Desarrollo de software en sistemas distribuidos

Servicios Web

Evolución de las aplicaciones

- Aplicación Monolítica
- Arquitectura Distribuida (2 capas y N capas)
- Arquitecturas Distribuidas con objetos (estándar CORBA)
- Integración de aplicaciones
- Arquitecturas Orientadas a Servicios

Arquitectura basada en servicios SOA

Las Arquitecturas Orientadas a Servicios están formadas por servicios de aplicación débilmente acopladas y altamente interoperables.

Se basa en tres conceptos técnicos:

- Servicios
- Interoperabilidad a través de un ESB
- Bajo acoplamiento

Josuttis Nicolai, "SOA in practice". O'Reilly, Pag 7-8. ISBN 978-0-596-52955-0. 2007

Conceptos tecnicos de SOA

- **Servicios**

Son piezas funcionales que resuelven un aspecto del negocio. Pueden ser simples (almacenar los datos de un cliente) o compuestos (el proceso de compra de un cliente)

- **ESB (Enterprise Service Bus)**

Es la infraestructura que habilita una alta interoperabilidad entre servicios en un contexto distribuido.

- **Bajo acoplamiento**

Concepto de reducir las dependencias entre sistemas. Es una idea muy difícil de desplegar, mantener y corregir.

Características de SOA

- Reuso de componentes de software existentes.
- Inteoperabilidad entre aplicaciones y tecnologías heterogéneas
- Flexibilidad para componer, integrar y escalar soluciones.

Principios de diseño SOA

- Contrato de servicio estándar

  Expresan el propósito del servicio (WSDL, Esquema XML, IDL)

- Bajo acoplamiento del servicio

  El acoplamiento mide el nivel de conexión entre dos elementos. El bajo acoplamiento de un servicio está dado por la independencia entre las prestaciones del servicio y su implementación.

- Abstracción del servicio

  Promueve el ocultamiento de detalles y las interfaces mínimas.

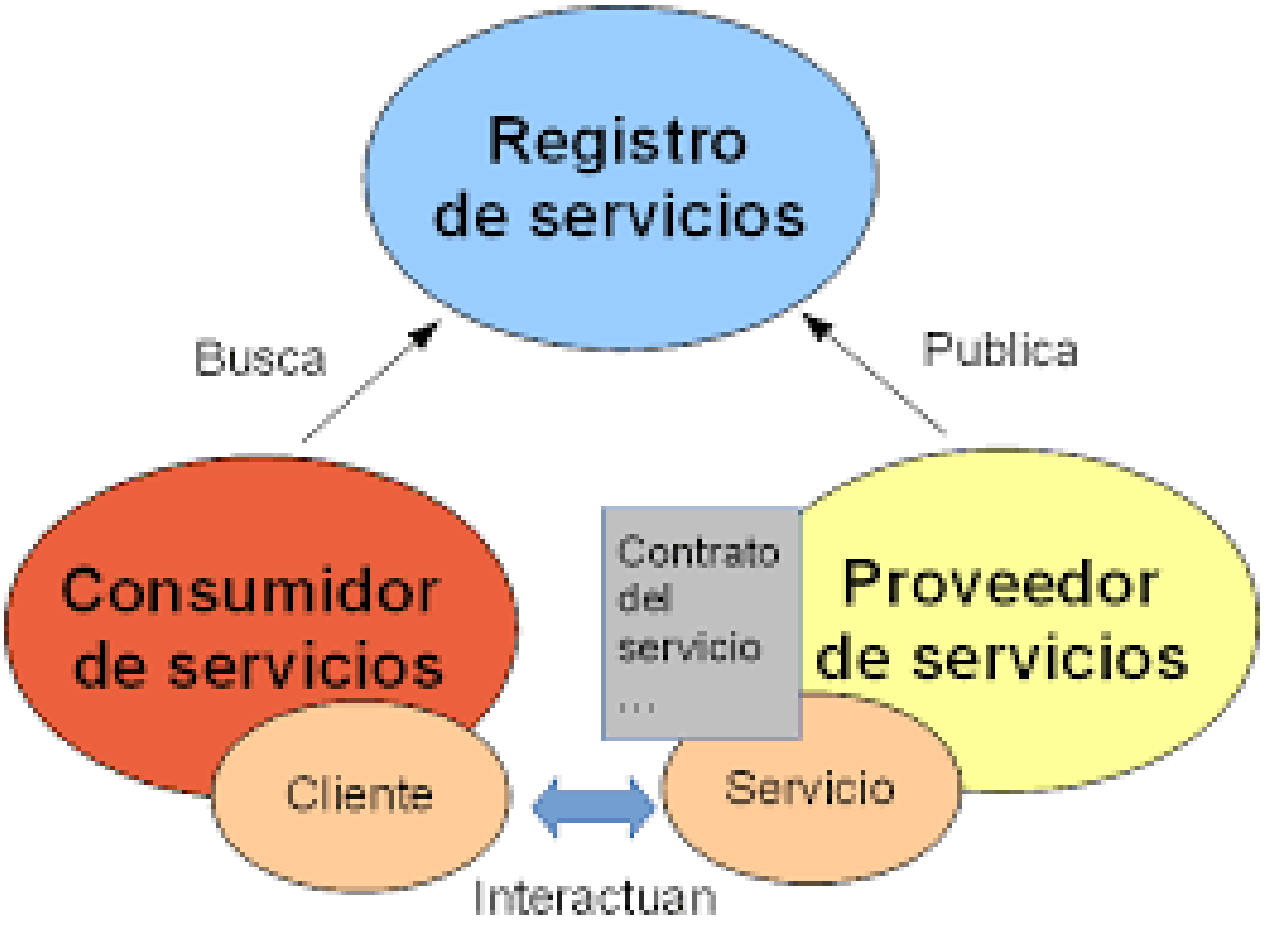
- Reusabilidad del servicio

  Servicio como recurso independiente del contexto funcional.

Principios de diseño SOA

- Auntonomía del servicio
  Los servicios deben tener el control sobre su entorno y recursos
- Falta de estado del servicio
  La gestión del estado de un servicio pueden comprometer su disponibilidad.
- Capacidad de encontrar el servicio
  Los servicios deben definir: propósito, capacidad y limitación de recursos, independientemente de la modalidad de registro.
- Componobilidad del servicio
  La composición de servicios permite escalar la complejidad de las arquitecturas SOA.

Actores SOA


Registro de servicios
Busca
Publica
Consumidor de servicios
Contrato del servicio
...
Proveedor de servicios
Cliente
Servicio
Interactuan

Servicios Web

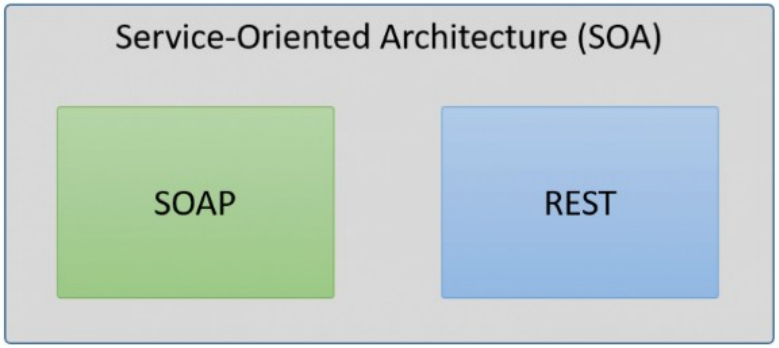
- Nuevo paradigma de desarrollo de aplicaciones interoperables entre si y basadas en internet.
- Permiten que las compañías vinculen el software
- Middleware que provee una interfaz entre clientes y servidores
- Los clientes acceden a las operaciones publicadas en la interfaz del servicio web

Características de los servicios

- Son el bloque fundamental de una Arquitectura orientada a servicios
- Poseen interfaces bien definidas, independientes de su implementación.
- Los clientes consumen los servicios sin la necesidad de entender cómo estos ejecutan sus pedidos.
- Los servicios puede ser descubiertos dinámicamente.
- Los servicios pueden componerse.
- Proveen funcionalidad de grano grueso.
- Cumplen con un objetivo de negocio predeterminado
- No dependen del contexto o estado de otros servicios
- Transparencia de localización(locationally transparent)

Implementaciones de los Servicios Web

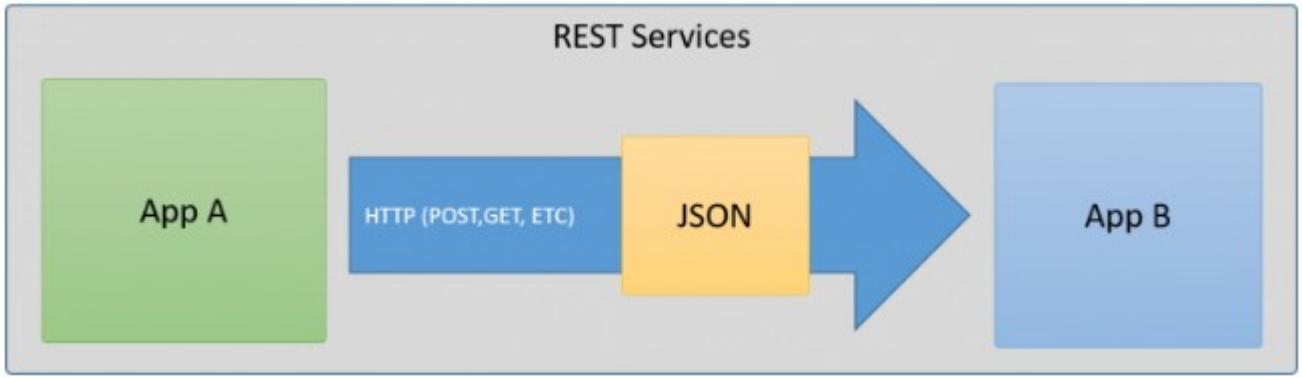
- SOAP como REST son tecnologías que implementan la arquitectura SOA.



	REST	SOAP
Formato del mensaje	XML,json, otros	XML dentro de SOAP
Definición del interfaz	No es necesario	WSDL
Transporte	HTTP	HTTP, JMS, FTP, etc.

Servicios Web RESTful

- Las arquitecturas REST se construyen sobre recursos identificados por URIs.
- Tienen una interfaz común, por lo que las únicas operaciones permitidas son GET, POST, PUT y DELETE.
- No tienen estado ni acceden a información de contexto. Toda la información para entender la petición debe encontrarse en la misma.
- Utiliza codigos de respuesta nativos (404)
- Permite transmitir cualquier tipo de datos, XML, JSON, Text.



Servicios Web RESTful – Reglas de diseño

- Deben se stateless

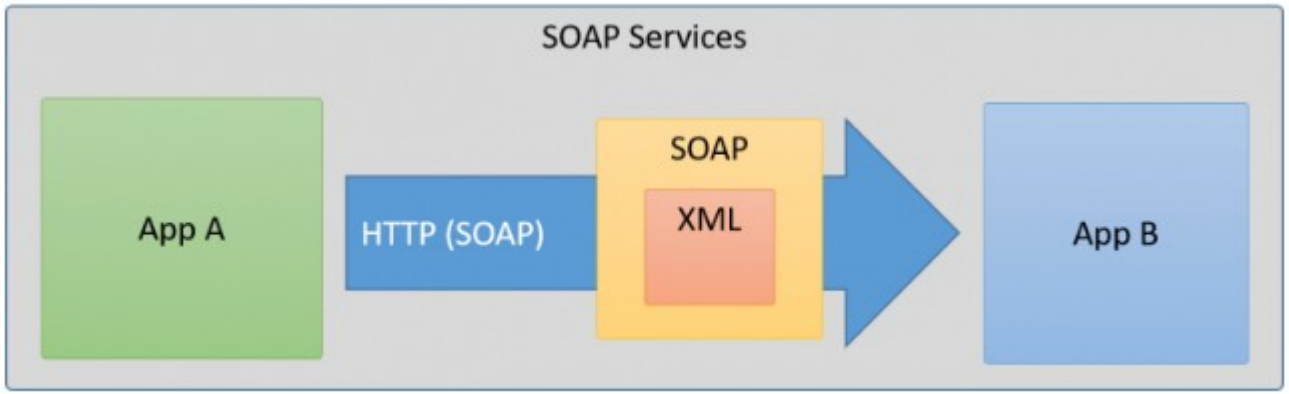
  Dado que cada solicitud realizada por un cliente debe ser siempre independiere de la anterior

- Utilizar el verbo HTTP correcto

  No solo Get y POST, por ejemplo PUT o DELETE
- Utilizar nombres de recursos apropiados.
- Utilizar Json como respuesta.
- Considerar la conectividad.
- Utilizar SSL para todas la peticiones
- Autenticación

Cada solicitud debe enviarse con un mecanismo de autenticación, como por ejemplo OAuth 2.

Servicios Web SOAP

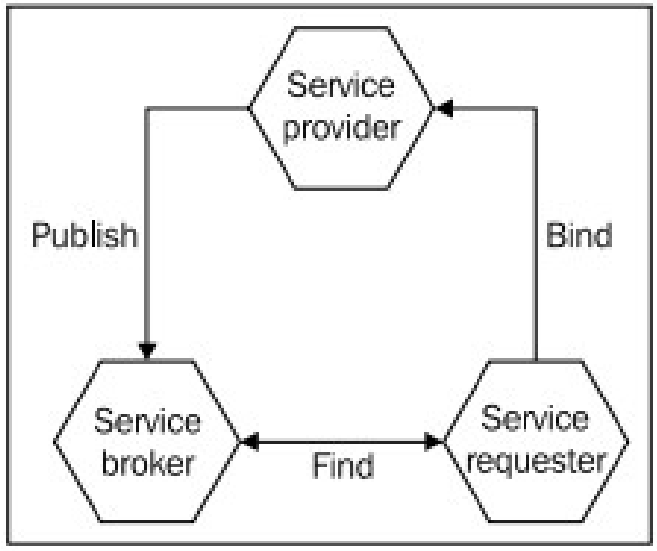


- Comunicación bajo el protocolo SOAP
- Solo soporta formato XML, menor flexilidad

Servicios Web SOAP – conceptos

Requisitos:
1. Definir el servicio web.
2. Publicar el servicio web.
3. Debe poder ubicarse.
3. Invocar el servicio web.
4. Despublicar en caso de ser necesario.

Descripción del Servicio: WSDL
Descubrir el Servicio: UDDI



Servicios Web SOAP

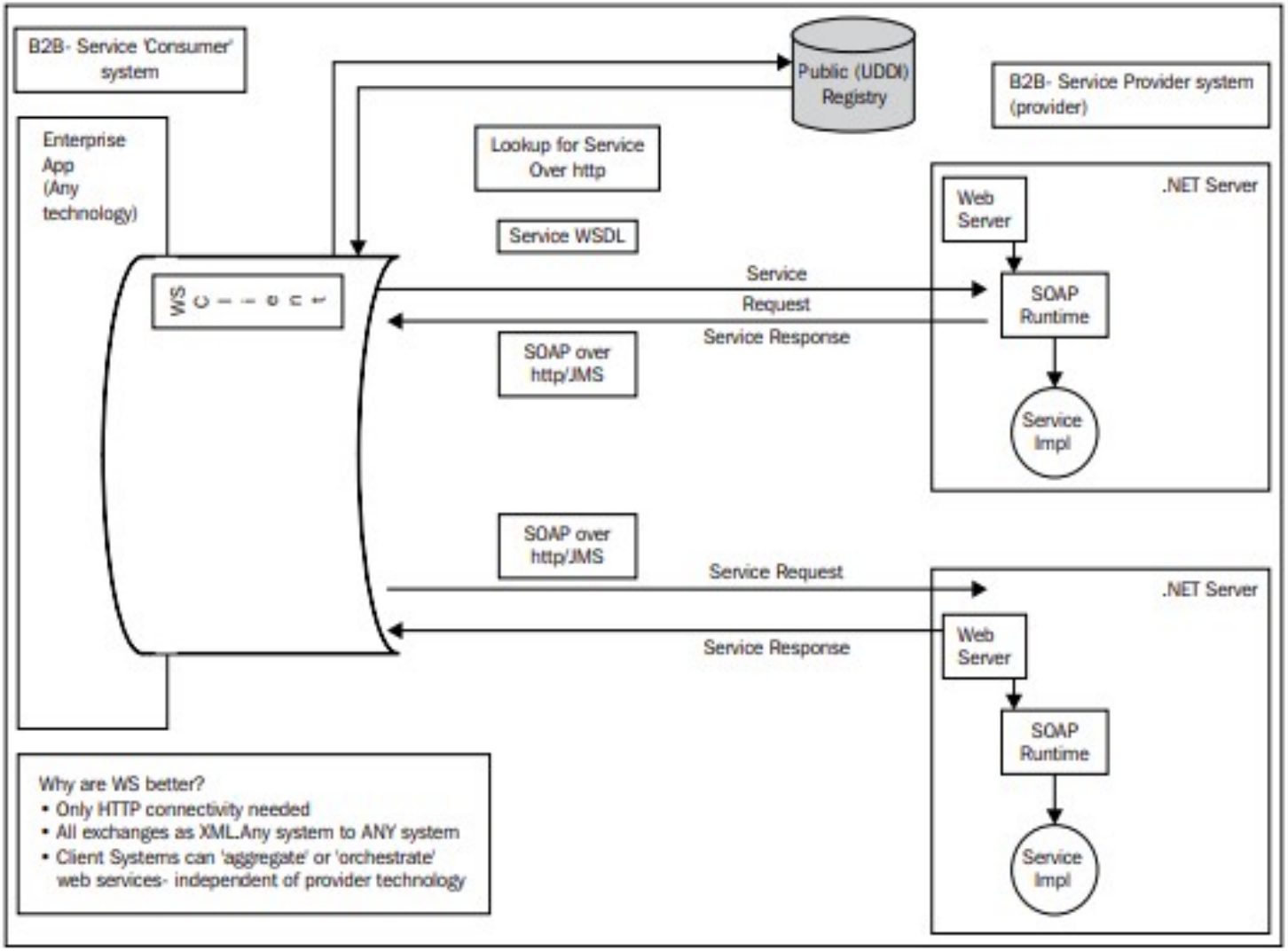
Los servicios web SOAP utilizan SOAP para:

- Invocar WS a través de interfaces descriptas en WSDL.
- Codificar los parámetros de entrada y de salida en XML.

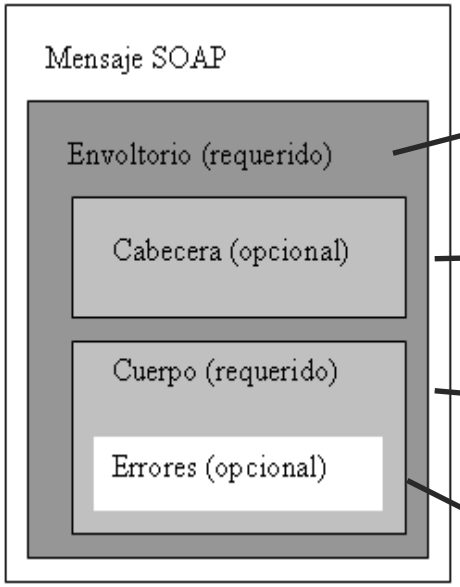
Para ello deben:

- Traducir las peticiones de servicios en invocaciones de lógica
- Traducir los parámetros entre distintos lenguajes de programación

Creación, registro, búsqueda y utilización del los Servicios Web



Estructura de un mensaje SOAP



Elemento raiz: guarda informacion del espacio de nombres y estilo de codificación

Cabecera: permite proporcionar Servicios a la carga del cuerpo. Ej: firma digital

Cuerpo: contiene la información que debe procesar el destino

Error: informa los errores. El protocolo contempla 4, pero el programador puede definir los propios

faultCode	Código de texto utilizado para indicar la clase de error.
faultString	Texto para explicación del error.
faultActor	Indica quien causo el error. Esto es útil en caso de que el mensaje SOAP viaje a través de múltiples nodos.
detail	Este elemento permite contener mensajes de errores específicos de la aplicación.

Estructura de codificación de errores

Pila de protocolos de los Servicios Web

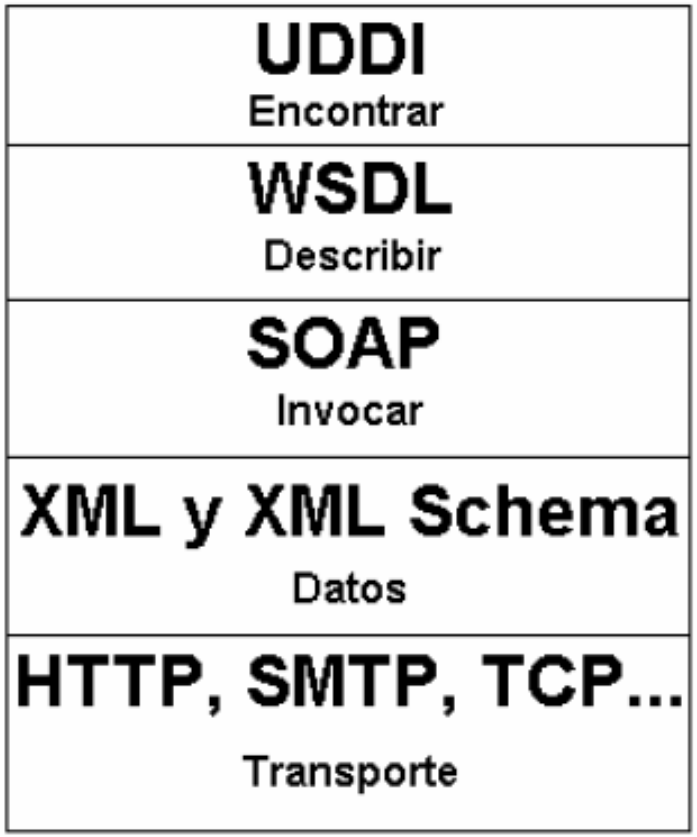
Ofrece un directorio de servicios en Internet

Ofrece un modo de definir los servicios

Permite invocar métodos de los servicios.

Permite a los consumidores de servicios enviar y recibir mensajes a y de los servicios

Son protocolos abiertos de Internet. Dan soporte a las capas superiores.



Estructura de un documento WSDL

Definitions: es la raíz de cualquier documento WSDL. Define nombre del Servicio Web y los múltiples namespaces a ser utilizados por el documento.

Types: define los tipos de datos usados por el cliente y el servidor.

Message: define el nombre del mensaje y las partes del mismo que pueden ser parámetros o valores de retorno.

PortType: define las diferentes invocaciones a servicios y el patrón de comunicación que se utilizara para el intercambio de mensajes.

Binding: describe de que forma el servicio va a ser comunicado.

Service: determina la dirección en la cual se invocara el servicio.

El protocolo UDDI

- El proyecto UDDI es en sí mismo un Servicio Web que establece un registro de búsqueda de servicios organizado de manera jerárquica.
- Consta de dos partes:
  - Un registro de información de los Servicios Web y el documento WSDL asociado a cada uno de ellos.
  - Un conjunto de documentos WSDL utilizados para manipular y buscar sobre el registro mismo.

Información almacenada en UDDI

La información que se almacena en el registro UDDI se clasifica en tres formas:

- Páginas blancas: contiene información general de quien publica el servicio.
- Páginas amarillas: contiene información clasificada por tipo de negocio, como productos y servicios que comercializa.
- Páginas verdes: contiene información técnica acerca de los Servicio Web que expone

La descripción de Servicios Web en UDDI no está limitada únicamente a servicios basados en SOAP, sino también servicios CORBA, RMI, e incluso simples páginas Web.

Última versión de UDDI: http://www.uddi.org/pubs/uddi_v3.htm

Lecturas recomendadas

- SOA with REST Tomas Erl
- Designing an Enterprise Application Framework for SOA
- USING OPEN-SOURCE SOA IN AN ENTERPRISE DEPLOYMENT
- Understanding the Business Benefits of an Open Source SOA Platform