Role of Middleware and Mobile Web Services in Cloud Computing

Anil Dudhe[1]

[1] Sr. Lecturer, ASCT Department, IGNOU-I^{2}IT Centre of Excellence for Advanced Education and Research, Pune

ABSTRACT

The large groups of mobile users have made possible the awareness of intelligent environments for human and machines to interact with each other, interact with information processing devices, and avail mobile services through various types of networks and systems ubiquitously. This has opened the doors for business and coalition. These days all businesses are moving on the cloud. So cloud interoperability is essential for mobile devices in order to nurture the decoupling of the handset to a specific cloud vendor. However, developer need to implement different Web API's with respect to cloud vendor within a native mobile platform. To counter this problem of interoperability across multiple clouds to perform processing intensive services from mobile phones and to provide the platform independency feature for the mobile cloud applications, this paper proposed a Mobile Cloud Middleware (MCM) which provides an exclusive interface that responds in JSON format according to the cloud services requested through RESTful Web Services.

Key words: Cloud, Web Services, Restful, Middleware.

Introduction

The Cloud Security Alliance (2009) declares that the "cloud describes the use of a collection of services, applications, information, and infrastructure comprised of pools of compute, network, information, and storage resources. These components can be rapidly orchestrated, provisioned, implemented and decommissioned, or scaled up and down" [1]

Web services provides standards abstraction layer over dissimilar network transports, operating systems, and programming languages. Web Services achieves its goal in a technology neutral manner. It provides well-defined interfaces for distributed functionalities, which are independent of the hardware platform, the operating system, and the programming language. So distributed functionalities, or services, which may be running on different hardware platforms, may be running in different operating systems, or may be written in different programming languages, can communicate through Web Service interfaces [2-3].

The middleware is software unit plays role in between software components or applications and user enabling services to interact over a network. Cloud middleware is an abstraction layer that hides system complexity and enables seamless communication mechanism between the cloud computing components [4, 5, and 6].

Middleware is distributed software that allows the applications to interact with the underlying operating systems, network protocol stacks, and hardware. Its primary job is to bridge the gap between application programs and the lower-level hardware and software infrastructure in order to coordinate how application components are connected and how they interoperate [7].

The rest of the paper is organized as follows: in section 2 describes related work, section 3 describes methodology adapted, section 4 present the implementation part, and finally, section 5 concludes the paper status.

Related Work

Web services performance is tested [2, 3] and found that the message size of RESTful web service is smaller than messages of Conventional SOAP or XML web service. That's prove REST provides better solution for the majority of implementations with greater flexibility and lower overhead to the constrained devices.

In paper [4] author has proposed an architecture that allows the consumption of Web Services on mobile devices. Hosting Web Services on mobile devices gains in importance when it comes to deliver real-time contextual data, such as current location or real-time heart rate. In addition to the characteristics of the available network, the usability of the Mobile Host depends on computational resources of the device itself.

JSON works more rapidly and uses very few resources than XML shown in result [5]. JSON and XML have its unique strength, but the implication of performance and resource utilization must be implied during decisions making between data interchange formats.

Middleware are based on XML, SOAP, Web services, and service-oriented architecture support that RESTful [8]

Study of different middleware [9] is presented to evaluate different a set of functionalities, metrics and related weights.

Interoperability is achieved for mobile web services by middleware [10] to access enterprise application or data stores. In this paper author has suggested various solutions for deploying mobile middleware systems as a client on the mobile device.

JSON is superior to XML [11]. According to author, to analyze XML one has to consider child node and parent node while JSON appears to be very simple and more readable in format. JSON can be represented in brief with less transmission amount. In interchange of data JSON format is poor in some information that does not prove to exert big impacts truthfully.

Method

The middleware performs all the necessary adaptation to the mobile clients as Cloud services are managed by service provider. Real-time services like, latest News, Blog, Twitter service, Face book etc will be updated to mobile clients via HTTP or email immediately after receiving.

Here developing an independent architecture for connecting mobile devices to the existing and new Web Services which has following parts:

1 Mobile users,

1. Middleware and

2. Cloud services.

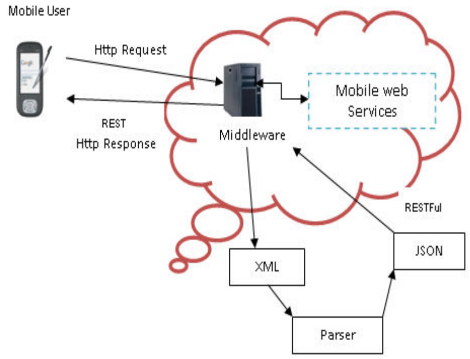

Fig 1: Request handling process

Figure 1: Handling Process

When Mobile user request for Web Services the middleware does following steps:

i. The middleware read existing Web Services designed using SOAP and XML.

ii. Then parse or convert it into JSON format and pass it for further process.

iii. The middleware stores a copy of result with the service ID in the database and returns the optimized result to the mobile user.

The middleware provide RESTful Web Service interface for the mobile clients.

Implementation

XML basics: XML is an industry standard for defining mark-ups. It used to describe data through different user defined tags.

```
<?xml version="1.0" encoding="UTF-8"?>
<Books>
   <book id="1">
```

```
        <title>JavaScript: The Definitive Guide, 5th Edition </title>
    <author>David Flanagan</author>
    <publisher>
          <name> O'Reilly </name>
          <date> August 2006 </date>
       <pages>1018</pages>
    </publisher>
  </book>
</Books>
```

Figure 2: XML data (file name: Book.xml)

JSON

JSON is a text-based and language independent data interchange format. JSON is built on two structures: Objects, enclosed in curly braces {} and Arrays encloses within square brackets []. Its syntax is human readable. Output of figure 4:

```
{
  "Books": {
    "book" : {
      "@attributes": {
        "id" : "1"
      },
      "title" : " JavaScript: The Definitive Guide, 5th Edition ",
      "author" : " David Flanagan ",
      "publisher" : {
        "name" : " O'Reilly ",
        "date" : " August 2006 ",
        "page" : "1018",
        }
    }
  }
}
```

Figure 3: JSON data

Data Exchange from XML to JSON

The middleware sends XML data to the browser, which is processed by parser like DOM or SAX APIs [12]. These parsers convert it into JSON format which is very light and consider as JavaScript native data structure in the browser-side code. So executes very fast in browser compare to XML.

JSON-formatted output received from the server can be captured using following statements:

```
InputStream is = ConvertXMLtoJSON.class.getResourceAsStream("Book.xml");
String xml = IOUtils.toString(is);
 XMLSerializer xmlSerializer = new XMLSerializer();
JSON output = xmlSerializer.read( xml );
System.out.println( output.toString(2) );
```

Figure 4: XML-JSON conversion code

Conclusion

JSON is catching the Web development market and it can valuable alternative to XML in some state. This paper discusses concepts about XML-JSON conversion at the middleware server layer. Detailed implementation, result and performance will be discussed in the next paper.

References

Amrani C, Filali K, Ahmed K, Tidiane A, Telolahy S and El-Ghazawi T., (2012),A Compartive Study of Cloud Computing Middleware., 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing ,690-693.

Cloud Security Alliance (CSA),(2009), Security guidance for critical areas of focus in cloud computing v2.1. Retrieved from Cloud Security Alliance website: http://www.cloudsecurityalliance.org/guidance/csaguide.v2.1.pdf.

Hamad H , Saad M, and Abed R, (2010), Performance Evaluation of RESTful Web Services for Mobile Devices, *International Arab Journal of e-Technology*, 1(3).

J. Gallard, A. Lebre, C. Morina, T. Naughtonb, S. L. Scott and G. Vallee, (2012), Architecture for The next generation system management tools, *Future Generation Computer Systems*, 28 (1),136-146 .

Kosteln P, Sarnovsk´y M and Furdik K, (2011), The Semantic Middleware For Networked Embedded Systems applied In the Internet of Things and Services Domain, Scalable Computing: Practice and Experience, Scalable Computing: Practice and Experience, . 12(3), 307–315.

Mizouni, R., Serhani, M.A., Dssouli, R., Benharref, A. and Taleb, I. ,(2011),Performance Evaluation of Mobile Web Services,Ninth IEEE European Conference on Web Services (ECOWS), DOI: 10.1109/ECOWS.2011.12. Page 184 – 191.

N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta, (2009), Comparison of JSON and XML Data Interchange Formats: A Case Study", in Proc. CAINE, 157-162.

Pawar C and Wagh R,(2012),A review of resource allocation policies in cloud computing, *World Journal of Science and Technology*, 2, 3 .

Schantz R and Schmidt D, (2001), Middleware for Distributed Systems: Evolving the Common Structure for Network-centric Applications,Encyclo-pedia of Software Eng., Wiley & Sons, New York, http://www.cs.wustl.edu/~schmidt/PDF/middleware chapter.pdf.

T. Mather, S. Kumaraswamy, S. Latif, (2009),Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance (Theory in Practice), O'Reilly Media.

Yahui Y.,(2012),Impact Data-exchange based on XML, 7th International Conference on Computer Science & Education doi: 10.1109/ICCSE.2012.6295268, 1147 – 1149.