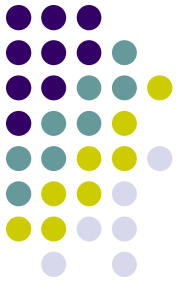
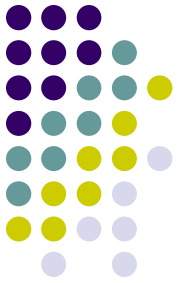

Web Services



Agenda



- Web Services
- UDDI
- WSDL
- SOAP



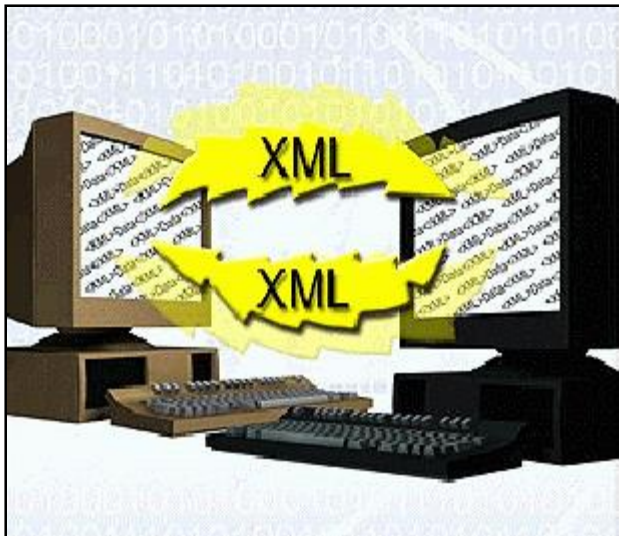
Web Services



Web Services

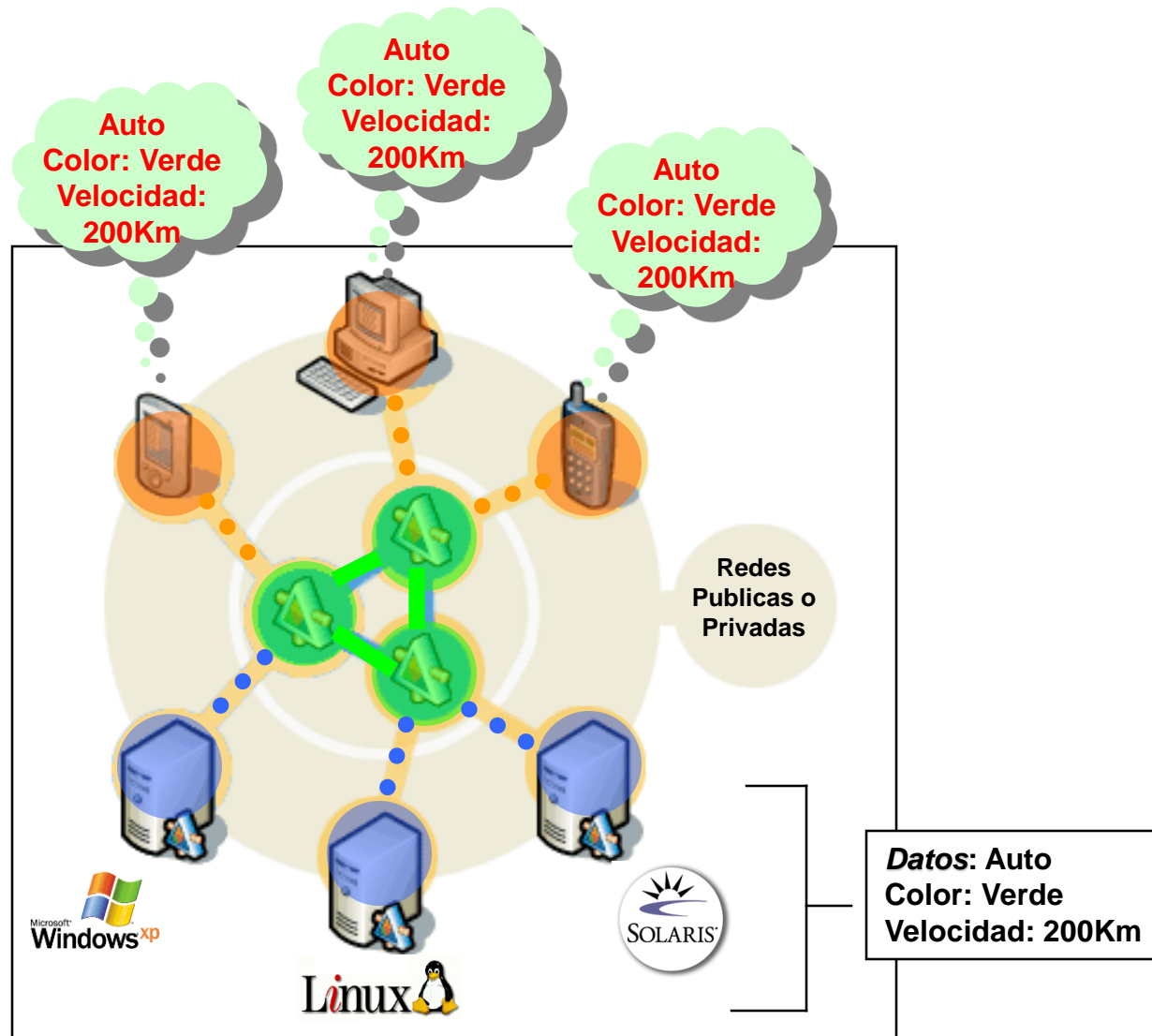
Son aplicaciones o parte de ellas independientes del lenguaje de programación y del sistema operativo, que pueden ser fácilmente publicadas, localizadas e invocadas a través de redes privadas o publicas como Internet.

El concepto de Servicio Web se apoya en el XML



```
<?xml version="1.0" ?>  
- <AUTO>  
  <COLOR>verde</COLOR>  
  <VELOCIDAD>200Km/h</VELOCIDAD>  
  <MOTOR>6 cilindros</MOTOR>  
</AUTO>
```







Ventajas

Interoperabilidad: los Servicios Web aseguran una interoperabilidad completa entre sistemas gracias a que la comunicación se lleva a cabo en XML.

Fácil implementación: crear un Servicio Web es algo muy simple en todas las plataformas.

Accesibilidad: los Servicios Web pueden ser completamente descentralizados y distribuidos sobre Internet y accedidos a través de una gran variedad de dispositivos.

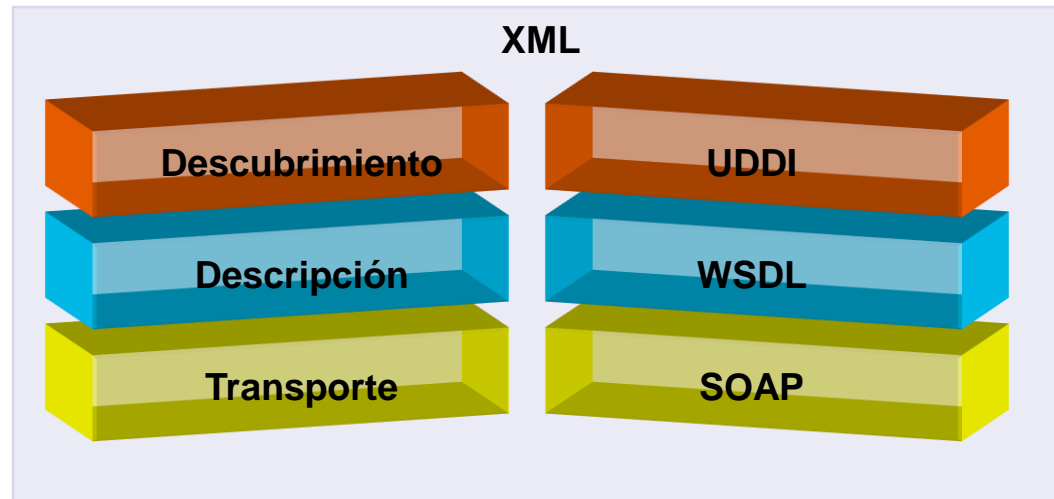
Especificaciones universalmente aceptadas: los Servicios Web se basan en especificaciones estándar para el intercambio de datos, mensajería, búsqueda, descripción y coordinación de los procesos.

El software como un servicio, en lugar de un producto empaquetado

Estándares



Los servicios Web se registran, anuncian e invocan utilizando los siguientes servicios y protocolos; la base de los mismos es el XML





Estándares - UDDI



Universal Description, Discovery and Integration: es un protocolo para describir Servicios Web. Este estándar permite a las empresas registrarse en un directorio de Internet o local para anunciar sus servicios, de tal forma que las compañías puedan encontrarse unas a otras y realizar transacciones en la Web. El proceso de registro y consultas se realiza utilizando mecanismos basados en XML y HTTP. Es correcto comparar a UDDI con la guía telefónica o las páginas amarillas.



Estándares - WSDL



Web Service Description Language: es un lenguaje para la descripción de Servicios Web. Gracias a WSDL se puede conocer cuál es la funcionalidad de un Servicio Web, cómo utilizar sus servicios, cómo invocarlos y cuál es la respuesta esperada a los mismos si es que la hay. También WSDL permite definir cuál es la estructura del mensaje SOAP; es decir que tipos de datos contiene, longitud, etc.

Estándares - SOAP



Simple Object Access Protocol: es el principal componente de los Servicios Web, dado que es un protocolo para el intercambio de información entre proveedores y consumidores de Servicios Web. La comunicación dentro de los Servicios Web se realiza pura y exclusivamente a través del envío de mensajes SOAP basados en XML.

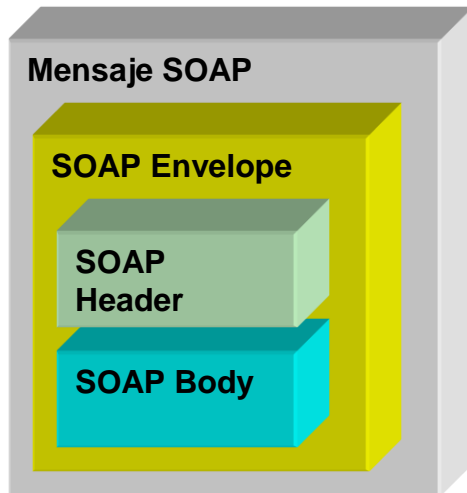


Estándares - SOAP

Características generales de SOAP

- Protocolo simple de utilizar.
- Independiente de la plataforma y del lenguaje de programación.
- Extensible.
- Trabaja con la infraestructura de Internet. No hay que hacer configuraciones especiales en routers, firewall, proxy servers, etc.
- Varios proveedores y organizaciones han soportado la implementación de SOAP. W3C ha definido a SOAP como una especificación, y ninguna organización tendrá el control sobre SOAP, por lo tanto se ha convertido en otro estándar de Internet, tales como HTML y XML.

Estándares - SOAP



Un mensaje SOAP consta de la siguiente estructura.

SOAP Envelope: es el elemento de mayor nivel dentro de la estructura de un mensaje y es un encabezado que identifica a un mensaje SOAP.

SOAP Header: es utilizado como una forma de extender el contenido de un mensaje, sin afectar a las partes que se están comunicando.

SOAP Body: es el lugar donde reside el mensaje propiamente dicho con los datos correspondientes.

Estándares - SOAP



<SOAP-ENV:Envelope

SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">

<SOAP-ENV:Body>

<ns1:getDVDs xmlns:ns1="http://dvdonline-interface">

<in0 xsi:type="xsd:string">La caza del Octubre Rojo</in0>

<in1 xsi:type="xsd:int">10</in1>

</ns1:getDVDs>

</SOAP-ENV:Body>

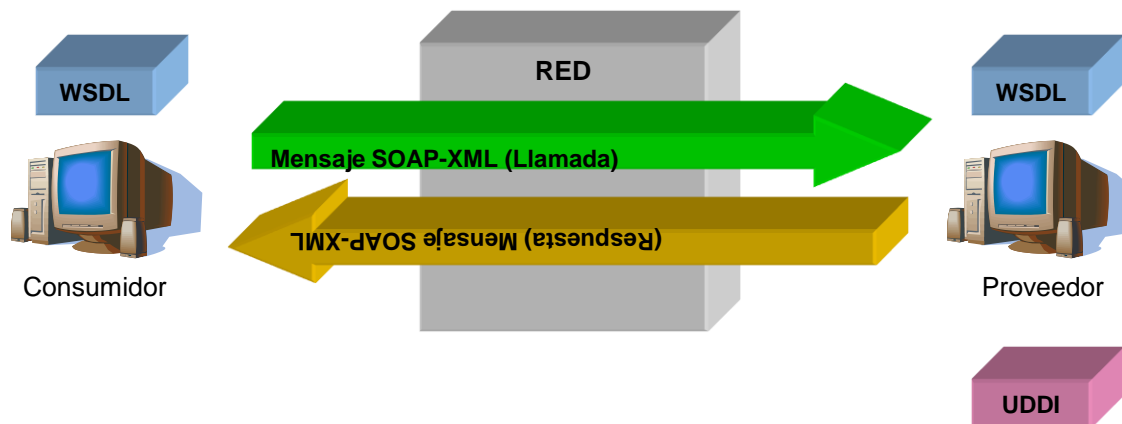
</SOAP-ENV:Envelope>



Tipos de mensaje SOAP

SOAP trabaja con dos tipos de mensajes: **RPC** y **Document**.

SOAP-RPC: este tipo de mensaje está basado en el modelo de llamada a procedimientos remotos RPC, donde una aplicación (cliente) realiza una llamada a una función (método) que no reside en el mismo ámbito de la aplicación, sino que por ejemplo se encuentra en otra computadora (servidor) en la red. Una vez procesada dicha llamada, el cliente recibe la respuesta correspondiente.



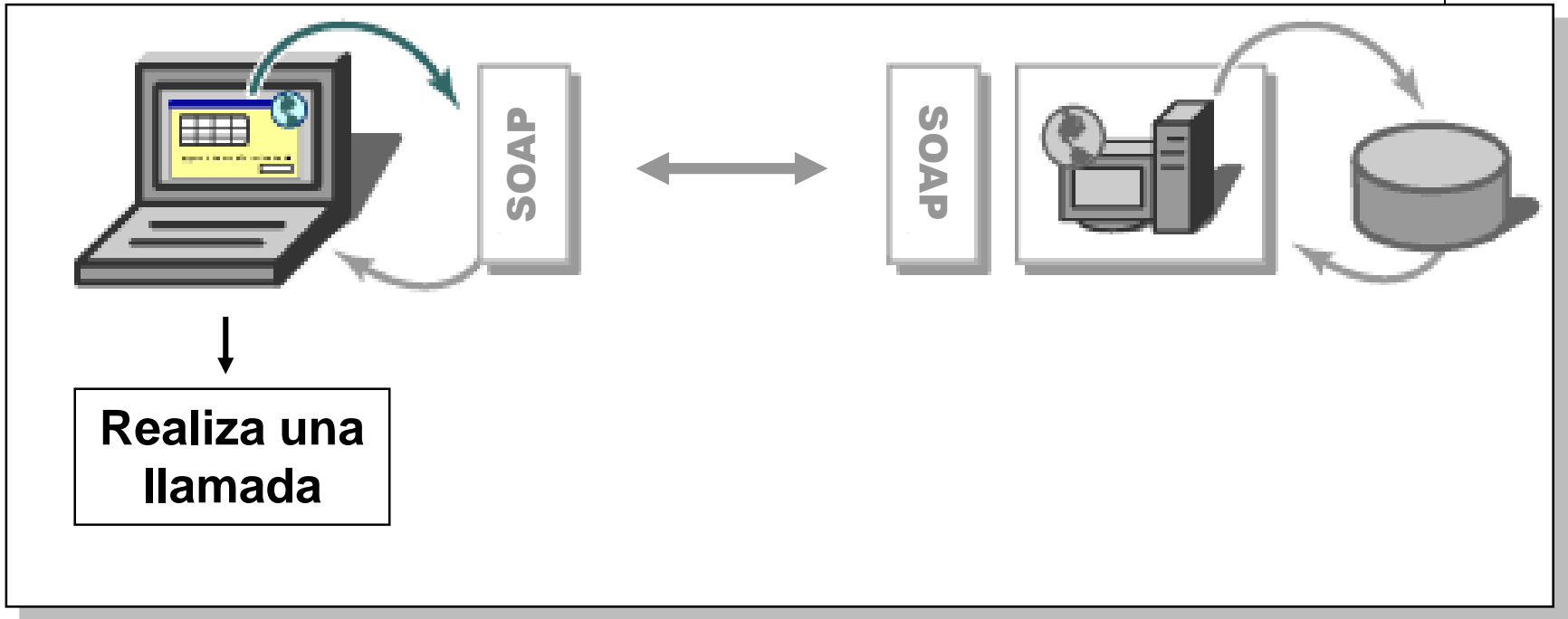
SOAP - RPC



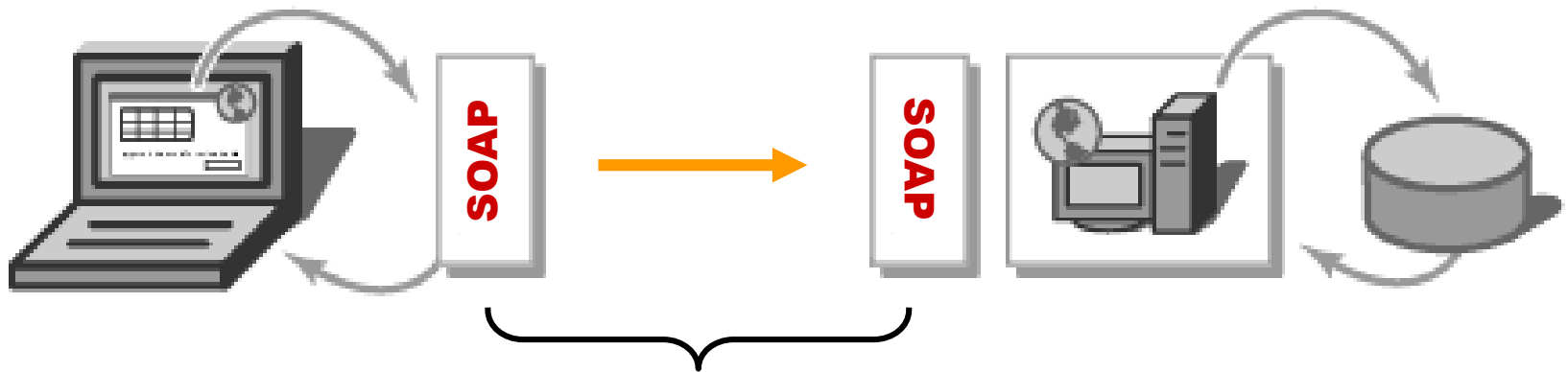
Lo importante de este tipo de mensajes (RPC) es que el cliente sólo recibe la respuesta a su llamada (sincrónica) y no realiza ningún tipo de procesamiento o cálculo. Todo este proceso queda a manos del servidor y se dice que el procesamiento es remoto (o distribuido).

Este tipo de tecnología no es nuevo en el mundo de la informática (RMI, CORBA, DCOM, .NET Remoting), pero el punto fuerte que posee en el ámbito de los Servicios Web es su posibilidad de interoperatividad entre sistemas dispares. Ej.: una aplicación ejecutándose en Windows, podría llamar a una función de otra aplicación corriendo en Linux, todo esto utilizando Servicios Web y XML.

SOAP - RPC

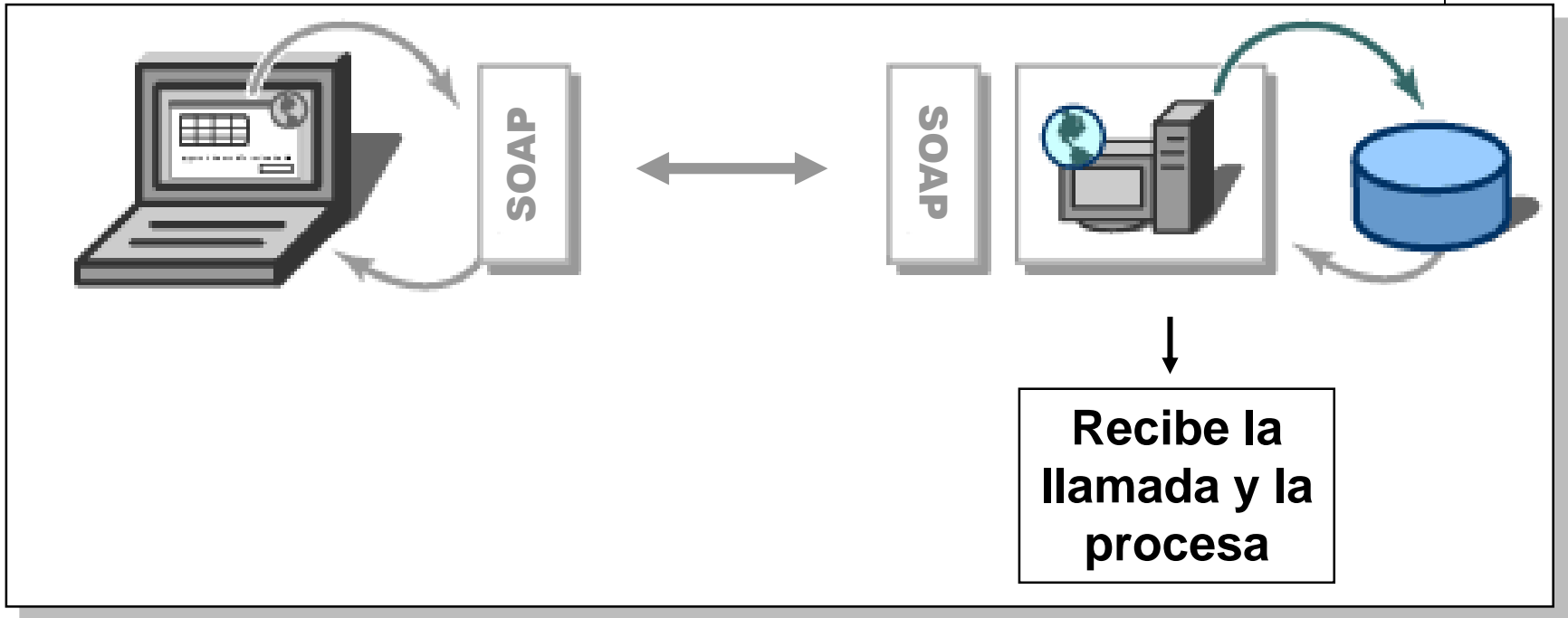


SOAP - RPC

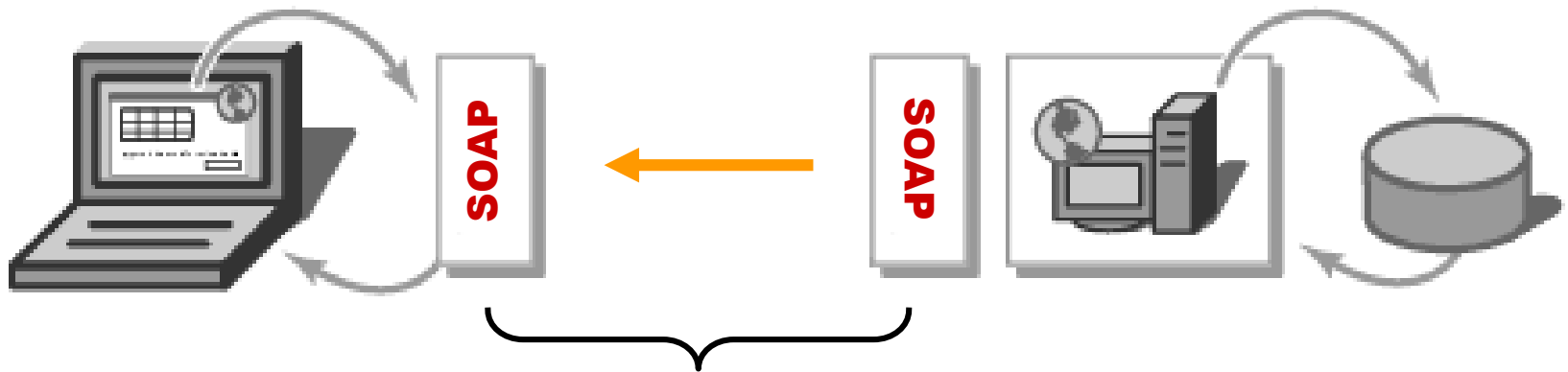


```
<buscarObjeto>  
  <oid xsi:type="xsd:string">4507</oid>  
  <tipo xsi:type="xsd:string">auto</tipo>  
</buscarObjeto>
```

SOAP - RPC

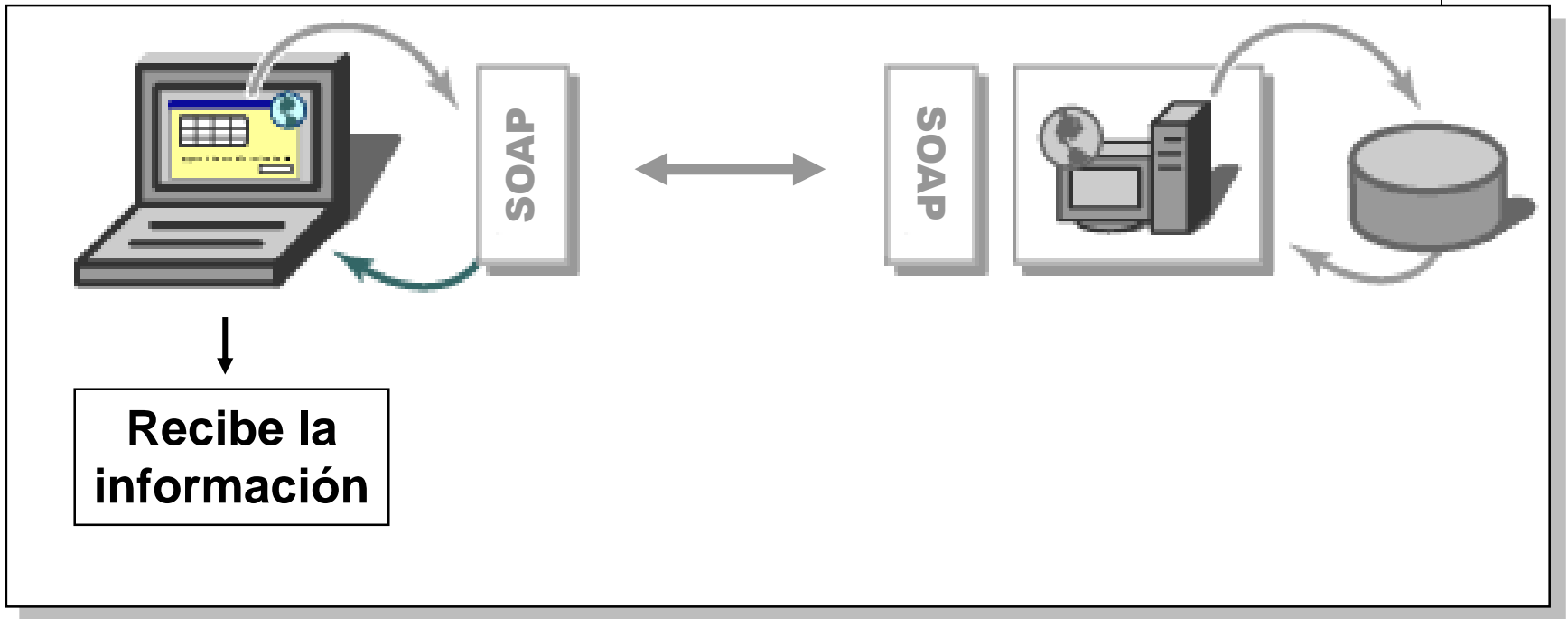


SOAP - RPC



```
<Auto>  
<item xsi:type="xsd:string">BMW-99</item>  
<item xsi:type="xsd:integer">1234678</item>  
<item xsi:type="xsd:date">12/12/2003</item>  
</Auto>
```

SOAP - RPC



SOAP - RPC



Ejemplo de llamada SOAP-RPC

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns1:getDVDs xmlns:ns1="http://dvdonline-interface">
      <in0 xsi:type="xsd:string">Drama</in0>
      <in1 xsi:type="xsd:int">10</in1>
      <in2 xsi:type="xsd:string">FSD-4235634 </in2>
    </ns1:getDVDs>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP - RPC



Ejemplo de respuesta SOAP-RPC

```
<?xml version="1.0" encoding="UTF-8"?>
  <SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP-ENV:Body>
      <ns1:getDVDsResponse
        SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        xmlns:ns1="http://dvdonline-interface">
        <getDVDsResult xsi:type="SOAP-ENC:Array"
          SOAP-ENC:arrayType="xsd:string[1]"
          xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
          <item xsi:type="xsd:string">La caza al Octubre Rojo</item>
        </getDVDsResult>
      </ns1:getDVDsResponse>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

SOAP - Document



SOAP-Document: este tipo de mensajes tiene la particularidad de no contar con reglas de codificación para el intercambio de información (contrariamente al modelo de trabajo RPC). Esto permite enviar y recibir cualquier tipo de información en XML arbitrariamente, brindando más libertad de trabajo.

Las partes se tienen que poder de acuerdo en el formato del mensaje y en la validez del mismo mediante un esquema.



SOAP - Document

Ejemplo de mensaje SOAP Document

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <dvdos:order xmlns:dvdos="http://dvdonline.com" dvdos:ID="_a89-sd9sd-334">
      <dvdos:customer>
        <dvdos:name>David Fraser</dvdos:name>
        <dvdos:address>123 Main St.</dvdos:address>
        <dvdos:city>Atlanta</dvdos:city>
        <dvdos:state>GA</dvdos:state>
        <dvdos:zip>30324</dvdos:zip>
        <dvdos:shipper dvdos:accountId="_123-3443-223" dvdos:name="Special"/>
      </dvdos:customer>
      <dvdos:dvd dvdos:ID="34werer2">
        <dvdos:title>Big Blue Rocks</dvdos:title>
        <dvdos:artist>Donald Cameron</dvdos:artist>
        <dvdos:releaseDate>12-12-2001</dvdos:releaseDate>
        <dvdos:listPrice>12.32</dvdos:listPrice>
        <dvdos:price>3.33</dvdos:price>
      </dvdos:dvd>
    </dvdos:order>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


Desventajas



- Para realizar transacciones no pueden compararse en su grado de desarrollo con los estándares abiertos de computación distribuida como CORBA, RMI, etc.
- Su rendimiento es bajo si se compara con otros modelos de computación distribuida, tales como RMI, CORBA, o DCOM. Es uno de los inconvenientes derivados de adoptar un formato basado en texto. Y es que entre los objetivos de XML no se encuentra la concisión ni la eficacia de procesamiento.
- Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear o auditar la comunicación entre programas a ambos lados de la barrera.
- Existe poca información de servicios web para algunos lenguajes de programación
- Gran problema de interoperatividad en algunos casos

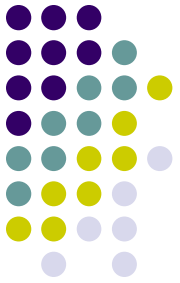
Otros Estándares

WS-I

WS-Reliable

WS-Transaction

WS-Security





Web Services en J2EE

Java permite exponer como Web Services tanto clases comunes (POJOS) como EJB-SB sin estado. Estos últimos tienen la ventaja de contar con todos los servicios ya existentes dentro de un servidor de aplicaciones J2EE.

```
// WS mediante una interfaz del tipo Remote
@WebService
public interface PlaceBidWS {
    public Long addBid(String bidderId, Long itemId, Double bidPrice);
}

@Stateless(name = "PlaceBid")
public class PlaceBidBean implements PlaceBidWS, PlaceBid {
    ...
}
```



Web Services en J2EE

```
import javax.jws.WebService;  
import javax.jws.WebMethod;
```

```
//WS sin la interfaz directamente en el SB
```

```
@WebService
```

```
@Stateless(name = "HolaMundo")
```

```
public class HolaMundoServiceBean implements ..... {
```

```
    @WebMethod
```

```
    public String holaMundoWS() {  
        return "Hola Mundo!!!";
```

```
    }
```

```
}
```



Web Services en J2EE

Tipos de WS soportados por Java

RPC: Adoptado por la mayoría de la implementaciones inicialmente. En la actualidad se esta comenzando a utilizar el modelo Document mas que el RPC.

```
@WebService(targetNamespace = "urn:xxxxxxxxxxxxxxxxx")  
@SOAPBinding(style = SOAPBinding.Style.RPC )
```

Document: Es el nuevo tipo soportado por Java por defecto. Plantea un modelo mejorado que RPC, permitiendo combinar XML con un esquema . Brindando la posibilidad que contar con mas interoperabilidad y dinamismo (Ej: Java - .NET).

```
@WebService(targetNamespace = "urn:xxxxxxxxxxxxxxxxx")  
@SOAPBinding(style = SOAPBinding.Style.DOCUMENT)
```

Web Services en J2EE



Estilos de codificación WS soportados por Java

Encoded: El mensaje es enviado con reglas de codificación externa que indican como decodificar el mensaje (clave-valor). Este tipo de codificación no es soportada por la nueva especificación WS-I.

Literal: el mensaje es enviado como simple XML conjuntamente con un esquema para que las partes se puedan comunicar de manera simple.

¿Preguntas?



Gracias por su atención