SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

AN AUTONOMOUS INSTITUTION

Approved by AICTE, New Delhi and Affiliated to
Anna University, Chennai

V SEMESTER

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

19TS501 - Cloud Computing

(Regulation – 2019)

UNIT – II

Topic: 2.3. Web Services

Academic Year 2024 – 2025 (Odd Semester)

2. 3. Web Services

Web services are the prominent technology for implementing SOA systems and applications.

They leverage Internet technologies and standards for building distributed systems. Several aspects make Web services the technology of choice for SOA.

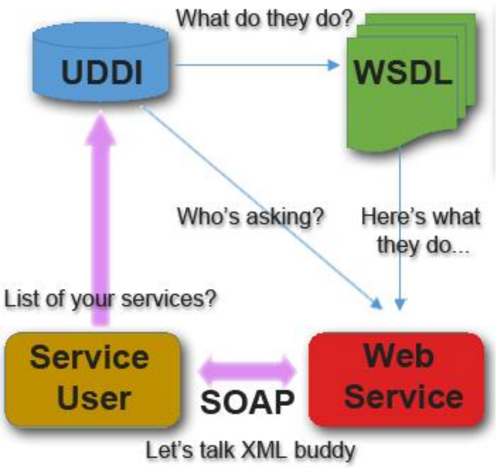
First, they allow for interoperability across different platforms and programming languages.

Second, they are based on well-known and vendor independent standards such as HTTP, SOAP, XML and WSDL.

Third, they provide an intuitive and simple way to connect heterogeneous software systems, enabling the quick composition of services in a distributed environment.

Finally, they provide the features required by enterprise business applications to be used in an industrial environment.

They define facilities for enabling service discovery, which allows the system architect to more efficiently compose SOA applications and service metering to assess whether a specific service complies with the contract between the service provider and the service consumer.



The concept behind a Web service is very simple.

Using as a basis the object oriented abstraction, a Web service exposes a set of operations that can be invoked by leveraging Internet based protocols.

The semantics for invoking Web service methods is expressed through interoperable standards such as XML and WSDL, which also provide a complete framework for expressing simple and complex types in a platform independent manner.

Web services are made accessible by being hosted in a Web server

HTTP is the most popular transport protocol used for interacting with Web services.

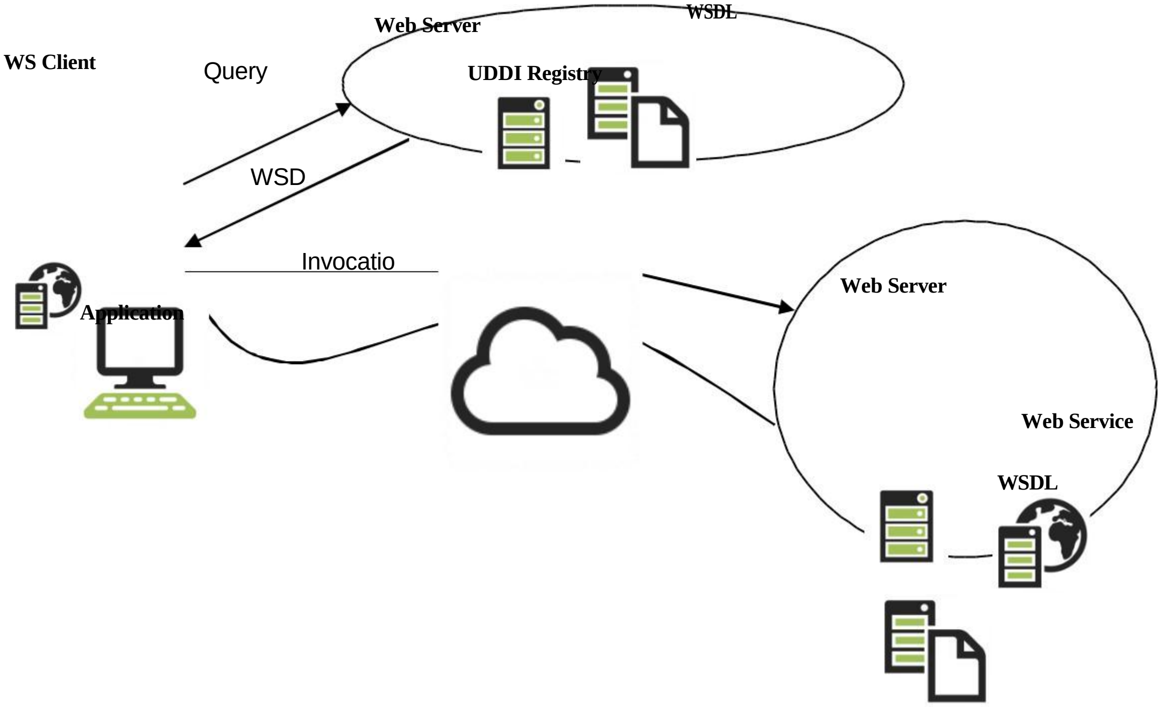


Figure 2.1 Reference scenario for Web Services

Figure 2.1 describes the common use case scenarios for Web services.

System architects develop a Web service with their technology of choice and deploy it in compatible Web or application servers.

The service description document is expressed by means of Web Service Definition Language (WSDL), can be either uploaded to a global registry or attached as a metadata to the service itself.

Service consumers can look up and discover services in global catalogs using Universal Description Discovery and Integration (UDDI).

The Web service description document allows service consumers to automatically generate clients for the given service and embed them in their existing application.

Web services are now extremely popular, so bindings exist for any mainstream programming language in the form of libraries or development support tools.

This makes the use of Web services seamless and straightforward with respect to technologies such as CORBA that require much more integration effort.

Moreover, being interoperable, Web services constitute a better solution for SOA with respect to several distributed object frameworks, such as .NET Remoting, Java RMI, and DCOM/COM1, which limit their applicability to a single platform or environment.

Besides the main function of enabling remote method invocation by using Web based and interoperable standards, Web services encompass several technologies that put together and facilitate the integration of heterogeneous applications and enable service oriented computing.

Figure 2.2 shows the Web service technologies stack that lists all the components of the conceptual framework describing and enabling the Web services abstraction.

These technologies cover all the aspects that allow Web services to operate in a distributed environment, from the specific requirements for the networking to the discovery of services.

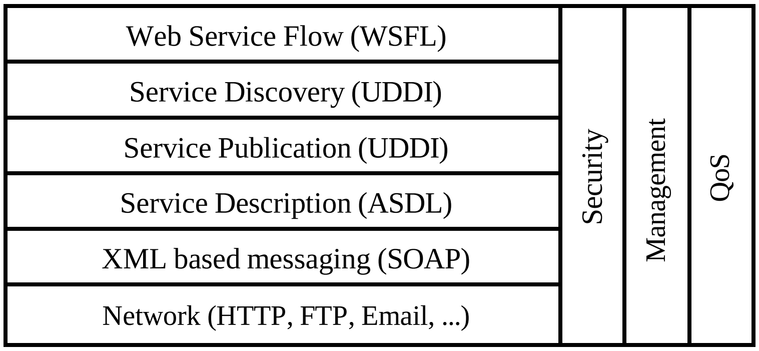

Web Service Flow (WSFL)
Service Discovery (UDDI)
Service Publication (UDDI)
Service Description (ASDL)
XML based messaging (SOAP)
Network (HTTP, FTP, Email, ...)
Security
Management
QoS

Figure 2.2 Web services technologies stack

The backbone of all these technologies is XML, which is also one of the causes of Web service's popularity and ease of use.

XML based languages are used to manage the low level interaction for Web service method calls (SOAP), for providing metadata about the services (WSDL), for discovery services (UDDI), and other core operations.

In practice, the core components that enable Web services are SOAP and WSDL.

Simple Object Access Protocol (SOAP) is an XML based language for exchanging structured information in a platform-independent manner, constitutes the protocol used for Web service method invocation.

Within a distributed context leveraging the Internet, SOAP is considered an application layer protocol that leverages the transport level, most commonly HTTP, for IPC.

SOAP structures the interaction in terms of messages that are XML documents mimicking the structure of a letter, with an envelope, a header, and a body.

The envelope defines the boundaries of the SOAP message.

The header is optional and contains relevant information on how to process the message.

```
Host : www.sample.com
Content-Type: application/soap+xml; charsetutf-8
Content-Length: <Size>

<?xml version= “1.0”>
<soap: Envelope xmlns:soap= “http://www.w3.org/2001/12/soap- envelope”
soap:encodingStyle= “http://www.w3.org/2001/12/soap- enoding” >
<soap:Header></soap:Header>
<soap:Body xmlns=http://www.sample.com/stock>
<m:GetPrice>
<m: StockName>DELL</m:StockName>
</m:GetPrice>
</soap:Body>
</soap: Envelope>

POST /StockPrice HTTP/1.1 Host :
www.sample.com
Content-Type: application/soap+xml; charsetutf-8
Content-Length: <Size>

<?xml version= “1.0”>
<soap: Envelope xmlns:soap= “http://www.w3.org/2001/12/soap- envelope”
soap:encodingStyle= “http://www.w3.org/2001/12/soap- enoding” >
<soap:Header></soap:Header>
<soap:Body xmlns=http://www.sample.com/stock>
<m:GetPriceResponse>
<m: Price>58.5</m:Price>
</m:GetPriceResponse>
</soap:Body>
</soap: Envelope>
```

Figure 2.3 SOAP Message

In addition to that it contains information such as routing and delivery settings, authentication, transaction contexts and authorization assertions.

The body contains the actual message to be processed.

The main uses of SOAP messages are method invocation and result retrieval.

Figure 2.3 shows an example of a SOAP message used to invoke a Web service method that retrieves the price of a given stock and the corresponding reply.

Despite the fact that XML documents are easy to produce and process in any platform or programming language, SOAP has often been considered quite inefficient because of the excessive use of markup that XML imposes for organizing the information into a well-formed document.

Therefore, lightweight alternatives to the SOAP/XML pair have been proposed to support Web services.