

5.5 WEB SERVICE DESCRIPTION LANGUAGE (WSDL)

WSDL stands for Web Services Description Language. It is the standard format for describing a web service. WSDL was developed jointly by Microsoft and IBM.

Features of WSDL

- WSDL is an XML-based protocol for information exchange in decentralized and distributed environments.
- WSDL definitions describe how to access a web service and what operations it will perform.

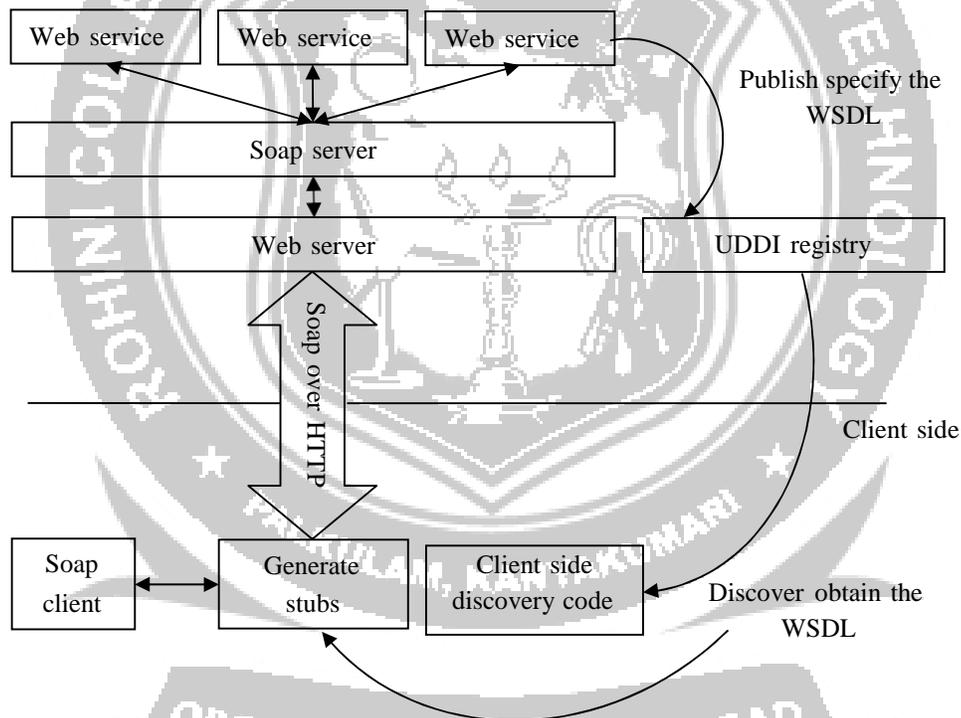


Fig 5.9 Architecture of Web service

- WSDL is a language for describing how to interface with XML-based services.
- WSDL is an integral part of Universal Description, Discovery, and Integration (UDDI), an XML-based worldwide business registry.
- WSDL is the language that UDDI uses.

WSDL addresses this need by defining an XML grammar for describing network services as collections of communication endpoints capable of exchanging messages. WSDL service definitions provide documentation for distributed systems and serve as a recipe for automating the details involved in applications communication.

Steps in providing service:

The following figure illustrates the use of WSDL. At the left is a service provider. At the right is a service consumer. The steps involved in providing and consuming a service are:

- ❖ A service provider describes its service using WSDL. This definition is published to a repository of services. The repository could use Universal Description, Discovery, and Integration (UDDI). Other forms of directories could also be used.

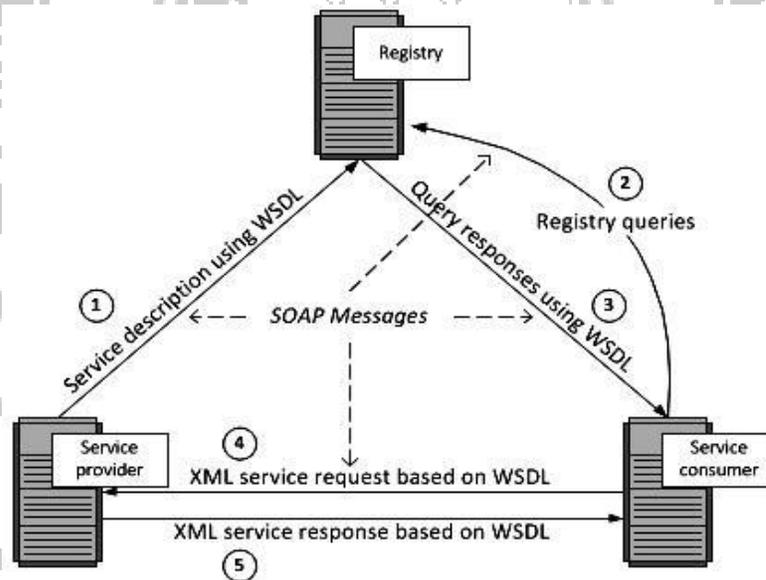


Fig 5.10 WSDL service

- ❖ A service consumer issues one or more queries to the repository to locate a service and determine how to communicate with that service.
- ❖ Part of the WSDL provided by the service provider is passed to the service consumer. This tells the service consumer what the requests and responses are for the service provider.

- ❖ The service consumer uses the WSDL to send a request to the service provider.
- ❖ The service provider provides the expected response to the service consumer.

WSDL Document

A WSDL document defines services as collections of network endpoints, or ports. In WSDL, the abstract definition of endpoints and messages is separated from their concrete network deployment or data format bindings. A port is defined by associating a network address with a reusable binding, and a collection of ports define a service.

WSDL Elements

WSDL breaks down web services into three specific, identifiable elements that can be combined or reused once defined: Types, Operations and Binding. A WSDL document has various elements, but they are contained within these three main elements, which can be developed as separate documents and then they can be combined or reused to form complete WSDL files. A WSDL document contains the following elements:

- **Definition:** It is the root element of all WSDL documents. It defines the name of the web service, declares multiple namespaces used throughout the remainder of the document, and contains all the service elements described here.
- **Data types:** The data types to be used in the messages are in the form of XML schemas.
- **Message:** It is an abstract definition of the data, in the form of a message presented either as an entire document or as arguments to be mapped to a method invocation.
- **Operation:** It is the abstract definition of the operation for a message, such as naming a method, message queue, or business process, that will accept and process the message.
- **Port type:** It is an abstract set of operations mapped to one or more end-points, defining the collection of operations for a binding; the collection of operations, as it is abstract, can be mapped to multiple transports through various bindings.
- **Binding:** It is the concrete protocol and data formats for the operations and messages defined for a particular port type.
- **Port:** It is a combination of a binding and a network address, providing the target address of the service communication.
- **Service:** It is a collection of related end-points encompassing the service definitions in the file; the services map the binding to the port and include any extensibility definitions.
- **Documentation:** This element is used to provide human-readable documentation and can be included inside any other WSDL element.
- **Import:** This element is used to import other WSDL documents or XML Schemas.

Document Structure of WSDL

```

<definitions> <types>
definition of types..... </types>
<message> definition of a message .....</message>
<portType> <operation> definition of a operation. .... </operation>
</portType>
<binding> definition of a binding. ....</binding>
<service> definition of a service..... </service> </definitions>

```

CONSUMING A WEB SERVICE

A web service can be consumed (or called) by a client application. Different types of client applications can consume a web service. In today's software environment, almost every application needs a web service to enhance its functionality. The important advantage of a web service is that it returns its results in xml format, which can be consumed by different types of clients like browser based clients, rich desktop clients, spreadsheets, wireless devices, interactive voice response(IVR) systems and other business applications.

A client application discovers a web service, and then uses services provided by the web service. This process is known as **consuming a Web service**.

Creating Web Ports

Web ports are specially configured ports that you use to consume (call) Web services. A Web port can contain multiple operations that represent a mix of one-way (request only) and two-way (request-response) Web methods. Each operation in a Web port represents one method of a Web service.

Adding Web References

A Web reference is a description of a Web service that is available to the project. A Web reference includes:

- A Universal Resource Locator (URL) for the Web service.
- A WSDL file that offers information about the service such as available methods, ports, and message types.
- A reference map (Reference.map).

When the user add a Web reference, all the Web methods for that Web service must be compatible with the Server.

DATABASE DRIVEN WEB APPLICATION

Web services enable application-to-application interaction over the Web, regardless of platform, language, or data formats. The key ingredients, including Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI), have been adopted across the entire software industry. The Database Web services technology is a database approach to Web services. It works in the following two directions:

- Accessing database resources as a Web service
- Consuming external Web services from the database

Oracle Database can access Web services through PL/SQL packages and Java classes deployed within the database.

Using Oracle Database as Web Services Provider

Web Services use industry-standard mechanisms to provide easy access to remote content and applications, regardless of the platform and location of the provider and implementation and data format. Client applications can query and retrieve data from Oracle Database and call stored procedures using standard Web service protocols. There is no dependency on Oracle-specific database connectivity protocols. This approach is highly beneficial in heterogeneous, distributed, and disconnected environments. Using Oracle Database as a Web service provider offers the following features:

- Enhances PL/SQL Web services
- Exposes Java in the database as Web services
- Provides SQL query Web services
- Enables DML Web services

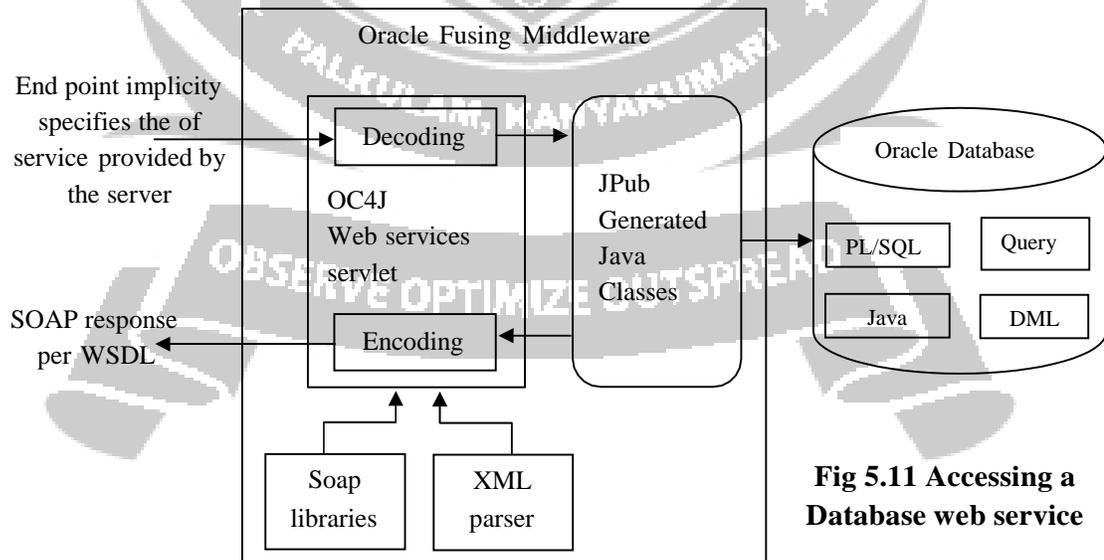


Fig 5.11 Accessing a Database web service

Using Oracle Database as Web Services Consumer

The storage, indexing, and searching capabilities of a relational database can be extended to include semi-structured and non-structured data, including Web services. By calling Web services, the database can track, aggregate, refresh, and query dynamic data produced on-demand, such as stock prices, currency exchange rates, and weather information.

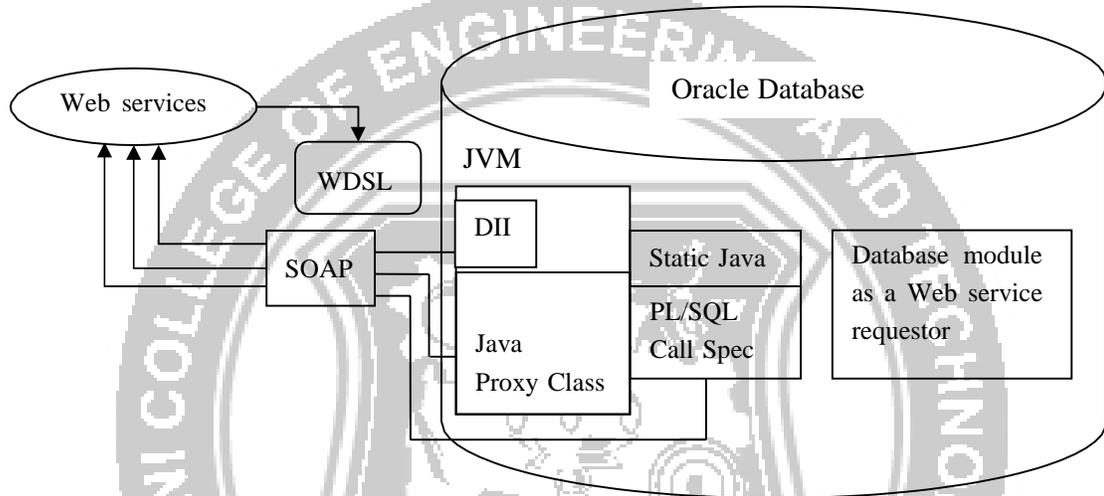


Fig 5.12 Calling web services within a database

Web Service Data Sources (Virtual Table Support)

To access data that is returned from single or multiple Web service invocations, create a virtual table using a Web service data source. This table lets the user to query a set of returned rows as though it were a table.

The client calls a Web service and the results are stored in a virtual table in the database. The result sets can be passed from function to function. This enables the user to set up a sequence of transformation without a table holding intermediate results. By using Web services with the table function, a range of input values can be manipulated from single or multiple Web services as a real table.

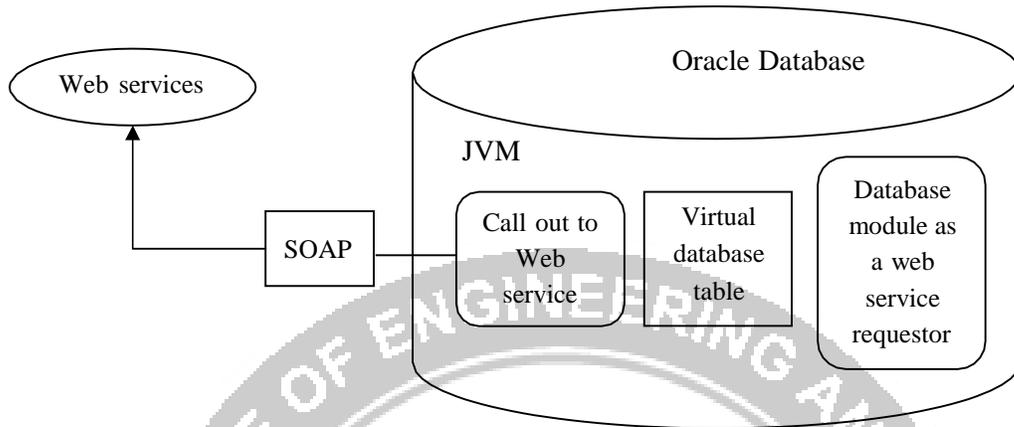


Fig 5.13 Virtual table

