Desarrollo de Aplicaciones Distribuidas

AUTORES:
Alejandro Calderón Mateos
Javier García Blas
David Expósito Singh
Laura Prada Camacho

Departamento de Informática
Universidad Carlos III de Madrid
Julio de 2012

APLICACIONES DE INTERNET: SOAP

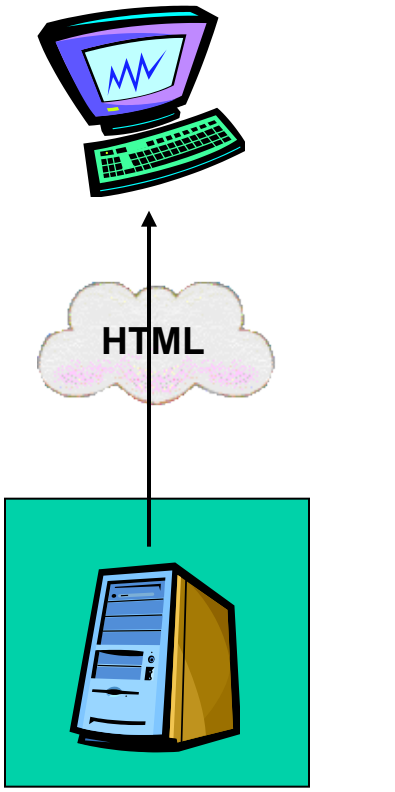
Contenidos

1. SOAP
   1. Introducción
   2. Arquitectura
   3. Ejemplo de aplicación
      - Desarrollo de un servicio privado

Paradigmas de Servicios de red, ORB, etc.

Espacio de objetos, aplicaciones colaborativas
Servicios de red, Servicios Web ***object request broker*, agentes móviles**
procedimientos remotos, métodos remotos
Cliente-servidor, *peer-to-peer*
Paso de mensajes

Evolución de la Web...

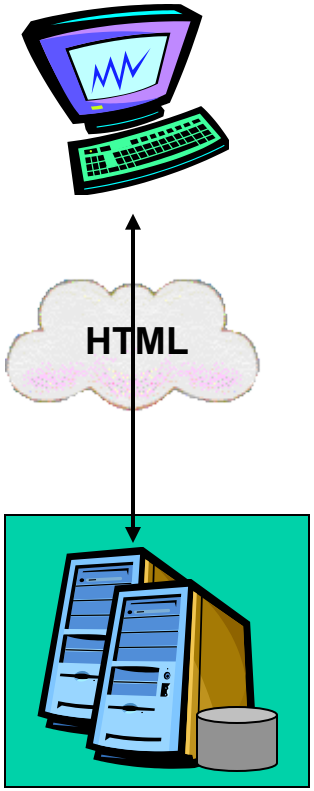


Generación 1
HTML estático

- El navegador Web pide una página Web indicando su identificador URI en la petición.
- El servidor Web busca el fichero almacenado que se corresponde con la URI pedida, y lo envía como respuesta.
- Se utiliza el protocolo HTTP para la transferencia de contenido.
- Contenido diverso:
  - Páginas HTML
  - Imágenes: PNG, JPEG, etc.
  - Vídeos: mov, AVI, etc.
  - Sonidos: MP3, .wav, etc.

http://www.w3.org/

Evolución de la Web...



Generación 2
Aplicaciones Web

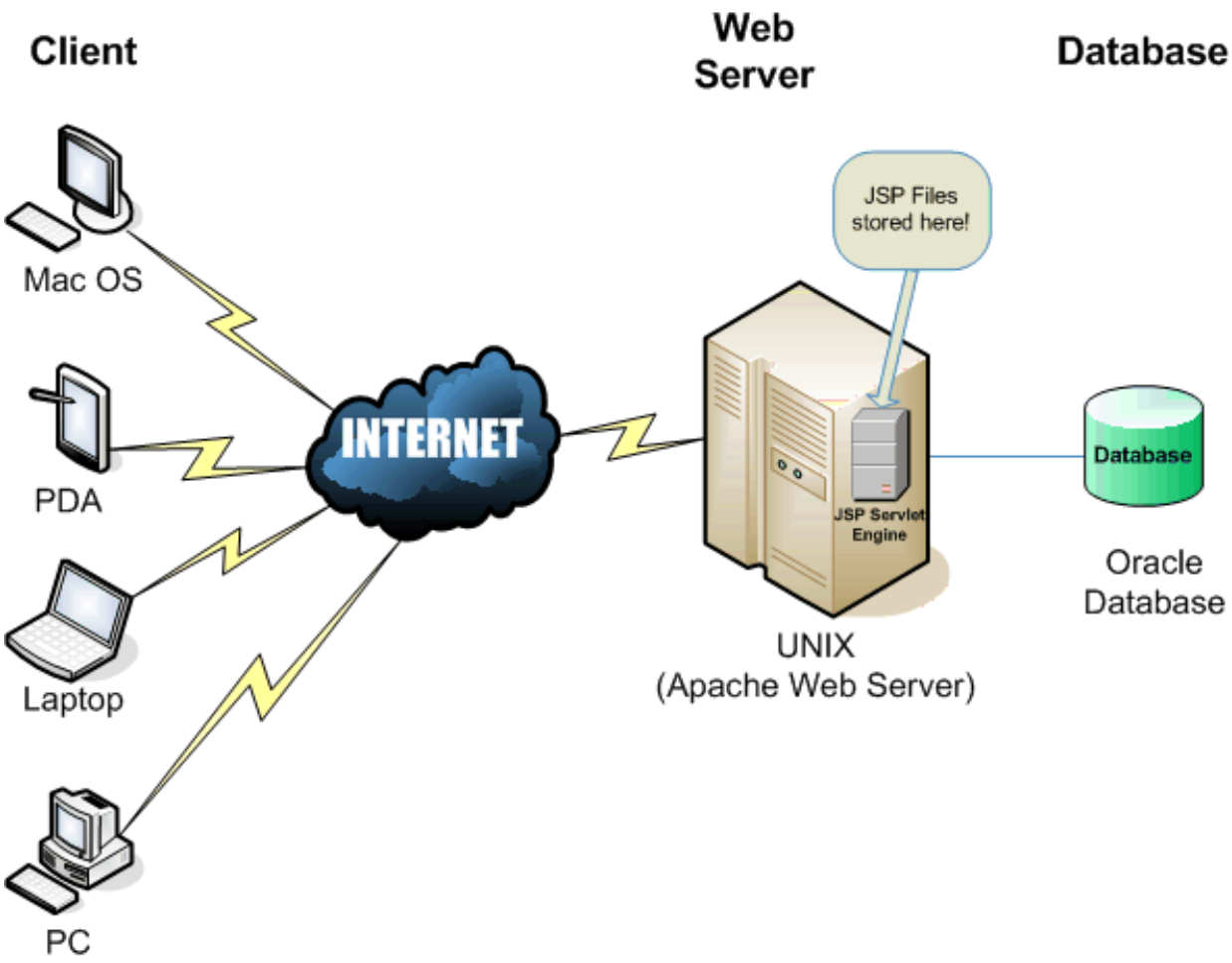
- Se añade la posibilidad de enviar datos al servidor (POST o GET) a través de formularios.
- Dos estrategias:
  - En el servidor:
    - Ejecución de programa en el servidor al que se le pasa los datos del formulario, y cuya salida se envía al cliente: CGI, *servlets* de Java, lenguajes embebidos (PHP, JSP, ASP, etc.)
  - En el cliente:
    - Además de páginas, imágenes, videos, etc. transferencia de aplicaciones para el navegador Web: *applets* de Java, flash, Adobe AIR, Microsoft Silverlight, etc.
    - Ejecución en el navegador Web del cliente de ciertas operaciones (libera al servidor de parte de la carga)

Ejemplo de la generación 2: *Servlet*

- La palabra *servlet* se deriva de la anterior *applet:*
  - Un *applet* es un programa en Java que se ejecutan en el navegador Web.
  - Un *servlet* es un programa que se ejecuta en un servidor Web.
- Un *servlet* permite generar páginas Web dinámicas a partir de los parámetros de la petición que envíe el navegador web.
- Los *servlets* forman parte de J2EE (*Java 2 Enterprise Edition*), que es una ampliación de J2SE (*Java 2 Standard Edition*).
- Un *servlet* es un objeto Java que implementa la interfaz javax.servlet.Servlet o hereda para algún protocolo específico (ej: javax.servlet.HttpServlet).
- Un *servlet* es un objeto que se ejecuta en un servidor o contenedor J2EE.

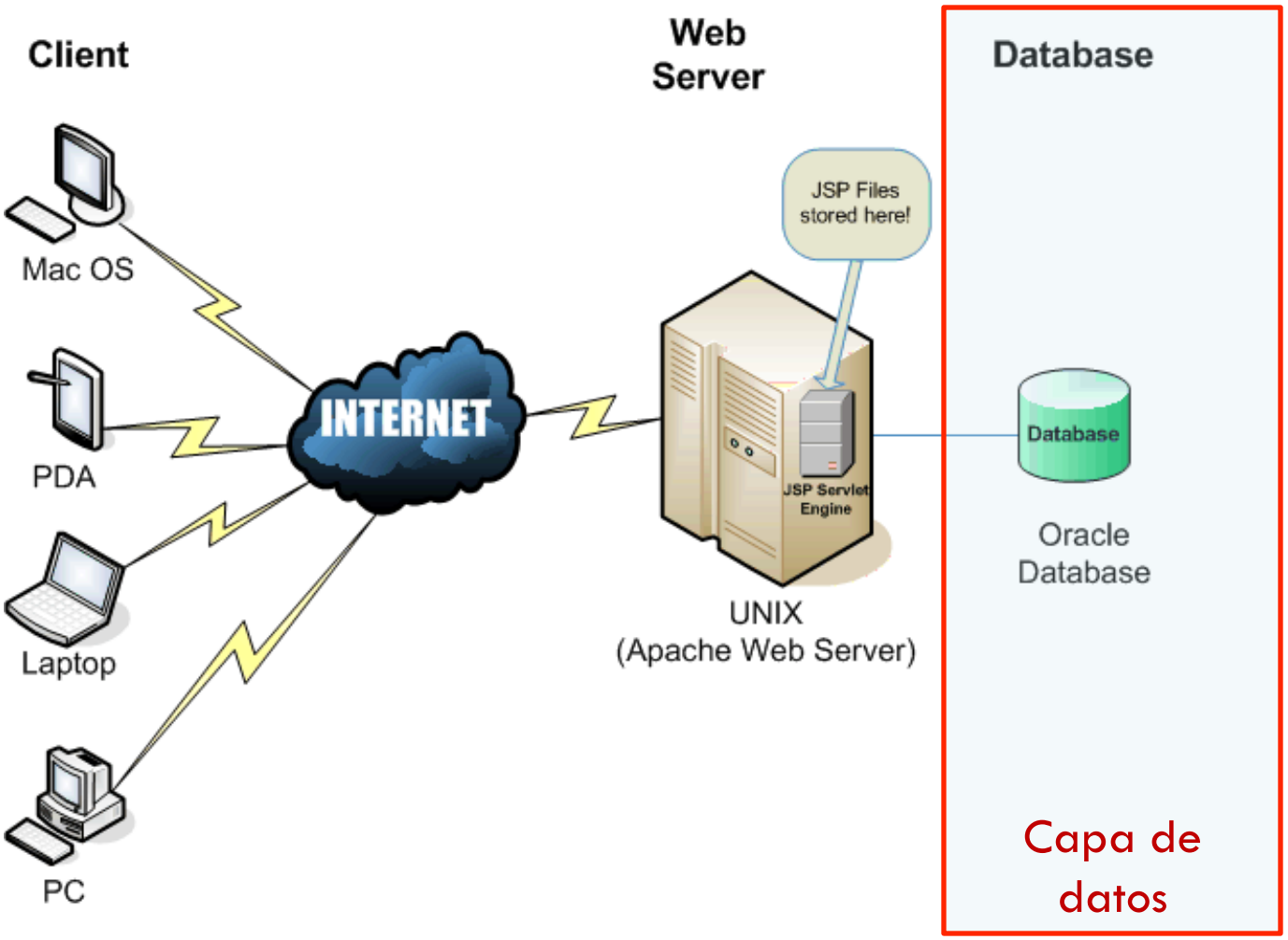
http://profesores.elo.utfsm.cl/~agv/elo323/2s06/projects/CastilloGonzalez/Proyecto_RII_Tomcat.html

Arquitectura en tres capas (*3-tier*)

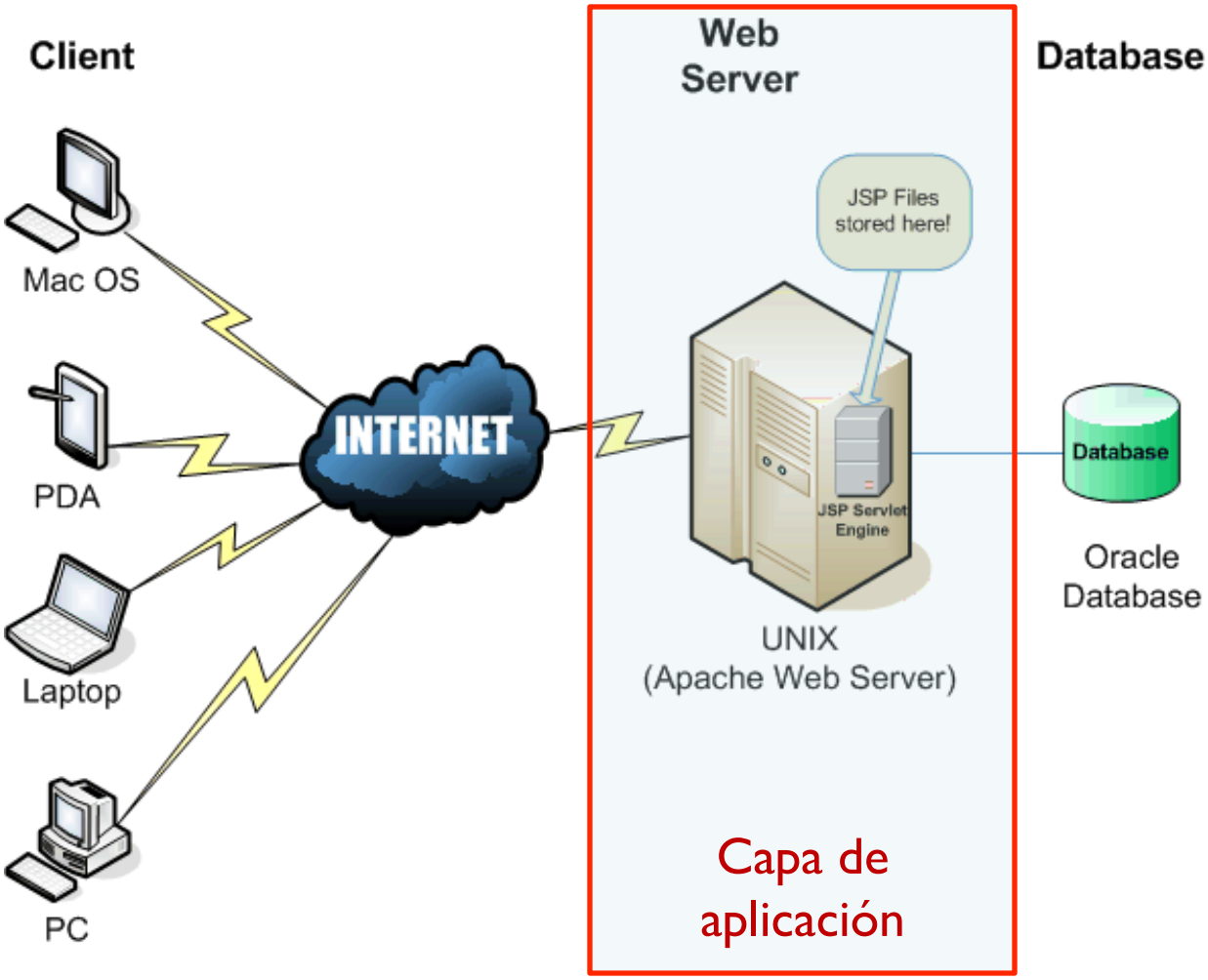


http://www.visualbuilder.com/jsp/tutorial/introduction-to-jsp/

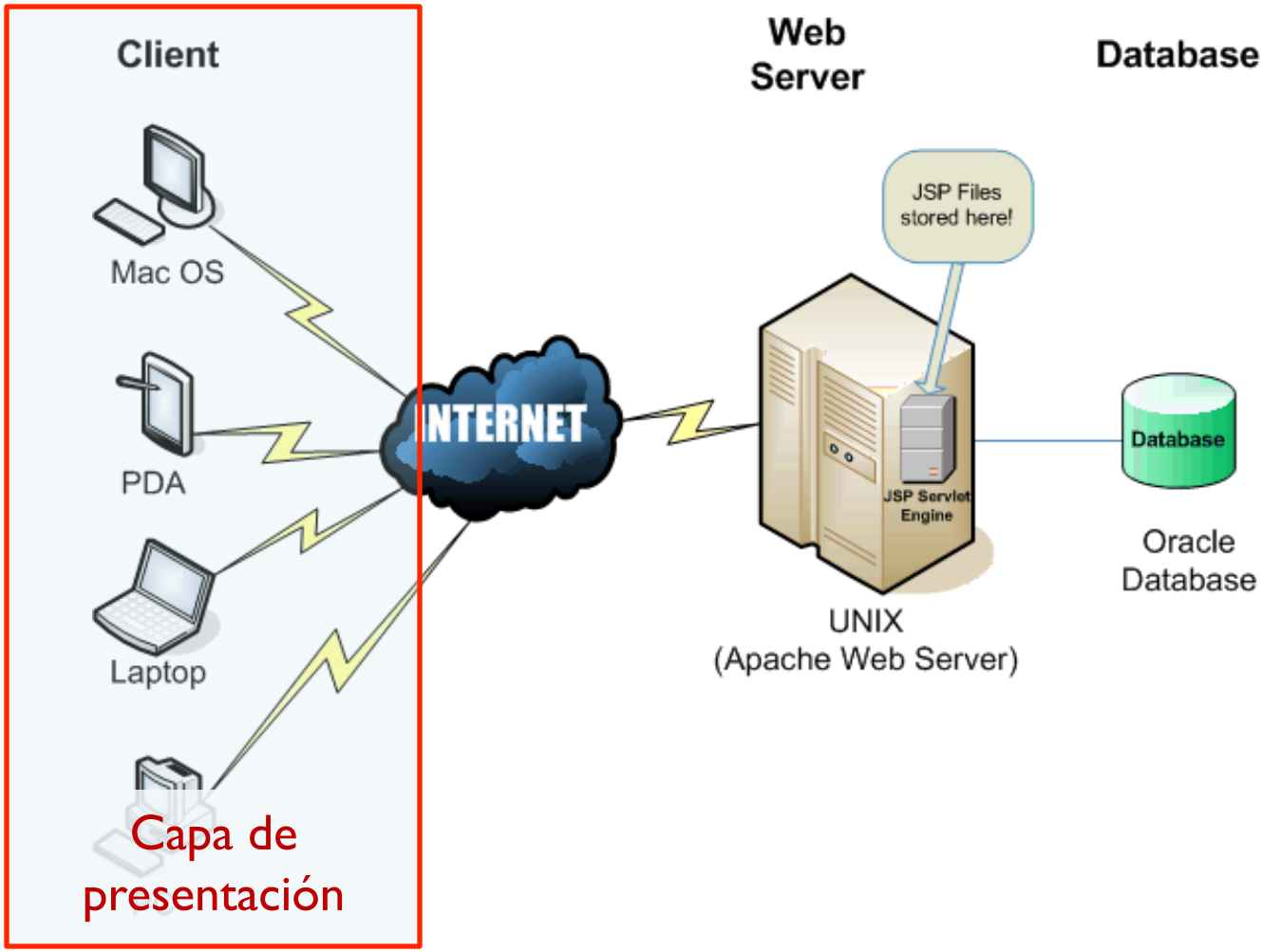
Arquitectura en tres capas (*3-tier*)



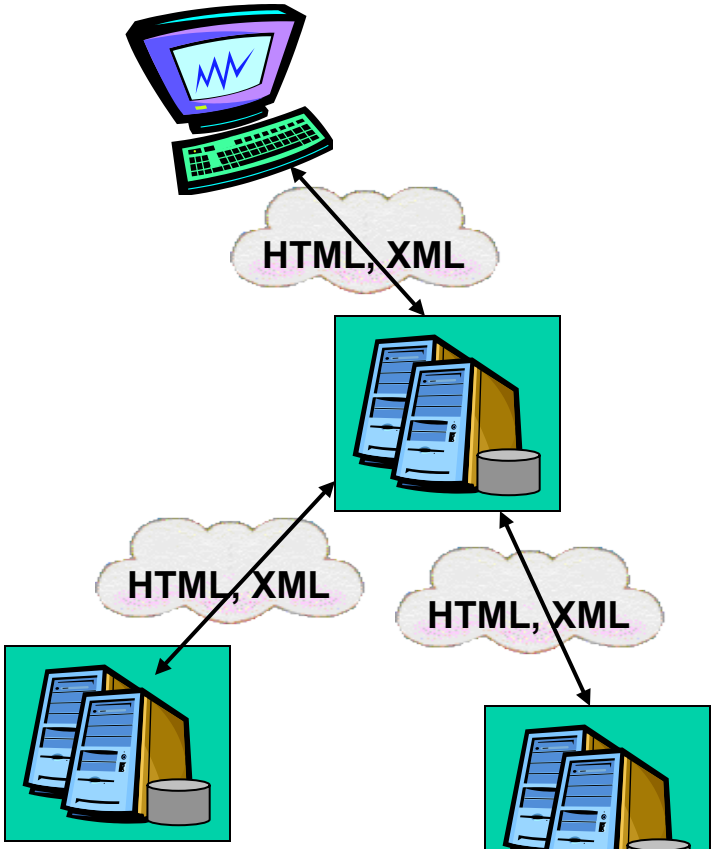
Arquitectura en tres capas (*3-tier*)



Arquitectura en tres capas (*3-tier*)

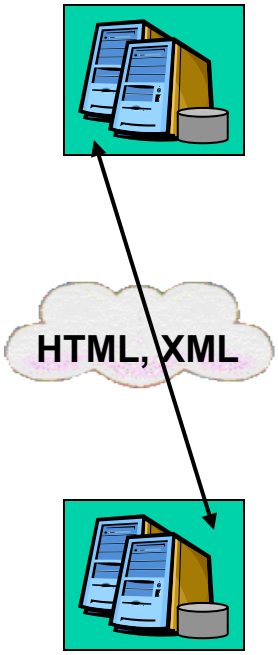


Evolución de la Web...



- Aparece b2b (*business to business*)
  - Necesidad de comunicar procesos de empresas sobre internet
    - Ej.: agencia de viaje que reserva avión y hotel
- Problema de la segunda generación:
  - Muy diversas tecnologías:
    - *Applets*, CGI, Lenguajes de *Scripts*, etc.
  - Desarrollos muy centrados en la interacción con la persona.
  - Por seguridad, los cortafuegos (*firewalls*) de muchas empresas solo dejan pasar tráfico HTTP (puerto 80) y cierran el resto:
    - Dificultad para usar Java RMI o CORBA
- Tercera generación: servicios Web

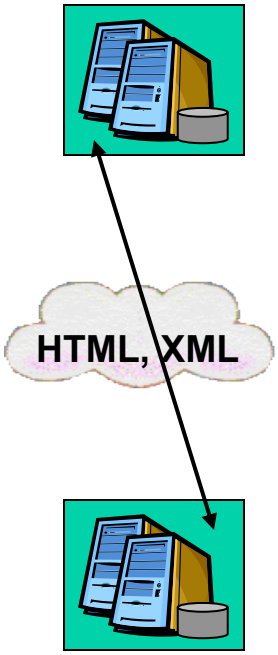
Servicio Web



- Un servicio web (en inglés, *Web Service*) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones en redes de ordenadores como Internet.
  - Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos.
  - La interoperabilidad se consigue mediante la adopción de estándares abiertos.
- Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web.

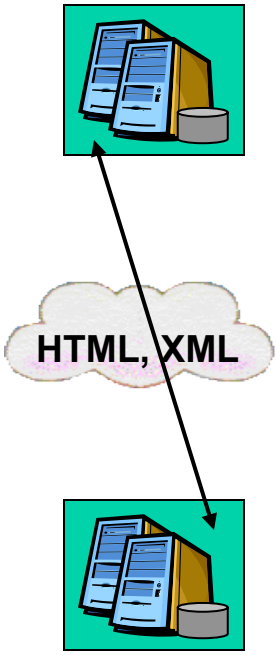
http://es.wikipedia.org/wiki/Servicio_web

Servicio Web



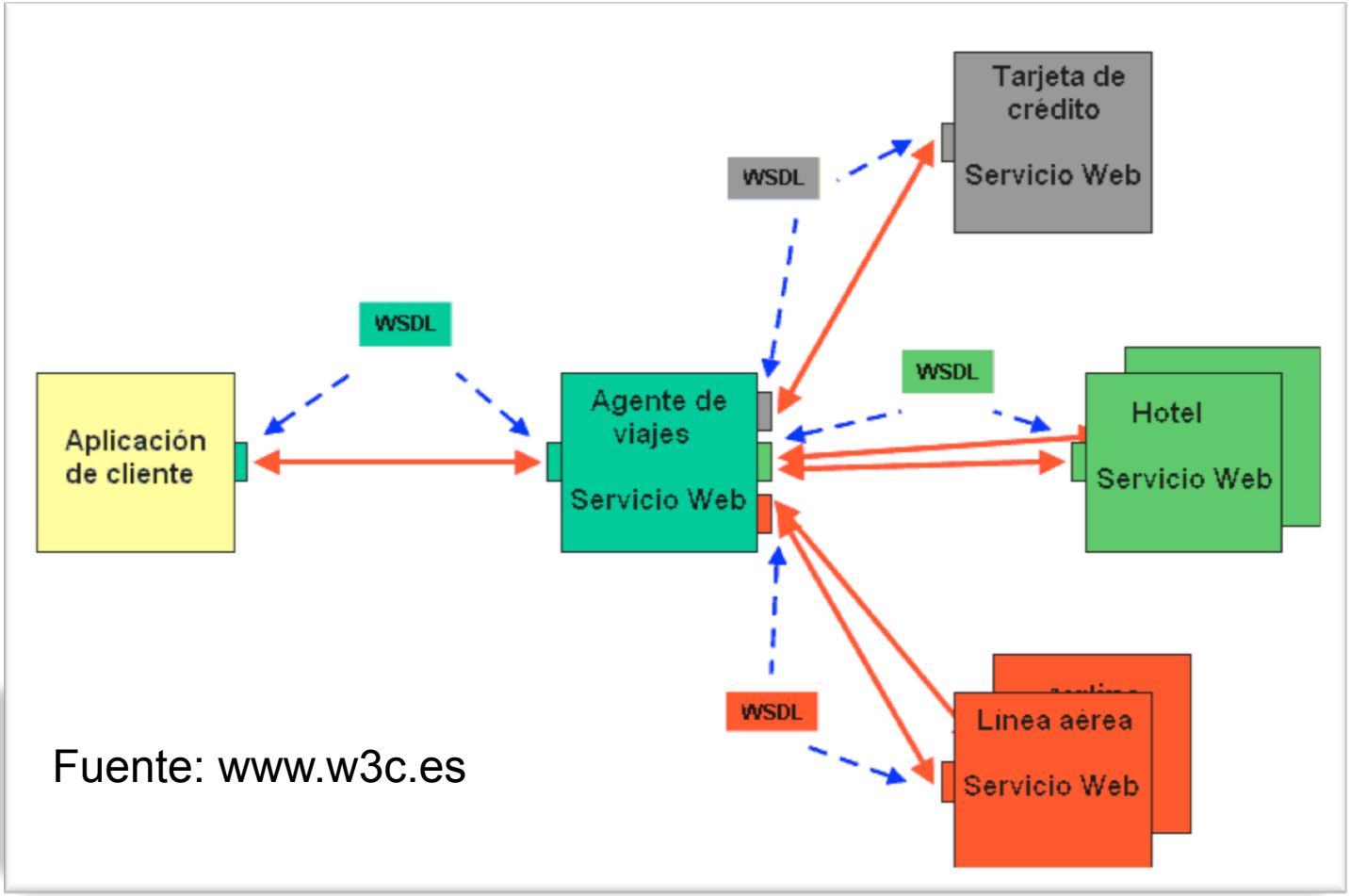
- Principales protocolos usados:
  - HTTP: transporte utilizado
  - SOAP: empaqueta la información y la transmite entre el cliente y el proveedor del servicio
  - XML: describe la información, los mensajes
  - UDDI: lista de servicios disponibles
  - WSDL: descripción del servicio

Servicio Web



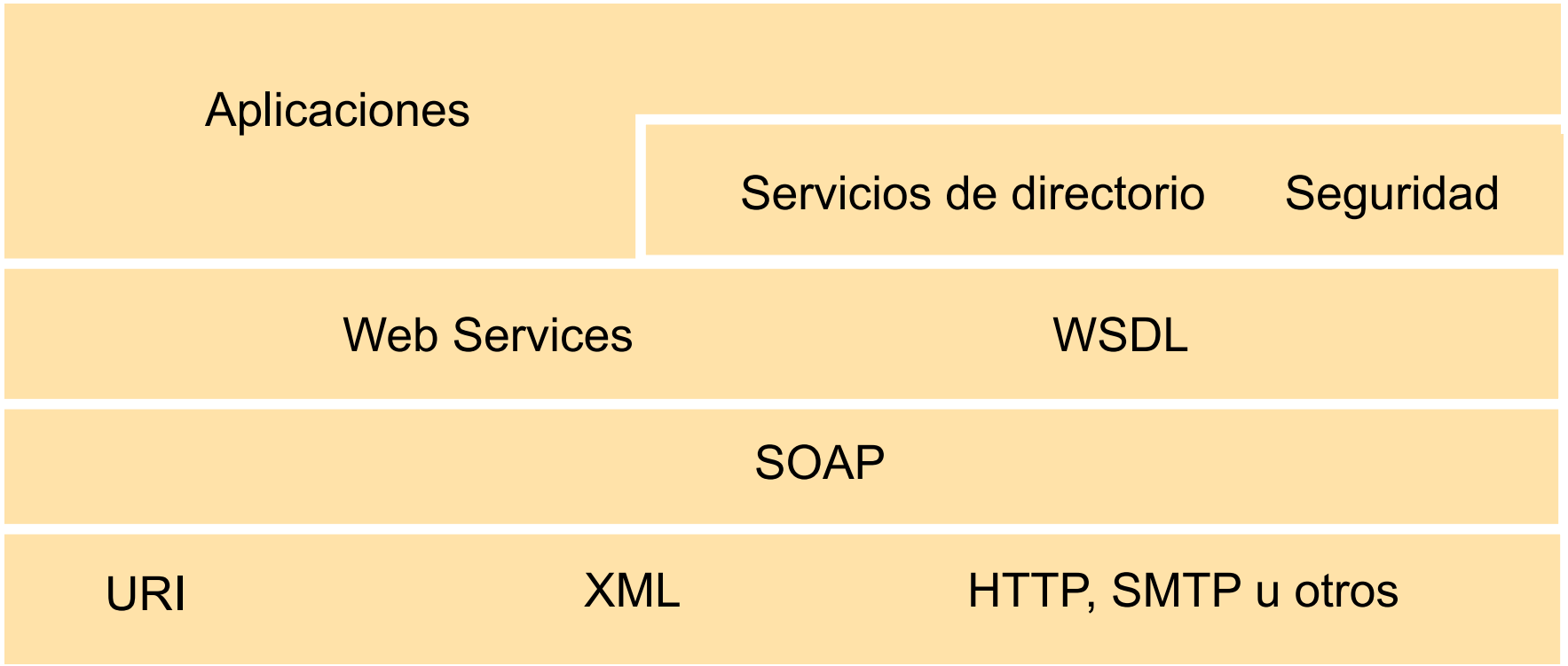
- Ventajas:
  - Paso de cortafuegos
    - Difícil en otros entornos como Java RMI o CORBA
  - Interoperabilidad
  - Compatibilidad
    - Especificaciones abiertas
    - Implementaciones compatibles a priori
- Inconvenientes:
  - HTTP es un protocolo simple y sin estado, por lo que no dispone de servicios de apoyo.
    - Ej.: servicios de transacciones mejor en CORBA.
  - Rendimiento es más bajo que otras soluciones.
    - Ej.: mandar datos binarios comparado con RMI, CORBA o DCOM.
    - Preciso conversión a XML, lo que añade una mayor sobrecarga.
  - Potenciales problemas de seguridad.
    - Dado que los firewall dejan pasar el tráfico HTTP, puede ser preciso asegurar el acceso a los servicios.

Combinación de servicios Web

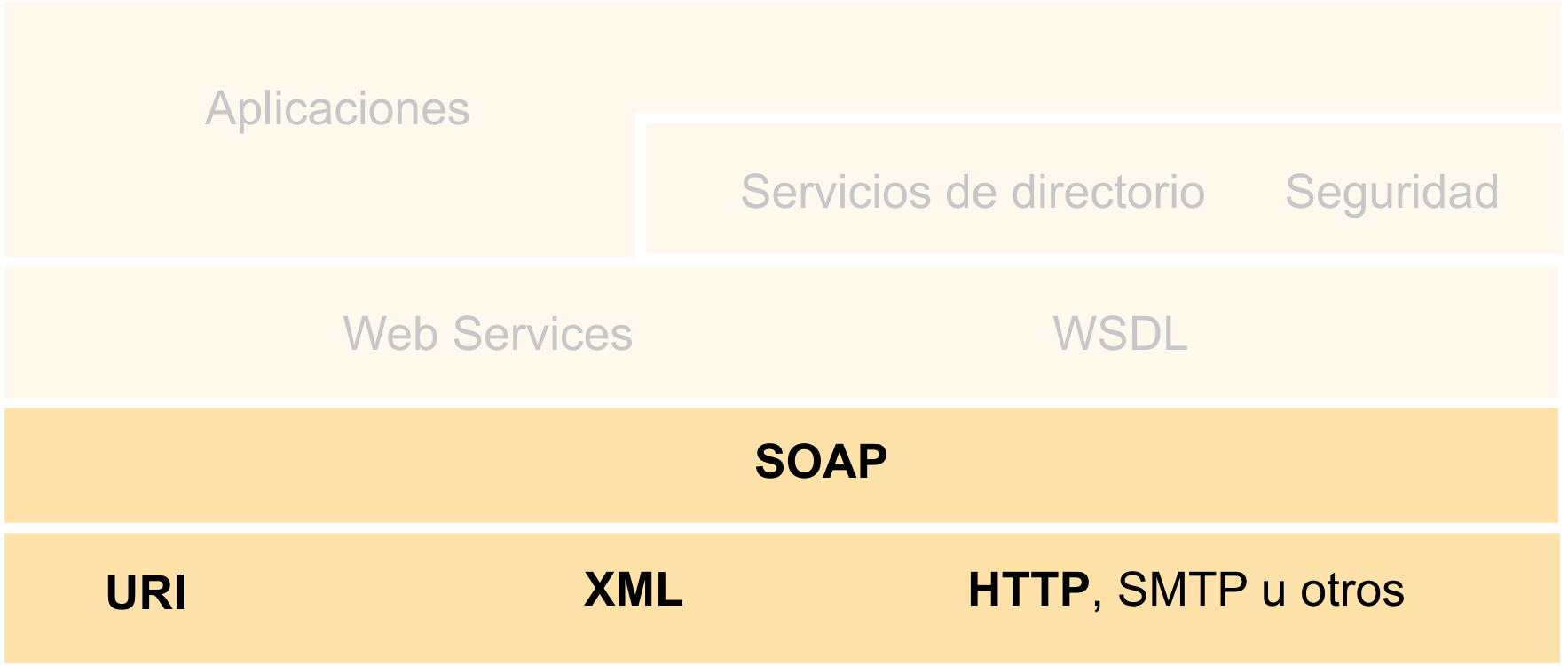


Fuente: www.w3c.es

Componentes e infraestructura



Componentes e infraestructura



Contenidos

1. Introducción:
   1. Paradigma de servicios de red
2. **SOAP**
   1. Introducción
   2. **Arquitectura**
   3. Ejemplo de aplicación
      - Desarrollo de un servicio privado

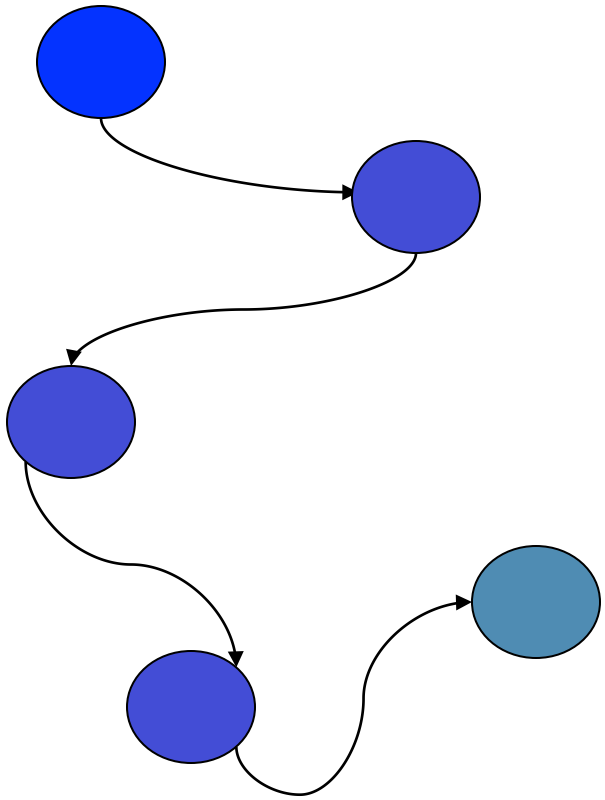
SOAP

- *Simple Object Accces Protocol*
  - http://*www.w3.org*

- SOAP especifica:
  - Cómo representar los mensajes en XML
  - Como combinar mensajes SOAP para un modelo petición-respuesta
  - Cómo procesar los elementos de los mensajes
  - Cómo utilizar el transporte (HTTP, SMTP, …) para enviar mensajes SOAP

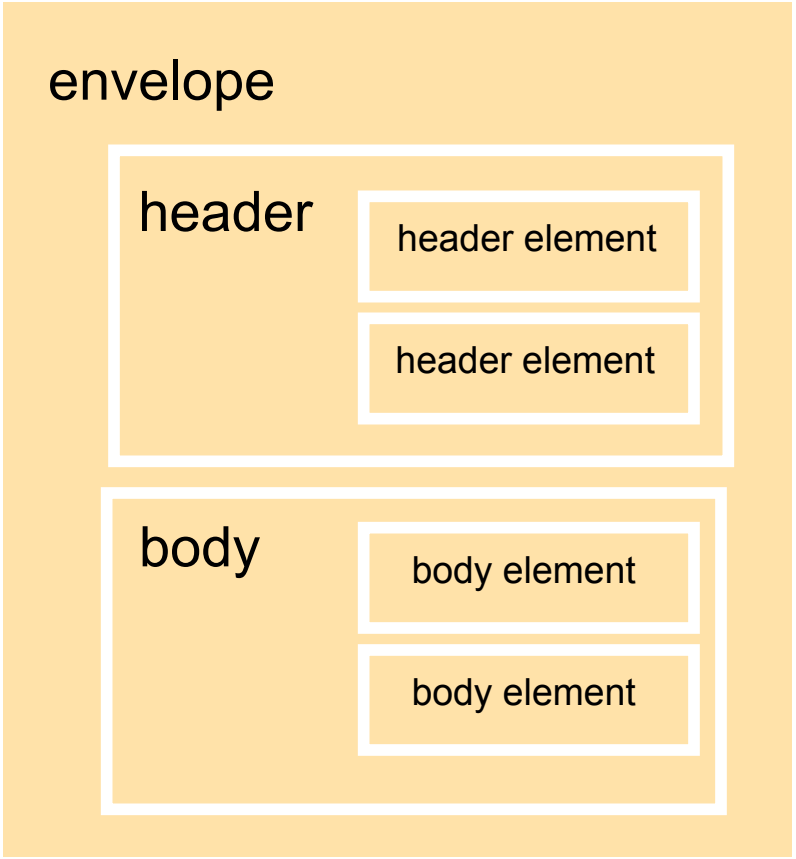
Nodo SOAP

- Nodo que transmite, recibe, procesa y responde un mensaje SOAP
- Tipos de nodo:
  - Emisor SOAP
  - Receptor SOAP
  - Intermediario



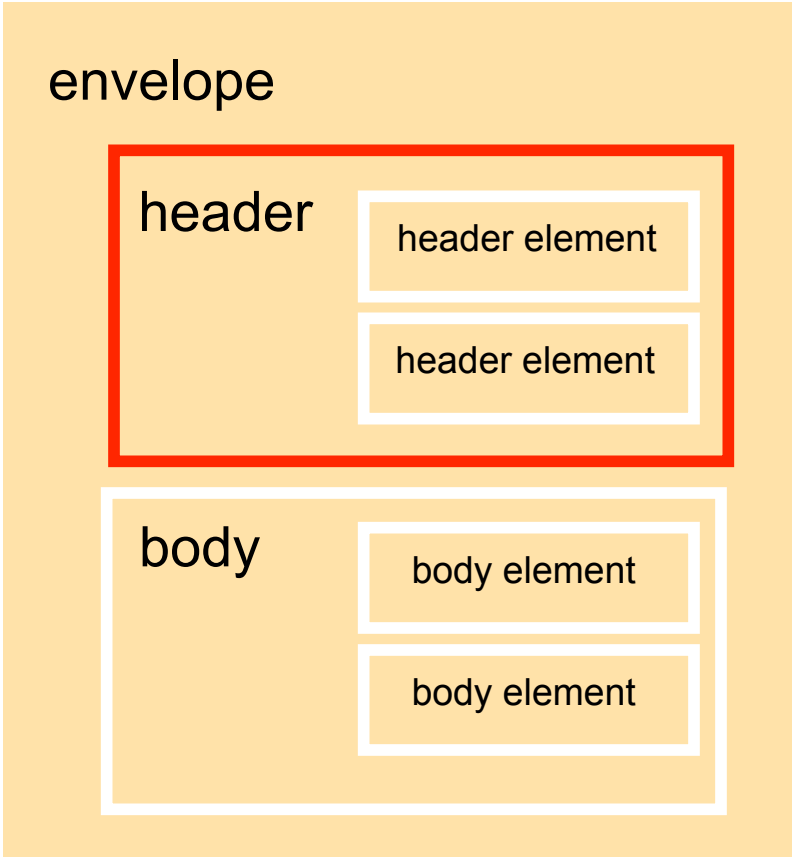
Mensaje SOAP

- Unidad básica de comunicación entre nodos SOAP
- El mensaje es transportado en un *envelope*
  - Encabezado opcional
  - Cuerpo
- Los elementos XML anteriores son definidos como un esquema en el espacio de nombres XML
  - Esquema definido en http://www.w3.org



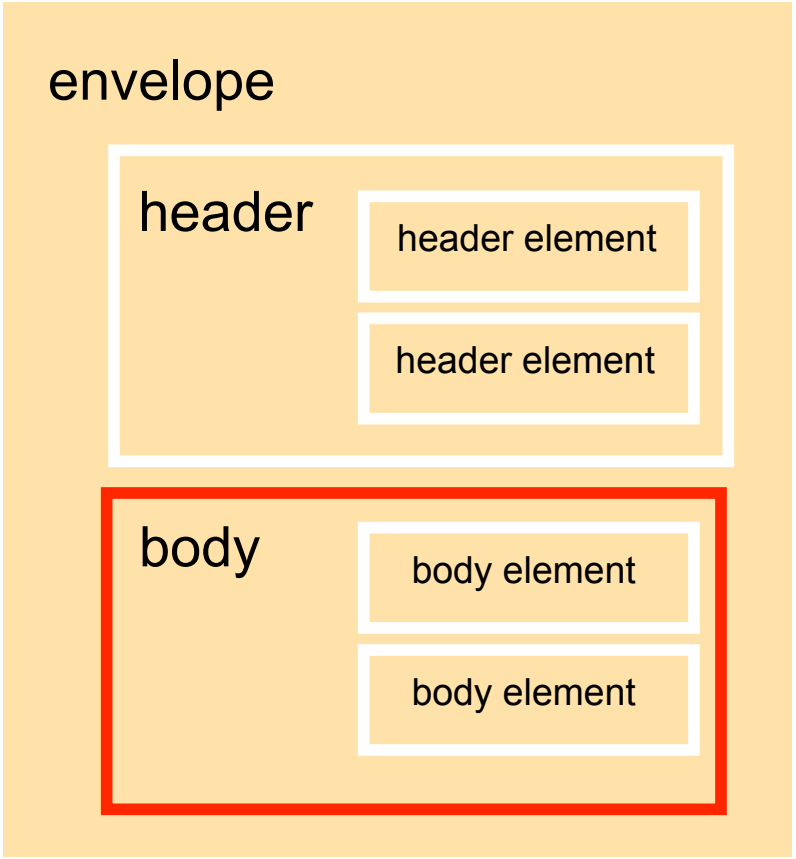
Mensaje SOAP: encabezado

- Elemento opcional
- Incluye información de control:
  - Identificador de transacción para su uso con un servicio de transacciones
  - Un identificador de mensajes para relacionar mensajes entre sí
    - Los servicios son autónomos e independientes entre sí
  - Un nombre de usuario, una clave pública, etc.



Mensaje SOAP: cuerpo

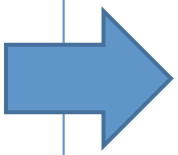
- Incluye la información:
  - Mensaje
  - Referencia al esquema XML que describe el servicio
- En los mensajes de una comunicación cliente/ servidor (RPC):
  - El elemento *body* contiene una *petición* o una *respuesta*.



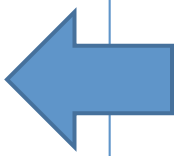
Serialización en XML

```
Float precio;

Precio=ObtenerPrecio(mesa);
```



```
<ObtenerPrecio>
  <item>mesa</item>
</ObtenerPrecio>
```



```
<ObtenerPrecioResponse>
  <precio>134.5</precio>
</ObtenerPrecioResponse>
```

Transporte de mensajes SOAP

- Protocolo HTTP
  - Estilo RPC:
    - Petición: en HTTP POST
    - Respuesta: en la respuesta al POST
  - Envío de información:
    - Con HTTP POST
    - Con HTTP GET
- Protocolo SMTP
  - La especificación indica cómo encapsular mensajes SOAP en mensajes con el formato usado en SMTP
    - Ejemplo: grandes volúmenes de datos binarios

Contenidos

1. Introducción:
   1. Paradigma de servicios de red
2. **SOAP**
   1. Introducción
   2. Arquitectura
   3. **Ejemplo de aplicación**
      - **Desarrollo de un servicio privado**

Plataforma de desarrollo

- gSOAP
  - Conjunto de herramientas para el desarrollo de aplicaciones basadas en servicios Web en C/C++
  - http://www.cs.fsu.edu/~engelen/soap.html