WSDL - Web Services Description Language

List of Topics

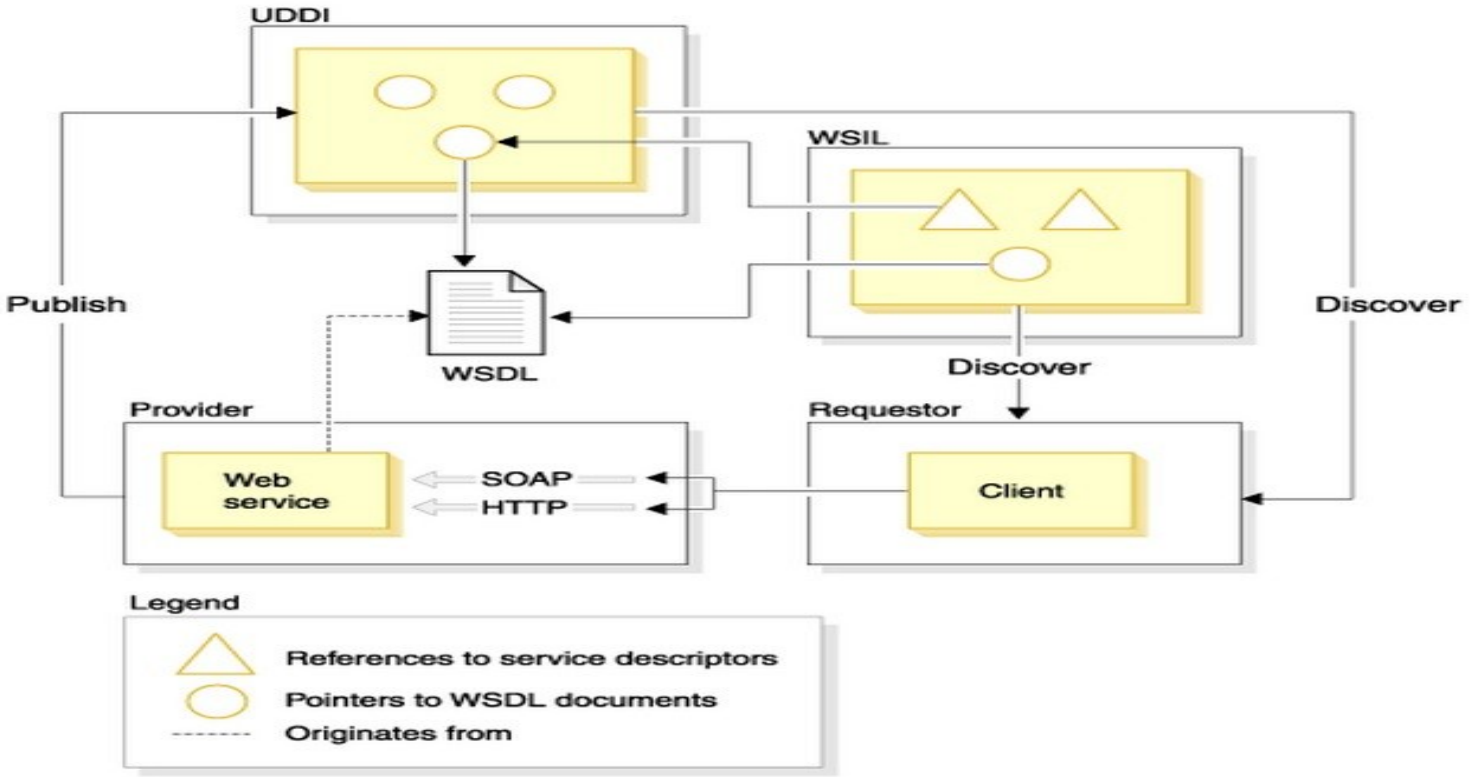
- WSDL Description
- WSDL Elements
  - Definition
  - Type
  - Message
  - Port type
  - Binding
  - Port
  - Service
  - WSDL 1.0 and 2.0
  - Summary

WSDL Describes Web Services

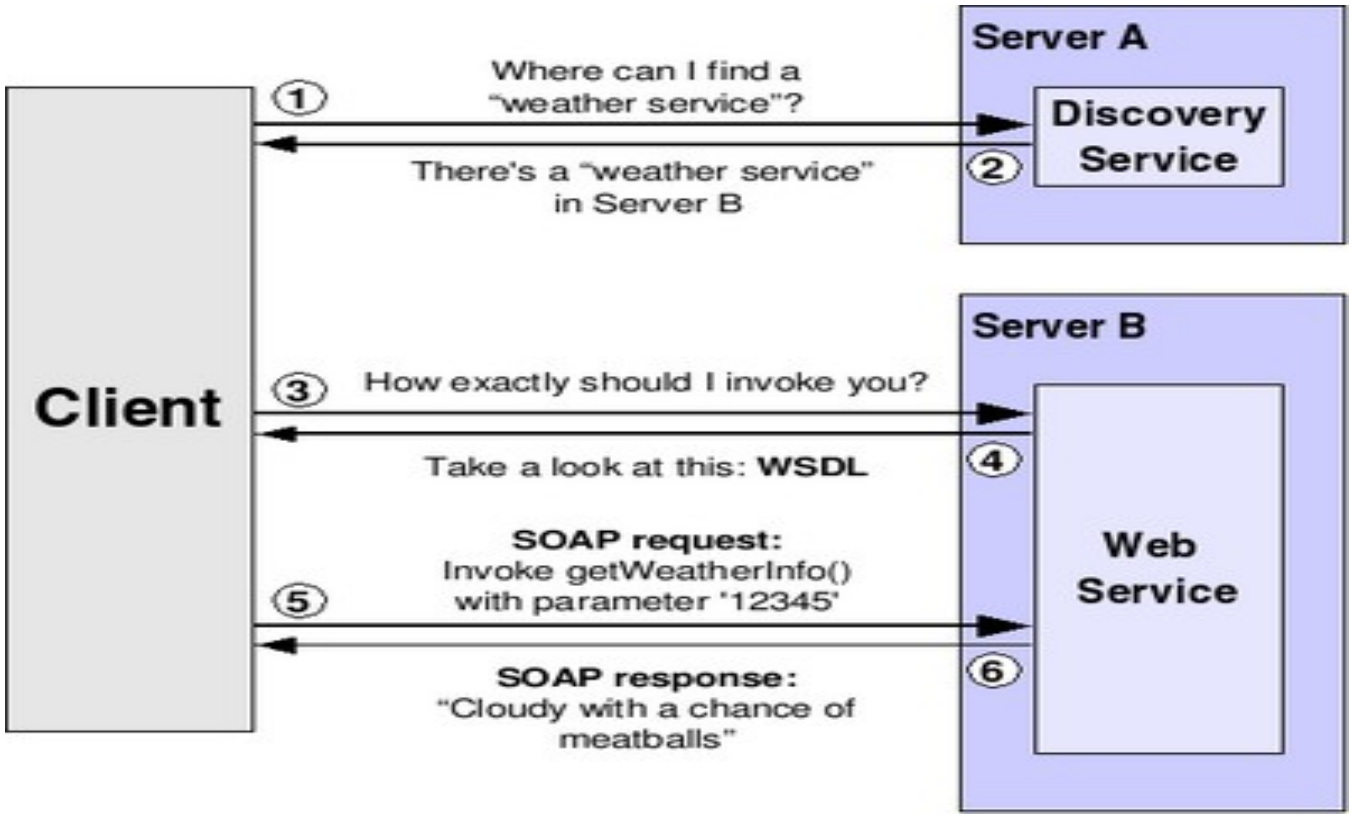
- WSDL stands for Web Services Description Language.
- WSDL is a document written in XML. The document describes a Web service. It specifies the location of the service and the operations (or methods) the service exposes.
- A WSDL description of a web service (also referred to as a *WSDL file*) provides a machine-readable description of how the service can be called what parameters it expects and what data structures it returns.
- WSDL was developed jointly by Microsoft and IBM initially.
- WSDL is pronounced as 'wiz-dull' and spelled out as 'W-S-D-L'

WSDL Usage

WSDL is often used in combination with SOAP and XML Schema to provide web services over the Internet. A client program connecting to a web service can read the WSDL to determine what functions are available on the server. The client can then use SOAP to actually call one of the functions listed in the WSDL.



WSDL example



Structure of a WSDL document

<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>

A WSDL document has various elements, but they are contained within these main elements, which can be developed as separate documents and then they can be combined or reused to form complete WSDL files.

WSDL Elements overview

- **Definition:** Element must be 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 - in the form of XML schemas or possibly some other mechanism - to be used in the messages
- **Message:** An abstract definition of the data in the form of a message presented either as an arguments to be mapped to a method invocation.
- **Operation:** An 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 :** An abstract set of operations mapped to one or more end points, defining the collection of operations for a binding

WSDL Elements overview (cont..)

- **Binding:** the concrete protocol and data formats for the operations and messages defined for a particular port type.
- **Port:** a combination of a binding and a network address, providing the target address of the service communication.
- **Service:** a collection of related end points encompassing the service definitions in the file.

WSDL Elements

- **<definition>** Element : It must be the root element of all WSDL documents. It defines the name of the web service.

```
<definitions name="HelloService"
    targetNamespace="http://www.examples.com/wsdl/HelloService.wsdl"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://www.examples.com/wsdl/HelloService.wsdl"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"> .....
</definitions>
```

  - <**Type**> Element:
  - A Web service needs to define its inputs and outputs and how they are mapped into and out of services. WSDL <types> element take care of defining the data types that are used by the web service.
  - It describes all the data types used between the client and server.
  - WSDL uses the **W3C XML Schema specification** as its default choice to define data types
  - If the service uses only XML Schema built-in simple types, such as strings and integers, then types element is not required (**Refer HelloService.wsdl**)

Type Elements cont…

- WSDL allows the types to be defined in separate elements so that the types are reusable with multiple Web services.

```
<types>
  <schema targetNamespace="http://example.com/stockquote.xsd"
          xmlns="http://www.w3.org/2000/10/XMLSchema">
    <element name="TradePriceRequest">
      <complexType>
        <all>
          <element name="tickerSymbol" type="string"/>
        </all>
      </complexType>
    </element>
    <element name="TradePrice">
      <complexType>
        <all>
          <element name="price" type="float"/>
        </all>
      </complexType>
    </element>
  </schema>
</types>
```

Elements cont…

- **<message>** Element:
  - It describes the data being exchanged between the Web service providers and consumers.
  - Each Web Service has two messages: input and output.
  - The input describes the parameters for the Web Service and the output describes the return data from the Web Service.
  - Each message contains zero or more **<part>** parameters, one for each parameter of the Web Service's function.
  - Each **<part>** parameter associates with a concrete type defined in the **<types>** container element.

```
<message name="SayHelloRequest">
   <part name="firstName" type="xsd:string"/>
</message>
<message name="SayHelloResponse">
   <part name="greeting" type="xsd:string"/>
</message>
```

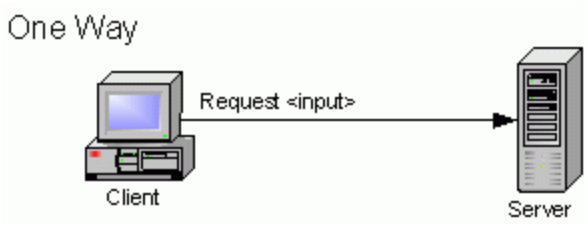
Elements cont…

- **<portType>** Element: It combines multiple message elements to form a complete one-way or round-trip operation.
- **One-way :**

  The service receives a message. The operation therefore has a single *input* element. which is the client's request to the server. **No response is expected.**

```
<operation name="weatherUpdate">
<input message="tns:weatherUpdate"/>
</operation>

<message name="weatherUpdate">
<part name="weatherData" type="wsx:WeatherSummary"/>
</message>
```



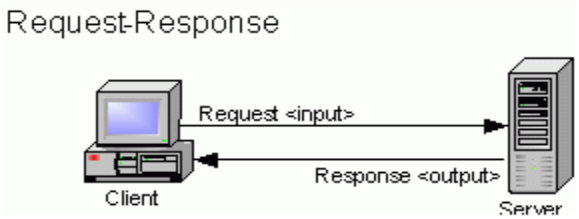
PortType Elements cont…

- **Request-response:**

  The service receives a message and sends a response. The operation therefore has one *input* element, followed by one *output* element.

```
<operation name="subscribe">
<input message="tns:subscribe"/>
<output message="tns:subscribeResponse"/>
</operation>
```

```
<message name="subscribe">
<part name="host" type="xsd:string"/>
<part name="port" type="xsd:string"/>
<part name="transportName" type="xsd:string"/>
</message>
```



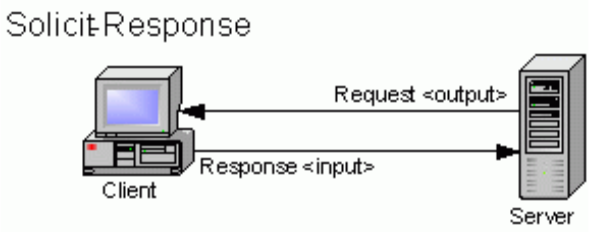
PortType Elements cont…

- **Solicit-response:**

  The solicit-response operation includes one output element, which is the server's request to the client, followed by one input element, which is the client's response back to the server.

```
<operation name="verifySubscription">
<output message="tns:verifySubscription"/>
<input message="tns:verifySubscriptionResponse"/>
</operation>

 <message name="verifySubscriptionResponse">
<part name="status" type="xsd:boolean"/>
 </message>
```



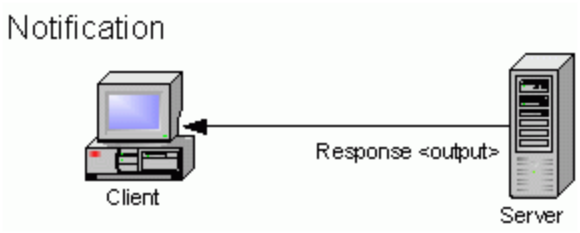
PortType Elements cont…

- **Notification :**

  The service sends a message. The operation therefore has a single *output* element.

```
<operation name="weatherNotification">
<output message="tns:weatherNotification"/>
</operation>

<message name="weatherNotification">
<part name="weatherData" type="wsx:WeatherSummary"/>
</message>
```



Elements cont…

- **<port>** element**:**
  - The port element has two attributes - the name attribute and the binding attribute.
  - The name attribute provides a unique name among all ports defined within in the enclosing WSDL document.
  - The binding attribute refers to the binding using the linking rules defined by WSDL.
  - Binding extensibility elements (1) are used to specify the address information for the port.
  - A port MUST NOT specify more than one address.
  - A port MUST NOT specify any binding information other than address information.

```
<service name="Hello_Service">
     <documentation>WSDL File for HelloService</documentation>
     <port binding="tns:Hello_Binding" name="Hello_Port">
        <soap:address
           location="http://www.examples.com/SayHello/">
     </port>
  </service>
```

Elements cont…

- **<binding>** element
  - provides specific details on how a *portType* operation will actually be transmitted over the wire.
  - The bindings can be made available via multiple transports, including SOAP over HTTP GET, HTTP POST
  - The bindings provide concrete information on what protocol is being used to transfer *portType* operations.
  - The bindings provide information where the service is located.
  - For SOAP protocol, the binding is **<soap:binding>**, and the transport is SOAP messages on top of HTTP protocol.
  - You can specify multiple bindings for a single *portType*.

<binding name="Hello_Binding" type="tns:Hello_PortType">

The name attribute defines the name of the binding, and the type attribute points to the port for the binding, in this case the "tns:Hello_PortType" port.

Binding Elements cont…

- **SOAP Binding**

  WSDL 1.1 includes built-in extensions for SOAP 1.1. This enables you to specify SOAP specific details, including SOAP headers, SOAP encoding styles, and the SOAPAction HTTP header. The SOAP extension elements include:

- **soap:binding**

  This element indicates that the binding will be made available via SOAP.
  The *style* attribute indicates the overall style of the SOAP message format. A style value of rpc specifies an RPC format.

  The *transport* attribute indicates the transport of the SOAP messages. The value *http://schemas.xmlsoap.org/soap/http* indicates the SOAP HTTP transport, whereas *http://schemas.xmlsoap.org/soap/smtp* indicates the SOAP SMTP transport.

- **soap:operation**

  This element indicates the binding of a specific operation to a specific SOAP implementation. The soapAction attribute specifies that the SOAPAction HTTP header be used for identifying the service.

Binding Elements cont…

- **soap:body**

  This element enables you to specify the details of the input and output messages. In the case of HelloWorld, the body element specifies the SOAP encoding style and the namespace URN associated with the specified service.

```
<binding name="Hello_Binding" type="tns:Hello_PortType">
<soap:binding style="rpc"
   transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="sayHello">
   <soap:operation soapAction="sayHello"/>
   <input>
      <soap:body
         encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="urn:examples:helloservice"
         use="encoded"/>
   </input>
   <output>
      <soap:body
         encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="urn:examples:helloservice"
         use="encoded"/>
   </output>
</operation>
</binding>
```

Elements cont…

- **<service>** Element:
  - Defines the ports supported by the Web service. For each of the supported protocols, there is one port element. The service element is a collection of ports.
  - Web service clients can learn from the service element where to access the service, through which port to access the Web service, and how the communication messages are defined.
  - The service element includes a documentation element to provide human-readable documentation.

```
<service name="Hello_Service">
    <documentation>WSDL File for HelloService</documentation>
    <port binding="tns:Hello_Binding" name="Hello_Port">
       <soap:address
          location="http://www.examples.com/SayHello/">
    </port>
 </service>
```

HelloService.wsdl

- **Content of HelloService.wsdl file (see attached file)**

- **Analysis of the wsdl file**

  Definition : HelloService

  Type : Using built-in data types and they are defined in XMLSchema.

  Message :

  - sayHelloRequest : firstName parameter
  - sayHelloresponse: greeting return value

  Port Type: sayHello **operation** that consists of a request and response service.

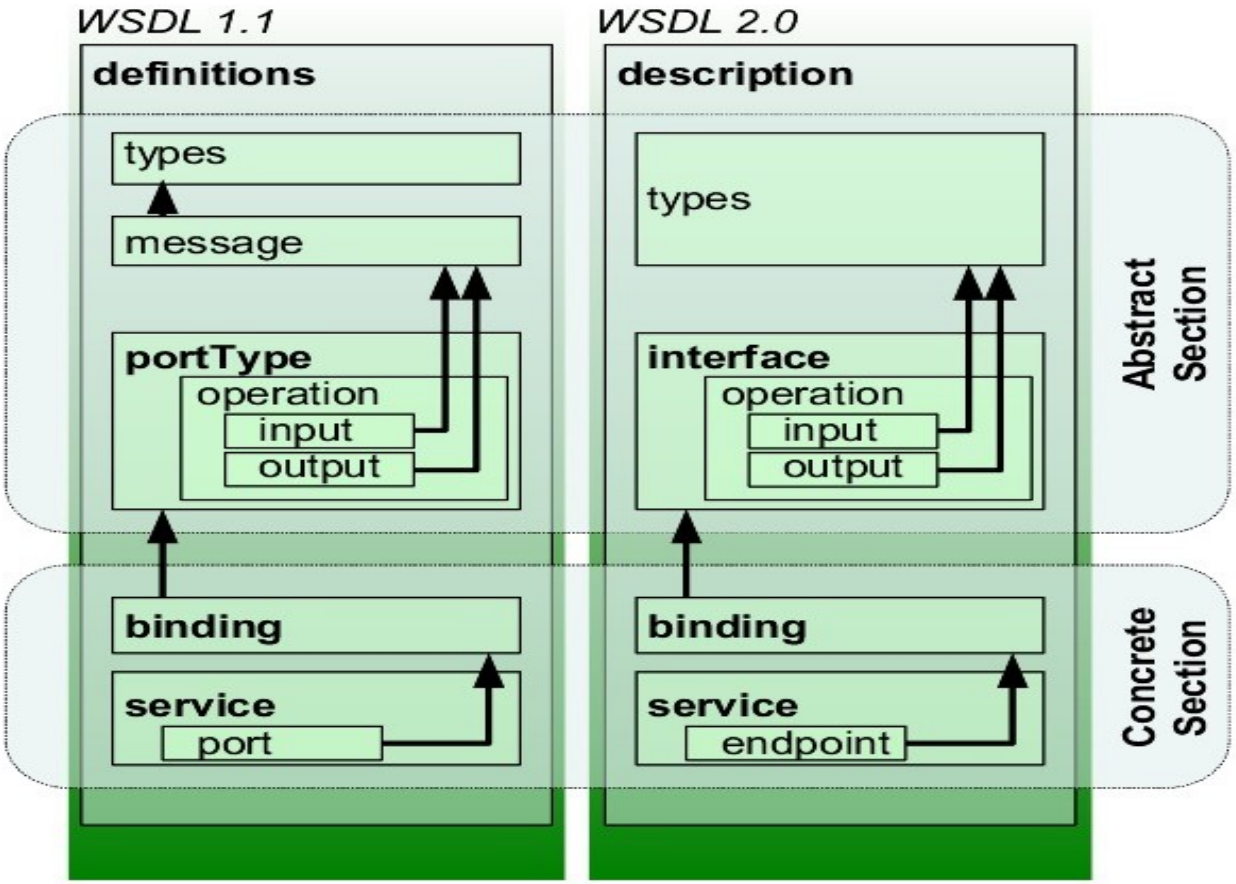
  Binding: Direction to use the SOAP HTTP transport protocol.

  Service: Service available at http://www.examples.com/SayHello/.

  Port: Associates the binding with the URI http://www.examples.com/SayHello/ where the running service can be accessed.

HelloService.wsdl

WSDL 1.1 and 2.0 difference



Difference between 1.1 and 2.0

WSDL 1.1 Term	WSDL 2.0 Term	Description
Service	Service	Contains a set of system functions that have been exposed to the Web-based protocols.
Port	Endpoint	Defines the address or connection point to a Web service. It is typically represented by a simple HTTP URL string.
Binding	Binding	Specifies the interface and defines the SOAP binding style (RPC/Document) and transport (SOAP Protocol). The binding section also defines the operations.
PortType	Interface	Defines a Web service, the operations that can be performed, and the messages that are used to perform the operation.
Operation	Operation	Defines the SOAP actions and the way the message is encoded, for example, "literal." An operation is like a method or function call in a traditional programming language.

Difference cont…

Message	**n/a**	Typically, a message corresponds to an operation. The message contains the information needed to perform the operation. Each message is made up of one or more logical parts. Each part is associated with a message-typing attribute. The message name attribute provides a unique name among all messages. The part name attribute provides a unique name among all the parts of the enclosing message. Parts are a description of the logical content of a message. In RPC binding, a binding may reference the name of a part in order to specify binding-specific information about the part. A part may represent a parameter in the message; the bindings define the actual meaning of the part. Messages were removed in WSDL 2.0, in which XML schema types for defining bodies of inputs, outputs and faults are referred to simply and directly.
Types	Types	Describes the data. The XML Schema language (also known as XSD) is used (inline or referenced) for this purpose.

WSDL 2.0

- WSDL 2.0 became a W3C recommendation on June 2007. WSDL 1.2 was renamed to WSDL 2.0 because it has substantial differences from WSDL 1.1. The changes are the following:
- Added further semantics to the description language
- Removed message constructs
- Operator overloading not supported
- PortTypes renamed to interfaces
- Ports renamed to endpoints

WSDL and UDDI

Universal Description, Discovery and Integration (UDDI) is a directory service where businesses can register and search for Web services.

What is UDDI

- UDDI is a platform-independent framework for describing services, discovering businesses, and integrating business services by using the Internet.
- UDDI is a directory for storing information about web services
- UDDI is a directory of web service interfaces described by WSDL
- UDDI communicates via SOAP

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

Who is Supporting UDDI?

- UDDI is a cross-industry effort driven by all major platform and software providers like Dell, Fujitsu, HP, Hitachi, IBM, Intel, Microsoft, Oracle, SAP, and Sun, as well as a large community of marketplace operators, and e-business leaders.
- Over 220 companies are members of the UDDI community.

Summary:

- **Web Services**

  Web services are open standard (XML, SOAP, HTTP etc.) based Web applications that interact with other web applications for the purpose of exchanging data.
- **UDDI**

  UDDI is an XML-based standard for describing, publishing, and finding Web services.
- **SOAP**

  SOAP is a simple XML-based protocol that allows applications to exchange information over HTTP.