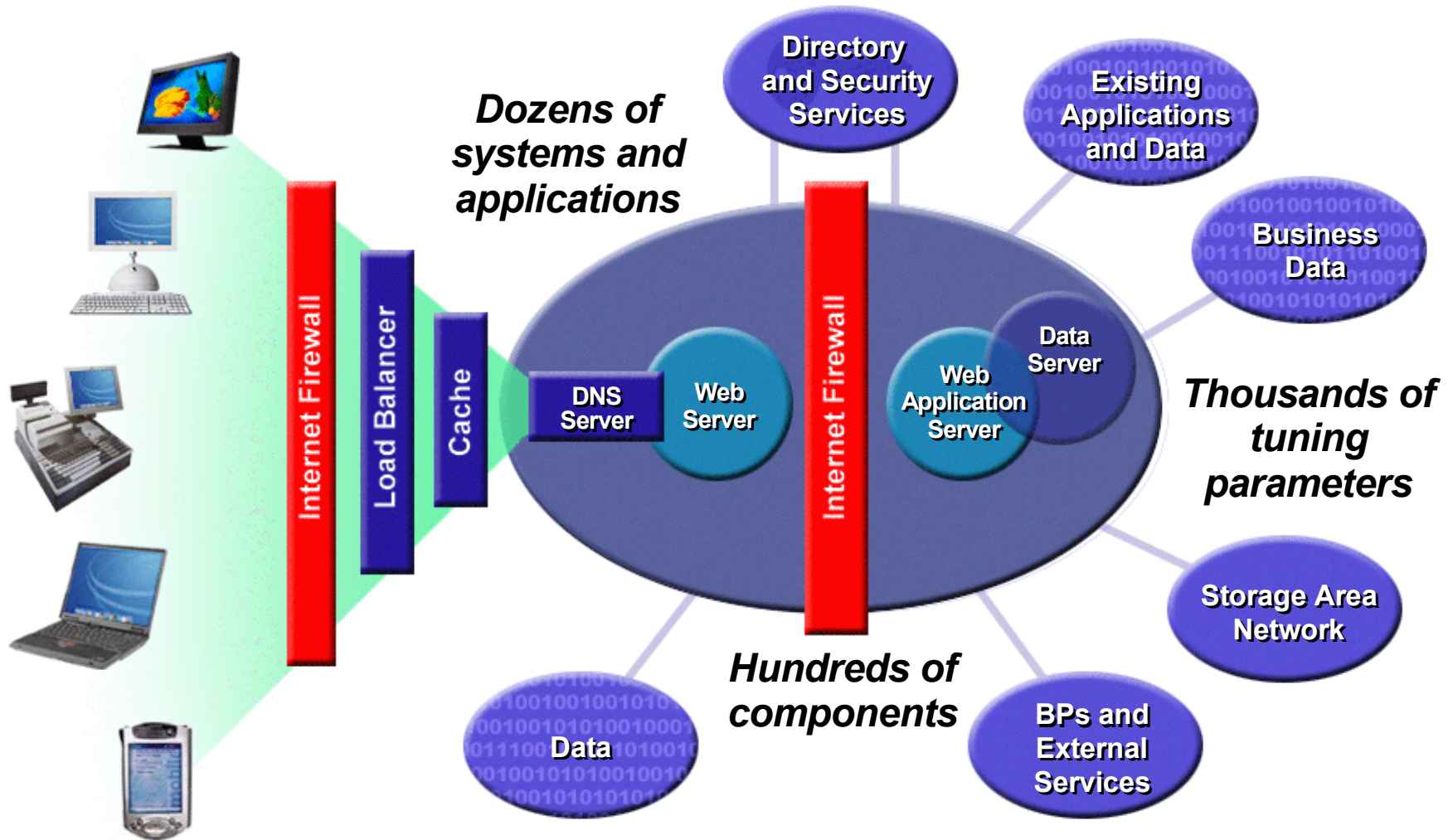


AI

**Sistemas Distribuidos y AI:
Computación Autonomica y SOA**

infraestructuras heterogéneas complejas son una realidad!



Tendencias

- Administración de los sistemas es cada vez más difícil
- Sistemas heterogéneos están cada vez más conectados
- Las arquitecturas no pueden planificar las interacciones entre los componentes
- Más carga debe ser asumida en tiempo de ejecución



Necesitamos sistemas informáticos autogestionados
Comportamiento especificado por los administradores de
sistemas a través de políticas de alto nivel

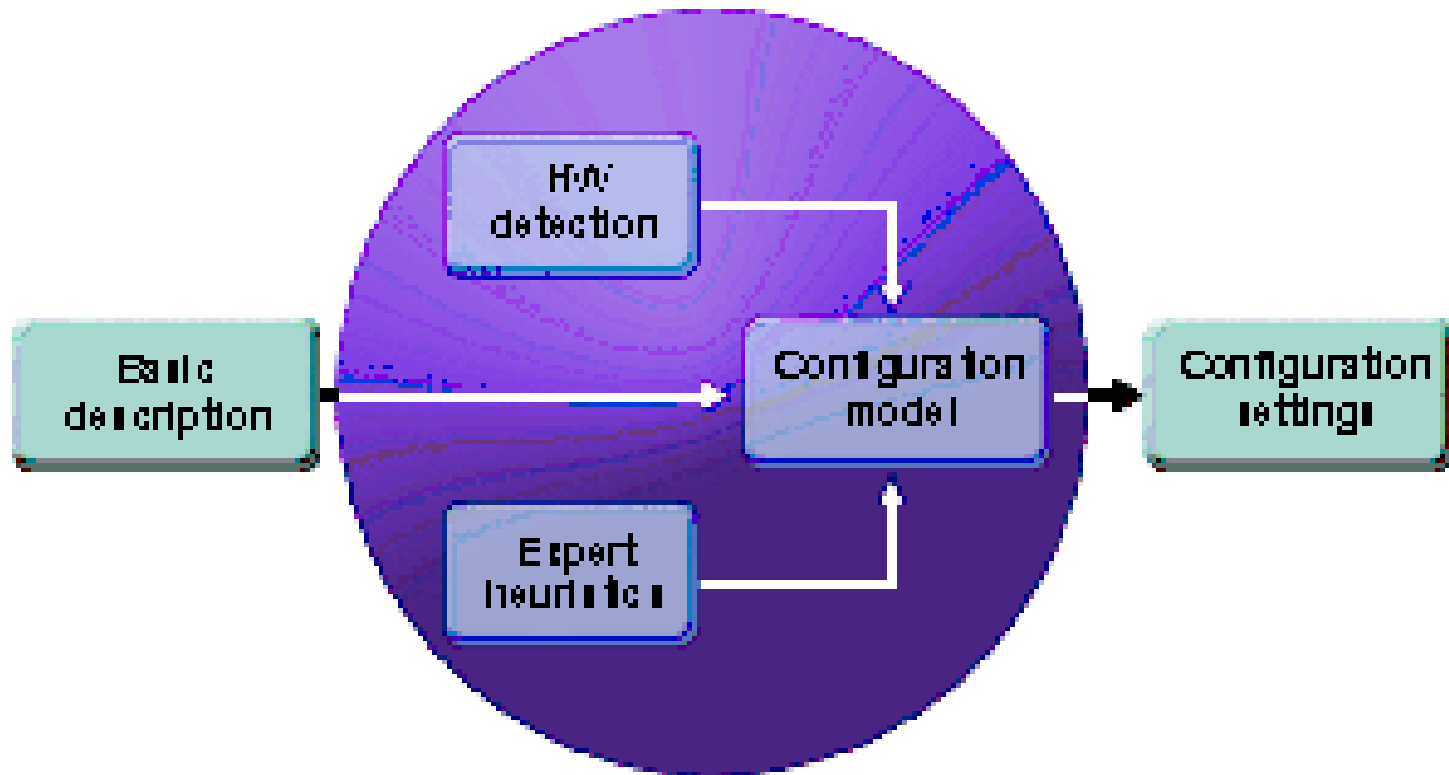
Computación Autonómica

- Modelo de autogestión inspirado en el sistema nervioso autónomo del ser humano
- Sistemas son capaces de ser auto-administrado a alto nivel, manteniendo la complejidad del sistema invisible para el usuario

Ha sido usada para auto-gestionar capacidades como:

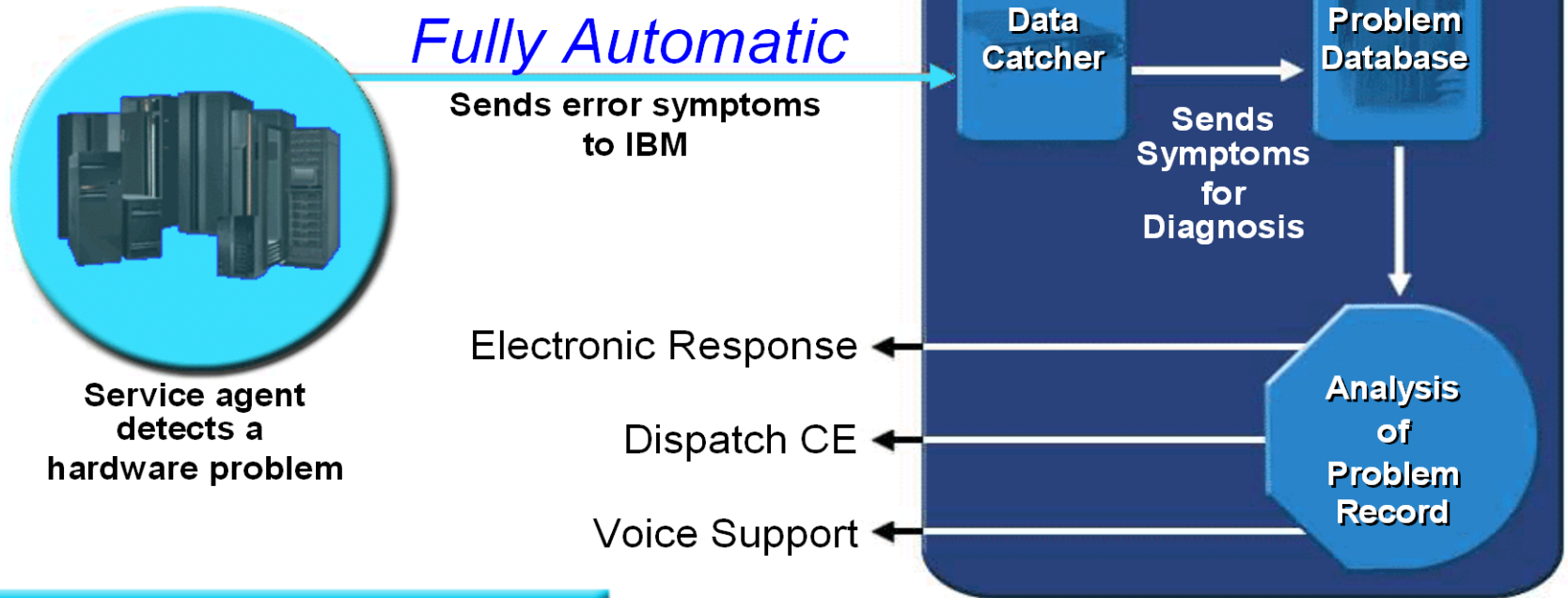
- Auto-configuración
- Auto-sanación
- Auto-Optimización
- Auto-Protección

Ejemplo Auto-Configuración: DB2 Configuration Advisor



Ejemplo auto-cura:

IBM Electronic Service Agent

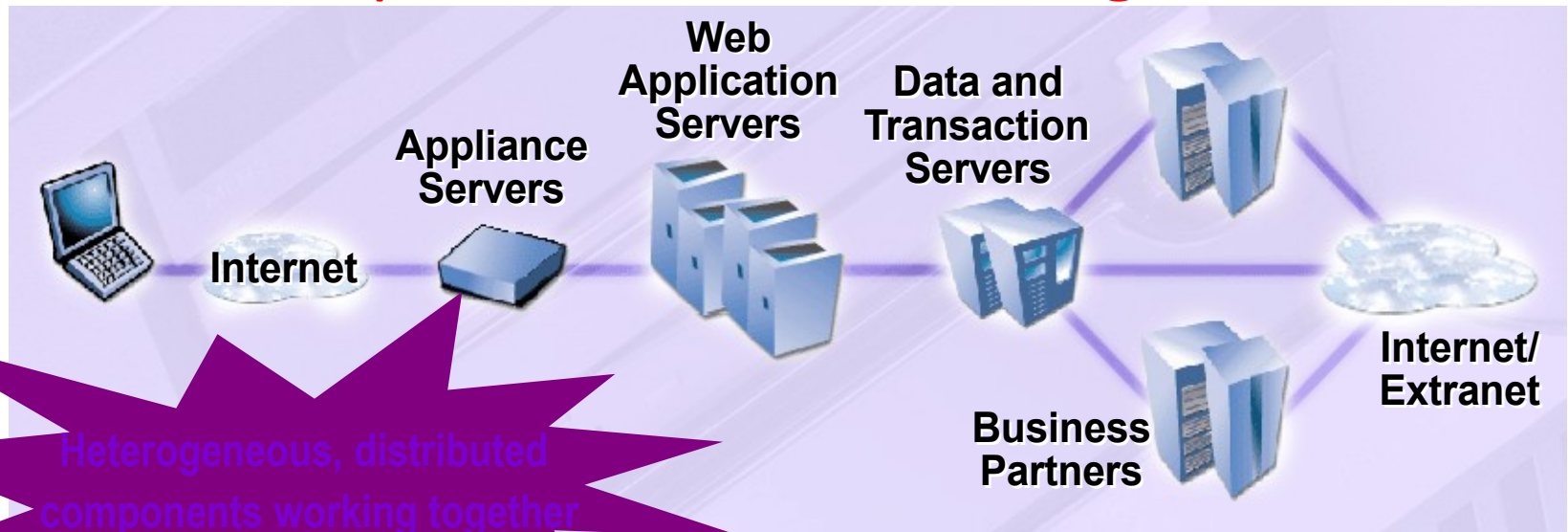


- ***Faster problem resolution***
- ***Higher availability/resiliency***
- ***Lower maintenance cost***

""IBM's eService Agent allows me to sleep soundly knowing the system is being monitored 24x7."

Alex Tambellini, 7-Eleven Stores Pty Ltd.

Ejemplo auto-optimización: Enterprise Workload Management

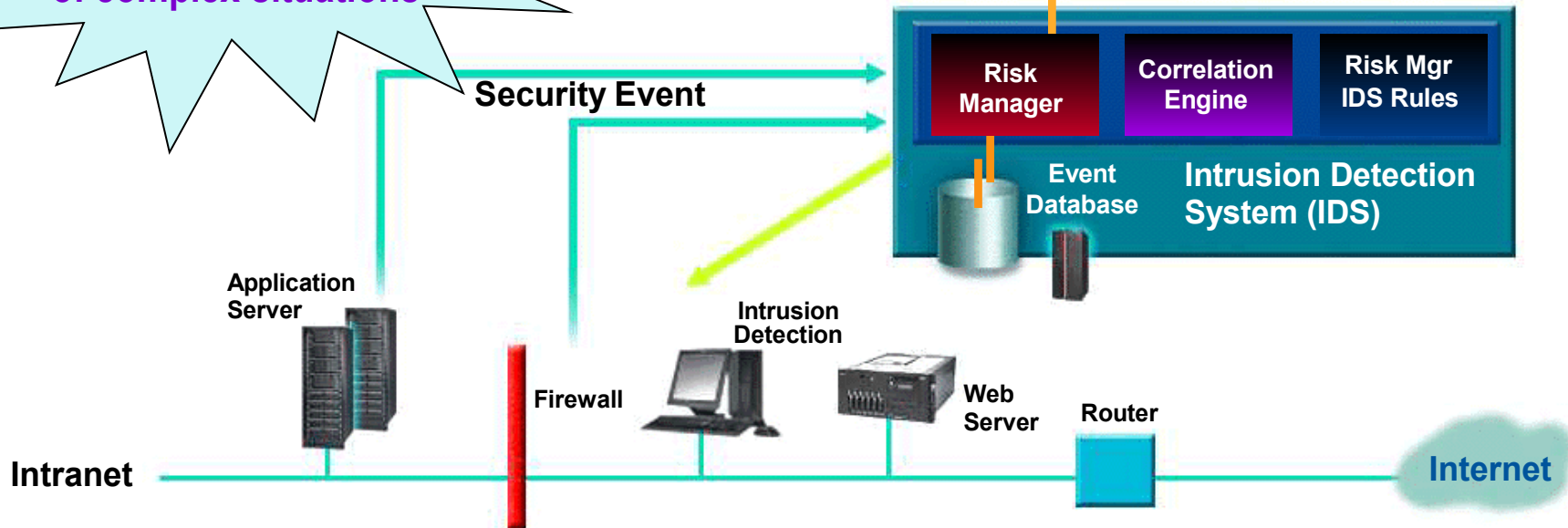


Self-tuning, end-to-end performance management

- Dynamic allocation of network resources
- Workload balancing & routing
- Cross platform reporting
- Policy-based for various classes of users & applications

Ejemplo auto-protección: IBM Tivoli Risk Manager

Rapid / automated analysis
of complex situations



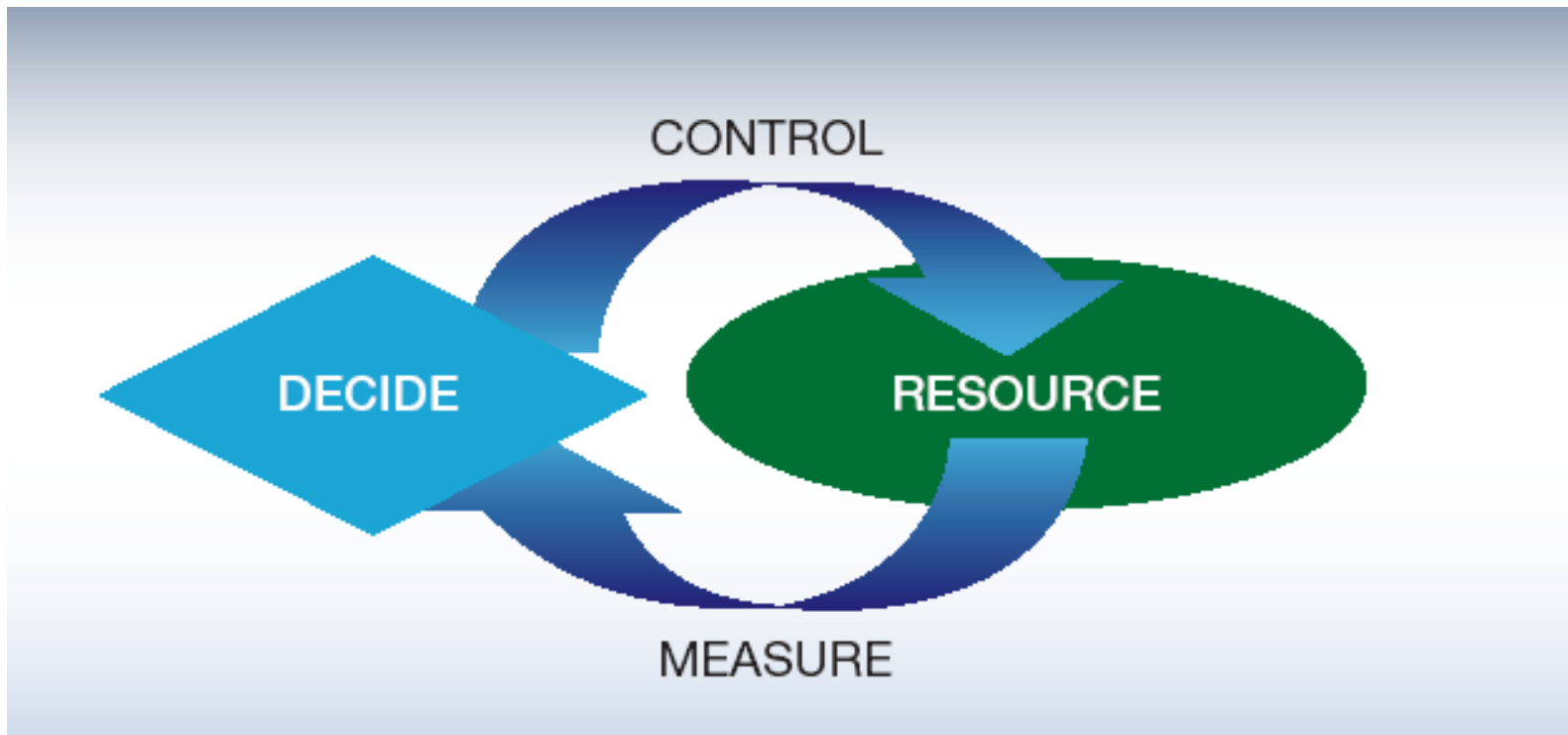
- Automate incident response
- Protect systems and data
- Help prevent service disruptions

"The Tivoli security management software portfolio is helping our clients extend their businesses to the Internet while providing security and privacy..."

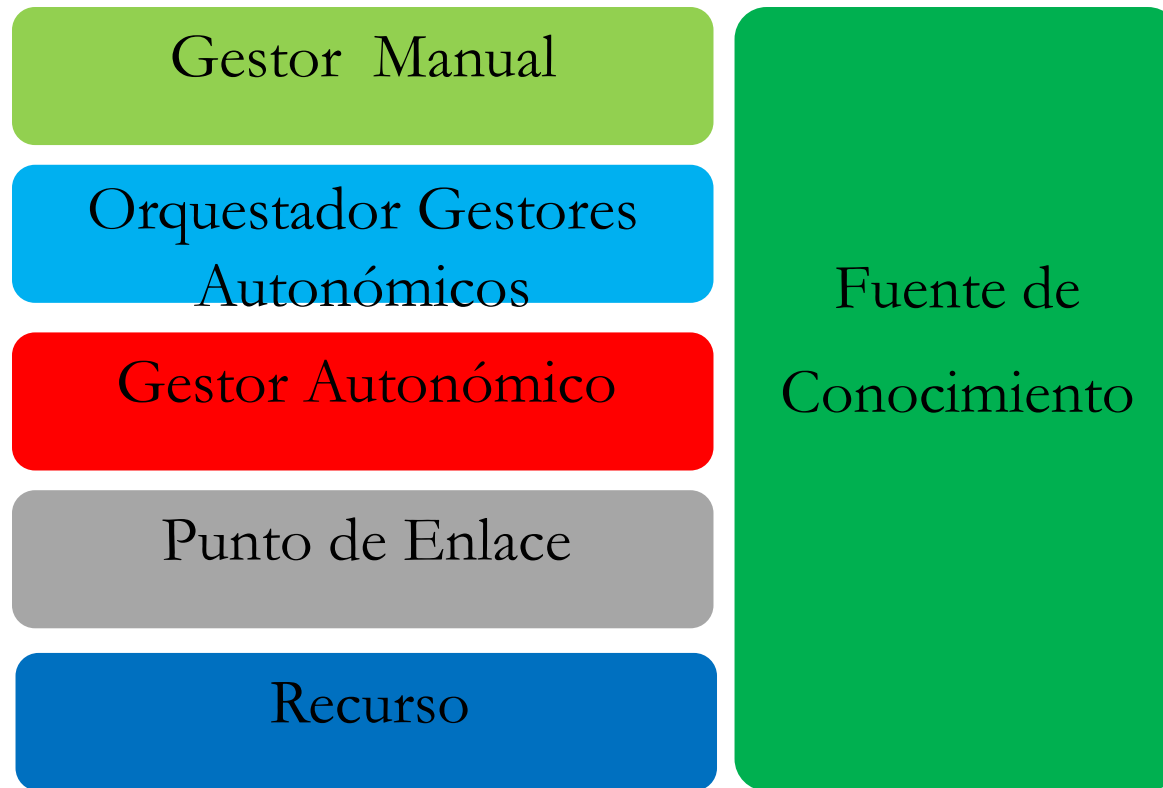
Mark Ford, Principal
Deloitte & Touche

Computación Autonomica

Lazo de control inteligente



Computación Autónoma

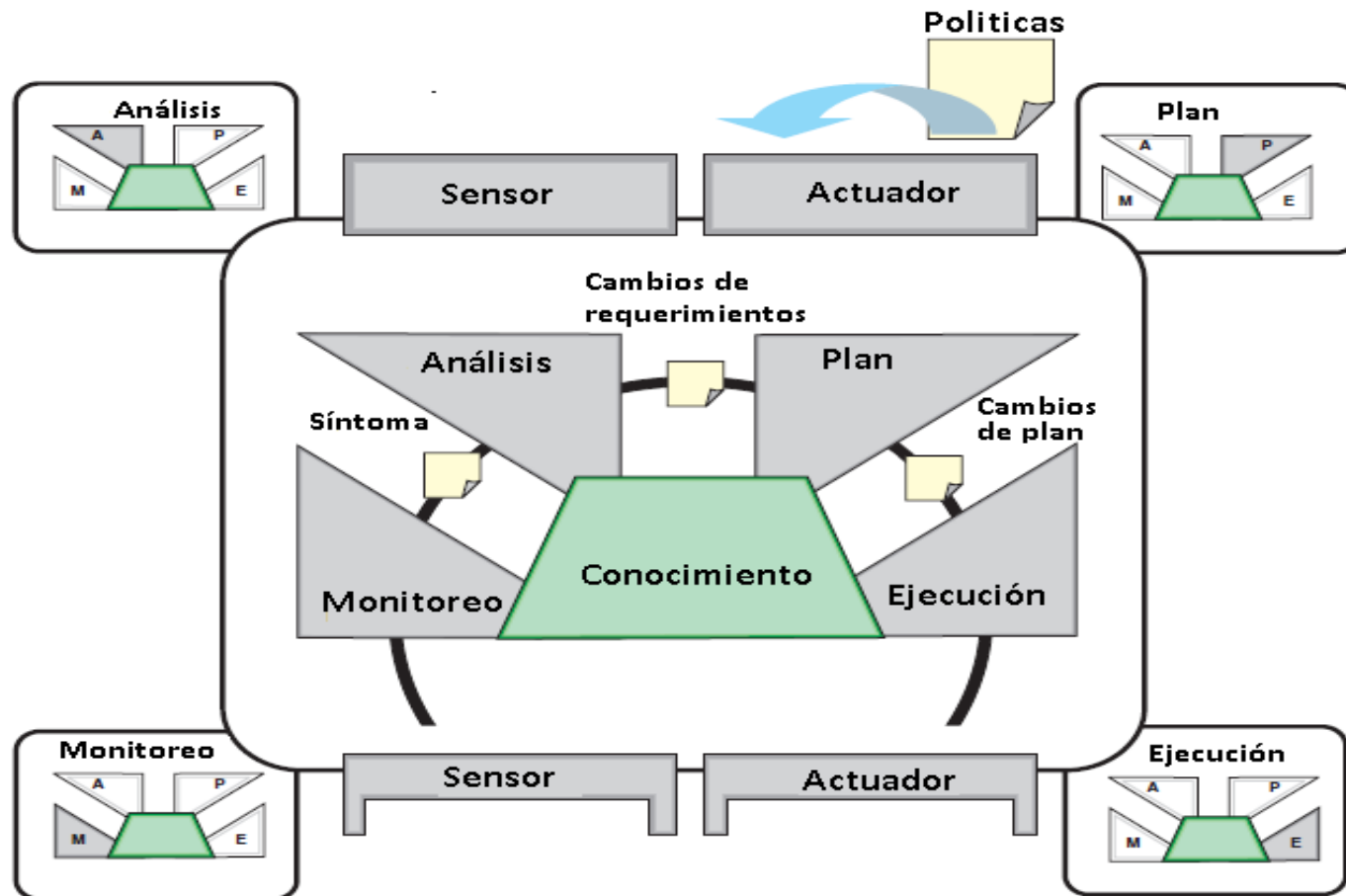


Computación Autonomica

- **Recurso Gestionado:** puede ser cualquier tipo de recursos (hardware o software) que puede ser gestionado. El recurso gestionado se controla a través de sus sensores y actuadores.
- **Puntos de Enlace:** Enlaza a los sensores y/o actuadores requeridos para la gestión de los recursos.
- **Gerente Autonomico:** Implementa los lazos de control inteligentes que automatizan las tareas de autorregulación/autogestión de las aplicaciones
- **Orquestador de los Gerentes Autonomicos:** Debido a que un sistema autonomico puede contar con varios gerentes autonomicos que necesitan trabajar en conjunto para garantizar el funcionamiento correcto de los recursos, este nivel proporciona el canal de comunicación para la coordinación entre ellos
- **Manejador Manual:** Permite a los humanos configurar los gerentes autonomicos para realizar su tarea de autogestión, proveyendo para esto una interfaz hombre-máquina que permite conectar al hombre con el gerente autonomico.
- **Fuentes de Conocimiento:** Proporciona acceso a los conocimientos requeridos para la gestión autonomica del sistema. Los distintos componentes del gerente autonomico

Computación Autónoma

Gestor Autónomo

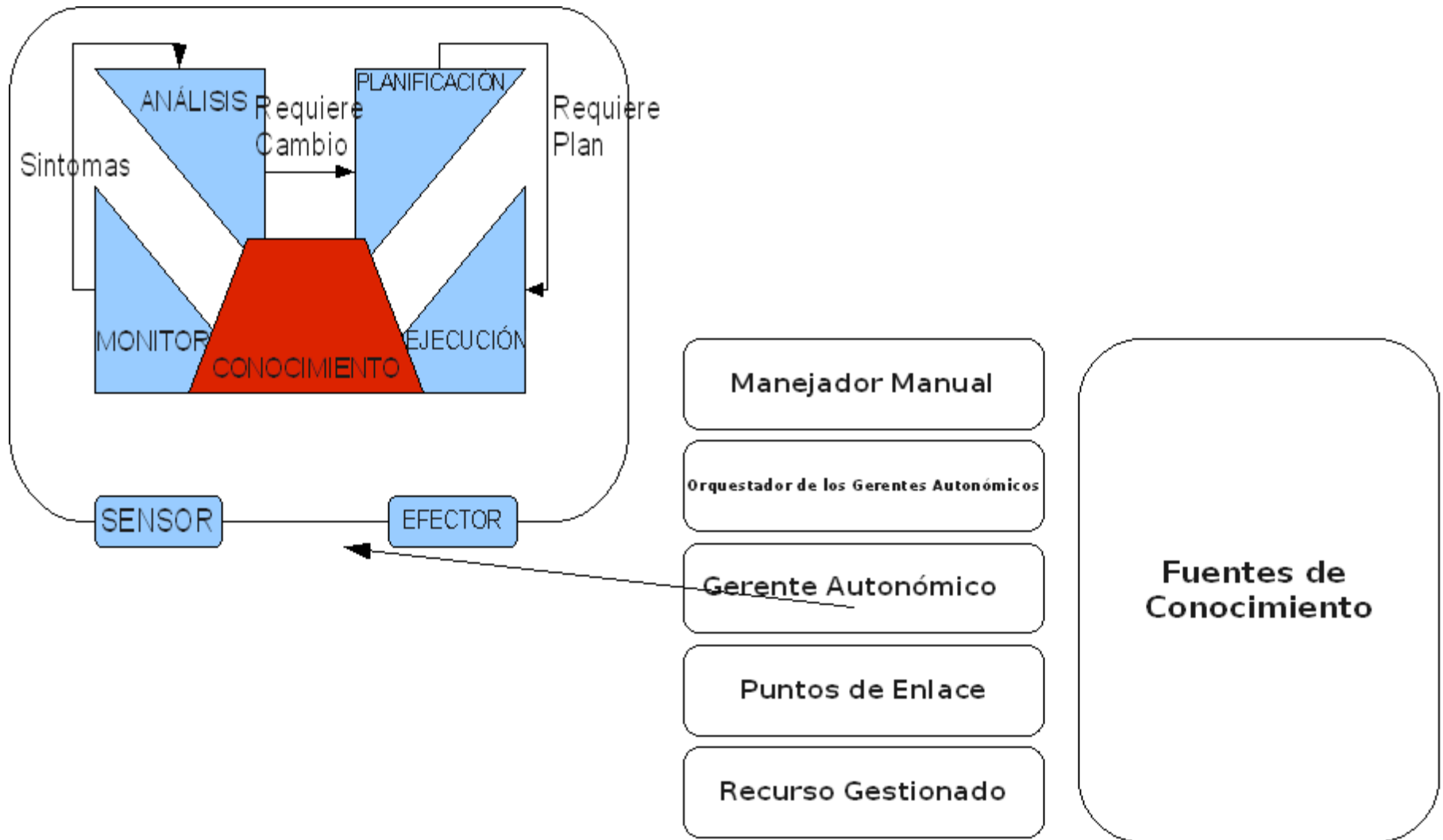


Computación Autónoma

Gestor Autónomo

- **Monitor:** recolecta los eventos/datos de los sensores de lo que esta aconteciendo en los recursos gestionados, convirtiéndolos en un conjunto de patrones o síntomas que pueden ser analizados por el componente de Análisis.
- **Análisis:** Provee mecanismos para observar y analizar los patrones provistos por el monitor, para determinar las causas de los problemas en los recursos gestionados..
- **Planificación:** Genera el conjunto de operaciones a ejecutar sobre el recurso gerenciado, a fin de alcanzar las metas y objetivos propuestos;.
- **Ejecución:** Ejecuta el conjunto de acciones necesarias contenidas en el Plan. Provee una interfaz para la comunicación entre el el gestor autónomo y el recurso gerenciado (interactúa con los actuadores).

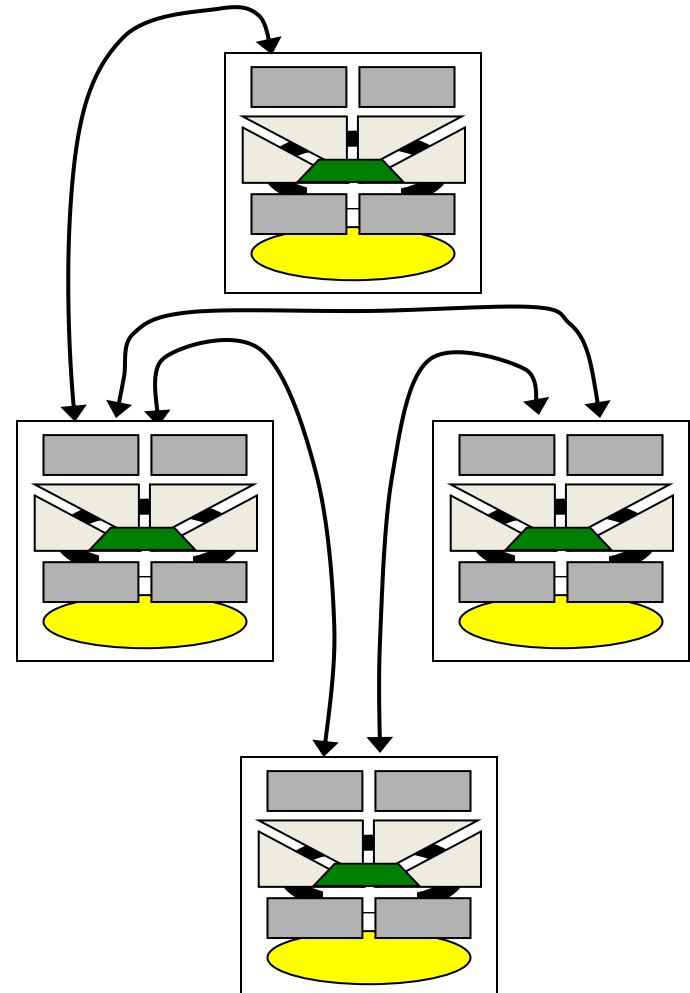
Computación Autónoma



Interacción Elementos Autonómicos

Relaciones

- Dinámicas, efímeras
- Formado por acuerdo
 - Puede ser negociado
- Por espectro
 - Peer-to-peer
 - Jerárquica
- Sujeta a políticas



Arquitectura Orientada a Servicios (SOA)

SOA consiste en un patrón de desarrollo para implementar aplicaciones distribuidas flexibles, que exhiben bajo acoplamiento entre sus componentes de software

SOA

Representa un modelo en la cual la lógica de la aplicación es descompuesta en pequeñas y distintas unidades lógicas, que al agruparse, comprenden la lógica de la aplicación como un todo.

- De manera individual, estas unidades son independientes y pueden ser distribuidas.
- Así, SOA es un enfoque para crear sistemas por la composición de servicios autónomos, independientes de las tecnologías sobre las que descansan.
- SOA es utilizado para organizar y utilizar recursos distribuidos, permitiendo interacciones entre proveedores y consumidores de servicios heterogéneos, a través de distintas plataformas, usando diferentes lenguajes y paradigmas de programación, e incluso, empleando diferentes middlewares.

SOA desde el punto de vista del negocio

resolver los siguientes requerimientos

- Mejorar la flexibilidad y agilidad de los sistemas.
- Proporcionar una visión integrada de los distintos “silos” de la organización.
- Mejorar la cobertura de las necesidades de negocio.
- Reducir el impacto de la evolución de la tecnología en las aplicaciones de negocio

SOA desde el punto de vista de la tecnología

Favorece la reutilización y la reducción del “time to market”:

- Aumenta el grado de reutilización al desacoplar las capas de una aplicación.
- Permite reutilizar las aplicaciones existentes mediante la encapsulación en servicios.
- Permite la utilización de servicios de terceros.
- Permite reaprovechar las plataformas existentes.
- Aumenta la flexibilidad:
- Simplifica la adaptación de los sistemas existentes.
- Evita el desarrollo de interfaces punto a punto entre los sistemas.
- Aumenta la interoperabilidad entre sistemas, permitiendo tanto la externalización como la prestación de servicios.

SOA Mejora la productividad de los procesos:

- Aumenta el nivel de automatización de los procesos, reduciendo el número de actividades manuales.
- Permite monitorizar la actividad del negocio (cuadros de mando).
- Permite realizar un análisis estadístico de los flujos de negocio reales en base a indicadores clave de negocio, permitiendo la identificación de puntos de mejora a optimizar.
- Permite evaluar el impacto y beneficio de variantes en los procesos mediante simulación.

SOA Mejora el proceso de construcción de software:

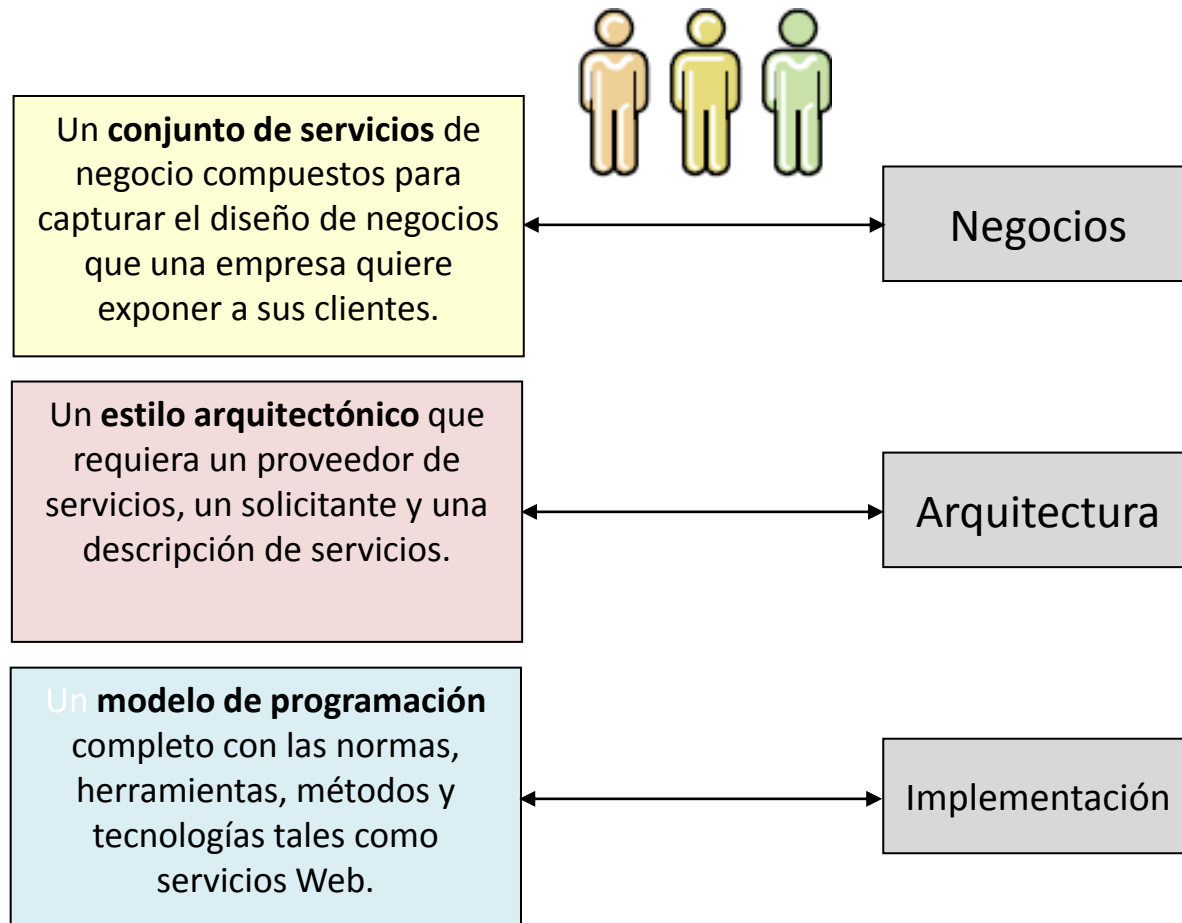
- Favorece la industrialización.
- Mejora la especificación de los requerimientos de negocio.
- Proporciona una filosofía de desarrollo común a todos los negocios y canales.
- Mejora la calidad.
- Desacopla el desarrollo de servicios y de procesos.
- Mejora el mantenimiento (procesos autodocumentados).

SOA Mejora la usabilidad de las aplicaciones:

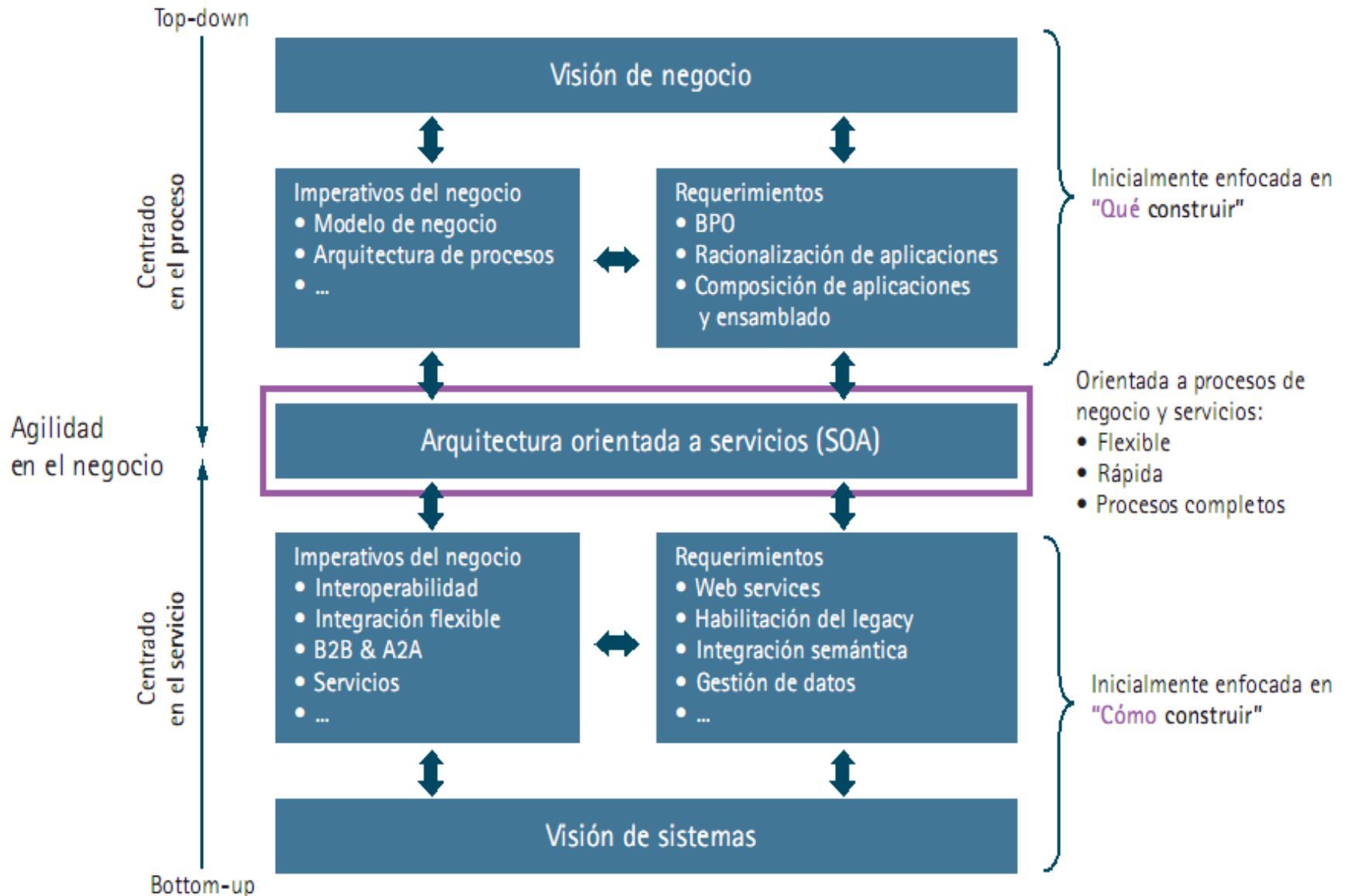
- Permite presentar al usuario la información dispersa en distintos sistemas y de forma integrada.
- Permite alcanzar un mayor nivel de automatismo en las aplicaciones en procesos complejos de workflow.
- Permite utilizar tecnologías de presentación avanzadas como Web 2.0.

Arquitectura Orientado a Servicios

Perspectivas



Vista organizacional



Arquitectura Orientada a Servicios (SOA)

SOA está basada en los siguientes conceptos:

- **Servicios:**
 - *Contrato del Servicio*: Especificación formal o informal del servicio, incluyendo todos los propósitos, funcionalidades, restricciones y uso del servicio.
 - *Interfaz del Servicio*: Especifica cómo se puede acceder al servicio, incluyendo el formato de los datos y las operaciones. También identifica el mecanismo para invocar el servicio.
 - *Implementación del Servicio*: Implementación física del servicio que proporciona la capacidad requerida o la lógica de negocio.

Arquitectura Orientada a Servicios (SOA)

SOA está basada en los siguientes conceptos:

- **Participantes del Servicio:**
 - *Consumidor del Servicio:* Entidad que hace uso del servicio ofrecido por un proveedor de servicios. El consumidor busca en el repositorio de servicios e identifica los detalles del servicio, incluyendo su interfaz. Una vez que el servicio ha sido localizado, el consumidor lo invoca utilizando el mecanismo apropiado.
 - *Proveedor del Servicio:* Entidad que ofrece un servicio específico o una funcionalidad específica. El proveedor generalmente registra la funcionalidad que provee la interfaz y que tiene que ser invocada para hacer uso del servicio que se encuentra en el repositorio de servicios.

Arquitectura Orientada a Servicios (SOA)

SOA está basada en los siguientes conceptos:

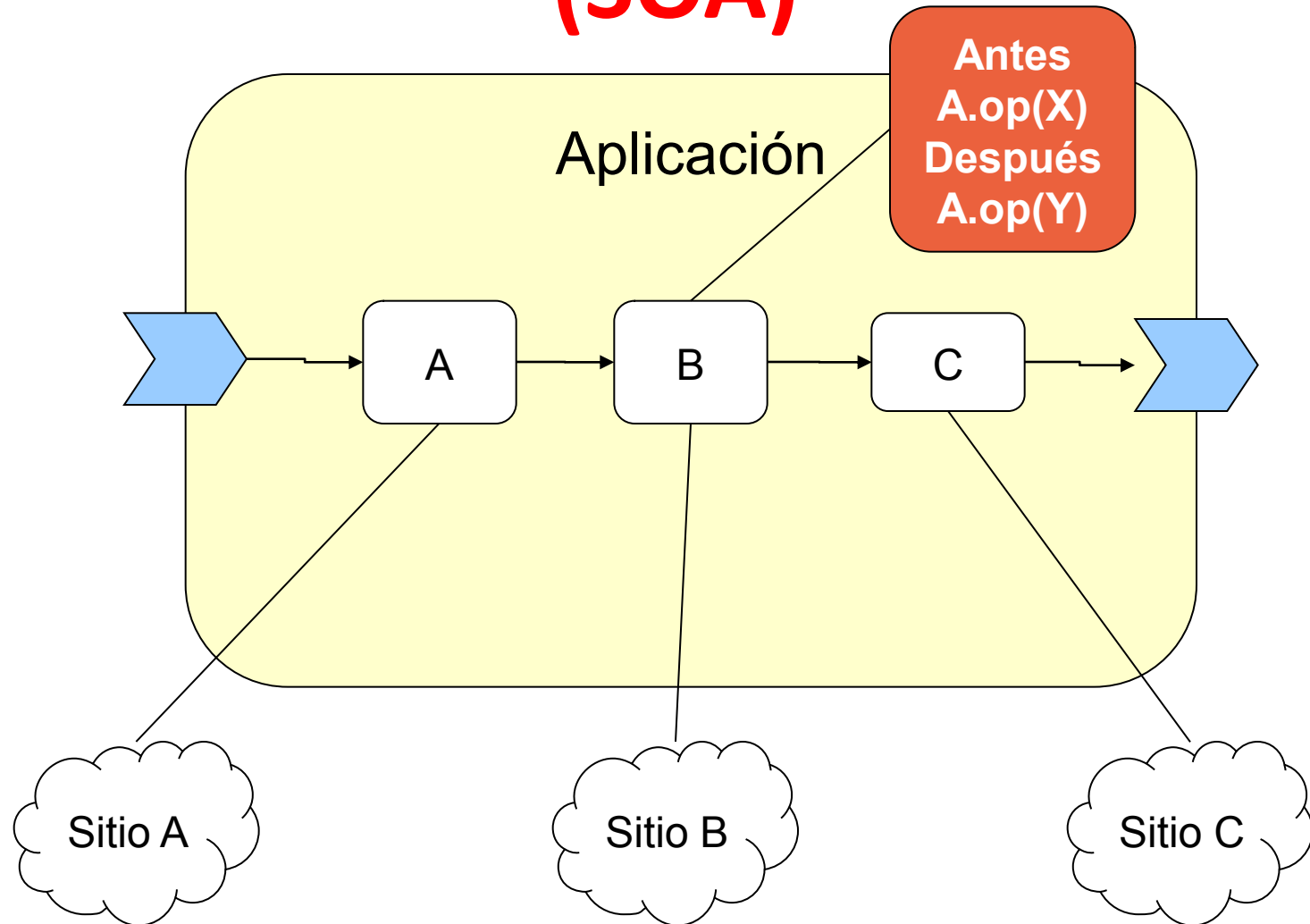
- **Repositorio de Servicios:** Proporciona las instalaciones necesarias para descubrir y utilizar los servicios. Un repositorio almacena los detalles acerca de los servicios que se pueden invocar (por ejemplo, contrato del servicio) y la forma de invocarlos (por ejemplo, interfaces de servicio, ubicación física, etc.)
- **Bus de Servicio:** Proporciona la conectividad necesaria entre los participantes del servicio. Un bus de servicio puede ser capaz de conectar a los participantes con las tecnologías y entornos heterogéneos.

Arquitectura Orientada a Servicios (SOA)

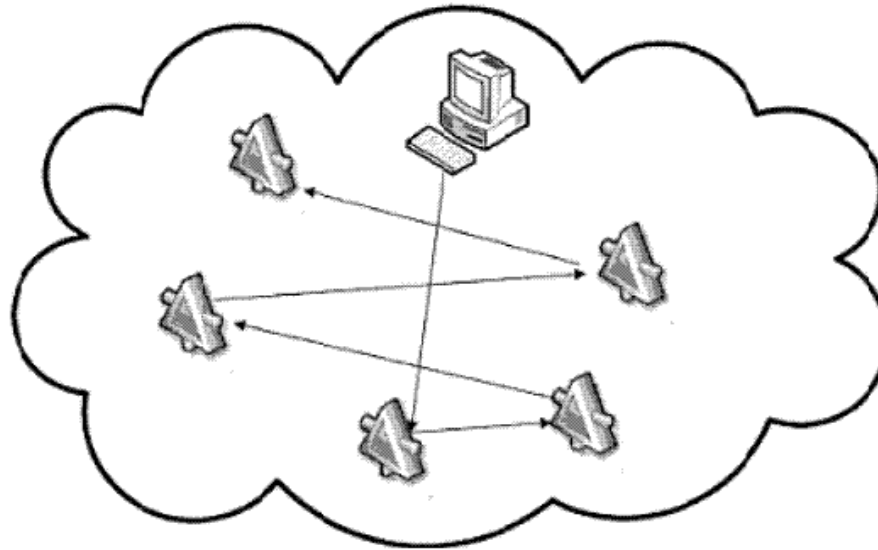
servicio es una funcionalidad concreta del sistema, que puede estar interactuando con otros (servicios) a través de una estructura muy bien definida de intercambio de mensajes.

- Un servicio realiza una tarea concreta y sin memoria,
- Un ejemplo es un servicio de verificación de la cédula de identidad de una persona, al ser invocado por usuarios, sistemas, e incluso otros servicios, a través de una interfaz estándar predefinida, debe dar la misma respuesta.
- Un servicio al ensamblarlo o acoplarlo con otros servicios (composición) , realiza tareas más complejas, como por ejemplo en el caso anterior, determinar la solvencia de un trabajador en el IVSS.

Arquitectura Orientada a Servicios (SOA)



Arquitectura Orientada a Servicios (SOA)



- La arquitectura orientada a servicios es un paradigma para **organizar y utilizar capacidades distribuidas**, definidas como servicios, que pueden estar **controladas bajo diferentes propietarios e implementadas bajo diferentes tecnologías**.
- La arquitectura orientada a servicios define la base para que un **conjunto de servicios independientes puedan colaborar entre sí**.

Arquitectura Orientada a Servicios (SOA)

Propiedades SOA

- **Vista lógica:** El servicio es una vista abstracta y lógica de programas reales, bases de datos, procesos de negocio, etc., definida en términos de qué hace (nada se define respecto al como se hace) .
- **Orientado a mensajes:** El servicio está formalmente definido como un intercambio de mensajes entre entidades suministradoras de servicios y la entidad en si que los orquesta .
- **Orientado a la descripción:** Un servicio se describe a través de meta-datos procesables por el ordenador. La descripción da soporte a la naturaleza pública de SOA: solo aquellos detalles que se exponen públicamente son importantes para la utilización del servicio, por lo cual deben estar incluidos en la descripción.

Arquitectura Orientado a Servicios

SOA define las siguientes capas de software:

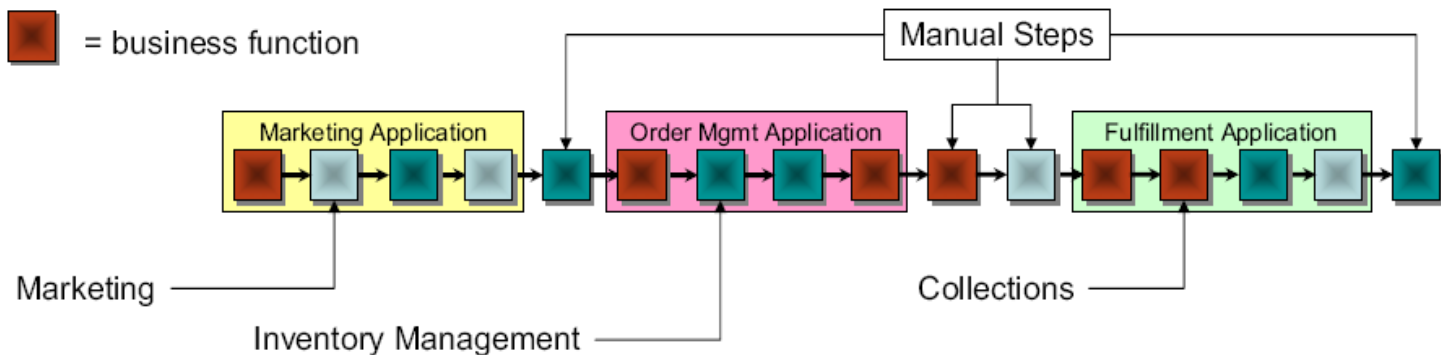
- **Aplicaciones básicas** - Sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad;
- **De exposición de funcionalidades** - Donde las funcionalidades de la capa aplicativa son expuestas en forma de servicios (generalmente como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

Arquitectura Orientado a Servicios

Transformación a SOA

1.- Procesos de negocios tradicionales

Las funciones de negocio están fuertemente unidas con las aplicaciones y tienen interfaces únicas y propietarias, imposibilitando el re-uso.

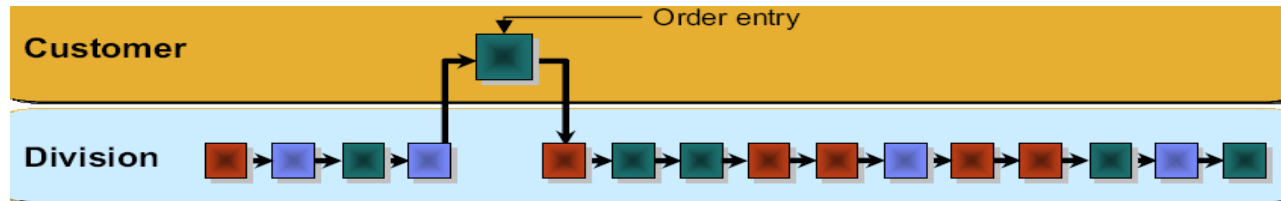


Arquitectura Orientado a Servicios

Transformación a SOA

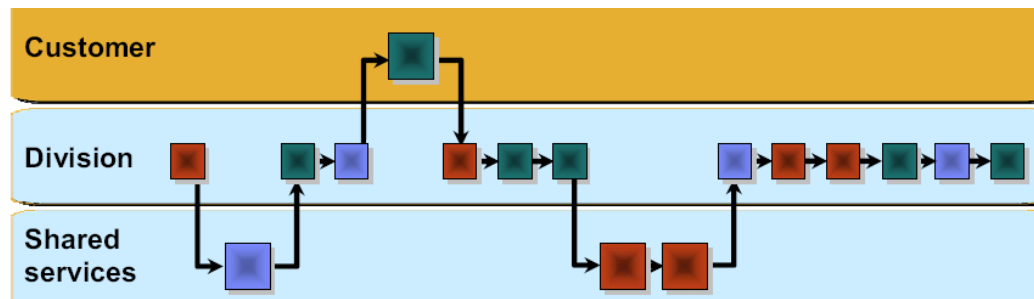
2.- Flexibilidad bajo demanda: autoservicio de clientes

El primero paso para incrementar la flexibilidad de la organización es por incursionar en nuevos canales



3.- Flexibilidad bajo demanda: servicios compartidos

El segundo paso es incrementar la eficiencia de los clientes a través de la creación de servicios compartidos y eliminando la redundancias en la entrega de servicios

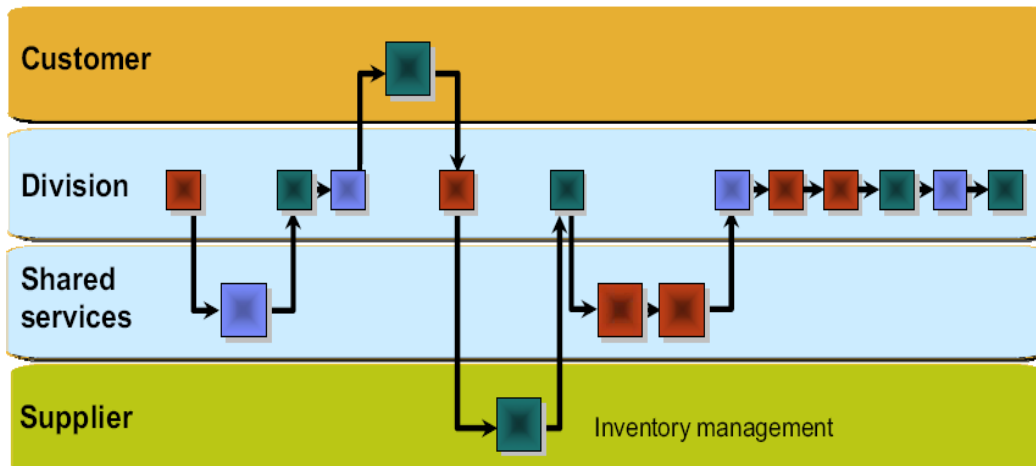


Arquitectura Orientado a Servicios

Transformación a SOA

4.- Flexibilidad bajo demanda: inventario manejado por proveedor

La función de la administración del inventario ha sido reubicada al proveedor. Esto añade eficiencia a todo el proceso, eliminando los servicios duplicados.

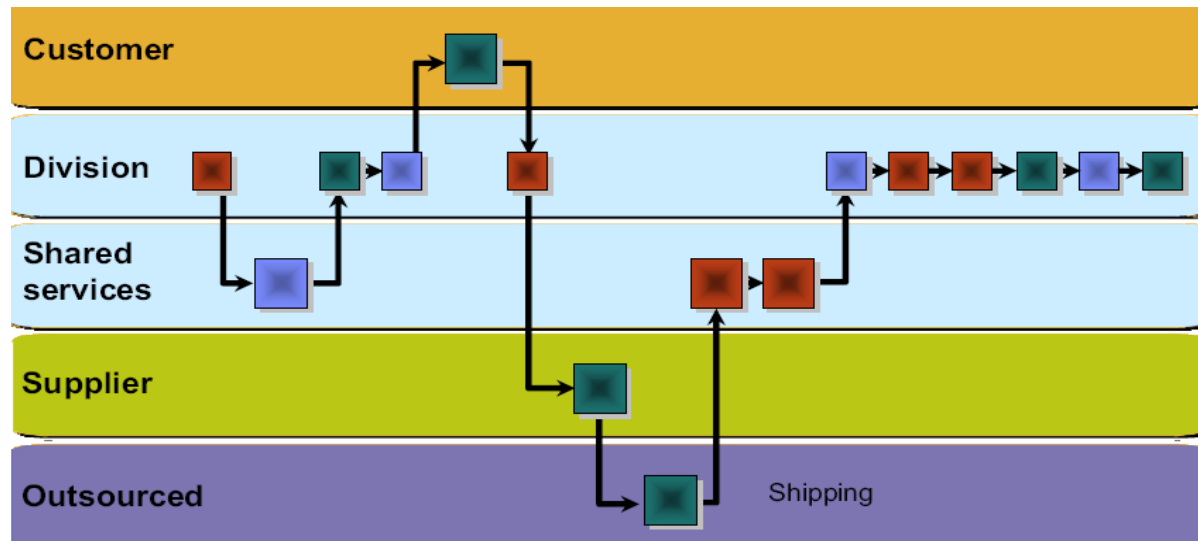


Arquitectura Orientado a Servicios

Transformación a SOA

5.- Flexibilidad bajo demanda: outsource (externalizar)

Otra manera de incrementar la efectividad de la organización es a través de outsourcing de componentes que puedan ser desarrollados por especialistas

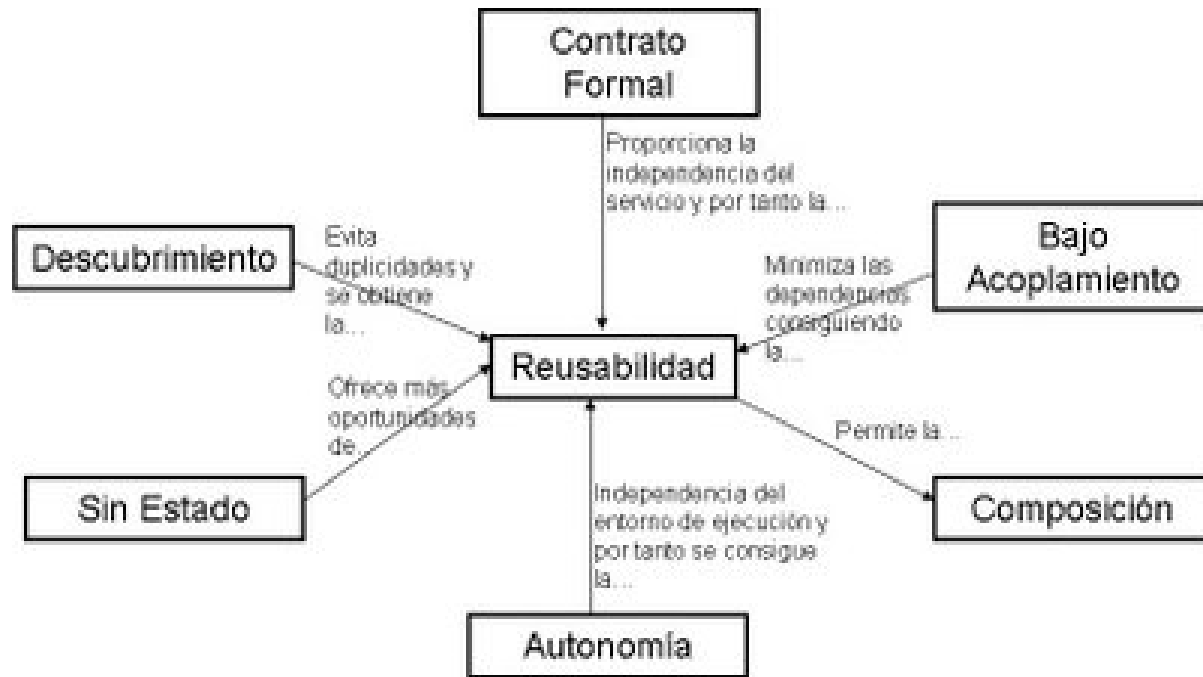


Arquitectura Orientada a Servicios (SOA)

Para que un servicio siga el paradigma SOA, debe poseer propiedades:

- Debe ser débilmente acoplado, .
- Debe poseer un contrato de servicio que muestre la descripción de la tarea que realiza y la manera de comunicarse con él.
- Debe ser autónomo con respecto a la función que realiza.
- Debe ser reusable por terceros en distintas tareas más grandes.
- Debe ocultar los detalles de la lógica de la tarea que realiza al mundo exterior.
- Debe permitir su acoplamiento y coordinación para formar servicios compuestos.
- Debe minimizar la retención de la información específica de una actividad.
- Debe ser diseñado hacia el exterior, tal que se pueda describir donde se encuentra y como puede ser alcanzado a partir de mecanismos de descubrimiento de servicios.

Servicios



Servicios

Existen básicamente tres tipos de servicios, divididos en base a sus funcionalidades:

- **Servicios de negocio:** Son los servicios que representan una tarea de negocio, y que forman parte de un proceso de negocio. Este tipo de servicios suelen ser poco reutilizables porque están orientados a resolver una tarea muy puntual.

Servicios

- **Servicios controladores:** Son los encargados de recibir las peticiones de los clientes y realizar las llamadas necesarias a otros servicios (en la secuencia adecuada) para devolver una respuesta. Es decir, son los servicios encargados de coordinar al resto de servicios. Si analizamos bien este tipo de servicios, nos daremos cuenta de que representan a los procesos de negocio que queremos implementar, ya que un proceso de negocio no es más que un conjunto de tareas ejecutadas en una determinada secuencia para obtener un objetivo.

Servicios

- **Servicios de utilidad:** Son aquellos servicios que se caracterizan por representar una tarea altamente reutilizable. Existen dos tipos, los servicios orientados al negocio que representan una tarea de negocio altamente reutilizable entre aplicaciones y los servicios tecnológicos encargados de encapsular una determinada tecnología y por tanto altamente reutilizables (ej: servicio de acceso a bases de datos relacionales).

Servicios

Una aplicación SOA la podemos dividir en tres capas.

1. La capa de recepción de peticiones (servicios controladores),
2. la capa de tareas (servicios de negocio)
3. la capa de lógica reutilizables (servicios de utilidad).



Arquitectura Orientada a Servicios (SOA)

Servicios Web es una tecnología que provee un eficiente camino para compartir la lógica distribuida de aplicaciones, a través de múltiples máquinas que están corriendo varios sistemas operativos y usando diferentes ambientes de desarrollo en Internet.

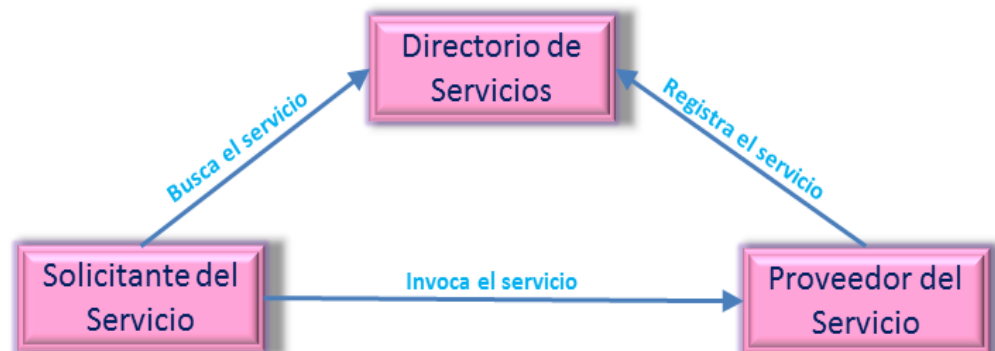
Un servicio web es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

- Para ello, se han definido un conjunto de mecanismos estandarizados para exponer y consumir datos y lógica de aplicaciones sobre protocolos de Internet.

Arquitectura Orientada a Servicios (SOA)

funciones de un Servicio Web:

- **Proveedor del Servicio:** encargado de implementar el servicio, así como hacerlo disponible en Internet.
- **Solicitante del Servicio:** Consumidor del servicio web, este solicitante utiliza un servicio existente en la Web al abrir una conexión de red y enviar una petición en lenguaje de marcas extensible (XML).
- **Registro del Servicio:** Directorio centralizado en servicios, este registro provee un repositorio central donde los desarrolladores pueden publicar nuevos servicios, así como encontrar los existentes.



Arquitectura Orientada a Servicios (SOA)

La pila de protocolos de un servicio web

- **Servicio de Transporte:** Esta capa es responsable del transporte de mensajes entre aplicaciones, incluye los protocolos HTTP), SMTP (FTP) y otros protocolos.
- **Mensajería XML:** Esta capa es la responsable de codificar los mensajes en un formato común de lenguaje de marcas extensible (XML), para que los mensajes puedan ser entendidos. Actualmente, esta capa incluye dos protocolos el protocolo de llamada a procedimiento remoto (XML-RPC) y el protocolo de acceso simple a un objeto (SOAP).
- **Descripción del Servicio:** Esta capa es la responsable de describir la interfaz pública de un servicio web específico, esta definición es manejada por medio del lenguaje de descripción de un servicio web (WSDL).
- **Descubrimiento del Servicio:** Esta capa es responsable de centralizar servicios dentro de un repositorio común, y proveer una funcionalidad de fácil publicación y búsqueda de los mismos, para esto se usa el protocolo de descripción universal, descubrimiento e integración (UDDI).

Arquitectura Orientada a Servicios (SOA)



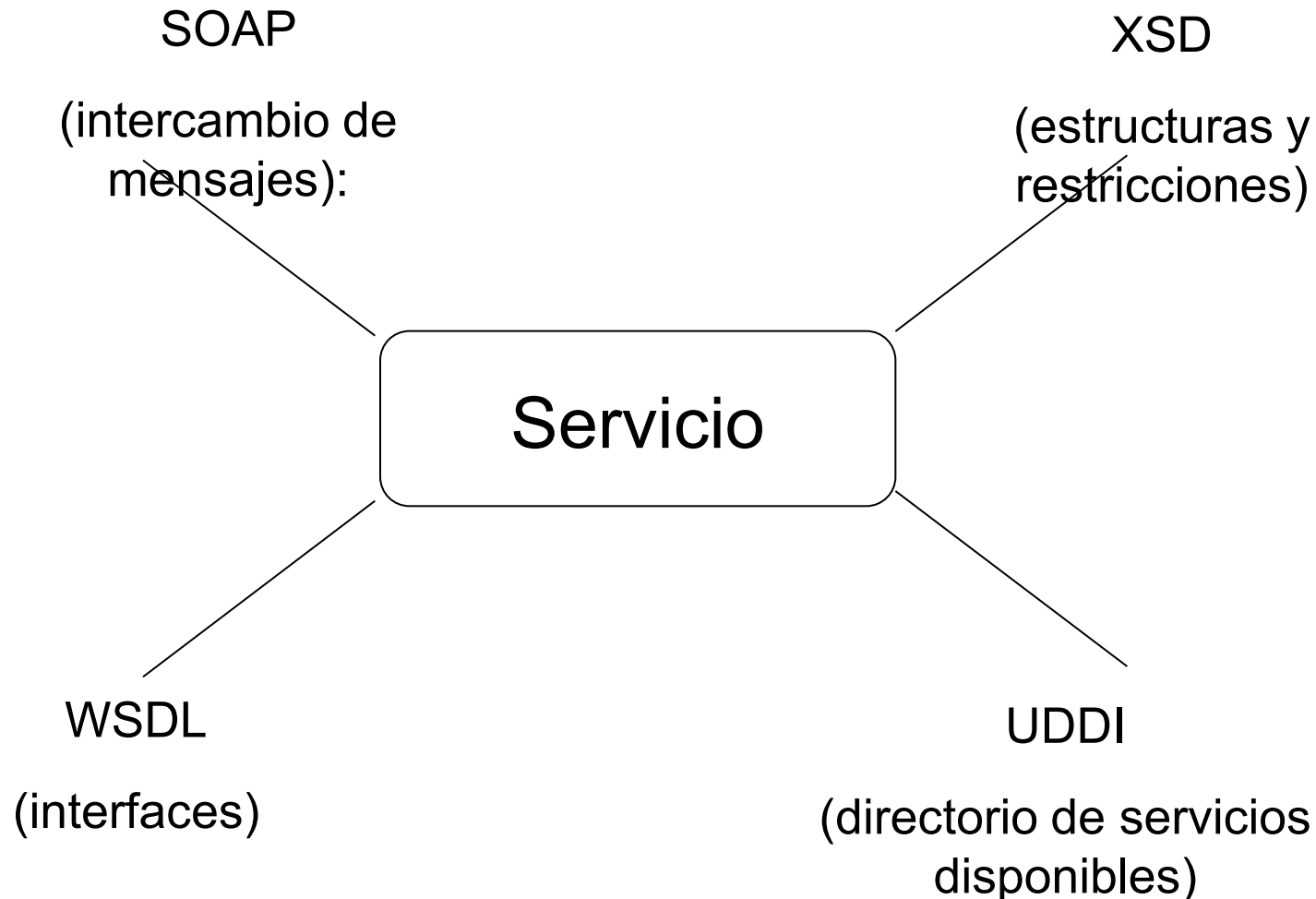
Arquitectura Orientada a Servicios (SOA)

protocolos y estándares :

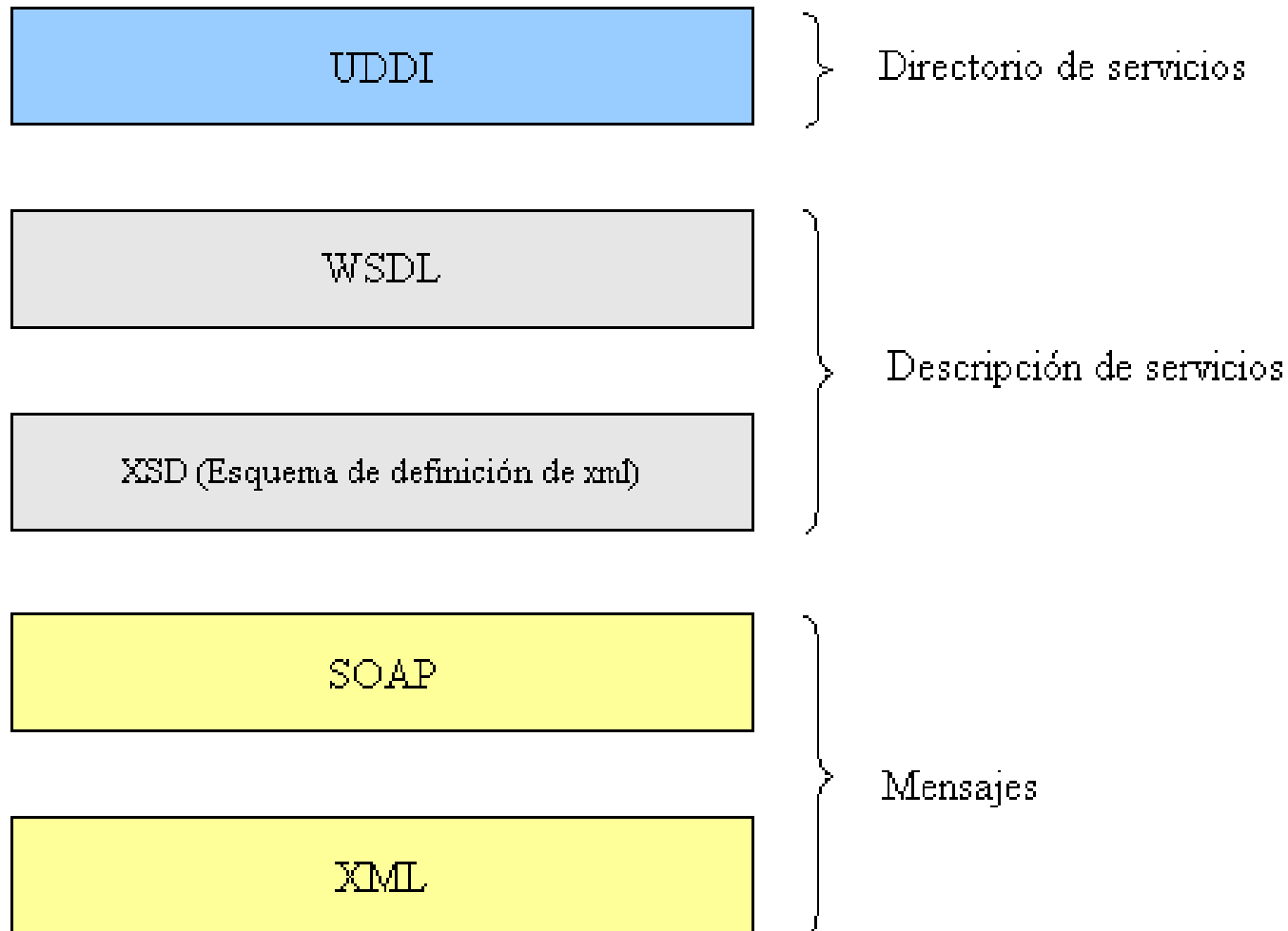
- **SOAP (Simple Object Access Protocol):** Es un protocolo basado en XML para el intercambio de mensajes, el cual es independiente de cualquier protocolo de transporte.
- **XSD (XML Schema Definition):** Es un lenguaje utilizado para describir las estructuras y restricciones de los contenidos en un documento XML.
- **WSDL (Web Services Description Language):** Es un lenguaje de definición de interfaces basado en XML, el cual separa la definición de la función de la implementación de un servicio. Provee a los desarrolladores y sistemas, la manera de entender la operación de los servicios y los mecanismos de invocación.
- **UDDI (Universal Description, Discovery and Integration):** Define una especie de directorio de servicios disponibles, y sus proveedores.

Arquitectura Orientada a Servicios (SOA)

Protocolos y estándares:



Arquitectura Orientada a Servicios (SOA)



Arquitectura Orientada a Servicios (SOA)



SOAP

“un protocolo de servicios web que define la manera de comunicar dos objetos por medio del intercambio de mensajes basados en XML”



SOA

- Mensaje SOAP es un documento XML que consta de los siguientes elementos
 - **Envelope (sobre):** Define un marco de referencia general para expresar que es lo que contiene el mensaje, quien debe recibirlo, y si es opcional u obligatorio.
 - **Header (cabecera):** Contiene meta información referente al mensaje, contiene la información referente del receptor y un conjunto de opciones de entrega.
 - **Body (cuerpo):** Contiene la información de la llamada y de la respuesta.
- Los objetivos principales de SOAP son:
 - Establecer un protocolo de invocación de servicios remotos, basado en protocolos estándares de Internet: HTTP para la transmisión y XML para la codificación de datos.
 - Independencia de plataforma, lenguaje de desarrollo e implementación.

SOA

WSDL (Web Service Description Language).

En esencia, WSDL es un contrato entre el proveedor del servicio y el cliente, mediante el que el proveedor del servicio indica:

- Qué funciones se pueden invocar
- Qué tipos de datos utilizan esas funciones.
- Qué protocolo de transporte se utilizara para el envío y recepción de los mensajes (generalmente por medio de mensajes SOAP).
- Cómo se accede a los servicios (Generalmente mediante URL)

SOA

Integración de servicios

- **Unión en tiempo de desarrollo:** El desarrollador es responsable de localizar toda la información necesaria desde el repositorio de servicios, con el fin de programar el consumidor de servicios para acceder al servicio.
- **Unión en tiempo de ejecución:** El servicio se invoca dinámicamente.
 - *Unión por nombre en tiempo de ejecución:* El nombre del servicio y la interfaz se conoce en tiempo de desarrollo y el consumidor del servicio se pueden programar antes.
 - *Unión por propiedades en tiempo de ejecución:* Una o varias interfaces de servicios se conocen en tiempo de desarrollo y el servicio adecuado se descubre en tiempo de ejecución según sus propiedades.
 - *Unión por reflexión en tiempo de ejecución:* La interfaz de servicio no se conoce en tiempo de desarrollo y el consumidor la necesita para interpretar de forma dinámica la semántica del servicio.

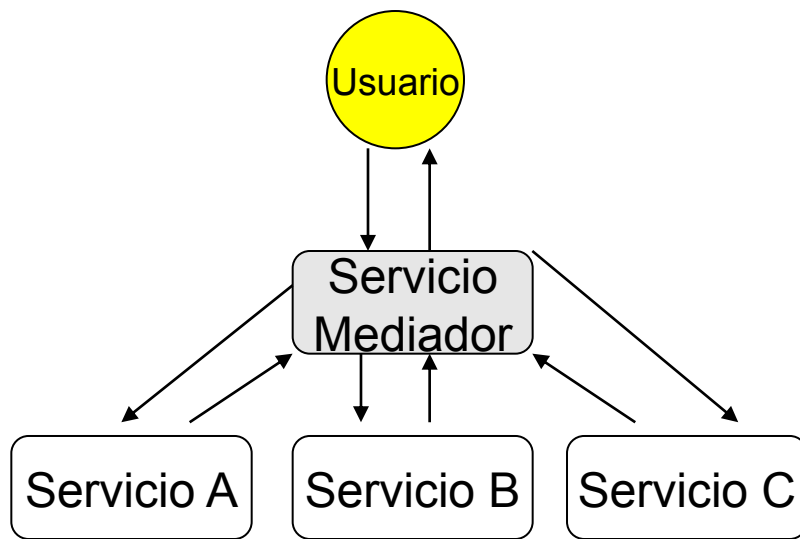
SOA

Integración de servicios

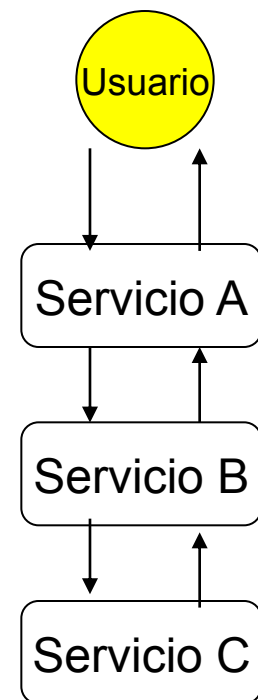
- **Orquestación:** Es diseñado para componer y ejecutar una serie de tareas, las cuales requieren un conductor central que siempre esta a cargo de la ejecución.
- **Coreografía:** Es un conjunto de relaciones punto a punto entre servicios individuales. No existe un conductor central, por lo que los servicios participantes interactúan con otros basándose en conjuntos definidos de contratos.

Arquitectura Orientada a Servicios (SOA)

Composición de Servicios



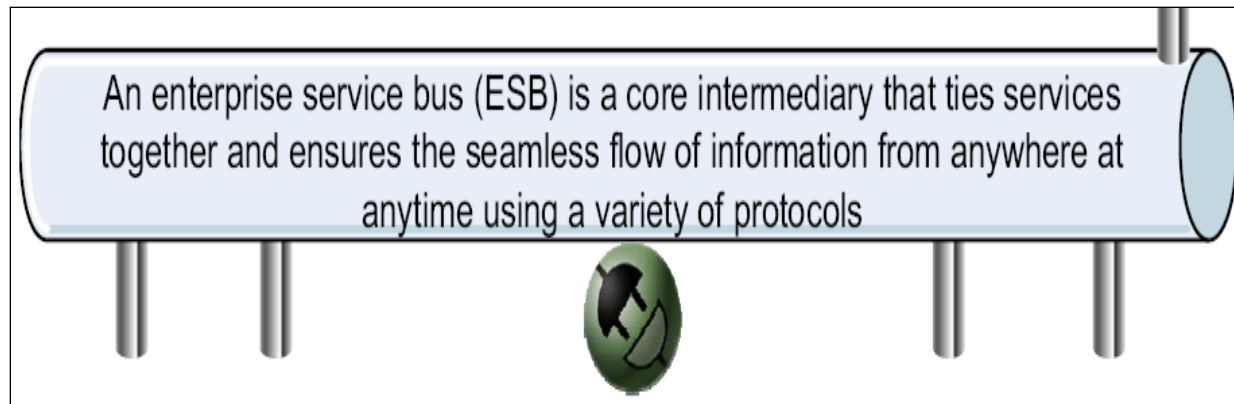
Orquestación



Coreografía

SOA: Puntos de Entrada

- ✓ Conectividad de Servicios para SOA, diseñado para ayudar a simplificar tu ambiente de TI con más seguridad, confiabilidad y escalabilidad que permita conectarte dentro y más allá de tu negocio.

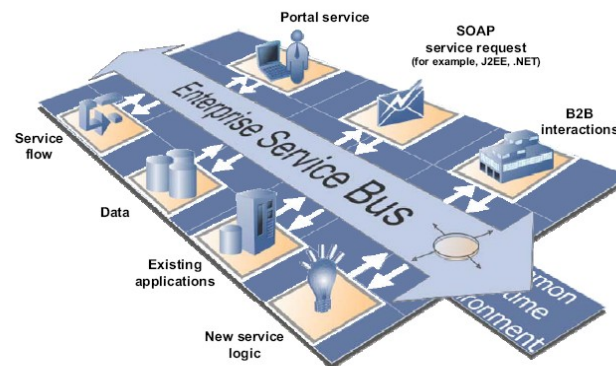


ESB – Enterprise Server Bus

Un Bus de Servicio Empresarial (ESB) es un intermediario:

- Red de servicios colaboradores.
- Un conjunto de aplicaciones expuestas como servicios, además de servicios intermediarios que se preocupan por la seguridad y monitoreo.
- Se puede añadir nuevos servicios, sin afectar a los ya existentes.

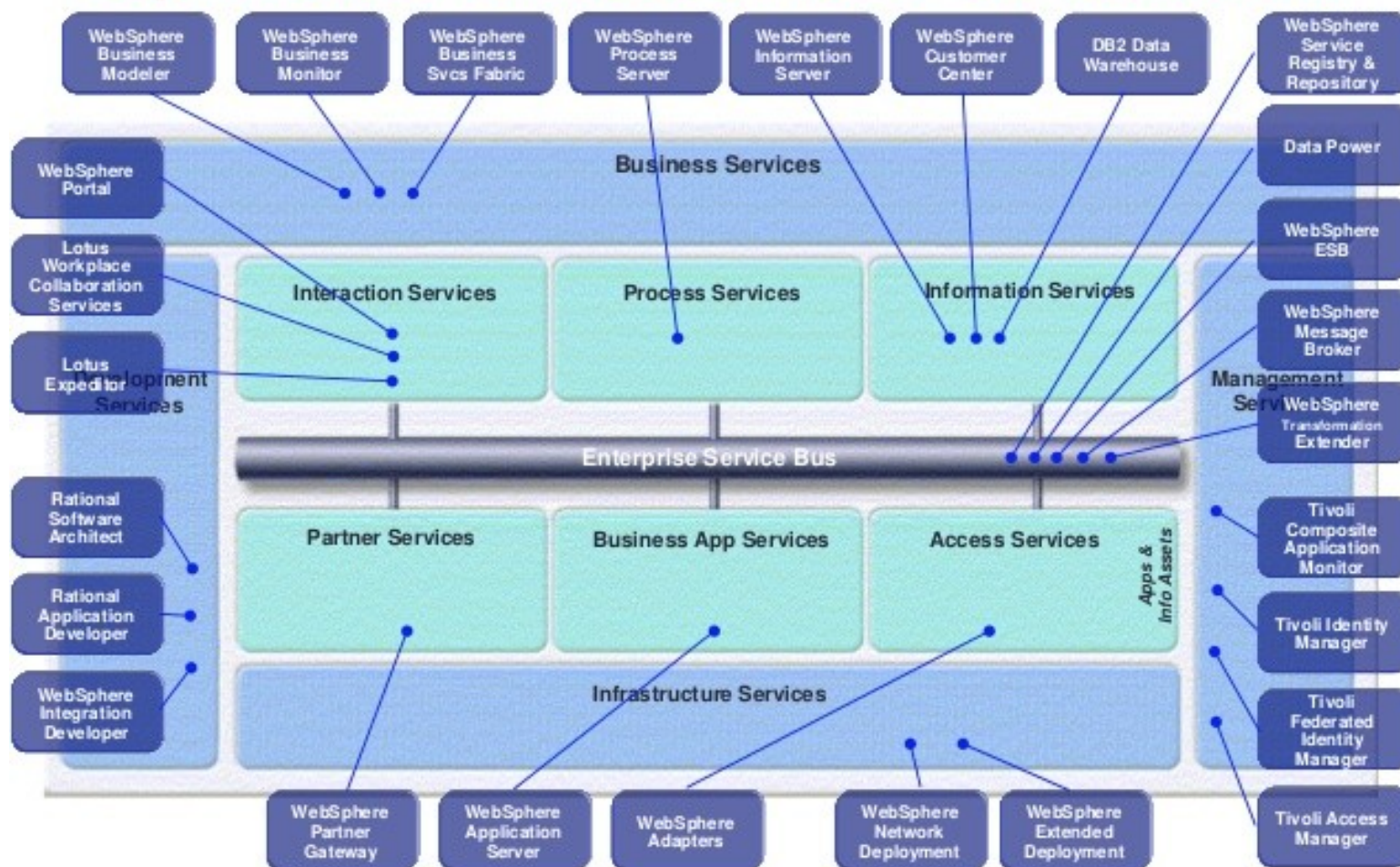
Service-oriented architecture and the ESB



ESB – Enterprise Server Bus

47

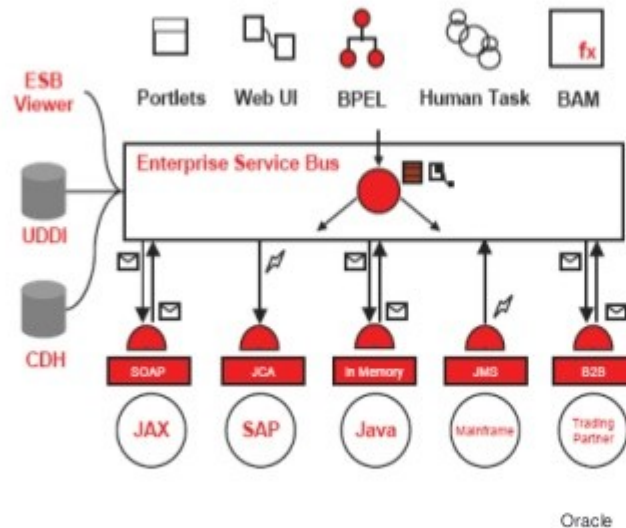
Arquitectura Orientada a Servicios – Servicios SOA-IBM



ESB – Enterprise Server Bus

54

SOA → Solución Tecnológica



- **Gestión de la comunicación:**
 - Adaptadores: Tecnología y aplicaciones
 - Transformación y Mensajería
 - Enrutamiento basado en contexto
 - Publicación y suscripción
 - Registro UDDI
- **Gestión de Procesos**
 - Orquestación de servicios
 - Gestión de flujos de trabajo
 - Gestión de eventos
- **Seguridad**
 - Autenticación
 - Control de acceso
 - Verificación de firma
 - Encriptación

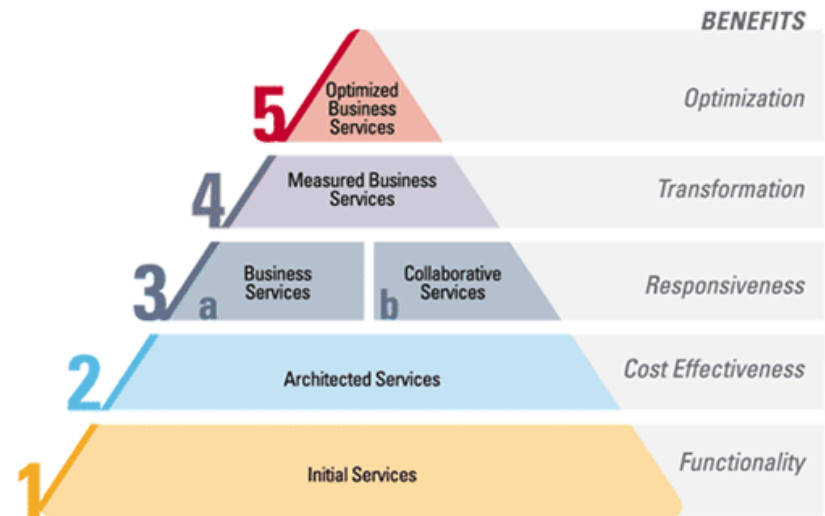
Relación entre un Servicio Web y SOA

SOA no implica el uso de tecnologías de servicio web, hay implementaciones SOA que no utilizan servicios web. Sin embargo los servicios Web son la principal tecnología para habilitar y realizar SOA hoy en día, porque sus características para complementar los conceptos fundamentales de SOA incluyen:

Múltiples Acoplamientos

Interoperabilidad

Amplio Apoyo a la Industria



Arquitectura Orientada a Servicios (SOA)

Fallas de Servicios:

- Físicas: no disponible.
- De desarrollo:
 - Incompatibilidad de parámetros.
 - Cambios de interfaces.
 - Inconsistencia de flujos.
 - Mal entendimiento de la funcionalidad.
- De Interacción:
 - Contenido.
 - Tiempo.

Arquitectura Orientada a Servicios (SOA)

Métodos de Reparación:

- **Servicio:**
 - Reintentar (Retry).
 - Substituir el Servicio.
 - Completar Incompatibilidad de Parámetros.
 - Reasignar el Servicio.
- **Flujo:**
 - Substituir el Flujo.
 - Rehacer.
 - Conductas Alternativas.
 - Saltar servicio.
 - Cambio de Parámetros

SOA basada en Computación autónoma

