

Índice

- Arquitectura orientada a servicios (SOA)
- Servicios web REST
- Servicios web SOAP
- Java WS

SOA

- En la actualidad hay muchas soluciones para aplicaciones empresariales de dimensión industrial
- Todas tienen ventajas e inconvenientes
- Lo más importante es tener claro cuando aplicar cada una

Arquitectura orientada a Servicios (SOA)

SOA

Tipo de aplicación	Solución	Uso de marcos (GUI, transacción, persistencia, etc.)	Ejemplos implementación J2EE
Empresarial	Arquitectura multicapa	No	JSPs + SAs POJOs + DAOs POJOs
		Sí	JSFs + SAs POJOs + mapeado O/R
+ Lógica distribuida	+ RPC (Remote Procedure Call)	No	JSPs + SAs POJOs accesibles con RMI + DAOs POJOs
		Sí	JSFs + SAs EJBs de sesión + ODs EJBs de entidad
+ Plataformas heterogéneas	+ SOA	No	JSPs + SAs POJOs accesibles como servicios web + DAOs POJOs
		Sí	JSFs + SAs EJBs de sesión accesibles como servicios web + ODs EJBs de entidad

Solución aplicable en base a los requisitos de aplicación

-RMI - Remote Method Invocation
 Protocolo de comunicación y API de Java, que permite que dos objetos que viven en dos JVMs (Java Virtual Machine) distintas y nodos de red distintos se comuniquen de manera transparente. Utiliza TCP/IP como protocolo de transporte.

-JSP (Java Server Pages) es una tecnología que permite mezclar HTML estático con HTML generado dinámicamente.

- POJO (Plain Old Java Object)

-JSF (Java Server Faces) tecnología y framework para aplicaciones Java que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE.

-EJB (Enterprise JavaBeans)

Componente Java responsable de ejecutar una transacción. Existen tres tipos de EJBs: Sesión, Entidad y de Mensaje

-- **EJB Entidad** Componente transaccional que implanta el ciclo de vida de los datos. Cada campo de un EJB de entidad equivale a una columna de una tabla relacional. Cada EJB de entidad se asocia con una tabla relacional.

--**EJB de sesión** Componente transaccional que ofrece métodos para realizar transacciones que involucran a recursos de la capa de persistencia. Un EJB de sesión invoca a uno o varios EJBs de Entidad. La transacción tiene escrita la lógica de la operación.

SOA

- Ya hemos visto la arquitectura multicapa
- La invocación remota de objetos en la práctica está muy ligada a plataformas de implementación
- Ahora nos centraremos **en arquitecturas orientadas a servicios,** **más independientes de las plataformas de implementación**

SOA

- La arquitectura orientada a servicios (SOA) es un modelo arquitectónico que intenta mejorar la eficiencia, agilidad y productividad de un empresa colocando los servicios como método principal a través del cual se representa la lógica de la solución

SOA

- En principio la arquitectura en sí debería diferenciarse de los elementos de la informática orientada servicios, pero en muchos contextos se utiliza de forma indistinta

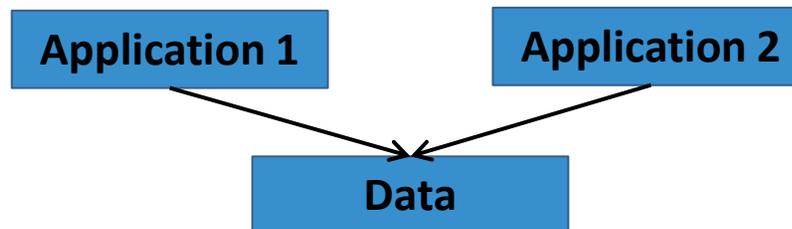
Objetivos y beneficios de SOA

- Podemos enumerar los siguientes:
 - Aumento de la interoperabilidad
 - Aumento de la federación { La federación se refiere a la unión de entornos diferentes manteniendo una gestión independiente
 - Aumento de las opciones de diversificación de plataformas
 - Mayor alineamiento del negocio y la tecnología
 - Aumento del retorno de la inversión
 - Aumento de la agilidad de la organización
 - Reducción de la carga de TICs

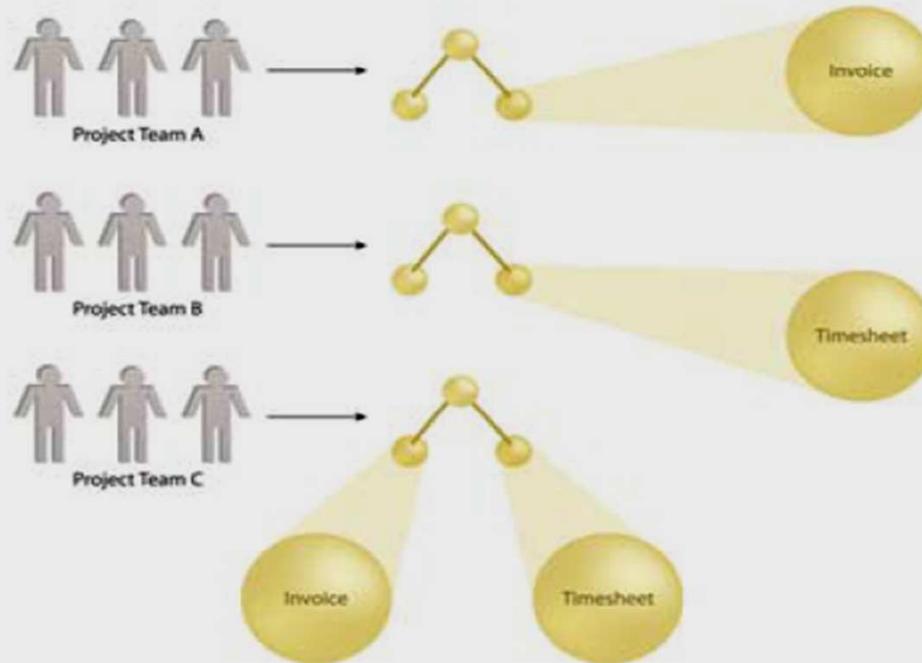
Objetivos y beneficios de SOA

- Aumento de la interoperabilidad

- En este contexto, interoperabilidad significa compartir datos entre aplicaciones
- La exposición de los servicios de una aplicación a través de SOA facilita dicha interoperabilidad



Objetivos y beneficios de SOA



Aumento de la interoperabilidad en SOA

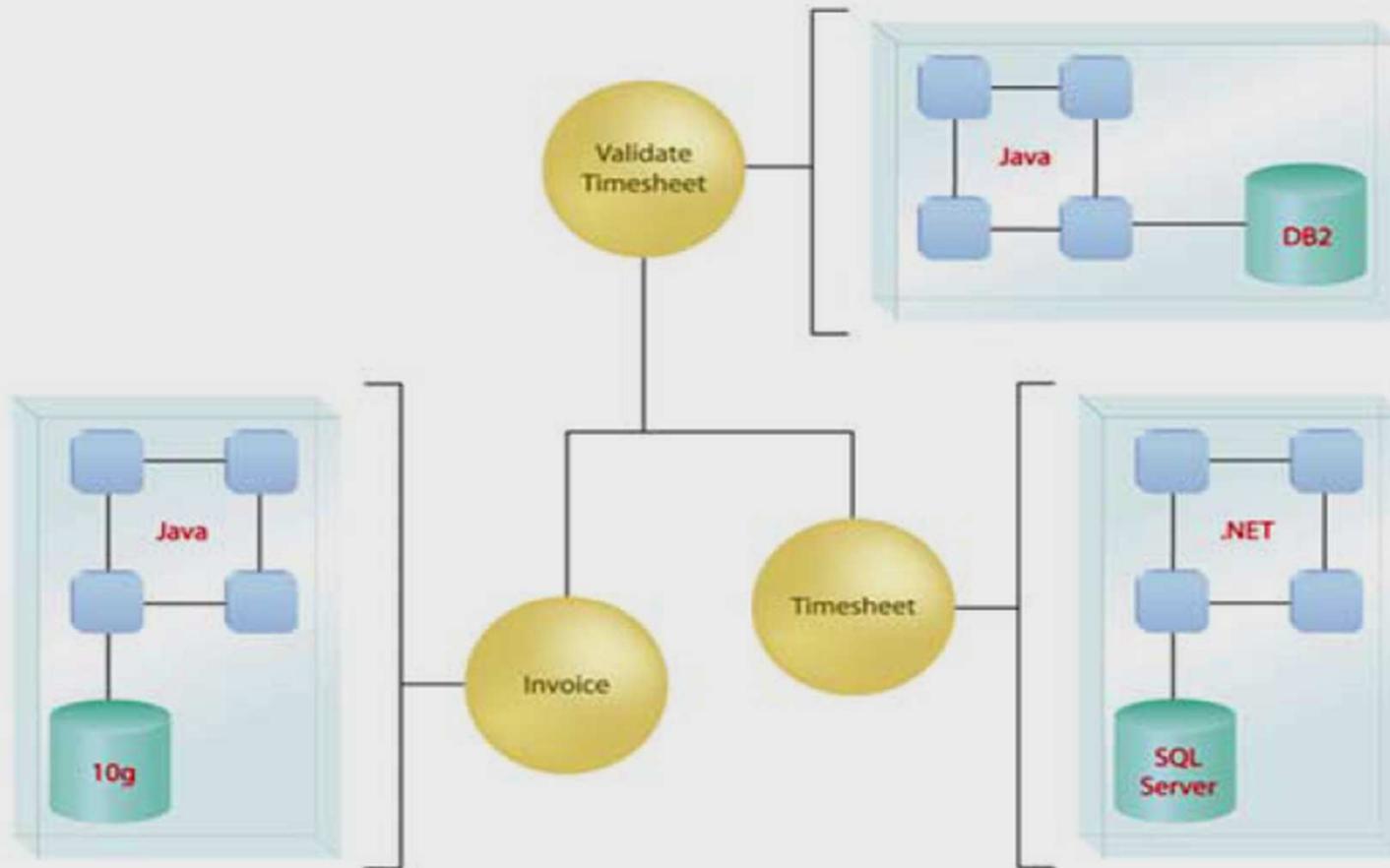
Objetivos y beneficios de SOA

- Aumento de la federación

- Un entorno informático federado es aquel donde los recursos y aplicaciones están unidos manteniendo su autonomía y autogobierno

- La exposición de servicios mediante SOA favorece de manera natural la federación de funcionalidades

Objetivos y beneficios de SOA



Servicios implementados en distintas plataformas

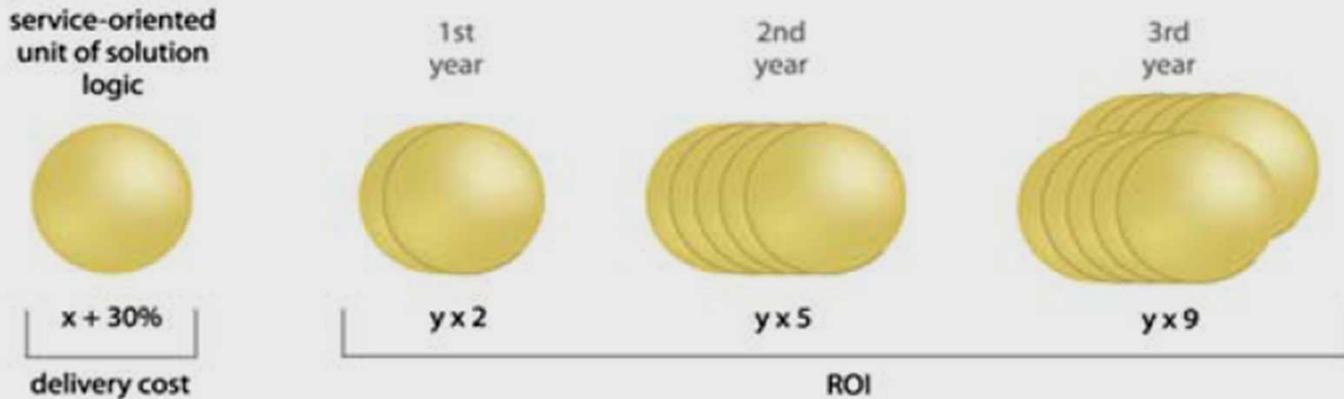
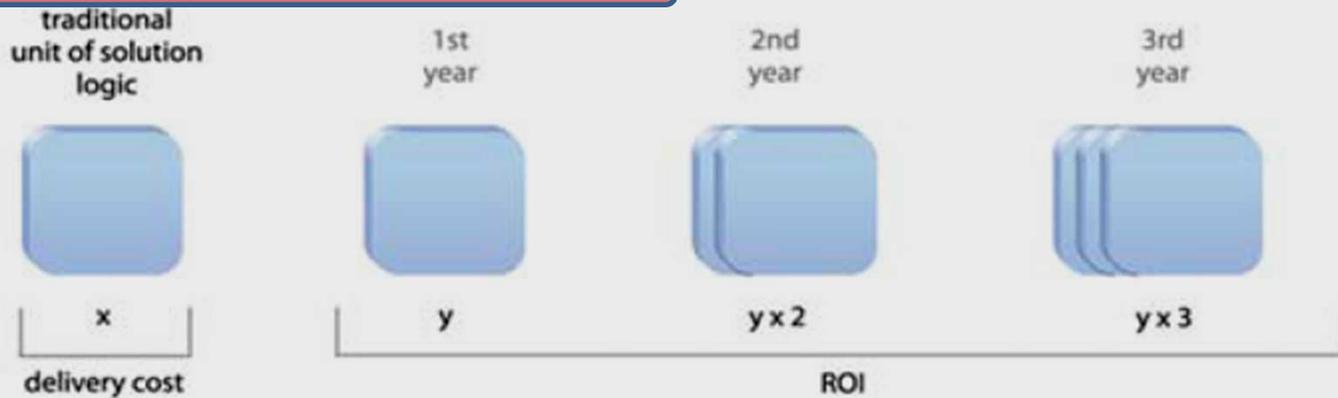
Objetivos y beneficios de SOA

- Mayor alineamiento del negocio y la tecnología

- El alineamiento busca que el sistema informático soporte lo antes posible los requisitos cambiantes de una empresa
- En la medida que el sistema esté diseñado en torno a servicios de grano fino, la evolución del sistema se verá facilitado, y así el alineamiento entre negocio y tecnología

Objetivos y beneficios de SOA

- Aumento del retorno de la inversión

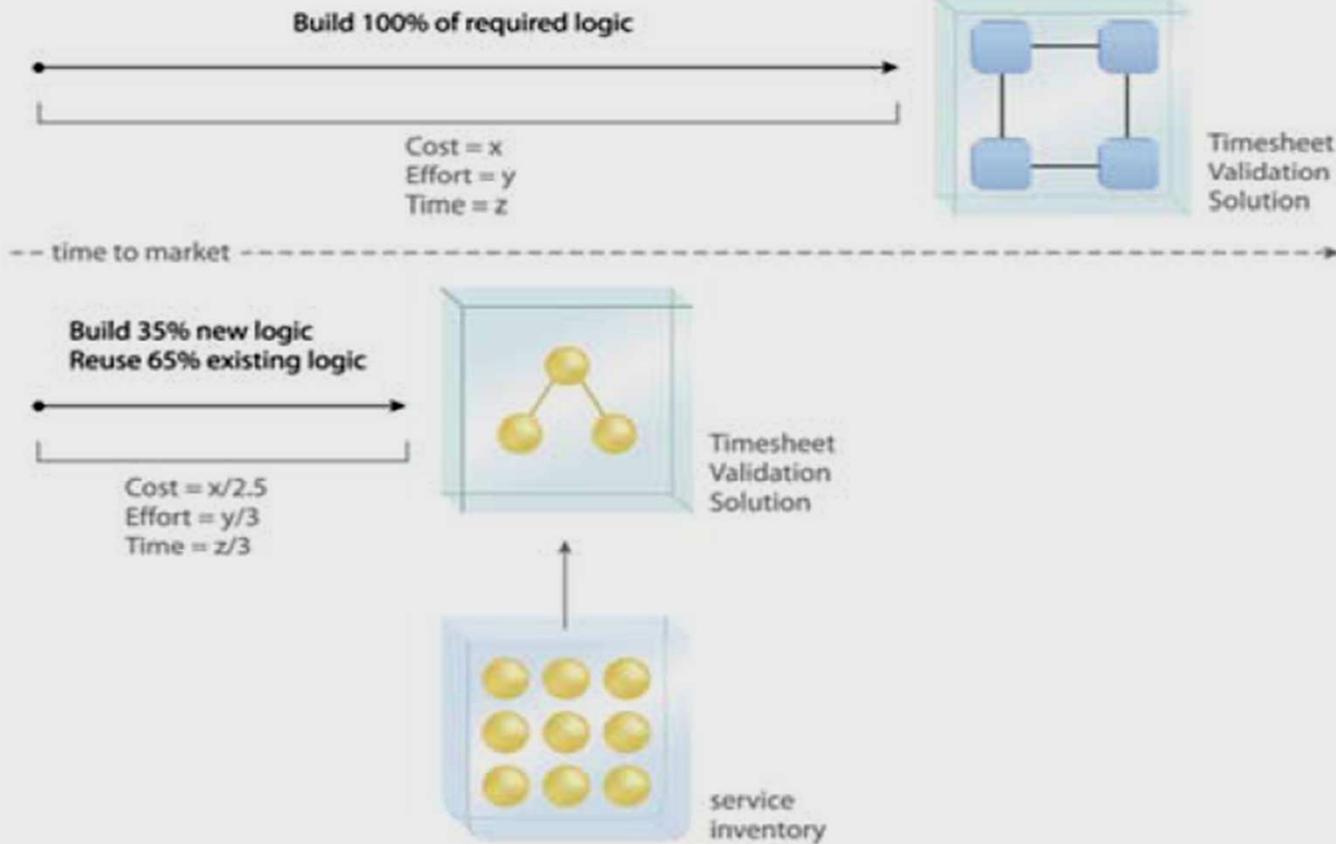


Comparación del retorno de inversión en soluciones clásicas y SOA

Objetivos y beneficios de SOA

- 4 • Aumento de la agilidad de la organización
 - La agilidad es la eficiencia con la que una organización puede responder al cambio
 - En la medida de que haya un buen soporte informático y un alineamiento entre negocio y sistema, cosas que favorecen SOA, la organización será más ágil

Objetivos y beneficios de SOA

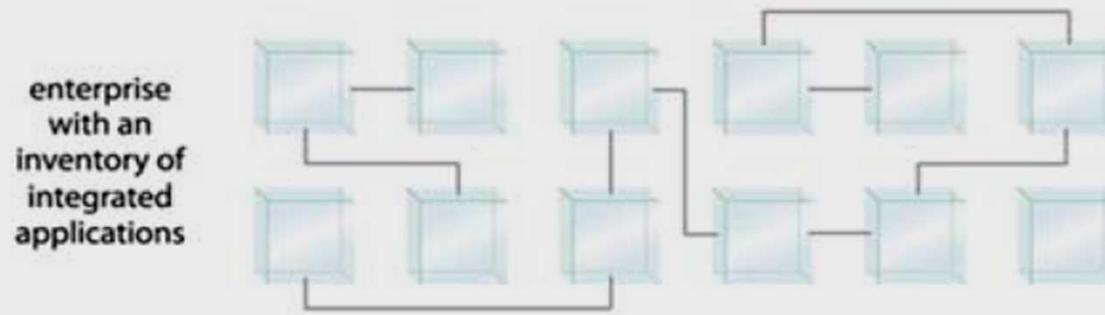


Comparación del tiempo de cambio en una aplicación tradicional y una SOA

Objetivos y beneficios de SOA

- 5 • Reducción de la carga de las TICs
 - A menor carga de cualquier componente de una empresa mayores ventas
 - La arquitectura orientada a servicios permite tener un departamento de TICs más ágil que disminuye la carga de la organización

Objetivos y beneficios de SOA



the same enterprise with an inventory of services



La organización en servicios permite tener departamentos de informática menos costosos

Principios de diseño SOA

- Hay una serie de principios a la hora de diseñar servicios:
 - 1 – Contrato de servicio estándar
 - 2 – Bajo acoplamiento del servicio
 - 3 – Abstracción del servicio
 - 4 – Reusabilidad del servicio
 - 5 – Autonomía del servicio
 - 6 – Falta de estado del servicio

Principios de diseño SOA

7 – Capacidad de encontrar el servicio

8 – Componibilidad del servicio

1 - **Contrato de Servicio Estándar**

– Los servicios expresan su propósito y capacidades mediante un contrato de servicio. Por ejemplo:

- WSDL (Web Services Description Language)
- Esquema XML
- Descripción WS-Policy

– Este es el principio más importante

Principios de diseño SOA

- 2 - **Bajo acoplamiento de servicios**
 - El acoplamiento es el nivel de conexión o relación entre dos objetos
 - El bajo acoplamiento de servicios promueve la independencia entre las responsabilidades del servicio y su implementación
- 3 - **Abstracción del servicio**
 - Este principio promueve ocultar el mayor número posible de detalles del servicio
 - Es decir, promueve interfaces mínimos

Principios de diseño SOA

4 - Reusabilidad de servicios

- Este principio promueve posicionar los servicios como recursos empresariales independientes de contextos funcionales concretos
- Así, los servicios son útiles para más de un propósito

Principios de diseño SOA

- 5 -
- **Autonomía de servicios**
 - Este principio determina que los servicios deben ser capaces de llevar a cabo su funcionalidad de forma consistente y fiable
 - Para esto, la implementación de dichos servicios debe tener un importante grado de control sobre su entorno y recursos

Principios de diseño SOA

5 - Componibilidad de servicios

– Según crece la sofisticación de las soluciones SOA, crece la complejidad de las configuraciones de composición de servicios

– Por tanto, la capacidad de componer servicios es fundamental en aplicaciones avanzadas

Principios de diseño SOA

- En los principios anteriores falta el concepto de *interoperabilidad*
- No se ha hecho explícito, ya que se supone como requisito fundamental de SOA y de cada uno de los principios anteriores

Principios de diseño SOA

- La aplicación consistente de estos ocho principios de diseño producen diseños con:
 - Mayor consistencia en la representación de la funcionalidad y los datos
 - Dependencias reducidas entre las unidades de solución lógica
 - Conocimiento reducido del diseño de la lógica y de su implementación
 - Mayor capacidad para reutilizar un elemento de solución lógica para múltiples propósitos

Principios de diseño SOA

- Mayor oportunidad para combinar unidades de solución lógica en diferentes configuraciones
- Mayor nivel de predicción del comportamiento
- Incremento en la disponibilidad y escalabilidad
- Mayor conocimiento de las soluciones lógicas disponibles

Problemas de arquitecturas no SOA

- Hay algunos problemas que las arquitecturas no SOA pueden presentar:
 - Puede haber funcionalidad redundante
 - Esto hace que el proceso de desarrollo no sea eficiente
 - Provoca la existencia de funcionalidad innecesaria en la empresa
 - Esto lleva a plataformas heterogéneas, infraestructuras complejas y arquitecturas enrevesadas
 - La integración se vuelve un desafío constante

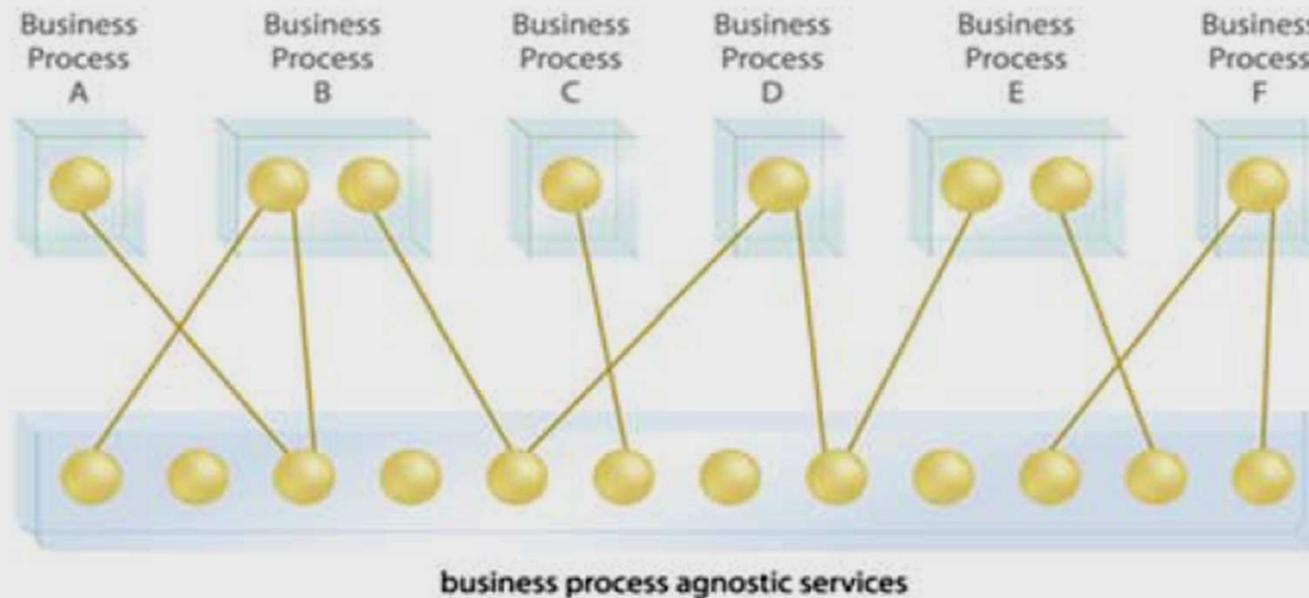
Ventajas de la arquitectura SOA

- La arquitectura SOA presenta distintas ventajas:
 - Mayor cantidad de lógica independiente de la solución
 - Menor cantidad de lógica específica para la aplicación
 - Reducción en el volumen de lógica
 - Interoperabilidad inherente

Ventajas de la arquitectura SOA

- Mayor cantidad de lógica independiente de la solución
 - Dentro de una solución SOA la unidades de lógica (servicios) encapsula funcionalidad que no es específica a una aplicación o servicio de negocio
 - Estos elementos son elementos TICS reutilizables e independientes de la solución

Ventajas de la arquitectura SOA



Servicios como elementos de lógica reutilizables

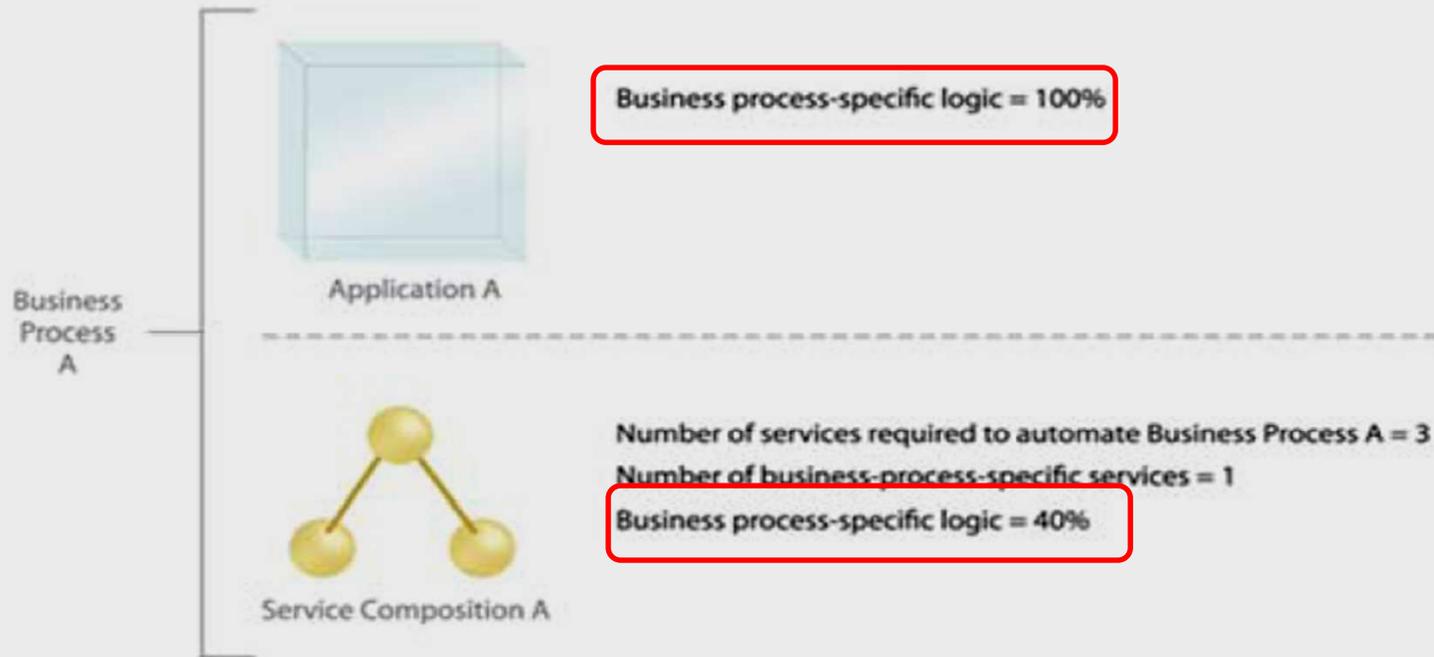
Ventajas de la arquitectura SOA

- Menor cantidad de lógica específica para la aplicación

- Al hacer servicios que no son específicos para una aplicación o negocio, se disminuye la cantidad de lógica específica para la aplicación

- Esto hace difuminar el concepto de aplicación independiente

Ventajas de la arquitectura SOA

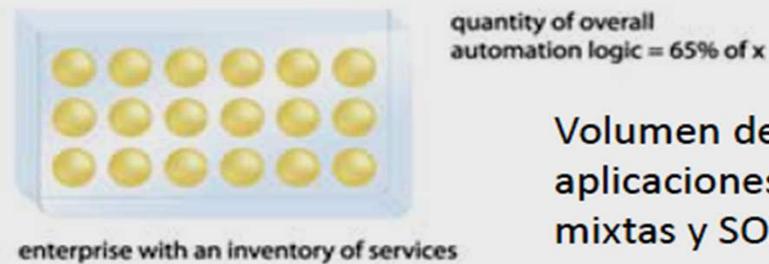
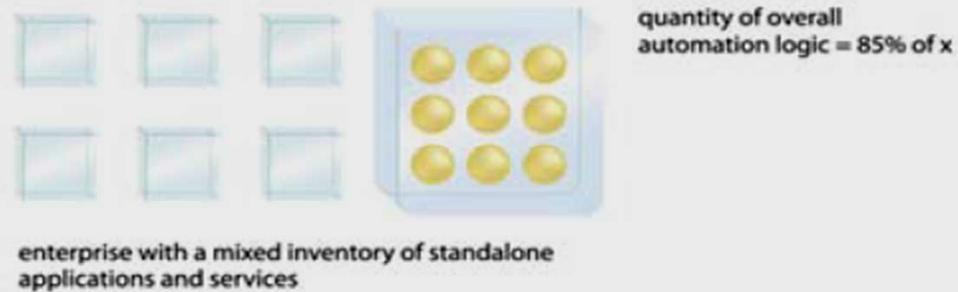
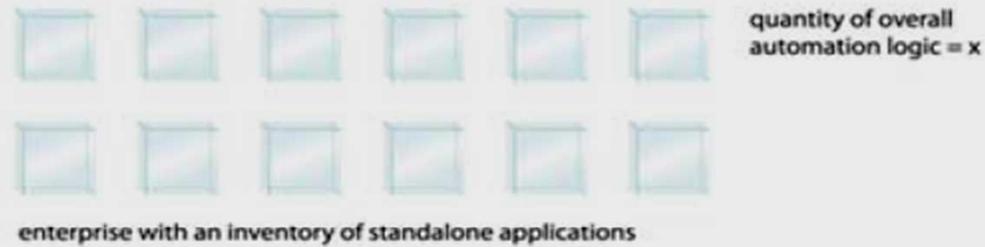


Reducción de lógica específica de la solución

Ventajas de la arquitectura SOA

- Reducción en el volumen de la lógica
 - El volumen de la lógica de la solución se ve **disminuido** al estar compartidos y reutilizados los servicios por diferentes procesos de negocio

Ventajas de la arquitectura SOA



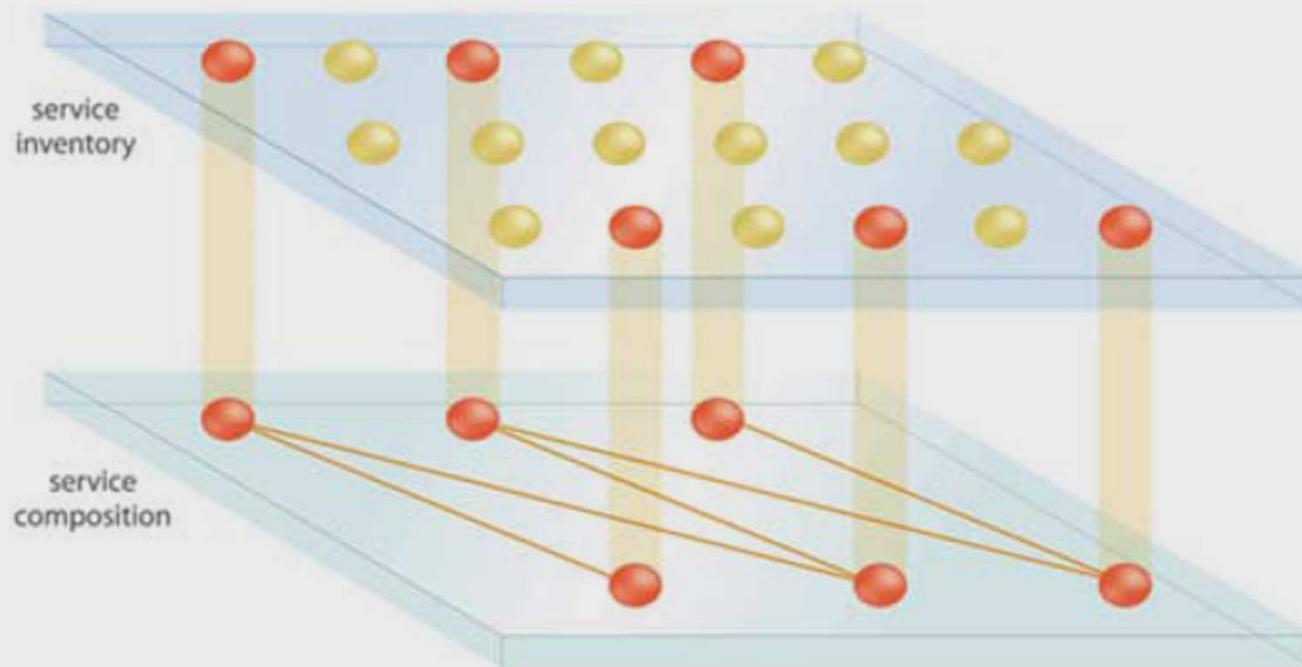
Volumen de lógica en aplicaciones clásicas, mixtas y SOA

Ventajas de la arquitectura SOA

• Interoperabilidad inherente

- Las soluciones SOA conllevan soluciones lógicas alineadas
- Así, los contratos de los servicios y sus modelos de datos subyacentes, determinan un mecanismo que aumenta la interoperabilidad

Ventajas de la arquitectura SOA



Un conjunto de servicios consistentes aumenta la interoperabilidad y comonibilidad

Desafíos de la arquitectura SOA

- La arquitectura SOA también presenta una serie de desafíos:
 - Aumento en la complejidad del diseño
 - Necesidad de estándares de diseño y de arquitectura
 - Enumeración previa de todos los servicios antes de su implementación
 - Puede dificultar la aplicación de metodologías ágiles

Desafíos de la arquitectura SOA

– Fuerza la presencia de una autoridad en la gestión de los servicios

- Por tanto, sólo debe aplicarse una solución SOA cuando sea estrictamente necesario