

Servicios Web Distribuidos

M.C. Carlos Rojas Sánchez¹

¹Licenciatura en Informática
Universidad del Mar Campus Puerto Escondido

∴ SISTEMAS DISTRIBUIDOS ∴

Servicio Web

Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Servicio Web

Un web service es una vía de intercomunicación e interoperabilidad entre máquinas conectadas en Red. En el mundo de Internet se han popularizado enormemente, ya se trate de web services públicos o privados. Generalmente, la interacción se basa en el envío de solicitudes y respuestas entre un cliente y un servidor, que incluyen datos.

Definición

Servicio Web

World Wide Web Consortium (W3C)

Un servicio web es un sistema de software diseñado para admitir la interacción interoperable de máquina a máquina a través de una red. Tiene una interfaz descrita en un formato procesable por máquina (específicamente WSDL). Otros sistemas interactúan con el servicio web de la manera prescrita por su descripción usando mensajes SOAP, típicamente transmitidos usando HTTP con una serialización XML junto con otros estándares relacionados con la web.

Conceptos básicos

URL Localizador Uniforme de Recursos

UDDI Descripción, descubrimiento e integración universales

WSDL Lenguaje de definición de servicios web

SOAP Protocolo de acceso a un objeto simple - (XML-RPC)

REST Transferencia de estado representacional

Pila de protocolos de los WebServices

- UDDI - ofrece un directorio de servicios
- WSDL - ofrece un modo de definir los servicios
- SOAP - Permite invocar métodos de los servicios
- XML - Permite enviar y recibir mensajes a y de los servicios
- HTTP,TPC ... - Protocolos abiertos de internet

Arquitectura de un Servicio Web

- Un servicio web simple está caracterizado por cuatro estándares: XML, SOAP, UDDI y WSDL, los cuales al trabajar juntos proporcionan una funcionalidad básica de tipo “solicitud/ respuesta”.
- Los servicios web simples pueden utilizarse para entregar de forma eficiente información como noticias, inventarios y reportes de clima a los sitios web, mientras que un servicio web complejo puede involucrar transacciones más elaboradas entre varias partes, involucrando socios de negocios o proveedores y basado en los estándares de e-bussines.

Arquitectura

- Utilizando servicios web, un programador puede emplear cualquier lenguaje que desee, y un consumidor de servicios web puede usar http estándar para invocar métodos en los tipos definidos en el servicio web.

Arquitectura



Patrón de interacción y arquitectura de servicio web

Arquitectura

- 1** El Proveedor del Servicio genera el contrato de integración WSDL, donde se describirá la especificación del Servicio Web. Registra este WSDL en el directorio UDDI o Registro del Servicio.
- 2** Un consumidor del servicio solicita un Servicio Web, a través del UDDI lo localiza en el Registro de Servicio y dispone de su ubicación física.
- 3** El consumidor, o cliente del servicio, a través de la especificación del descriptor (WSDL) envía una solicitud (Request) para un servicio particular al Web Service Listener, que se encarga de recibir y enviar los mensajes en formato SOAP.
- 4** El Proveedor del servicio recibe el mensaje SOAP del consumidor y ejecuta la operación relacionada con dicha solicitud. El resultado de la misma se devuelve como mensaje SOAP al consumidor.
- 5** El consumidor recibe el SOAP con la respuesta y/o resultado de la operación y lo procesa.

Arquitectura

Al definir un servicio web, se requiere contar con lo siguiente:

1 Para colocar “disponible” el servicio

- 1** El primer paso es definir el servicio web (usando XML)
- 2** Una vez se ha definido la funcionalidad del servicio, es necesario publicarlo para que otros servicios y aplicaciones puedan acceder a él.

2 Para “localizar” el servicio

- 1** Cuando un consumidor desea acceder a un servicio web, debe contar con un servicio de descubrimiento, que permita conocer la ubicación exacta del servicio, es decir, se debe contar con un directorio donde se tengan listas las referencias a los servicios disponibles. Esto se logra gracias a UDDI.

Tecnologías de Servicios Web

- La primera introducción de los web services en el mundo de Internet vino de la mano de SOAP. SOAP es un protocolo que define cómo deben de realizarse las comunicaciones entre máquinas. SOAP usa XML como lenguaje de intercambio de datos con una estructura compleja que es capaz de albergar todo tipo de datos sobre la solicitud o respuesta generada.

SOAP

¿Qué es?

Es un formato de mensaje XML utilizado en interacciones de servicio web. Los mensajes SOAP habitualmente se envían sobre HTTP o JMS, pero se pueden utilizar otros protocolos. El uso de SOAP en un servicio web específico se describe mediante la definición WSDL. Hay dos versiones de SOAP en uso común: SOAP 1.1 y SOAP 1.2.

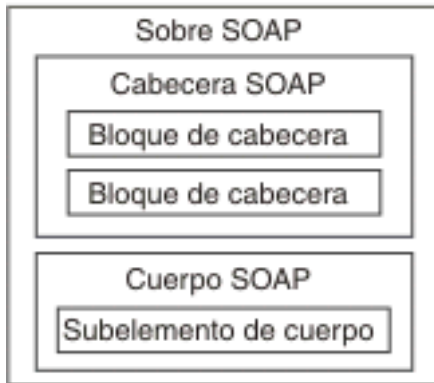
SOAP

La estructura de un mensaje SOAP

- Un mensaje SOAP se codifica como un documento XML, que consta de un<Envelope>, que contiene un elemento opcional<Header>y un elemento obligatorio<Body>. El <Fault> elemento, contenido en <Body>, se utiliza para notificar errores.

SOAP

La estructura de un mensaje SOAP



SOAP

La estructura de un mensaje SOAP

- I El sobre SOAP <Envelope> es el elemento raíz en cada mensaje SOAP y contiene dos elementos hijo, un valor opcional<Header> y un elemento obligatorio<Body>.
- II La cabecera SOAP <Header> es un subelemento opcional del sobre SOAP y se utiliza para pasar información relacionada con la aplicación que debe procesar los nodos SOAP a lo largo de la vía de acceso del mensaje.

SOAP

La estructura de un mensaje SOAP

- III El cuerpo SOAP <Body> es un subelemento obligatorio del sobre SOAP, que contiene información destinada al destinatario final del mensaje.
- IV El error SOAP <Fault> es un subelemento del cuerpo SOAP, que se utiliza para notificar errores.

SOAP

Ejemplo: Mensaje SOAP

```
<?xml version='1.0' Encoding='UTF-8' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <m:reservation xmlns:m="http://travelcompany.example.org/reservation"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
      <m:reference>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d</m:reference>
      <m:dateAndTime>2007-11-29T13:20:00.000-05:00</m:dateAndTime>
    </m:reservation>
    <n:passenger xmlns:n="http://mycompany.example.com/employees"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
      <n:name>Fred Bloggs</n:name>
    </n:passenger>
  </env:Header>
  <env:Body>
    <p:itinerary xmlns:p="http://travelcompany.example.org/reservation/travel">
      <p:departure>
        <p:departing>New York</p:departing>
        <p:arriving>Los Angeles</p:arriving>
        <p:departureDate>2007-12-14</p:departureDate>
        <p:departureTime>late afternoon</p:departureTime>
        <p:seatPreference>aisle</p:seatPreference>
      </p:departure>
      <p:return>
        <p:departing>Los Angeles</p:departing>
        <p:arriving>New York</p:arriving>
        <p:departureDate>2007-12-20</p:departureDate>
        <p:departureTime>mid-morning</p:departureTime>
        <p:seatPreference></p:seatPreference>
      </p:return>
    </p:itinerary>
  </env:Body>
</env:Envelope>
```

REST

- REST usa el propio protocolo HTTP para la comunicación entre máquinas. HTTP es ampliamente soportado por todos los sistemas y de hecho para la transferencia de datos en la web se usa HTTP.
- REST se caracteriza por no tener estado. Es decir, el servidor no es capaz de recordar el estado de la anterior solicitud REST que pudo, o no, hacer un cliente. Por ello, el cliente tiene que enviar en cada solicitud todo el estado de su sesión, lo que se suele hacer mediante un token que le «ayude a recordar» al servidor.

REST

Se apoya en los métodos básicos de HTTP, como son:

- Post: Para crear recursos nuevos.
- Get: Para obtener un listado o un recurso en concreto.
- Put: Para modificar.
- Patch: Para modificar un recurso que no es un recurso de un dato, por ejemplo.
- Delete: Para borrar un recurso, un dato por ejemplo de nuestra base de datos.

REST

Todos los objetos se manipulan mediante URI, por ejemplo, si tenemos un recurso usuario y queremos acceder a un usuario en concreto nuestra URI seria `/user/identificadordelobjeto`, con eso ya tendríamos un servicio USER preparado para obtener la información de un usuario, dado un ID.

XML vs JSON

- XML está basado en etiquetas, como HTML. Es más tradicional pero también es un lenguaje más avanzado, que presenta diversas utilidades para su extensión, validación de la información y sintaxis de los datos.
- JSON es un lenguaje más nuevo, basado en sintaxis Javascript. Generalmente, es más ligero y requiere mucho mejor carga del servidor para su procesamiento.

Computación en la nube

Computación en la nube

- La computación en la nube transforma la infraestructura de TI en una utilidad: le permite "conectarse" a la infraestructura a través de Internet y utilizar los recursos informáticos sin instalarlos ni mantenerlos en las instalaciones.

Computación en la nube

Computación en la nube es el acceso bajo demanda, a través de Internet, a recursos informáticos: aplicaciones, servidores (servidores físicos y Servidores virtuales), almacenamiento de datos, herramientas de desarrollo, capacidades de red y más, hospedadas de forma remota centro de datos gestionado por un servicios en la nube proveedor (o CSP). El CSP ofrece estos recursos en un plan de suscripción mensual o los factura según el uso.

Computación en la nube

El término "computación en la nube" también se refiere a la tecnología que hace que la nube funcione. Esto incluye alguna forma de infraestructura de TI; servidores, software de sistema operativo, redes y otra infraestructura que se abstrae mediante software especial, de modo que la TI se pueda agrupar y dividir independientemente de los límites físicos del hardware. Por ejemplo, un único servidor de hardware se puede dividir en varios servidores virtuales.

Computación en la nube

Comparación con la TI local tradicional

- Menores costos de TI: la nube le ayuda a minimizar algunos o la mayoría de los costos y el esfuerzo que implica comprar, instalar, configurar y gestionar su propia infraestructura local.
- Mejore la agilidad y el tiempo de obtención de valor: con la nube, su organización puede empezar a utilizar aplicaciones empresariales en minutos, en lugar de esperar semanas o meses para que TI responda a una solicitud, adquiera y configure el hardware e instale el software. La nube también le permite capacitar a ciertos usuarios, específicamente desarrolladores y científicos de datos, para acceder por sí mismos a la infraestructura de software y soporte.
- Escale más fácil y rentablemente: la nube le proporciona flexibilidad, ya que en lugar de adquirir una cantidad excesiva de recursos que no va a utilizar durante períodos lentos, puede aumentar o disminuir la capacidad en respuesta a alzas y caídas en el tráfico. También puede aprovechar la red global de su proveedor de nube para acercar sus aplicaciones a usuarios de todo el mundo.



Computación en la nube

Ventajas de la computación en nube

La computación en nube ofrece a las empresas modelos prácticos para acceder a las ofertas de infraestructura, plataforma y software de pago por uso. Con la computación en nube, las empresas pueden liberar capital, optimizar el mantenimiento de TI, modernizar y escalar los enfoques empresariales, convertir la seguridad y la flexibilidad en servicios y soluciones, ayudar a los clientes de nuevas maneras, y hacer crecer su empresa en las condiciones de mercado siempre cambiantes.

Tipos de modelos de implementación de computación en nube

Privada

El modelo de nube privada hace referencia a los recursos de computación en nube que se usan exclusivamente en una misma empresa u organización. La nube privada puede almacenarse en el centro de datos interno o un proveedor de servicios puede alojarla en la nube.

Tipos de modelos de implementación de computación en nube

Pública

Cuando una organización implementa un modelo de nube pública, toda la infraestructura de hardware, software y soporte es propiedad del proveedor de servicios que la administra y la proporciona exclusivamente por Internet. Puede acceder a estos servicios y administrar la cuenta mediante un navegador web.

Tipos de modelos de implementación de computación en nube

Pública

- Amazon Web Services (AWS)
- Google Cloud
- IBM Cloud
- Microsoft Azure
- Oracle Cloud

Tipos de modelos de implementación de computación en nube

Híbrida

Un modelo de computación en nube híbrido combina las nubes públicas y privadas para compartir datos y aplicaciones. Las nubes híbridas conectan la infraestructura y las aplicaciones entre recursos en nube con los recursos existentes que no se encuentran en la nube. Este enfoque le aporta a su empresa más flexibilidad y más opciones de implementación.

Tipos de servicios de computación en nube

Los servicios de computación en nube ofrecen modelos convenientes de pago por uso que eliminan los gastos y el mantenimiento costosos. Los proveedores de nube alojan una variedad de ofertas de infraestructura, plataforma y software en las instalaciones que ellos “alquilan”, lo que le aporta a la organización la flexibilidad de activar o desactivar los servicios de computación en nube en función de los requisitos cambiantes.

Tipos de servicios de computación en nube

Infraestructura como servicio (IaaS)

La IaaS son los componentes de la TI basada en la nube. En este modelo, un proveedor de nube aloja los componentes de infraestructura que tradicionalmente se almacenan en centros de datos internos. Por ejemplo, el hardware de servidores, almacenamiento y redes, junto con el hipervisor (capa de virtualización), suelen alojarse en las instalaciones. Con IaaS, su organización puede elegir cuándo y cómo desean administrar las cargas de trabajo sin necesidad de comprar, administrar y respaldar la infraestructura subyacente. IaaS permite que la infraestructura esté en funcionamiento rápidamente, con un modelo de pago por uso.

Tipos de servicios de computación en nube

Plataforma como servicio (PaaS)

PaaS se suma al modelo de IaaS, pero suele ser algo específico de las herramientas de hardware y software para el desarrollo de aplicaciones. Además de proporcionar componentes de infraestructura, los proveedores de nube también alojan y administran sistemas operativos y middleware que los desarrolladores necesitan para crear y ejecutar aplicaciones. PaaS ofrece un modelo a petición de pago por uso.

Tipos de servicios de computación en nube



Software como servicio (SaaS)

Con SaaS, los proveedores de nube alojan y administran toda la infraestructura, además de las aplicaciones para usuarios finales. Cuando su empresa elige un modelo de SaaS, no necesita instalar nada; los usuarios podrán iniciar sesión y comenzar a usar de inmediato la aplicación del proveedor de nube que se ejecuta en la infraestructura. SaaS lo libera para que pueda dedicarse únicamente a pensar en cómo usará su empresa dicho software, en vez de pensar en cómo mantenerlo. SaaS está disponible a petición o por suscripción.

Referencias

- https://www.cisco.com/c/es_mx/solutions/cloud/what-is-cloud-computing.html?ccid=cc000071
- <https://www.ibm.com/mx-es/topics/cloud-computing>

Bibliografía I

-  Andrew S. Tanenbaum and Maarten Van Steen.
Sistemas Distribuidos: Principios y Paradigmas .
Prentice Hall. 2008, 2ª Edición.
-  Coulouris, G., Dollimore, J., and Kindberg, T..
Distributed Systems: Concepts and Design
Addison-Wesley. 1994, 1ª Edición. 2ª Edición.