Web Services Demystified

Abstract

The distributed systems were aimed at enabling the interactions of computers for computation hungry applications. The distributed systems have been going well when it comes to the integration between computers with in the same domain of authority. But cross-domain cooperation for problem solving has always been a problem. Primary reasons being the proprietary nature of the protocols used and the dependence of the inherent infrastructure. Web Services have been designed to overcome these problems. This article gives a straightforward explanation of the architecture and the features of web services

Article Summary

The invention of the World Wide Web was the greatest revolution in the fields of commerce and industry resulting in an entirely new paradigm of the e-commerce. In the same context the web services, as many veterans of the e-commerce industry are forecasting are going to bring about the same revolution for the e-commerce solution providers. The design of web services seems to be the ideal for overcoming the problems being faced for integration of distributed services. Web Services have enables the maximum exploitation of the services available in distributed systems due to the use of standard protocols and platform independent libraries. Now it is possible to make an application that would be just a merge of different web services hence exploiting the reuse of code to the maximum. The purpose of this article is to give an easy to understand introduction to web services, a technology that enjoys a high position in the future strategies of the big players like Microsoft, IBM and SUN etc. The design issues about the web services technology are also discussed in details with a nutshell touch of the advantages brought about by the web services.

Key Words

Web Services, Service Oriented Architecture, Distributed Systems, XML, SOAP, WSDL, UDDI

Introduction

Rightly said the computer technology is one of the most rapidly growing technologies of the world. Form 1940 to 1980 computers were large in size and cost tens of thousands of dollars. This hampered their wide spread usage. But the development of microprocessors in mid 80s resulted in both their reduced size and reduced cost. And then the invention of high-speed networks further resulted in the low costs of the intense computation. The problems previously

required costly mainframes now could be solved on multiple of computers connected together via appropriate networks. This initiated the era of distributed computing.

The basic aim behind the distributed computing was the elimination of the needs of the costly mainframes by solving the problems via coordination and communication of computers. The technologies developed to harness the power of distributed computing like Microsoft's DCOM (Distributed Component Object Model), Java RMI (Remote Method Invocation) and the CORBA (Common Object Resource Broker Architecture) were able to leverage the computing power of distributed computers only to some an extent. The primary reason behind their poor performance was their dependence on the inherent infrastructure. DCOM was primarily developed for coordination between computers running Microsoft Windows. (Although the DCOM has been ported to various other platforms but it has never gained acceptance on these platforms). RMI enables interaction between programs written in Java. CORBA provided interaction between the programs written in different programming languages but writing a service via CORBA is a rather complex issue. This is because of the differences in the programming languages among which the CORBA may have to translate.

The second problem with all these techniques is their connection-oriented nature especially where the coordinating computers are not located under the same administrative domain. To communicate through these technologies the applications have to communicate through some specified ports resulting in security problem. Their connection-oriented nature also causes problems in their load balancing. Once the connection between service provider and service requester is intact the next request cannot be routed to another service provider. Hence we can conclude that following features are required to truly benefit from distributed architecture namely Interoperability between different platforms, access friendliness, greater reuse of existing techniques and infrastructure, programming language independence.

Keeping in view these requirements ad needs Microsoft introduced a new distributed computing technique in the name of the Web Services.

Web Services

Formally webservices.org defines web services as

"Web Services are encapsulated, loosely coupled contracted functions offered via standard protocols"

- Encapsulated means that functionality of the function is hidden from outside.
- Loosely Coupled means that invoker and invoked entities (functions) are independent from changes in each other.
- Contracted means they are publicly available.
- Standard protocols imply that they are accessible via common in use protocols.

Architecture of Web Services

Since the web services are another technology for distributed computing they follow the same Service Oriented Architecture SOA as being followed by other distributed computing techniques like RMI, DCOM, and CORBA etc.

Service Oriented Architecture

The service-oriented architecture consists of the 3 entities namely the service provider service requester and the service broker. All the techniques based on the service oriented architecture follow the following basic steps as shown in the figure 1

1. The service provider publishes its service descriptions to the service broker.
2. The service requester queries the service broker (lookup service) for the information of the service provider.

3. The service requester binds it to the service provider

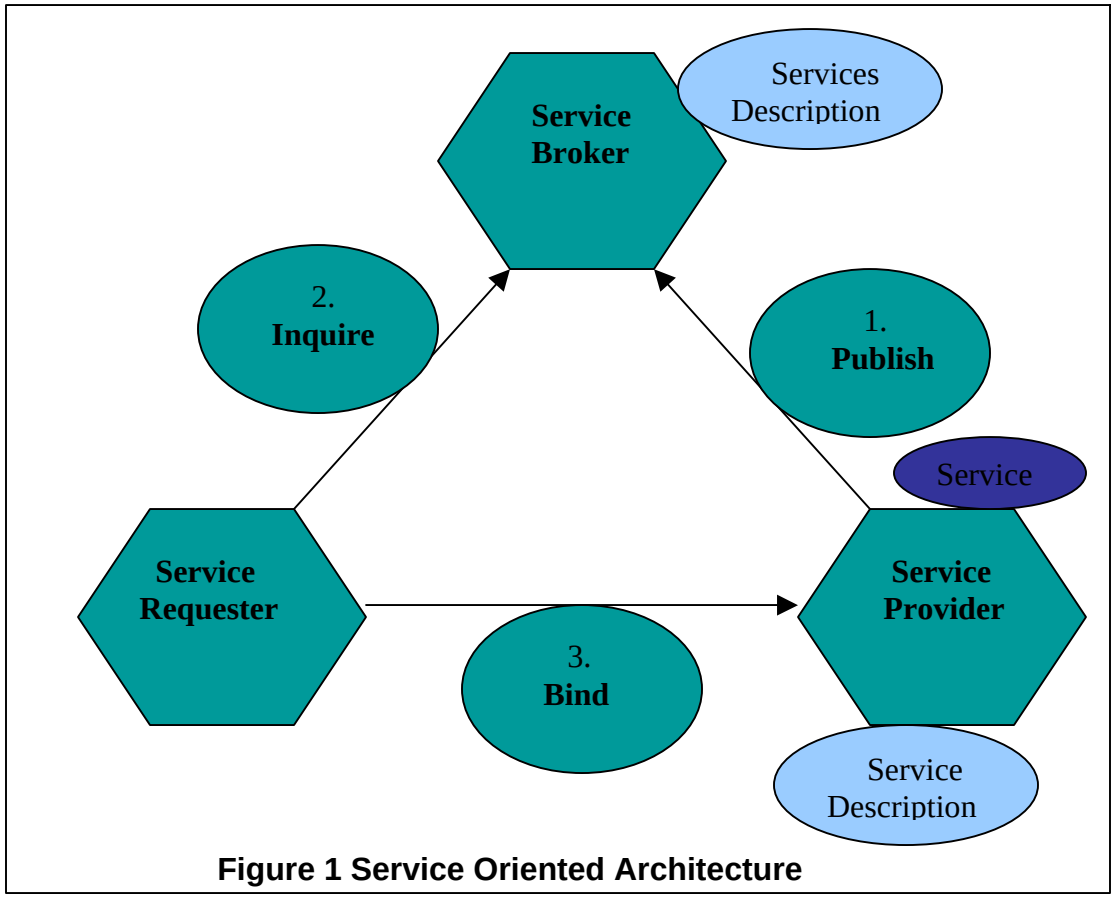

Figure 1 Service Oriented Architecture

Web Services Building Blocks

The building blocks of the web services technology or the web services programming stack is as shown in the figure 2. This stack shows that the major functions required for publishing and locating web services along with the standard protocols associated with them.

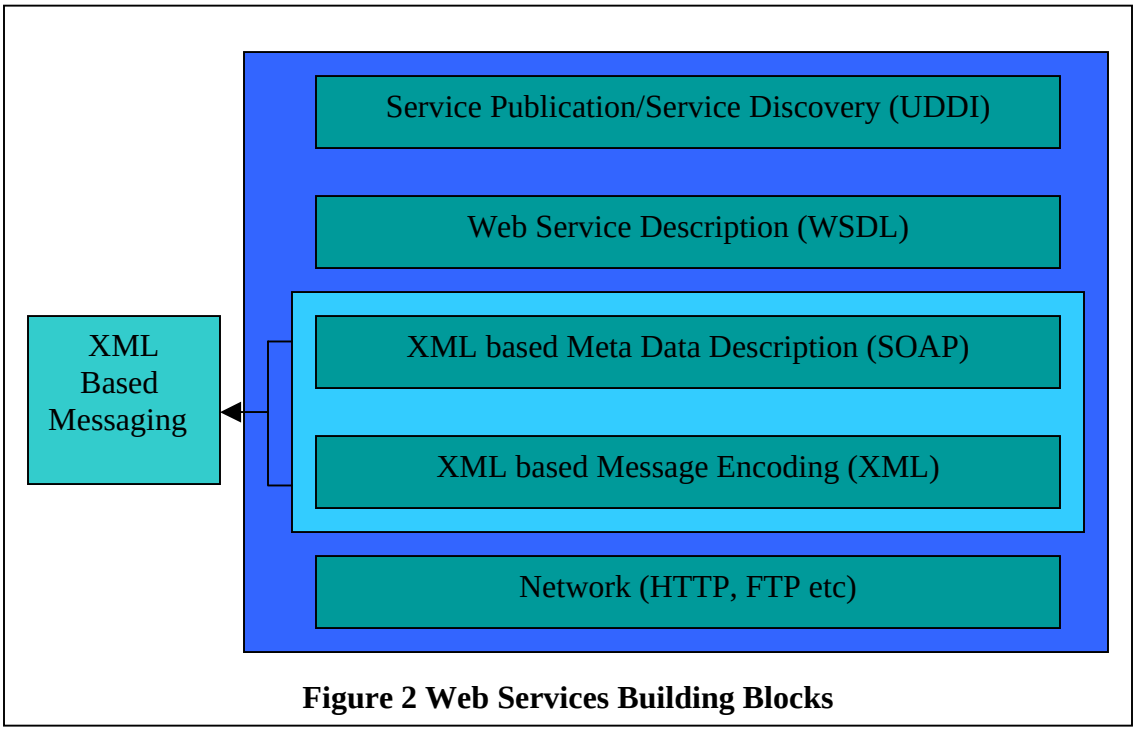

Figure 2 Web Services Building Blocks

Service Publication / Discovery

Web Services developers need to publish their web services so that the web services requester can bind to them while the web services requesters should first locate the web services so that they can bind to it.

Universal Discovery Description and Integration (UDDI) protocol provides both of these functionalities.

Web Services Description

Once the client requesting the web services has located the web service it needs further information like over which protocol the web service expects to receive queries. The Web Service Description Language (WSDL) handles this and many other web service description features.

XML based Messaging

XML based Meta Data Description

Some times additional data may be passed along with the web service encoded message which is important for the proper working of the web service but that is not a part of the message e.g. the web service request may have to pass through many intermediate nodes before reaching the final destination hence it must be predefined whether intermediate destinations should process data or not. The Simple Object Access Protocol handles such factors. SOAP is an XML based protocol the reason being that it had to be platform independent hence XML was the obvious option.

XML based Message Encoding

All the data transferred between the service provider and the service requester should be understandable to both. As the web services are aimed to provide the maximum of the platform independence hence the message encoding should not be infrastructure dependent. Hence XML is the obvious option for encoding the web service messages.

Network

The basic layer of the web services stack is the network layer. The design needs of the web services required the inherent network protocols to be connectionless, widespread and independent of the infrastructure. So obvious option was Hyper Text Transfer Protocol (HTTP). It is wide spread in use over the Internet. Further most enterprise firewalls allow HTTP traffic. And most importantly it is connectionless hence supporting the load balancing.

Benefits of Web Services

The Web Services provide the following advantages

- Inter Operability (does not employ infrastructure dependent protocols and libraries)
- Ubiquitous (employs wide spread protocols of the internet)
- Load Balancing (employs connection less protocols hence the request to one server can be directed to other servers)
- Reuse (encourages product development using existing products).

Hence it can be easily conclude that web services are the most appropriate technique for the expansion of business horizons in this fast developing information age.

"What's the most pervasive element of IT's future that most of us know the least about? My guess would be Web services, that transparent aspect of computing that will become increasingly critical to the smooth operation of just about everything."

Bob Sutor, Director, Web Services Strategy, IBM. [1]

References

www.ebizq.net/topics/web_services/ features/3184.html?pp=1

By

Muhammad Raza ur Rehman
netwizio@yahoo.com
Military College of Signals
NUST

Author Biography

Author is a final year student of computer software engineering in the College of Signals, National University of Sciences and Technology Rawalpindi Pakistan. The interests of author are in distributed systems, artificial intelligence, computer networks, network security, wireless communication etc.

Contact Information

Mailing Address: hno 1684 st no 79-a I-10/1 Islamabad Pakistan
Email address: netwizio@yahoo.com
Telephone 92-051-4440964