Introducción al middleware

Arquitectura Orientada a Servicios

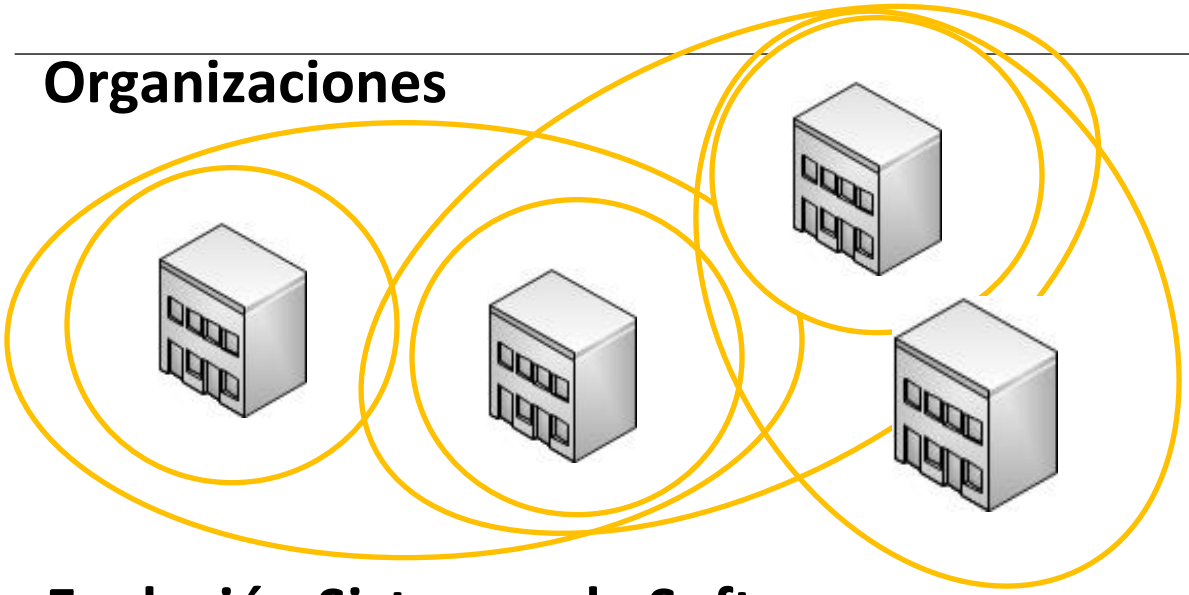
Agenda

- Computación Orientada a Servicios
  - Motivación y Principales Conceptos
  - Principios
  - Beneficios y Desafíos
- Arquitectura Referencia para SOA
- Middleware para SOA
- SOA y otras Tecnologías

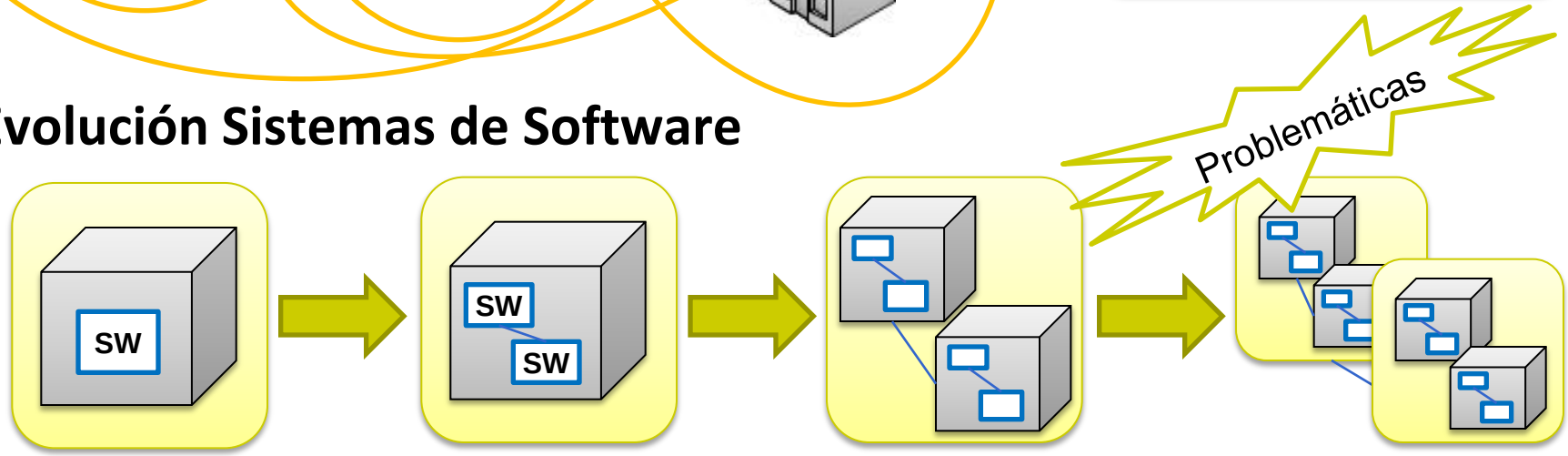
Computación Orientada a Servicios

Motivación

Organizaciones



Evolución Sistemas de Software

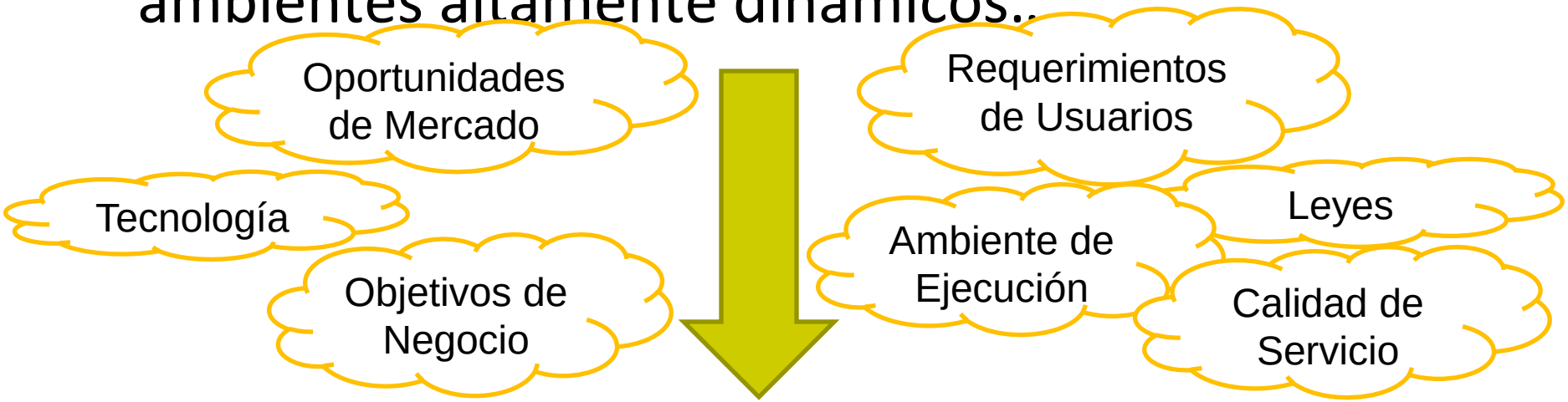


(Di Nitto et al, 2008)

Computación Orientada a Servicios

Motivación

- Los Sistemas de Software actuales operan en ambientes altamente dinámicos..



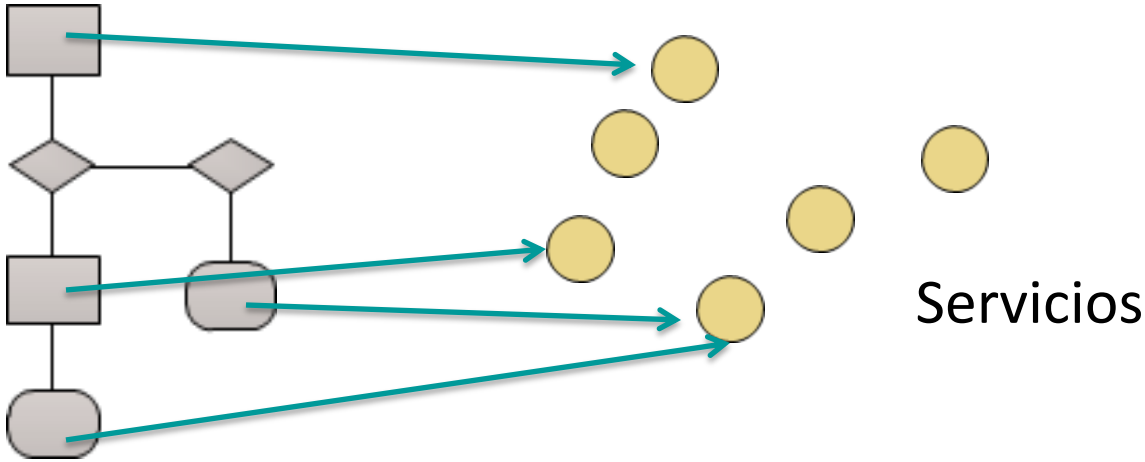
…necesitan cada vez más ser capaces de adaptarse ágilmente ante distintos tipos de cambios…

Computación Orientada a Servicios

Conceptos: Service Oriented Computing, SOC

- SOC es un paradigma de computación que utiliza servicios como elementos fundamentales para dar soporte al desarrollo rápido, y de bajo costo, de aplicaciones distribuidas en ambientes heterogéneos.

Aplicaciones Basadas en Servicios



(Papazoglou and Heuvel 2007)

Computación Orientada a Servicios

Conceptos: Servicios

- Los Servicios son:
  - entidades de software autónomas, auto-contenidas e independientes de la plataforma
  - proveen funcionalidades de negocio
  - tienen una interfaz pública
  - pueden ser descubiertos, invocados y combinados de forma dinámica

(Papazoglou and Heuvel 2007)

Computación Orientada a Servicios

Conceptos: Arquitectura Orientada a Servicios

- La puesta en práctica del paradigma SOC requiere la implementación de Arquitecturas Orientadas a Servicios (SOAs)

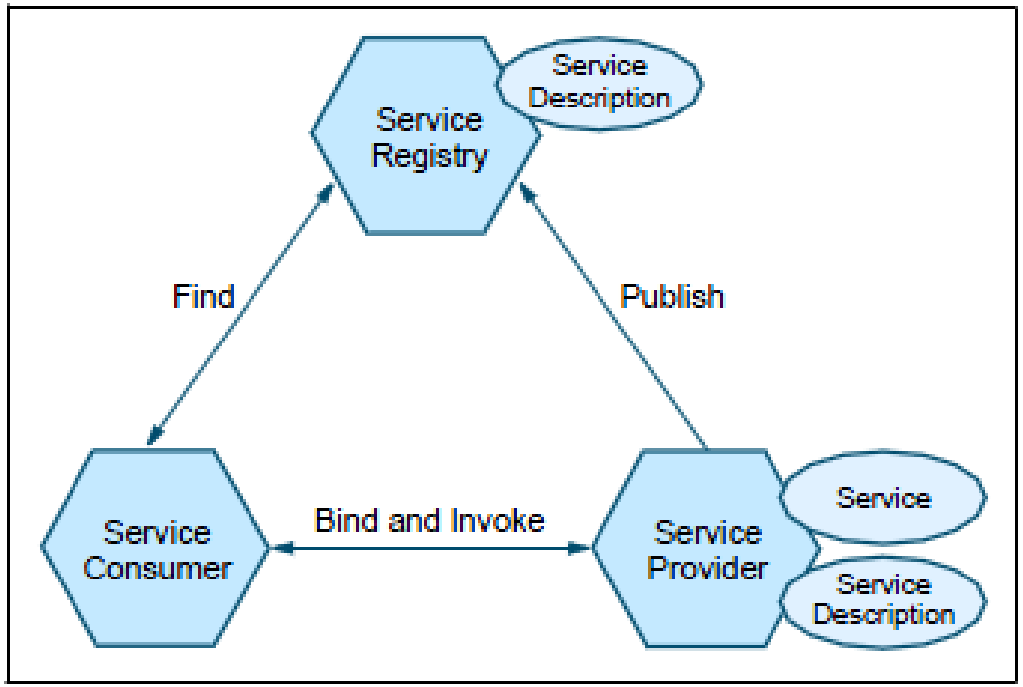
- Una SOA es una forma lógica de diseñar un sistema de software para proveer servicios, a usuarios finales, aplicaciones u otros servicios, a través de interfaces públicas que pueden ser descubiertas.

(Papazoglou and Heuvel 2007)

Computación Orientada a Servicios

Conceptos: Arquitectura Orientada a Servicios

- Las interacciones en una SOA siguen el paradigma “find, bind and invoke”



(Endrei et al., 2004)

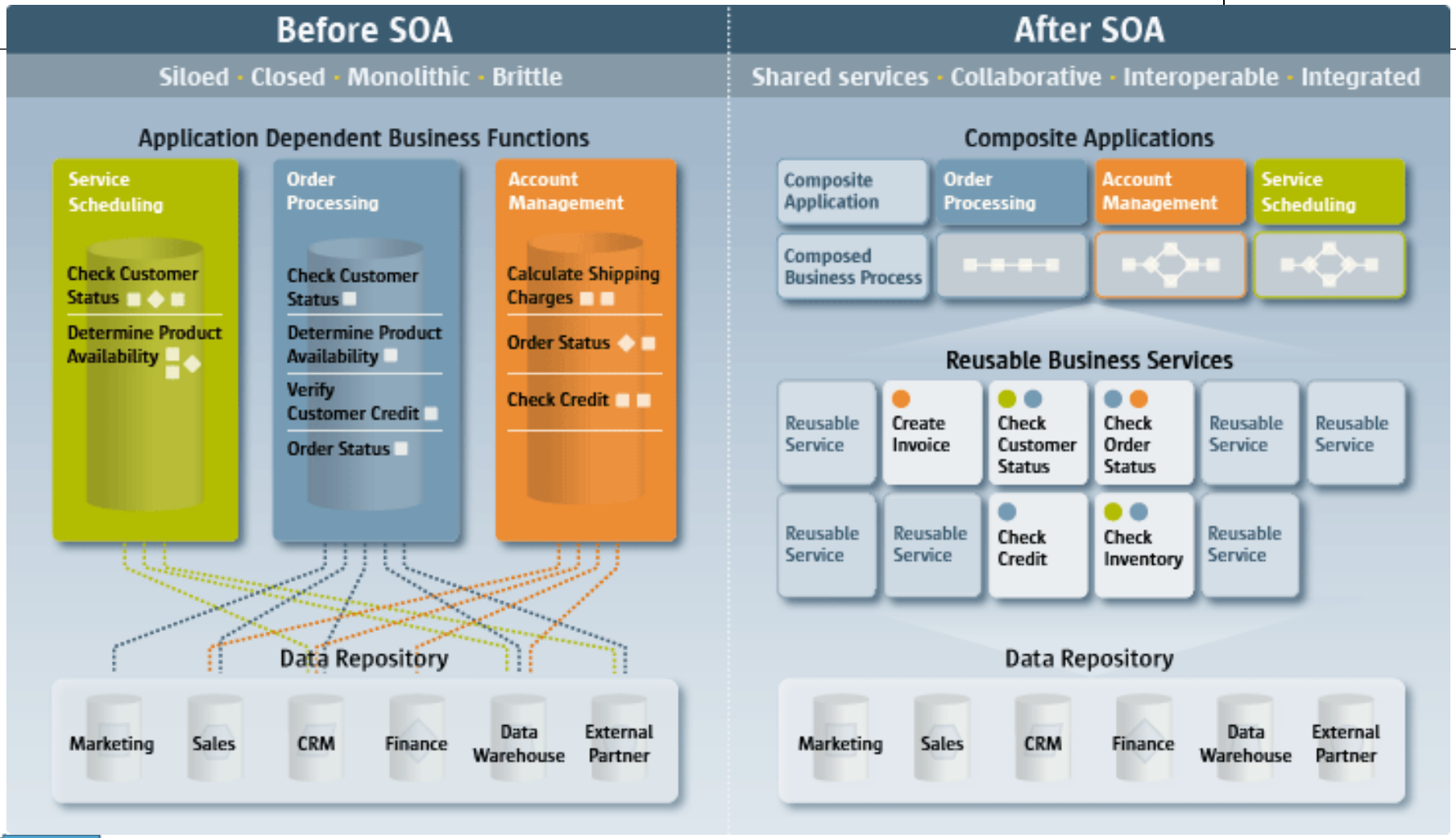
Computación Orientada a Servicios

Conceptos: Arquitectura Orientada a Servicios

- En una SOA, los servicios se mapean a funcionalidades de negocio
- Cada servicio tiene una interfaz bien definida que le permite ser publicado, descubierto e invocado
- Una organización puede publicar sus servicios:
  - de forma externa, para interactuar con sus socios de negocio
  - de forma interna a la organización

Computación Orientada a Servicios

Conceptos: Arquitectura Orientada a Servicios



http://ibima.net/articles/JSSD/2012/169423/

Computación Orientada a Servicios

Principios

- Standardized Service Contracts
- Service Loose Coupling
- Service Abstraction
- Service Reusability
- Service Autonomy
- Service Statelessness
- Service Discoverability
- Service Composability

(Erl, 2008)

Computación Orientada a Servicios

Principios: Standarized Service Contracts

- Los servicios en el mismo inventario se deben ajustar a los mismos estándares para el diseño de contratos
- Esta estandarización se debe dar a nivel de:
  - Funcionalidad
  - Modelo de Datos
  - Políticas

(Erl, 2008)

Computación Orientada a Servicios

Principios: Service Loose Coupling

- El contrato del servicio debe estar idealmente desacoplado de los detalles de tecnología e implementación
- Esto promueve un ambiente donde los servicios y sus consumidores pueden evolucionar con un mínimo impacto entre ellos

(Erl, 2008)

Computación Orientada a Servicios

Principios: Service Abstraction

- Los contratos de los servicios sólo deben incluir la información esencial y la información de los servicios se limita a la que se publica en su contrato
- El objetivo principal de este principio es evitar la proliferación de información innecesaria
- Apunta a lograr el balance adecuado de “information hiding”

(Erl, 2008)

Computación Orientada a Servicios

Principios: Service Reusability

- Los servicios contienen y expresan lógica agnóstica y se pueden posicionar como recursos empresariales reutilizables
- Algunos objetivos:
  - Permitir que la lógica sea reutilizada (ROI)
  - Aumentar la agilidad de negocio

(Erl, 2008)

Computación Orientada a Servicios

Principios: Service Autonomy

- Los servicios son autónomos, esto es, tienen un alto control sobre su entorno de ejecución
- La autonomía representa la independencia con que el servicio ejecuta su lógica
- Beneficios:
  - Se aumenta la confiabilidad (reliability)
  - Se aumenta la previsibilidad

(Erl, 2008)

Computación Orientada a Servicios

Principios: Service Statelessness

- Los servicios minimizan el consumo de recursos difiriendo el manejo de su información de estado cuando es necesario
- Esto apunta a aumentar la escalabilidad y poder manejar más solicitudes de forma confiable

(Erl, 2008)

Computación Orientada a Servicios

Principios: Service Discoverability

- Los servicios se complementan con metadatos comunicativos a través de los cuales pueden ser descubiertos e interpretados de forma efectiva
- Esto apunta a incrementar el reuso de los servicios y reducir la posibilidad de desarrollar servicios cuya funcionalidad se solape

(Erl, 2008)

Computación Orientada a Servicios

Principios: Service Composability

- Los servicios participan en composiciones de forma efectiva, sin importar el tamaño o complejidad de la composición
- Este principio permite la agilidad ya que promueve la construcción de sistemas a partir de servicios existentes

(Erl, 2008)

Computación Orientada a Servicios

Beneficios

- SOA permite a las organizaciones aprovechar sus activos informáticos, encapsulándolos como servicios que proveen funciones de negocio

- SOA permite mayor agilidad para salir al mercado, posibilitando componer servicios de negocio en base a otros ya existentes

(Papazoglou and Heuvel 2007)

Computación Orientada a Servicios

Beneficios

- La posibilidad de utilizar y combinar servicios fácilmente significa en general:
  - menos duplicación de recursos
  - más potencial de reutilización
  - menos costos
- Los principios de SOC soportan y promueven la interoperabilidad

(Papazoglou and Heuvel 2007)

Computación Orientada a Servicios

Desafíos

- Los beneficios reales de SOA se obtienen en general a mediano o largo plazo
- Con SOA se introducen nuevos roles, productos y procesos
- Las tecnologías, estándares y productos para dar soporte a SOA aún están madurando
- Identificación de Servicios y Definición de Contratos de Servicios

(Alluri, 2009)

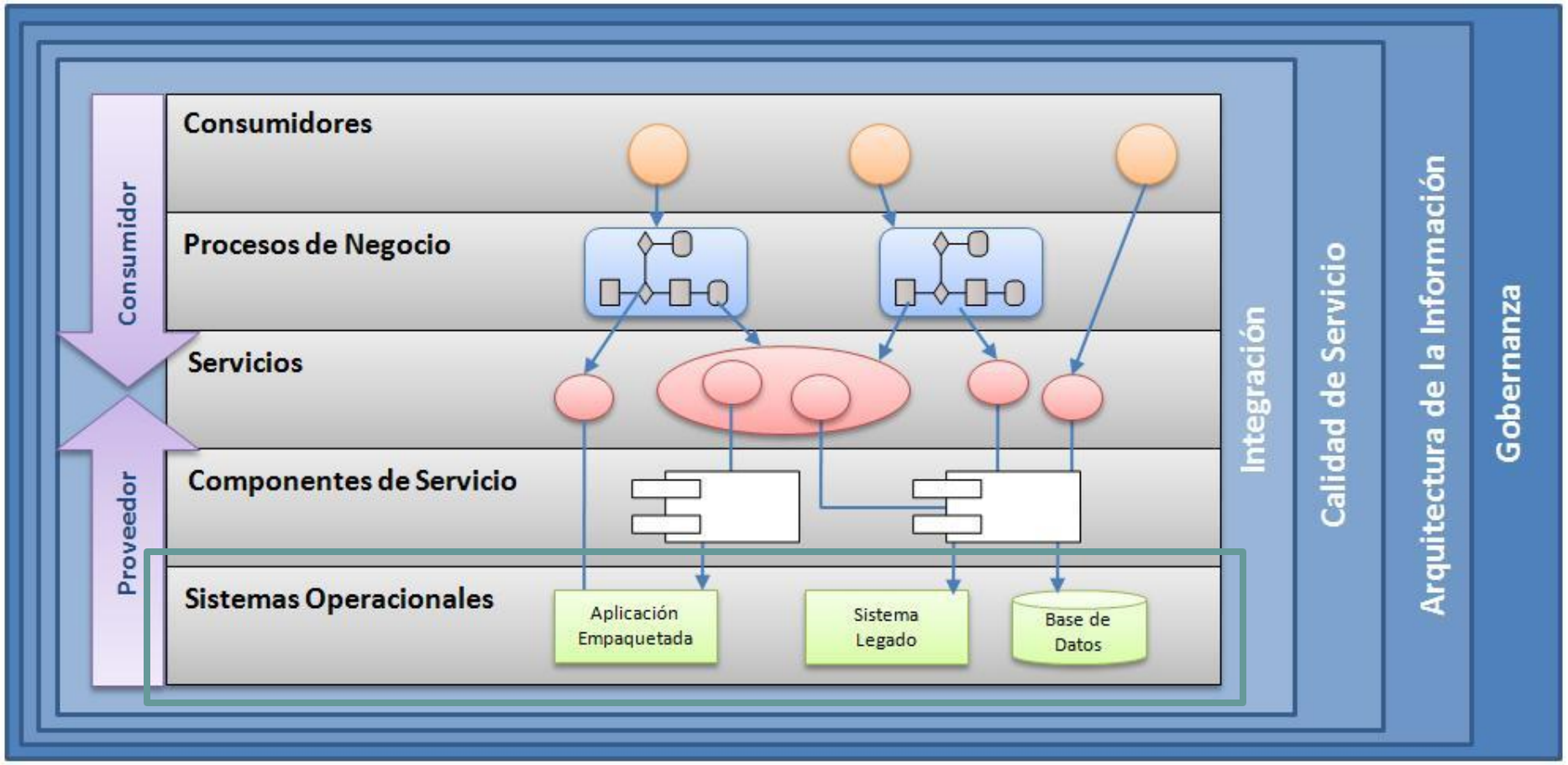
Computación Orientada a Servicios

Desafíos

- Gobernanza de SOA
  - Asegurar que los servicios y las soluciones SOA en una organización se adhieren a las políticas, lineamientos y estándares definidos en base a los objetivos, estrategias y regulaciones aplicadas en la organización
  - Consiste en:
    - Establecer y comunicar las políticas que los empleados deben seguir
    - Brindar a los empleados las herramientas para cumplir con dichas políticas
    - Proveer visibilidad de los niveles de cumplimiento
    - Mitigar las desviaciones

Arquitectura Referencia para SOA

Capas de una SOA



(Arsaniani, 2004)

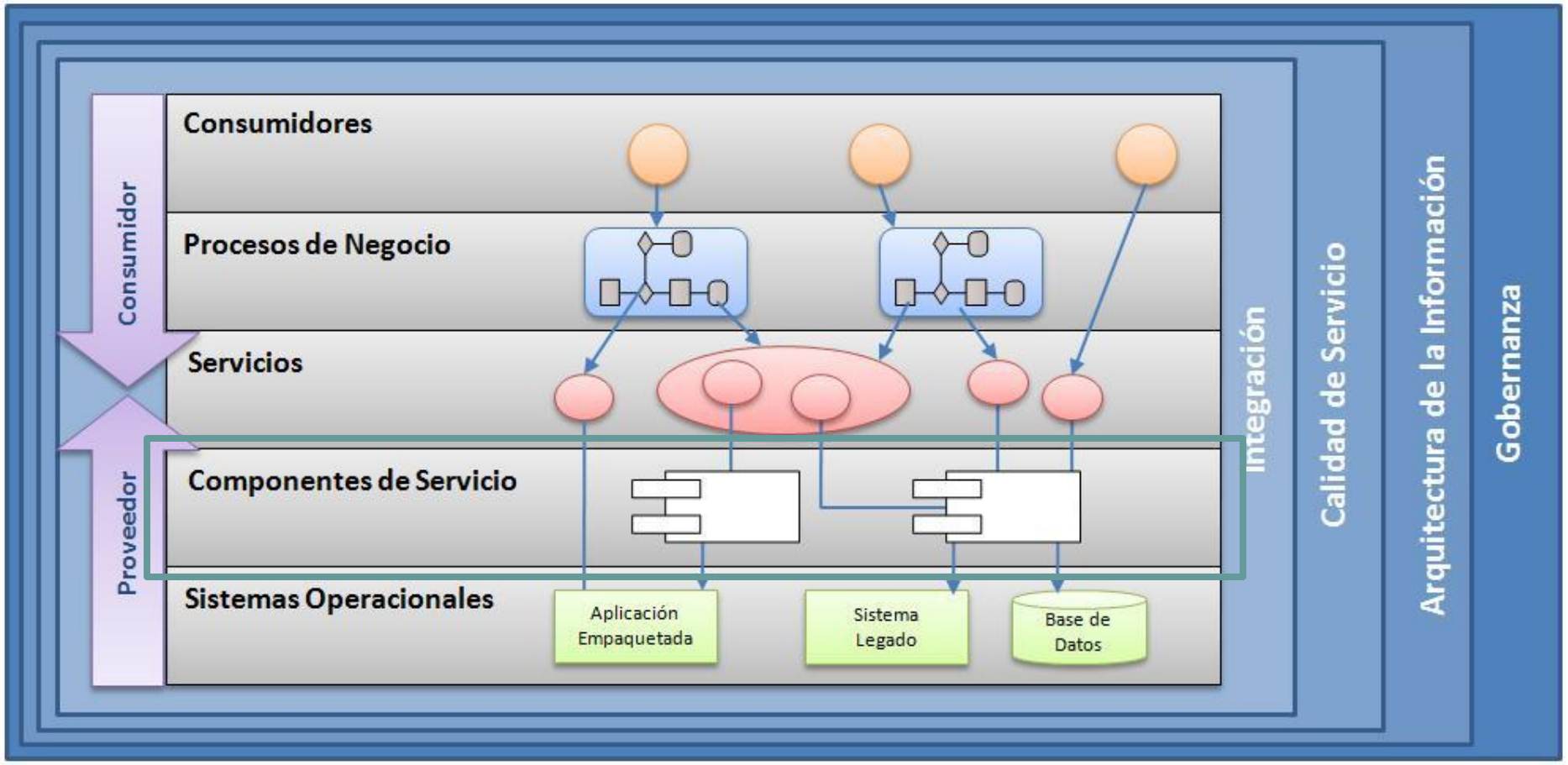
Arquitectura Referencia para SOA

Capa: Sistemas Operacionales

- Activos de aplicaciones que dan soporte a las actividades de negocio de la organización
- Puede incluir aplicaciones monolíticas desarrolladas a medida sobre las plataformas Java EE o .Net, sistemas legados y BDs
- También se incluyen aplicaciones y soluciones empaquetadas, por ej., ERPs y CRMs

Arquitectura Referencia para SOA

Capas de una SOA



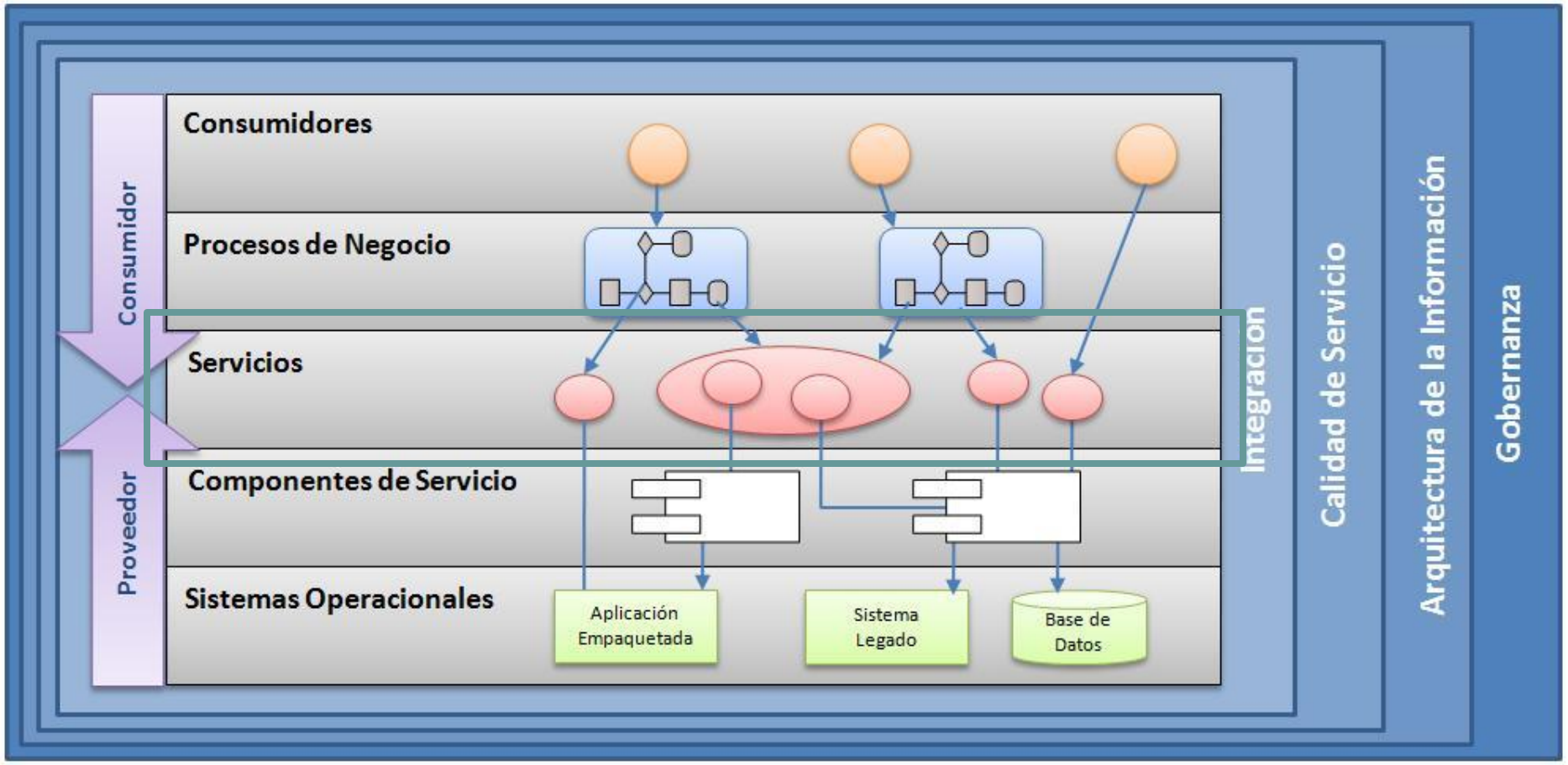
Arquitectura Referencia para SOA

Capa: Componentes de Servicios

- Incluye componentes de software, cada uno de los cuales provee la implementación de un servicio, u operación de un servicio
- Se pueden utilizar funcionalidades de varios sistemas de la capa de Sistemas Operacionales
- Esta capa garantiza la alineación entre la implementación y la descripción del servicio

Arquitectura Referencia para SOA

Capas de una SOA



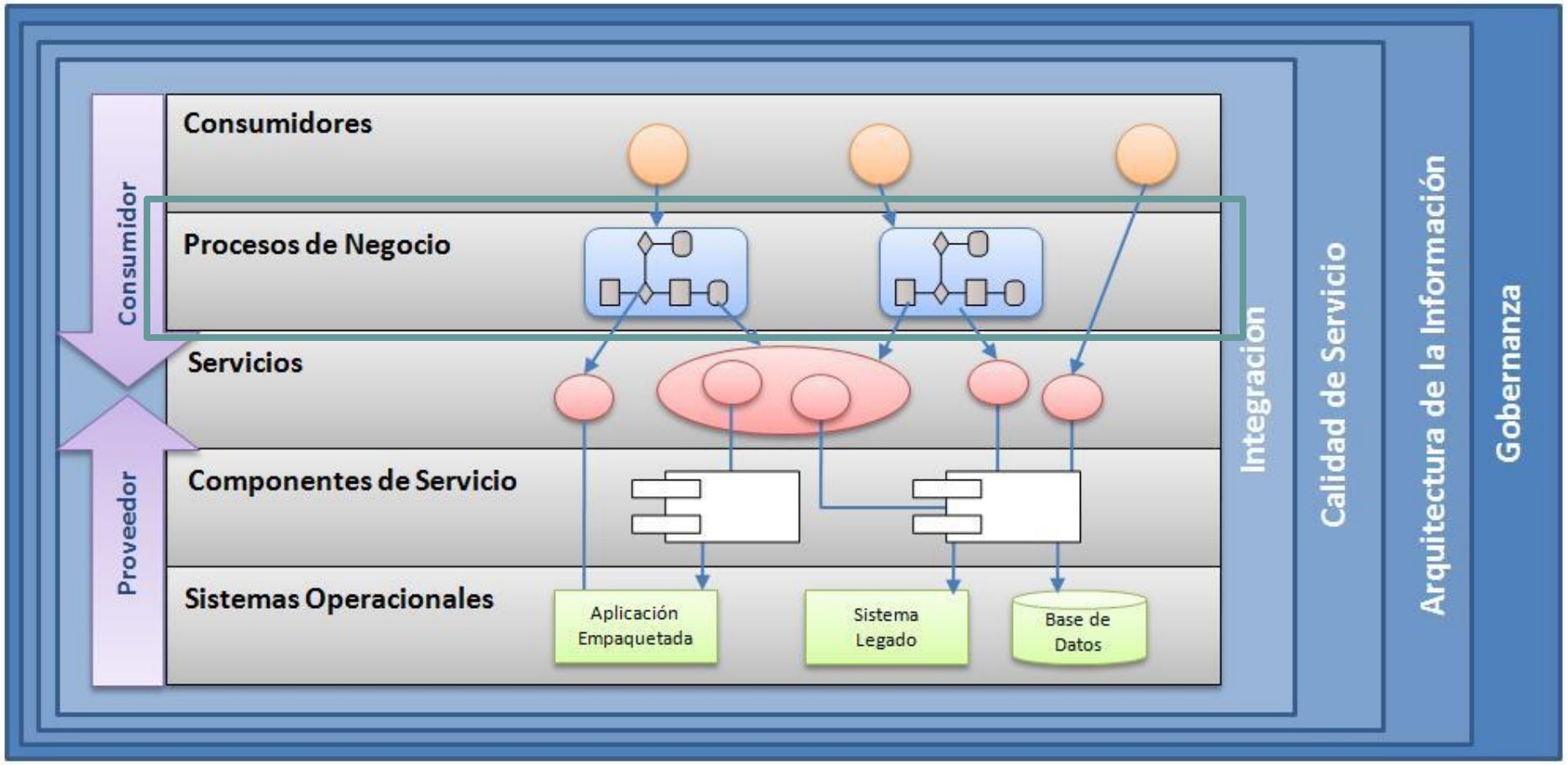
Arquitectura Referencia para SOA

Capa Servicios

- Consiste de los servicios definidos en una SOA
- Un servicio es una descripción abstracta de una o más funcionalidades de negocio
- Esta descripción provee a los consumidores la información necesaria para invocar la funcionalidad de negocio expuesta por un proveedor de servicios

Arquitectura Referencia para SOA

Capas de una SOA



Arquitectura Referencia para SOA

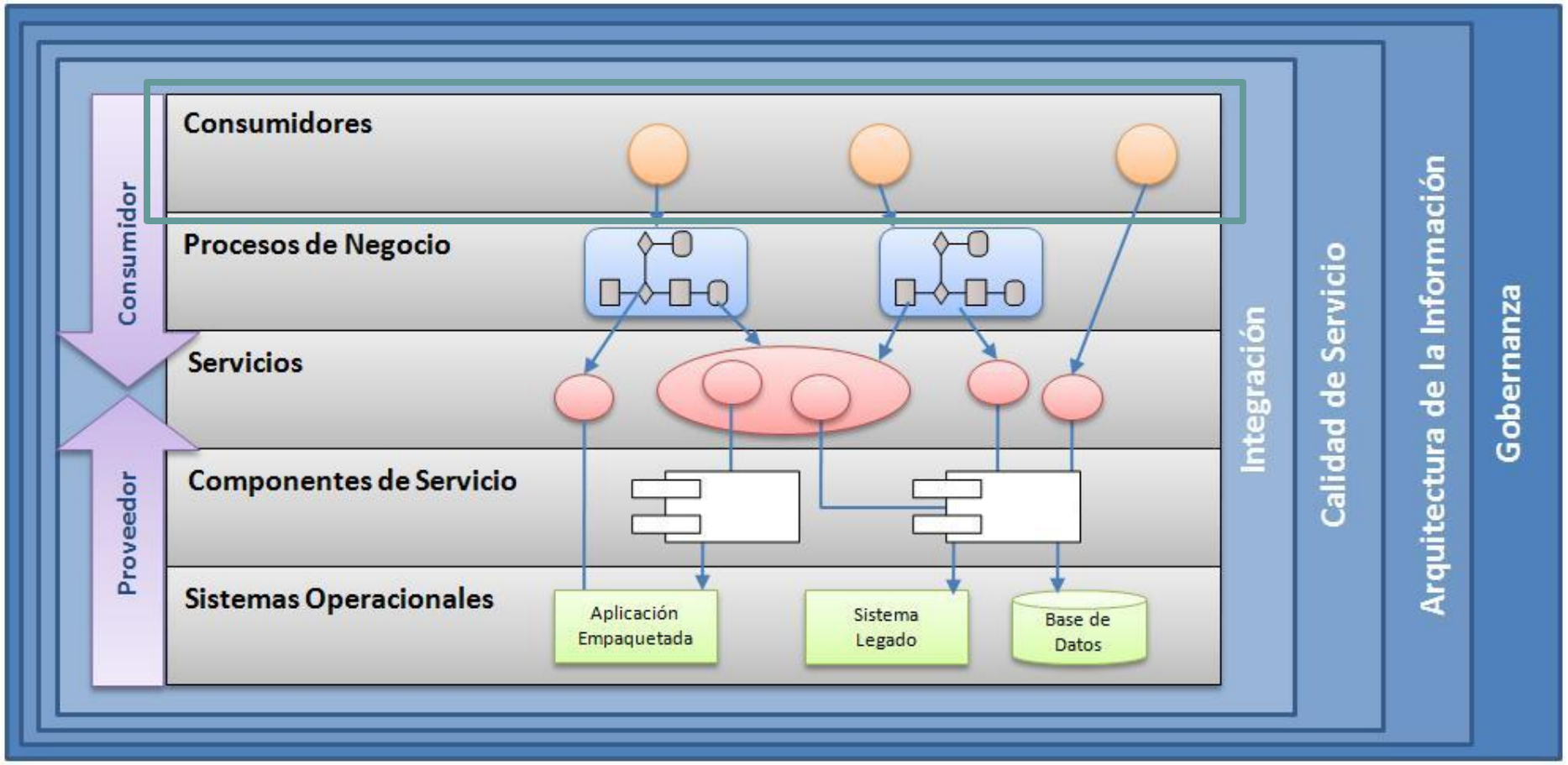
Capa: Procesos de Negocio

- En esta capa las organizaciones combinan los servicios expuestos en la Capa de Servicios para crear composiciones de servicios que dan soporte a sus procesos de negocio

- Esto redunda, generalmente, en una mayor reutilización y agilidad de negocio

Arquitectura Referencia para SOA

Capas de una SOA



Arquitectura Referencia para SOA

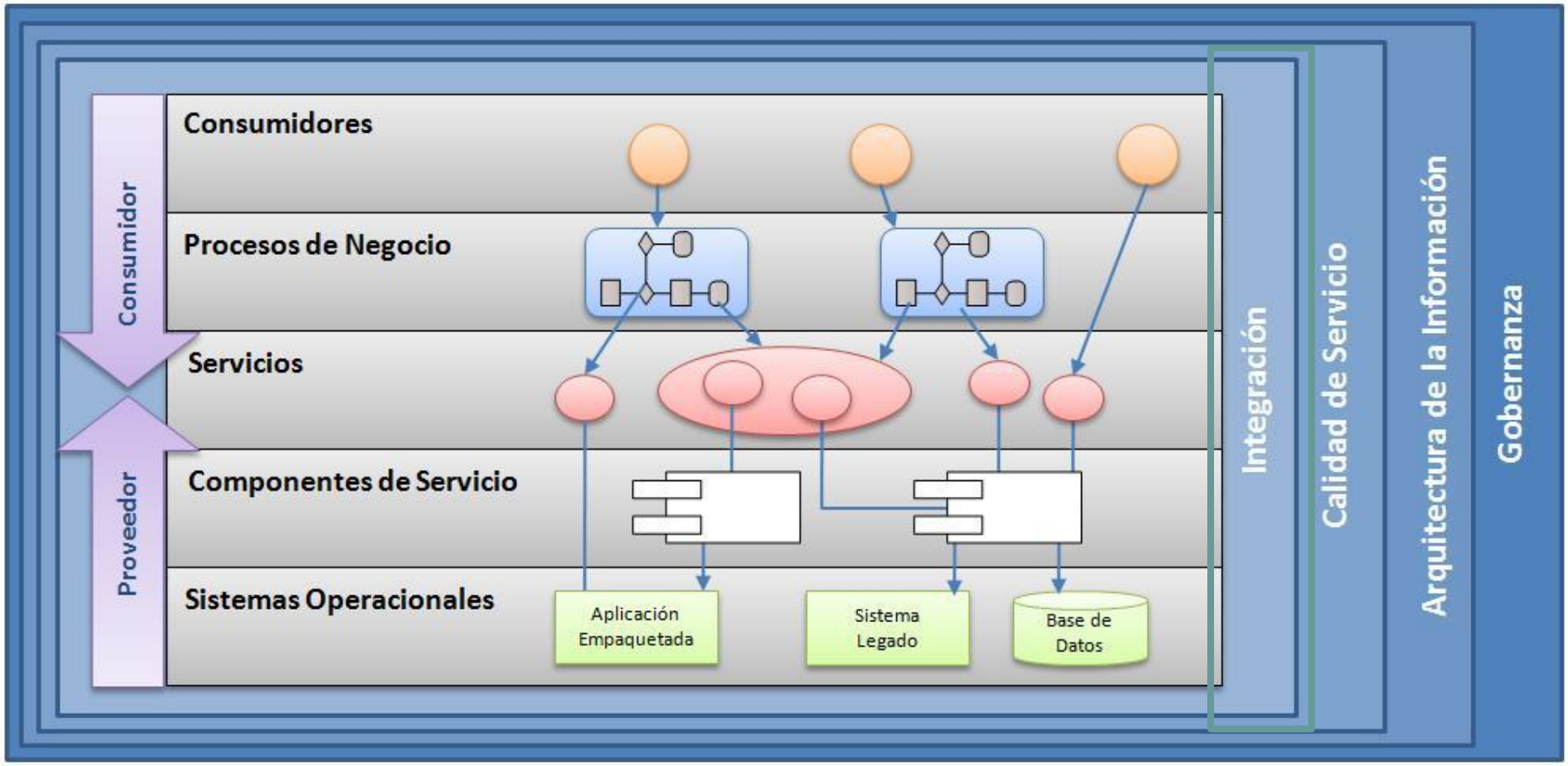
Capa: Consumidores

- Esta capa maneja la interacción con el usuario u otros sistemas en la SOA

- A través de esta capa, las organizaciones pueden ofrecer funcionalidades y datos a aplicaciones o usuarios, de acuerdo a requerimientos o preferencias específicas

Arquitectura Referencia para SOA

Capas de una SOA



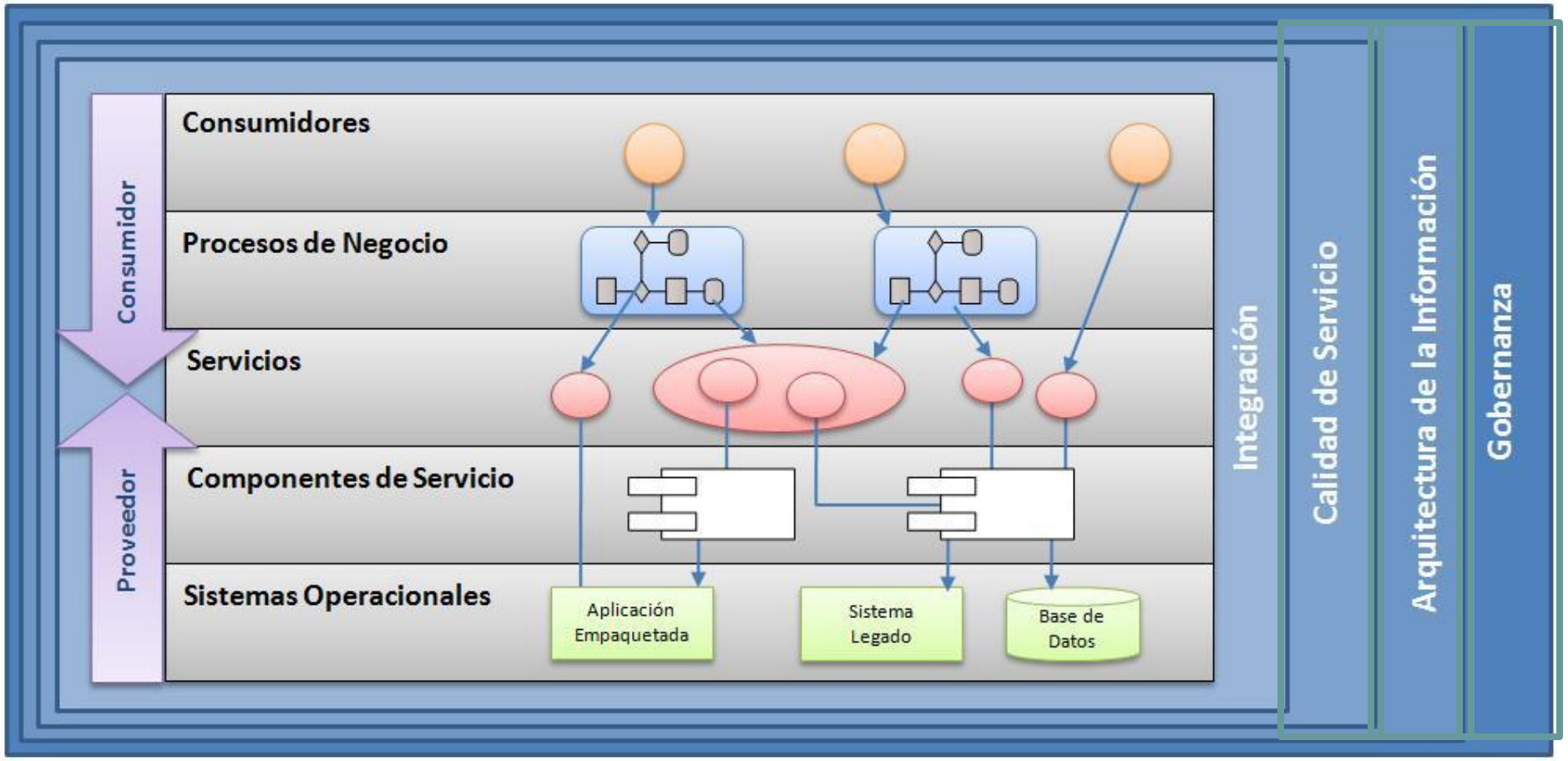
Arquitectura Referencia para SOA

Capa: Integración

- Provee la capacidad de mediar, transformar, realizar ruteos y transportar solicitudes de servicios, desde el consumidor al proveedor.
- A través de un conjunto de capacidades de mediación, esta capa posibilita la integración de los servicios en una SOA
- Generalmente se implementa a través de productos tipo Enterprise Service Bus (ESB)

Arquitectura Referencia para SOA

Capas de una SOA



Middleware para SOA

Web Services

- El término Web Service nace aproximadamente en el año 2000
- Surge como una necesidad de la industria en las áreas:
  - Business to Business (B2B)
  - Enterprise Application Integration (EAI)

Middleware para SOA

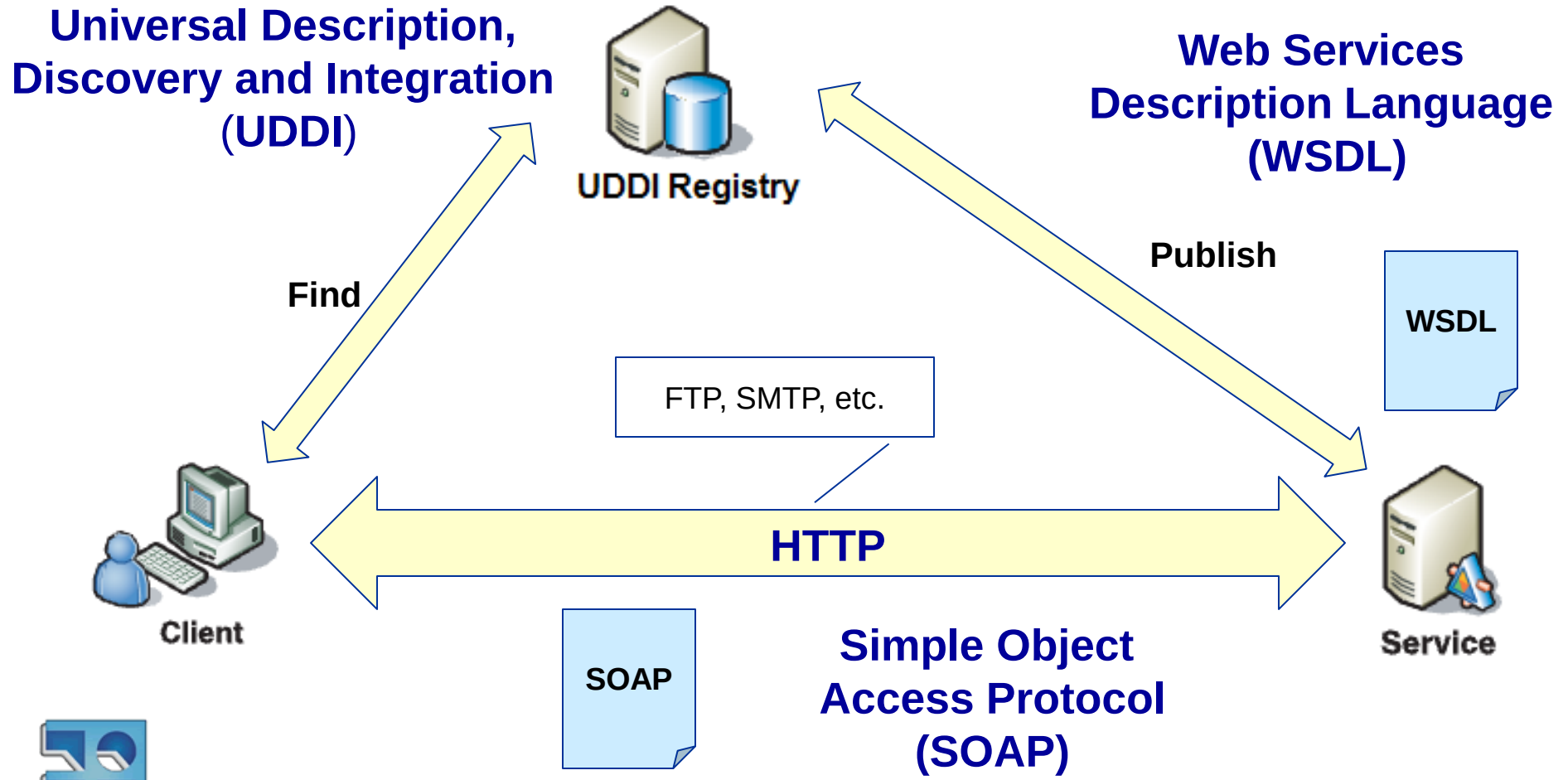
Web Services: Una Definición

- Un Web Service es una aplicación de software identificada por una URI, cuyas interfaces y formas de acceso pueden ser definidas, descriptas y descubiertas como artefactos XML, y soporta la interacción directa con otros componentes de software utilizando mensajes basados en XML, intercambiados a través de protocolos basados en internet

  http://www.w3.org/TR/ws-desc-reqs/#definitions

Middleware para SOA

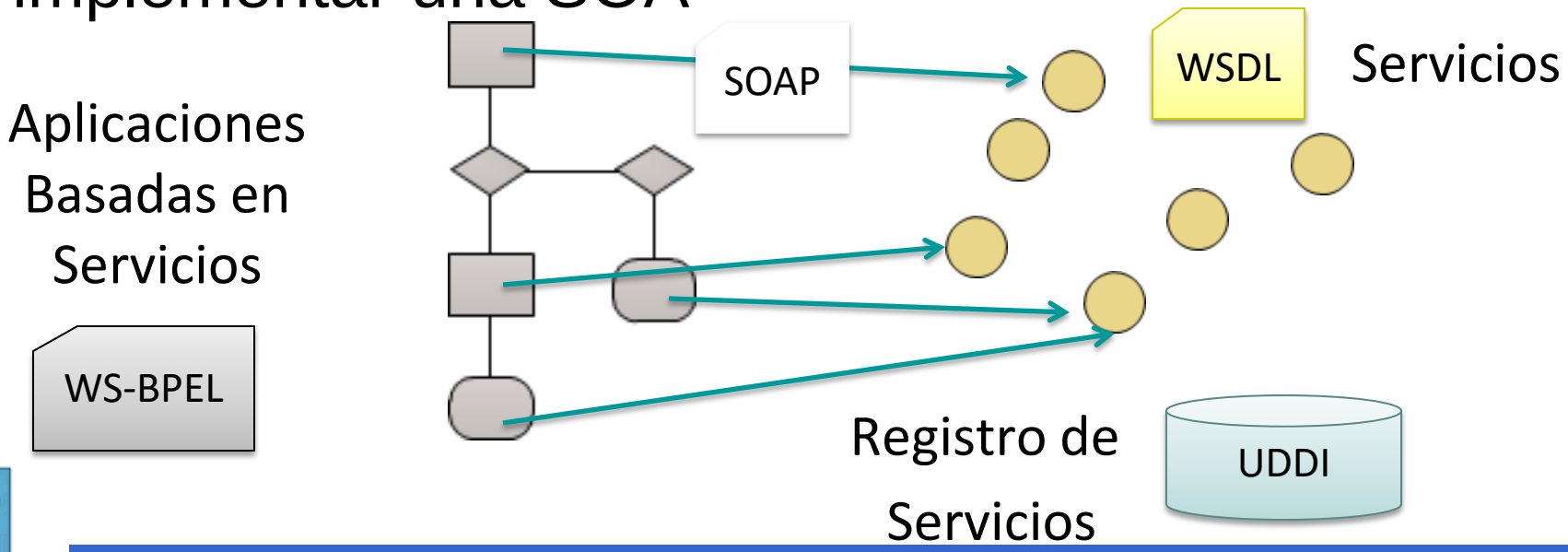
Web Services: Estándares Básicos



Middleware para SOA

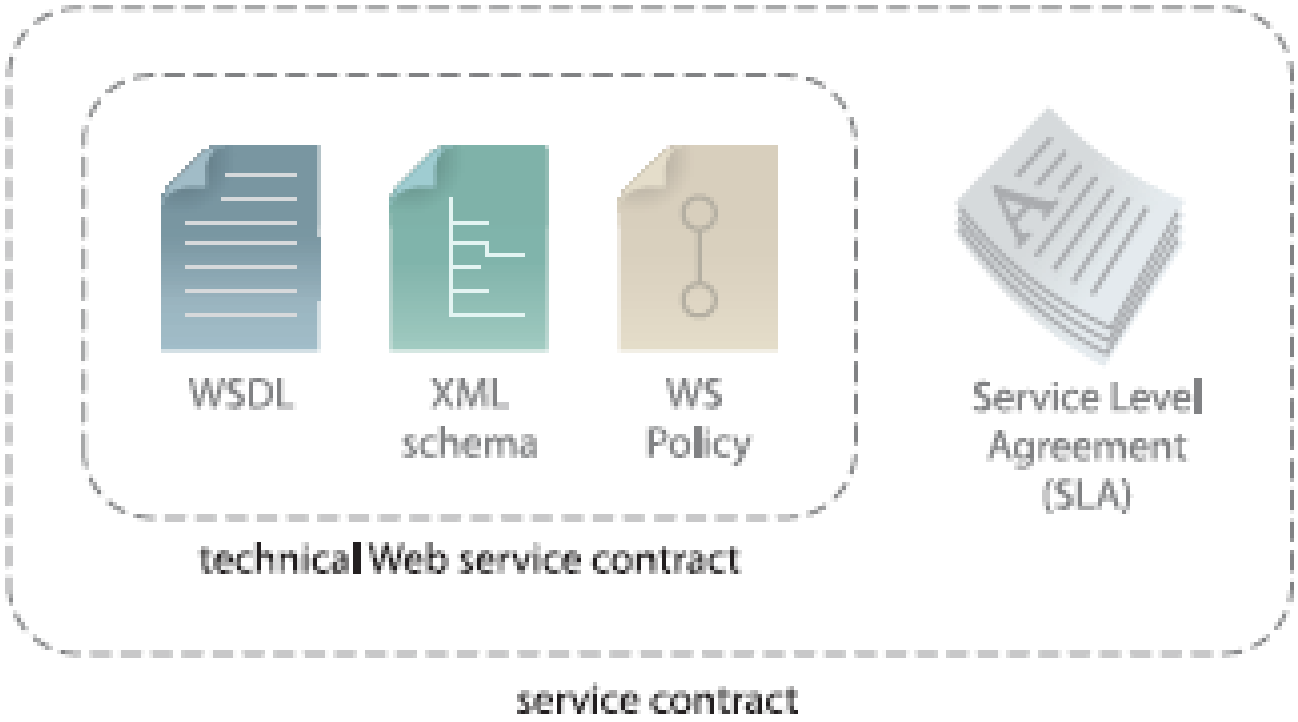
Web Services

- Si bien los principios de SOC no dependen de una tecnología en particular, dadas sus características, los Web Services se han convertido la tecnología preferida para implementar una SOA



Middleware para SOA

Web Services: Contrato de un Servicio



(Erl et al., 2008)

Middleware para SOA

Web Services

- La tecnología de Web Services representa el uso de estándares y tecnologías para la invocación e interoperabilidad
- Los Servicios SOA realizan una actividad clave de un proceso de negocios y se describen como servicios de negocio
- Estos servicios de negocio pueden ser expuestos como Web Services pero
  - Servicio SOA != Web Service

http://j2eearchitec.blogspot.com/2013/01/SOA-Basics-III.html

Middleware para SOA

Web Services: Limitaciones

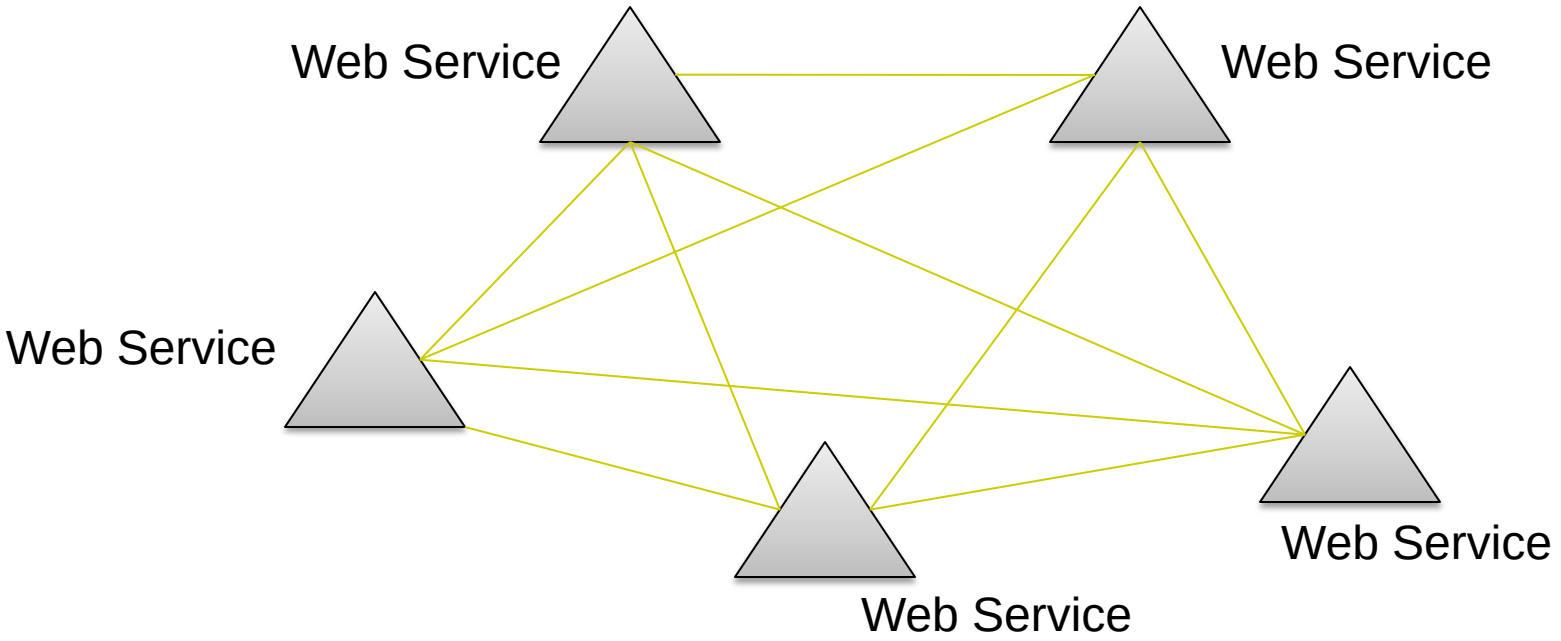
- Si bien los Web Services proveen una base sólida para la implementación de una SOA, existen alguna problemáticas que pueden afectar la flexibilidad y mantenibilidad de arquitecturas a escala empresarial

(Endrei et al., 2004)

Middleware para SOA

Web Services: Limitaciones

- La integración tipo “spaghetti” no favorece la administración y monitoreo de la SOA



(Endrei et al., 2004)

Middleware para SOA

Web Services: Limitaciones

- Las interacciones punto a punto significan a menudo que cuando la interfaz del servicio cambia, los consumidores deben ser modificados para contemplar este cambio

- A escala pequeña esto suele no representar un problema, pero en arquitecturas de escala empresarial, esta situación puede requerir cambios a varias aplicaciones

(Endrei et al., 2004)

Middleware para SOA

Web Services: Limitaciones

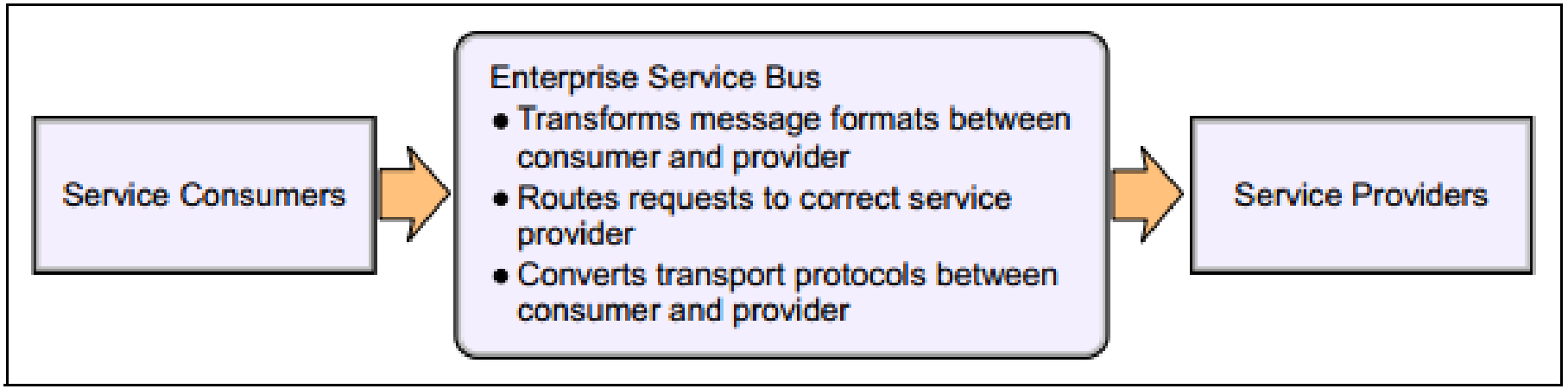
- Estas interacciones punto a punto también requieren que cada consumidor maneje un protocolo adecuado para cada proveedor que necesita utilizar.

- El tener que instalar distintos adaptadores de protocolos en varias aplicaciones cliente agrega costos y complica la mantenibilidad.

(Endrei et al., 2004)

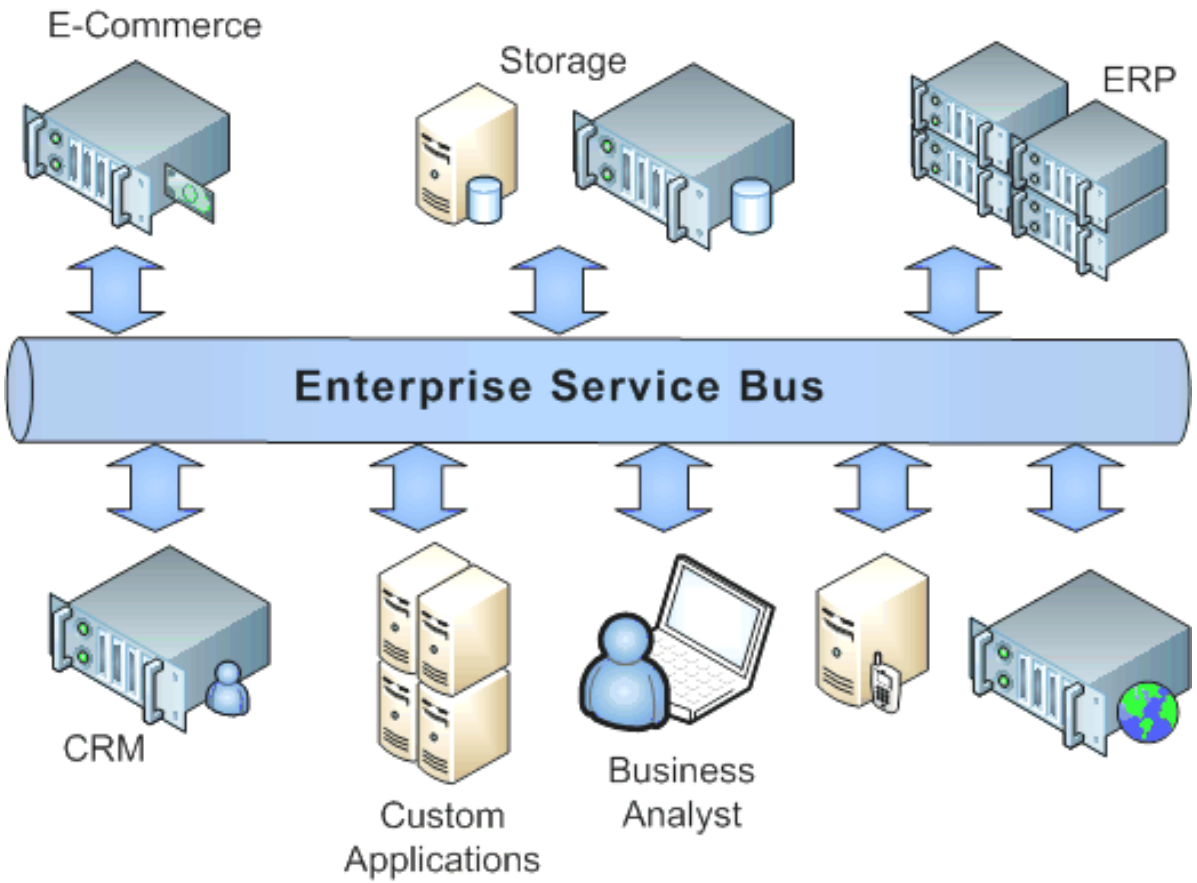
Middleware para SOA

Enterprise Service Bus



(Endrei et al., 2004)

Integraciones NxN



Middleware para SOA

Enterprise Service Bus

- Un ESB es una **plataforma de integración basada en estándares** que combina **mensajería**, **web services**, **transformación de datos** y **ruteo inteligente** para conectar y coordinar de forma **confiable** la interacción de un gran número de **aplicaciones diversas** a través de empresas extendidas (empresas + socios de negocios) con **integridad transaccional**.

(Chappell 2004)

Middleware para SOA

Enterprise Service Bus

- Los ESB proveen varios servicios de infraestructura para implementar una SOA
- Permiten el bajo acoplamiento entre servicios
- Facilitan el aprovechamiento de sistemas legados (adaptadores y conectividad)
- Facilitan la administración de la SOA (capacidades de monitoreo y administración)

Middleware para SOA

Enterprise Service Bus

- Los ESB proveen una infraestructura de integración administrable y distribuida consistente con los principios de SOA
- Permiten la interacción entre servicios heterogéneos brindando virtualización de ubicación, proveedor e interfaz
- Brinda soporte tanto para Web Services como para tecnologías y estándares de comunicación tradicionales

Resumen

Arquitecturas Orientadas a Servicios

- SOC paradigma de computación que utiliza servicios como elementos fundamentales
- SOA forma lógica de diseñar un sistema de Software para dar soporte a SOC
- SOC y SOA brindan varios beneficios pero introduce también desafíos
- Arquitectura Referencia SOA
  - Capas funcionales y no funcionales

Resumen

Arquitecturas Orientadas a Servicios

- Web Services
  - Dadas sus características constituyen la tecnología preferida para implementar una SOA (SOA != WS)
- Enterprise Service Bus
  - Provee distintas capacidades que facilitan la implementación de una SOA (SOA != ESB)

SOA y Otras tecnologías

Business Process Management

- Un Proceso de Negocio consiste de un conjunto de actividades que se realizan de forma coordinada para lograr un objetivo de negocio

- BPM incluye conceptos, métodos y técnicas para dar soporte al diseño, administración, configuración, ejecución y análisis de los procesos de negocio

(Behara, 2006)

Business Process Management (BPM)

- El objetivo de BPM es administrar el ciclo de vida de un proceso desde sus metas de negocio hasta su:
  - definición
  - despliegue
  - ejecución
  - medición
  - análisis
  - modificación
  - re-despliegue

(Behara, 2006)

SOA y Otras tecnologías

Business Process Management

- Los BPM Systems (BPMS) son un conjunto integrado de herramientas, que permiten coordinar la ejecución de procesos de negocio dando soporte al:
  - modelado
  - diseño
  - medición
  - monitoreo
  - análisis
  - optimización
  - mejora continua

(Behara, 2006)

SOA y Otras tecnologías

Business Process Management

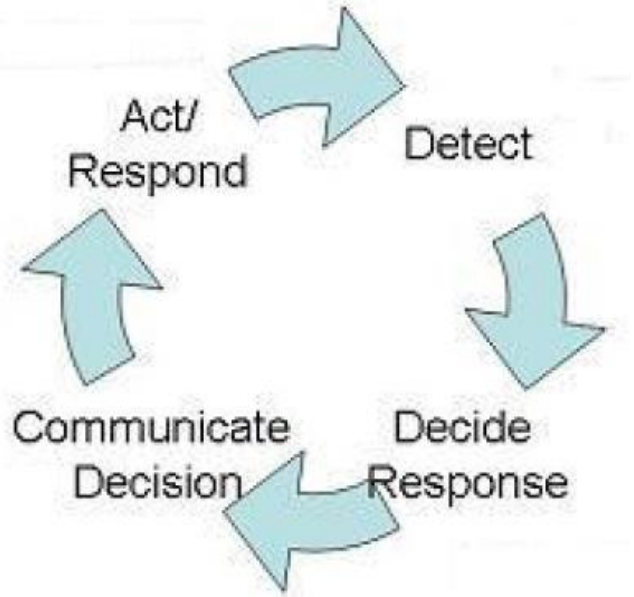
- Si bien BPM y SOA pueden existir de forma independiente, la combinación de ambos enfoques trae mayores beneficios
- Una SOA puede aprovechar BPM para administrar el ciclo de vida de los procesos
- La capa de servicios provee una plataforma ideal para construir procesos de negocio

(Behara, 2006)

SOA y Otras tecnologías

Reglas de Negocio

- La agilidad de negocio se define típicamente como la habilidad de detectar una necesidad de cambio, analizar las posibles respuestas a ese cambio, decidir cuál es la respuesta apropiada, comunicarla



(Taylor, 2011)

SOA y Otras tecnologías

Reglas de Negocio

- SOA favorece la agilidad de negocio encapsulando lógica de negocio en servicios
- Puede además existir la necesidad de cambiar la lógica de negocio implementada en los servicios
- La lógica de negocio de un servicio puede necesitar variar o adaptarse a cambios más frecuentemente
  - La lógica de un servicio que tiene en cuenta el entorno de negocio actual podría variar de forma diaria

(Taylor, 2011)

SOA y Otras tecnologías

Reglas de Negocio

- La lógica de negocio de un servicio puede ser compleja y “pesada”
  - Ante la necesidad de un cambio, se puede dificultar encontrar la parte del servicio a cambiar
- La lógica de negocio de un servicio puede ser difícil de implementar para usuarios técnicos (que no conocen el negocio)
- También existe la necesidad de cumplir normativas, lo que puede presentar complicaciones si el servicio cambia o las normativas cambian

(Taylor, 2011)

SOA y Otras tecnologías

Reglas de Negocio

- En todos estos casos existe la necesidad de que los servicios se adapten al cambio de forma más efectiva que los programas (escritos y compilados con herramientas tradicionales)

- Si esto no ocurre, la agilidad de negocio global puede verse comprometida

(Taylor, 2011)

SOA y Otras tecnologías

Reglas de Negocio

- Las reglas de negocio son abstracciones de las políticas y prácticas de una empresa
- El enfoque de reglas de negocio es una metodología donde las reglas son “utilizadas” por los servicios, pero no están “incluidas” en los servicios

- Estas reglas se especifican en un lenguaje que puede entender tanto especialistas de negocio como de TI
- Las reglas son utilizadas por una organización para llevar a cabo su negocio

(Taylor, 2011)

SOA y Otras tecnologías

Reglas de Negocio

- Una regla de negocio es una declaración que abstrae lógica de decisión y se maneja de forma separada del código

- Las reglas de negocio tiene la forma IF – THEN
  - La condición del IF se evalúa contra un hecho (fact)
  - Si se cumple la condición, se ejecutan la acción del THEN

- Los hechos son información del mundo real

(Taylor, 2011)

- Ejemplo de Regla en Drools

```
rule "check balance"
    when
        Account( balance < 100 ) // condition
    then
        System.out.println("Account balance is " + "less than 100"); // consequence
end
```

SOA y Otras tecnologías

Reglas de Negocio

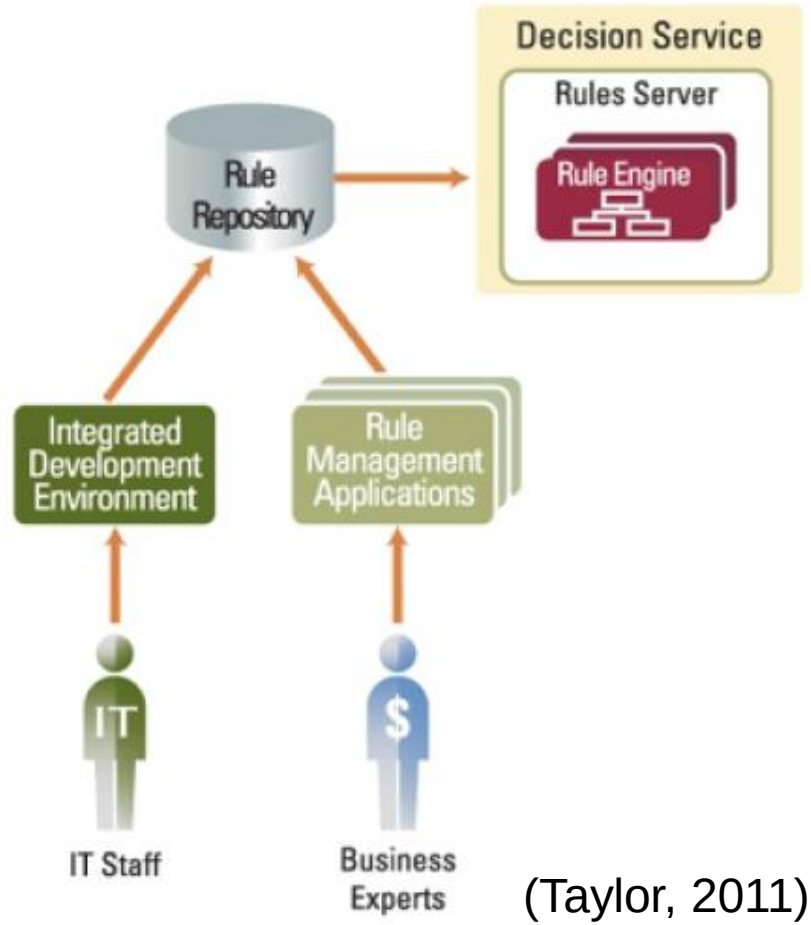
- Un Business Rule Engine (BRE) utiliza un conjunto de reglas de negocio para analizar hechos
- Las reglas pueden ser administradas por un Business Rules Management System (BRMS)
- Un BRMS es un conjunto de herramientas que en general brinda soporte al modelado, administración, almacenamiento y ejecución de reglas de negocio

(Taylor, 2011)

SOA y Otras tecnologías

Reglas de Negocio

- ¿Cuándo usar un BRMS?
  - Reglas de negocio dinámicas
  - Reglas de negocio compartidas por aplicaciones
  - …



(Taylor, 2011)

SOA y Otras tecnologías

Reglas de Negocio

- Las principales alternativas para posicionar las reglas de negocio en una SOA son utilizarlas para:
  - Implementar servicios de negocio
  - Implementar servicios de decisión

(Taylor, 2011)

SOA y Otras tecnologías

Reglas de Negocio

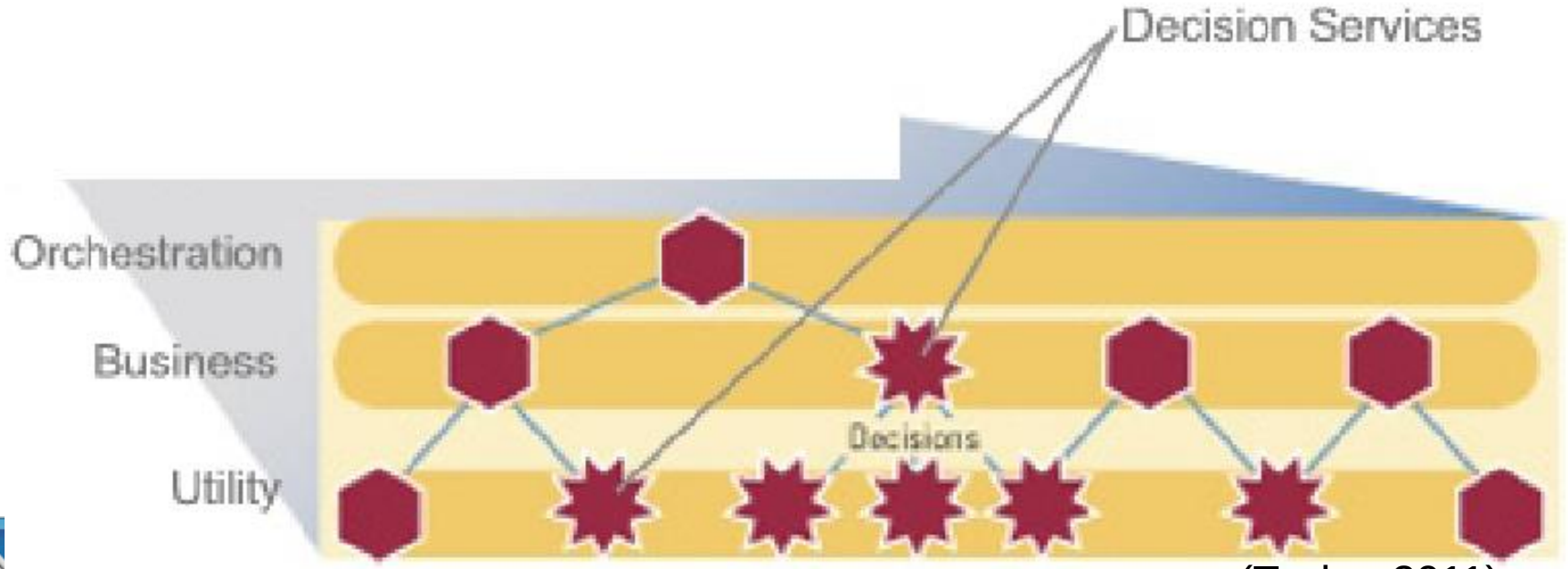
- Implementar servicios de negocio
  - En este caso la implementación de cada servicio interactúa directamente con la plataforma de reglas de negocio de acuerdo a sus necesidades

(Taylor, 2011)

SOA y Otras tecnologías

Reglas de Negocio

- Implementar servicios de decisión
  - Estos servicios puede posicionares como recursos genéricos o específicos de negocio que brindan una respuesta a una pregunta específica



(Taylor, 2011)

SOA y Otras tecnologías

Reglas de Negocio

- En una SOA las Reglas de Negocio permiten incrementar la agilidad de negocio

- BPM vs Reglas de Negocio
  - Se pueden ver como enfoques complementarios
  - En general las reglas de negocio no apuntan a dar soporte a procesos de larga duración
  - En general, las opciones de decisión disponible en BPM son más simples

(Taylor, 2011)

SOA y Otras tecnologías

Procesamiento de Eventos Complejos (CEP)

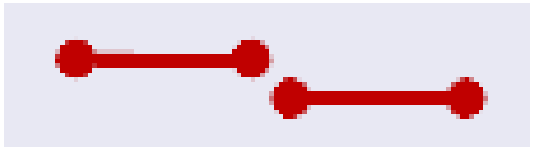
- Un evento se define en general como “algo que ocurre o es percibido como que ocurre”
- En el contexto de un sistema de información, un evento es un registro de un hecho que ocurre en determinado sistema o dominio
- Ejemplos:
  - Inicio de una solicitud de crédito
  - Aprobación / rechazo de una solicitud
- Pueden tener atributos asociados
  - Ej: Timestamp

http://www.complexevents.com/wp-content/uploads/2011/08/EPTS_Event_Processing_Glossary_v2.pdf

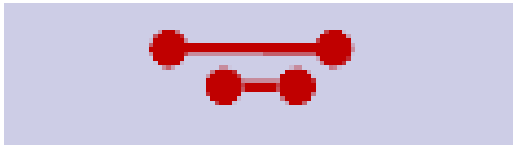
SOA y Otras tecnologías

Procesamiento de Eventos Complejos (CEP)

- Eventos pueden ocurrir en un instante o durante un intervalo de tiempo (tienen una duración asociada)
- Consecutivos
  - B comienza una vez que finalizó A



- Concurrentes
  - B ocurre mientras está ocurriendo A



http://www.jboss.org/drools/drools-fusion.html

SOA y Otras tecnologías

Procesamiento de Eventos Complejos (CEP)

- Un evento complejo es un evento que representa, o enmarca, un conjunto de eventos más simples (consecutivos, concurrentes, etc)

- Se llama procesar eventos complejos a la realización de operaciones como creación, lectura, transformación y desechado sobre instancias de eventos complejos

http://www.complexevents.com/wp-content/uploads/2011/08/EPTS_Event_Processing_Glossary_v2.pdf

SOA y Otras tecnologías

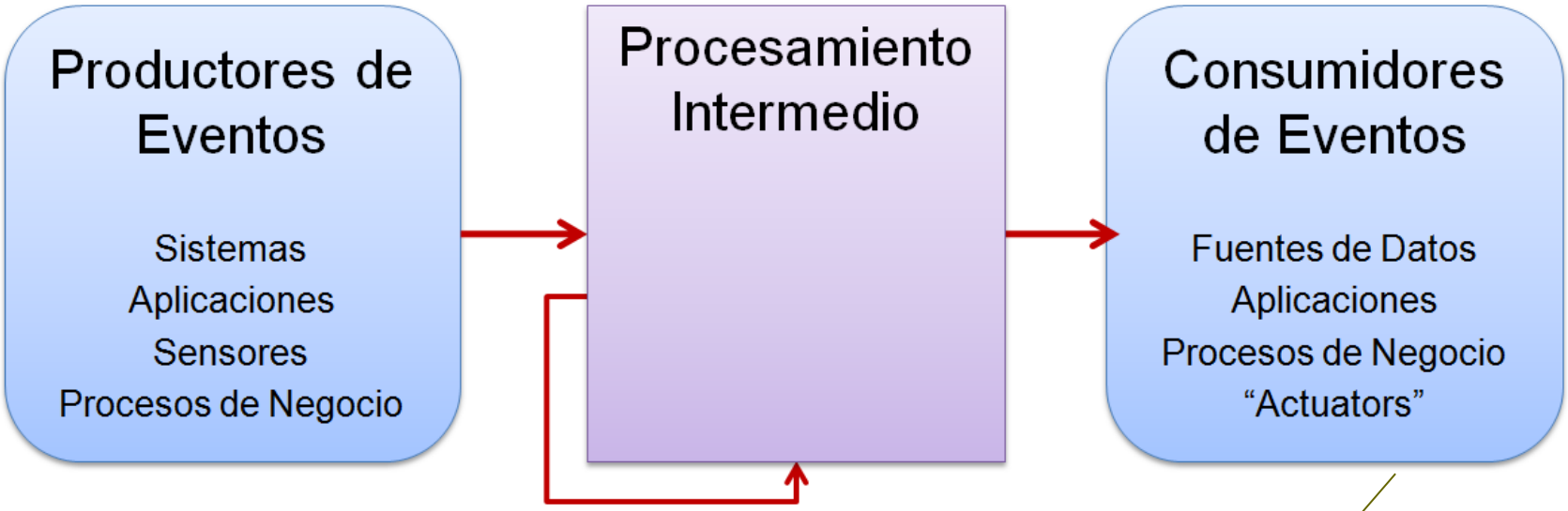
Procesamiento de Eventos Complejos (CEP)

- En un ambiente distribuido, los eventos pueden originarse en diferentes sistemas
- CEP permite reunir esta información y aplicar reglas pre-definidas (patrones) en tiempo real
- Es posible detectar situaciones que de otra forma podrían pasar desapercibidas

(Etzion & Niblett, 2010)

SOA y Otras tecnologías

Procesamiento de Eventos Complejos (CEP)



Productores y consumidores son independientes, uno no sabe nada del otro

(Etzion & Niblett, 2010)

SOA y Otras tecnologías

Procesamiento de Eventos Complejos (CEP)

- Ejemplo: *detección de fraudes*
  - Proceso de negocio: *Solicitudes de Crédito*
  - Eventos: *solicitudes realizadas por la misma persona en 1 hora*
  - Regla: *si la cantidad de solicitudes en 1 hora supera cierta cantidad bloquear la cuenta y pasarla a estudio*

SOA y Otras tecnologías

Procesamiento de Eventos Complejos (CEP)

```
rule "Sound the alarm"
when
    $f : FireDetected( )
    not( SprinklerActivated( this after[0s,10s] $f ) )
then
    // sound the alarm
end
```

```
rule "Sound the alarm"
when
    $f : FireDetected( )
    not( SprinklerActivated( this after[0s,10s] $f ) )
then
    insert (new FireNotHandled());
end
```

SOA y Otras tecnologías

Procesamiento de Eventos Complejos (CEP)

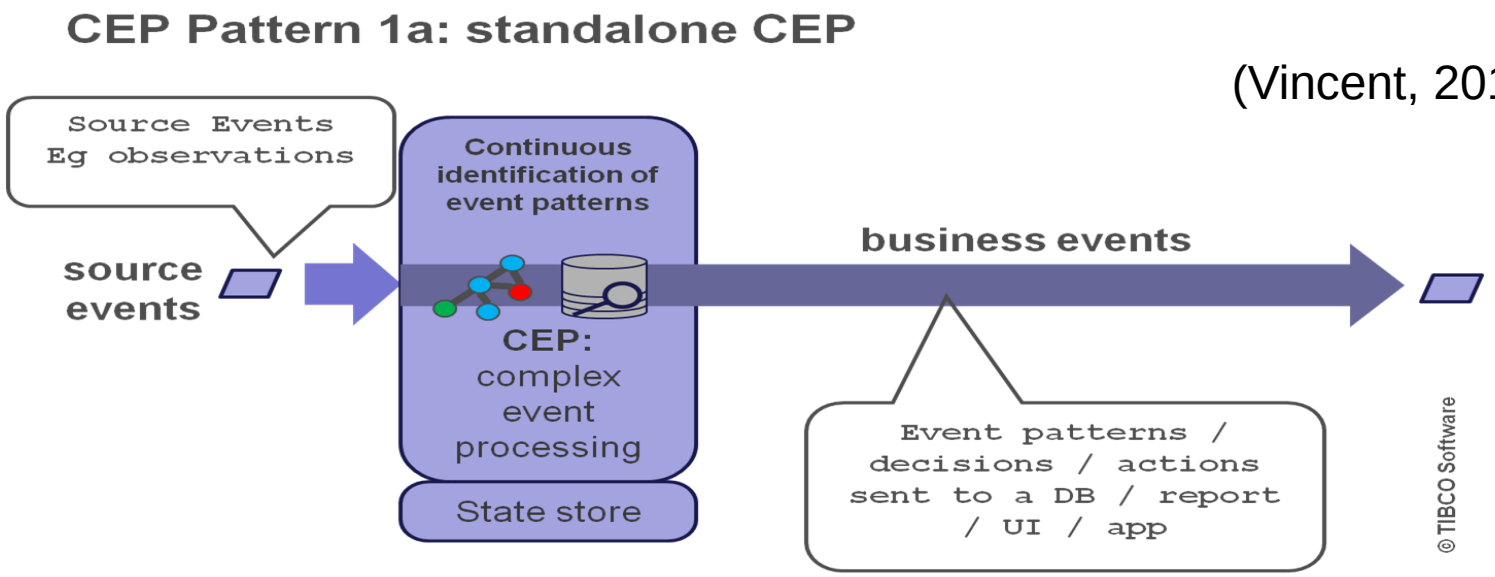
- Existen algunos patrones de cómo aprovechar CEP en contextos BPM y SOA:
  - CEP Independiente
  - CEP Enriqueciendo Procesos de Negocio (BP) y Servicios de Negocio (BS)
  - CEP Monitoreando BPs y BSs
  - Decisiones Basadas en CEP para BPs y BS.
  - Control dinámico de BPs y BSs
  - Embebiendo BPs y BSs en CEP

(Vincent, 2010)

SOA y Otras tecnologías

Patrones CEP: CEP Indenpendiente

- Se utiliza un motor CEP para producir eventos de negocio a partir de eventos provenientes de distintas fuentes, que no necesariamente tienen un sentido de negocio.

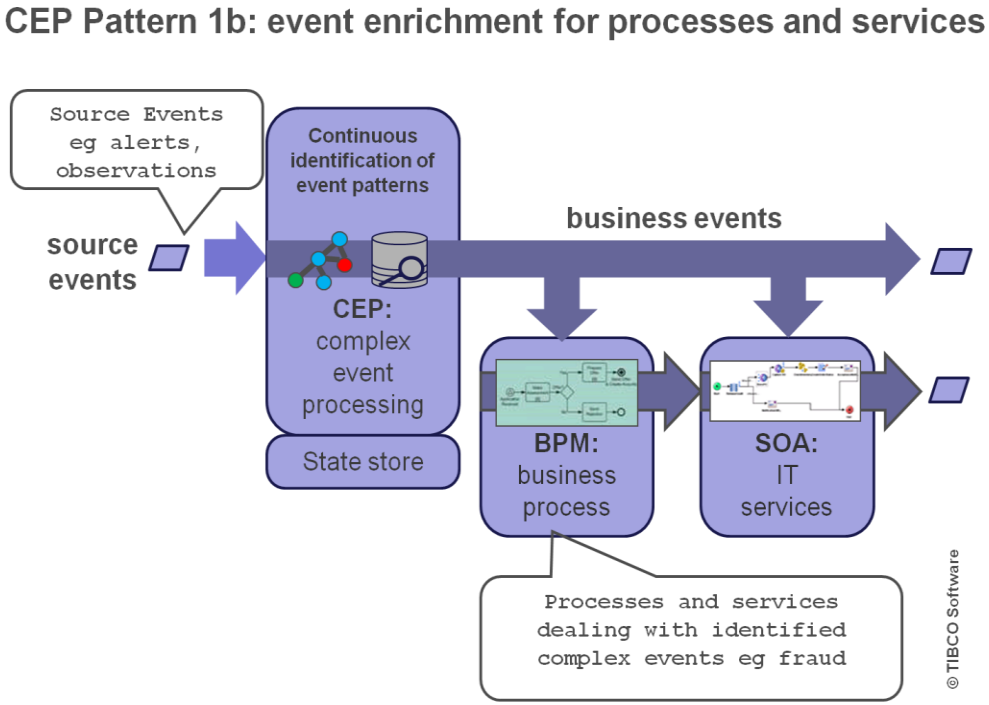


(Vincent, 2010)

SOA y Otras tecnologías

Patrones CEP: CEP enriqueciendo Procesos y Servicios de Negocio

- Constituye la vista convencional de CEP, en la cual se detectan eventos complejos de interés y útiles para procesos y servicios.



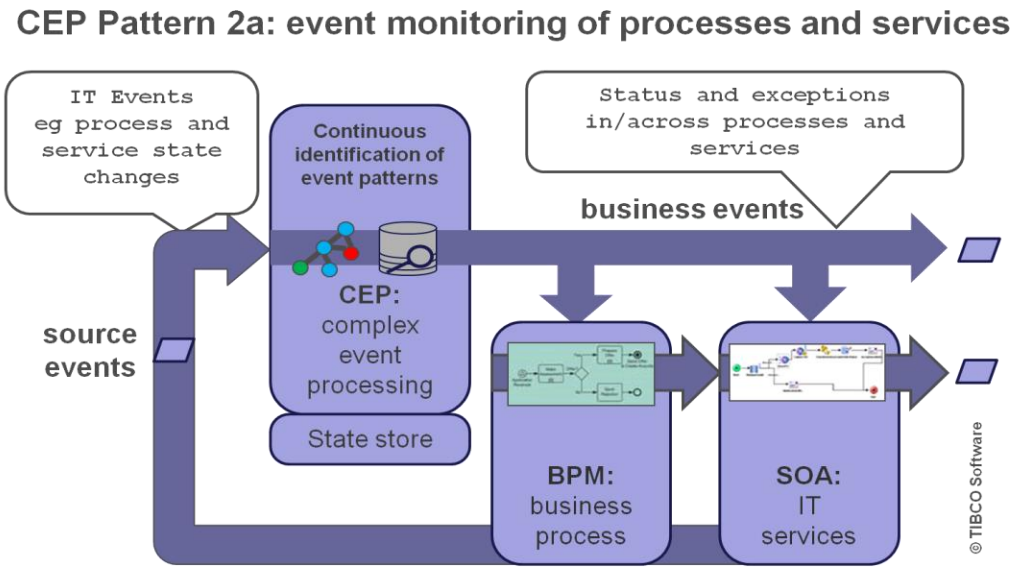
(Vincent, 2010)

SOA y Otras tecnologías

Patrones CEP: CEP monitoreando Procesos y Servicios de Negocio

- En este patrón los eventos fuente provienen de los propios procesos o servicios.
- De esta forma, es posible monitorearlos para detectar excepciones o discrepancias.

(Vincent, 2010)

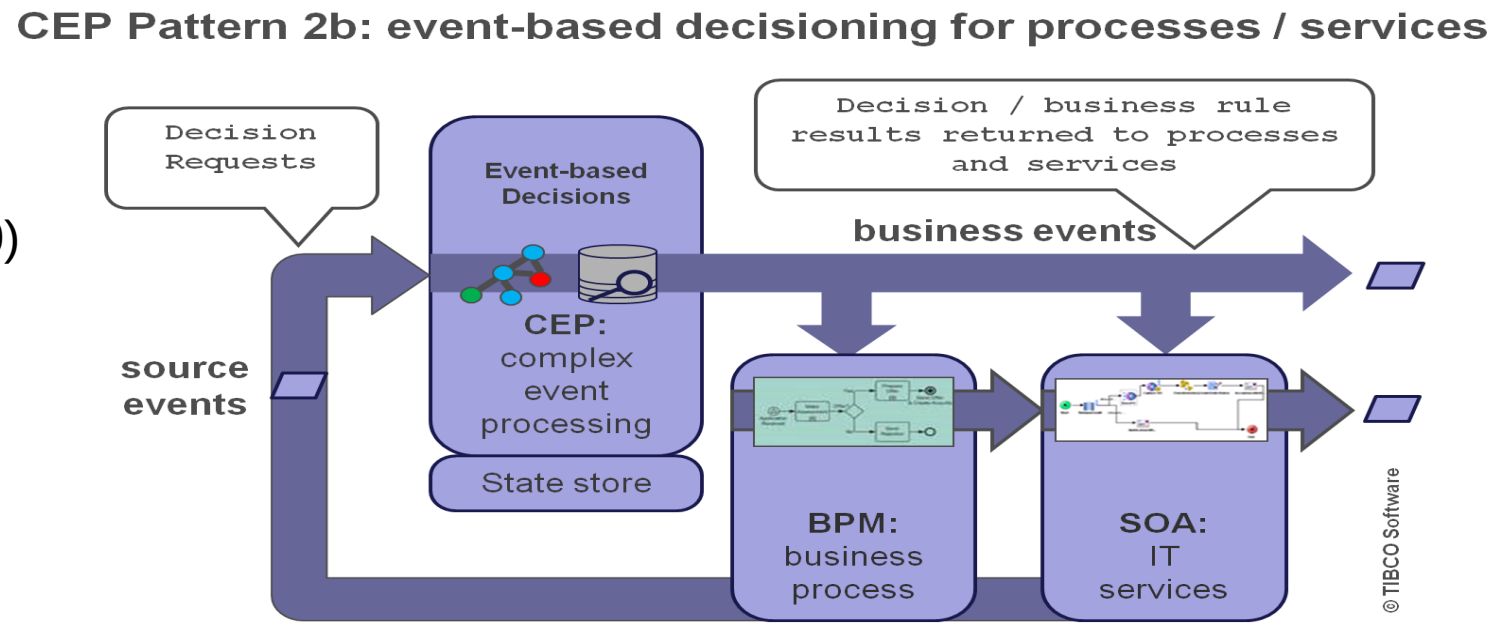


SOA y Otras tecnologías

Patrones CEP: Decisiones basadas en CEP

- Se utiliza la capa CEP, desde la capa de procesos o servicios, como un componente compartido para la gestión de decisiones.

(Vincent, 2010)



Preguntas

Referencias

- Alluri, R. SOA Adoption Challenges, 2009.
- Arsaniani, A. Service-oriented modeling and architecture, 2004.
- Chappell, David. 2004. Enterprise Service Bus: Theory in Practice. O'Reilly Media.
- Davis, Jeff. 2009. Open Source Soa. 1st ed. Manning Publications.

Referencias

- Di Nitto, C. Ghezzi, A. Metzger, M. Papazoglou, and K. Pohl. A journey to highly dynamic, selfadaptive service-based applications. Automated Software Engineering, vol. 15, no. 3, pp. 313-341, Dec. 2008.
- Endrei, Mark et al. 2004. “IBM Redbooks. Patterns: Service-Oriented Architecture and Web Services.”
- Erl, Thomas. 2008. SOA Principles of Service Design. 1st ed. Prentice Hall.

Referencias

- Erl, Anish Karmarkar, Priscilla Walmsley, Hugo Haas, L. Umit Yalcinalp, Kevin Liu, David Umit Orchard, Andre Tost and James Pasley. Web Service Contract Design and Versioning for SOA. 2009.
- Keen, Martin et al. 2004. “IBM Redbooks. Patterns: Implementing an SOA using an Enterprise Service Bus.”
- Krishna Behara. BPM and SOA: A Strategic Alliance. 2006.

Referencias

- Papazoglou, Michael, and Willem-Jan Heuvel. 2007. “Service oriented architectures: approaches, technologies and research issues.” The VLDB Journal 16:389-415.
- Papazoglou, Paolo Traverso, Schahram Dustdar, Frank Leymann. Service-Oriented Computing: State of the Art and Research Challenges.
- Rademakers, Tijs, and Jos Dirksen. 2008. Open-Source ESBs in Action: Example Impl. in Mule and ServiceMix. 1st ed. Manning Publications.

Referencias

- Vicent. Hpw does CEP fit into BPM abd SOA environments? 2010.
- Weske, Business Process Management: Concepts, Languages, Architectures.2007.