

# SOA

## Arquitectura Orientada a Servicios

### I

**Profesor:** Alejandro Sirvent Llamas

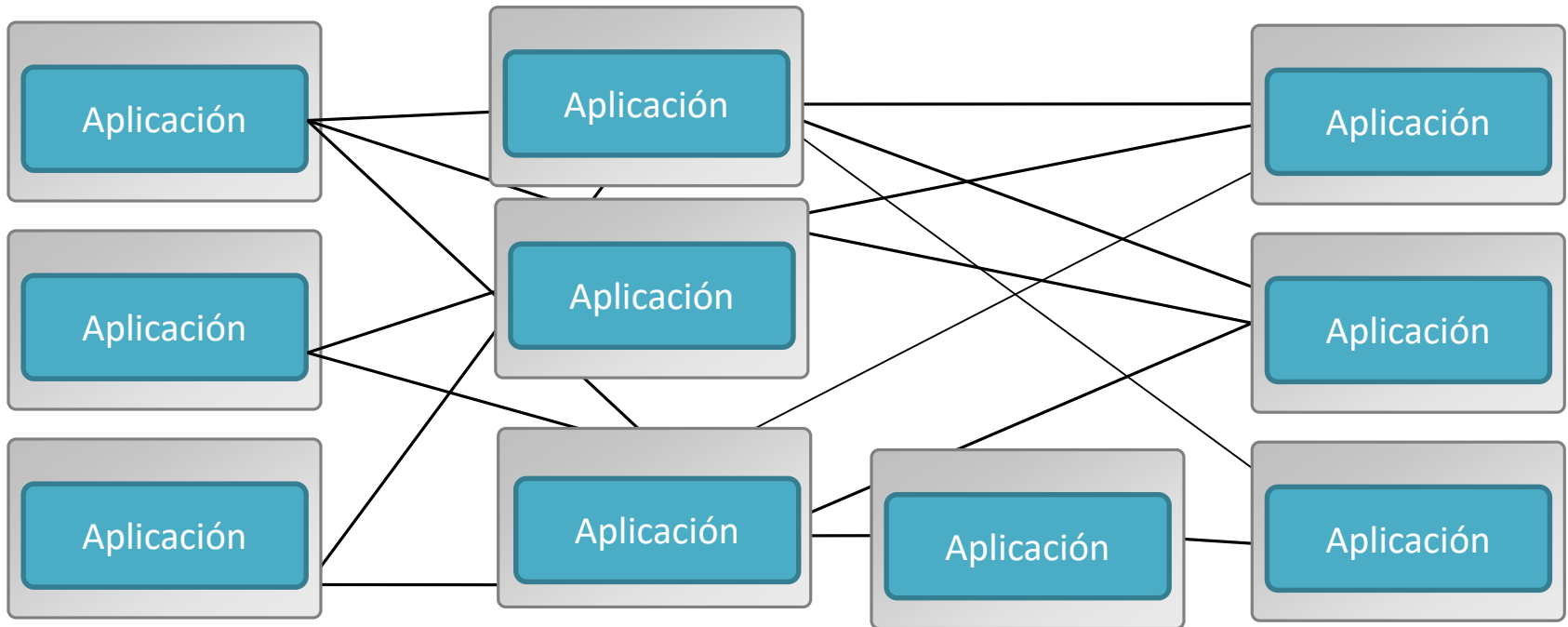
**Curso:** 2019-2020

- Una organización compleja cuenta con **sistemas y aplicaciones muy distintas**, focalizadas por áreas:
  - finanzas,
  - logística,
  - recursos humanos
- Con fines muy concretos, implementadas en **tecnologías muy diversas**.
- Para **ganar en productividad** es importante que estos sistemas:
  - Intercambien la información necesaria,
  - Eviten replicas,
  - Eviten inconsistencia de datos,
  - Que las interacciones sean mantenibles...

- Por otra parte, para **mantener la competitividad**, cada vez es más **necesario ir a modelos B2B**, intercambiando información con los sistemas informáticos de nuestro entorno:
  - Proveedores,
  - Clientes,
  - Bancos,
  - Administraciones públicas...
- En este modelo, **nuestra organización** (sus sistemas) **actúa** como **proveedor** o **consumidor** de **servicios** de negocio, que en este contexto, serán intercambios de información.

- **¿Mejora esto nuestra productividad o nuestra competitividad?**
- Ejemplos:
  - Una **integración electrónica** con nuestros **clientes** puede situarnos por delante de otros competidores con integraciones manuales, ya que **seremos capaces de ofrecerles** lo que necesitan con una **mayor celeridad**.
  - Una **integración electrónica** con nuestros **proveedores** puede **facilitar** la gestión y seguimiento de los **pedidos**.
  - Nos puede permitir **mejorar la calidad** de nuestra información **sumándole** la información de estos **sistemas externos**, permitiéndonos obtener un **conocimiento** mejor de la situación de nuestra organización.

- Partimos de una necesidad



## • Partimos de una necesidad

- Las conexiones entre aplicaciones:
  - Se realizan punto a punto,
  - Implementándolas para cada necesidad concreta,
  - Cada cuál utilizando el sistema que le plazca: ficheros, conexiones RMI...
- **Acoplamiento** entre sistemas.
  - Cualquier modificación en un extremo, aunque sea pequeña, afecta a la otra parte
- **Descontrol**: difícil gestión de las interconexiones, cuando hay problemas, cuestan de encontrar...

# • Partimos de una necesidad

## - Ejemplo:

- Un Banco quiere exponer la funcionalidad **Consultar Saldos**, el cliente del Banco puede acceder a esta información por **Cajero, Portal Web, Teléfono y otros**.
- En el caso de punto a punto cada uno de éstos podría tener su propia orquestación (llamar a consultar saldo TDD, TDC, cuenta ahorro, corriente) y cada una se comunica directamente con cada uno de los sistemas destino.
- Se tendrían **funcionalidades repetidas** e interconexiones repetidas.

## • Servicios

- **Servicios Web**, tecnología que permite intercambios de datos entre aplicaciones mediante un conjunto de protocolos y estándares.
- **Interoperabilidad**... conectar distintos estándares, distintos modelos de datos, distintos ...
- **Reusabilidad** ... re-utilizamos librerías y códigos...
- **Desacoplar** ... sistemas que sean independientes al mantenerse y desarrollarse.

Diagram illustrating the components of a Service-Oriented Architecture (SOA):

- Consumidor** (Consumer): Represented by a blue box at the top.
- Servicio** (Service): Represented by a blue box inside a larger grey box in the center.
- Productor** (Producer): Represented by a purple box at the bottom.

Arrows indicate interactions:

- A double-headed vertical arrow connects the **Consumidor** and the **Servicio** box.
- A double-headed vertical arrow connects the **Servicio** box and the **Productor**.

Servicio

Productor



- **¿Qué es un servicio?**

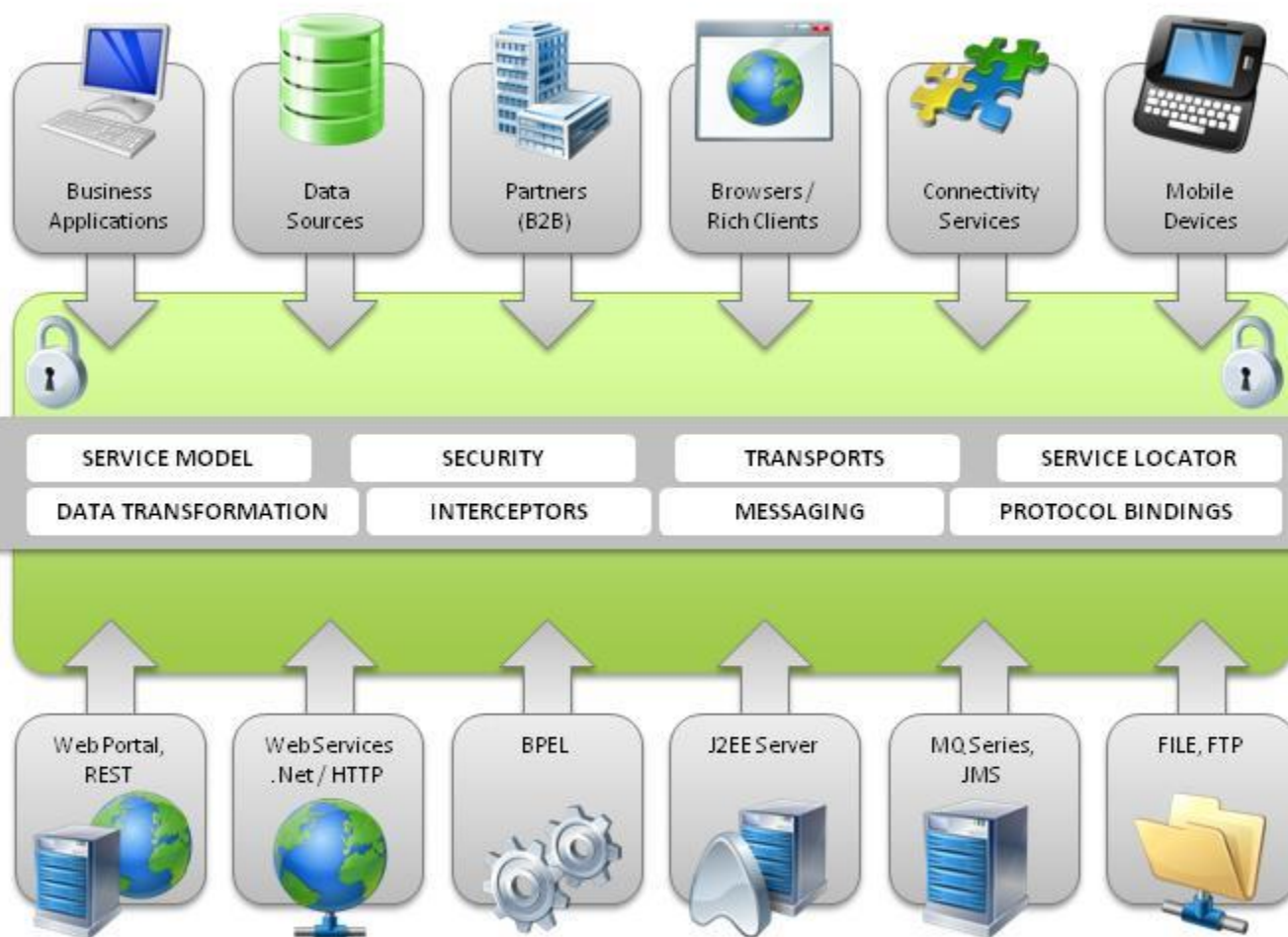
*“Un servicio es una **funcionalidad reutilizable y auto-contenida**, con un **interfaz perfectamente definido y descrito**, no sólo en cuanto a la información que intercambia y el mecanismo utilizado, sino también con las condiciones de utilización tales como Acuerdos a Nivel de Servicio y Políticas de Seguridad.”*

- Service Oriented Architecture
  - No existe un consenso para definir SOA
  - Confusión
    - Metodología?
    - Arquitectura?
    - Tecnología?
- ¿Qué no es SOA?
  - SOA no es lo mismo que Servicios Web
  - Un término comercial
  - Un producto
  - Una iniciativa IT, únicamente
  - Componentes de software reusables
  - Usar ESB o EAI.
    - Suelo decirse que EAI es el pasado y que SOA es el presente → (“transportation is a thing of past now road is here”).

# Diferencias fundamentales entre: ESB, EAI y SOA

- ESB, es la abreviatura de **bus de servicios empresariales** :
  - Es una **arquitectura** de software que **permite** la **integración** de aplicaciones empresariales y servicios para arquitecturas complejas, tales como plataformas de infraestructura middleware.
  - Su principal función es **proporcionar** las **conexiones** entre las **aplicaciones** que se comunican.
    - Como si fuese un router.
  - Se **utiliza** comúnmente en la integración de aplicaciones empresariales (**EAI**) o la arquitectura orientada a servicios (**SOA**).

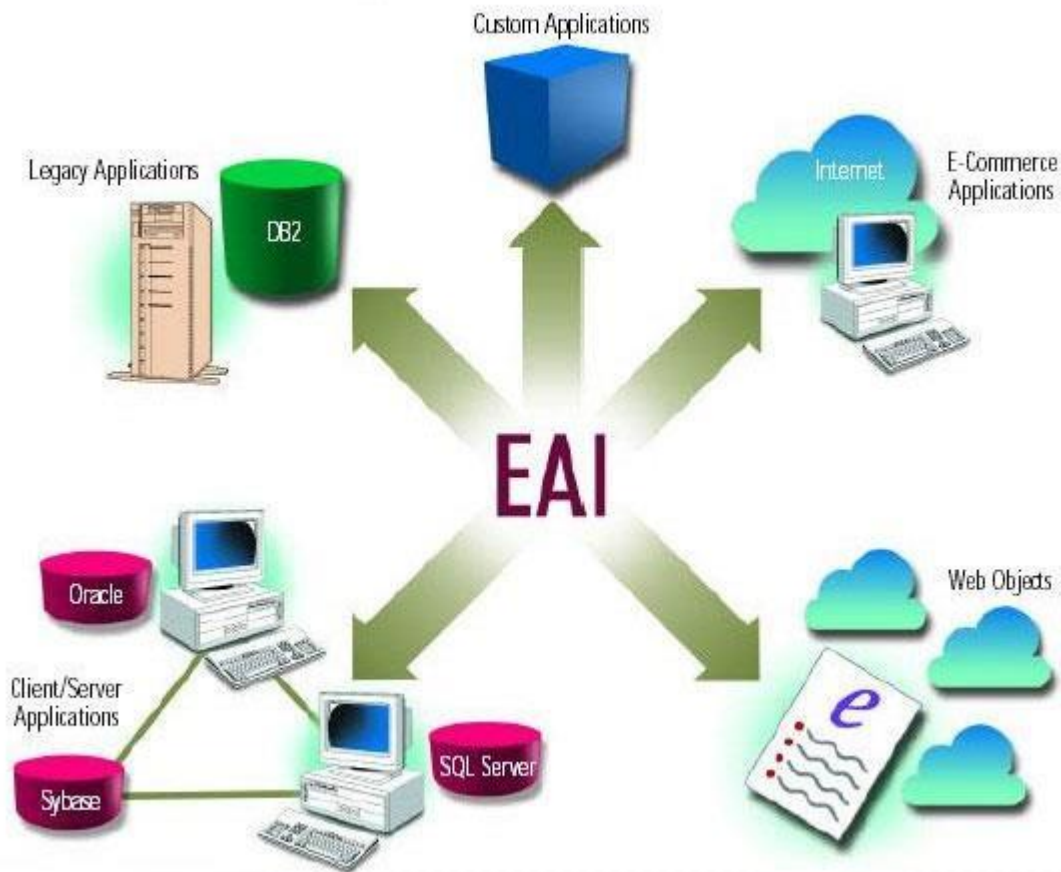
- ESB, es la abreviatura de **bus de servicios empresariales** :
  - La interacción y la comunicación entre los componentes están al otro lado del bus, que tiene una **función similar a la del bus del PC**:
    - Para manejar la transferencia de datos o el intercambio de mensajes entre los servicios, sin necesidad de escribir código real.
  - ESB proporciona:
    - Enrutamiento.
    - Transformación de datos.
    - Traducción a petición de un cliente.
    - Eventos de interpretación.
  - A menudo es necesario **transformar** los **mensajes** en un **formato** que la aplicación puede interpretar.
  - Un ESB también se utiliza para **cambiar** el **contenido** de los **datos o ejecutar** los **servicios** a través de un **motor de reglas**.



- EAI, abreviatura de **integración de aplicaciones empresariales**:
  - Hace uso de software y sistemas informáticos para permitir la **integración** de datos a través de aplicaciones **dentro de una sola organización**
    - Simplificación de los procesos de negocio entre aplicaciones conectadas y fuentes de datos.
  - La puesta en común de datos y procesos de negocio entre aplicaciones son sus principales propósitos.
  - EAI también **define** un **conjunto de principios** para la **integración** de múltiples sistemas para las arquitecturas de comunicación, como middleware orientado a mensajes (**MOM**).

- EAI, abreviatura de **integración de aplicaciones empresariales**:
  - Sirve como un sistema que puede proporcionar un **servicio de negocio** para **simplificar** los **datos** de información **entre** las **diversas aplicaciones**, lo que hace posible integrar fácilmente cuando sea necesario.
  - Es un componente fundamental de las estrategias de las empresas de hoy.
  - Diferenciándose de EAI, que trata de vincular las aplicaciones empresariales para que puedan comunicarse entre sí y llevar a cabo “lotes” de transferencias de datos,
    - Es **SOA** quien proporciona las "transferencias de datos transaccionales", sin necesidad de software de terceros.
  - **SOA es diferente del enfoque EAI, ya que no depende de una solución de terceros.**





- SOA

- Es un paradigma de integración que se basa en un principio fundamental del diseño y proporciona servicios **interoperables** arquitectónicos - de ahí su nombre.
- Abarca:
  - Sistemas heredados,
  - Componentes de software
  - Capacidades de procesamiento de mensajes.
- Su **función principal** es la **integración** de los **servicios** y el desarrollo de aplicaciones.
- SOA es el nexo que une las metas de negocio con el sistema de software.

- SOA
  - **Permite automatizar los procesos de negocio**, con la especificación de los sistemas de negocio de modelado orientados al servicio dentro de una aplicación arquitectónica.
  - Se utiliza como esquema de diseño de soluciones a nivel de aplicación y de nivel empresarial para el modelado de un entorno informático.
  - Aparentemente, **SOA es una solución para superar todo tipo de carencias de los EAI** y proporcionar un medio para que los servicios sean **reutilizables**.
  - Soporta ESB.
    - Plataforma ideal para la adaptación del transporte, la adaptación de servicios y los servicios comunes.



# Definición SOA

- Service Oriented Architecture
  - OASIS (Modelo de Referencia SOA)
    - «Un paradigma para organizar y utilizar capacidades distribuidas que podrían estar bajo el control de diferentes propietarios... SOA es un medio para organizar soluciones que promueve la reusabilidad, escalabilidad e interoperabilidad.»

- Service Oriented Architecture

- Según *Gartner*

- «*The main objective with SOA initiatives is to enable a more agile, flexible and standardized approach to designing, developing and deploying functionality that is often scattered through out established IT systems. »*
    - *El principal objetivo de las iniciativas SOA es permitir un enfoque más ágil, flexible y estandarizado para el diseño, desarrollo y despliegue de las funcionalidades que a menudo se dispersan fuera de lo establecido por los sistemas de TI.*

- Beneficios de SOA

- Reusabilidad

- “Las funcionalidades actuales de Negocio pueden ser reutilizadas. También nuevas funcionalidades deben ser diseñadas en base a la reusabilidad”*

- Interoperabilidad.

- “La interoperabilidad se define en términos de facilidad para integrar sistemas heterogéneos para que puedan intercambiar información”*

- Sistemas desacoplados

- “Se manifiesta en la posibilidad de evolucionar las aplicaciones y sistemas de forma separada siempre y cuando se respeten las interfaces”*

- Escalabilidad

- “Las aplicaciones son flexibles para adaptarse al cambio de los requerimientos de negocio”*

- Coste Eficiencia

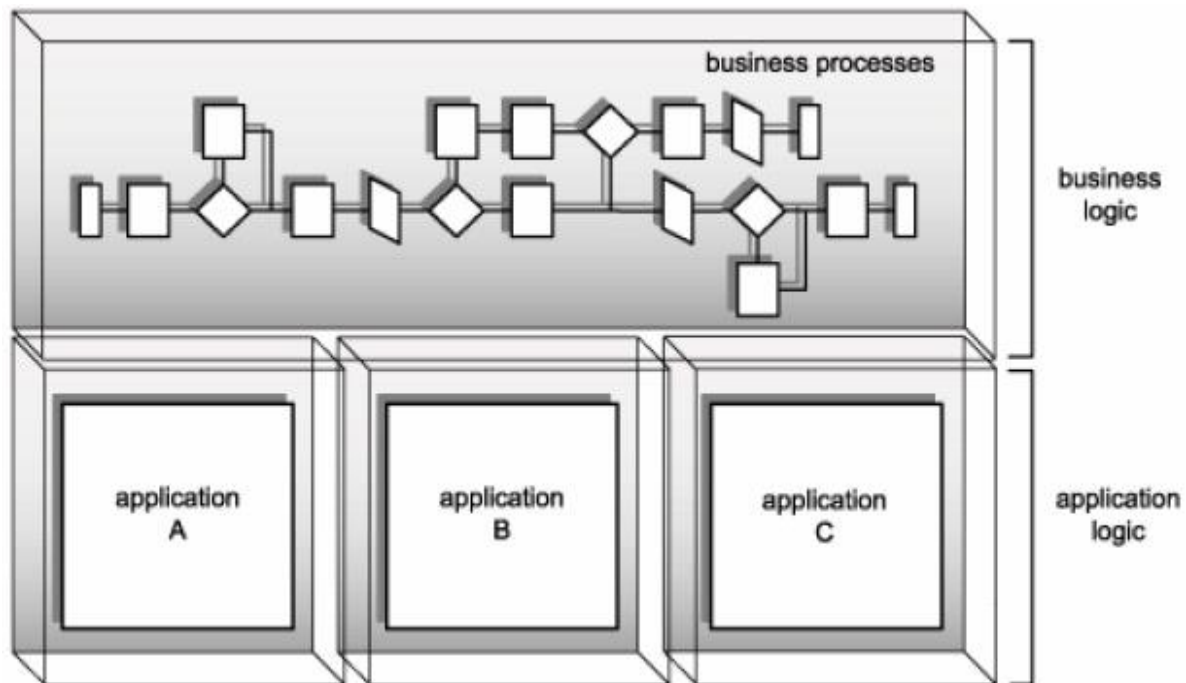
- “SOA es altamente eficiente en costes ya que los recursos están basados en estándares”*



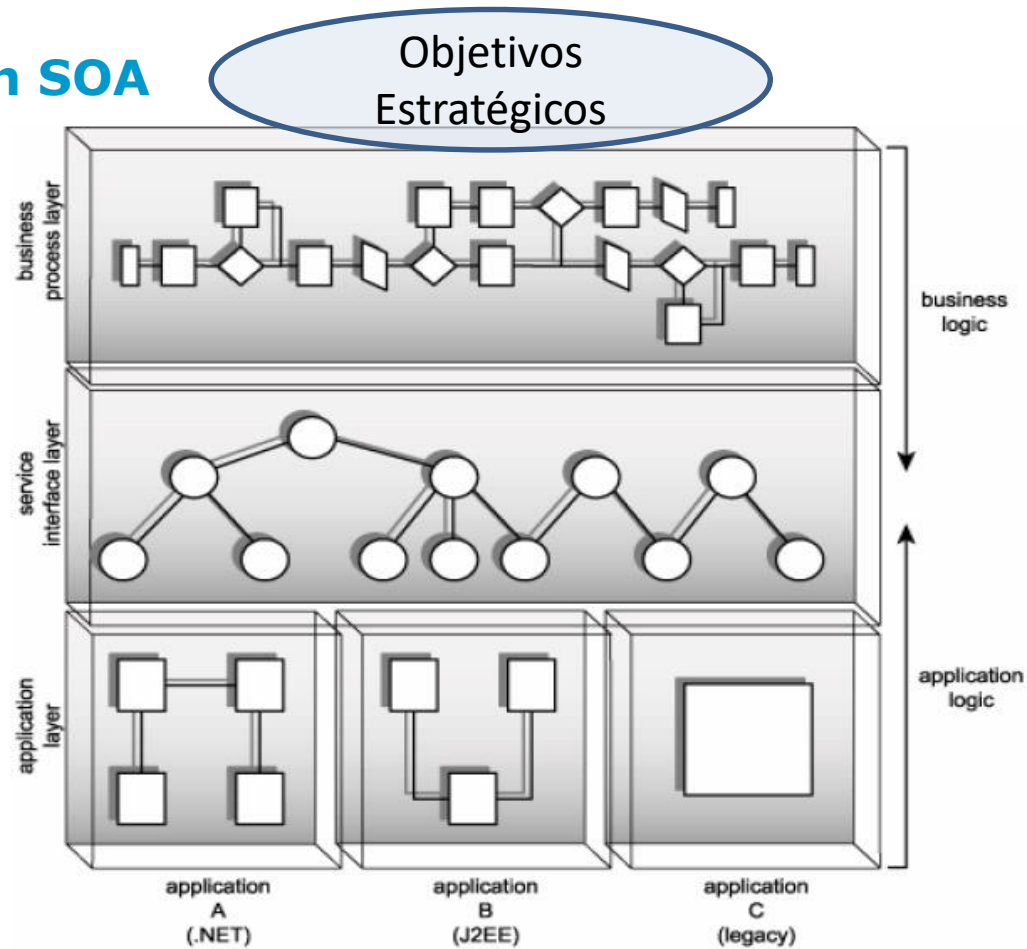
- Paradigma que facilita la alineación del negocio y la tecnología
  - Modelo ágil ante el cambio
    - Mejor respuesta → aumento de competitividad
    - Menos coste en el diseño e implementación de soluciones IT
    - Más esfuerzo en la definición y análisis
      - Enfoque *top-down* puro

- ¿Por qué SOA?
  - Competencia del mercado
    - Soluciones flexibles y rápidas
    - Mejores servicios
    - Adaptación rápida al cambio
  - Demanda de colaboración
    - Compartir información entre clientes y proveedores
    - Mejora de la conectividad
    - Integración
  - Apertura en nuevos mercados
  - Cambios tecnológicos
    - Sistemas heredados obsoletos
    - Sistemas rígidos que dificultan los cambios

## Visión clásica

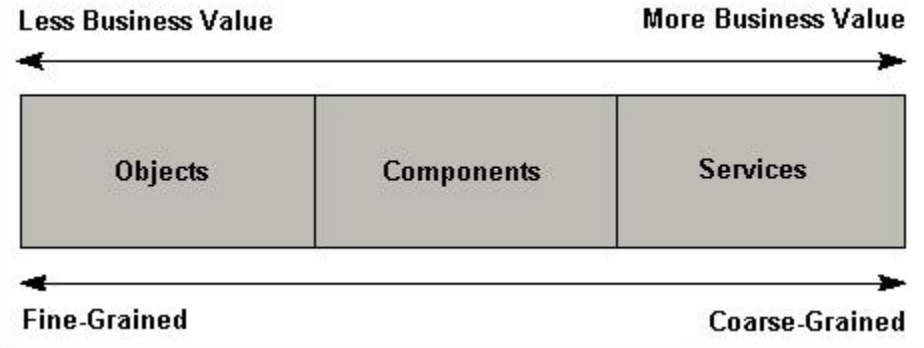


## Visión SOA



- Componentes de SOA

- Proveedor de servicio
- Consumidor de servicio
- Definición de servicio
- Intermediario de servicio (service broker)
  - El service broker, está destinado a ser un registro de los servicios, y almacena la información sobre los servicios disponibles y que pueden utilizarlos.
  - Por ejemplo, UDDI, que fue originalmente concebido como un registro de servicios web ahora se considera un Service Broker SOA.
- Registro de servicio



- Principios (T. Erl y Marks & Bell)

- Entidades autónomas y autocontenidas
- Reusables
- Duraderos
- Contratos bien definidos
- Interoperables
- Con capacidad de composición
- Funcionalidad de alto nivel de abstracción
  - Funcionalidades de grano grueso (coarse grained)
  - Abstracción de la lógica subyacente
- Acoplamiento débil
- Con capacidad de descubrimiento, publicación y localización
- Alineados con el negocio
- Sin estado

Por definición una operación de servicio de **grano grueso** tiene un **alcance más amplio** que un servicio de grano fino.

Requiere un **aumento** de la **complejidad del diseño**, pero puede reducir el número de llamadas necesarias para completar una tarea.



MTIS

SOA

# Modelos de Servicios

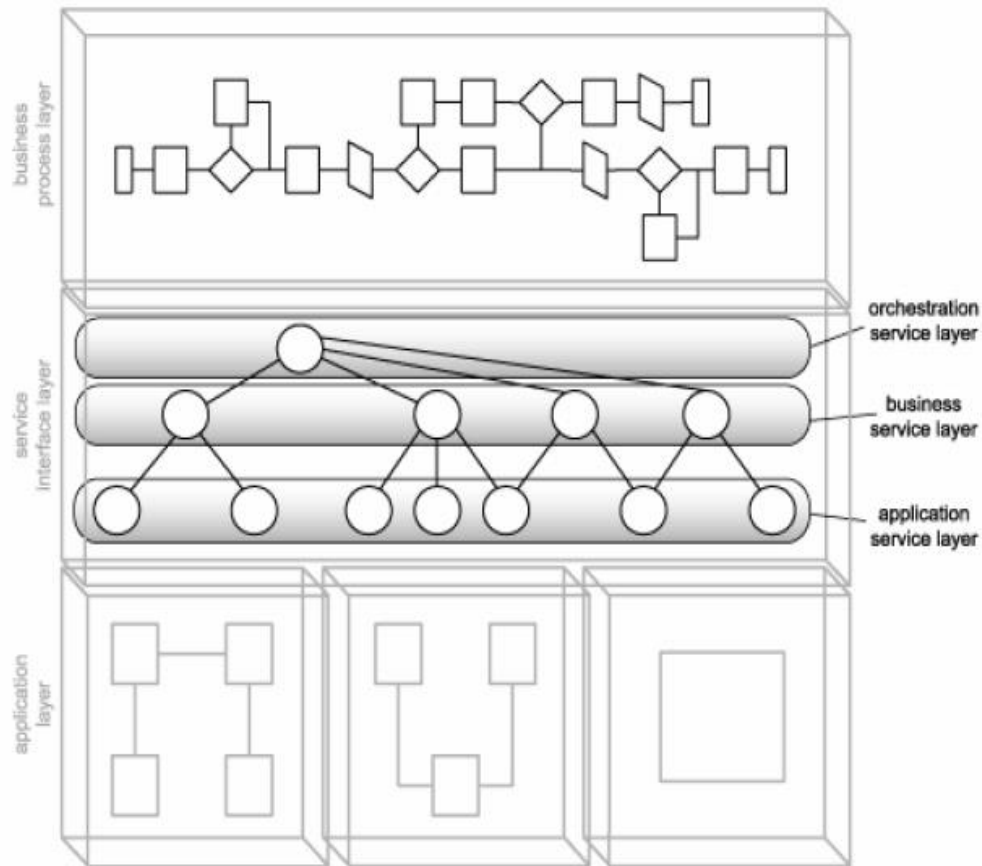
- En (Marks & Bell, 2006) se definen dos tipos de servicios:
  - Los primeros son denominados **servicios de negocio**.
    - Éstos representan conceptos de negocio y sus relaciones, son normalmente asociados con la ejecución de funciones del negocio de la organización o del dominio de negocio.
  - Los segundos son llamados ***servicios tecnológicos*** *y representan funcionalidades de* aplicaciones que son utilizadas por los anteriores.

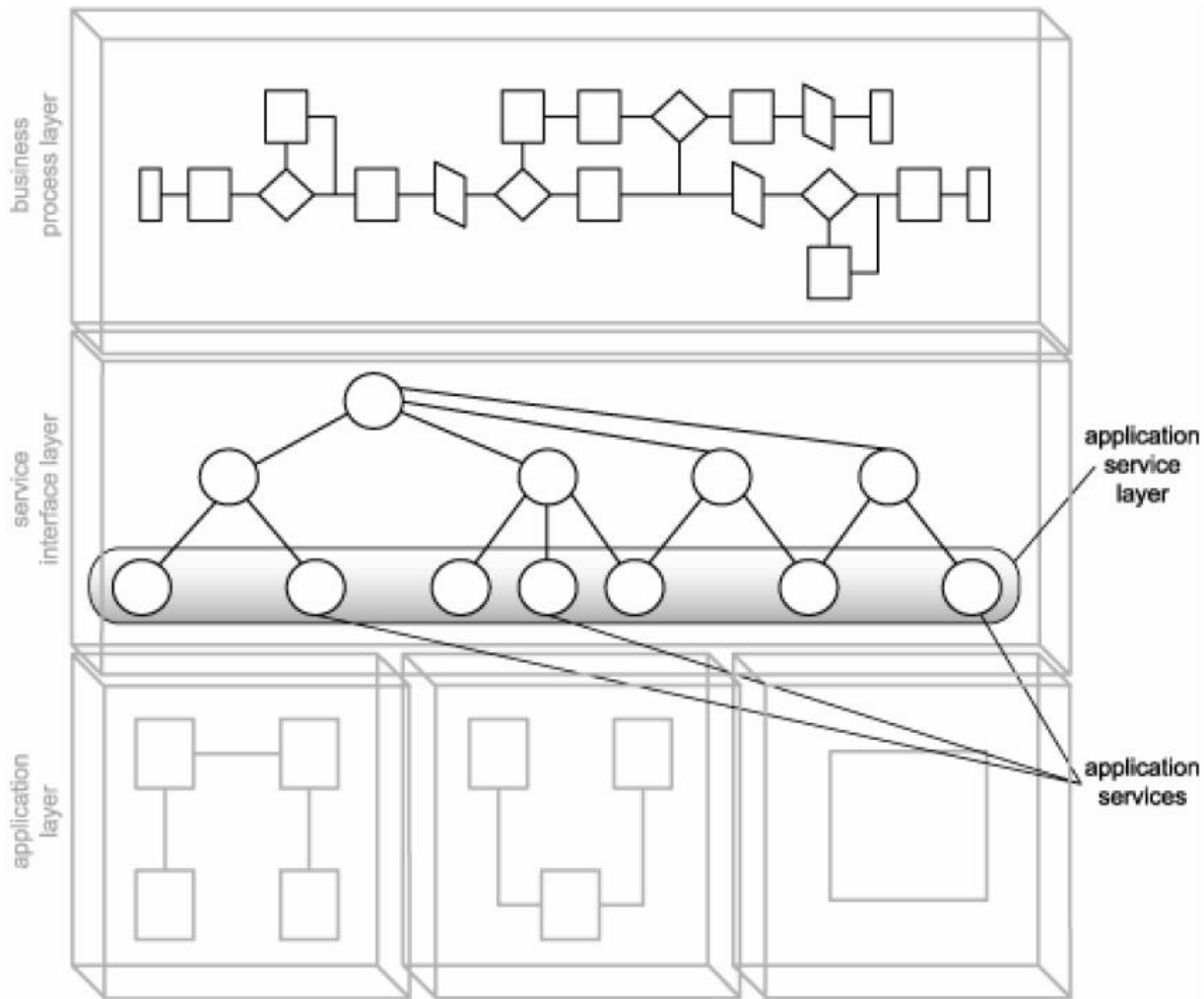


- Clasificación ([Marks & Bell, 2006](#))
  - Servicios de negocio
    - Conceptos de negocio
    - Mayor abstracción
    - Ejem. *Abrir una cuenta bancaria*
  - Servicios Técnicos
    - Conceptos tecnológicos
      - Funcionalidades transversales
    - Menor nivel de abstracción
    - Abstraen de la tecnología subyacente
    - Utilizados por los de negocio
    - Ejem. *Seguridad*
  - Enfoque muy genérico
    - Poco preciso
    - Puede dificultar su gobierno

- Clasificación (Erl, 2009)
  - Modelo basado en capas funcionales
    - Patrón *Logical Inventory Layer Pattern*
      - El inventario de servicios es estructurado en varias capas lógicas funcionales
    - 3 capas funcionales
      - Capa de servicios de aplicación
      - Capa de servicios de negocio
      - Capa de servicios de orquestación

## Arquitectura SOA





- Capa de servicios de aplicación (Características)
  - Capa más baja del modelo
    - Utilizados por las capas superiores
  - Exponen **funcionalidad** específicamente **tecnológica**.
  - Su propósito es proporcionar funciones **reutilizables** relacionadas con el procesamiento de datos en entornos nuevos o de aplicaciones heredadas.
  - Exponen la funcionalidad dentro de un contexto de procesamiento específico.
  - Hacen uso de los recursos disponibles dentro de una determinada plataforma.

- Capa de servicios de aplicación (Características)
  - Son una solución **agnóstica**.
  - Son **genéricos y reutilizables**.
  - Pueden ser utilizados para lograr la integración de punto a punto con otros servicios de aplicaciones .
  - A menudo son inconsistentes en cuanto a la granularidad de la interfaz que exponen .
  - Pueden consistir en una mezcla de servicios desarrollados a medida y servicios de terceros que se han comprado o arrendado.

- Ejemplos típicos de los modelos de servicio implementadas como servicios de aplicación:
  - **Utility services**
    - Funcionalidades tecnológicas de tipo genérico
    - Independientes del dominio de aplicación
      - » Agnósticos
        - Hace referencia a que la lógica del servicio se centra en la entidad objeto del servicio, olvidando por completo el uso que se le vaya a dar en cualquier sistema de información que vaya a usar el servicio.
    - Conocidos también como
      - » Servicios de soporte
      - » Servicios de infraestructura
  - **Wrappers services**
    - Integración de aplicaciones o sistemas heredados

- Utility services:
  - Cualquier servicio web genérico o agente de servicio diseñado para su **reutilización** potencial pueden ser clasificados como utility service.
  - La clave para alcanzar esta clasificación es que la **funcionalidad sea reutilizable**, que no sea específica en la naturaleza y que sea completamente genérica.
  - Se utilizan dentro de SOA de la siguiente manera:
    - como servicios que permiten la reutilización de las características con SOA.
    - como servicios de intermediación de una solución agnóstica.
    - como los servicios que promueven la interoperabilidad característica intrínseca de SOA.
    - como los servicios con el más alto grado de autonomía.



- Ejemplo Utility services:
- Tenemos descritos ocho Servicios Web, ¿Cuáles son servicios de negocios y cuales Utility services?
  - Servicio de Pagos
  - Servicio de Política de seguridad
  - Servicio de envío de la factura
  - Servicio de contabilidad
  - Servicio de Balanceo de carga
  - Servicio de Cumplimentación de ordenes
  - Servicio de Orden de Compra
  - Perfil de proveedor de servicio

- Ejemplo Utility services:
- Tenemos descrito ocho Servicios Web, ¿Cuáles son servicios de negocios y cuales Utility services?
  - Servicio de Pagos = servicio de negocio
  - **Servicio de Política de seguridad = utility service**
  - Servicio de envío de la factura = servicio de negocio
  - Servicio de contabilidad = servicio de negocio
  - **Servicio de Balanceo de carga = utility service**
  - Servicio de Cumplimentación de ordenes = servicio de negocio
  - Servicio de Orden de Compra = servicio de negocio
  - Perfil de proveedor de servicio = servicio de negocio
- El equilibrio de carga y servicios de Política de seguridad se clasifican como utility service, porque **proporcionan funcionalidad genérica que puede ser reutilizada por distintos tipos de aplicaciones.**
- La lógica de la aplicación de los servicios restantes es específica a una determinada tarea o solución de negocio, lo que los convierte en los servicios centrados en el negocio.

- *Wrappers services*

- Es un tipo de servicio de integración que **encapsula y expone la lógica** que reside dentro de un **sistema heredado**.
- Son proporcionados comúnmente por sistemas heredados y por tanto, introducen con frecuencia interfaces no estandarizadas.
- "El servicio de envoltorio", que también es más ampliamente conocido como "legacy wrapper" se utiliza para envolver un componente heredado para **evitar problemas de acoplamiento en soa**.
- *Podría decirse que todo en **SOA es un envoltorio**. Es decir, un servicio web es sólo una fachada, y no hay manera de decir si estamos frente a "aplicaciones heredadas" o "un conjunto de servicios complejos".*

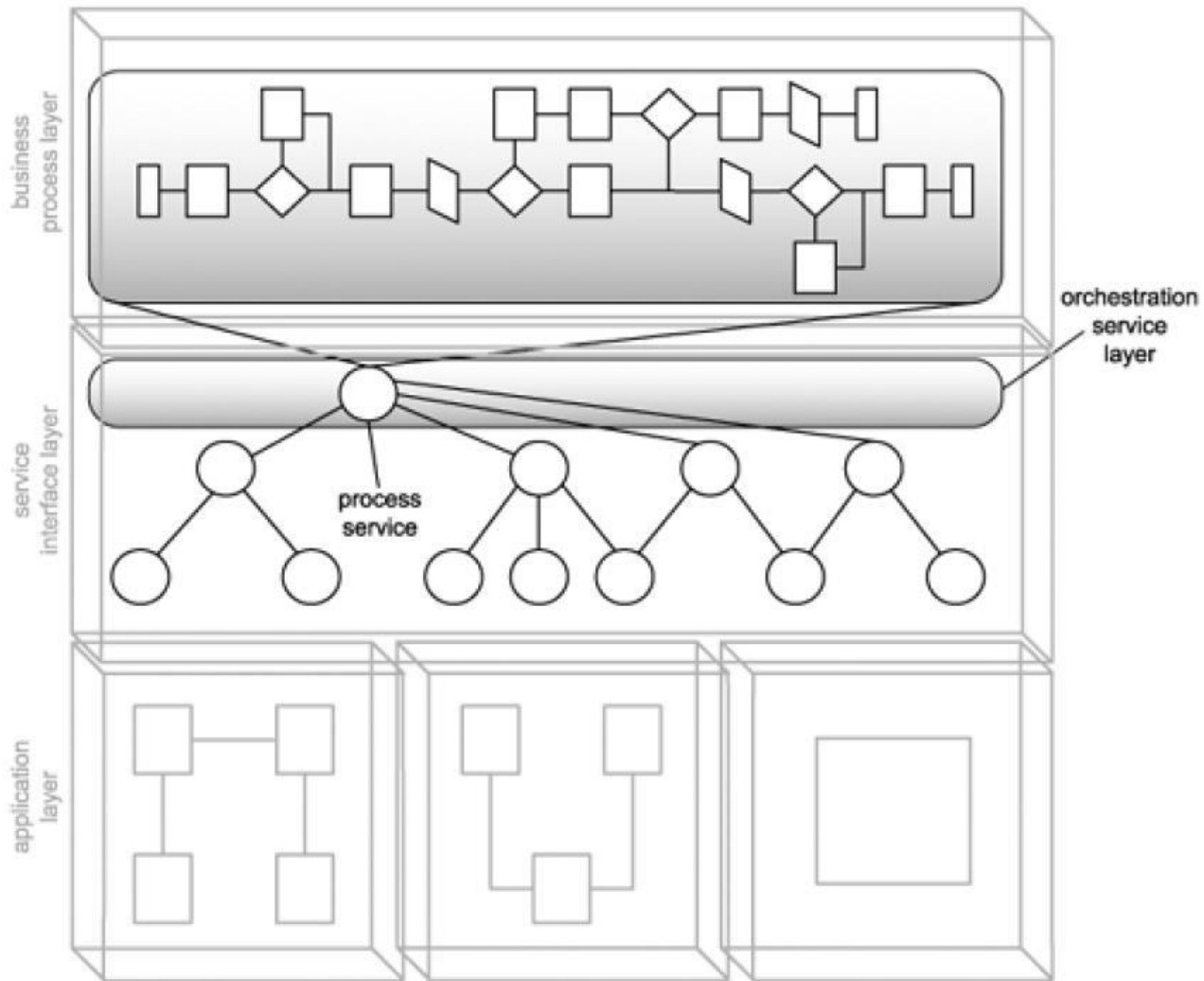
- Capa de servicios de negocio
  - Capa intermedia del modelo
    - Utilizados por la capa de orquestación
  - Mientras que los servicios de aplicación son los encargados de representar la tecnología y la lógica de la aplicación, la capa de servicios de negocio *presenta un servicio que trata sólo de representar la lógica de negocio.*
  - Servicios que impliquen lógica de negocio
  - Tipos
    - Orientados a la tarea (*task-centric business services*)
    - Orientados a la entidad (*entity-centric business services*)

- Capa de servicios de negocio
  - Tipos
    - Orientados a la tarea (*task-centric business services*)
      - Servicios que exponen lógica de proceso de negocio
      - Generados desde la primera toma de requerimientos de negocio
        - » Casos de uso y definición de procesos
      - Ejemplos como *VerificarFactura* o *GenerarHistoricoDeInformes*
    - Orientados a la entidad (*entity-centric business services*)

- Capa de servicios de negocio

- Tipos

- Orientados a la tarea (*task-centric business services*)
    - Orientados a la entidad (*entity-centric business services*)
      - Servicios que exponen lógica de entidades de negocio
      - Más a largo plazo desde el análisis de requerimientos de negocio
        - » Como resultado de la alineación entre servicios de negocio y modelos de negocio existentes en la organización
      - Ejemplos como *CuentasBancarias o Facturas*



- Capa de servicios de orquestación
  - Capa superior del modelo
  - Más cercana al negocio de la organización
    - Servicios que representan los procesos de negocio de la organización mediante composición de otros servicios
    - Coordina el flujo de trabajo de los servicios
  - Incrementa la agilidad operacional de la organización
  - **Process services**
    - *Representan procesos de negocio* de la organización.
    - Componen otros servicios que proporcionan un conjunto específico de funciones, independiente de las reglas de negocio y la lógica específica del escenario, requerida para ejecutar una instancia de proceso.



- Agnostic services

- Servicios más propensos a ofrecer lógica reutilizable :

- Entity-centric business services, están diseñados para proporcionar un conjunto de características que proporcionan datos de gestión relacionado sólo con sus correspondientes entidades.
      - Por tanto, son servicios agnósticos.
      - Los mismos Entity-centric business services pueden (y deben ) ser reutilizados por diferentes procesos o servicios a las empresa.
    - Servicios de aplicaciones, idealmente se construyen de acuerdo con el modelo de servicio público .
      - Esto los convierte en muy genéricos , reutilizables, como una solución agnóstica. Diferentes soluciones orientadas a servicios, pueden (y deben ) reutilizar los mismos servicios de aplicación.

- **Agnostic services**
  - Facilitan la integración
  - Reaprovechamiento de soluciones heredadas
  - Agilidad y flexibilidad en el negocio
    - Rápida adaptación a los cambios
  - Reusabilidad de servicios

- Agnostic services
  - Reducción de costes TI
  - Aumento de productividad
  - Reducción del time to market
    - Salida a mercado o producción
  - Independencia de la tecnología
    - Cambio de tecnologías sin cambio en servicios
    - Cambio de procesos de negocio sin cambiar servicios y sistemas TI subyacentes

# Errores más comunes acerca de SOA

- Errores más comunes acerca de SOA
  - Una aplicación que utiliza servicios web esta orientada al servicio:
    - Para aprovechar todo el potencial de la arquitectura SOA, es necesario estandarizar la forma en que se sitúan los servicios Web y diseñarlos de acuerdo con los principios SOA.

- Errores más comunes acerca de SOA
  - SOA es un término de marketing utilizado para los servicios Web como una marca:
    - El término "SOA" de manera adecuada o no adecuada se utilizan en exceso para la comercialización de los servicios web.
    - Se ha convertido en una palabra de moda.
    - SOA no es un invento de los medios o de algún departamento de marketing.
    - Representa una arquitectura diferenciada, basada en un conjunto de principios distintos.
    - SOA actualmente implica el uso de un conjunto de tecnologías utilizadas para hacer realidad los principios fundamentales de SOA.
      - La plataforma de tecnología de elección es actualmente , los servicios Web y todo lo que conllevan estos.

- Errores más comunes acerca de SOA
  - SOA simplifica la computación distribuida:
    - Los principios detrás de SOA son relativamente simples en la naturaleza.
      - Sin embargo, la aplicación de estos principios en la vida real puede llegar a ser una tarea compleja.
    - Aunque SOA ofrece un potencial de beneficio significativo, estos se consiguen mediante la inversión en el análisis minucioso y la adhesión a los principios de diseño de la orientación a servicios.

- Errores más comunes acerca de SOA
  - SOA simplifica la computación distribuida:
    - Cuando los requisitos de integración emergen.
    - Cuando existe un suficiente número de servicios componible.
    - Cuando los principios de orientación a servicios están bien integrados en una organización,
      - Es cuando SOA puede simplificar la entrega de requisitos de automatización.



- Errores más comunes acerca de SOA
  - Una aplicación con servicios web que utiliza las extensiones WS-\* es orientada a servicios.
    - Lo que los convierte en parte de una arquitectura orientada a servicios es que la **forma en la propia arquitectura esté bien diseñada**.
    - La **mayoría** de las soluciones que emplean seriamente el uso de **WS-\* extensions, son orientadas a servicios**.
    - La tasa de adopción de SOA coincide con WS-\*, en el desarrollo y productos middleware.
  - “Si entiendes los servicios Web no vas a tener problema en construir SOA”
    - Los principios fundamentales de orientación a servicios son independiente de la tecnología.

- Errores más comunes acerca de SOA
  - No usar estándares
  - Falta de plan de transición
  - No comprender las necesidades de procesamiento SOA
  - Desentenderse de plataformas de productos y desarrollo de estándares
    - Limitarse a una tecnología

# Metodología SOA

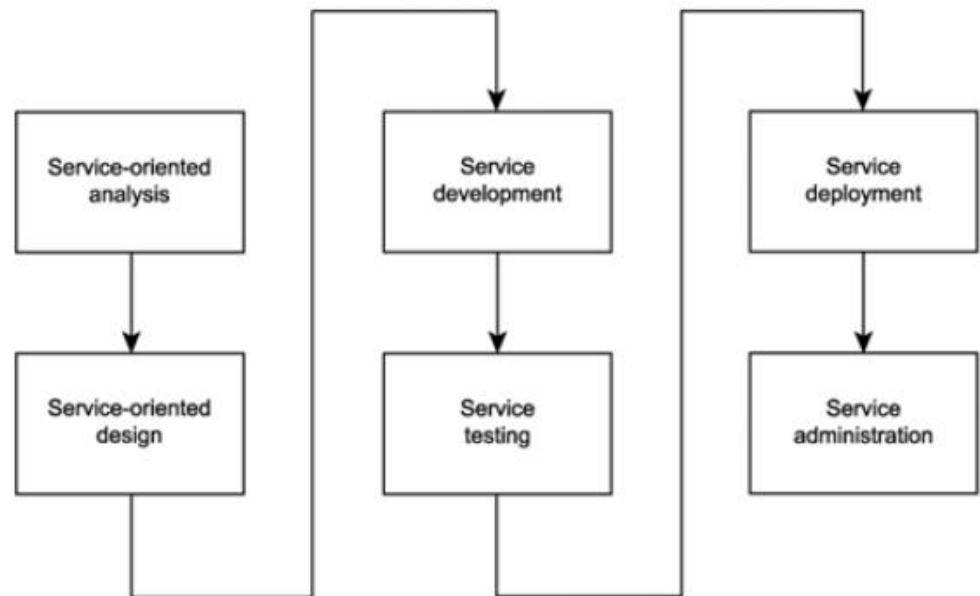
- Metodología soa
  - No hay una metodología de facto.
  - El ciclo de vida de SOA básico consiste en una serie de **fases** similar a las utilizadas regularmente en el desarrollo de proyectos.
  - SOA introduce **consideraciones específicas** en cada **fase** de la construcción y entrega de servicios.

- Metodología soa

- Existen diferentes estrategias de cómo organizar las etapas del ciclo de vida para permitir la entrega de capas de servicios especializados.
- Proceso Iterativo.
  - Identificación y modelado de servicios.
- Estrategias centradas en el análisis y diseño.
  - Descomposición de servicios.
  - Define los principales servicios necesarios para la compañía
    - catálogo de servicios.

- Estrategias
  - Buttom-up (First code)
  - Top-down (First definition)
  - Meet-In-the-middle (Agile strategy)

- Fases generales
  - Identificar los servicios
  - Construir los servicios
  - Ejecutar los servicios
  - Gestionar los servicios



- Múltiples propuestas de grandes empresas y principales expertos en SOA
  - SOMA de IBM (Service Oriented Modeling Architecture)
  - OASIS SOA Adoption Blueprints
    - <https://www.oasis-open.org/committees/download.php/15071>
  - MSOAM de Thomas Erl (The Mainstream SOA Methodology)
  - SOMF de Michael Bell (Service Oriented Modeling Framework)
- Otros enfoques centrados en el diseño software
  - UML profile for SOA (IBM)

- Planificación Adopción de SOA
  - Es durante esta etapa inicial donde **se toman las decisiones de planificación fundacional**.
  - Evaluar y determinar una serie de factores, incluyendo:
    - alcance de inventario de servicio planificado y el Estado de destino final .
    - hitos que representan estados intermedios de destino.
    - línea de tiempo para la realización de los hitos y el esfuerzo general de adopción.
    - La financiación disponible y modelo de financiación adecuado.
    - Sistema de gobierno .
    - Sistema de gestión.
    - Metodología .
    - la evaluación de riesgos .
- Además, los requisitos previos deben ser definidos con el fin de establecer criterios utilizados para determinar la viabilidad global de la adopción de SOA.

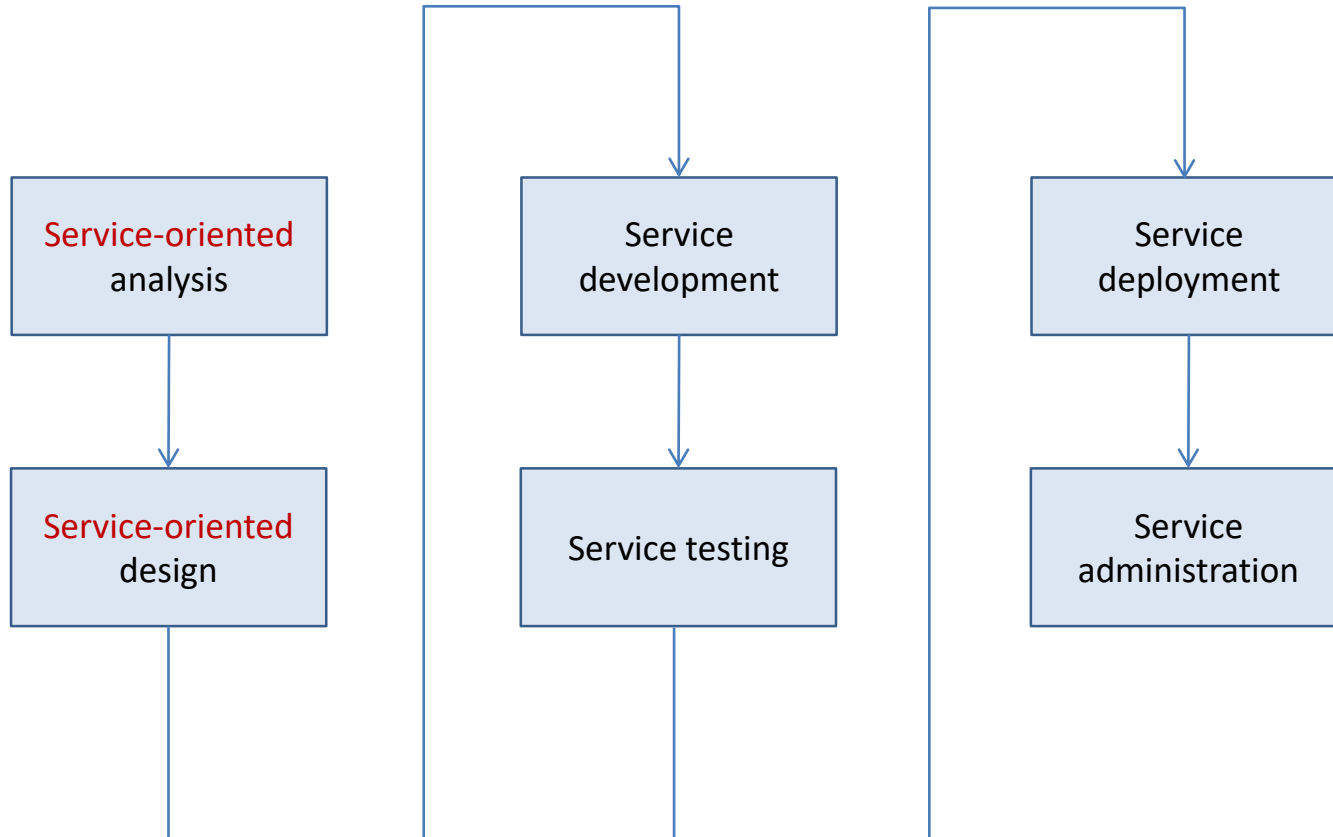


# Introducción MSOAM

The Mainstream SOA Methodology

Thomas Erl

- MSOAM
  - Fases del ciclo de vida proyecto SOA



- Service Oriented Analysis

- Proceso de análisis formal para obtener información inicial del dominio de negocio.
- Aplicado iterativamente por cada proceso de negocio.
- Para su éxito es necesaria la colaboración entre analistas de negocio y arquitectos de tecnología.
- Apoyado por el subproceso de modelado (service modeling) .
- Resultado desde el punto de vista conceptual.
  - Service Inventory Blueprints.
    - Modelo de capas de servicios.
    - Servicios candidatos.
    - Capacidades candidatas de los servicios.
    - Composiciones candidatas entre servicios

- Service Oriented Design
  - Parte del *Service Inventory Blueprints*.
  - Distingue entre servicios candidatos (conceptuales) y servicios (físicos).
  - Dedicado a la producción de contratos de servicios.
  - Representación formal del enfoque *contract-first*.

- Estrategias SOA

1. Fase previa de planificación

1. Puede incluirse en el análisis

2. Top-down

1. Enfoque centrado en el análisis de negocio previo
2. Parte de los procesos de negocio de la organización
3. Este enfoque es el más adecuado cuando se va a implementar un nuevo sistema
4. Comienza con la fase de análisis de inventario
  1. *Service Inventory Blueprints*
5. *Analizar como es el sistema para identificar los procesos implementados para asignarlos con el sistema y los procesos necesarios*

- Estrategias SOA

- Top-down

- Ventajas

- Alineación completa con los objetivos de negocio.
      - Maximiza la reusabilidad.
      - Captura de reglas de negocio en un nivel alto.
      - Entender mejor el contexto de negocios.
      - Dependencias pueden ser identificadas antes en el nivel más alto.
      - Las interacciones entre los diversos actores.
      - Conocemos antes la complejidad dentro de la empresa en términos de los patrones de intercambio de mensajes .

- Inconvenientes

- Incrementa los costes del proyecto → etapa de análisis.

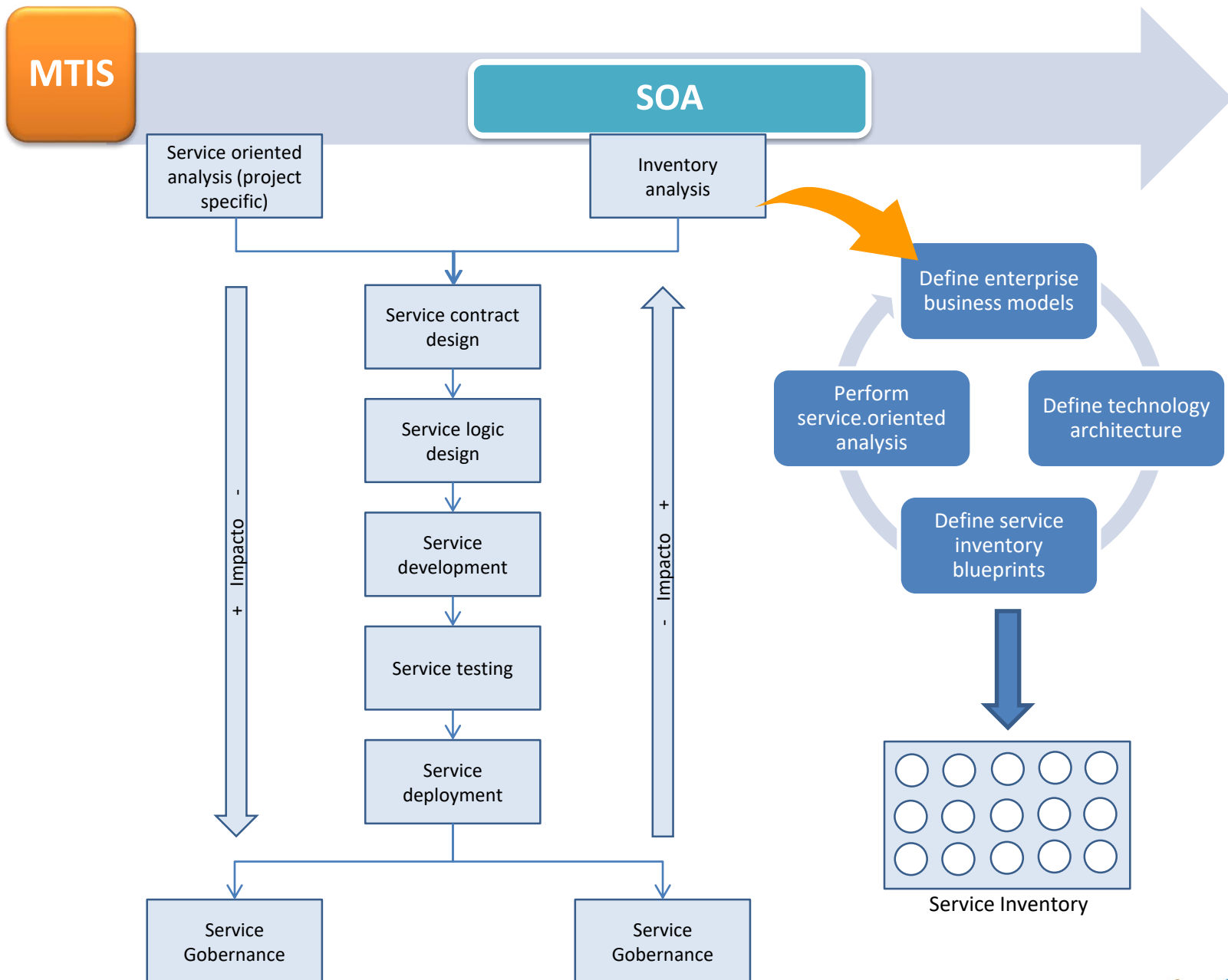
- Estrategias SOA
  - Bottom-up
    - Enfoque centrado en los requerimientos de la aplicación.
    - Generalmente se adopta este enfoque para la modernización de los sistemas heredados o sistemas que no son de SOA.
    - Requerimientos de composición e integración .
      - Implementación a partir de procesos existentes en aplicaciones.
      - Funcionalidades existentes en sistemas heredados.
    - Análisis para la redundancia y la re-factorización del esfuerzo.

- Estrategias SOA

- Bottom-up

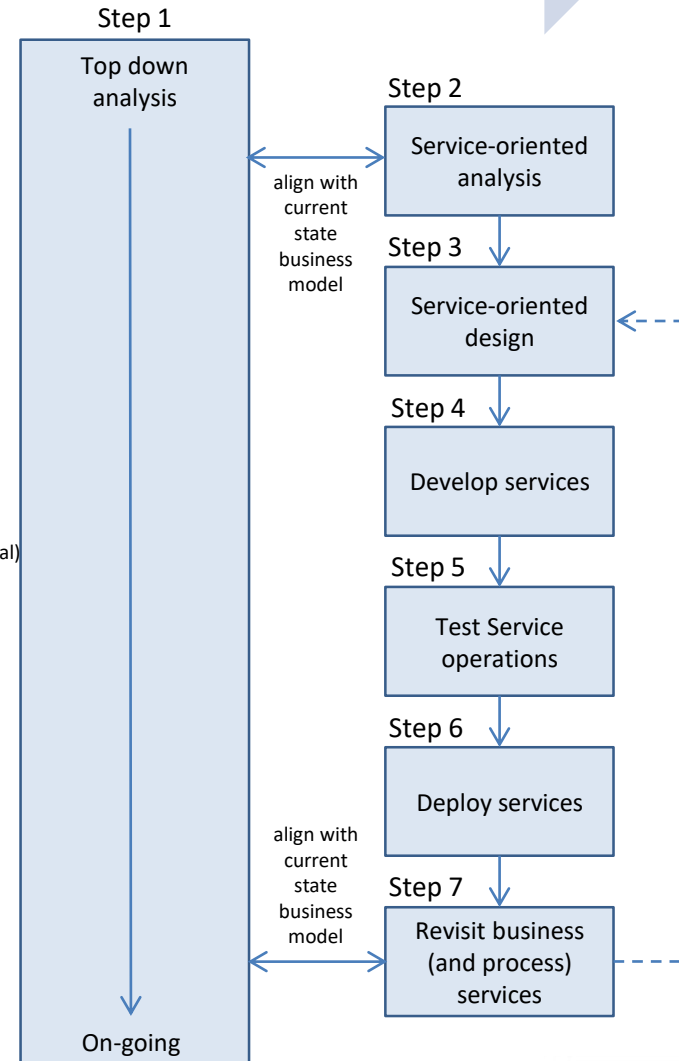
- Este enfoque no requiere necesariamente el enfoque de procesos de negocio (identificación de los procesos, el modelado y luego utilizar los servicios para poner en práctica esos procesos).
    - El más usado hoy en día
      - Aprovechar ventajas de Servicios Web.
    - Ventajas
      - Reduce el esfuerzo y tiempo para el desarrollo de servicios.
    - Problemas
      - Servicios de corta duración.
      - Incrementa el mantenimiento y las actualizaciones.





## • Estrategias SOA

- Enfoque intermedio (meet-in-the-middle)
  - Estrategia ágil
  - Inicialmente comenzar con un enfoque *top-down* para crear un inventario de servicios
  - En la realización usar bottom-up aprovechando sistemas heredados
  - Fases
    - Step1, análisis previo
      - » Ontología  
( formulación de un exhaustivo y riguroso esquema conceptual)
      - » Entidades de negocio
      - » Enfoque reducido del *top-down*
    - Step2, análisis orientado a servicios
      - » Fase anterior en progreso
    - Step3, diseño orientado a servicios
      - » Se define las capas de servicios seleccionadas
      - » Se diseñan servicios individuales
    - ...
    - Step7, Revisión de los servicios de negocio
      - » Revisiones periódicas
      - » Mejoras y modificaciones producidas



- <http://www.innovativearchitects.com/KnowledgeCenter/ESB-EAI-SOA.aspx>
- <http://www.servicetechmag.com/I51/0611-2>



**Service-Oriented Architecture: Concepts, Technology, and Design**

By Thomas Erl

.....  
Publisher: **Prentice Hall PTR**

Pub Date: **August 04, 2005**

ISBN: **0-13-185858-0**

Pages: **792**