

Servicios web



Introducción

Un servicio web es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de computadoras como Internet .

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. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.



Para que sirven?



Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar.



Un servicio web se suele definir como una unidad de aplicación capaz de ofrecer datos o servicios de procesamiento a otras aplicaciones informáticas.

Así, una aplicación podría ofrecer distintos servicios a otras aplicaciones. Las características de estos servicios son:

- Que se solicitan a través del web
- Que los resultados de su ejecución también se devuelven a la aplicación peticionaria a través del web
- Que se tramitan según un modelo "estandarizado".





http://www.

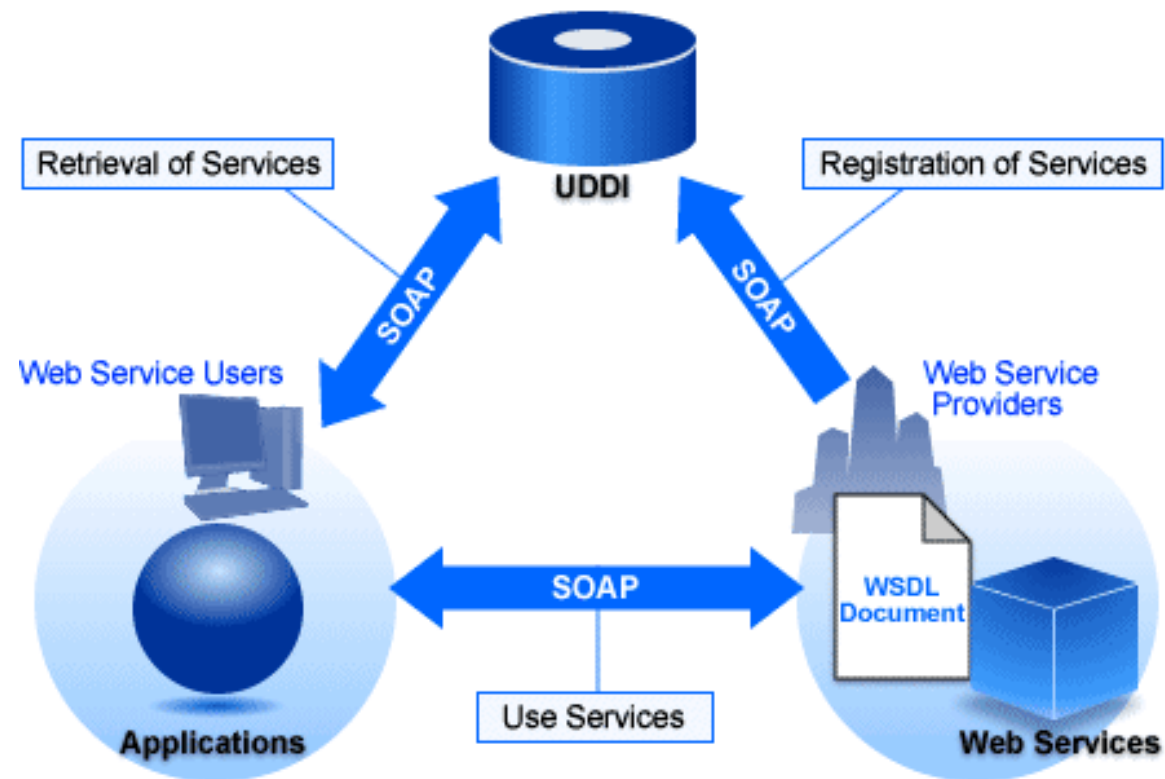
Estándares empleados
Web Services Protocol Stack: Así se denomina al conjunto de servicios y protocolos de los servicios Web.

Los servicios web se definen a partir de las siguientes especificaciones:

- SOAP (Simple Object Access Protocol)
- WSDL (Web Services Description Language)
- UDDI (Universal Description, Discovery and Integration)

Del mantenimiento de las dos primeras, *SOAP* y *WSDL* se encarga el *W3C*. En el caso de *UDDI*, se trata de un proyecto en el que participan distintas

La especificación *SOAP* indica cómo se deben codificar los mensajes que circularán entre las dos aplicaciones.



La especificación SOAP define dos modelos de mensajes:

-Un mensaje que se enviará desde la aplicación cliente a la aplicación servidor, solicitando la ejecución de un método al que se pasan una serie de parámetros.

-Un mensaje que se enviará desde la aplicación servidor a la cliente, y que contendrá datos XML con los resultados de la ejecución del método solicitado.



WSDL (Web Services Description Language)

Permite describir los distintos métodos o funciones que están disponibles en un servicio web, así como su signatura, es decir, el número de argumentos o parámetros que se les debe pasar, y el tipo de dato que devolverá la función como resultado.

Se establece una equivalencia entre el documento *WSDL* y un "contrato" que especifica los servicios que el servidor se compromete a ofrecer al cliente, siempre que éste los solicite de la forma adecuada



UDDI (Universal Description, Discovery and Integration)

A medida que el número de proveedores de servicios web aumente, será necesario disponer de un sistema de referencia que permita localizar estos servicios. Este es el propósito del proyecto *UDDI*

UDDI es un proyecto relacionado con los servicios web y que tiene importancia, si bien no es uno de los componentes básicos de la tecnología sobre la que se construye el paradigma de los servicios web



Ventajas de los servicios web

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad *firewall* sin necesidad



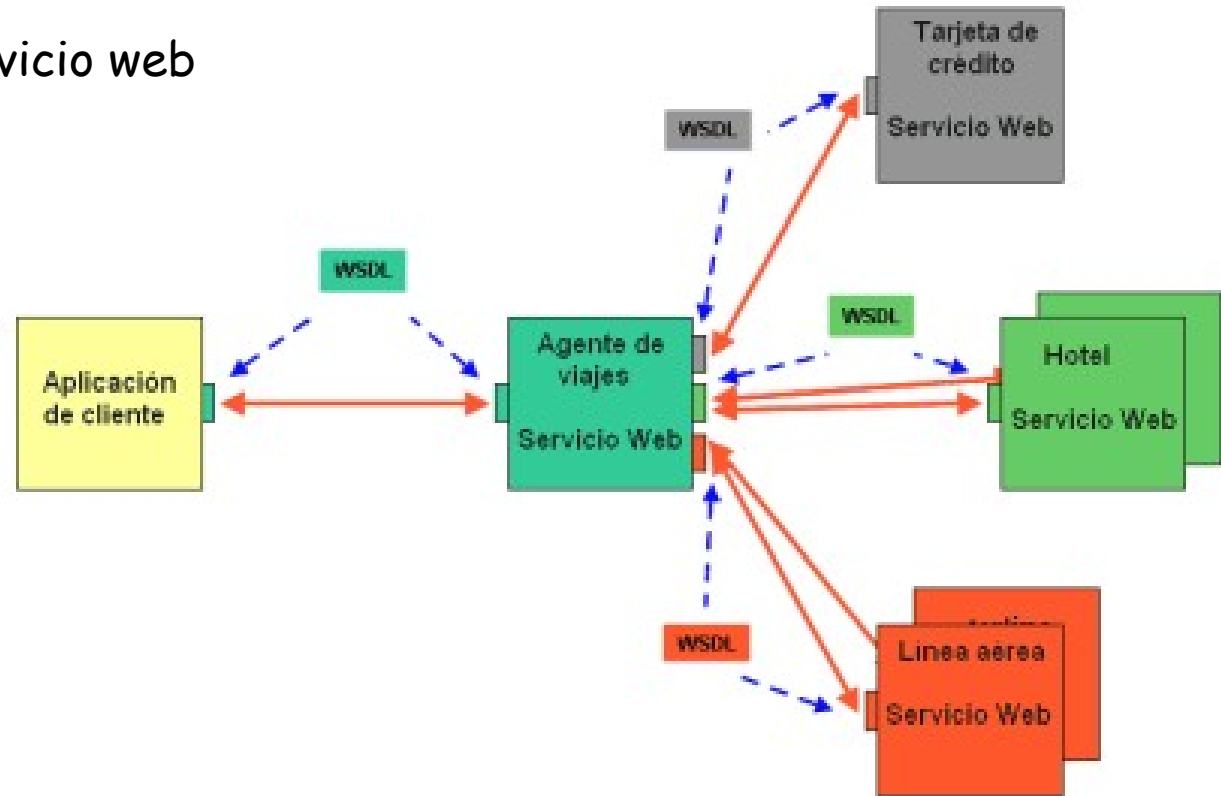
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar y abiertos.



Inconvenientes de los servicios Web

- Para realizar transacciones no pueden compararse en su grado de desarrollo con los estándares abiertos de computación distribuida como CORBA(Common Object Request Broker Architecture).
- Su rendimiento es bajo si se compara con otros modelos de computación distribuida, tales como RMI, CORBA o DCOM.
- 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.

Ejemplo de un servicio web



Según el ejemplo del gráfico, un usuario a través de una aplicación, solicita información sobre un viaje que desea realizar haciendo una petición a una agencia. La agencia de viajes ofrecerá a su cliente la información requerida. Para proporcionar al cliente la información que necesita, esta agencia de viajes solicita a su vez información a otros recursos en relación con el hotel y la línea aérea. La agencia de viajes obtendrá información sobre el hotel y la línea aérea.



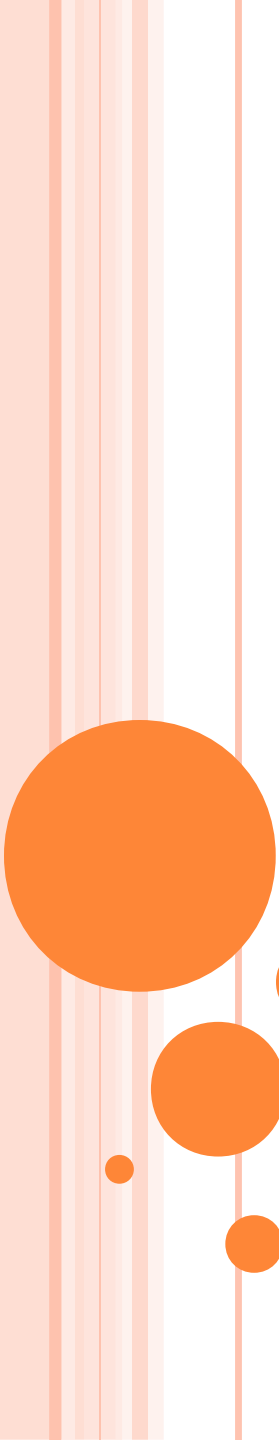
Razones para crear servicios Web

La principal razón para usar servicios Web es que se basan en HTTP sobre TCP (Transmission Control Protocol) en el puerto 80. Dado que las organizaciones protegen sus redes mediante *firewalls* -que filtran y bloquean gran parte del tráfico de Internet-, cierran casi todos los puertos TCP salvo el 80, que es, precisamente, el que usan los navegadores. Los servicios Web utilizan este puerto, por la simple razón de que no resultan bloqueados.




Una tercera razón por la que los servicios Web son muy prácticos es que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio.

De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro.



Esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más utilizada.

Se espera que para los próximos años mejoren la calidad y cantidad de servicios ofrecidos basados en los nuevos estándares



Gracias!

