Programación Web

TEMA 6. SERVICIOS WEB

Curso 2016-2017

Bibliografía

- E. Cerami, «Web Services Essentials», O'Reilly, 2002
- D. Chappell y T. Jewell, «Java Web Services», O'Reilly, 2002
- L.J. Mitchel, "PHP Web Services", O'Reilly, 2013.
- E. Newcomer, «Understanding Web Services», Addison-Wesley Professional, 2002
- W3C, «Web Services Architecture», http://www.w3.org/TR/ws-arch

Contenido

- Definición
- Conceptos básicos
- Arquitectura

DEFINICIONES

Servicio web

- Cualquier servicio ofrecido a través de Internet, a través de un sistema de mensajes XML
- Sistemas de mensajes XML:
  - XML-RPC
  - SOAP
  - HTTP GET/POST

Definición del W3C

- A Web service is a software system designed to support interoperable ***machine-to-machine interaction*** over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

Eric Newcomer

Web services provide a ***layer of abstraction*** above existing software systems, such as application servers, CORBA, .NET servers, messaging, and packaged applications. Web services work at a level of abstraction similar to the Internet and *are* ***capable of bridging*** any operating system, hardware platform, or programming language, just as the Web is.

Web services are Extensible Markup Language (XML) **applications mapped** to programs, objects,

or databases or to comprehensive business functions.

Using an XML document created in the form

of a message, a program sends a request to a Web service across the network, and, optionally,

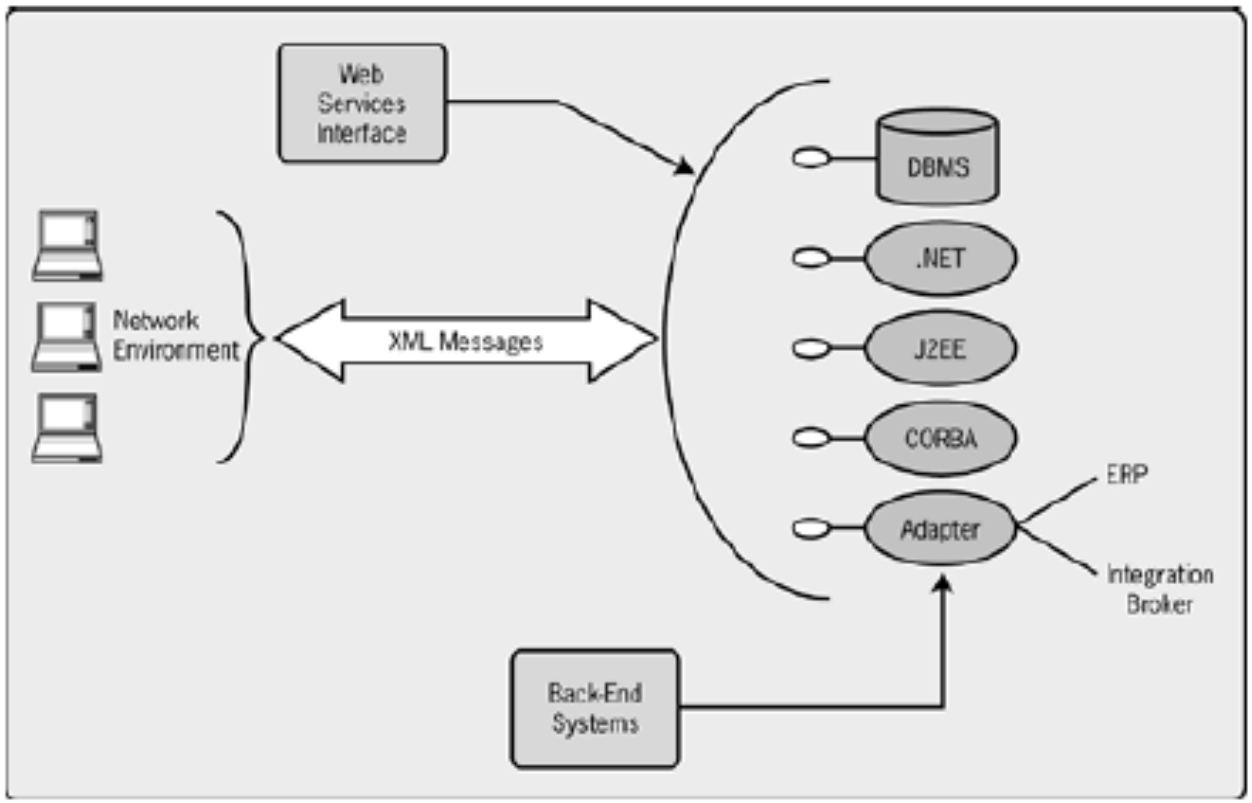
receives a reply, also in the form of an XML document.

Web services wrap, presenting to the network a standard way of interfacing with back-end software systems, such as database management systems, .NET, J2EE (Java2 Platform, Enterprise Edition), or CORBA (common object request broker architecture), objects, adapters to enterprise resource planning (ERP) packages, integration brokers, and others.

Web services interfaces

1. receive a standard XML message from the networking environment,
2. transform the XML data into a format understood by a particular back-end software system, and, optionally,
3. return a reply message.

The underlying software implementations of Web services can be created by using any programming language, operating system, or middleware system.


Web
Services
Interface
DBMS
.NET
J2EE
CORBA
ERP
Adapter
Integration
Broker
Network
Environment
XML Messages
Back-End
Systems

- Son componentes de aplicaciones
- Se comunican a través de protocolos abiertos
- Son autocontenidos y autodescriptivos
- Descubribles a través de UDDI
- Utilizables por otras aplicaciones

Ejemplo: búsqueda de información

- Mecanismo habitual: insertar datos en HTML

http://www.google.com/search?q=Skate+boots&btnG=Google+Search

Búsqueda de «Skate boots» en el motor de Google:

- search: servicio requerido
- «Skate+boots»: cadena buscada, enviada en HTML

Alternativa XML

```
<SOAP-ENV:Body>
  <s:SearchRequest
    xmlns:s="www.xmlbus.com/SearchService">
      <p1>Skate</p1>
      <p2>boots</p2>
      <p3>size 7.5</p3>
  </s:SearchRequest>
</SOAP-ENV:Body>
```

Ventajas del envío en XML

- Mejor control de los tipos de datos y la estructura de la información
- Más flexibilidad
- Más extensible
- …

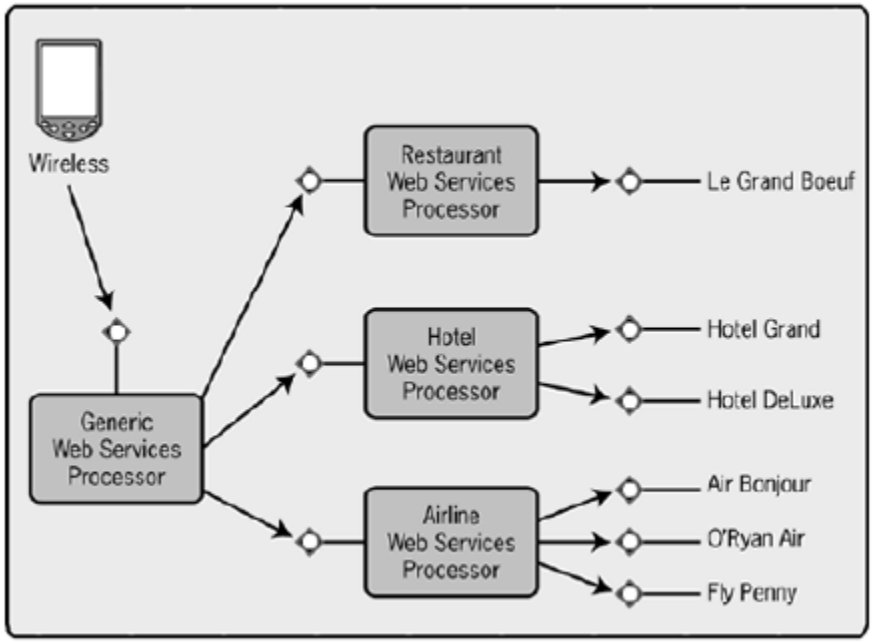
Desventajas

- Para transacciones, el desarrollo es mucho menos maduro que el de estándares de computación distribuida como CORBA
- El rendimiento es muy bajo, comparado con otros modelos de computación distribuida. RMI, CORBA
  - Entre los objetivos de XML no están la concisión, ni la eficiencia en el procesamiento

Ejemplos de aplicaciones

- Buscar y comprar bienes y servicios al mejor precio
  - Comprar billetes de avión; seguros de coche; hoteles, …
- Coordinar billetes de viajes y eventos
- Gestión de procesos de negocio: consecución, facturación y envío
- …

Coordinación de vuelo, alojamiento y restaurante



CONCEPTOS BÁSICOS

Entidades

- Agentes y servicios
- Clientes y proveedores
- Descripción y localización de servicios

Agentes y servicios

- El **agente** es el componente software que gestiona la comunicación (mensajes).
- El **servicio** es el recurso, caracterizado por la funcionalidad que se provee
- Ej.: el agente se puede implementar con distintos lenguajes de programación, mientras que el servicio es el mismo

Clientes y proveedores

- El proveedor es la entidad que ofrece un agente que implementa un servicio particular
- El cliente es la entidad que desea utilizar el servicio de un proveedor. Implementa un agente que dialoga con el agente del proveedor

Descripción de los servicios

- El protocolo de intercambio de mensajes para prestación de servicios web se describe en el lenguaje **WSDL**
- La descripción del protocolo (WSD) incluye formatos de mensajes, tipos de datos, protocolos de transporte y *serialización*
- También indicaciones en las que localizar el servicio

Interacción para un servicio web

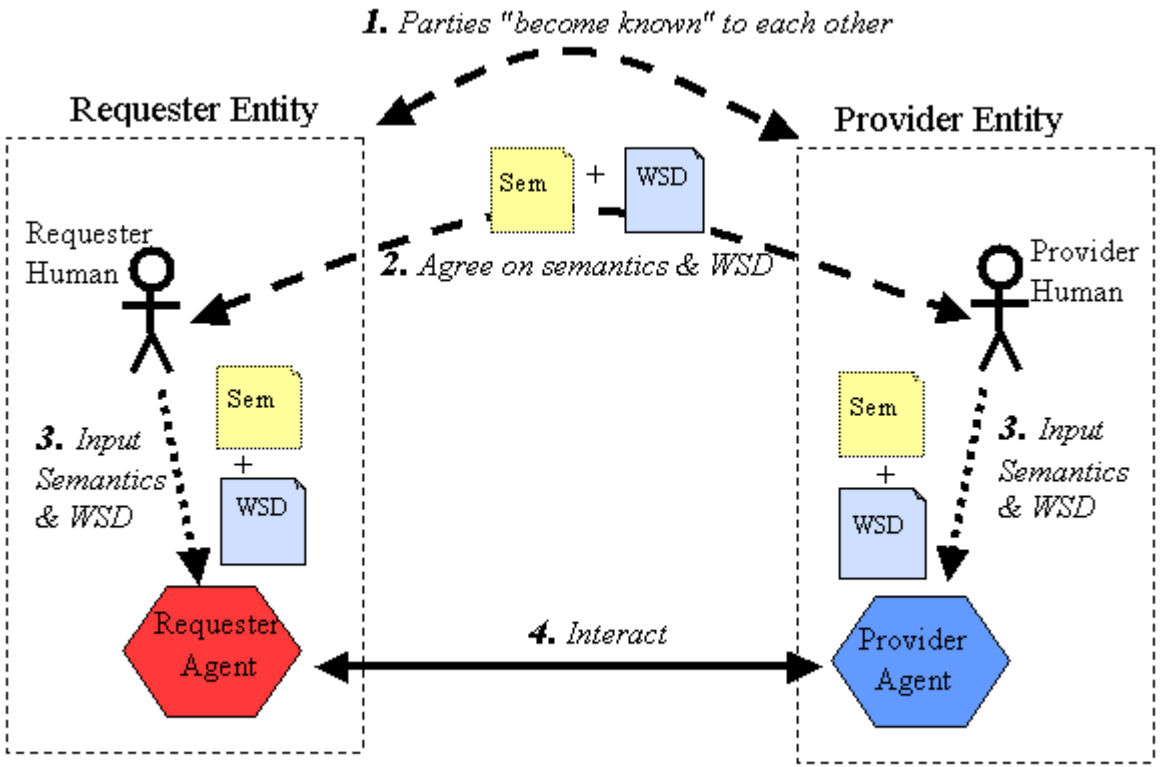


Figure 1-1. The General Process of Engaging a Web Service

ARQUITECTURA

Tecnología para servicios web

Programs that interact with one another over the Web must be able to:

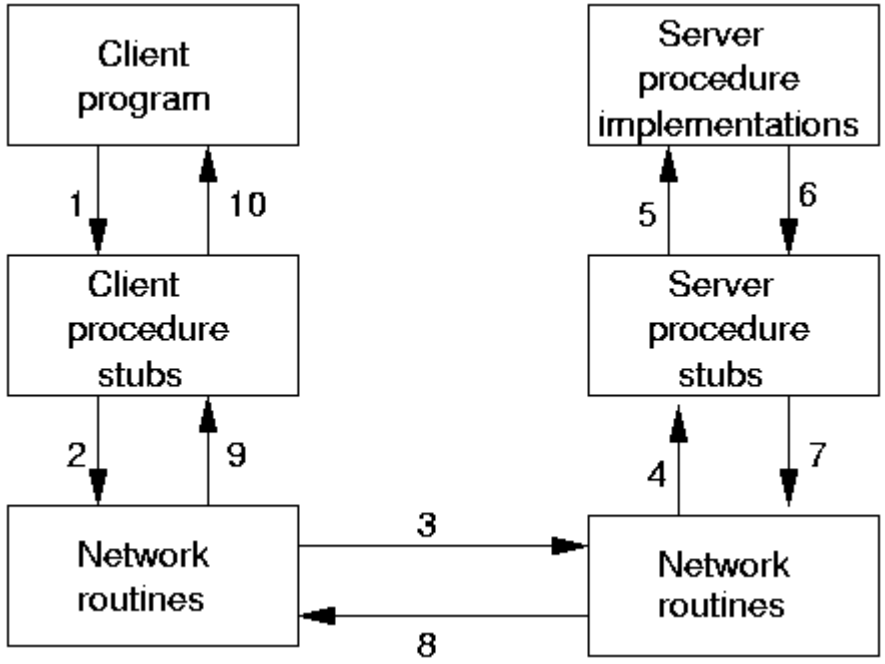
- find one another, discover information allowing them to interconnect,
- figure out what the expected interaction patterns are— a simple request/reply or more complicated process flow?—
- negotiate such qualities of service as security, reliable messaging, and transactional composition.

Estándares

- XML-RPC
- WSDL
- SOAP
- UDDI

RPC

- RPC implementa un proceso de llamada a un procedimiento, pero en una ubicación remota (no local)
- Existe una complejidad adicional en el proceso de comunicación entre el invocador y el llamado
- Protocolos existentes para computación distribuida



XML en Servicios Web

- XML se creó para superar las limitaciones de HTML, particularmente para propocionar mejor soporte en la creación y gestión de contenido dinámico
- Con XML puedes crear cualquier elemento que asocie significado a los datos: mucha flexibilidad
- Los esquemas restringen dicha flexibilidad

XML-RPC

Extensible Markup Language-Remote Procedure Call

- Protocolo que usa XML para ejecutar RPC
- Solicitudes codificadas en XML y enviadas vía POST
- Las respuestas XML se incrustan en el cuerpo de la respuesta HTTP
- Es independiente de plataforma

Ejemplo: weather service

```
<methodCall>
  <methodName>weather.getWeather </methodName>
     <params>
        <param><value>10016</value></param>
      </params>
</methodCall>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodResponse>
   <params>
       <param>
    <value><int>65</int></value>
    </param>
  </params>
</methodResponse>
```

WSDL

Web Services Description Language

- Lenguaje, basado en XML, para definir
  - Tipos de datos incluidos en los mensajes
  - Operaciones a realizar sobre los mensajes
  - La traducción de mensajes en redes de transporte (interconexión entre servicios)

Ejemplo: servicio weather (I)

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="WeatherService"
    targetNamespace="http://www.ecerami.com/wsdl/
WeatherService.wsdl"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/
soap/"
    xmlns:tns="http://www.ecerami.com/wsdl/
WeatherService.wsdl"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<message name="getWeatherRequest">
    <part name="zipcode" type="xsd:string"/>
</message>
```

Ejemplo: servicio weather (II)

```
<message name="getWeatherResponse">
<part name="temperature" type="xsd:int"/>
</message>

<portType name="Weather_PortType">
  <operation name="getWeather">
     <input message="tns:getWeatherRequest"/>
     <output message="tns:getWeatherResponse"/>
   </operation>
</portType>
```

servicio weather (III)

```
<service name="Weather_Service">
<documentation>WSDL File for Weather
Service</documentation>
<port binding="tns:Weather_Binding"
name="Weather_Port">
<soap:address
location="http://localhost:8080/soap/
servlet/rpcrouter"/>
</port>
</service>
</definitions>
```

SOAP

Simple Object Access Protocol

- Protocolo de comunicaciones que define un formato de *serialización* para la transmisión de documentos XML sobre una red y para representar interacciones RPC

Ejemplo: petición

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-
envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getWeather
       xmlns:ns1="urn:examples:weatherservice"
        SOAP-ENV:encodingStyle="http://www.w3.org/
2001/09/soap-encoding/">
        <zipcode xsi:type="xsd:string">10016</zipcode>
     </ns1:getWeather>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Ejemplo: respuesta

```
<?xml version='1.0' encoding='Utf-8'>
<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-
envelope/"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <SOAP-ENV:Body>
      <ns1:getWeatherResponse
         xmlns:ns1="urn:examples:weatherservice"
         SOAP-ENV:encodingStyle="http://www.w3.org/
2001/09/soap-encoding/">
         <return xsi:type="xsd:int">65</return>
      </ns1:getWeatherResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

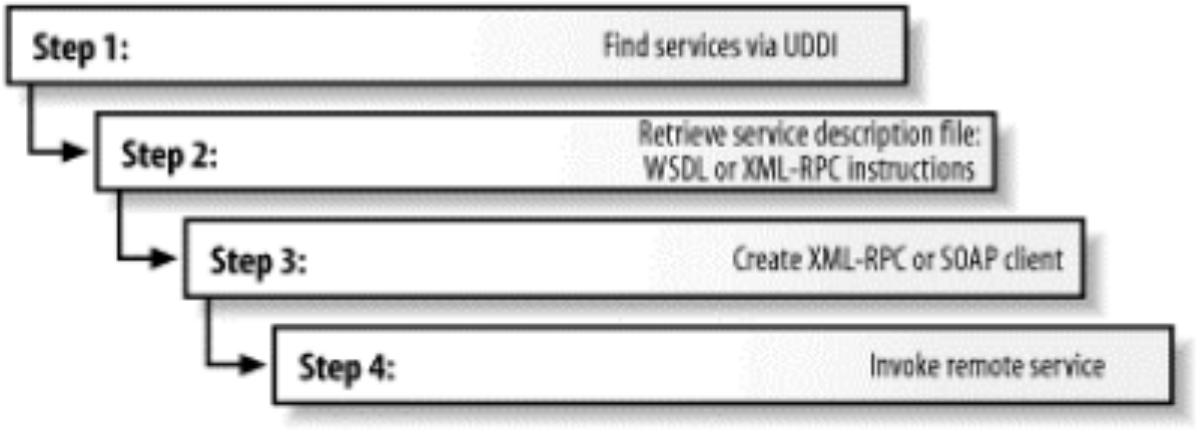
UDDI

Universal Description, Discover, and Integration

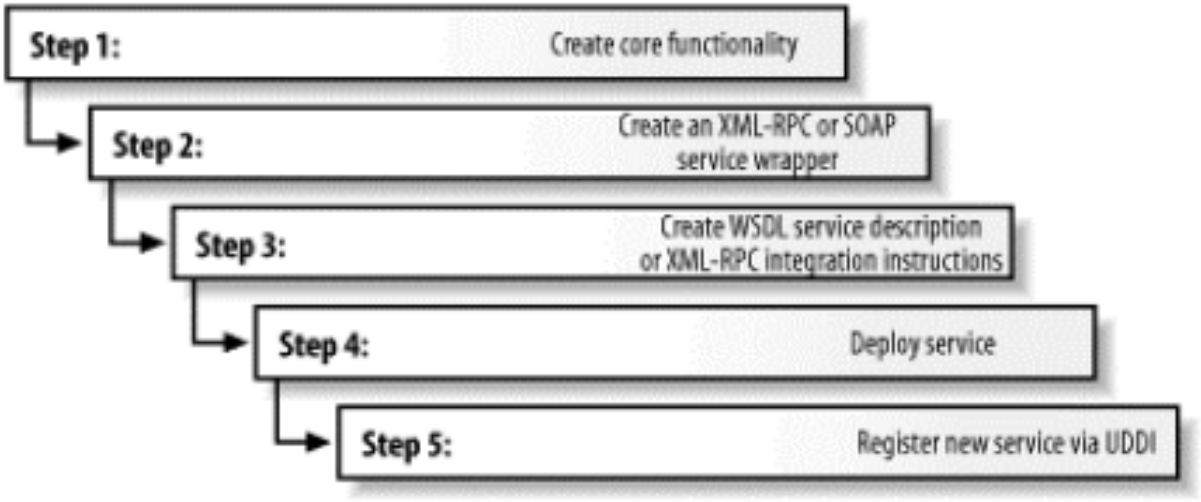
- Mecanismos de registro y descubrimiento de servicios web utilizados para almacenar y categorizar información de negocio y para recuperar direcciones de acceso a servicios web

Pila de protocolos

Perspectiva del cliente



Perspectiva del proveedor



Beneficios de los servicios web

- Exponer la funcionalidad a la red
- Conectar distintas aplicaciones (interoperatividad)
- Protocolo estandarizado
- Bajo coste de comunicación