application data caching, which you can use to programmatically store arbitrary objects, such as data sets, to server memory so that your application can save the time and resources it takes to recreate them.

Mashups let you take part in the reuse software culture. Instead of coding up some functionality that already exists, client mashups let you embed that functionality into your web pages[8]. Many mashups require little integration, especially the presentation mashups where hundreds of widgets are readily available. The Ajax Toolkit supplied by Salesforce makes creating client service mashups a walk in the park, and the web services API enables external service mashups too.

Mashups provide immediate benefit at little cost. The number of APIs that can be used in a mashup grows daily—providing a rich ecosystem of data and services that can be used to extend your Force.com applications. This all means a faster time to market with exciting new functionality that helps your workforce accomplish their daily tasks.

As an example consider the ubiquitous Google Maps mashup. You could take Google Maps functionality and data (the ability to show and navigate maps) and mash it up with data about your accounts to map account locations, providing popup detail bubbles as you mouse over their locations. In effect you would be mashing up the location data of your client with the mapping data and services provided by Google Maps, using Google Maps to display the result on your own site.

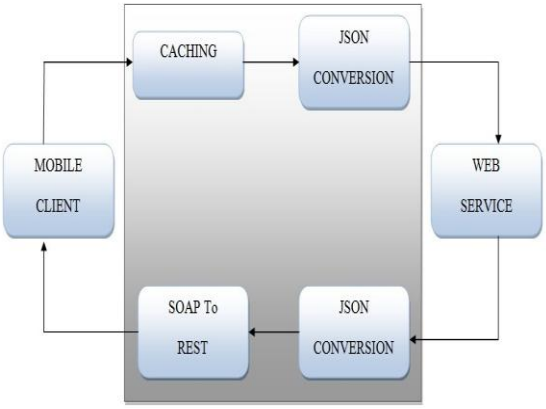


Fig.1 : System Architecture

Cloud Computing :

Cloud computing is a technology that uses the internet and central remote servers to maintain data and applications. Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access[9]. This technology allows for much more efficient computing by centralizing storage, memory, processing and bandwidth.

A simple example of cloud computing is Yahoo email, Gmail, or Hotmail etc. You dont need a software or a server to use them. All a consumer would need is just an internet connection and you can start sending emails. The server and email management software is all on the cloud (internet) and is totally managed by the cloud service provider Yahoo , Google etc. The consumer gets to use the software alone and enjoy the benefits.

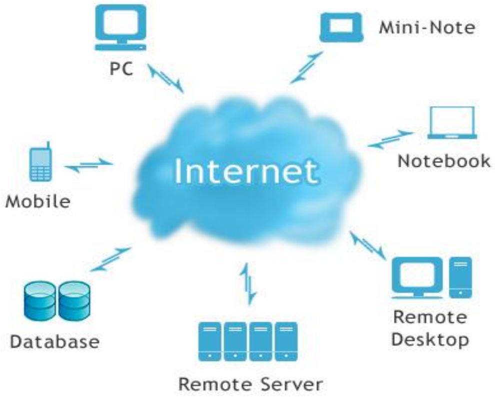


Fig.3 : Cloud Computing

Cloud is made feasible through the deployment and interoperability of three platform types[10]. These three layers are:

1]IaaS - Infrastructure as a Service :

Cloud Providers offering Infrastructure as a Service tout data-center space, and servers; as well as network equipment such as routers/switches and software for businesses. These data-centers are fully outsourced, you need not lift a finger, upgrade an IOS or re-route data. Although this is the base layer, it allows for scalability and reliability; as well as better security than an organization may have in a local co-lo or local datacenter.

2]PaaS - Platform as a Service :

Provisioning a full hardware architecture and software framework to allow applications to run is the essence of Platform-as-a-Service. There's a huge market for customers who require flexible, robust web-based applications. But, in order for these applications to run, there needs to be platform supporting it that is just as robust and flexible. Cloud providers offer this environment and framework as a service.

3]SaaS - Software as a Service :

Software-as-a-Service (which I'll refer to simply as SaaS) is the process of provisioning commercially available software but giving access over the net. The customer doesn't have to worry about software licenses, since they are handled by the service provider. The provider also handles upgrades, patches, or bug fixes. Some examples of this software might be office productivity software, which you may access online, like Google Docs.

III. Aim

An efficient and scalable architecture for connecting mobile devices to the WS.

IV. Proposed methodology

We are developing a cross-platform architecture for connecting mobile devices to the WS. The architecture includes a platform independent design of mobile service client and a middleware for enhancing the interaction between mobile clients and WS.

When WS are executed through the middleware, the follow steps are involved in the middleware.

The mobile client sends a HTTP GET request with an identifier of a WS to the middleware.

The middleware deals with interactions to the WS (and generates SOAP WS client if necessary).

The middleware extracts (JSON or XML parsing) the required service results from the original service result and form a new service results in JSON format.

The middleware is responsible for consuming the Cloud Services whether they are SOAP or RESTful WS and delivers the service result to the mobile client. On the mobile client, users can define WS and later execute the pre-defined WS on the fly. The middleware provide RESTful WS interface for the mobile clients.

JSON :

JSON or JavaScript Object Notation, is a lightweight text-based open standard designed for human-readable data interchange. It is derived from the JavaScript scripting language for representing simple data structures and associative arrays, called objects. Despite its relationship to JavaScript, it is language-independent, with parsers available for many languages.

JSON format is often used for serializing and transmitting structured data over a network connection. It is used primarily to transmit data between a server and web application, serving as an alternative to XML.

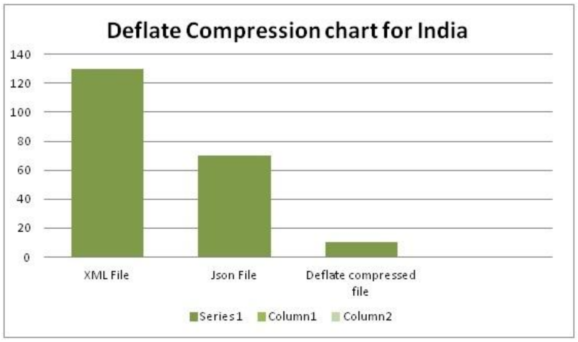


Fig.3 : JSON vs XML

REST :

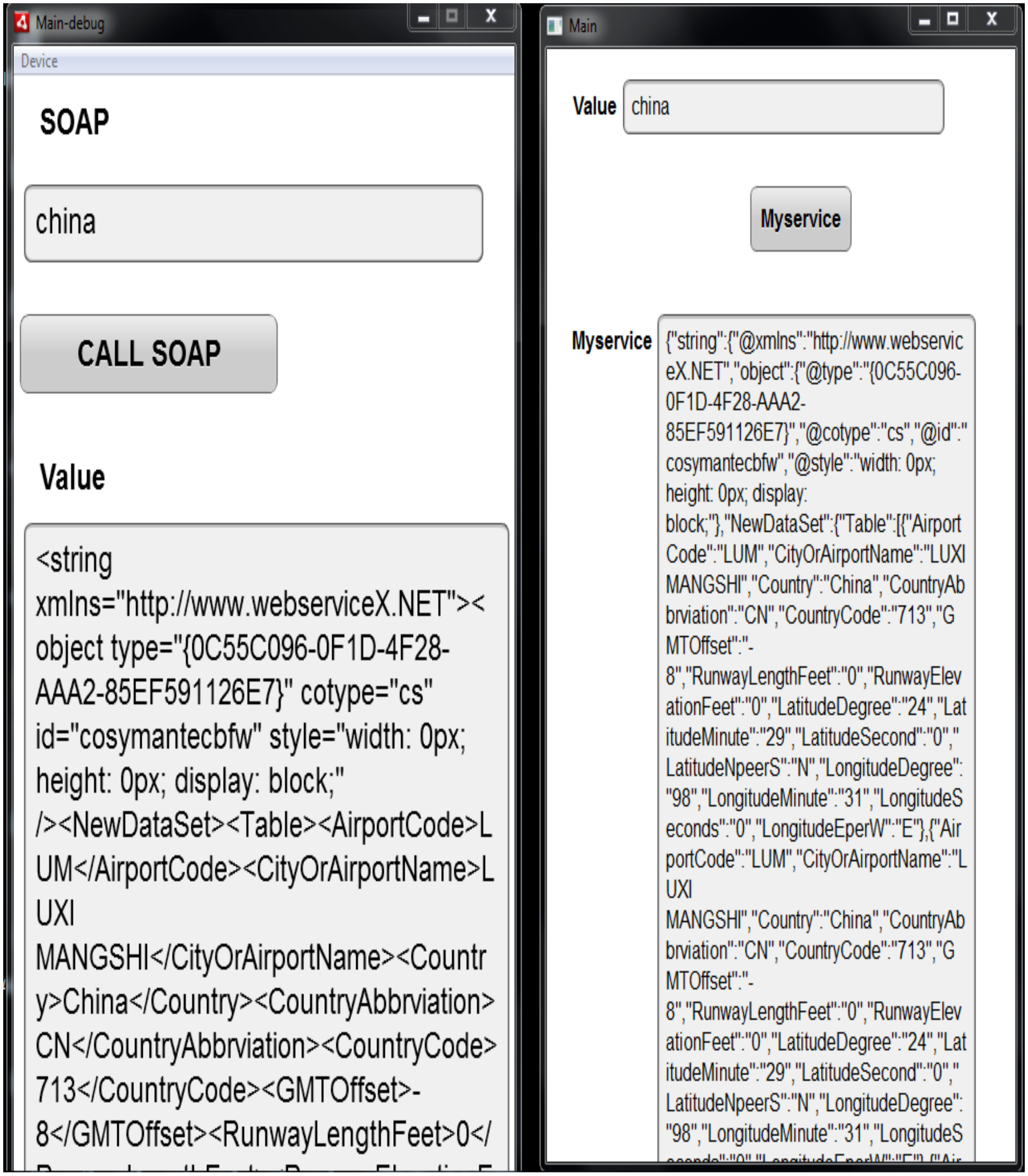
Representational state transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. The term representational state transfer was introduced and defined in 2000 by Roy Fielding in his doctoral dissertation.

SOAP :

SOAP, originally defined as Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks. It relies on Extensible Markup Language (XML) for its message format, and usually relies on other Application Layer protocols, most notably Hypertext Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission.

REST vs SOAP :

- REST has no WSDL interface definition .
- REST is over HTTP, but SOAP can be over any transport protocols such HTTP, FTP, STMP, JMS etc.
- SOAP is using soap envelope, but REST is just XML.



```
<string
xmlns="http://www.webserviceX.NET"><
object type="{0C55C096-0F1D-4F28-
AAA2-85EF591126E7}" cotype="cs"
id="cosymantecbfw" style="width: 0px;
height: 0px; display: block;"
/><NewDataSet><Table><AirportCode>L
UM</AirportCode><CityOrAirportName>L
UXI
MANGSHI</CityOrAirportName><Countr
y>China</Country><CountryAbbrviation>
CN</CountryAbbrviation><CountryCode>
713</CountryCode><GMTOffset>-
8</GMTOffset><RunwayLengthFeet>0</
```

```
{"string":{"@xmlns":"http://www.webservic
eX.NET","object":{"@type":"{0C55C096-
0F1D-4F28-AAA2-
85EF591126E7}","@cotype":"cs","@id":"
cosymantecbfw","@style":"width: 0px;
height: 0px; display:
block;"},"NewDataSet":{"Table":[{"Airport
Code":"LUM","CityOrAirportName":"LUXI
MANGSHI","Country":"China","CountryAb
brviation":"CN","CountryCode":"713","G
MTOffset":"-
8","RunwayLengthFeet":"0","RunwayElev
ationFeet":"0","LatitudeDegree":"24","Lat
itudeMinute":"29","LatitudeSecond":"0","
LatitudeNpeerS":"N","LongitudeDegree":
"98","LongitudeMinute":"31","LongitudeS
econds":"0","LongitudeEperW":"E"},{"Air
portCode":"LUM","CityOrAirportName":"L
UXI
MANGSHI","Country":"China","CountryAb
brviation":"CN","CountryCode":"713","G
MTOffset":"-
8","RunwayLengthFeet":"0","RunwayElev
ationFeet":"0","LatitudeDegree":"24","Lat
itudeMinute":"29","LatitudeSecond":"0","
LatitudeNpeerS":"N","LongitudeDegree":
"98","LongitudeMinute":"31","LongitudeS
econds":"0","LongitudeEperW":"E"},{"Air
```

Fig.4 : SOAP IN XML TO REST IN JSON CONVERSION

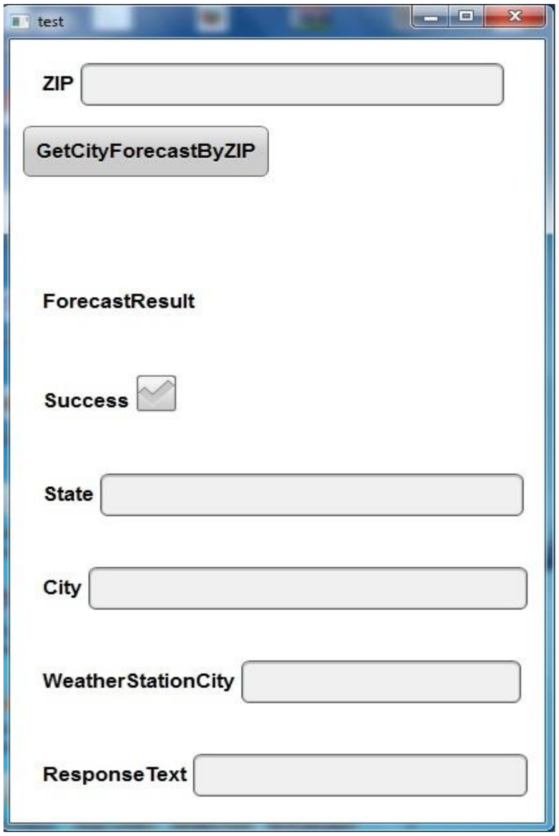


Fig.5 : User Interface

V. CONCLUSION

- It is possible for mobile clients to consume WS .
- Adaptation is needed for mobile clients to interact with WS .
- Middleware can provide a platform for consuming multiple services and returning a favorable highly optimized result.The above process is called a mashup.
- Middleware caching helps improve the system performance.
- It is possible to efficiently transform SOAP Web service protocol to REST web service protocol.
- Middleware services is completely platform independant hence can be consumed by any mobile(IOS,Android,BlackBerry OS) or desktop(Adobe Air) application.

REFERENCES

[1] Portio Research Mobile Factbook, Portio Research, 2009.
[2] S. Yates, It's Time To Focus On Emerging Markets For Future Growth, Forester, 2007.
[3]"Web Service Introduction"http://www.w3schools.com/webservices/ws_intro.asp
[4]"Web Service"-http://msdn.microsoft.com/en-us/library/ms950421.aspx
[5]"Mobile Web Service"-http://www.ibm.com/developerworks/wireless/library/wi-websvc/
[6]"Middleware For Mobile Devices"-Wikipedia.
[7]"Caching In ASP.NET"-http://msdn.microsoft.com/en-us/library/xsbfdd8c%28v=vs.71%29.aspx
[8]"Mashup"-http://wiki.developerforce.com/page/Mashups:_The_What_and_Why
[9]"Cloud Computing"-http://www.wikinvest.com/concept/Cloud_Computing
[10]"Categories Of Cloud"-http://www.securityweek.com/three-categories-cloud-computing-whats-your-flavor.
[11] Code Project -http://www.codeproject.com/
[12]"Sample Web Services"-http://webservicex.net/ws/default.aspx
[13]http://www.youtube.com/?tab=w1&gl=IN