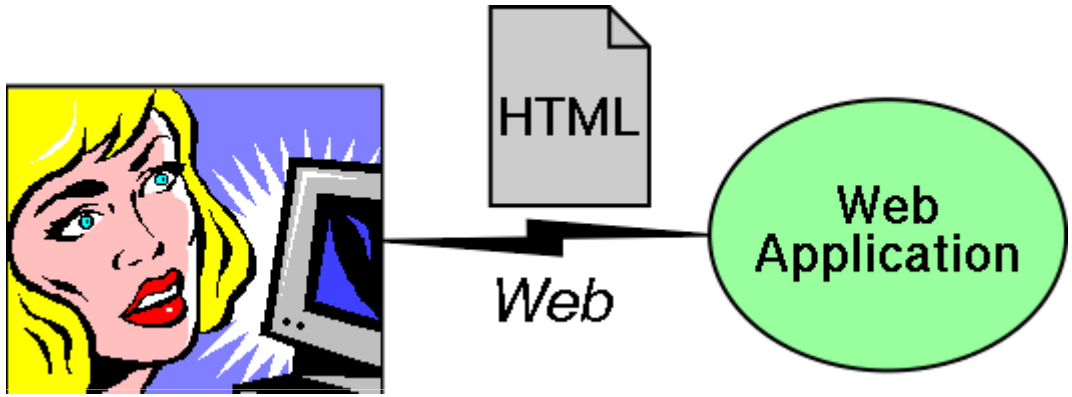
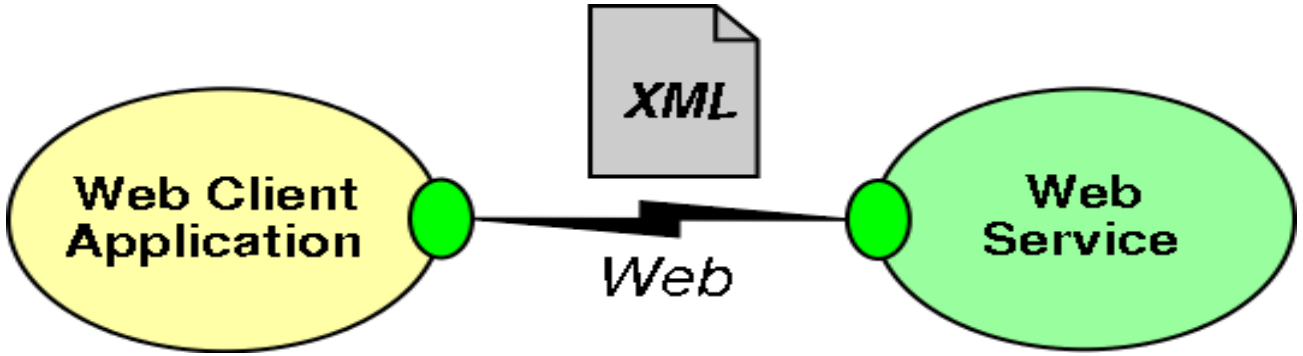
Java Web Service

Web Services Concepts

- A web application uses Web technologies to provide functionality to an end user
- A web service uses Web technologies to provide functionality to another software application

Traditional web application/Web service





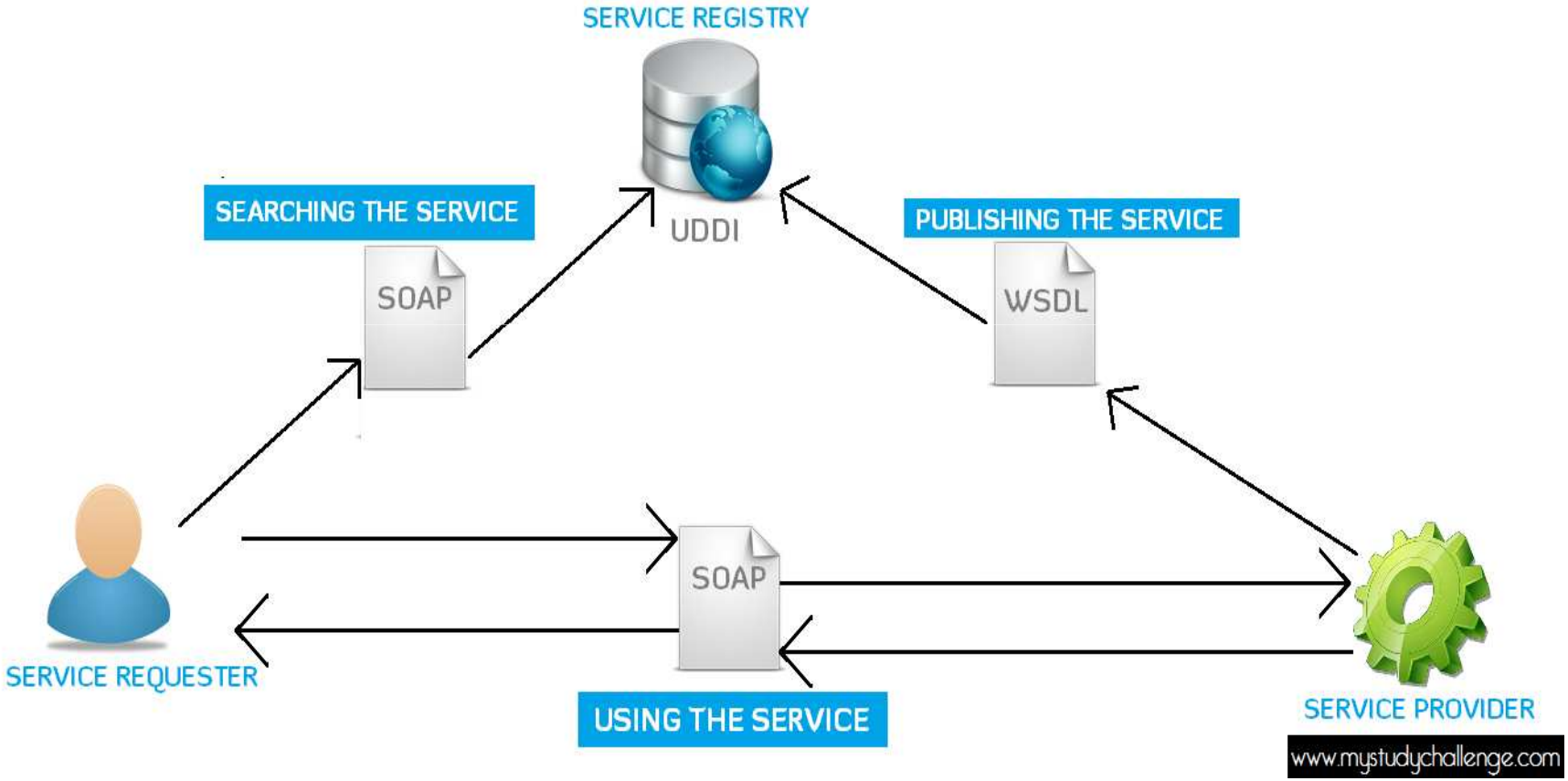
What are Web Services?

A simpler definition, and perhaps more useful, might be: "a Web service is a software application, accessible on the Web through a URL, that is accessed by clients using XML-based protocols, such as Simple Object Access Protocol (SOAP) sent over accepted Internet protocols, such as HTTP. Clients access a Web service application through its interfaces and bindings, which are defined using XML artifacts, such as a Web Services Definition Language (WSDL) file."

Advantages of Web service

- Not based on a programming language: *Java, .Net, C, C++, Python, Perl, …*
- Not based on a programming data model: objects vs non-objects environments.
- Convergence of SOA (Service-Oriented Architecture) and Web.
- Based on web technologies
- Do not need huge framework of memory

Web service Architecture



WS built on existing standards

- Extensible Markup Language (XML)
- The HTTP (Hypertext Transfer Protocol) standard is allowing more systems to communicate with one another.
- SOAP (Simple Object Access Protocol) (built on XML) standardizes the messaging capability on different systems.
- UDDI (Universal Description, Discovery, and Integration) standardizes the publishing and finding of Web services.
- WSDL (Web Services Description Language) standardizes the description of Web services so providers and requesters are speaking the same language.

Simple Object Access Protocol

- SOAP is a communication protocol designed to communicate via Internet.
- SOAP can extend HTTP for XML messaging.
- SOAP provides data transport for Web services.
- SOAP can exchange complete documents or call a remote procedure.
- SOAP can be used for broadcasting a message.
- SOAP is platform- and language-independent.
- SOAP is the XML way of defining what information is sent and how.
- SOAP enables client applications to easily connect to remote services and invoke remote methods.

Universal Description, Discovery, and Integration.

- UDDI is a specification for a distributed registry of web services.
- UDDI is a platform-independent, open framework.
- UDDI can communicate via SOAP, CORBA, Java RMI Protocol.
- UDDI uses Web Service Definition Language(WSDL) to describe interfaces to web services.
- UDDI is seen with SOAP and WSDL as one of the three foundation standards of web services.
- UDDI is an open industry initiative, enabling businesses to discover each other and define how they interact over the Internet.

Web Services Description Language

- WSDL is an XML based protocol for information exchange in decentralized and distributed environments.
- WSDL is the standard format for describing a web service.
- WSDL definition describes how to access a web service and what operations it will perform.
- WSDL is a language for describing how to interface with XML-based services.

WSDL Document Structure

```
<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>
```

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

```
<portType name="Hello_PortType">
      <operation name="sayHello">
        <input message="tns:SayHelloRequest"/>
        <output message="tns:SayHelloResponse"/>
      </operation>
   </portType>
   <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>

  <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>
</definitions>
```

WSDL - Example

- **Definitions** : 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 a 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.