

Arquitectura orientada a servicios

Ingeniería de Sistemas



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela de Ingeniería Informática

Service Oriented Architecture (SOA)

- Es un paradigma para organizar y utilizar capacidades distribuidas que pueden estar bajo el control de varios propietarios (dominios).
- Provee medios uniformes para ofrecer, descubrir, interactuar y utilizar capacidades para producir los efectos deseados consistentes con precondiciones y expectativas medibles.

(OASIS, Organization for the Advancement of Structured Information Standards)

Service Oriented Architecture (SOA)

Una estrategia general de organización de los elementos de IT, de forma que una colección heterogénea de sistemas distribuidos y aplicaciones complejas se pueda transformar en una red de recursos integrados, simplificada y sumamente flexible.

(Microsoft, 2006)

Service Oriented Architecture (SOA)

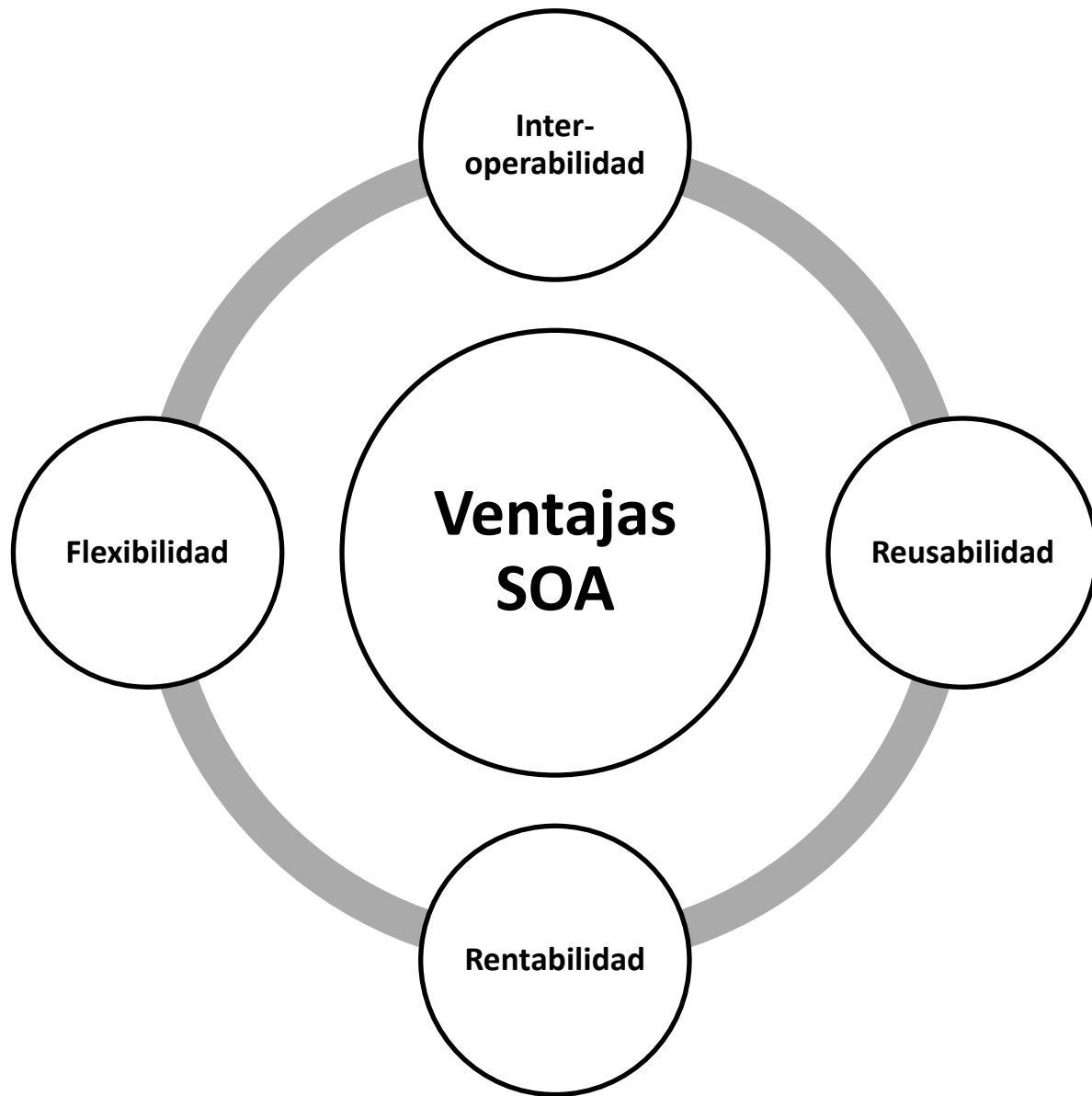
Marco de trabajo conceptual que permite a las organizaciones unir los objetivos de negocio con la infraestructura de TIC integrando los datos y la lógica de negocio de sus sistemas separados.

(Marsili, 2007).

Service Oriented Architecture (SOA)

- Es un paradigma de arquitectura para diseñar y desarrollar sistemas distribuidos.
- Las soluciones SOA han sido creadas para satisfacer los objetivos de negocio, facilitar la integración con sistemas legados, alineación directa con los procesos de negocio reduciendo costos de implementación, innovación de servicios a clientes y una adaptación ágil ante cambios incluyendo reacción temprana ante la competitividad.

[Manifiesto SOA](http://www.soa-manifesto.org/default_spanish.html): http://www.soa-manifesto.org/default_spanish.html



Principales ventajas de SOA

- **Interoperabilidad:** Permite mejorar la toma de decisiones. Al tener las aplicaciones integradas e intercambiando datos se dispone de información más exacta y actualizada.
- **Flexibilidad:** Logra optimizar y aprovechar los sistemas existentes, y adaptarse de manera rápida a nuevas aplicaciones que surgen por nuevas demandas.

Principales ventajas de SOA

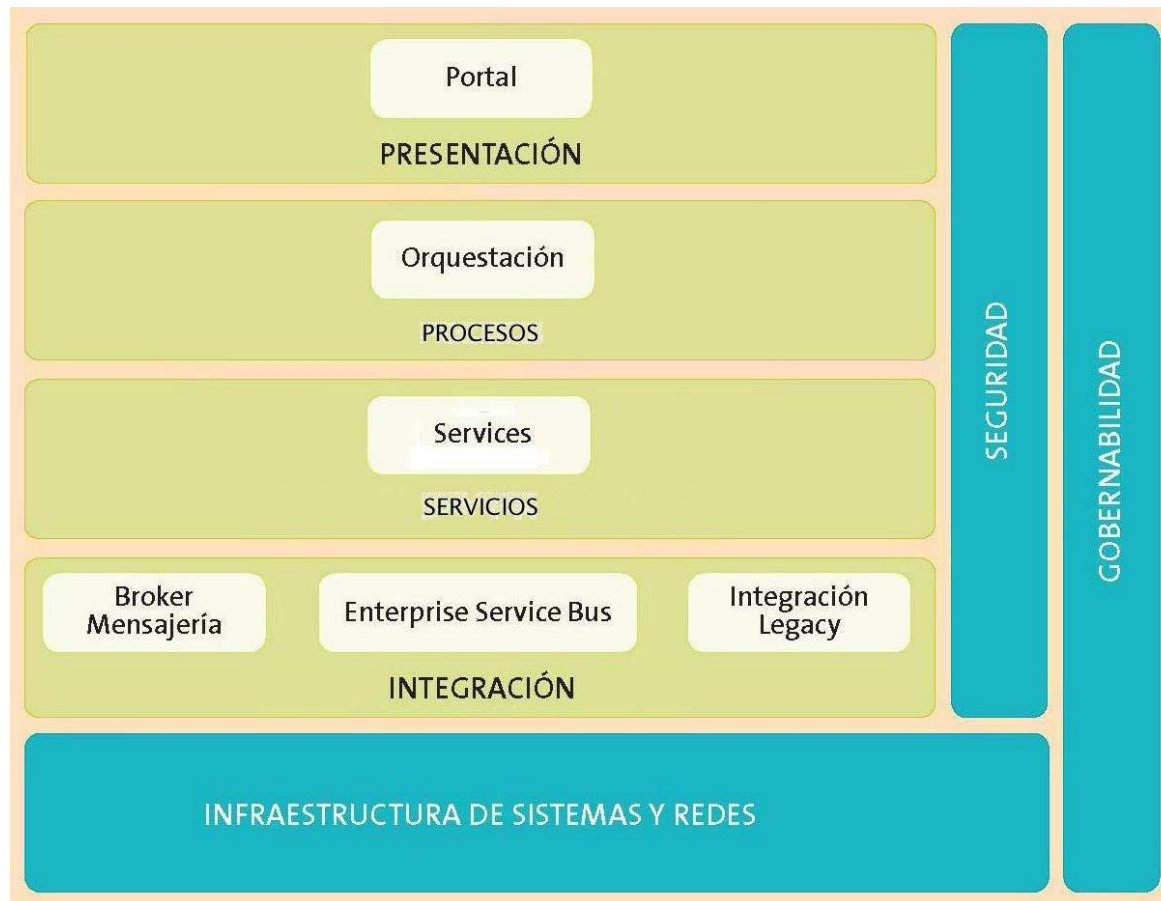
- **Reusabilidad:** Facilita la creación de un repositorio de servicios que se pueden combinar en servicios de mayor nivel y aplicaciones compuestas en respuesta a nuevas necesidades de la empresa. Se reduce el costo del desarrollo de soluciones y de los ciclos de prueba, se eliminan redundancias y se consigue su puesta en valor en menos tiempo.
- **Rentabilidad:** Relacionada con la reusabilidad y con el uso eficiente de los recursos. Ayuda a aumentar la productividad del personal al tener un mejor acceso a los sistemas y a la información.

Sistemas de información

La arquitectura SOA se usa para **definir la estrategia y modelo de orquestación** (coordinación) de los diferentes componentes tecnológicos (servicios) que soportan las necesidades de una organización.

Es un paradigma de arquitectura para sistemas de información, que busca el **mínimo acoplamiento entre sus componentes y promueve su reutilización**, favoreciendo la identificación de un conjunto de servicios en red y la definición de los procesos por los que interactúan.

Arquitectura basada en SOA



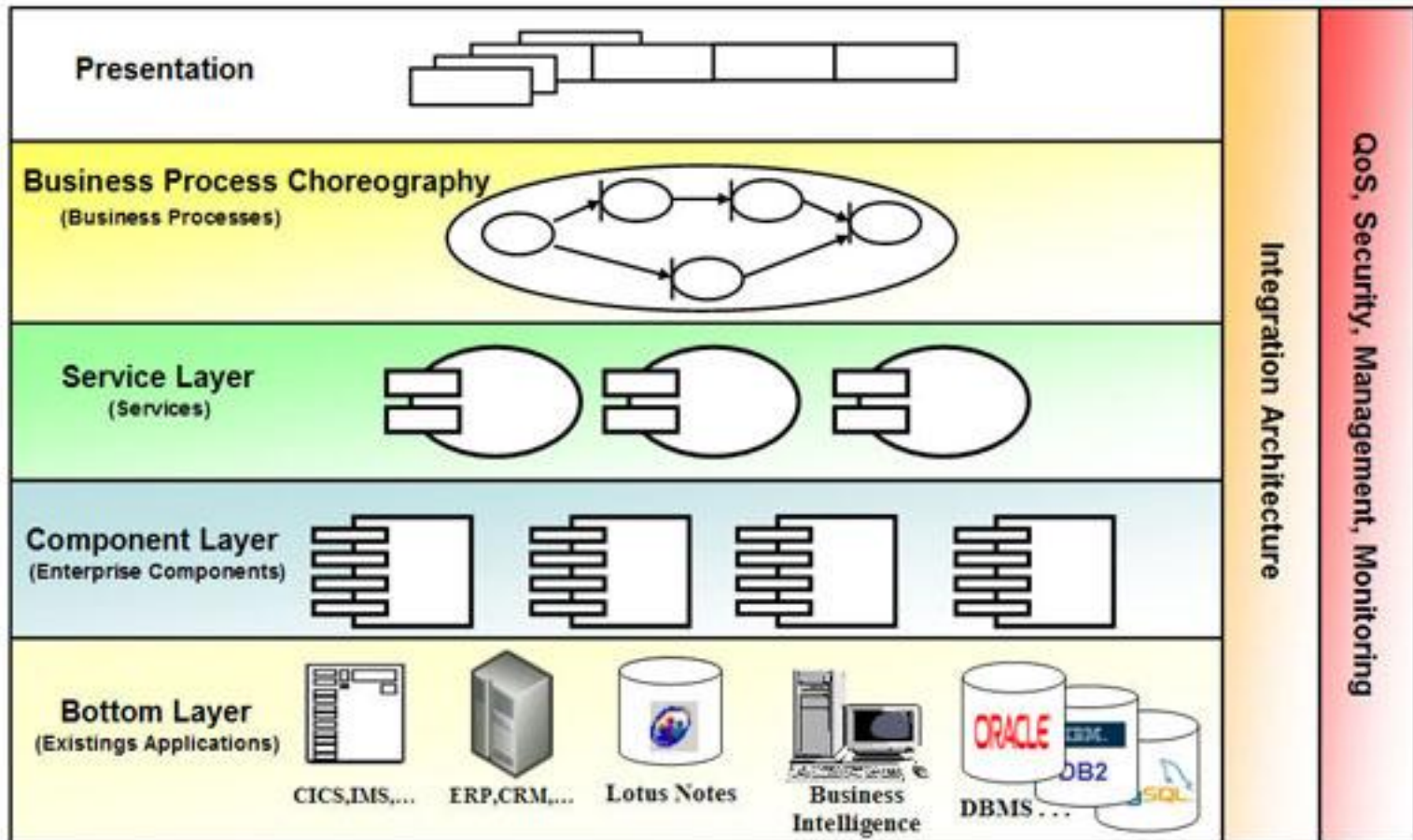
Ventajas para las organizaciones

- SOA permite **soportar servicios débilmente acoplados** (independientes) para posibilitar la flexibilidad en el negocio de manera interoperable e independiente de los componentes tecnológicos.
- SOA permite **mejorar el alineamiento de las TI con las necesidades de negocio**, y adaptarse de forma rápida a las presiones del mercado.

Arquitectura empresarial y SOA

- La Arquitectura Empresarial (AE) actúa como fuerza integradora entre aspectos estratégicos del negocio, operacionales y tecnológicos de la organización.
- El enfoque orientado a servicios en los frameworks de AE contribuye a lograr el alineamiento necesario.
- El objetivo es construir servicios comunes que sean orquestados a través de una capa de intermediación (middleware) para la implementación de los procesos de negocio.

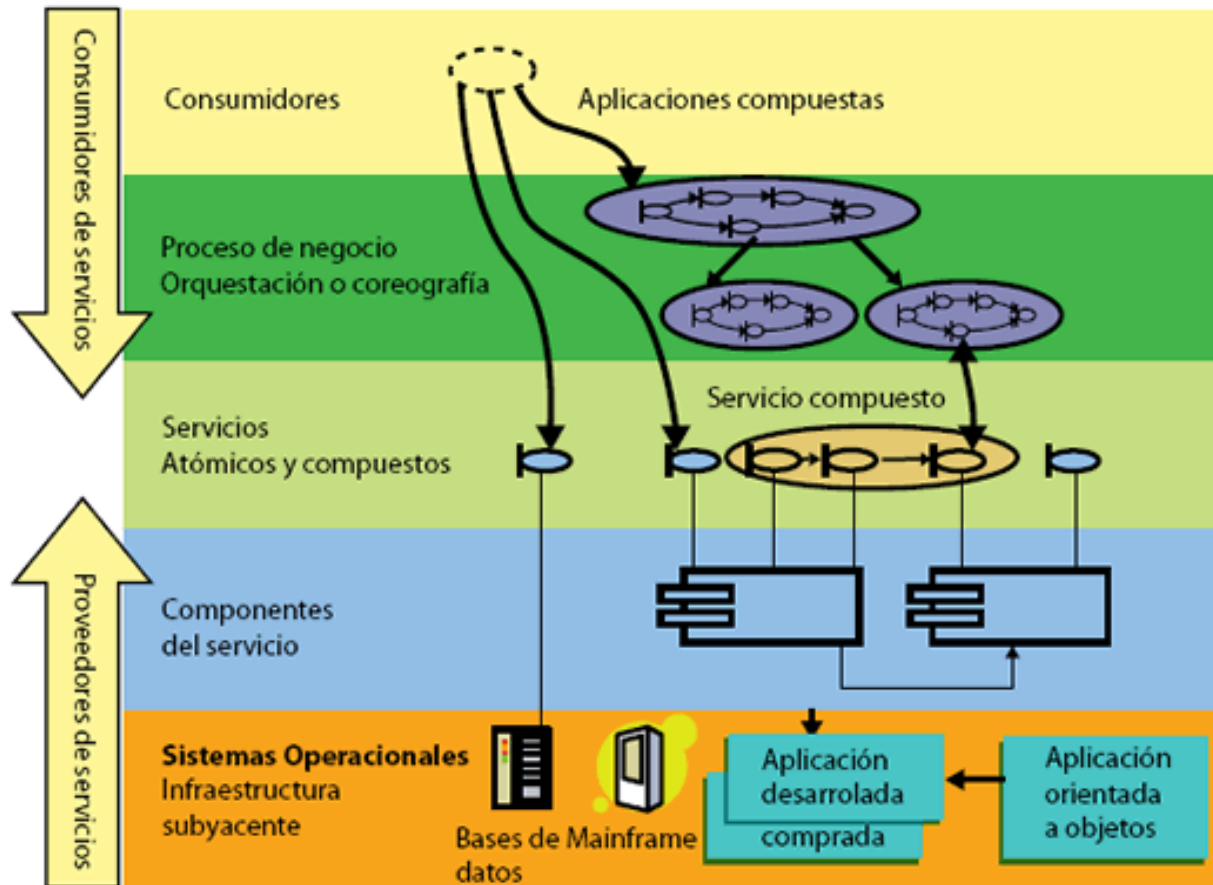
Capas de SOA



Capas de SOA

De entrega	Donde los servicios son desplegados a los usuarios finales
De composición de procesos	Que define el proceso en términos del negocio y sus necesidades, y que varía en función del negocio
De integración de servicios	Facilitan el intercambio de datos entre elementos de la capa aplicativa orientada a procesos empresariales internos o en colaboración
De exposición de funcionalidades	Donde las funcionalidades de la capa aplicativa son expuestas en forma de servicios (generalmente como servicios web)
Aplicaciones básicas	Sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad

Relación entre capas



Consumidores de servicios

- La capa de acceso o presentación donde están los consumidores de los servicios (aplicaciones compuestas, móviles, etc).
- La capa donde se encuentran los procesos de negocio modelados y donde ocurre la orquestación o la coreografía de los procesos.
- La capa donde está el catálogo de servicios a nivel empresarial que pueden ser atómicos o compuestos. Estos servicios podrían ser utilizados por procesos de negocio o directamente por consumidores. La aplicación de diferentes prácticas arquitectónicas (patrones de diseño, etc.) hacen de esta capa una arquitectura. En estas tres capas encontramos a los consumidores de servicios.

Proveedores de servicios

- La capa donde están los componentes del servicio que son los elementos que se utilizan para construir los servicios. Es importante notar que pueden ser utilizados por uno o más servicios.
- La capa de sistemas operacionales e infraestructura. Aquí es donde yacen las aplicaciones de software existentes, compradas o desarrolladas internamente, así como la infraestructura tecnológica con la que cuenta la empresa (mainframes, bases de datos, redes, etc.). Los servicios podrían interactuar directamente con elementos de esta capa o hacerlo a través de los componentes.

SOA Governance

Recomienda que las personas, la tecnología y los procesos deben coordinarse para crear y administrar las políticas y acciones necesarias que aseguren la alineación con las líneas de negocio.

Principios SOA

1. **Los servicios deben ser reusables.** Todo servicio debe ser diseñado y construido pensando en su reutilización dentro de la misma aplicación, dentro del dominio de aplicaciones de la empresa o incluso dentro del dominio público para su uso masivo.
2. **Los servicios deben proporcionar un contrato formal.** Todo servicio desarrollado, debe proporcionar un contrato en el cual figuren: el nombre del servicio, su forma de acceso, las funcionales que ofrece, los datos de entrada de cada una de las funcionalidades y los datos de salida. De esta manera, todo consumidor del servicio, accederá a este mediante el contrato, logrando así la independencia entre el consumidor y la implementación del propio servicio. En el caso de los Servicios Web, esto se logrará mediante la definición de interfaces con WSDL.

Principios SOA

3. **Los servicios deben tener bajo acoplamiento.** Los servicios tienen que ser independientes. Para lograr ese bajo acoplamiento, lo que se hará es que cada vez que se vaya a ejecutar un servicio, se accederá a él a través del contrato, logrando así la independencia entre el servicio que se va a ejecutar y el que lo llama. Si conseguimos un bajo acoplamiento, los servicios podrán ser totalmente reutilizables.
4. **Los servicios deben permitir la composición.** Todo servicio debe ser construido de tal manera que pueda ser utilizado para construir servicios genéricos de más alto nivel, el cual estará compuesto de servicios de más bajo nivel. En el caso de los Servicios Web, esto se logrará mediante el uso de los protocolos para orquestación(WS-BPEL) y coreografía (WS-CDL).

Principios SOA

5. **Los servicios deben de ser autónomos.** Todo servicio debe tener su propio entorno de ejecución. De esta manera el servicio es totalmente independiente y nos podemos asegurar que así podrá ser reutilizable desde el punto de vista de la plataforma de ejecución.
6. **Los servicios no deben tener estado.** Un servicio no debe guardar ningún tipo de información. Esto es así porque una aplicación está formada por un conjunto de servicios, lo que implica que si un servicio almacena algún tipo de información, se pueden producir problemas de inconsistencia de datos. La solución, es que un servicio sólo contenga lógica, y que toda información esté almacenada en algún sistema de información sea del tipo que sea.
7. **Los servicios deben poder ser descubiertos.** Todo servicio debe poder ser descubierto de alguna forma para que pueda ser utilizado, consiguiendo así evitar la creación accidental de servicios que proporcionen las mismas funcionalidades. En el caso de los Servicios Web, el descubrimiento se logrará publicando los interfaces de los servicios en registros UDDI.

Orquestación y Coreografía

Orquestación: Una aplicación de software BPM a modo de director de orquesta, ejecuta y controla el flujo de datos a través de los servicios que constituyen el proceso de negocio.

Coreografía: La secuencia y dependencia de las interacciones entre los múltiples servicios que implementan un proceso de negocio así como las condiciones necesarias para su invocación previamente definidas.

Proceso SOA

El diseño de los procesos de negocio y los procesos de tecnología **deben estar integrados en un mismo flujo para que funcionen como una unidad**, garantizando la independencia de cada capa.

SOMA

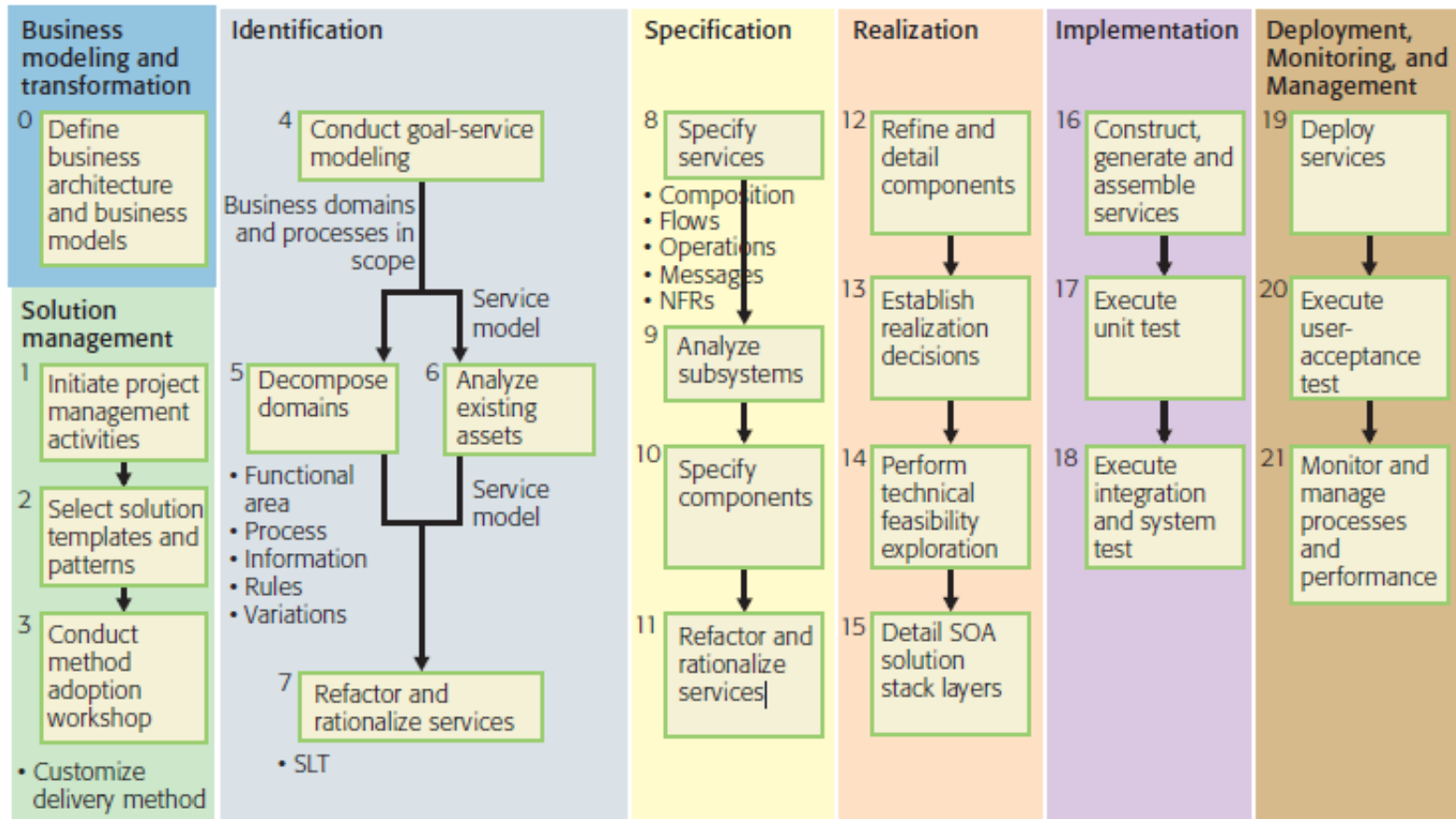
Service-oriented modeling and architecture

Es una metodología que propone un ciclo de vida de desarrollo para el diseño de soluciones SOA, que proporciona las técnicas y normas necesarias para el análisis, diseño, implementación, pruebas y despliegue de servicios, componentes, procesos, información y políticas necesarios para el diseño y la construcción de soluciones robustas y reutilizables de soluciones SOA en la empresa. (Arsanjani et al., 2008)

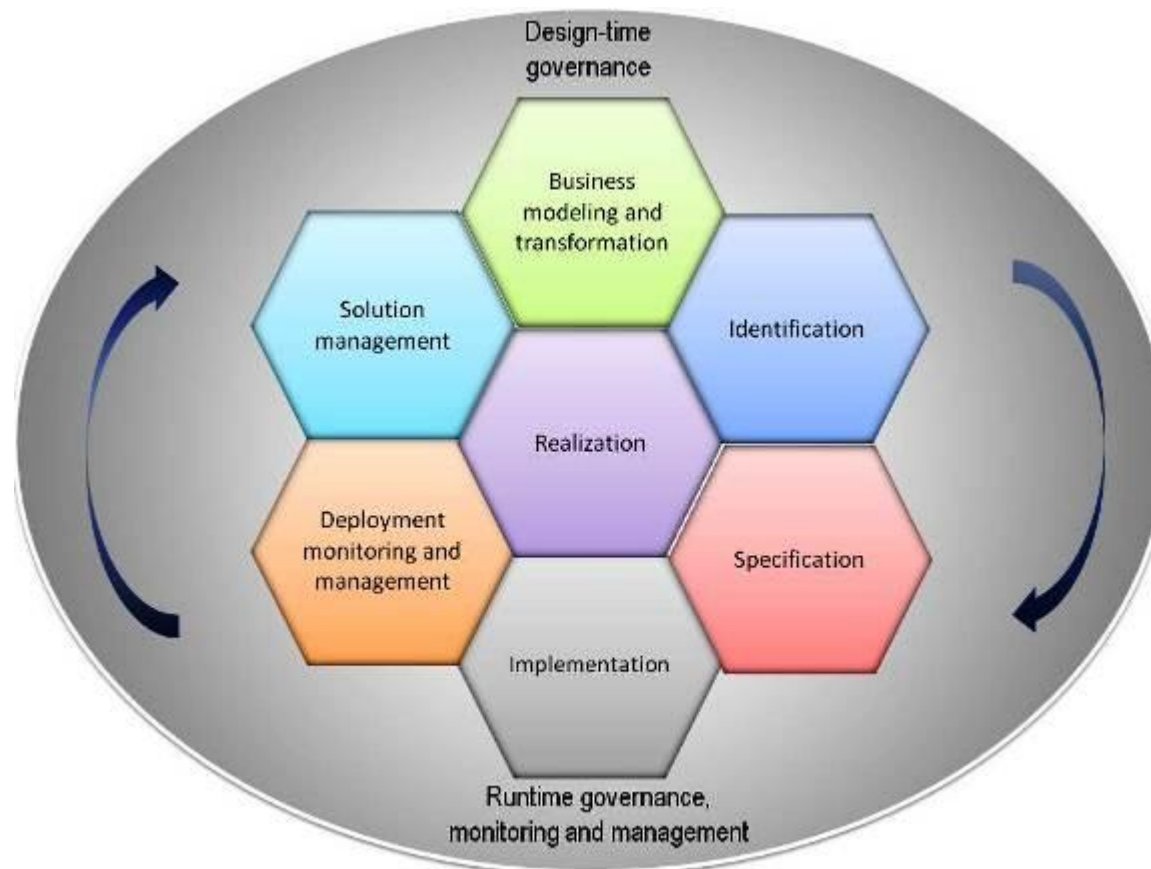
Fases de SOMA

1. Modelado y transformación del negocio: Fase del modelado, simulación y optimización del negocio completo.
2. Gestión de la solución: Como las soluciones SOA son híbridas por naturaleza, es conveniente tener unas plantillas donde se encuadre el proyecto.
3. Identificación: La fase de identificación tiene como objetivos principales el descubrimiento de los servicios, los componentes y su proceso de integración.
4. Especificación: Se elabora el modelo de los servicios en términos de las dependencias de servicios, eventos, reglas, operaciones, mensajes, requisitos no funcionales, etc.
5. Realización: Se valida el desarrollo realizando una exploración de viabilidad de la solución.
6. Implementación y ensamblado: Se construye la solución.
7. Despliegue, monitorización y gestión: Se despliega la solución, se valida por los usuarios y se pone en producción.

Ciclo de vida SOMA



Enfoque fractal de SOMA



(Arsanjani et al., 2008)

SOMA-ME

- IBM ha desarrollado también un framework de desarrollo SOMA-ME, Service-Oriented Modeling Environmet.
- El objetivo de dicho framework es el diseño de soluciones SOA utilizando SOMA basado en MDD (Model-Driven Design).

Procesos de negocio

- Los procesos de la organización se descomponen en partes reutilizables y manejables, que pueden ser implementadas como *Servicios*.
- Un servicio es una unidad de trabajo ejecutada por el proveedor del mismo para obtener un resultado que es requerido por un consumidor.

Funciones de negocio

Los servicios corresponden a **funciones de negocio** que cuando son invocadas realizan una tarea específica (consultar estado de cuenta del cliente, realizar un pedido, solicitar cita...) común a casi todas las organizaciones.

Servicios de negocio

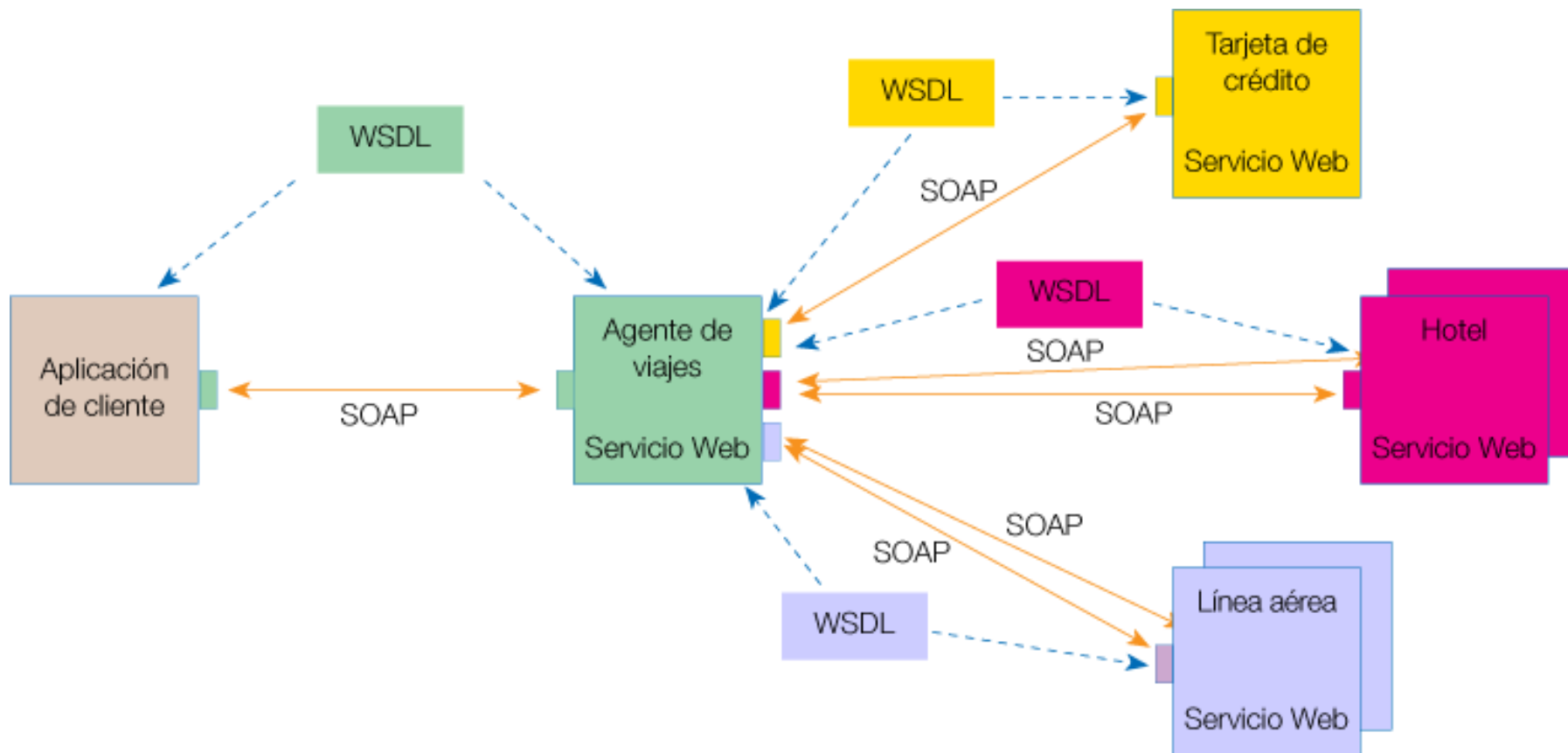
Una arquitectura SOA consta de un conjunto de **servicios de negocio** que soportan la realización de procesos de negocio de principio a fin de **una forma dinámica y reutilizable**, utilizando descripciones de servicios basadas en interfaces.

Servicios compuestos

- Un servicio puede estar compuesto de otros servicios más simples.
- Cada servicio debe aportar algún *valor* al negocio. Tiene que estar orientado a satisfacer una necesidad concreta del negocio.

Red de servicios

- Los servicios hacen disponibles sus recursos a otros servicios de la red, bajo un esquema de servicios independientes a los que se accede de un modo estandarizado.
- La comunicación entre servicios se realiza mediante su interfaz que se define siguiendo un estándar y es independiente de la plataforma, del lenguaje de implementación y de la tecnología subyacente ocultando los detalles de la implementación.



Ejemplo de arquitectura basada en servicios web (www.w3c.es)

Estándares adoptados por SOA

- BPEL y WS-Coordination que proporcionan métodos de definición, coordinación y soporte para flujos de trabajo, procesos de negocio y aplicaciones distribuidas.
- Utilización de servicios Web a través de estándares basados en SOAP, WSDL, UDDI, DML, HTTP y JMS.
- Tecnologías basadas en el concepto de servicios (no necesariamente servicios web): JMS y CORBA.



Servicios Web

- Los servicios Web se utilizan actualmente tanto dentro como fuera de los límites de una organización.
- Se han convertido en la tecnología común para proporcionar puntos de integración entre las aplicaciones.
- Los servicios Web exponen las operaciones de ciertas aplicaciones o sistemas de información.
- La combinación o composición de varios servicios Web implica la integración de las aplicaciones subyacentes y sus funcionalidades.

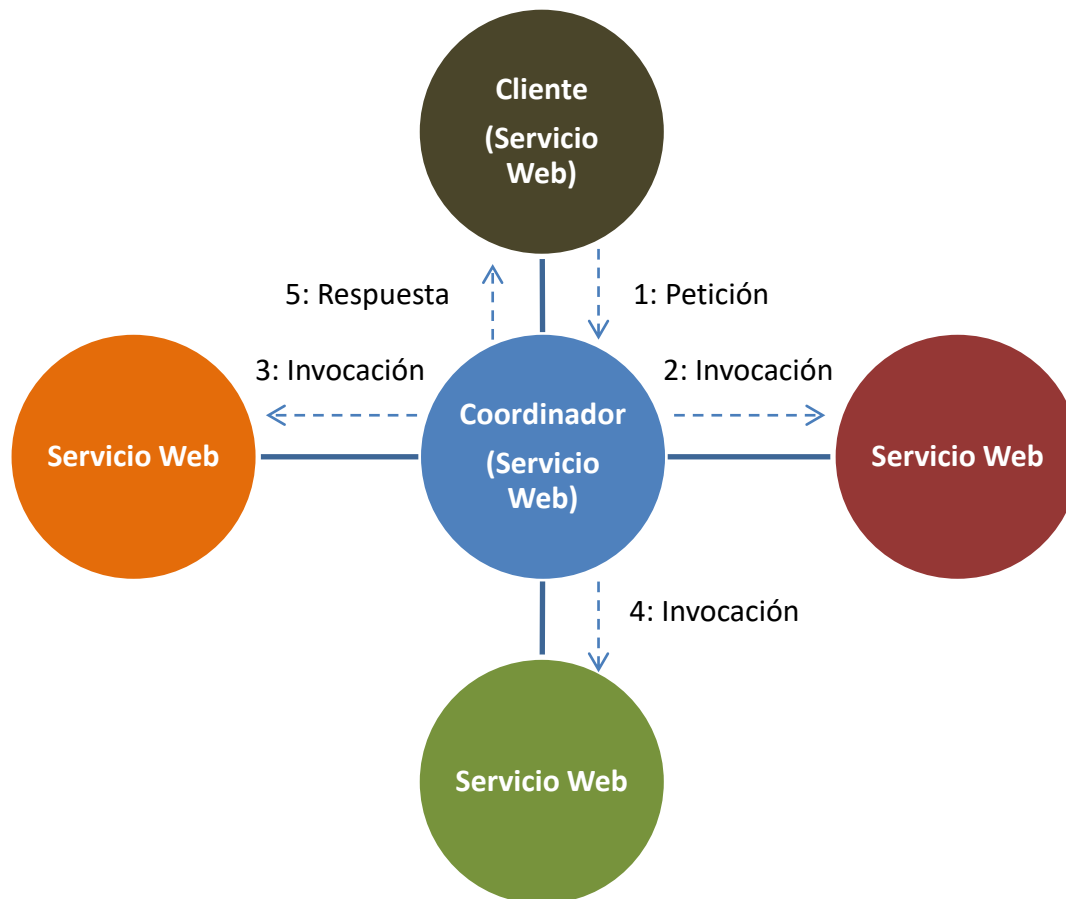
Combinación de servicios

- La orquestación es una de las posibles formas de combinación de servicios Web para obtener nuevos procesos de negocio y servicios de más alto nivel.
- Los servicios Web proporcionan una tecnología de interfaces común que unifica el modelo de integración para todos los servicios independientemente de su origen.
- Los beneficios de los servicios Web, tales como su descubrimiento en tiempo de ejecución y su bajo acoplamiento, contribuyen a la orquestación de servicios Web proporcionando un acercamiento al modelado y ejecución de procesos de negocio en tiempo real.
- La idea es construir orquestaciones con el objetivo de reducir el esfuerzo total de producir nuevos servicios y aplicaciones basadas en servicios.

Orquestación de servicios

- Un proceso central lleva el control de los servicios Web implicados en la realización de una tarea y coordina la ejecución de las diferentes operaciones sobre dichos servicios Web.
- Los servicios Web orquestados no saben que están implicados en un proceso de composición y que forman parte de un proceso de negocio de mayor nivel.
- Solamente el coordinador central de la orquestación conoce la meta, por lo que la orquestación se centraliza mediante definiciones explícitas de las operaciones y del orden en el que se deben invocar los servicios Web.

Orquestación de servicios



Ventajas de la orquestación

- La coordinación de los procesos componentes está gestionada de forma centralizada por un coordinador conocido que puede ser otro servicio Web.
- Los servicios Web pueden incorporarse sin saber que están formando parte de un proceso de negocio de mayor nivel.
- Pueden definirse escenarios alternativos para cuando ocurra algún fallo.