*IOSR Journal of Computer Engineering (IOSR-JCE)*
*e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 15, Issue 4 (Nov. - Dec. 2013), PP 53-56*
*www.iosrjournals.org*

# Survey on Restful Web Services Using Open Authorization (Oauth)

K. V. Kanmani, P. S. Smitha
*PG Student Velammal Engineering College, Chennai-66.*
*Assistant ProfessorVelammal Engineering College, Chennai-66.*

***Abstract:*** *Web services are application based programming interfaces (API) or web APIs that are accessed through Hypertext Transfer Protocol (HTTP) to execute on a remote system hosting the requested services. A RESTFUL web service is a budding technology, and a light weight approach that do not restrict the client-server communication. The open authorization (OAuth) 2.0 protocol enables the users to grant third-party application access to their web resources without sharing their login credential data. The Authorization Server includes authorization information with the Access Token and signs the Access Token. An access token can be reused until it expires. An authentication filter is used for business services. This paper presents a secure communication at the message level with minimum overhead and provides a fine grained authenticity using the Jersey framework.*
***Keywords:*** *Open authorization (oauth), Restful web services, HTTP protocols and uniform resource identifier(URI).*

## I. Introduction

A web service is a methodology to communicate between two electronic devices over World Wide Web. The Web service is a virtual component that can be accessed through multiple formats and protocols [10]. Web service techniques are loosely coupled, distributed and heterogeneous software systems. Basically, there are two types of Web service enabling technologies: SOAP (Simple Object Access Protocol) based and REST (Representational State Transfer) styled [1]. The original HTTP (Hyper Text Transfer protocol) and HTML (Hyper Text Markup language) protocols are confirmed to be a cost-effective to the user interfaces [11]. A key factor of HTTP and HTML is the simplicity i.e. both HTTP and HTML are primarily text-based and can be implemented using a variety of operating systems and programming environment.

The term Representational State Transfer (REST) was introduced by Roy Fielding to identify an architectural style based on principles, addressability, uniform interface, and statelessness[1].In this section the Rest based services is a light weighted approach. REST does not restrict client-server communication to a particular protocol, but it is the most commonly used with HTTP because HTTP is the primary transfer protocol. The building blocks of the Web are called resources [3].Resources are manipulated through messages that have standard meanings on the Web called as HTTP methods. Resources are named with uniform resource identifier (URIs) [2].

The advantage of using HTML-based user interfaces (UIs) [7] is that they work well across devices with a capable Web browser. OAuth is used for implementing mash up applications that involves services from different service providers. Most of the new public web services from large vendors (Google, Yahoo, Amazon, and Microsoft) rely on REST as the technology for sharing and merging information from multiple sources.

OAuth 1.0 has various ways to make it into the project domain with the lack of performance optimization offered by the protocol. Companies like Microsoft, Google, and other large organizations projected OAuth WRAP (Web Resource Authorization Profiles) which is used to solve the performance and made possible by the enterprise to adopt easily [8]. OAuth is verified by the WRAP recommendation into OAuth 2.0.

The OAuth Web Resource Authorization Profiles (OAuth WRAP) permits the Resource to communicate the authorization and access the Resource to more trusted authorities. The Clients access a Resource that obtains an authorization from the trusted authority (Authorization Server). Once the authorization is provided with an Access Token to the client, a Refresh Token is obtained by a new Access Tokens.

The Authorization Server includes information about the authorization in the Access Token and signs the access Token. The Protected Resource checks whether the Access Token received was from a Client and issued by an Authorization Server and checks whether it is a valid one. The Protected Resource checks the contents of the Access Token to determine the authorization that was granted to the Client was an authorized one.

**The rest of the sections is organized as follows:**

Section II explains about Open authorization protocol. Section III explains about Rest principles. Section IV explains about the architectural models. Section V explains about
Techniques used for restful web services.

## II. Openauthorization(Oauth) Protocol

Securing Restful Web services involves securing the data, as well as the entire communication to protect the confidentiality and integrity of the data. The communication verifies the authentication and access control, to ensure whether the privacy is maintained. The security behind the web service is OAuth 2.0 protocol [6], which is adopted by major service providers. The OAuth 2.0 protocol enables users to grant third-party application access to the web resources without sharing their login credential data.

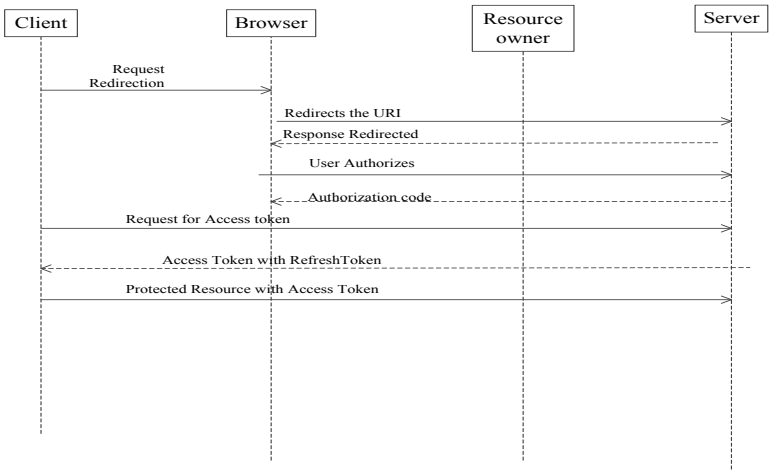


**FIG 1. Oauth protocol architecture flow diagram**

In fig 1. the client redirects the request to the browsers. The resource owner identifies a uniform resource identifier (URI) and redirects to the authorization server (AS). The authorization server authenticates from the resource owner to check whether the owner denies or accept the client request. An authentication code (AC) is generated from the AS and redirect to client. The client sends back the request with AC and URI to the AS to verify the code. An Access token is created and sent to the client.

The purpose of the token is to make it redundant for the resource owner to share its credentials with the client. The server receives the protected resources with the access token from the client. Rather than relying on a single password as the master key for every app that accesses an API, OAuth uses this token. The OAuth protocol enables a website or application (known as a service consumer) to access the protected resources from a web service (known as a service provider) through an API. With OAuth, the browser redirects the resource individually to verify authentication. Therefore, using verifiable and cacheable assertions reduces network transparency for clients [12]. Another emerging protocol is XAuth, an open platform for extending authenticated user services across the Web which has lot of security problems.

## III. Rest Principles

This paper proposes a secure communication at the message level with minimum overhead and also provides a fine authenticity, and confidentiality [5]. REST protocol helps to maintain the scalability of a server, for a very large number of clients. The advantage of including cache constraints is to improve the efficiency, scalability, and performance and reduce the latency of interactions. The methods used are GET, POST, PUT and DELETE operations[3] which are based on the http method used to create, retrieve, update, and delete operations on resources, respectively. The restful approach significantly reduces the transparencies that are caused by the required processing of SOAP-based messages due to the open and uniform identifying scheme.

**GET** - Used to Retrieve Data
**POST** - To Append Data into the server

**PUT** - Used for Inserts and Updates
**DELETE** - Used to delete data

In this paper we provide a security protocol to make message security implementation as lightweight [7] and efficient to respect the REST principles and how the resources are manipulated through a message signature and address communication security for Restful services at a fine grained level. REST interaction was a two-way process with large-grain data of hypermedia interaction can be processed in a data-flow network, and the filter components are applied to the data stream in order to transform the content [9].In REST, components are actively used to transform the content of messages because messages are self-descriptive and are visible to intermediates [14].

## IV. Architectural Model:

Fig 2. Explains how an Authorization Server gets the request from the User. The Client starts the authorization flow and obtains an agreement from the Authorization Server on behalf of the User's. At this point, if it is successful, the Authorization Server issues an authorization code (one-time token). Client exchanges the authorization code for an access token. The clients request the POST method which is designed to request the server to accept the data which are enclosed in the request message's body for storage. A feedback is given to the client from the resource server. The client sends the result to the user. One of the security threats in OAuth2 is a malevolent Client stealing the tokens asking for an arbitrary transmit, so that Authorization Servers protect the token against this by requiring Clients to register one or more redirect URIs.

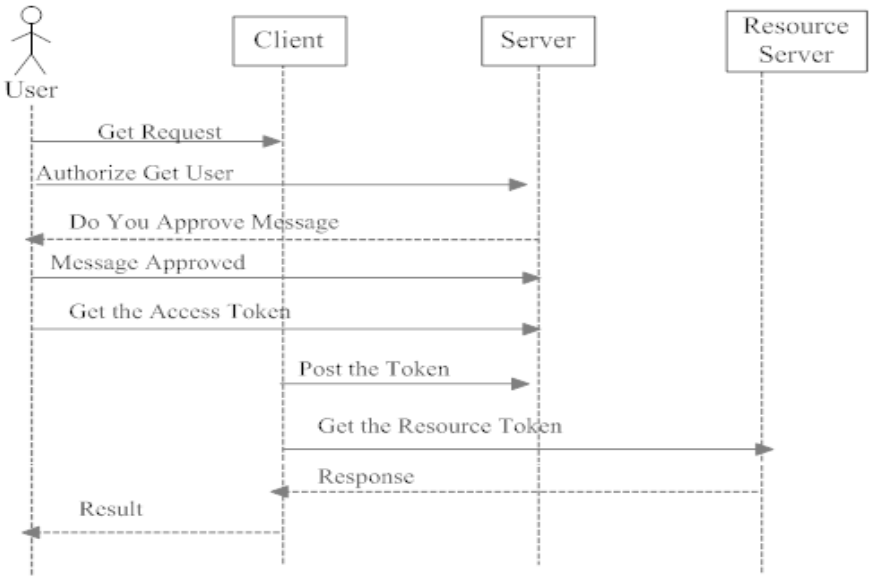


**Fig 2: Architectural flow diagram**

The Authorization Server is called due to the interface that provides users to confirm that they authorized to the Client to act on behalf of the server. The UAA (User agent authentication), and Oauth2 is used to, provide a simple form-based interface in the general case, but allow auto-approval of certain clients.

## V. Techniques Used For Restful Web Services:

The techniques used in Restful web services are

**1)Restlet** : Restlet is a lightweight, open source framework for the Java platform. Restlet is appropriate for both server and client Web applications. It supports major Internet transport, data format, and service description standards like HTTP and HTTPS, SMTP (simple mail transfer protocol).

**2)RESTEasy** : RESTEasy provides frameworks to build Restful Web Services and Restful Java applications. It is certified fully for the JAX-RS specification. JAX-RS is a new JCP (java community process) specification.

**3) Jersey**: Jersey is an open source that builds the production quality reference implementation of JSR-311: JAX-RS- Java API for Restful Web Services. HTTP uses Multipurpose Internet Mail Extensions (MIME)

media types to identify the data formats [2], Some of the common MIME are given in table 1 used by the restful services.

| MIME | Content type |
|---|---|
| Json | Application/json |
| XML | Application/xml |
| XHTML | Application/Xhtml+xml |

**Table 1. Common MIME types used by Restful services**

## VI. Conclusion:

Representational State Transfer is significantly grown using various techniques of a software architectural style for handling web-based integration, which can be used to contribute web services using data interchange format as well as OAuth as authorization protocol. This paper presents a literature survey on these various techniques and how each of these techniques has their own benefits and limitations. This paper discusses on how REST protocol is performed using open authorization. Compared to the SOAP-based integration approach, REST has many advantages such as service is addressable and can be connected to, interface is consistent, and resources can be cached. Moreover, Restful Web services are the Web service that have a simple description of the document and is easy to release, and provides a platform for the future work on web services.

## References:


[1] Serme, G.,de Oliveira, A.S. ; Massiera, J. and Roudier, Y.“ Enabling Message Security for RESTful Services”.In Proceedings Web Services (ICWS), 2012 IEEE 19th International Conference on web services pages 114 – 121.

[2] S. Vinoski, “RESTful Web Services Development Checklist,” Internet Computing, IEEE, vol. 12, no. 6, 2008, pp. 96-95.

[3] Paul Adamczyk, Patrick H. Smith, Ralph E. Johnson, and Munawar Hafiz “REST and Web Services: In Theory and in Practice” Pages (35-57), 2011, springer NewYork.

[4] D. Booth et al., “Web Services Architecture,” W3C Working Group Note, February 2004. http://www.w3.org/TR/ws-arch/.

[5] Belqasmi, F.Glitho, R. and Chunyan Fu “RESTful web services for service provisioning in next-generation networks” Communication Magazine, IEEE (Volume:49 , Issue: 12 on December 2011.

[6] San-Tsai Sun and Konstantin (Kosta) Beznosov “An empirical analysis of OAuth SSO systems”In proceedings of the 2012 ACM conference on computer and communication security, pages (378-390).

[7] Christian Prehofer, Jilles van Gurp, Vlad Stirbu, Sailesh Sathish, Pasi P. Liimatainen, Cristiano di Flora, and Sasu Tarkoma “Practical web-based smart spaces” Pervasive Computing, IEEE (Volume:9 , Issue: 3 on july-sept 2010.

[8] Noureddine, M. and Bashroush, R., A Performance Optimization Model towards OAuth 2.0 Adoption in the Enterprise Cybernetic Intelligent Systems (CIS), 2011 IEEE 10th International Conference on 1-2 Sept. 2011, pp 76-80.

[9] Fielding, R.T. and Taylor, R.N., “Principled design of the modern Web architecture” Software Engineering, 2000. Proceedings of the 2000 International Conference on June 2000, pg 407-416.

[10] Frank Leymann “Web Services: Distributed Applications without Limits” http://citeseerx.ist.psu.edu Conference held on 2003 pg: 26-28.

[11] Cesare Pautasso, Olaf Zimmermann and Frank Leymann “Restful web services vs. "big"' web services: making the right architectural decision” published in WWW '08 Proceedings of the 17th international conference on World Wide Web in 2008 ACM New York, NY, USA 2008 pg 805-814.

[12] http://docs.oracle.com/cd/E14571_01/web.1111/e13734/rest.htm#autoId0

[13] http://javadevelopment.wikispaces.com/file/view/OReilly+RESTful+Web+Services+Cookbook.pdf.


.