Integración de aplicaciones con SOA

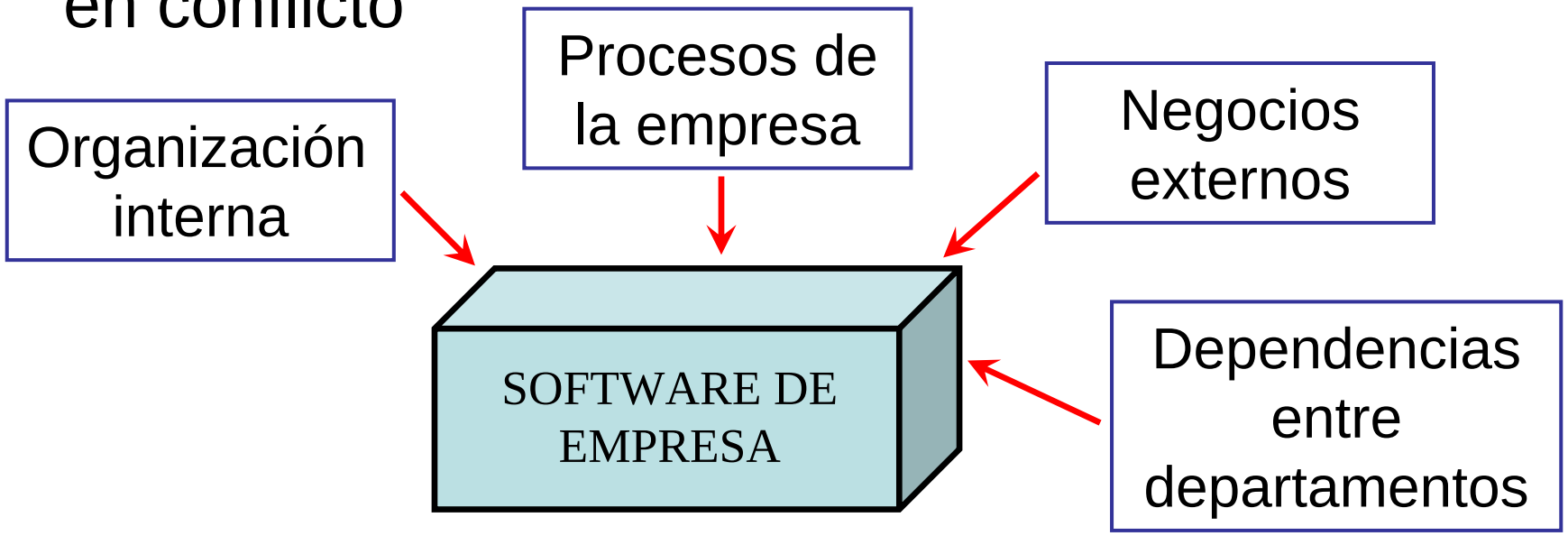
Sesión 2: Arquitecturas orientadas a servicios

Puntos a tratar

- Razones para introducir SOA
- El concepto de servicio
- Definición de SOA
- Capas en aplicaciones orientadas a servicios
- El gobierno SOA
- SOA y JBI
- SOA y Servicios Web
- SOA y BPM

Razones para introducir SOA

- El software de empresa está condicionado por requerimientos cambiantes que pueden entrar en conflicto



"El software de empresa es un animal diferente" (Dirk Krafzig)

Arquitectura del *Software de empresa*

Los arquitectos software utilizan la refactorización para luchar con el aumento continuo de la complejidad del sistema

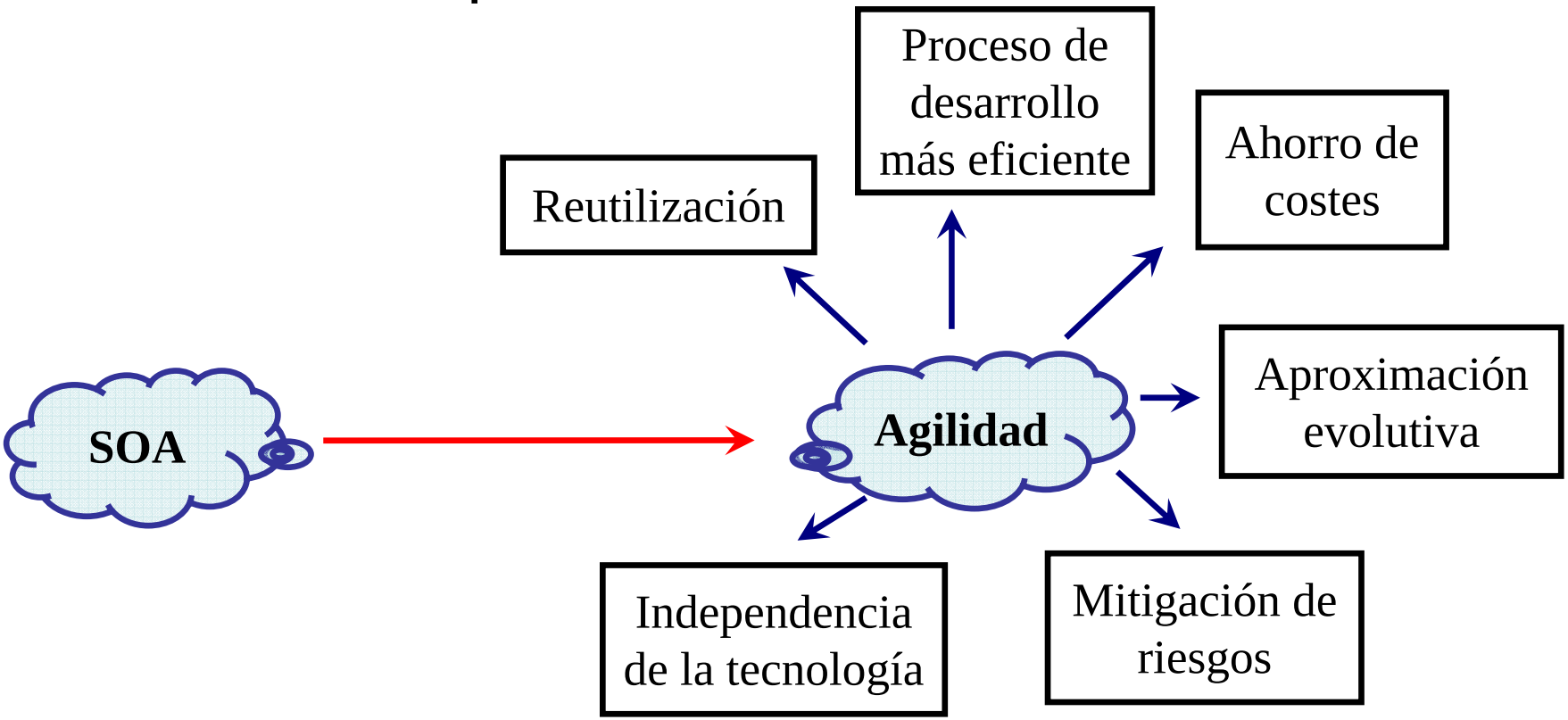
Carácterísticas deseables del sw de empresa

- Simplicidad
- Flexibilidad y mantenibilidad
- Reusabilidad
- Desacoplamiento entre funcionalidad y tecnología
- Objetivo: conseguir una empresa **ÁGIL**

SOA posee estas características

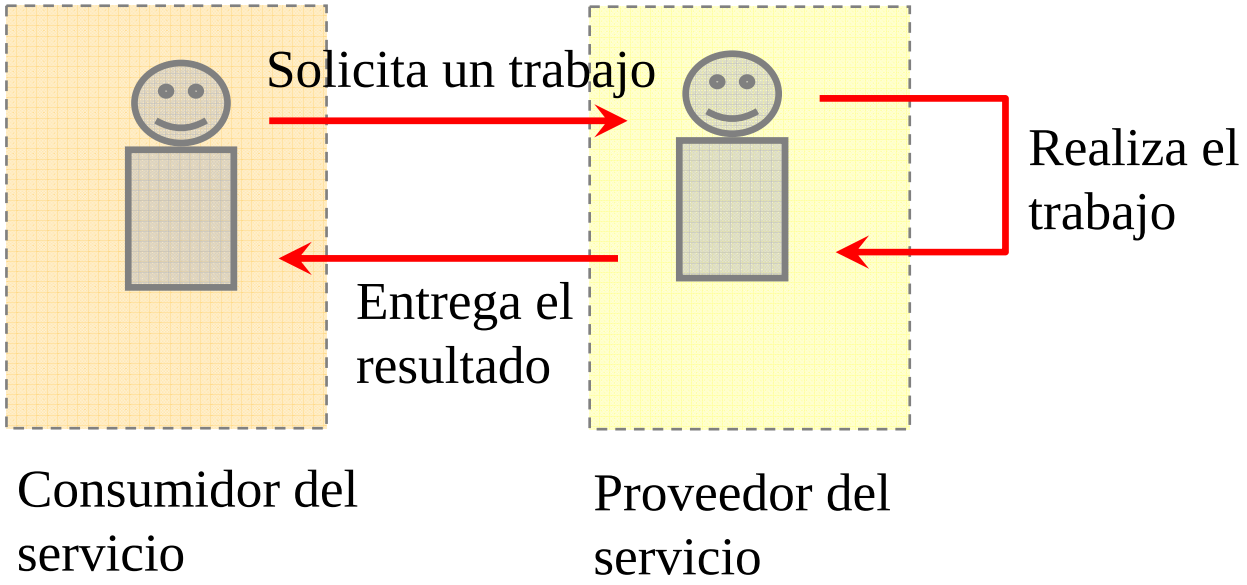
SOA y sus beneficios

- Los procesos y servicios pueden ser rápidamente creados, configurados y reorganizados sin necesidad de personal técnico



El concepto de "servicio"

- Módulo de aplicación autocontenido que es remotamente accesible



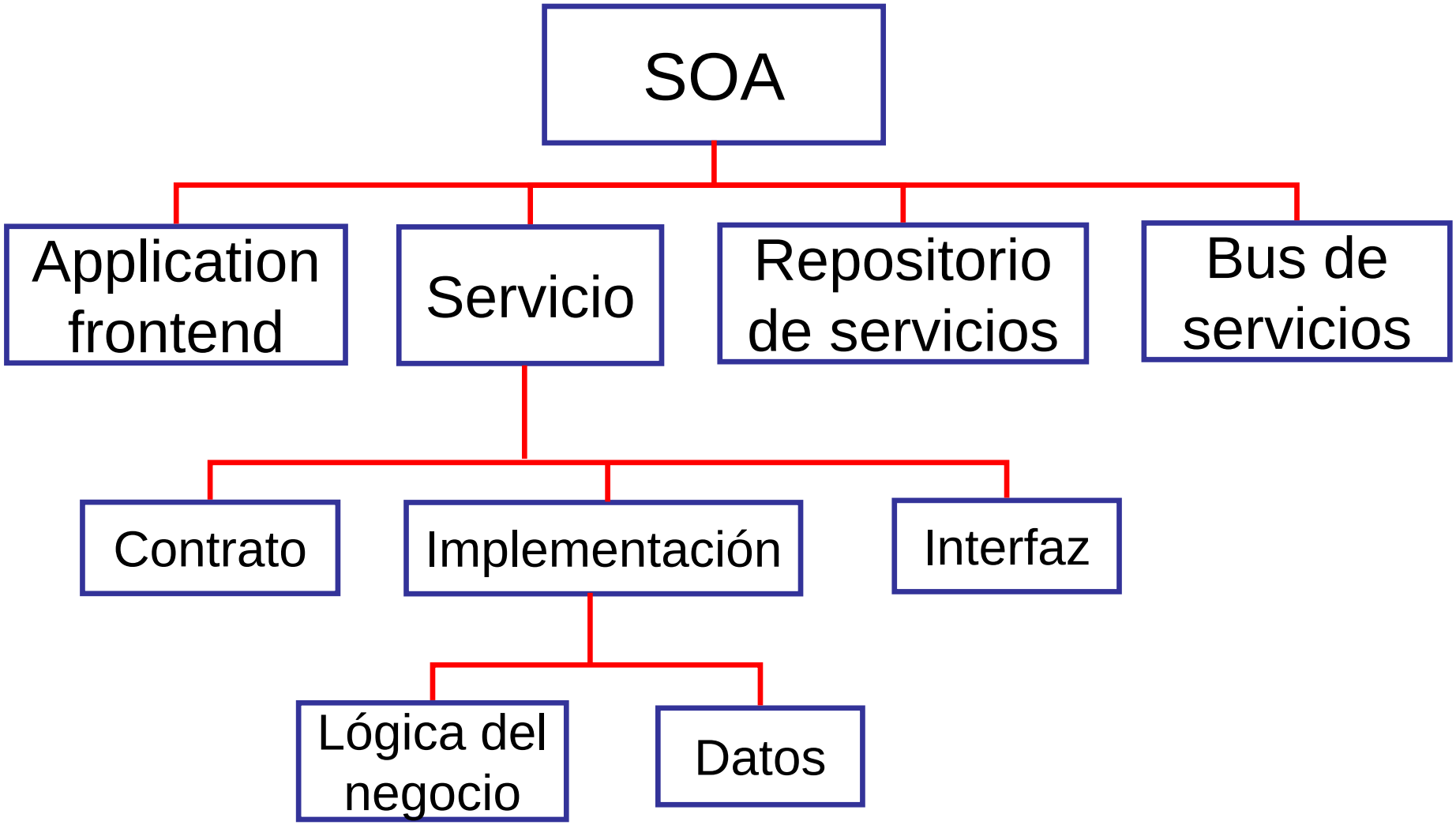
Características de un servicio

- Oculta detalles técnicos como la búsqueda y localización del servicio
- Proporcionan funcionalidad de negocio
- No se diseñan para un cliente específico
- Una SOA proporciona acceso "uniforme" a todos los servicios
- Diferencias significativas con objetos:
  - Interfaz orientada a datos (en vez de a comportamiento)
  - Desacoplamiento de datos y comportamiento
  - Un servicio provoca un cambio de estado

SOA: Definición

- Una arquitectura orientada a servicios (SOA) es una **arquitectura software** basada en los conceptos clave de *frontend* de aplicaciones, servicio, repositorio de servicios, y bus de servicios
- El concepto de una SOA se centra en la definición de una **infraestructura de negocio**

SOA: elementos que la componen

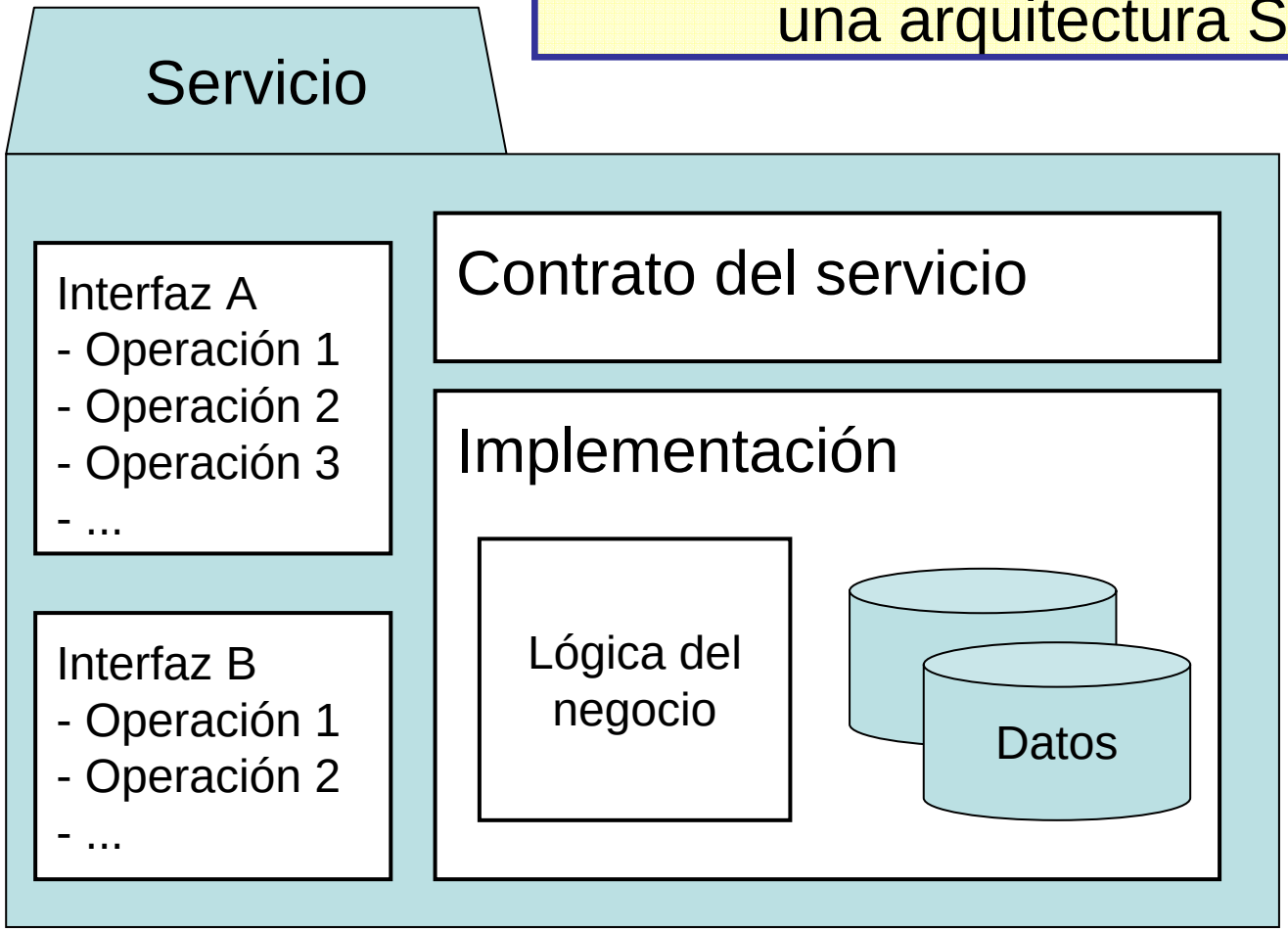


Frontend

- Inician y controlan todas las actividades de los sistemas corporativos
- Los *frontends* de aplicaciones son similares a las capas de nivel más alto en las arquitecturas multi-capa tradicionales

Servicios

Los servicios son el "CORAZÓN" de una arquitectura SOA



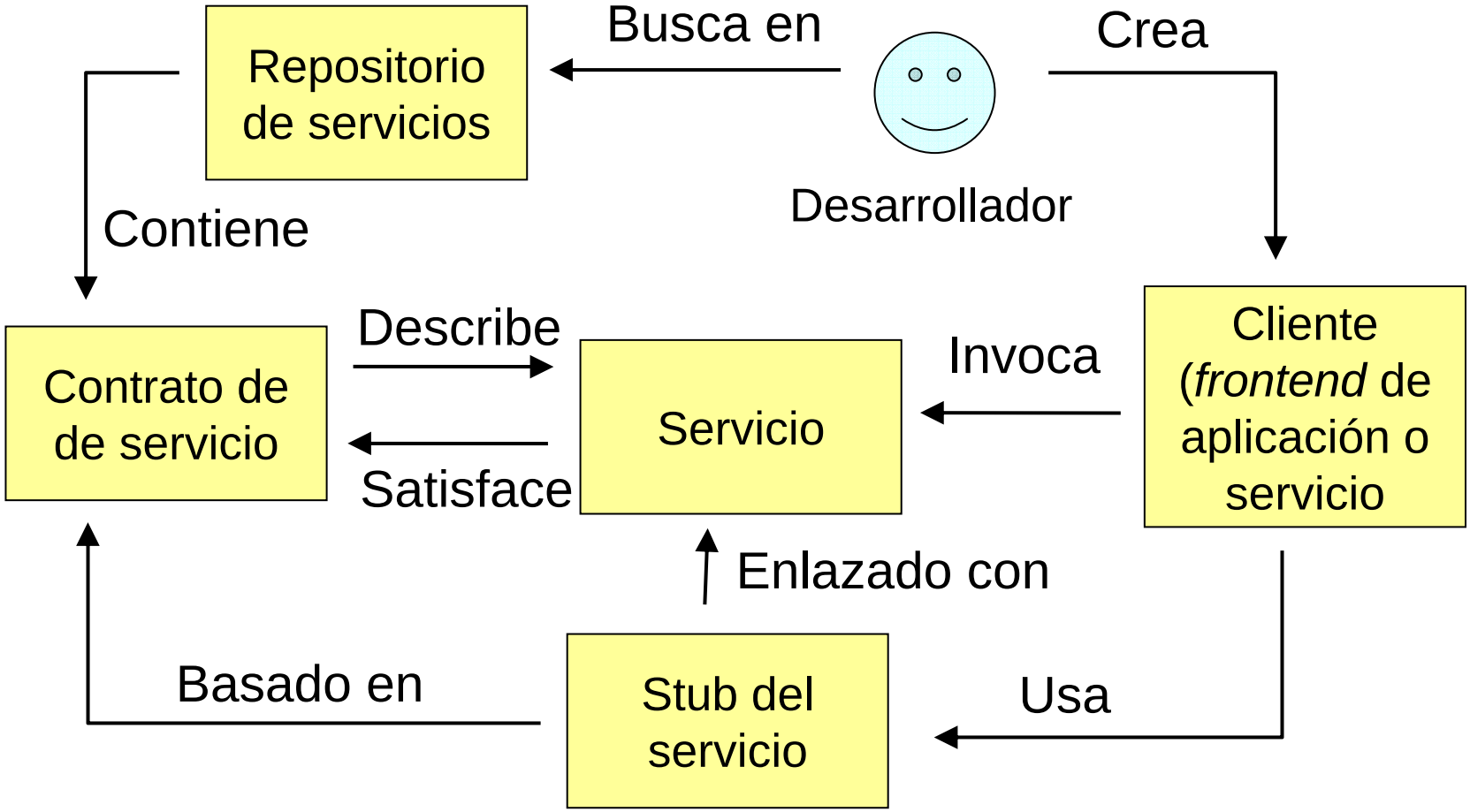
Los servicios deben ser...

- Débilmente acoplados
- De grano grueso (*coarse-grained*)
- Centrados en el negocio
- Reutilizables

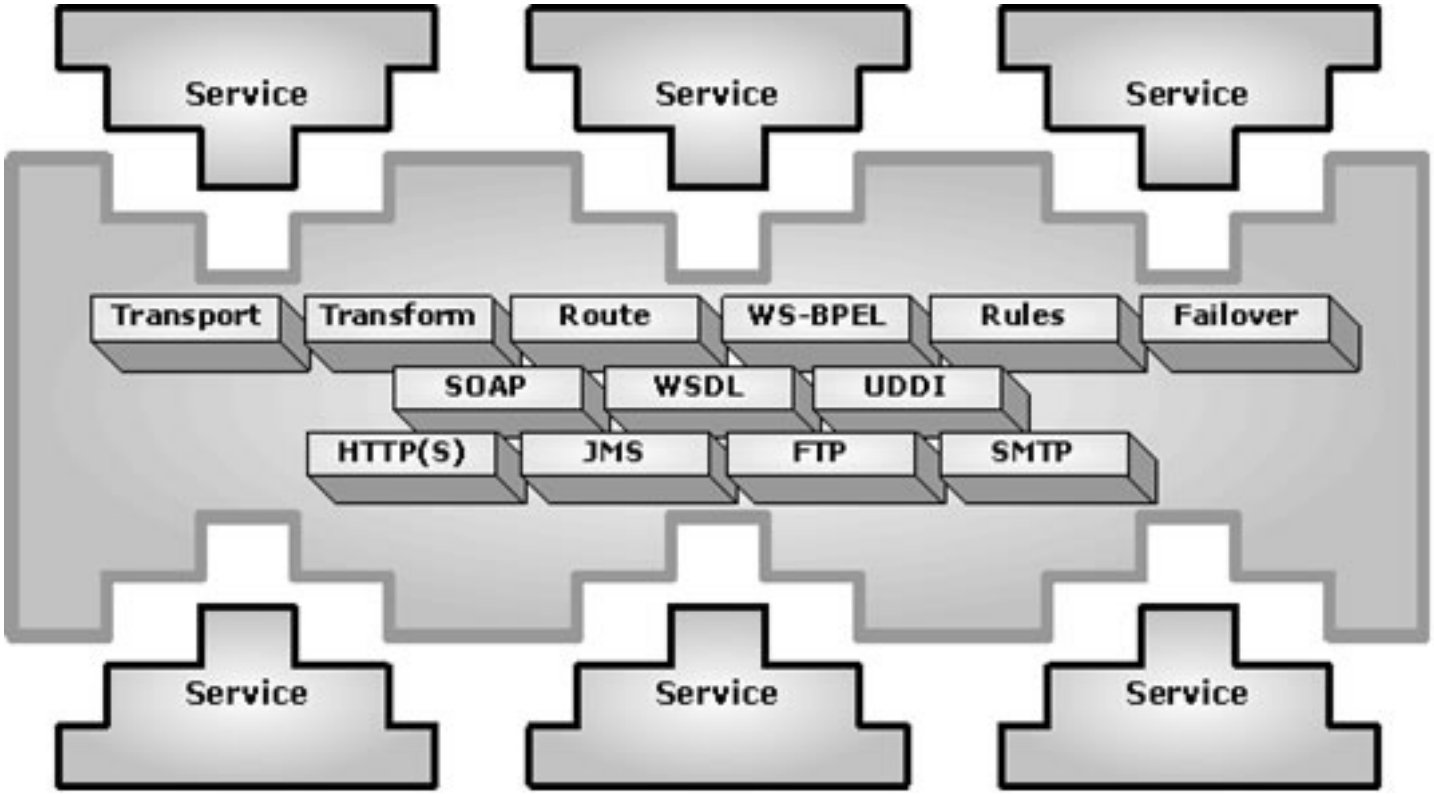
Repositorio de servicios

- Proporciona facilidades para encontrar servicios y adquirir toda la información para utilizar dichos servicios.
- Búsqueda y enlazado (*binding*) de servicios:
  - En tiempo de desarrollo
  - En tiempo de ejecución
- Niveles de búsqueda y enlazado dinámico:
  - Por nombre
  - Por propiedades
  - Basada en *reflection*

Búsqueda y *binding* de servicios en tiempo de desarrollo



Bus de servicios

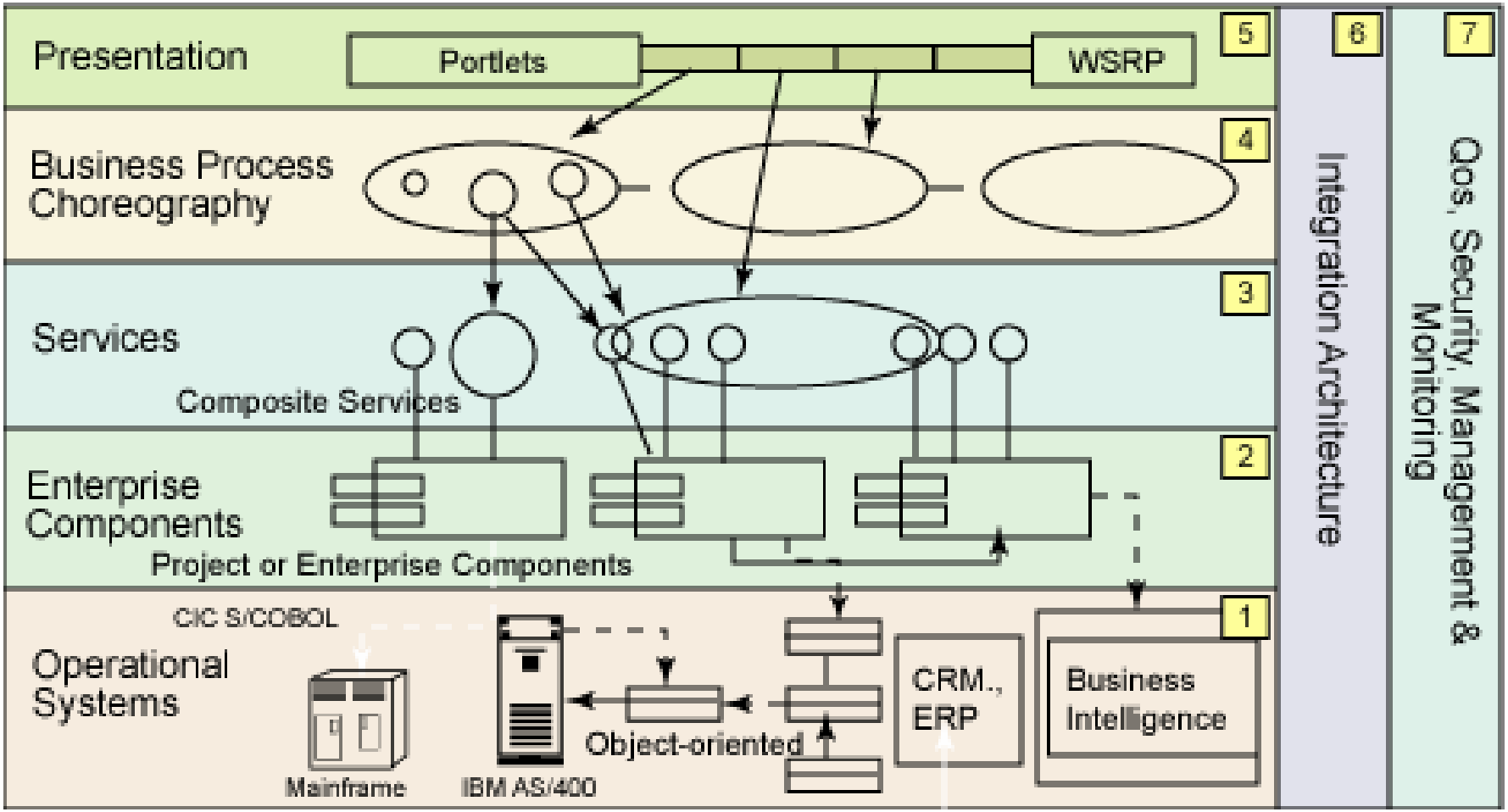


- Proporciona un entorno de ejecución para desplegar los servicios y permitir que se pueda definir la interacción entre dichos servicios (p.ej. orquestación)

Características de un bus de servicios

- Conectividad
- Heterogeneidad de tecnología
- Heterogeneidad de conceptos de comunicación
- Servicios técnicos

Capas en aplicaciones orientadas a servicios



Capa de servicios

- El mayor reto cuando construimos una aplicación orientada a servicios es crear una interfaz con el nivel adecuado de abstracción
- Tenemos dos posibilidades a la hora de construir un servicio:
  - aproximación *top-down*
  - aproximación *bottom-up*.

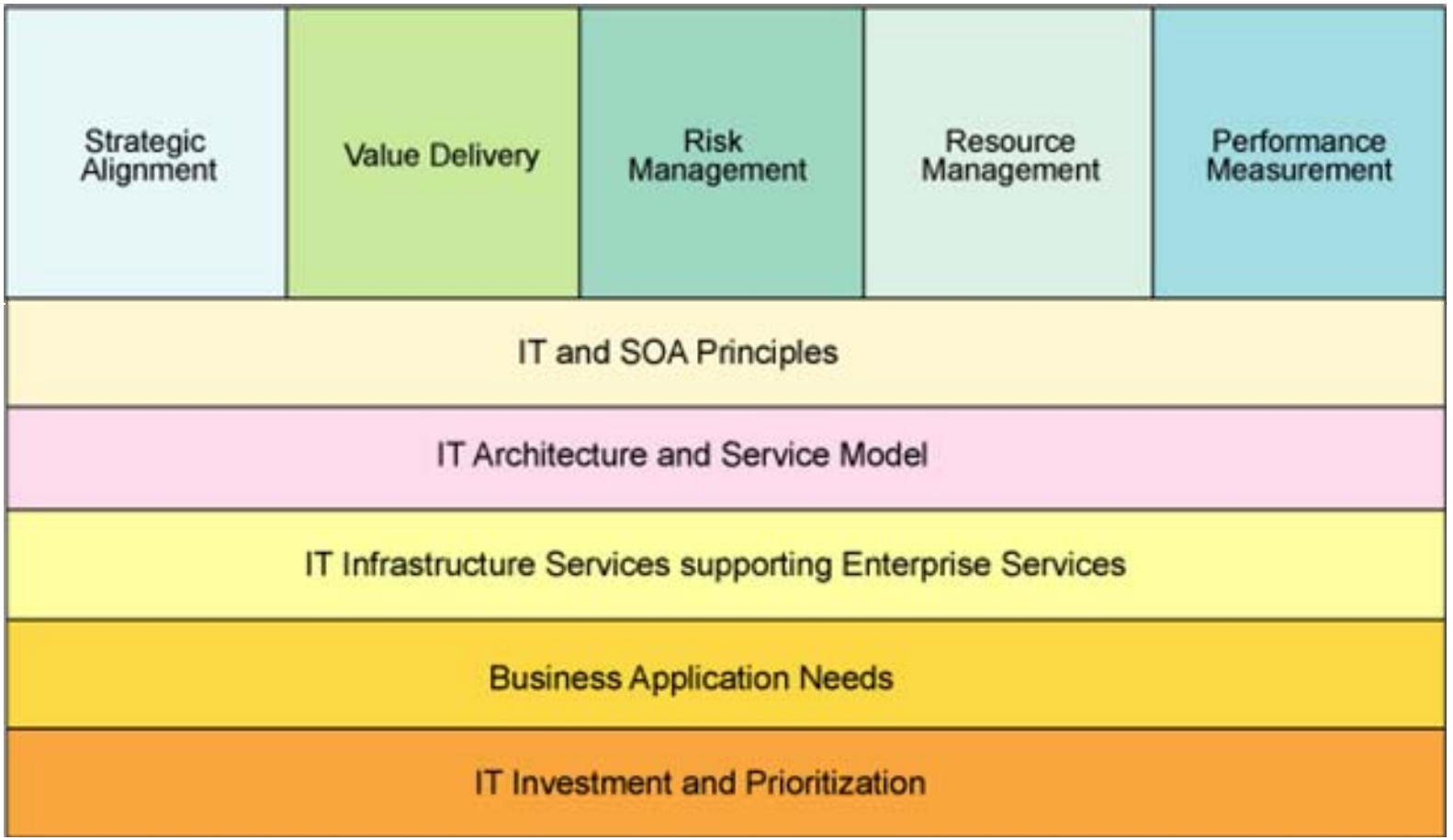
Capa de lógica de negocio

- El principal beneficio que proporciona SOA es la estandarización del modelado de procesos de negocio (orquestación de servicios)
  - BPEL: Estándar de OASIS
  - BPEL es un lenguaje de programación, pero su representación es XML

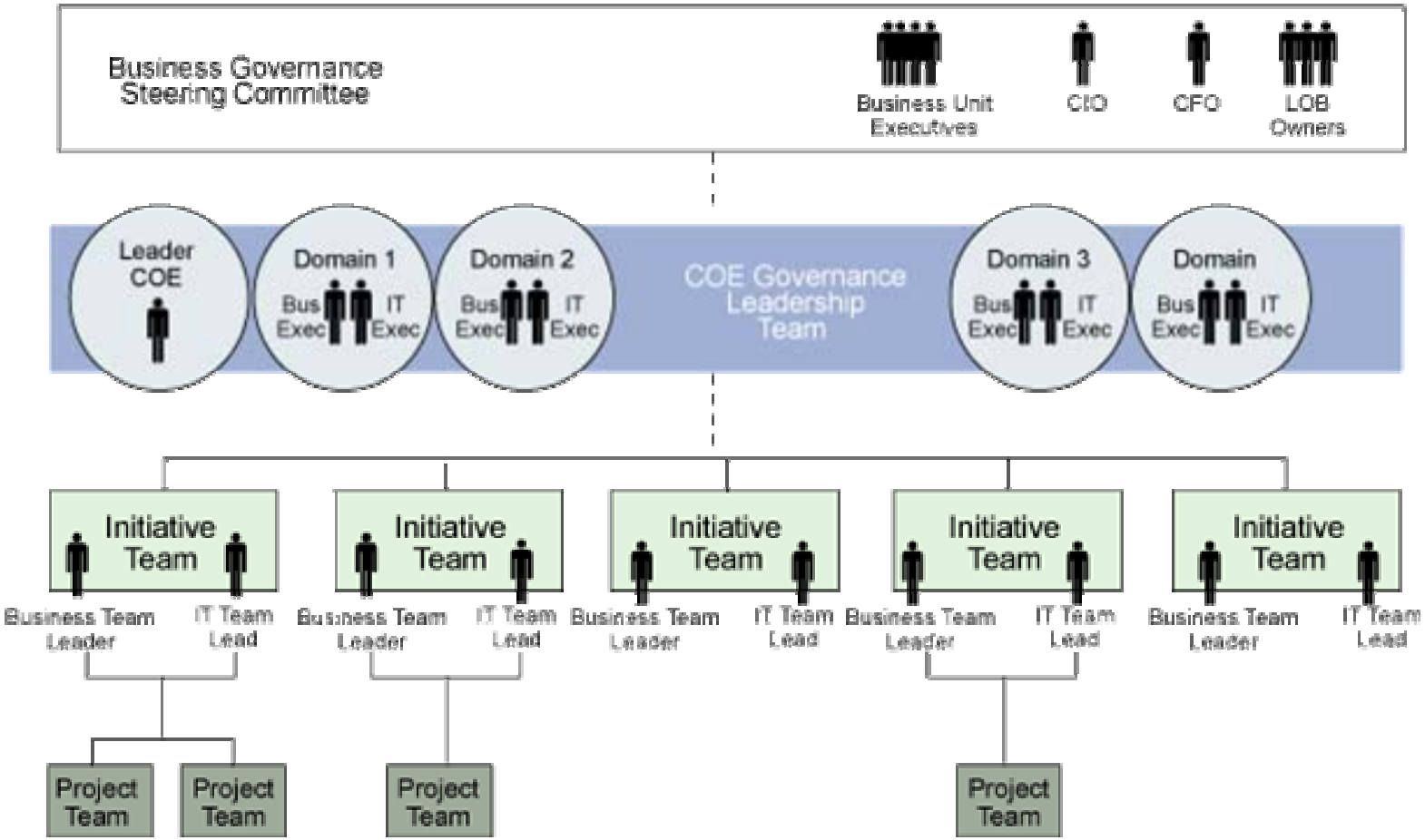
El gobierno SOA (SOA *governance*)

- *Governance*: acción o forma de gobernar
- Marco de toma de decisiones y responsabilidades que fomenta un comportamiento deseable en las IT
- Un equipo de gobierno de IT debe tratar tres cuestiones:
  - ¿Qué decisiones debemos tomar para asegurar una gestión y uso efectivos de la IT?
  - ¿Quién debe tomar dichas decisiones?
  - ¿Cómo se van a tomar dichas decisiones y cómo pueden monitorizarse?

Responsabilidades de gobierno



Implementación del gobierno



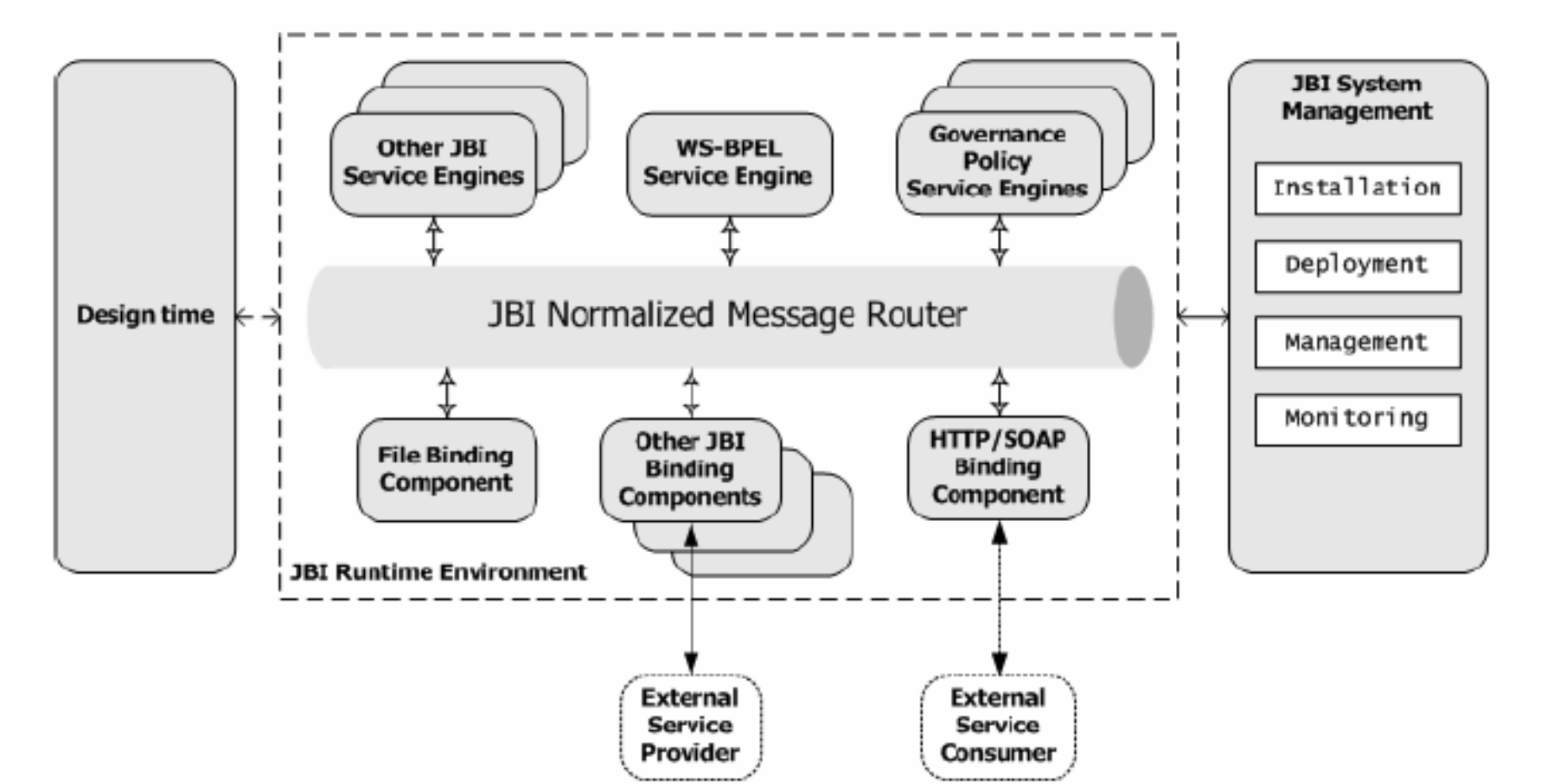
JBI

- JBI (*Java Business Integration*) es un estándar basado en Java que aborda las cuestiones principales sobre EAI y B2B, y que está basado en los paradigmas y principios que defiende SOA.
- Actualmente JSR 208 (JSR: Java Specification Request)
- JBI define una arquitectura basada en *plug-ins* en la que los servicios pueden ser *plugged* en el entorno de ejecución de JBI.

JSR 208: Ventajas

- Es en sí misma una arquitectura orientada a servicios
- Las máquinas de servicios podrían implementarse en cualquier lenguaje siempre y cuando soporten la definición SPI
- Pueden añadirse nuevas máquinas en el contenedor definiendo los mensajes que utilizarán para interactuar con el resto del sistema.
- Interfaces abiertas

Elementos clave de un entorno JBI



Modelo de componentes JBI

- Componentes de la máquina de servicios (SE: *Service Engine*)
  - responsables de la implementación de la lógica del negocio y otros servicios
- Componentes de enlazado (*BC: Binding components*)
  - se utilizan principalmente para proporcionar enlaces a nivel de transporte para los servicios desplegados

Modelo de mensajes JBI

- JBI utiliza un modelo de mensajes que desacopla los consumidores de servicios de los proveedores de servicios.
- El modelo de mensajes se define utilizando WSDL
- Se requiere que los servicios tengan interfaces, formadas por un conjunto de operaciones. Cada operación está formada por uno o más mensajes. Un interfaz puede tener uno o más *bindings* a nivel de transporte.

Elementos clave JBI

- Una forma flexible y abierta de ensamblar las máquinas de ejecución y las comunicaciones que consiguen una solución de integración SOA
- "*Service Assembly*" - Permite definir en un único documento todos los artefactos y servicios que forman una aplicación SOA

SOA y JBI

- SOA
  - Aproximación por capas
  - Elementos integrados y Compartidos: Servicios, Procesos,...
  - Basado en estándares: BPEL, JBI, WSDL,...
- JBI (and JBI-based ESB)
  - Estandariza los componentes de integración "*pluggables*"
  - Estandariza a administración de los servicios compuestos
  - Estandariza el intercambio de mensajes
  - Proporciona un meta-contenedor SOA con bajo acoplamiento

SOA y Servicios Web (I)

- SOA y Servicios Web **NO** son sinónimos
- SOA es un principio de diseño, mientras que los servicios Web son una tecnología de implementación
- Una de las principales ventajas de implementar una SOA con servicios Web es que los servicios Web están muy extendidos y constituyen una plataforma sencilla y sobre todo neutral

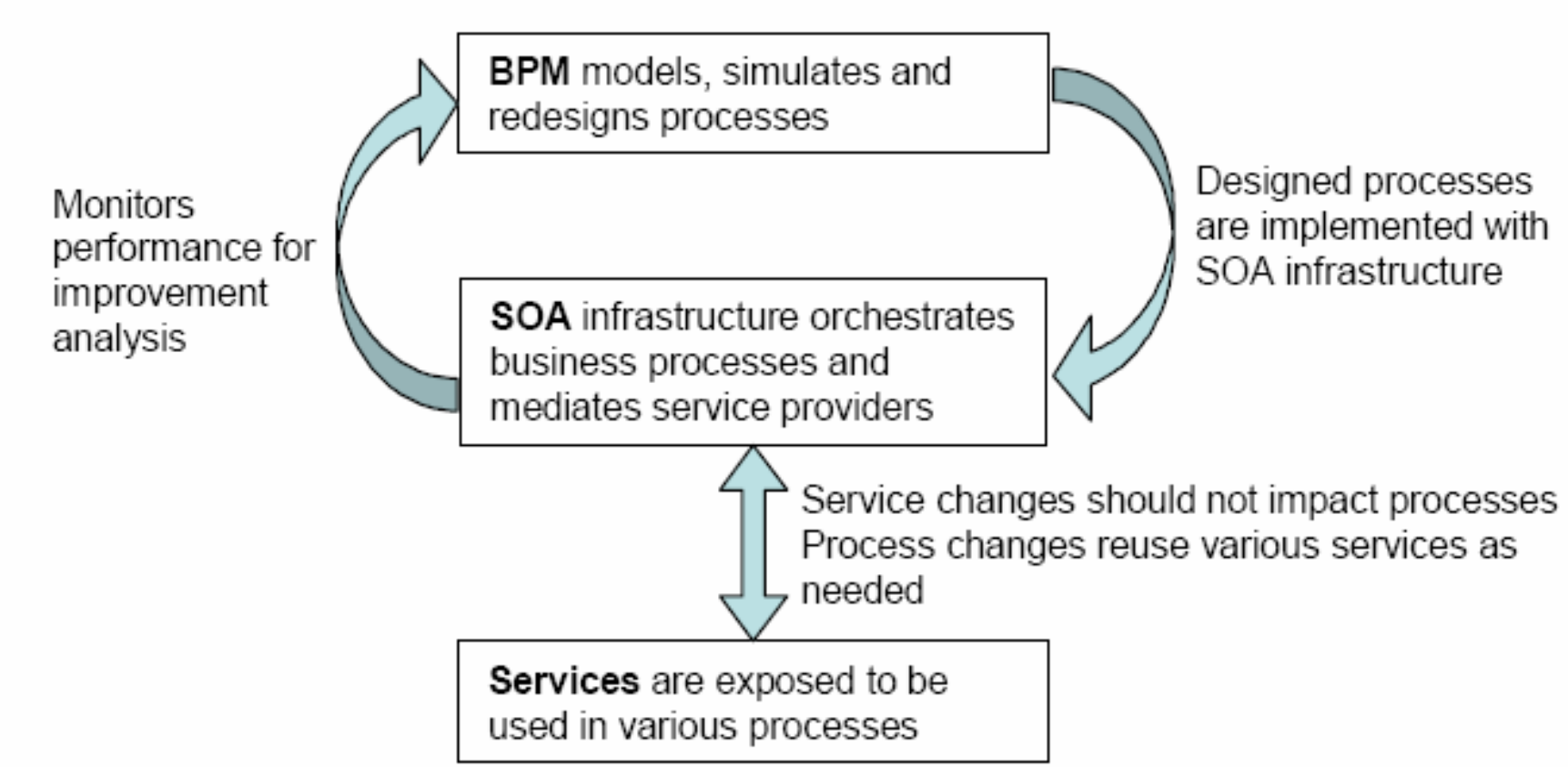
SOA y Servicios Web (II)

- La combinación de servicios Web y SOA proporciona una integración rápida
- Las aplicaciones pueden intercambiar datos más fácilmente utilizando un servicio Web definido en la capa de lógica de negocio
- El desarrollo de puntos de entrada orientados a servicios en la capa de lógica de negocio permiten a una máquina de gestión de procesos de negocio llevar a cabo un flujo automático de ejecución a través de los múltiples servicios

BPM

- BMP: *Business Process Management*
- La BMP se encarga de identificar, modelar, desarrollar, desplegar y gestionar sus procesos de negocio
- Ventajas de las BMP:
  - Reducen diferencias entre requerimientos del negocio y de las IT
  - Incrementan la productividad de los empleados
  - Incrementan la flexibilidad y agilidad de la empresa
  - Reduce los costes de desarrollo

Relación entre BMP y SOA



¿Preguntas...?