Name : - Mahamuni Rutik Sunil
Roll No. :- 17448
Subject :- Web Service

CASE STUDY

SOAP, UDDI AND SEMANTIC WEB

Introduction:-

The growth of the World Wide Web has created a virtual forum that allows rapid exchange of information between parties. However, there is no common manner for transferring application-specific data. The key protocols of the current web infrastructure are HTTP and HTML. HTTP concerns itself with how data should be transported between a server and client. HTML defines the predominate data format that is used to render text on the current infrastructure. The cloud of technology that should enable this level of peer-to-peer interaction is called Web Services. We will examine SOAP, a new initiative that defines a much stricter data format that allows integrity and allows for proper syntactic validation of the message. We will also introduce UDDI, which is a service for discovering available web services using a public directory. Finally, we will look at the Semantic Web. In addition to defining the syntax of these interactions using Web Services technologies, the Semantic Web may also be helpful to define the semantic meanings of these interactions.

SOAP:-

Simple Object Access Protocol (SOAP) Version 1.2 is defined by the W3C as “a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment” SOAP is meant to promote shared understanding of data in a way that machines can easily and correctly parse them. To achieve this goal of extensibility, SOAP uses XML as the principal data format. SOAP consists of several components and actors that work together. A SOAP envelope consists of the data to be transmitted. Each actor is represented by a server node that has a role in processing the message that defines its behavior and responsibilities. In addition, SOAP also has an error structure that allows for graceful handling of faults. Each of these will be further discussed in detail in the following sections.

Advantages of Soap

WS Security: SOAP defines its own security known as WS Security.
Language and Platform independent: SOAP web services can be written in any programming language and executed in any platform.

Disadvantages of Soap

Slow: SOAP uses XML format that must be parsed to be read. It defines many standards that must be followed while developing the SOAP applications. So it is slow and consumes more bandwidth and resource.
WSDL dependent: SOAP uses WSDL and doesn't have any other mechanism to discover the service

SOAP Fault:-

A SOAP fault is generated when an error occurs during the processing of the SOAP message. A fault may be generated by a SOAP intermediary or by a SOAP recipient. A SOAP faults is separate from binding related errors. A binding error is reported using the error mechanisms of the underlying transport protocol. When a SOAP fault occurs, no additional data may be returned. Therefore, it is not possible to return partial data and a SOAP fault in the same message. A SOAP fault must contain a code element which describes the type of error that occurred. Furthermore, it must also contain a reason element that should provide further explanation as to why the fault was generated. Optionally, the SOAP fault may indicate the node and role where the fault originated.

Problems with SOAP

In theory, SOAP creates a very fine line about what it can and cannot do. However, in application, the predominate use of SOAP has corrupted the integrity of its architecture. Most of these problems can be traced to architectural mismatches with the predominate protocol binding - HTTP.
Layering of resources and representations
Idempotent operations

UDDI

The Universal Description Discovery and Integration (UDDI) system defined as “a set of services supporting the description and discovery of businesses, organizations, and other Web services providers, the Web services they make available, and the technical interfaces which may be used to access those services.” The Universal Description

Discovery and Integration (UDDI) system is built as a mechanism on top of SOAP. Its primary goal is to allow multi-organizational collaboration stored within an UDDI registry

What is UDDI Based On?

- UDDI uses World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF) Internet standards such as XML, HTTP, and DNS protocols.
- UDDI uses WSDL to describe interfaces to web services
- additionally, cross platform programming features are addressed by adopting SOAP, known as XML Protocol messaging specifications found at the W3C Web site.

Challenges to UDDI

There are several challenges and issues with UDDI that may affect its acceptance and chances for long-term survival. The first is that UDDI has a non-uniform security model. The second is that UDDI may be limited by its ability to only have single ownership of an entity. Finally, the subscription model may place an undue burden upon the clients of the UDDI system.

- Non-uniform security model
- Duration and history of subscriptions
- Single ownership

Semantic Web

In addition to the umbrella of Web Services, there is also work on creating a Semantic Web. The W3C working group defines the Semantic web as way to "bring to the Web the idea of having data defined and linked in a way that it can be used for more effective discovery, automation, integration, and reuse across various applications Similarly to Web Services, the Semantic Web is looking to enhance the interaction of sites. If Web Services could be viewed as the syntactic agreement of how web sites can interact, then the Semantic Web is an agreement on what is being transferred. Therefore, in addition to agreeing on a common format of how data should be transferred, there is also a contract as to the meaning of the transferred data

Challenges of the Semantic Web

There are two major classifications that challenge the viability of a Semantic Web. The first one is that it is not a small technical feat to create such ontology's. However, the more serious challenge arises from a social perspective. Therefore, it may be possible to address the technical challenges, but it may not be possible to address the social challenges inherent in creating a Semantic Web.

- Ambiguity in Natural Languages
- Multiple Ontology's

- Intentional Non-Participation
- Unintentional Non-Participation

CONCLUSION:-

Two web services that implement the same functionality may still have different interfaces. SOAP merely allows for the separation between display of content from the content itself. Current technologies like HTML do not create this level of separation. Therefore, just removing the element of display from content is a significant win. However, SOAP by itself does not allow for inherent migration from content providers. UDDI is a necessary technology that is required to make the adoption of Web Services widespread. Without a good naming system, Web Services would become unmanageable. Yet, UDDI suffers from scalability issues that would essentially cripple it if it gained widespread acceptance that it seeks. Issues of security, replication, and subscription would overwhelm a widespread UDDI infrastructure. The Semantic Web tries to make semantic mismatch a slightly easier problem to handle. Yet, there is still no consensus on how to define true semantic meaning. For example, there is no RDF format that everyone agrees can fully describe a document. Each individual can come up with their own RDF schema.

CASE STUDY

REVIEW PAPER ON WEB SERVICE SECURITY

INTRODUCTION:-

Forty years ago, computers began to be connected to the Internet and data transfer among computers was already common. Since then, Internet has evolved to form a huge information space, in which users can move transparently from one machine to another. In the field of application programs, a similar development is ongoing. Distributed computing has been used as long as there have been computer networks. But at present, distributed applications are increasingly viewed and constructed as one vast computing medium .Applications which interacts between different machines to provide orchestrated services have now been deployed on a large scale. Introduce security concerns that do not exist in traditional distributed messaging techniques like RMI and CORBA. This is because the SOAP based XML messages can bypass easily traditional firewalls and this could lead to gain access to sensitive systems for non - authorized users just using the interfaces provided by the WSDL files for service description for example. The object of this paper is to explain the principles of web services architecture. It presents the concepts, standards, and the required infrastructure. It discusses and analyzes the limitations of this architecture from the security view giving also light on the challenges surrounding this aspect related to this technology.

DEFINITION OF WEB SERVICE:-

"A web service is any piece of software that makes itself available over the internet and uses a standardized XML messaging system. XML is used to encode all communications to a web service. For example, a client invokes a web service by sending an XML message, and then waits for a corresponding XML response

WEB SERVICES SECURITY:-

To secure Web services, a range of XML-based security mechanisms are needed to solve problems related to authentication, role-based access control, distributed security policy enforcement, message layer security that accommodate the presence of intermediaries. This is a principal condition to make Web services widely adopted, since no company wants to risk exposing their applications and business flows with no damage. Standardization organizations are proposing specifications in order to make these services more secure as traditional Security techniques doesn't provide security against Application level communication as they works on the Lower levels of the OSI stack of message transfer specially on transport layer. The most important standards

are:

1) XML Encryption 2) XML Signature 3) WS-Security 4) WS-Policy 5) WS-Security Policy

WEB SERVICES SECURITY REQUIREMENT:-

There are many security challenges for adopting Web services. The objective is to create an environment, where message level transactions can be conducted securely in an end-to-end fashion during transit and data storage. The requirements for providing end-to-end security for Web services are summarized in

Requirement	Explanation
Authentication	Authentication is needed in order to verify the identities of the requester and provider agents. In some cases, the use of mutual authentication may be needed since the participants may not necessarily be directly connected by a single hop. Several methods can be used to authenticate services (can be combined) including: passwords, certificates, Lightweight Directory Access Protocol (LDAP), Kerberos, and Public Key Infrastructure (PKI)
Authorization	Authorization is needed in order to control access to resources. Once authenticated, authorization mechanisms control the requester access to the requested resources on the system
Data Integrity and Data Confidentiality	Data integrity techniques ensure that information has not been altered, or modified during transmission without detection. Data confidentiality ensures that the data is only accessible by the intended parties. Data encryption and digital signature techniques are used for this purpose. It must be verified in End-to-End manner
Non-Repudiation	It is a security service that protects a party to a transaction against false denial of theoccurrence of that transaction by another party. It used to resolve probable disagreement.
Audit Trails	Audit trails are needed in order to trace user access and behavior. They can ensure system integrity through verification. It is often not possible to prevent the violationof obligations. Instead, if an audit guard detects a policy violation, some form of retribution or remediation must be enacted.
Distributed Security Policies	The architecture must be able to provide a security policy and enforce it across heterogeneous platforms with varying constrains and privileges

THE WEB SERVICE PROTOCOL STACK:-

Web services are built by using various related technologies. Illustrates the stack of standards on whichweb services are generally based on.

- **Service transport :** The service transport layer delivers messages between applications. This layer usually implements hypertext transfer protocol (HTTP), Simple Mail Transfer Protocol (SMTP) or file transfer protocol (FTP).
- **XML messaging :** This layer is responsible for encoding messages in a common XML format so that messages can be understood at both parties. This layer includes XML-RPC and SOAP
- **Simple object access protocol (SOAP) :** SOAP is a simple XML-based messaging protocol responsible for transferring data between different web services. SOAP allows communication among interacting web services by implementing a request/response model

Service description WSDL : The purpose of this layer is to define the public interface of a specific web service and its description. WSDL is based on XML

PERFORMANCES:-

Using WS-Security implies using signing and encryption. Those operations are costly in matter of resources (CPU and Memory) they can cut application throughput between 5 percent and 50 percent. A solution for this overhead could be the use of a dedicated hardware call XML Firewalls. Those hardware systems provide performance as they allow real-time processing of huge documents. But they cannot always be used as an optimal solution as this quality comes with a price and also they cannot be easily integrated with the already existing back-end software infrastructure.

CONCLUSION:-

We have seen that web services constitute a sort of automated services which communicate via Internet and rely on open Internet-based standards. We have seen also that Web services are invoked using messages instead of APIs or file formats. This works due to the independence of the service interface through the WSDL standard from the implementation. Nevertheless, web services are also subject to some limitations which include low performance, weak transaction management facilities but more critical, an immature and incomplete security framework although the presence of the WS Security framework that we presented and give a critical analysis. . It is important de note that Web Service security represents a key requirement for today's distributed interconnected electronic world. To date, the problem of security has been investigated very much in the context of standardization efforts; these efforts, however, been concentrated in adapting existing security techniques, such as encryption, for use in Web Services