

Indice

Hora 1

- 1 Middlewares
- 2 Remote Procedure Call (RPC)
- 3 Remote Object/Method Invocation (ROI/RMI)
- 4 Comunicación orientada a mensajes (MOC)
- 5 Comunicación orientada a streams (streaming)

Hora 2

- 6 Middlewares para la web: Web Services
 - 6.1 XML (eXtensible Markup Language)
 - 6.2 SOAP (Simple Object Access Protocol)
 - 6.3 WSDL (Web Services Description Language)
 - 6.4 UDDI (Universal Description Discovery and Integration)
 - 6.5 Integración
 - 6.6 Otros interfaces a servicios web
 - 6.7 Ejemplos de Web Services
- Referencias

6 Middlewares para la web: Web Services

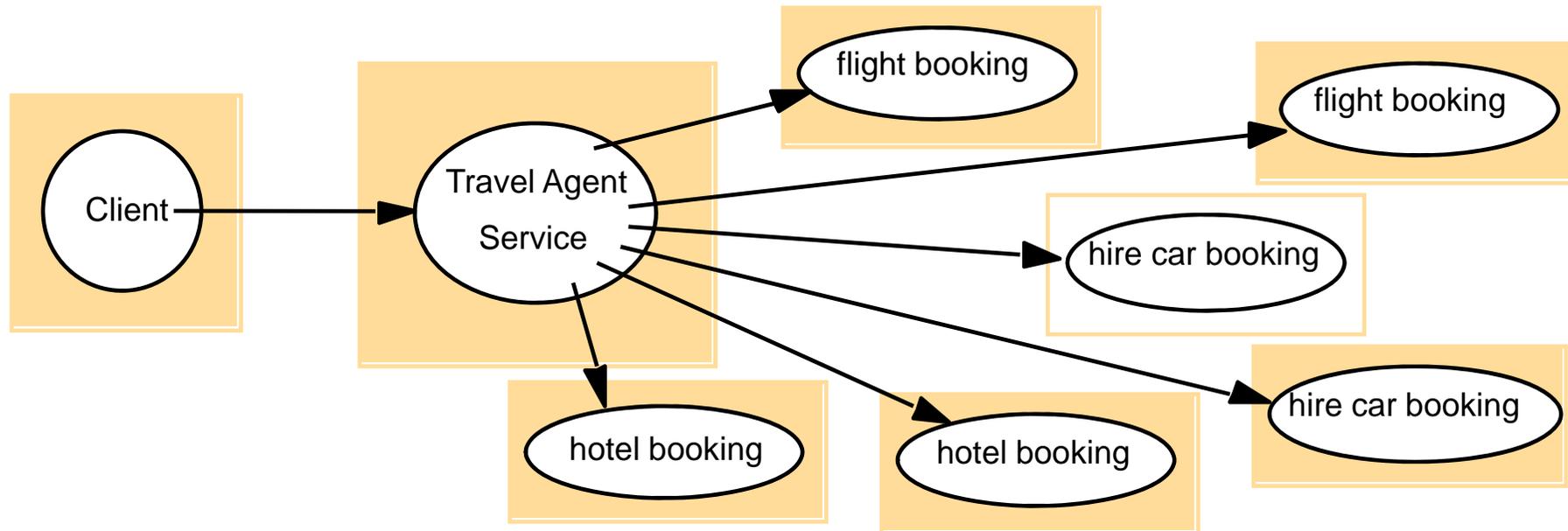
- ▶ La web se ha llenado de aplicaciones.
- ▶ La web está pensada para interacción con humanos.
- ▶ Queremos hacer la web accesible a otros ordenadores y aplicaciones.
 - Un nuevo middleware!
 - Acceso directo, no mediante la simulación de un cliente
- ▶ Ventajas:
 - Protocolos conocidos (HTTP)
 - HTTP es el tráfico que se permite normalmente en las redes (a diferencia de otros servicios en otros puertos)
 - Permite reutilizar infraestructura existente (servidor web)

Web Services

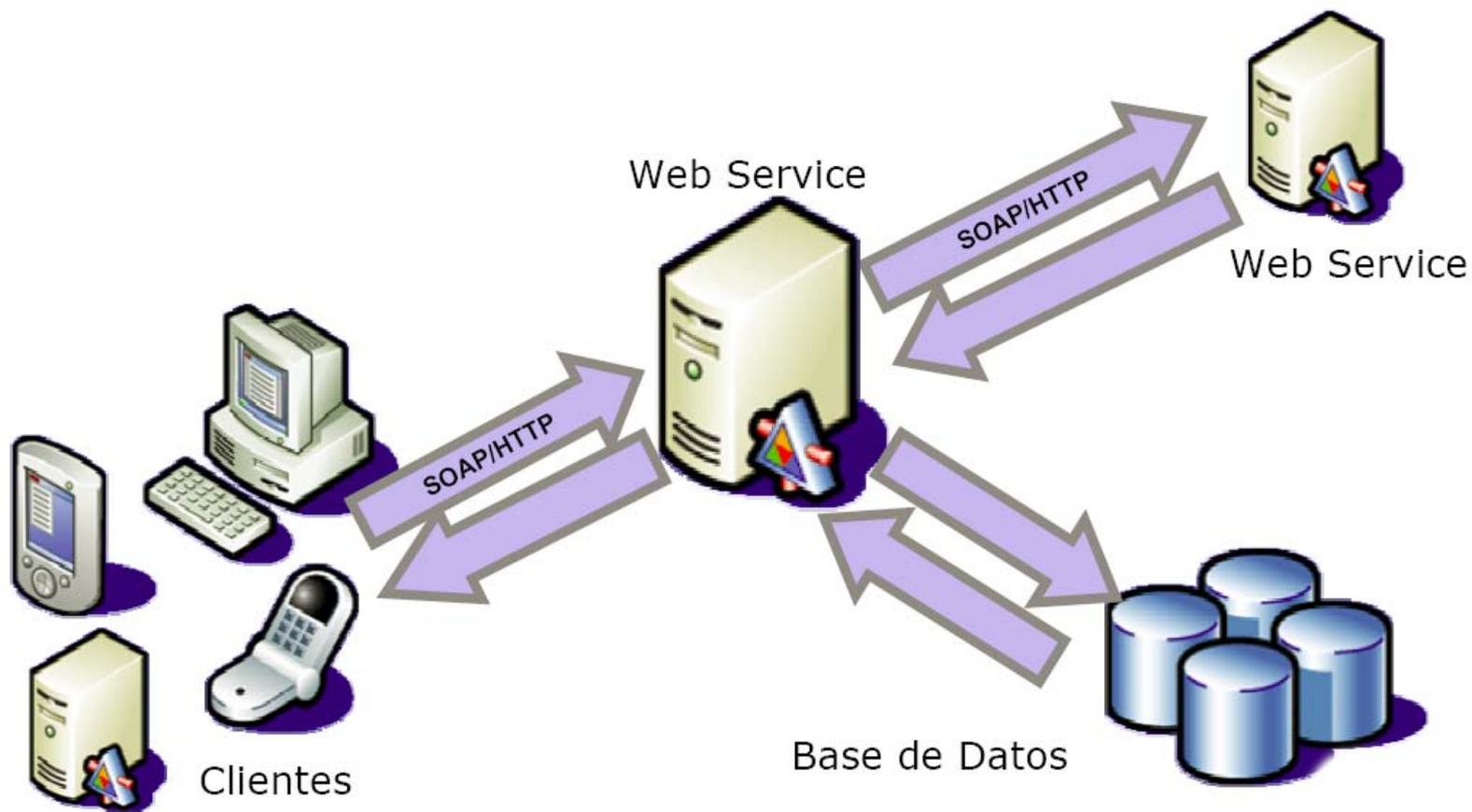
- ▶ ¿Qué es un servicio Web?
 - *Una aplicación Web identificada por un URI, cuyos interfaces y enlaces pueden ser definidos, descritos y descubiertos mediante artefactos XML.*
 - *Soporta la interacción directa con otras aplicaciones Web.*
 - *El interfaz define una colección de operaciones que son accesibles de forma remota usando mensajes XML mediante protocolos estándar de Internet.*
- ▶ Un Web Service (WS) consiste en una colección de operaciones que pueden ser usadas por un cliente a través de Internet.

Ejemplo de combinación de Web Services

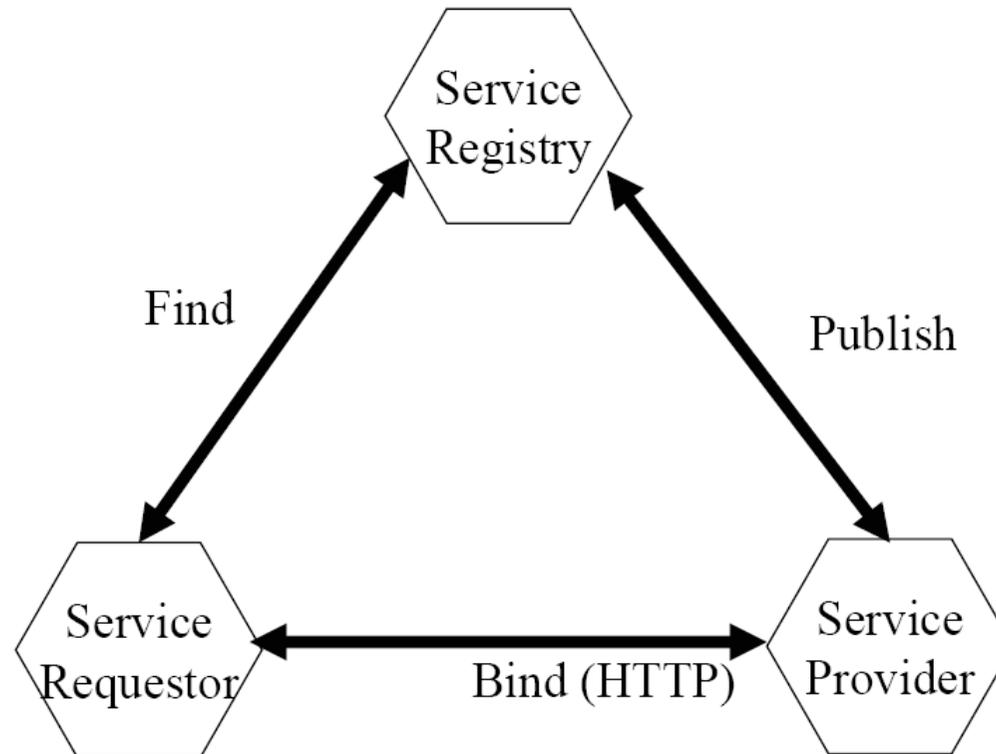
- ▶ Web de agencias de viajes hace uso de interfaces web services que proveen los proveedores de servicio final



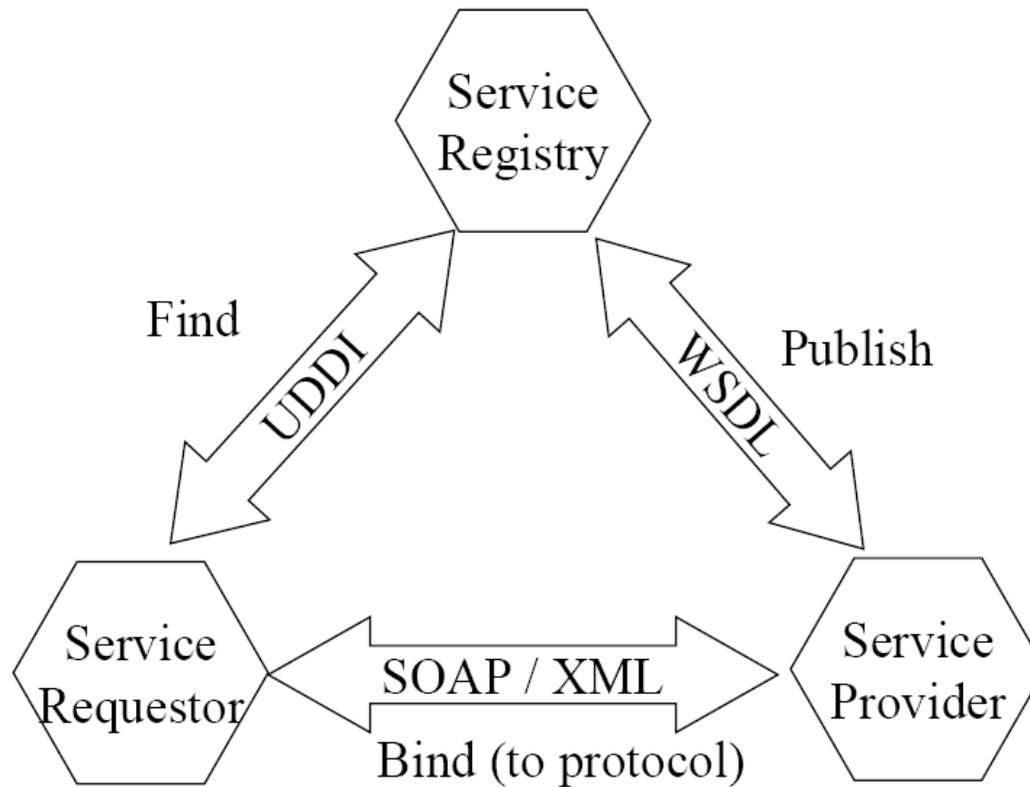
Web Services



Arquitectura orientada a servicios



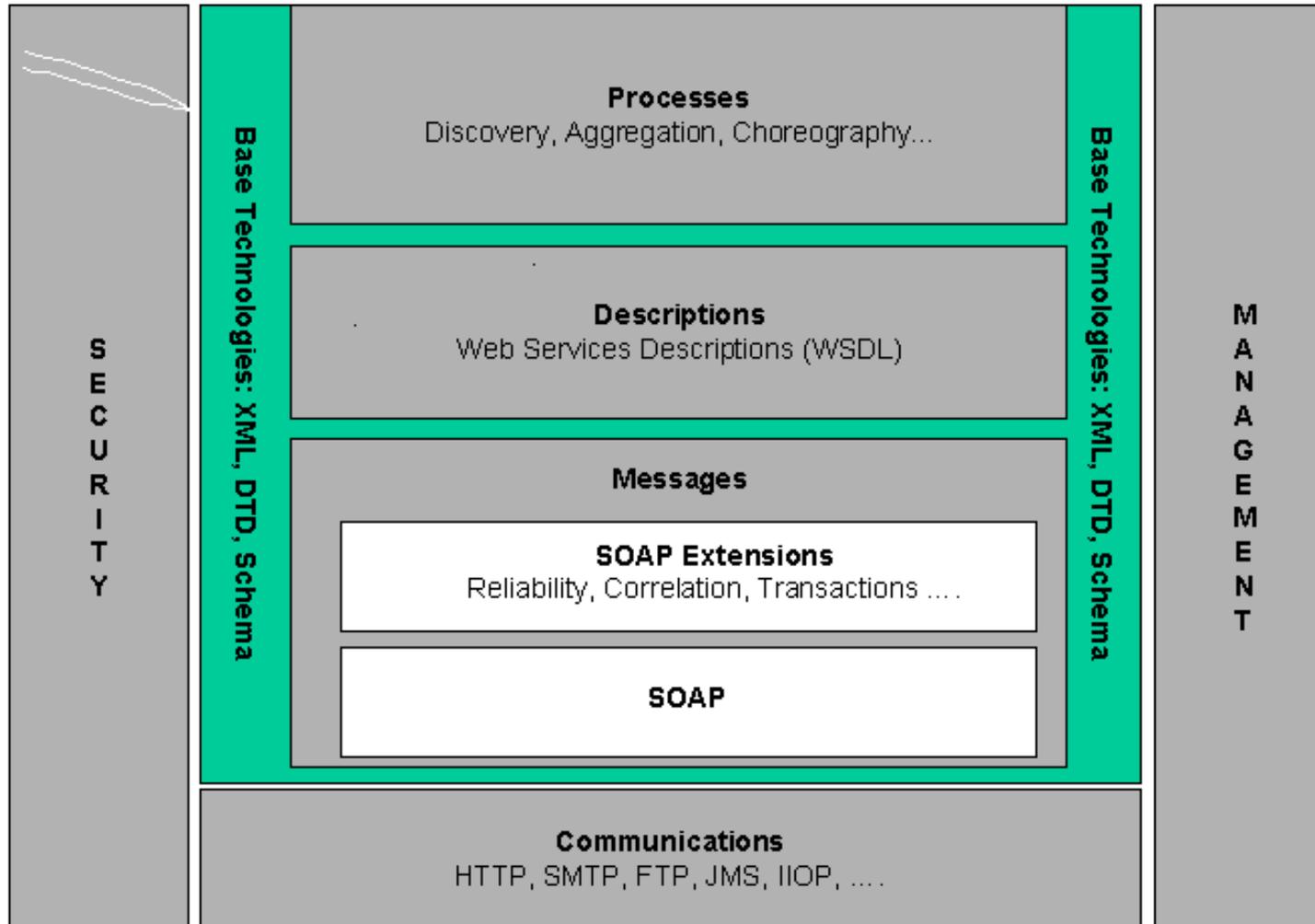
Arquitectura orientada a servicios



3 Componentes

- ▶ XML (eXtensible Markup Language)
 - Lenguaje estándar de representación e intercambio de datos.
- ▶ SOAP (Simple Object Access Protocol)
 - Mecanismo estándar de comunicación. Empaqueta la información y la transporta entre cliente y el proveedor del servicio.
- ▶ WSDL (Web Services Description Language)
 - Meta-lenguaje estándar para describir los servicios ofrecidos.
- ▶ UDDI (Universal Description Discovery and Integration)
 - Mecanismo para registrar y localizar aplicaciones basadas en WS.
- ▶ HTTP: protocolo principal de comunicaciones.

Componentes



6.1 XML (eXtensible Markup Language)

- ▶ XML define el formato textual para representar datos estructurados.
 - `<etiqueta atributo="valor">contenido</etiqueta>`
- ▶ Forma sencilla de describir estructuras de datos complejas y jerárquicas (serializaciones).
 - Separa la definición de estructuras de la presentación de las mismas.
- ▶ Características:
 - Flexibilidad
 - Facilidad de procesado
 - Independencia de plataformas – arquitecturas

XML ejemplo

```
<?xml version="1.0"?>
<catalogo>
  <libro ISBN="1-861003-11-0">
    <titulo>Professional XML</titulo>
    <autor>Didier Martin et al.</autor>
    <editorial>Wrox</editorial>
    <anyo>2000</anyo>
  </libro>
  <libro ISBN="0-07-212646-9">
    <titulo>XML Developer's Guide</titulo>
    <autor>Fabio Arciniegas</autor>
    <editorial>McGraw-Hill</editorial>
    <anyo>2001</anyo>
  </libro>
</catalogo>
```

XML Schema

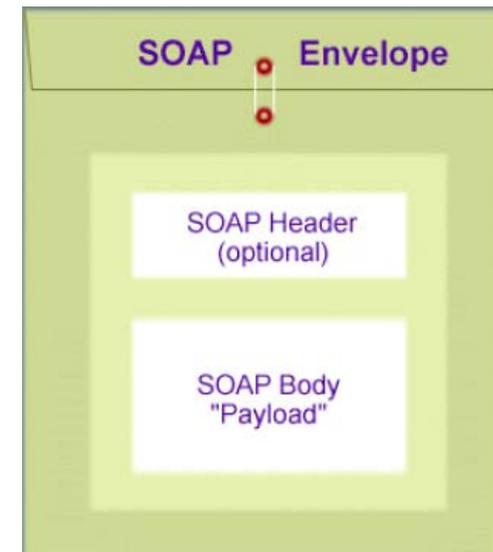
- ▶ Permiten fijar el formato de los datos para su introducción y verificación.

```
<xs:element name="alumno">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="nombre" type="xs:string"/>
      <xs:element name="apellidos" type="xs:string"/>
      <xs:element name="nacim" type="xs:gYear"
        minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute name="dni" type="xs:integer"/>
  </xs:complexType>
</xs:element>
```

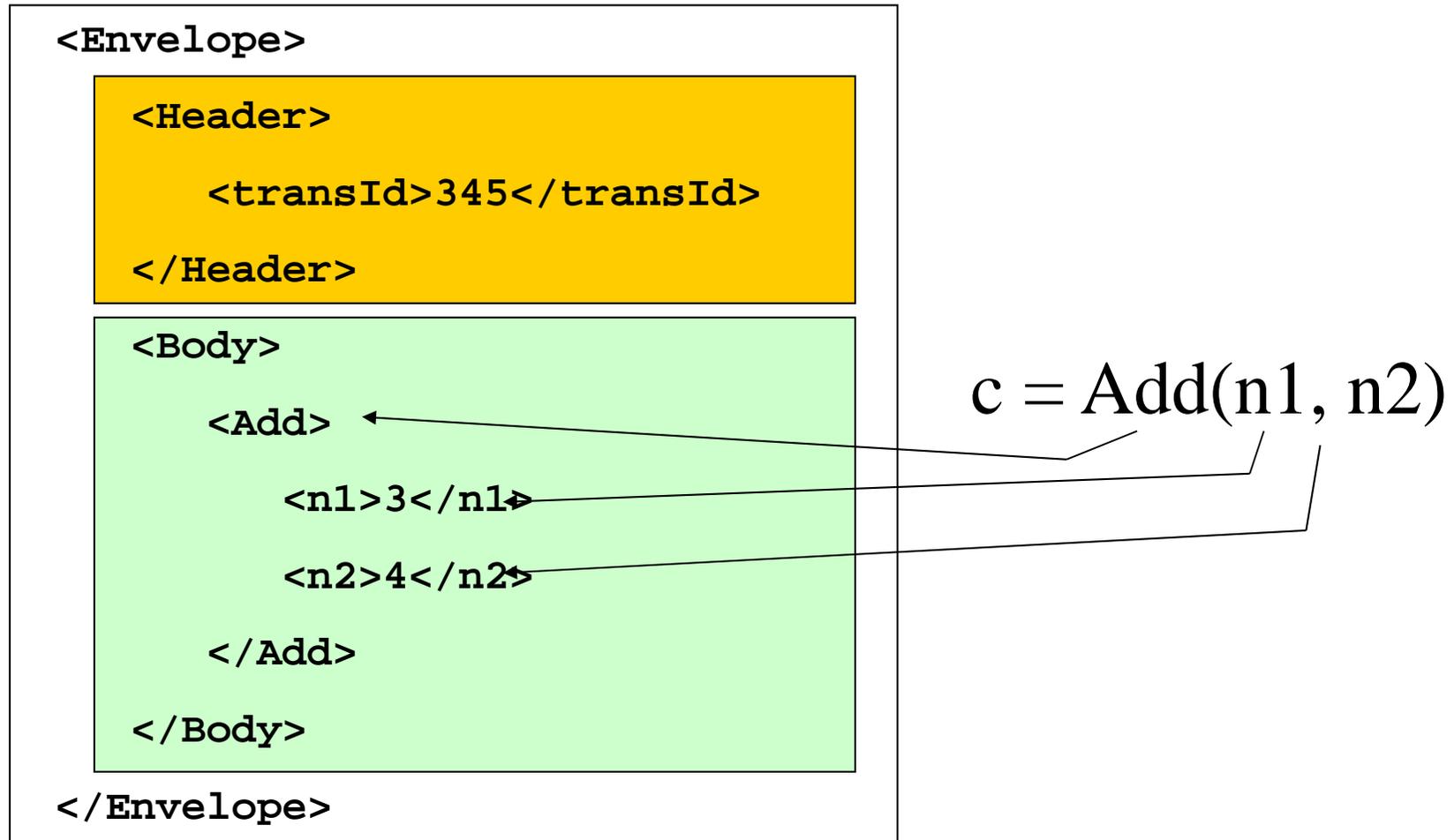
```
<alumno dni="9399390">
  <nombre>Juan</nombre>
  <apellidos>García García</apellidos>
  <nacim>1985</nacim>
</alumno>
```

6.2 SOAP (Simple Object Access Protocol)

- ▶ SOAP es un protocolo basado en XML para el intercambio de información de una forma descentralizada sobre entornos distribuidos.
- ▶ Define un mecanismo para el paso de instrucciones (comandos) y parámetros entre clientes y servidores (RPCs).
- ▶ Es totalmente independiente de la plataforma, el modelo de datos y el lenguaje de programación usado.
- ▶ Se transporta normalmente sobre HTTP.
- ▶ W3C define el uso de:
 - SOAP con XML como payload.
 - HTTP como transporte.
- ▶ Estructura:
 - Sobre (envelop). Contenedor.
 - Cabeceras (header) opcionales. Determina cómo el receptor ha de procesar el mensaje. Añade funcionalidades de autenticación, rutado de mensajes, pago, etc.
 - Cuerpo (body). Información o llamada RPC



SOAP ejemplo simple



SOAP sobre HTTP

POST /examples/stringer ← endpoint address
Host: www.cdk4.net
Content-Type: application/soap+xml
Action: http://www.cdk4.net/examples/stringer#exchange ← action

HTTP
 header

<env:envelope xmlns:env= namespace URI for SOAP envelope
<env:header> </env:header>
<env:body> </env:body>
</env:Envelope>

Soap
 message

6.3 WSDL (Web Services Description Language)

- ▶ Información XML para la descripción de servicios web que se proveen y de un conjunto de puntos finales donde se dan los detalles de la conexión
- ▶ Informalmente: WSDL define...
 - Qué operaciones puede realizar
 - Dónde está
 - Cómo es necesario invocarlo
- ▶ Estructura

```
<definitions>
  <!-- definiciones de tipos de datos -->
  <types>...</types>

  <!-- definiciones de tipos de mensajes -->
  <message name="...">...</message>

  <!-- definiciones de tipos de puerto -->
  <portType name="...">...</portType>

  <!-- definiciones de ligaduras -->
  <binding name="...">...</binding>

  <!-- definiciones de servicios -->
  <service name="...">...</service>
</definitions>
```

WSDL

- ▶ Tipos de datos
 - Todos los tipos de datos utilizados en los mensajes deben ser declarados en esta sección
 - Se puede utilizar cualquier sistema de definición de tipos de datos, aunque normalmente se utiliza XML Schema
- ▶ Ejemplo de tipos de datos

```
<wsdl:types>
  <element name="login">
    <complexType>
      <sequence>
        <element name="login" type="xsd:string"/>
        <element name="password" type="xsd:string"/>
      </sequence>
    </complexType>
  </element>
  <element name="loginResponse">
    <complexType>
      <sequence>
        <element name="loginReturn" type="xsd:string"/>
      </sequence>
    </complexType>
  </element>
</wsdl:types>
```

WSDL

- ▶ Tipos de mensajes
 - Todos los mensajes de entrada y salida de operaciones deben declararse en esta sección
 - A cada mensaje se le asocia un nombre
 - Un mensaje se compone de una o más partes, y para cada parte se indica su nombre y su tipo de datos
- ▶ Ejemplo de tipos de mensajes

```
<wsdl:message name="loginRequest">  
  <wsdl:part element="tns:login" name="parameters"/>  
</wsdl:message>
```

```
<wsdl:message name="findConceptsRequest">  
  <wsdl:part element="tns:findConcepts" name="parameters"/>  
</wsdl:message>
```

```
<wsdl:message name="SQAPSWebServiceException">  
  <wsdl:part element="impl:fault" name="fault"/>  
</wsdl:message>
```

WSDL

- ▶ Tipos de puertos
 - Una operación es una agrupación lógica de un conjunto de mensajes intercambiados
 - Un tipo de puerto es una agrupación lógica de operaciones relacionadas entre sí
- ▶ Una operación puede definir, como mucho, un tipo de mensaje de entrada y otro de salida:
 - En un sentido: llega un mensaje al servicio, y éste no responde (*input*)
 - Petición/respuesta iniciada por el cliente: llega un mensaje al servicio, y éste responde con otro mensaje (*input/output*)
 - Petición/respuesta iniciada por el servicio: el servicio envía un mensaje al cliente, y éste responde (*output/input*)
 - Notificación: el servicio envía un mensaje al cliente y éste no responde (*output*)

WSDL

► Ejemplo de tipos de puertos

```
<wsdl:portType name="SQAPSWebService">
  <wsdl:operation name="query">
    <wsdl:input message="impl:queryRequest" name="queryRequest"/>
    <wsdl:output message="impl:queryResponse" name="queryResponse"/>
    <wsdl:fault message="impl:SQAPSWebServiceException"
      name="SQAPSWebServiceException"/>
  </wsdl:operation>
  <wsdl:operation name="login">
    <wsdl:input message="impl:loginRequest" name="loginRequest"/>
    <wsdl:output message="impl:loginResponse" name="loginResponse"/>
    <wsdl:fault message="impl:SQAPSWebServiceException"
      name="SQAPSWebServiceException"/>
  </wsdl:operation>
</wsdl:portType>
```

WSDL

- ▶ Tipos de ligaduras (bindings)
 - Una ligadura especifica cómo formatear los mensajes para interaccionar con un servicio dado
 - A cada tipo de puerto se le asigna una o más ligaduras
 - WSDL es extensible en cuanto a la definición de nuevos tipos de ligaduras. Algunos predefinidos:
 - Ligadura SOAP: especifica cómo codificar los mensajes en mensajes SOAP
 - Ligadura HTTP/MIME: especifica cómo codificar los mensajes directamente en HTTP con MIME (sin SOAP)

WSDL

▶ Ejemplos de tipos de ligaduras

```
<wsdl:binding name="SQAPSSoapBinding" type="impl:SQAPSWebService">
  <wsdlsoap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="login">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="loginRequest">
      <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="loginResponse">
      <wsdlsoap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="SQAPSWebServiceException">
      <wsdlsoap:fault name="SQAPSWebServiceException" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
  (...)
</wsdl:binding>
```

WSDL

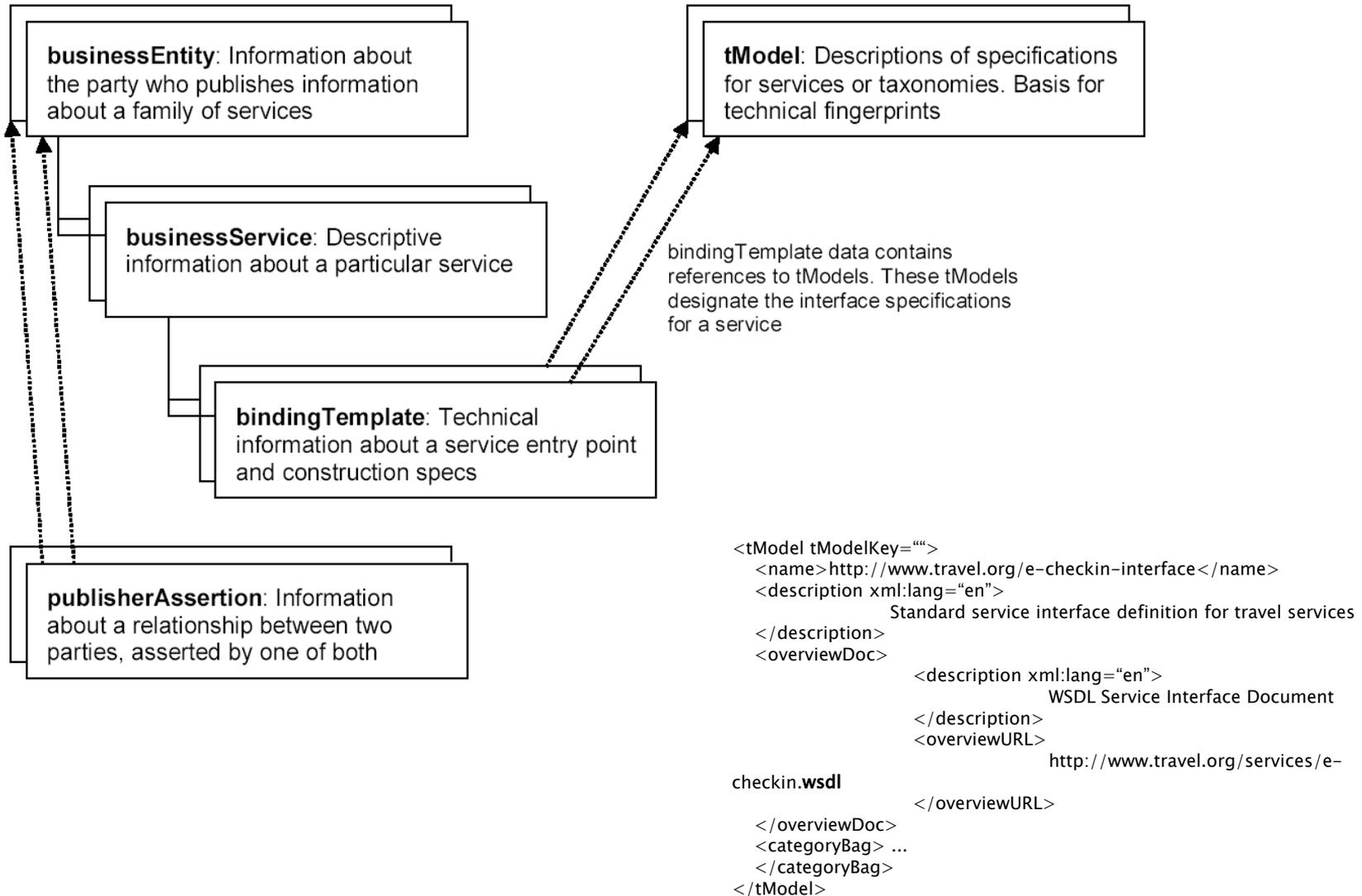
- ▶ Tipos de servicios
 - Un servicio representa un tipo de puerto, con una ligadura concreta, accesible a través de una dirección dada
 - Puede haber más de un servicio asociado al mismo tipo de puerto y misma ligadura, cambiando únicamente su dirección
- ▶ Ejemplo de tipo de servicio

```
<wsdl:service name="SQAPSWebServiceService">  
  <wsdl:port binding="impl:SQAPSSoapBinding" name="SQAPS">  
    <wsdlsoap:address location=  
      "http://bach.gast.it.uc3m.es:8180/axis/services/SQAPS"/>  
  </wsdl:port>  
</wsdl:service>
```

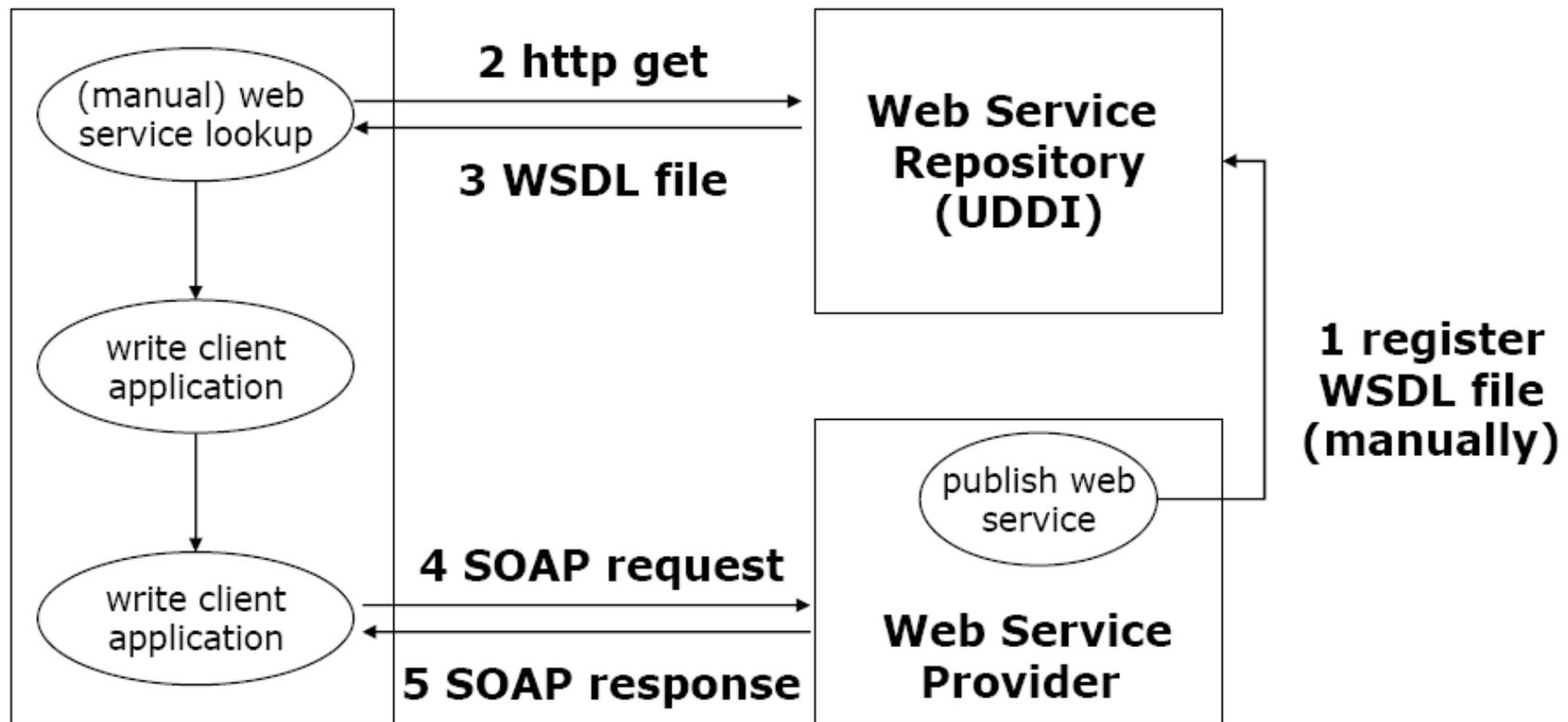
6.4 UDDI (Universal Description Discovery and Integration)

- ▶ UDDI Estructura de Datos estándar
- ▶ También basado en XML
- ▶ Es un servicio Web
- ▶ Soporta:
 - Gestión de taxonomía para ayudar las búsquedas
 - Búsquedas por personas y máquinas
- ▶ Los usuarios pueden hacer peticiones a un registro UDDI (búsquedas por criterio) y el resultado es un puntero al documento WDSL que caracteriza el servicio

Información de registro



6.5 Integración



Integración

Descripción Interfaz WSDL :

```
...  
<s:element name="MakeEncRequest">  
  <s:complexType>  
    <s:sequence>  
      <s:element ... name="principal" type="s:string" />  
      <s:element ... name="cipher" type="s:base64Binary" />  
    </s:sequence>  
  </s:complexType>  
</s:element>
```

Petición SOAP:

```
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope ...>  
  <soap:Body>  
    <MakeEncRequest ...>  
      <principal>adg@microsoft.com</principal>  
      <cipher>+oXHdaCXvK...=</cipher>  
    </MakeEncRequest>  
  </soap:Body>  
</soap:Envelope>
```

Respuesta SOAP:

```
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope ...>  
  <soap:Body>  
    <MakeEncRequestResponse ...>  
      <MakeEncRequestResult>  
        Accepted: with nonce=1  
      </MakeEncRequestResult>  
    </MakeEncRequestResponse>  
  </soap:Body>  
</soap:Envelope>
```

6.6 Otros interfaces a servicios web

- ▶ XML-RPC
 - Mediante métodos POST HTTP peticiones y respuestas XML que imitan una funcionalidad RPC sobre la web
- ▶ REST (REpresentational State Tranfer)
 - Patrón de diseño
 - Mediante métodos GET HTTP peticiones y respuestas XML/JSON que imitan una funcionalidad RPC sobre la web
 - En la propia URL se codifica la petición por lo que simplifica la depuración
 - Válido para peticiones de tamaño pequeño
- ▶ AJAX (Asynchronous JavaScript and XML)

6.7 Ejemplos de Web Services

- ▶ <http://aws.amazon.com/es/>
 - Permite acceder directamente a sus servicios, manejando directamente el “carrito de compra”.
 - Puedes actualizar tus paginas con su información.
 - Te permite contratar servicios de infraestructura

- ▶ <http://www.google.com/apis/>
 - Pueden invocar la búsqueda directamente usando una API.
 - Desde 2008 migrado a AJAX.

- ▶ <http://developer.yahoo.net/search/>
 - Servicios varios Yahoo.

- ▶ <http://www.websvcex.net/ws/default.aspx>
 - Directorio con múltiples web services de acceso libre

Referencias

- ▶ George Coulouris, Jean Dollimore and Tim Kindberg. Distributed Systems: Concepts and Design. Addison-Wesley. ISBN 0321263545, 2007, 4a edición
- ▶ Web services
 - <http://www.w3.org/2002/ws/>
 - <http://www-128.ibm.com/developerworks/webservices/>
- ▶ SOAP
 - <http://www.w3.org/TR/soap/>
- ▶ WSDL
 - <http://www.w3c.org/TR/wsdl>
- ▶ UDDI
 - <http://www.uddi.org>
 - Demo: <https://uddi.ibm.com/testregistry/registry.html>