



Programación Orientada a Servicios

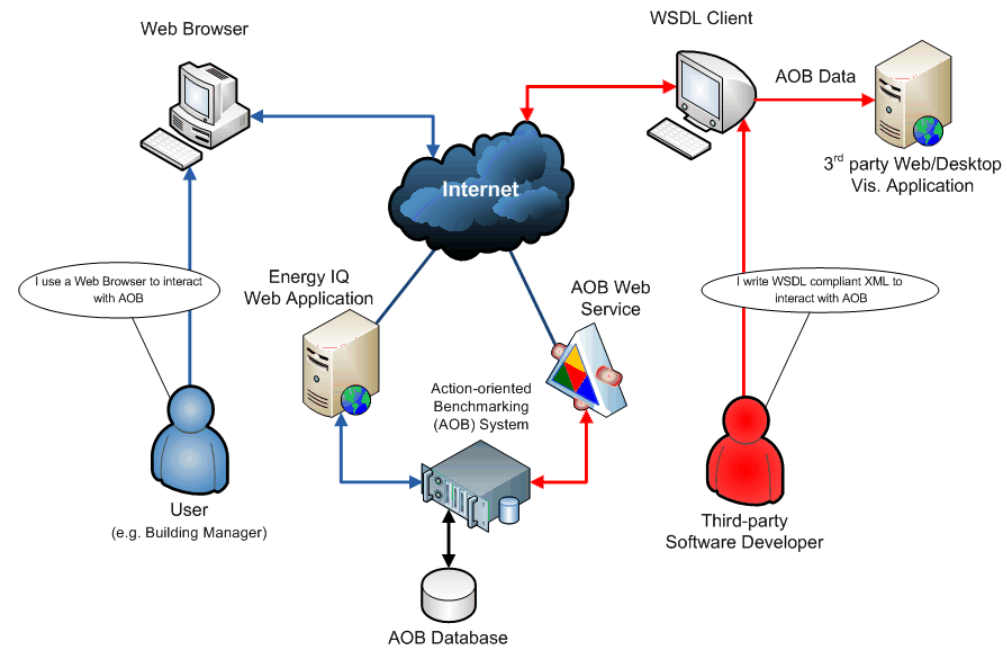
Introducción a los Servicios Web

Programa de
Ingeniería en Computación
UAM – Azcapotzalco

A cargo de:
Dra. Maricela Claudia Bravo Contreras
mcbc@correo.azc.uam.mx

Servicios Web

- Un **servicio web** es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
- Aplicaciones desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes públicas o privadas.



¿Qué es un Servicio Web?

3

- Componente de software **reutilizable** y distribuido que ofrece una funcionalidad concreta, **independiente** tanto del lenguaje de programación en que está implementado como de la plataforma de ejecución.
- Componente de aplicación programable al que se puede acceder mediante **protocolos estándar** de Internet.

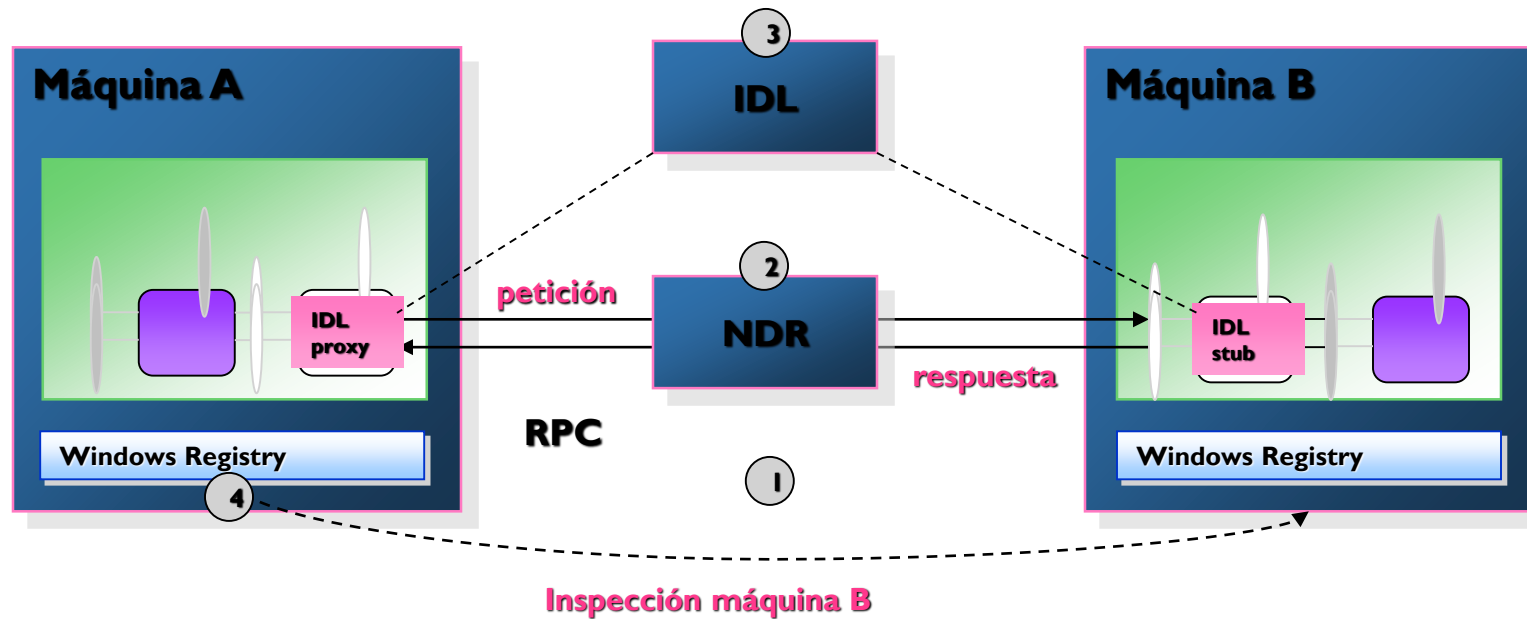
Introducción a los WS

Arquitecturas Distribuidas

- Objetos distribuidos para aplicaciones complejas
- Alternativas principales
 - DCOM (Microsoft)
 - CORBA (OMG)
 - RMI (Sun-Oracle)

Introducción a los WS

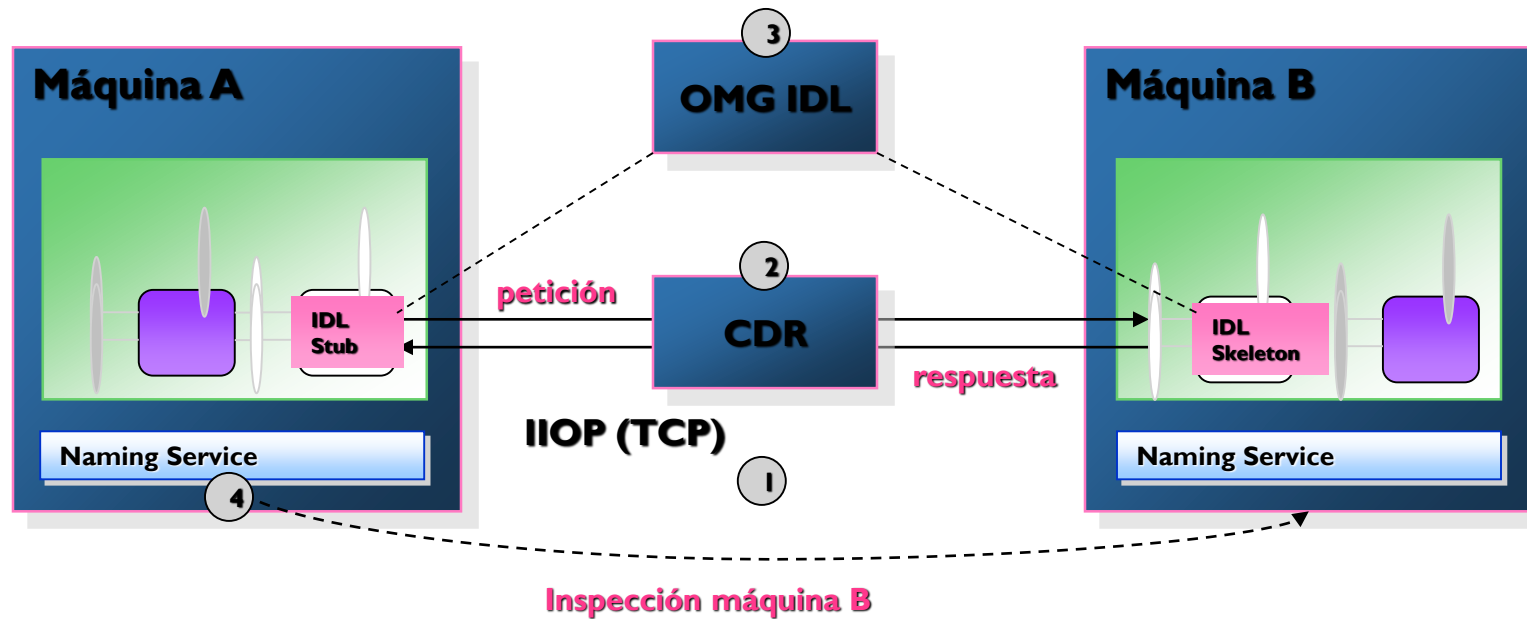
Arquitecturas Distribuidas (DCOM)



- 1 **Protocolo de comunicaciones**
- 2 **Formato de mensaje**
- 3 **Lenguaje de descripción**
- 4 **Mecanismo de localización**

Introducción a los WS

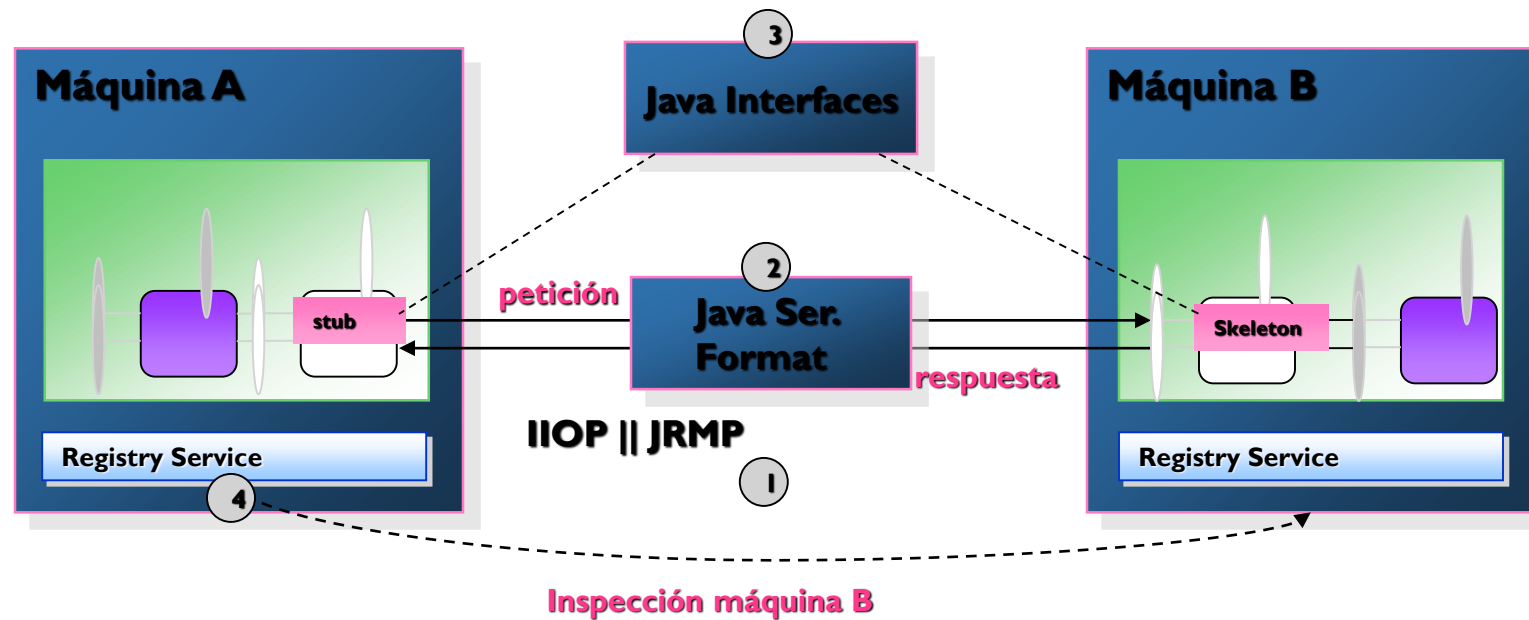
Arquitecturas Distribuidas (CORBA)



- 1 **Protocolo de comunicaciones**
- 2 **Formato de mensaje**
- 3 **Lenguaje de descripción**
- 4 **Mecanismo de localización**

Introducción a los WS

Arquitecturas Distribuidas (Java RMI)



- 1 **Protocolo de comunicaciones**
- 2 **Formato de mensaje**
- 3 **Lenguaje de descripción**
- 4 **Mecanismo de localización**

Introducción a los WS

Arquitecturas Distribuidas - problemas

	DCOM	CORBA	Java RMI
RPC Protocol	RPC	IIOP	IIOP or JRMP
Message Format	NDR	CDR	Java Ser. Format
Description	IDL	OMG IDL	Java
Discovery	Windows Registry	Naming Service	RMI Registry or JNDI

- Estas tecnologías no interoperan entre sí
- Es necesaria una arquitectura independiente:
 - Del lenguaje
 - De la plataforma
 - De las características de los objetos
 - Del mecanismo de llamada
- SOA : Service Oriented Architecture

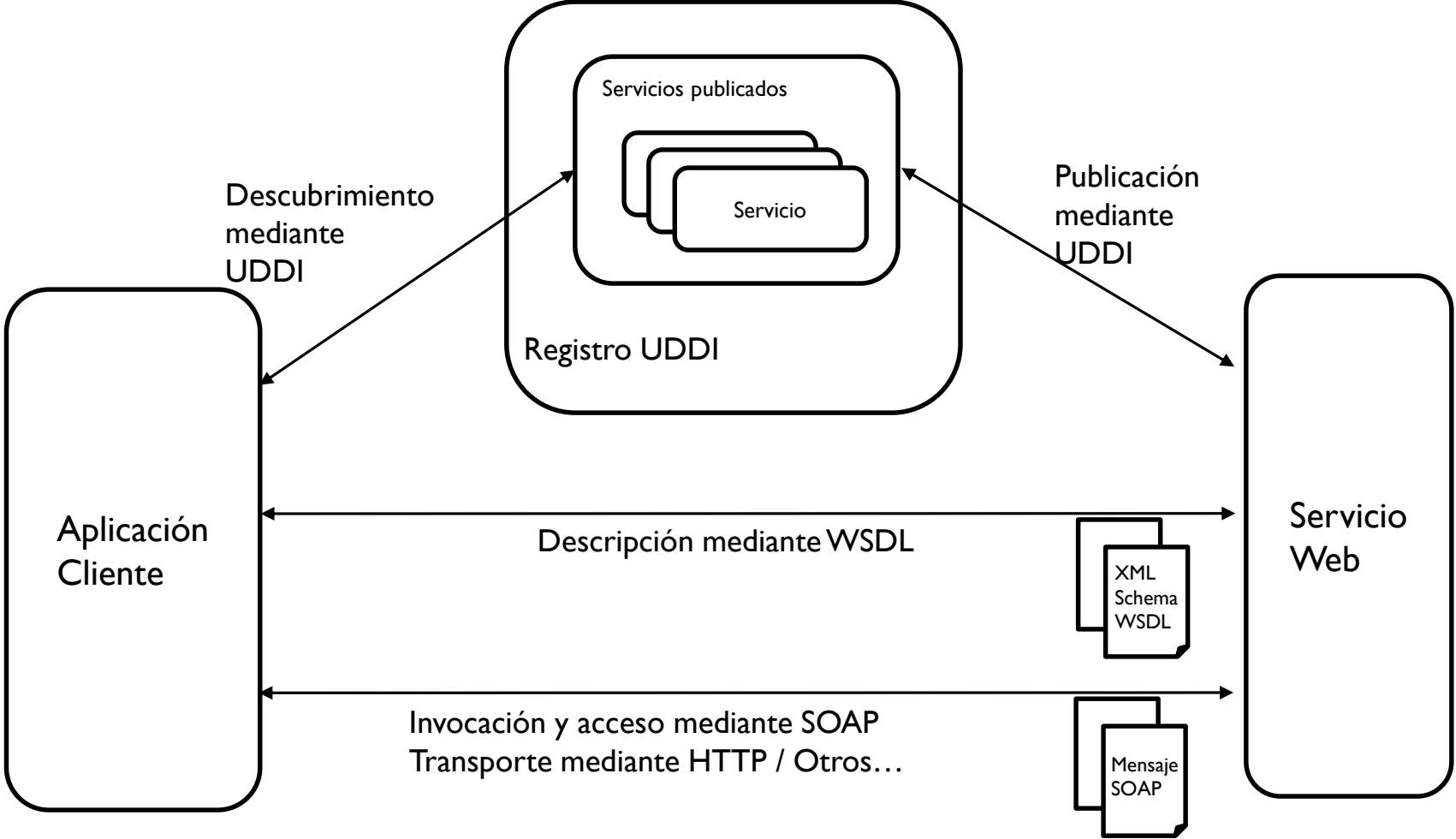
Introducción a los WS

Arquitecturas Distribuidas (SOA)

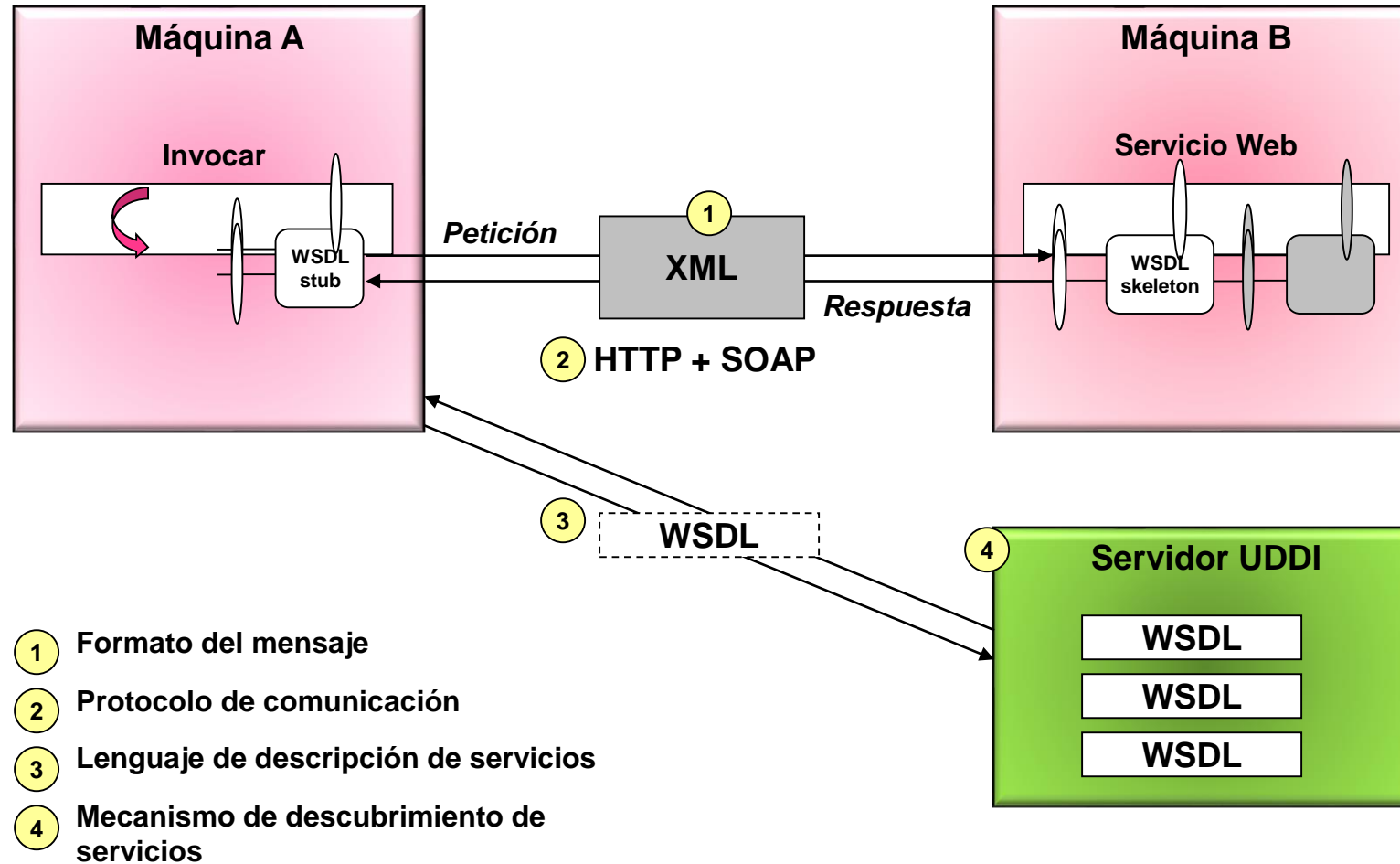
- Servicios Web son una implementación de SOA
- Se redefinen los elementos de las arquitecturas distribuidas, ahora basándolas en XML
 - Comunicación mediante protocolos estándares de Internet
 - HTTP, u otros: SMTP, FTP...
 - Formato del mensaje: SOAP
 - Definición de servicios: WSDL
 - Como localizador: UDDI

Introducción a los WS

Arquitecturas Distribuidas (SOA)



ORPC de Servicios Web



Características de los Servicios Web

12

- **Interoperabilidad**
- **Encapsulación**
- **Disponibilidad**
- Las características anteriores podrían ser parte también de otras arquitecturas como DCOM, CORBA, RMI.
- Sin embargo los servicios Web ofrecen comunicar diferentes aplicaciones realizadas en diversas plataformas sin ningún problema.



Interoperabilidad

- El IEEE define **interoperabilidad** como la habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.
- La habilidad de organizaciones y sistemas dispares y diversos para **interaccionar** con objetivos consensuados y comunes y con la finalidad de obtener beneficios mutuos.

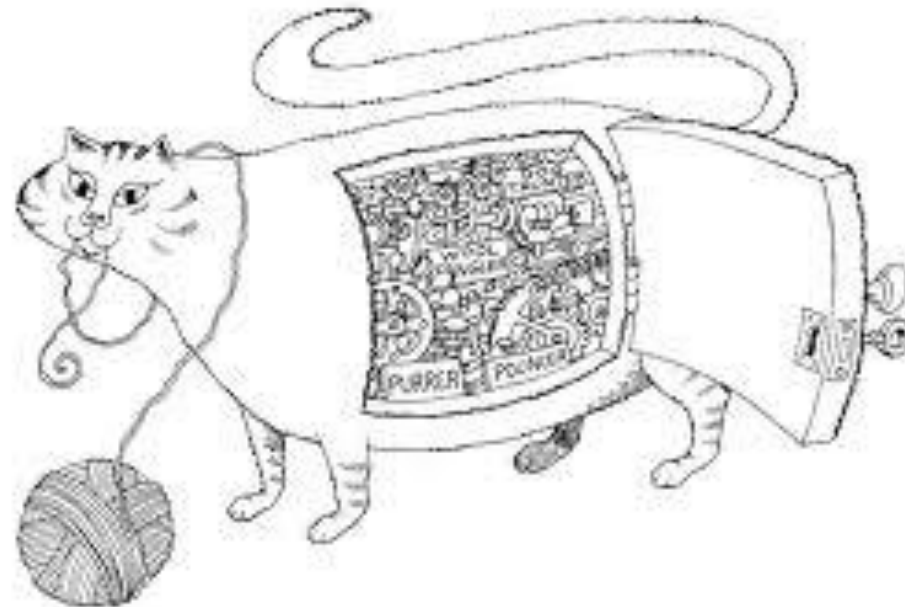


La **interacción** implica que las organizaciones involucradas compartan información y conocimiento a través de sus procesos de negocio, mediante el intercambio de datos entre sus respectivos sistemas de tecnología de la información y las comunicaciones.

Encapsulación

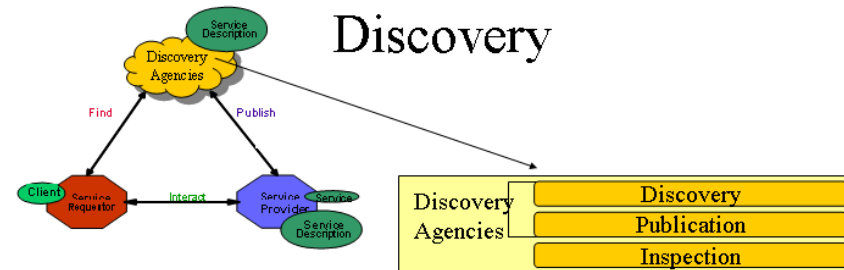
- Se denomina **encapsulamiento** al ocultamiento del estado, es decir, de los datos de un componente de manera que sólo se pueda cambiar mediante las operaciones definidas.
- Cada componente está aislado del exterior, es un módulo natural, y la aplicación entera se reduce a un agregado o rompecabezas de componentes.
- El aislamiento protege a los datos asociados de un componente contra su modificación por quien no tenga derecho a acceder a ellos, eliminando efectos secundarios e interacciones.

Una aplicación que utilice Servicios Web no necesita conocer los detalles de programación del componente.



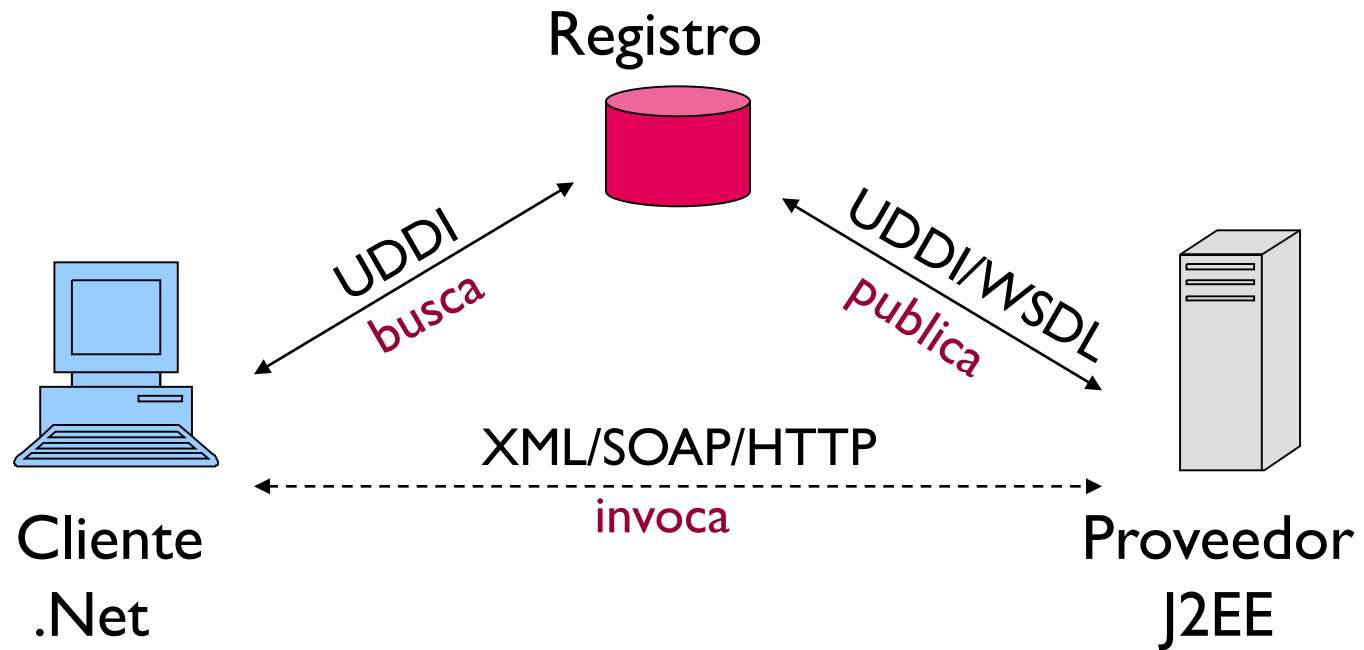
Disponibilidad

- Un componente debe poder publicarse con la suficiente información para que cualquier programador lo utilice en las aplicaciones que desarrolle



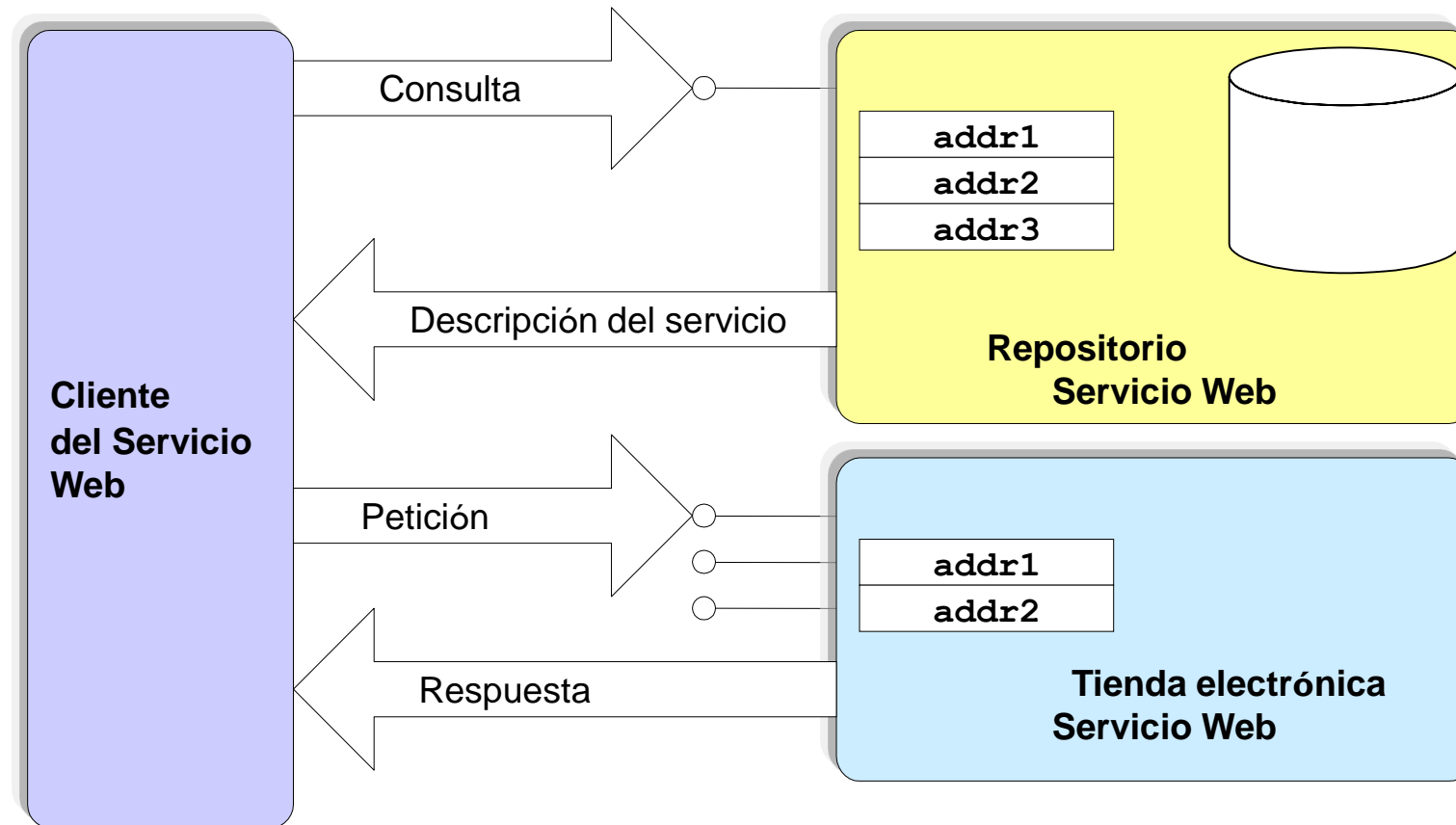
Arquitectura de Servicios Web

16



Arquitectura de Servicios Web

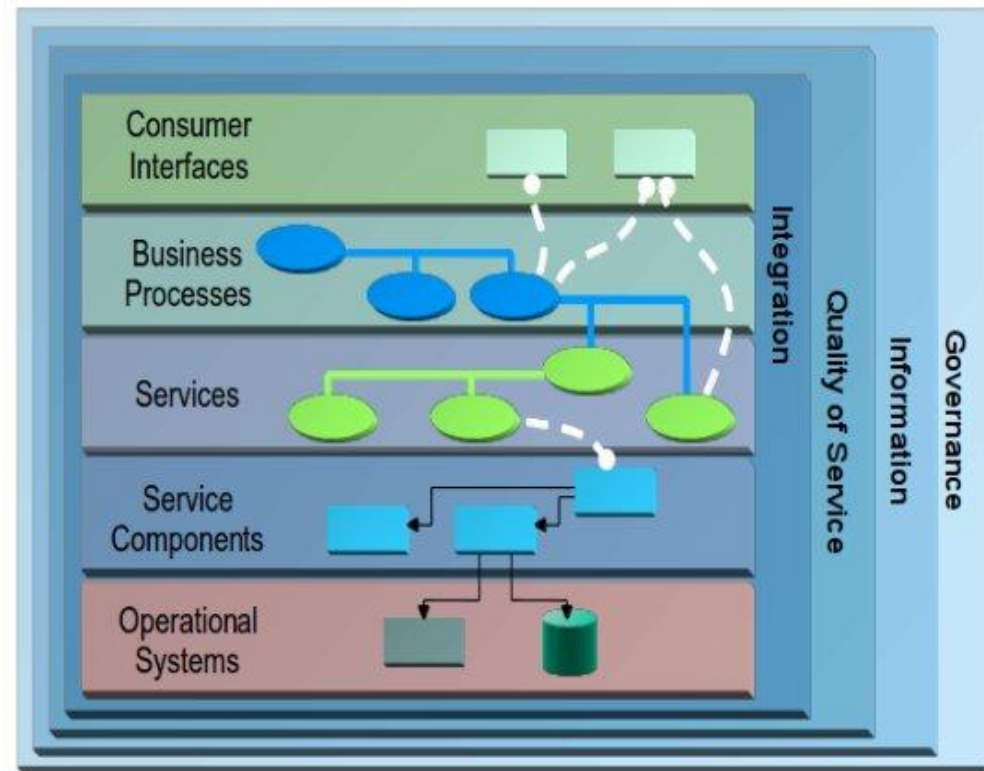
17



Ejemplo de las operaciones que se realizan para acceder un servicio Web que se encuentra en una tienda electrónica.

Arquitectura de servicios organizada en capas

- Razones para el manejo de capas:
 - Composición flexible
 - Reutilización
 - Estandarización funcional en las capas inferiores
 - Personalizar las capas superiores
 - Las políticas pueden variar por capa



Principios SOA

- Contrato de servicios estandarizado
- Bajo acoplamiento
- Abstracción
- Reutilización
- Autonomía
- Sin estado
- Descubrimiento
- Composabilidad

XML

- XML es una forma de representar datos. Es un estándar aceptado universalmente por W3C que puede ser utilizado para representar cualquier información estructurada.
- Lenguaje de marcado extensible
- Lenguaje para describir documentos con datos estructurados
- Estándar del sector W3C
- Esquemas XML – XSD
 - Basado en XML
 - Lenguaje para describir tipos de datos

Formato de los mensajes XML

Documento XML

```
<?xml version="1.0" encoding="UTF-16" ?>
<ns:student xmlns:ns="xyzzzy:abc">
  <name>David &#83;mith</name>
  <age>38</age>
</ns:student>
```

XML Schema

```
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="xyzzzy:abc"
  targetNamespace="xyzzzy:abc"
>
  <xsd:complexType name="person" >
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string"/>
      <xsd:element name="age" type="xsd:double" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="student" type="tns:person" />
</xsd:schema>
```



SOAP

SOAP

- Protocolo sencillo basado en XML que se utiliza para intercambiar información estructurada a través de Internet.
- Es utilizado para la comunicación basada en mensajes.
- Estándar del sector.
- Protocolo sencillo basado en XML.
- Admite distintos protocolos y formatos: HTTP, SMTP y MIME.

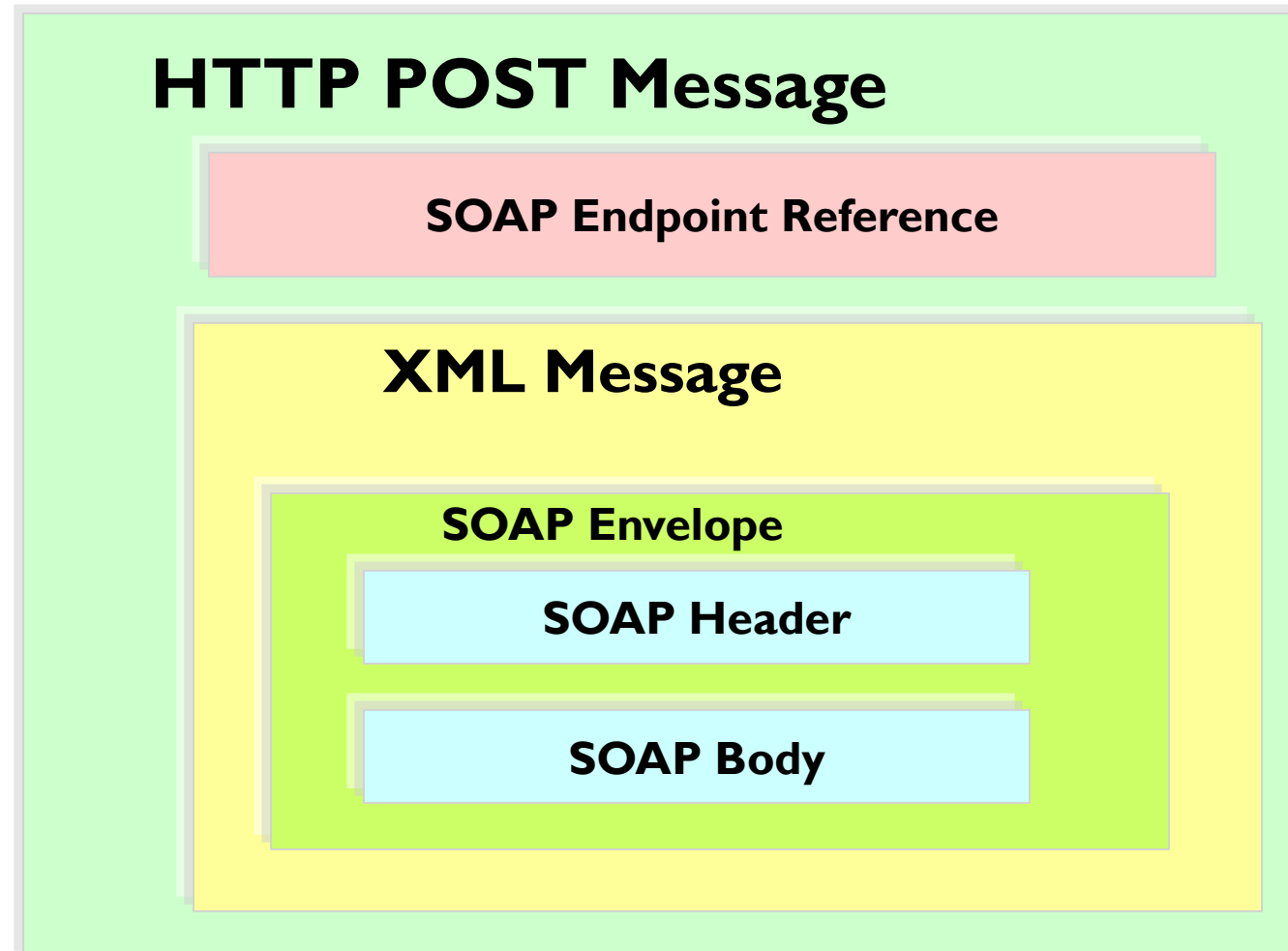
```
HTTP/1.1 500 Internal Server Error
Host: www.borland.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn

<SOAP-ENV:Envelope

  xmlns:SOAP-ENV=
"http://schemas.xmlsoap.org/soap/envelope/"

  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>
        SOAP-ENV:Server
      </faultcode>
      <faultstring>
        Server Error
      </faultstring>
      <detail>
        <d:message xmlns:d="some URI">
          No Symbol found
        </d:message>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
```

SOAP sobre HTTP



SOAP Endpoint Reference

IP Host Address	TCP Port No	Object Endpoint ID
144.122.230.16	80	/ProductCatalog#getPrice

POST /ProductCatalog HTTP/1.0
Host: http://www.srdc.metu.edu.tr
Content-Type: text/xml;
charset="utf-8"
Content-Length: 500
SOAPAction:

``http://www.srdc.metu.edu.tr/ProductCatalog#getPrice``

SOAP Envelope

```
<SOAP-ENV:Envelope
  xmlns="http://schemas.xmlsoap.org/soap/envelope/">

  < SOAP-ENV:Header>
    ...
  </ SOAP-ENV:Header>

  < SOAP-ENV:Body>
    ...
  </ SOAP-ENV:Body>
  ...
</ SOAP-ENV: Envelope>
```


SOAP Body

```
<SOAP-ENV:Header />
```

```
<SOAP-ENV:Body>
```

Method Name



```
<s:getPrice xmlns:s="http://www.srdc.metu.edu.tr/ProductCatalog">
```

```
<product_name xsi:type="xsd:string">Palm Pilot</product_name>
```

```
</s:getPrice>
```

```
</SOAP-ENV:Body>
```

```
</SOAP-ENV:Envelope>
```

Input Parameter



SOAP Response

HTTP/1.0 200 OK

Content-Type: text/xml; charset="utf-8"

Content-Length: 400

<SOAP-ENV:Envelope

xmlns:SOAP-ENV=" http://schemas.xmlsoap.org/soap/envelope/"

SOAP-ENV:encodingStyle=" http://schemas.xmlsoap.org/soap/encoding"

xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"

xmlns:xsd="http://www.w3.org/1999/XMLSchema">

<SOAP-ENV:Header />

<SOAP-ENV:Body>

<s:getPriceResponse xmlns:s="http://www.srdc.metu.edu.tr/ProductCatalog">

<product_price xsi:type="xsd:float">2000.</product_price>

</s:getPriceResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>



Standard Suffix



WSDL

Descripción del servicio

- Para lograr una llamada a un servicio utilizando SOAP se necesita saber
 - El URL
 - Los valores para las extensiones HTTP
 - El formato exacto del mensaje
- WSDL es un lenguaje de descripción de servicios escritos en XML
 - Describe los detalles de invocación al servicio

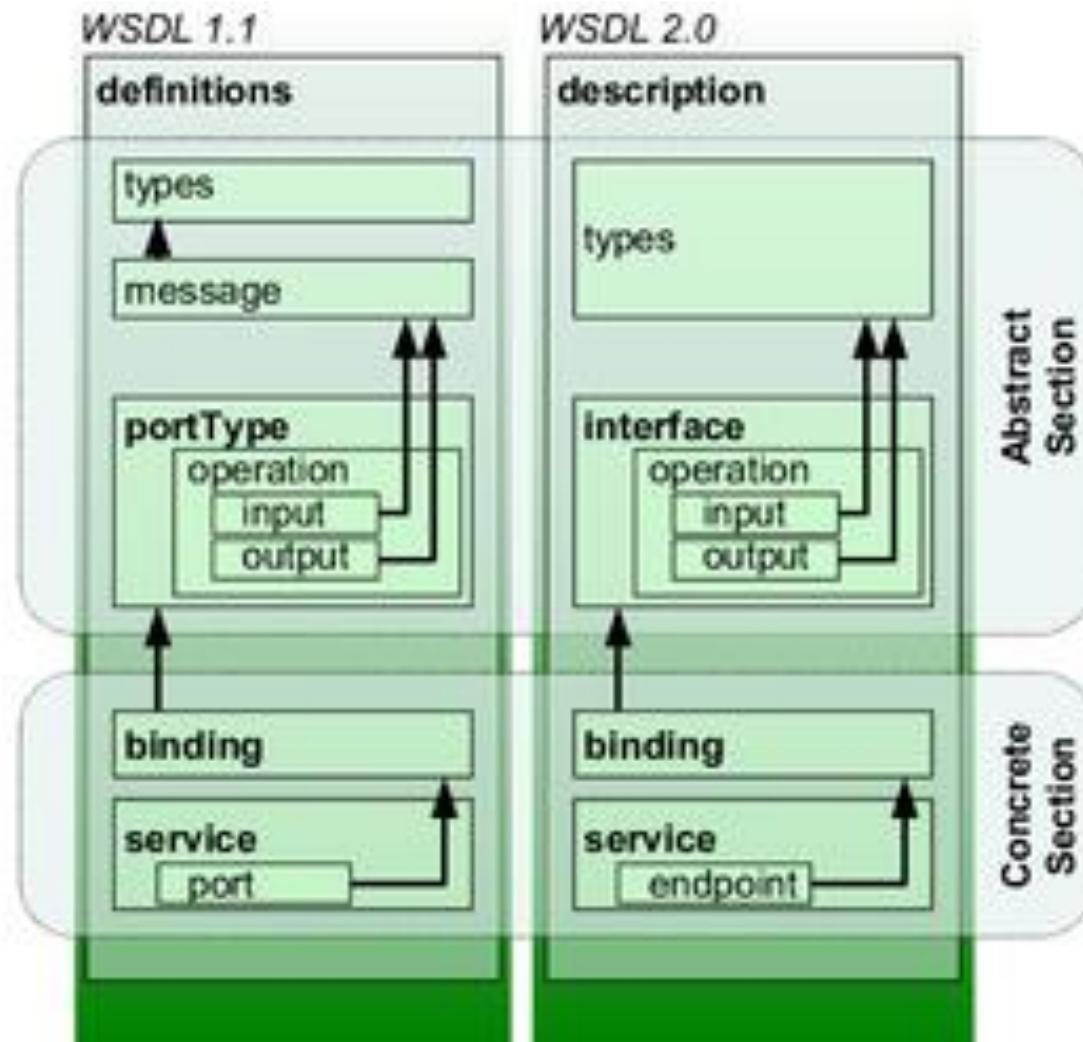
WSDL

- WSDL describe servicios de red usando una gramática XML.
- Mientras que SOAP especifica la comunicación entre un solicitante y un proveedor, WSDL describe los servicios ofrecidos por el proveedor.
- Un documento de WSDL cumple un papel similar a un archivo IDL en CORBA o al Interfaz Remoto en Java RMI.

WebServices Description Language

- Es una gramática XML, orientada a describir en forma estructurada, la funcionalidad de un Web Service y la forma en que esa funcionalidad se hace disponible.
- Describe un servicio, como una colección de “communication endpoints” (puertos) capaces de intercambiar mensajes.
- Cada port tiene un definición abstracta (port type) y una definición concreta (binding).
- Permite describir en forma abstracta operaciones y mensajes, prescindiendo de las especificaciones de protocolo y tipos de datos.
- Vincula las descripciones abstractas a una implementación concreta de protocolos y tipos de datos, permitiendo el reuso de las definiciones abstractas.
- Es extensible tanto en lo que respecta a tipos de datos (XSD) como a protocolos y formatos de mensajes.
- Provee documentación sobre el servicio que describe

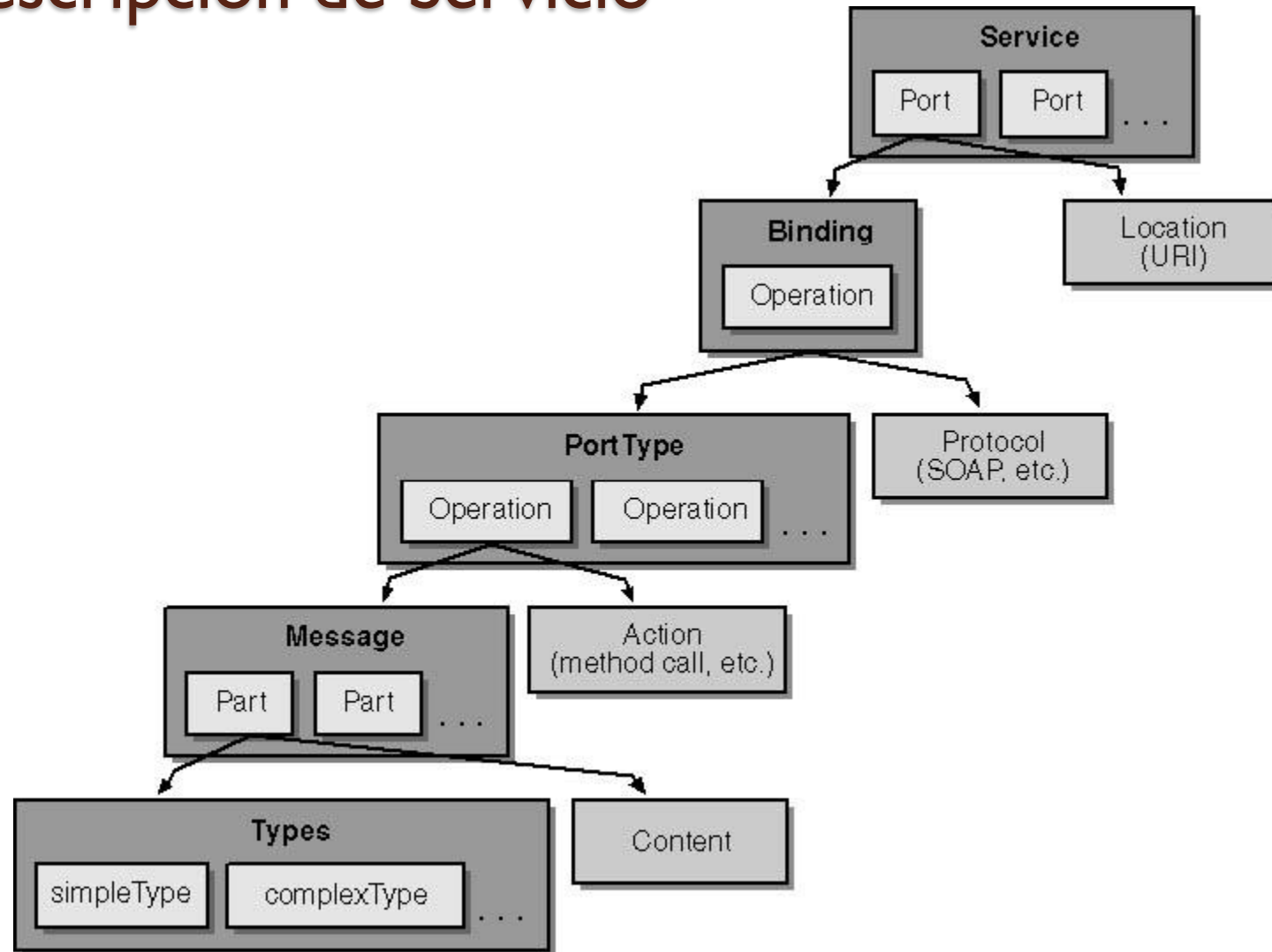
Versiones de WSDL



Componentes de una Descripción de Servicio

- **Service.** Conjunto de “ports” relacionados que implementan el servicio.
- **Port.** “Port type” + “Binding”
Es una descripción abstracta de una acción soportada por el servicio. Cada operación se corresponde a un mensaje de input o de output
- **Port type.** Colección de “operations” o “signatures” de los métodos que definen el intercambio ordenado de los mensajes.
- **Bindings.** especifica los protocolos que usa cada “port” y el “encoding”.
- **Message.** Descripción de los datos que van a ser transmitidos. Son una colección de “data values” de un tipo particular (utilizando XML Schema como mecanismo de tipación).

Componentes de una Descripción de Servicio

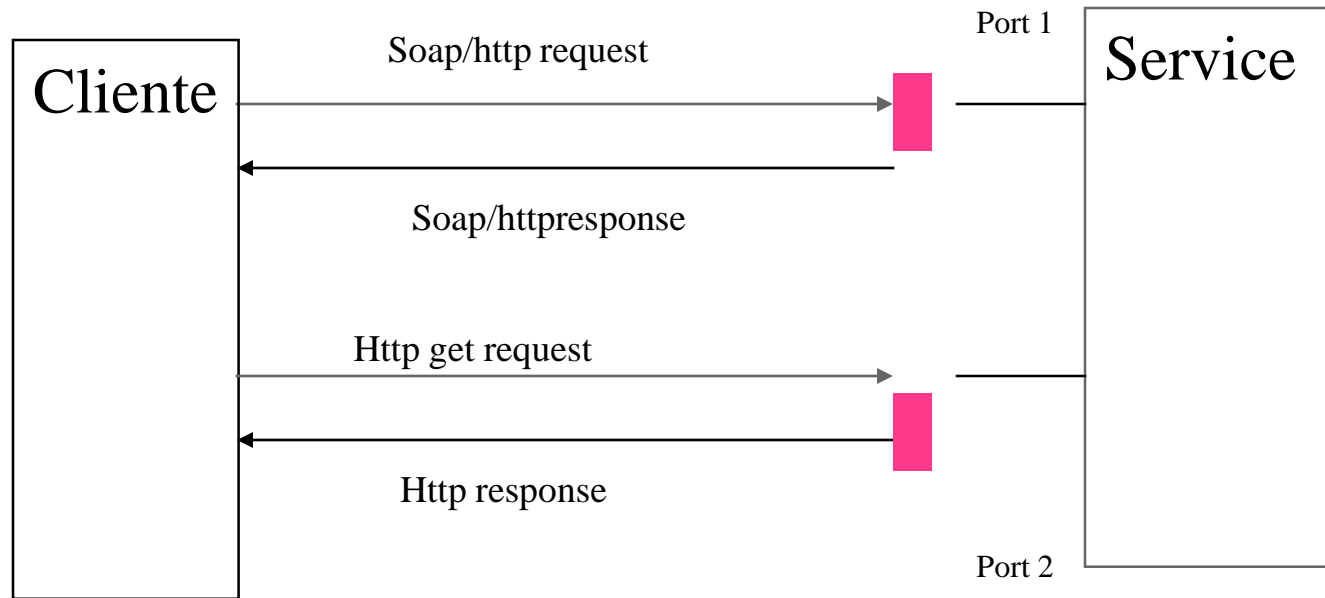


Descripción de un Servicio

1. Data types
 - `<wsdl:type/>`
2. Messages
 - `<wsdl:message/>`
3. Interfaces
 - `<wsdl:portType/>`
4. Services
 - `<wsdl:binding/>`
 - `<wsdl:service/>`

Prefijos para referenci de espacios de nombres

Prefijo	URI Namespace	Definición
wSDL	<code>http://schemas.xmlsoap.org/wSDL/</code>	WSDL namespace for WSDL framework
soap	<code>http://schemas.xmlsoap.org/wSDL/soap/</code>	WSDL namespace for WSDL SOAP binding
http	<code>http://schemas.xmlsoap.org/wSDL/http/</code>	WSDL namespace for WSDL HTTP GET & POST binding
Mime	<code>http://schemas.xmlsoap.org/wSDL/mime/</code>	WSDL namespace for WSDL MIME binding
soapenc	<code>http://schemas.xmlsoap.org/soap/encoding/</code>	Encoding namespace as defined by SOAP 1.1
soapenv	<code>http://schemas.xmlsoap.org/soap/envelope/</code>	Envelope namespace as defined by SOAP 1.1
xsi	<code>http://www.w3.org/2000/10/XMLSchema-instance</code>	Instance namespace as defined by XSD
xsd	<code>http://www.w3.org/2000/10/XMLSchema</code>	Schema namespace as defined by XSD
tns	varios	El "this namespace" (tns) se usa como convención para referirse al documento actual



Definición de tipos y espacios de nombres en WSDL

Uso de “import” para importar una referencia de un tipo de definición:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="HelloWorldDescription"
  targetNamespace="urn:HelloWorld"
  xmlns:tns="urn:HelloWorld"
  xmlns:types="urn:MyDataTypes"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:import namespace="urn:MyDataTypes"
      location="telephonenumber.xsd"/>
</wsdl:definitions>
```

Describiendo la interface del Webservice

```
<definitions ...>
  <wsdl:message name="sayHello_IN">
    <part name="name" type="xsd:string" />
  </wsdl:message>
  <wsdl:message name="SayHello_Out">
    <part name="greeting" type="xsd:string" />
  </wsdl:message>
  <wsdl:portType name="HelloWorldInterface">
    <wsdl:operation name="SayHello">
      <wsdl input message="tns:sayHello_IN" />
      <wsdl output message="tns:sayHello_Out" />
    </wsdl:operation>
  </wsdl:portType>
  ...
</definitions>
```

- Hay 4 tipos de operaciones posibles: input, output, input output, output input

Haciendo el “binding” con la interface del WebService

```
<wsdl:binding name="HelloWorldBinding"
              type="tns:HelloWorldInterface">
  <!-- define el protocolo de transporte y el estilo del mensaje SOAP-->
  <soap:binding style="rpc"  puede ser "rpc" o "document"
                transport="http://schemas.xmlsoap.org/soap/http"/>
  <!-- define el valor del header soapAction cuando HTTP es usado-->
  <wsdl:operation name="sayHello">
    <soap:operation soapAction="urn:Hello" />
    <!-- especifica si las partes del mensaje serán condif o literales-->
    <wsdl:input>
      <soap:body use="encoded" namespace="..."
                encodingStyle="..." />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="encoded" namespace="..."
                encodingStyle="..." />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
```

Binding de WSDL con HTTP-GET

```
<wsdl:binding name="HelloWorldBinding"
              type="HelloWorldInterface">
  <http:binding verb="GET"/>
  <wsdl:operation name="sayHello">
    <http:operation location="sayHello" />
    <wsdl:input>
      <http:urlEncoded />
    </wsdl:input>
    <wsdl:output>
      <mime:content type="text/plain" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
```

- La invocación podría ser <http://localhost/sayHello?name=John>
- La respuesta es un string de datos en formato MIME.

Herramientas

- Existen varias herramientas y entornos disponibles para generar automáticamente el documento WSDL correspondiente a un WebServices:
 - Membrane
 - Axis
 - VisualStudio.Net
 - IBM Web Services Toolkit
 - Apache Web Services Toolkit
 - Microsoft Soap Toolkit V2 Rc 0.

Observaciones

- WSDL es sin duda el estándar para describir servicios Web.
- Existen otros lenguajes que apuntan a resolver el mismo problema en ambientes más especializados.
- Uno de ellos es ebXML (orientado a transacciones comerciales) www.oasis.org

Desventajas:

- WSDL carece de la posibilidad de especificar una secuencia de las operaciones necesarias en un intercambio de mensajes (por ejemplo, un login previo).

Repositorios de Servicios Web

- <http://neptune.fulton.ad.asu.edu/WSRepository/repository.html>
- <http://www.visualwebservice.com/>
- <http://www.dneonline.com/calculator.asmx>

Temas de Investigación sobre Servicios Web

