Explore SOA (arquitectura orientada a servicios), una etapa importante en la evolución del desarrollo y la integración de aplicaciones.

**¿Qué es SOA (arquitectura orientada a servicios)?**

SOA, o arquitectura orientada a servicios, define una forma de hacer que los componentes de software sean reutilizables a través de interfaces de servicio. Estas interfaces utilizan estándares de comunicación comunes de tal manera que pueden incorporarse rápidamente a nuevas aplicaciones sin tener que realizar una integración profunda cada vez.

Cada servicio en una SOA incorpora el código y las integraciones de datos requeridas para ejecutar una función comercial completa y discreta (por ejemplo, verificar el crédito de un cliente, calcular un pago mensual del préstamo o procesar una solicitud de hipoteca). Las interfaces de servicio proporcionan un acoplamiento flexible, lo que significa que pueden llamarse con poco o ningún conocimiento de cómo se implementa la integración debajo. Los servicios se exponen utilizando protocolos de red estándar, como SOAP (protocolo simple de acceso a objetos) / HTTP o JSON / HTTP, para enviar solicitudes para leer o cambiar datos. Los servicios se publican de una manera que permite a los desarrolladores encontrarlos rápidamente y reutilizarlos para ensamblar nuevas aplicaciones.

Estos servicios se pueden construir desde cero, pero a menudo se crean exponiendo funciones de sistemas de registro heredados como interfaces de servicio.

De esta manera, SOA representa una etapa importante en la evolución del desarrollo y la integración de aplicaciones en las últimas décadas. Antes de que SOA surgiera a fines de la década de 1990, la conexión de una aplicación a datos o funcionalidades alojadas en otro sistema requería una integración compleja punto a punto, integración que los desarrolladores tuvieron que recrear, en parte o en su totalidad, para cada nuevo proyecto de desarrollo. Exponer esas funciones a través de SOA elimina la necesidad de recrear la integración profunda cada vez.

Hay que tener en cuenta que, aunque SOA y la arquitectura de microservicios más reciente comparten muchas palabras en común, solo están poco relacionadas y, de hecho, operan en diferentes ámbitos.

**¿Qué es un ESB?**

Un ESB, o bus de servicios empresariales, es un patrón mediante el cual un componente centralizado realiza la integración a los sistemas de fondo y luego hace que esas integraciones estén disponibles como interfaces de servicio. Realiza la traducción de modelos de datos, conectividad profunda, enrutamiento y, potencialmente, composición de múltiples solicitudes, y las pone a disposición como una única interfaz de servicio para su reutilización por nuevas aplicaciones. El patrón ESB generalmente se implementa utilizando un tiempo de ejecución de integración especialmente diseñado y herramientas que se adaptan bien a las capacidades anteriores, asegurando la mejor productividad posible.

En teoría, podría implementar un SOA sin un ESB, pero los propietarios de las aplicaciones tendrían que encontrar su propia forma única de exponer las interfaces de servicio, lo cual es mucho trabajo (incluso si las interfaces son finalmente reutilizables) y crea un desafío de mantenimiento significativo en el futuro. De hecho, los ESB fueron, eventualmente, considerados un elemento de facto de cualquier implementación SOA que los dos términos a veces se usan como sinónimos, creando confusión.

**Beneficios de SOA:**

En comparación con las arquitecturas que lo precedieron, SOA ofrece beneficios significativos para la empresa:

* Mayor agilidad empresarial; tiempo de comercialización más rápido: la eficiencia de ensamblar aplicaciones a partir de interfaces de servicio reutilizables, en lugar de reescribir y reintegrar con cada nuevo proyecto de desarrollo, permite a los desarrolladores crear aplicaciones mucho más rápidamente en respuesta a nuevas oportunidades de negocios.
* Capacidad para aprovechar la funcionalidad heredada en nuevos mercados: una SOA bien diseñada permite a los desarrolladores tomar fácilmente la funcionalidad 'bloqueada' en una plataforma o entorno informático y extenderla a nuevos entornos y mercados. Por ejemplo, muchas empresas han utilizado SOA para exponer la funcionalidad de los sistemas financieros basados en mainframe a la web, lo que permite a sus clientes servir a procesos e información a los que antes solo se podía acceder mediante la interacción directa con los empleados o socios comerciales de la compañía.
* Mejor colaboración entre la empresa y la TI: en una SOA, los servicios se pueden definir en términos comerciales (por ejemplo, "generar cotización de seguro" o "calcular el ROI de los equipos de capital"). Esto permite a los analistas de negocios trabajar de manera más efectiva con los desarrolladores en ideas importantes, como el alcance de un proceso comercial definido por un servicio o las implicaciones comerciales de cambiar un proceso, que pueden conducir a un mejor resultado.

Ejemplos de SOA:

* Delaware Electric recurrió a SOA para integrar sistemas que anteriormente no se comunicaban entre sí, lo que resultó en eficiencias de desarrollo que ayudaron a la organización a mantenerse solvente durante una congelación obligatoria de cinco años en las tarifas eléctricas.
* Cisco adoptó SOA para asegurarse de que su experiencia en pedidos de productos fuera consistente en todos los productos y canales al exponer los procesos de pedidos como servicios que las divisiones, adquisiciones y socios comerciales de Cisco podrían incorporar en sus sitios web.
* Independence Blue Cross (IBC) de Filadelfia implementó una SOA para garantizar que los diferentes componentes que se ocupan de los datos del paciente (agentes de servicio al cliente de IBC, consultorios médicos, usuarios del sitio web de IBC) trabajaban con la misma fuente de datos (una 'versión única del verdad').

**SOA e IBM Cloud:**

A medida que su empresa cambia su infraestructura de TI hacia un enfoque de nube híbrida, existe una alta probabilidad de que trabaje con una variedad de cargas de trabajo, incluidas aquellas basadas en SOA. IBM es uno de los pioneros de SOA, y las ofertas y servicios de IBM Cloud pueden aprovechar y ampliar sus inversiones SOA existentes.

Con la adopción rápida de SOA, surgen los siguientes retos clave:

* Idoneidad: Las pilas complejas de middleware que soportan servicios web, no son fáciles de mantener a lo largo del tiempo, debido a sus interdependencias. Lograr la idoneidad mientras de retiene la sostenibilidad no es así de fácil.
* Seguridad: Como mostrado arriba, los servicios web son usuarios pesados de XML, y existen muchas formas diferentes de ataque conocidas para XML hoy.
* Rendimiento: Desde una perspectiva computacional, el proceso de datos representado en XML es más costoso que las representaciones de datos en tipos de datos nativos. De esta manera, la conversión de XML para tipos de datos nativos entendidos por los ciclos de procesamiento de costos de la CPU. Poniendo en la cima cantidades constantemente en crecimiento de datos que necesitan ser procesados e intercambiadas entre sistemas, agrega de manera medible un valor a la demanda de más energía y velocidad de la computación.

**Capas de SOA:**

Capa de servicios de datos:

La capa de servicios de datos (DSL) proporciona una capa para el acceso de datos que es independiente del esquema físico. La finalidad de la capa de servicios de datos es proporcionar una interfaz coherente (llamada fachada de servicios de datos) para acceder a datos, independiente de la infraestructura de correlación de objetos relacionales (como EJB, DAS o JPA). A su vez, la infraestructura de correlación extraída se utiliza para transformar los datos recuperados de la base de datos en una colección de objetos Java. Esos objetos se implementan como objetos de datos de servicio físico (SDO).

Capa de componentes de servicio (Service Components Layer):

Es la implementación de un servicio y según IBM debe cumplir con SCA (Service Componente Architecture) y con SDO (Service Data Objects).

Personalmente, creo que este es un punto flaco de esta arquitectura. ¿qué pasa si la lógica de la capa operacional ya se ofrece como un servicio web? ¿es necesario hacer este componente? esto implica mucho trabajo para poca ganancia.

Capa de servicios (Services Layer):

Esta capa contiene los servicios de negocio de la compañía. Están implementados con una tecnología estándar independiente de la plataforma. Tiene un descriptor, WSDL, que igualmente es estándar y que define el interfaz del servicio (con qué parámetros se invoca y cuáles devuelve).

Se pueden crear servicios compuestos, combinando otros servicios más simples.

Capa de procesos de negocio (Business Process Layer):

En esta capa se ubican los servicios compuestos, formados por la combinación de servicios más sencillos de la capa anterior. También se encuentran los procesos BPM (coreografías de servicios) que modelan casos de uso específicos y procesos de negocio de la organización. Cada proceso se un flujo de tareas automáticas o humanas (tienen que ser realizados por una persona).

Capa de consumidores (Consumers Layer):

Aquí se encuentran todos los posibles consumidores de los servicios. Debido a la característica de multicanalidad de los propios servicios y dado la cantidad de dispositivos existentes actualmente para acceder a los mismos, los consumidores pueden ser de tipo muy heterogéneo: puede ser las aplicaciones de oficina, aplicaciones con interfaz web del canal de internet, pueden ser socios de la empresa (B2B), usuarios finales con sus smartphones, etc. etc.

Es importante recalcar que los servicios son “agnósticos” respecto a la forma de su consumo o a la presentación que se va a hacer al usuario final.

Todos los servicios están dados de alta en el registro de sevicios a fin de que los consumidores puedan localizar el servicio que necesiten y sean independientes de la localización física del servicio.

Capa de integración (Integration Layer):

Esta es una capa de infraestructura que proporciona las capacidades de mediación, enrutado y tranporte de la petición desde el cliente hasta el servicio concreto. Estas capacidades se implementan mediante un Enterprise Service Bus (ESB)