
Sistemas Operativos Distribuidos

Introducción a los Servicios Web (*Web Services*)

Evolución de la Web

- Pasado: Web de documentos
 - Páginas estáticas
 - Web como un enorme repositorio de información
 - Tecnologías: HTTP + HTML
- Presente: Web de aplicaciones
 - Páginas dinámicamente generadas por aplicaciones web
 - Aplicaciones exportan su interfaz a los usuarios a través de la Web
 - Entorno de transacciones comerciales (*Business to consumer*, B2C)
 - Tecnologías: CGI, ASP, PHP, JSP, *servlets*, ...
- Futuro (ya está aquí): Web de servicios (funciones/métodos)
 - “Bibliotecas” ofrecen servicios a **programas** (no a usuarios)
 - Web como una enorme API de servicios (Web de componentes)
 - Empresas de valor añadido (*Business to business*, B2B)
 - Base de Sistemas distribuidos sobre Internet
 - Servicio web: RPC sobre la Web usando XML

Aplicaciones web: Escenario típico

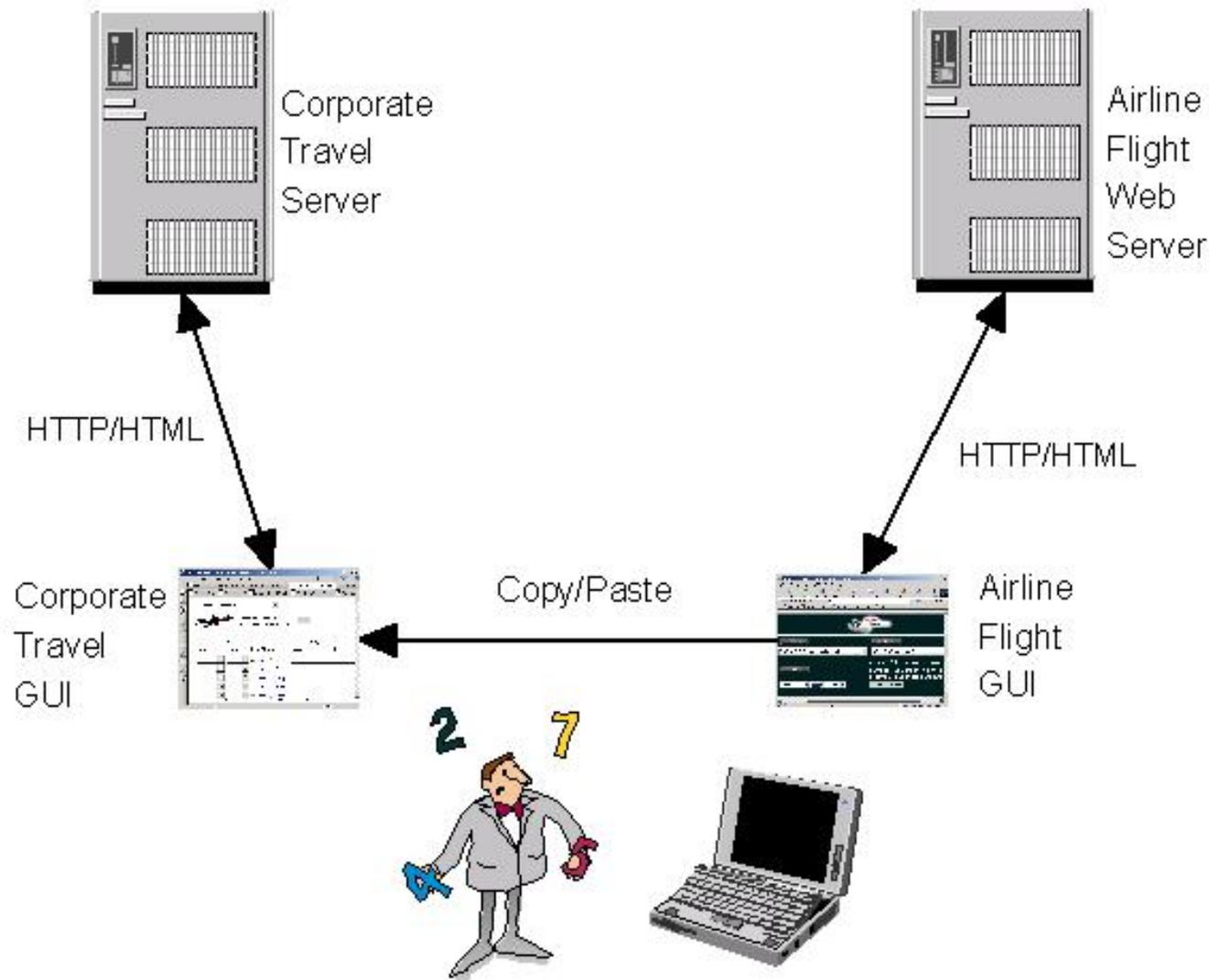


Figura extraída de “*Understanding Web Services*”: <http://www7.software.ibm.com/vad.nsf/Data/Document4362>

Servicios web: Escenario típico

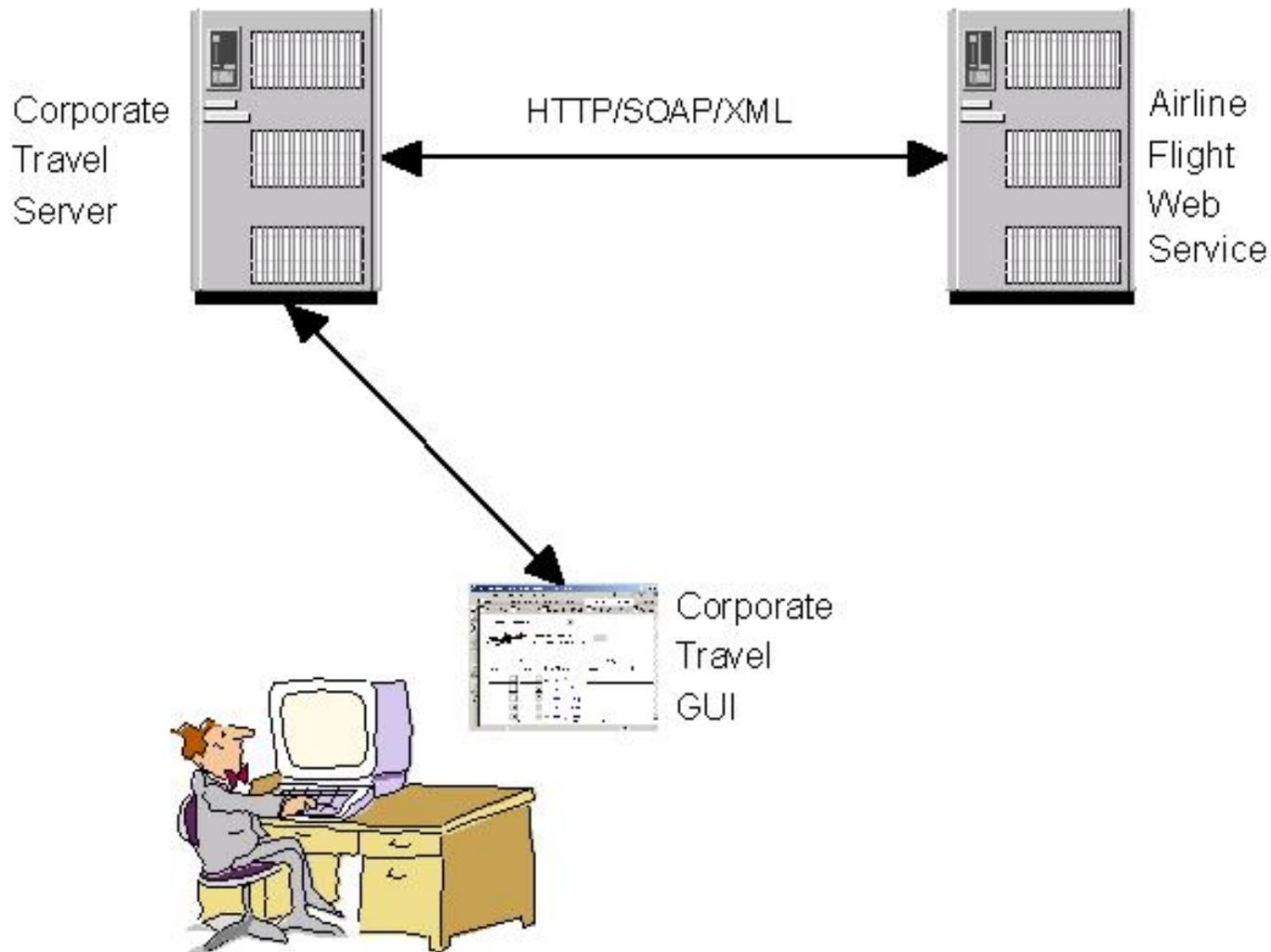


Figura extraída de “*Understanding Web Services*”:
<http://www7.software.ibm.com/vad.nsf/Data/Document4362>

Integración de servicios web

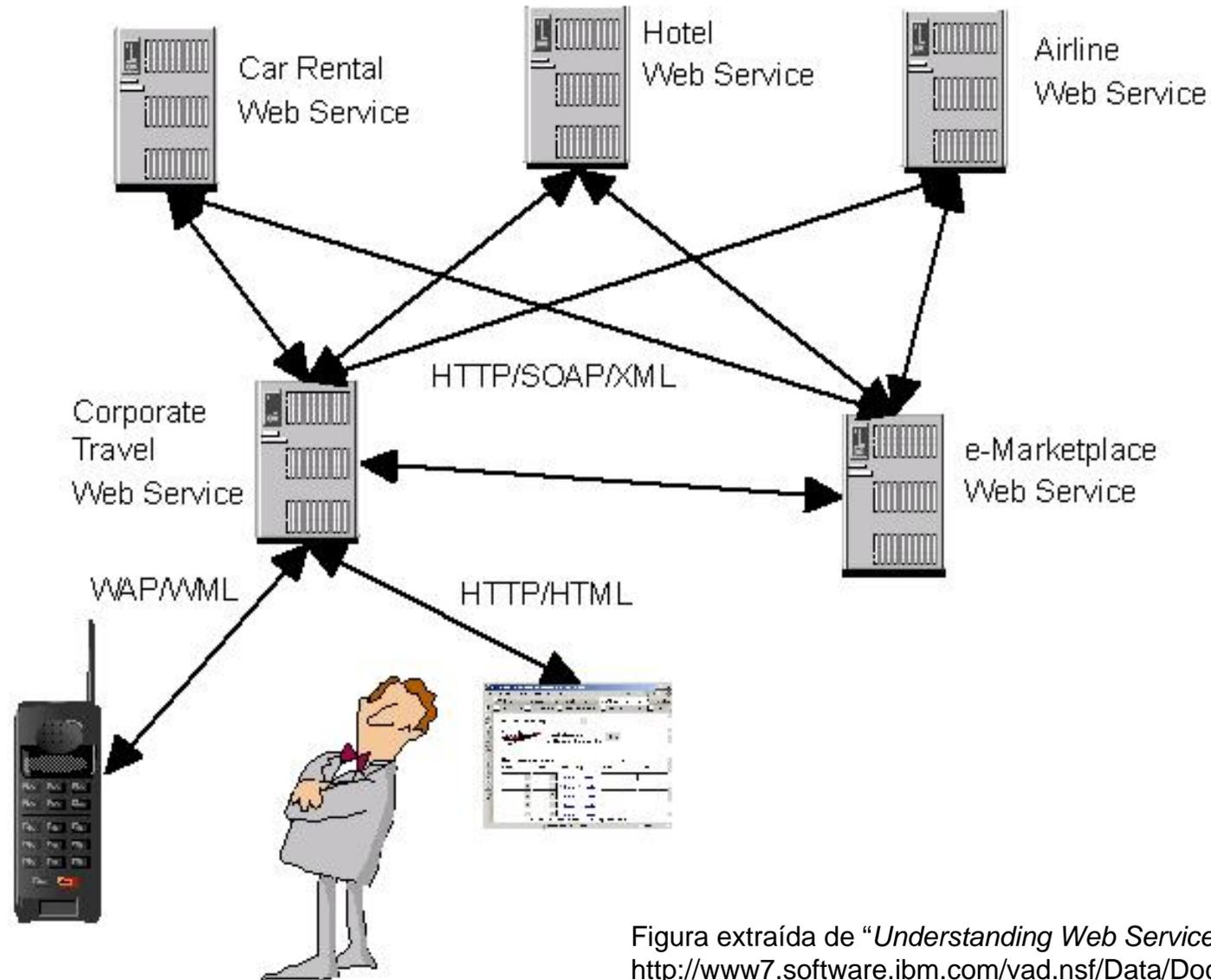


Figura extraída de “*Understanding Web Services*”:
<http://www7.software.ibm.com/vad.nsf/Data/Document4362>

Definición de servicio web

- Módulo que exporta un conjunto de funciones (métodos) a aplicaciones a través de la Web proporcionando independencia de plataformas hardware/software
- Similar a RPC o RMI pero integrado en la Web
 - ¿Reinventando la rueda? → ¿Por qué no usar CORBA?
- Estandarización controlada por un grupo del W3C:
 - <http://www.w3.org/2002/ws/>
- Mismas cuestiones que con RPC/RMI:
 - ¿Qué protocolo de transporte se usa? → HTTP
 - ¿Qué formato de representación se utiliza? → XML
 - ¿Qué protocolo de comunicación se usa? → SOAP
 - ¿Cómo se especifican los servicios exportados (IDL)? → WSDL
 - ¿Cómo localiza el cliente al servidor (*binding*)? → UDDI

Protocolo de transporte: HTTP

- Uso típico de operación POST de HTTP:
 - datos de formulario y página de respuesta

```
POST /~ssoo/consultaBD.cgi HTTP/1.0
```

```
Content-length: 76
```

```
.....
```

```
DNI=87654321&MAT=980000&Asignatura=sod&Curso=2002&Convocatoria=Jun&Tipo=acta
```

```
HTTP/1.1 200 OK
```

```
Content-Type: text/html; charset=iso-8859-1
```

```
.....
```

```
<HTML>
```

- Uso de POST para petición y respuesta de RPC
 - Universalmente disponible
 - Atraviesa el *firewall* de la organización

Formato de representación: XML

- Información de RPC codificada en XML
 - Muy flexible y potente
 - XML Schema permite definir con precisión los tipos de datos
- Ej: *float GetLastTradePrice(string symbol);*

Petición:

```
<GetLastTradePrice>  
  <symbol>DIS</symbol>  
</GetLastTradePrice>
```

Respuesta:

```
<GetLastTradePriceResponse>  
  <Price>34.5</Price>  
</GetLastTradePriceResponse>
```

Esquema:

```
<element name="GetLastTradePrice">  
  <complexType><all>  
    <element name="symbol" type="string"/>  
  </all></complexType>  
</element>  
<element name="GetLastTradePriceResponse">  
  <complexType><all>  
    <element name="Price" type="float"/>  
  </all></complexType>  
</element>
```

Protocolo de comunicación: SOAP

- *Simple Object Access Protocol (Candidate Recommendation)*
- SOAP = HTTP + XML
 - Especifica cómo mandar mensajes XML sobre HTTP
 - Define el contenedor del mensaje (también en XML)
 - Protocolo general, no sólo para RPC, también unidireccional
- Estructura de mensaje contenedor SOAP:
 - Sobre (*Envelope*): Cabecera (*Header*) [opcional] + Cuerpo (*Body*)
 - Cabecera : info. complementaria (p.ej. en RPC un ID de transacción)
 - Cuerpo: contiene el mensaje original
- SOAP para RPC:
 - En petición: Identificador en POST identifica destino de RPC
- Seguridad:
 - Usando HTTPS (SSL)
 - Nueva propuesta: *WS-Security*

Un ejemplo de SOAP en RPC

POST /StockQuote HTTP/1.1

.....

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="http://example.com/stockquote.xsd">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Petición

HTTP/1.1 200 OK

.....

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePriceResponse xmlns:m="http://example.com/stockquote.xsd">
      <Price>34.5</Price>
    </m:GetLastTradePriceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Respuesta

Definición de interfaz de servicio: WSDL

- *Web Service Description Language (Working Draft)*
- IDL para servicios Web basado en XML
- Documento WSDL describe servicio web:
 - Tipos de datos (XML Schema)
 - Funciones exportadas y sus mensajes de petición y respuesta
 - Protocolos usados: típicamente SOAP sobre HTTP
 - Dirección de servicio → URL con servidor y "componente"
 - P. ej. *<http://www.stockquoteserver.com/StockQuote>*
- Normalmente, generado automáticamente a partir de código de servicios

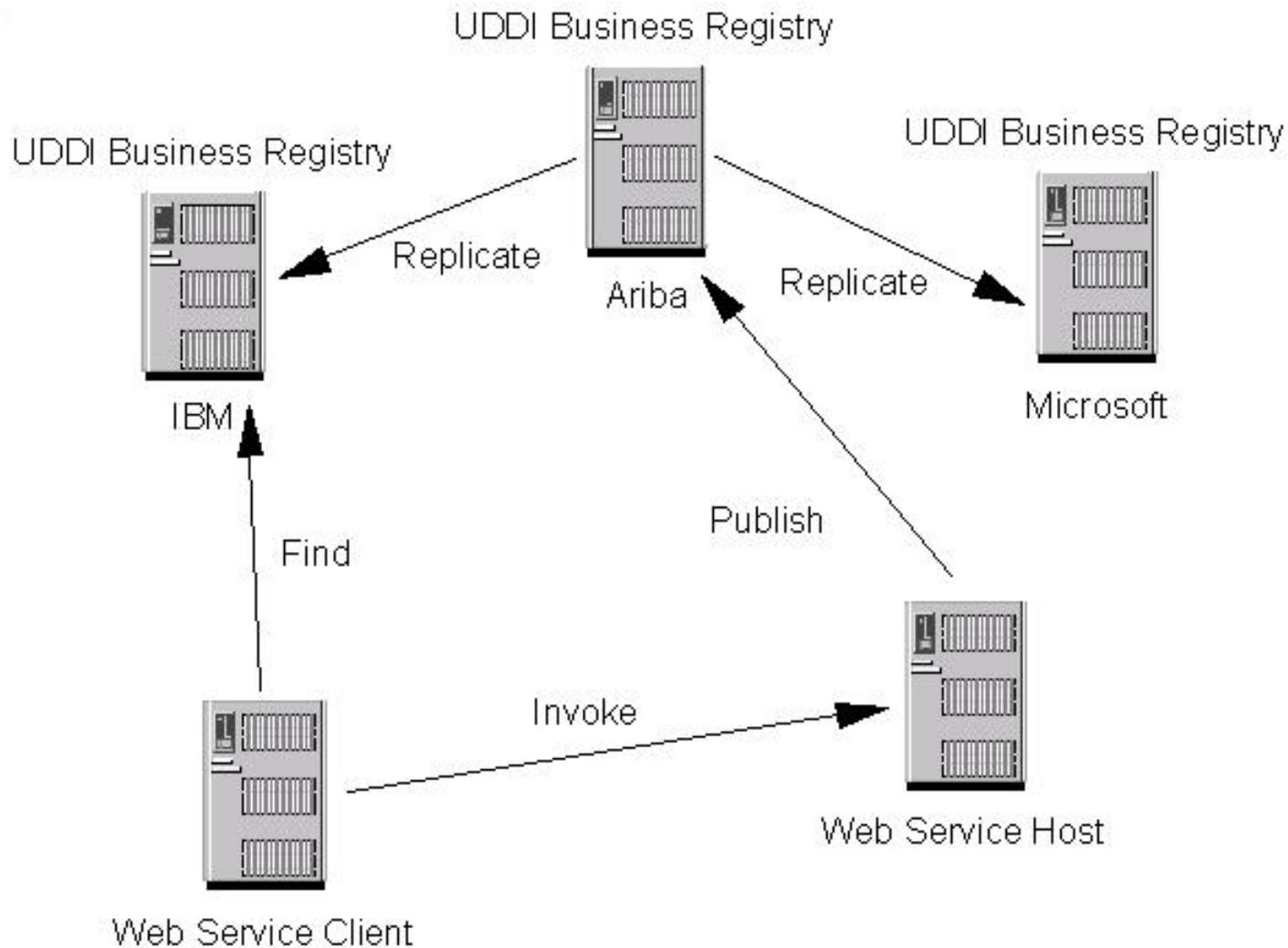
Desarrollo de un servicio Web

- Programación de biblioteca de servicio
 - En algunos entornos hay que incluir información específica
 - En VisualStudio .Net: etiqueta *[WebMethod]* sobre métodos exportados
- Generación automática de fichero WSDL
 - Generalmente, dentro de la generación de aplicación de servicio
 - En VisualStudio .Net: Proyecto de tipo *Web Service*
- En servidor: fichero WSDL informa sobre cómo activar servicio
 - Normalmente, lo hace un servidor web con soporte de servicios web
- Desarrollo de cliente:
 - Obtener fichero WSDL y generar proxy para aplicación cliente
 - En VisualStudio .Net: "*Add Web Reference*"

Localización del servicio: UDDI

- *Universal Description, Discovery, and Integration*
 - No estándar: Propuesta inicial de Microsoft, IBM y Ariba
- Registro distribuido de servicios web ofrecidos por empresas
- Información clasificada en 3 categorías (guías):
 - Páginas blancas: Datos de la empresa
 - Páginas amarillas: Clasificación por tipo de actividades
 - Páginas verdes: Descripción de servicios web (WSDL)
- Se accede a su vez como un servicio web
- Puede consultarse en tiempo de desarrollo o incluso dinámicamente en tiempo de ejecución
- Permite búsquedas por distintos criterios
 - Tipo de actividad, tipo de servicio, localización geográfica

Registro de un servicio web



Extesiones de protocolos

- ASAP (Asynchronous Service Access Protocol):
 - Solicitudes asíncronas (envío cliente -> servidor).
 - Extensión de SOAP.
 - Pensadas para transacciones de larga duración.
- DIME (Direct Internet Message Encapsulation):
 - Optimización seleccionando la codificación de porciones del mensaje.
 - Extensión de SOAP / SOAP MTOM.
 - Empaquetado más ligero.

Servicios web vs. RPC/RMI

- Servicio Web similar a RPC/RMI (Corba, DCOM)
 - ¿Hay un “ganador”? ¿Reinventando la rueda?
- Convivencia: Distintos ámbitos de aplicación
- Servicios web
 - Entornos de interacción “débilmente acoplados”
 - Exportar servicios fuera de la organización
 - Integrar aplicaciones de la empresa
 - Más estándar y extendido, menos problemas con *firewalls*
- RPC/RMI “tradicionales”
 - Aplicación distribuida “fuertemente acoplada”
 - Más funcionalidad y eficiencia
- Ejemplo de escenario de convivencia:
 - Exportar un servicio interno CORBA mediante un servicio web

Entornos de desarrollo de servicios web

- Número creciente de entornos de desarrollo
- Algunas implementaciones de interés:
 - .Net de Microsoft
 - *Web Services Project* de Apache
 - *Java Web Services Developer Pack*
 - *IBM WebSphere SDK for Web services (WSDK)*
 - WASP de Systinet
- Cursos sobre SOAP, WSDL y otras tecnologías web:
 - <http://www.w3schools.com/>