

Introducción al middleware



Arquitectura Orientada a
Servicios



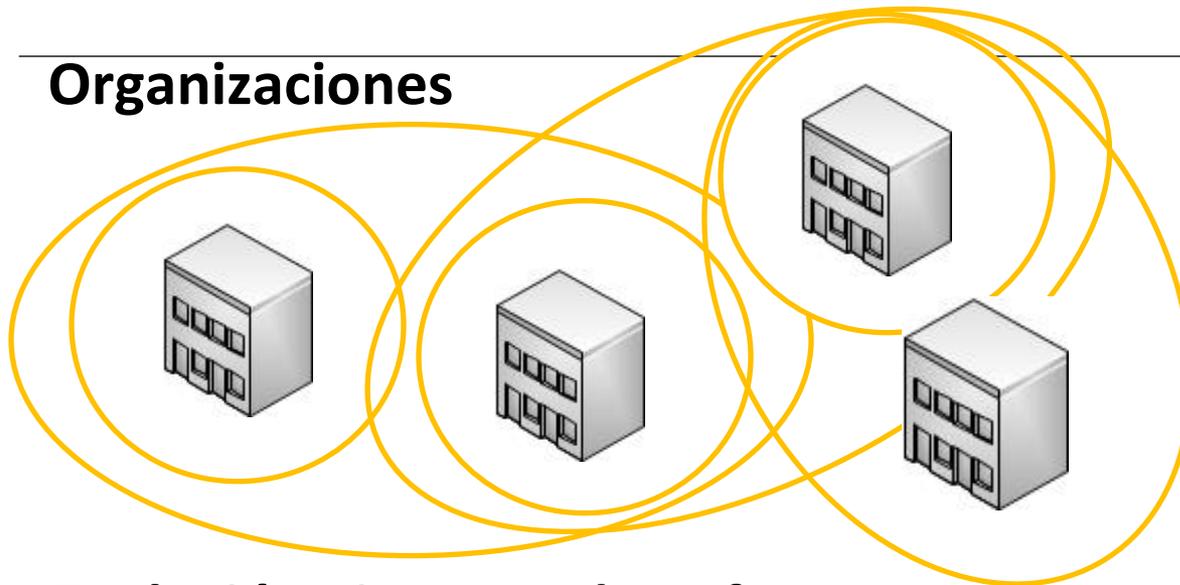
- ❑ Computación Orientada a Servicios
 - Motivación y Principales Conceptos
 - Principios
 - Beneficios y Desafíos
- ❑ Arquitectura Referencia para SOA
- ❑ Middleware para SOA



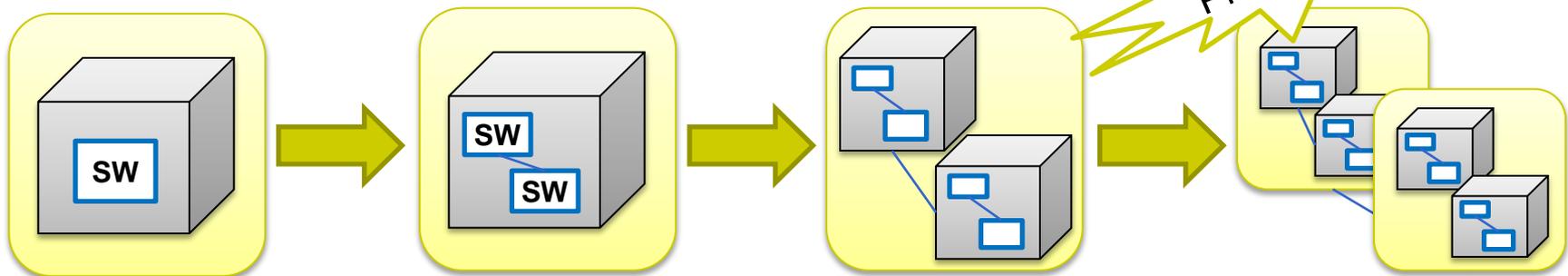
Computación Orientada a Servicios

Motivación

Organizaciones

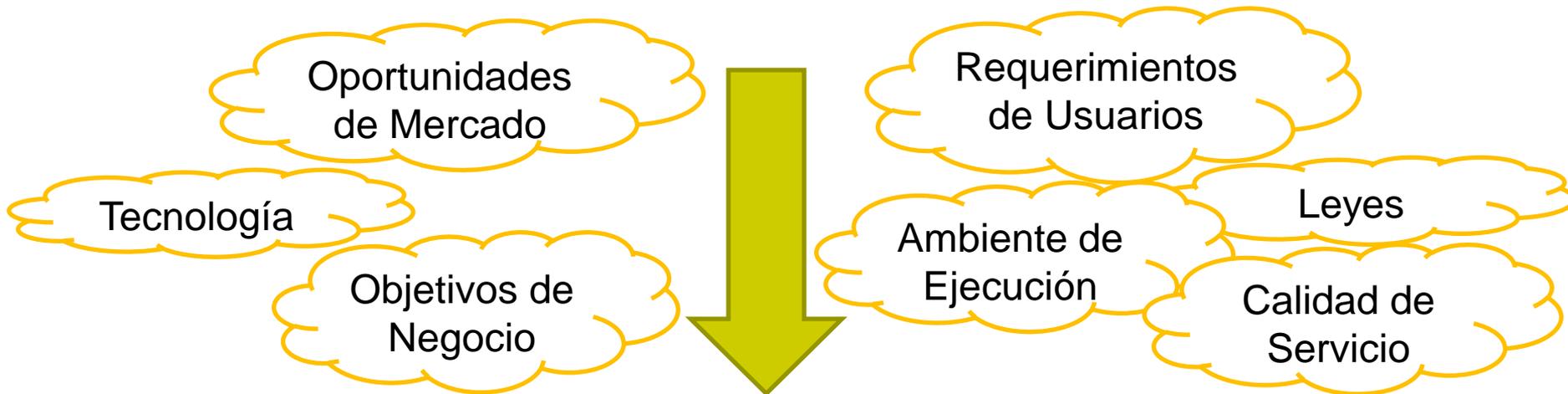


Evolución Sistemas de Software



(Di Nitto et al, 2008)

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



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

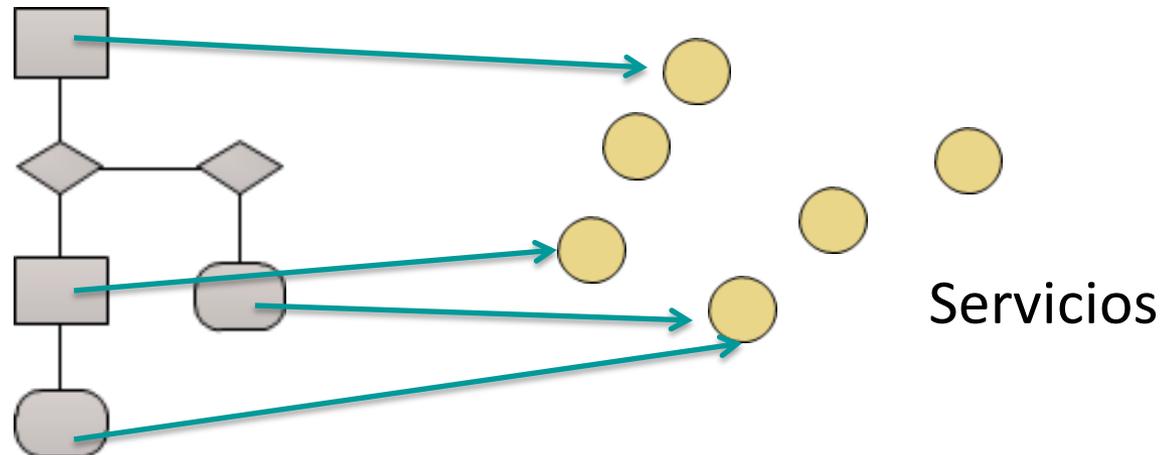


Computación Orientada a Servicios

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)



□ 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

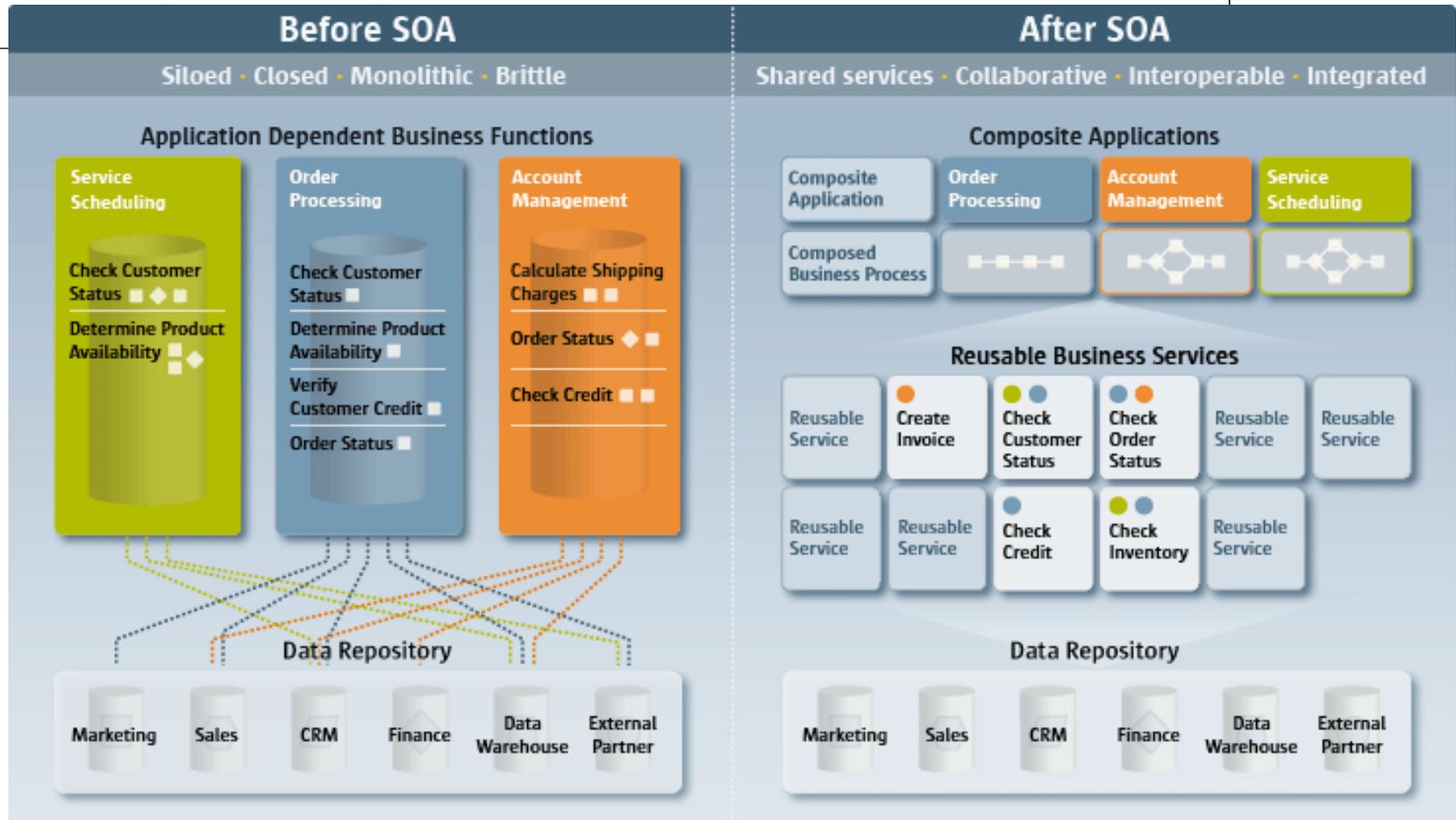
- ❑ La puesta en práctica del paradigma SOC requiere la implementación de Arquitecturas Orientadas a Servicios (SOAs)
- ❑ Una SOA es una forma lógica de diseñar un sistema de software para proveer servicios, a usuarios finales, aplicaciones u otros servicios, a través de interfaces públicas que pueden ser descubiertas.



(Papazoglou and Heuvel 2007)

Computación Orientada a Servicios

Conceptos: Arquitectura Orientada a Servicios

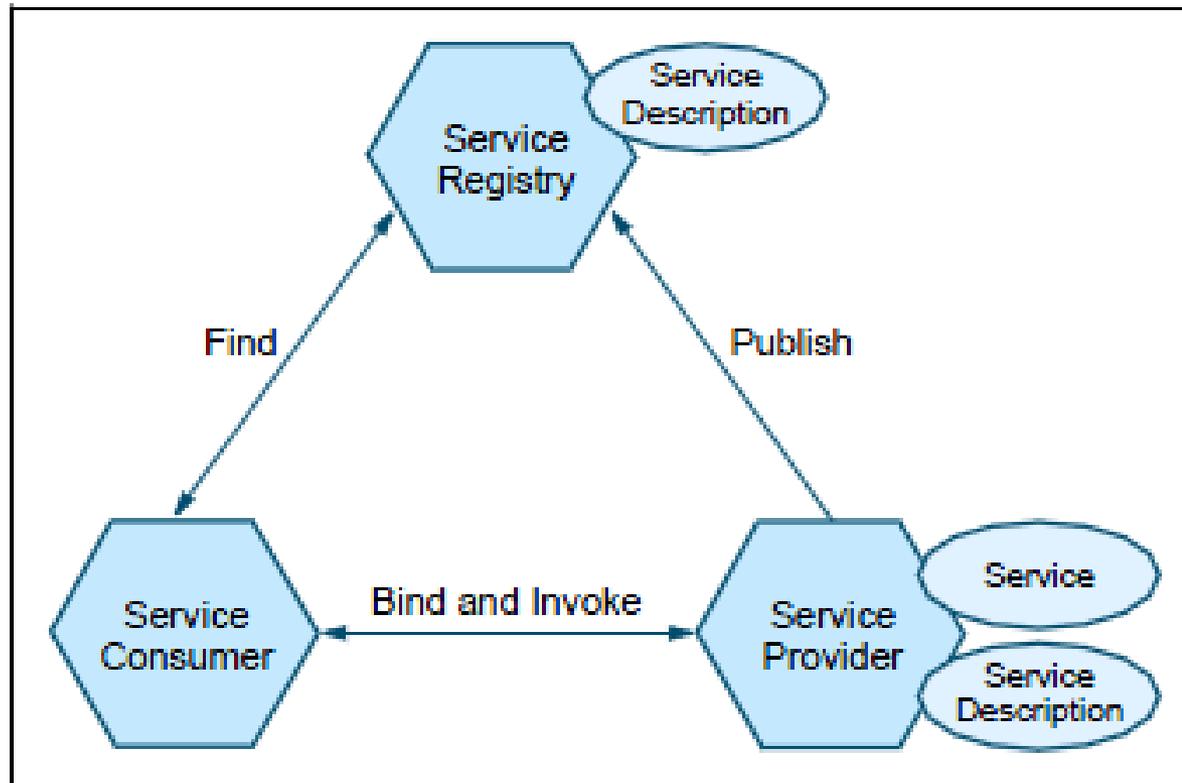


<https://ibimapublishing.com/articles/JSSD/2012/16942>

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

- Una organización puede publicar sus servicios:
 - de forma interna a la organización
 - de forma externa, para interactuar con socios de negocio



Computación Orientada a Servicios

Escenario SOA: Dentro de una Organización

- ❑ Mejorar el soporte a negocios dentro de la empresa
 - P. ej: Fusión de empresas, integración de sistemas pensados de forma independiente
- ❑ Ejemplo: ANTEL
 - ANTEL, ANCEL, ANTEL DATA
 - Tres sistemas de clientes, tres sistemas de lealtad, tres sistemas de stock...
 - A nivel empresa es necesario tener una visión unificada de su gestión y operativa.



Computación Orientada a Servicios

Escenario SOA: Negocios entre Organizaciones

- ❑ Mejorar la eficiencia de nuevos negocios entre empresas.
- ❑ Ejemplo:
 - una agencia de pagos y cobranzas
 - integración con UTE, ANTEL, OSE, etc
 - Banco y verificación de identidad de clientes
 - consultar información de identificación de DNIC online.



Computación Orientada a Servicios

SOA no es aplicable a todos los escenarios

- Ejemplo:
 - diseño de aplicación monolítica

- Todo está bajo control de un único "dueño".



Computación Orientada a Servicios

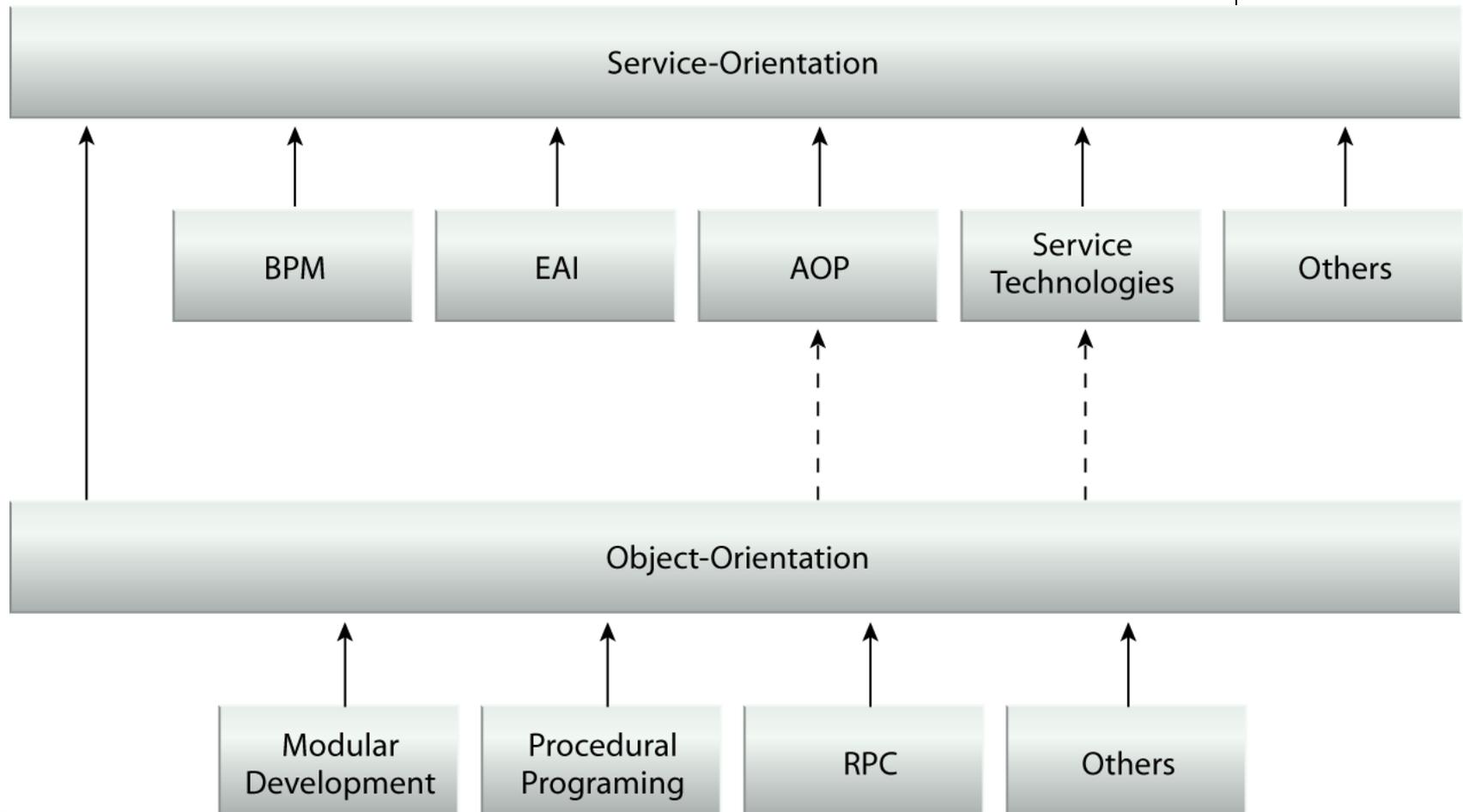
Conceptos: Arquitectura Orientada a Servicios

- ❑ SOA es la evolución de prácticas de desarrollo anteriores para el desarrollo de sistemas distribuidos.
- ❑ SOA aplica buenas prácticas anteriores preservando las características exitosas y alineando principios de diseño que permiten un desarrollo orientado a servicios.
- ❑ SOA provee lineamientos para realizar este tipo de integraciones



Computación Orientada a Servicios

Influencias sobre SOA



(Erl, 2008)



Computación Orientada a Servicios

Influencias sobre SOA

- ❑ Orientación a objetos
 - Reusabilidad, abstracción, encapsulamiento
- ❑ Tecnologías de Servicios
 - Abstracción, bajo acoplamiento y composición de servicios
- ❑ Business Process Management (BPM)
 - Capa de procesos de negocios, composición de servicios y adaptación al cambio
- ❑ Enterprise Application Integration (EAI)
 - Abstracción de servicios, servicios sin estado, bajo acoplamiento y composición de servicios.



Computación Orientada a Servicios

Objetivos de SOA

- Potenciar la **eficiencia**, **agilidad** y **productividad** de una empresa, posicionando a los servicios como principal solución lógica.



Computación Orientada a Servicios

Elementos de una SOA

- ❑ Servicios
- ❑ Composición de servicios
- ❑ Inventario de servicios



□ Los Servicios son:

- entidades de software autónomas, auto-contenidas e independientes de la plataforma
- proveen funcionalidades de negocio
 - Por ejemplo, verificar disponibilidad de crédito.
- tienen una interfaz pública
- pueden ser descubiertos, invocados y combinados de forma dinámica



(Papazoglou and Heuvel 2007)

□ Composición de Servicios:

- Es una agregación coordinada de varios servicios.
- La automatización de un proceso de negocio puede ser implementada como una composición de servicios.
- La aplicación de los principios de SOC conlleva la creación de servicios con capacidades agnósticas a los procesos de negocios.
- Esta capacidad agnóstica es la que les permite participar en múltiples composiciones.

(Erl, 2008)

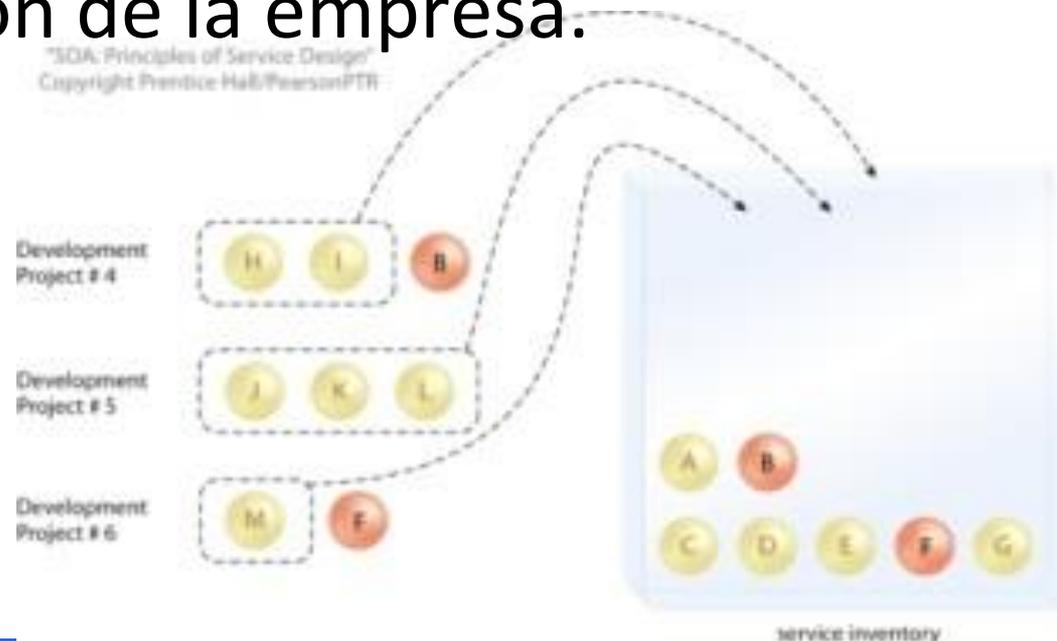


Computación Orientada a Servicios

Elementos de una SOA: Inventario de Servicios

□ Inventario de Servicios:

- Es una colección estandarizada y gobernada de servicios complementarios dentro una empresa o sección de la empresa.



(Erl, 2008)

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: Standardized 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”



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



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



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)

- ❑ 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)

- ❑ 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

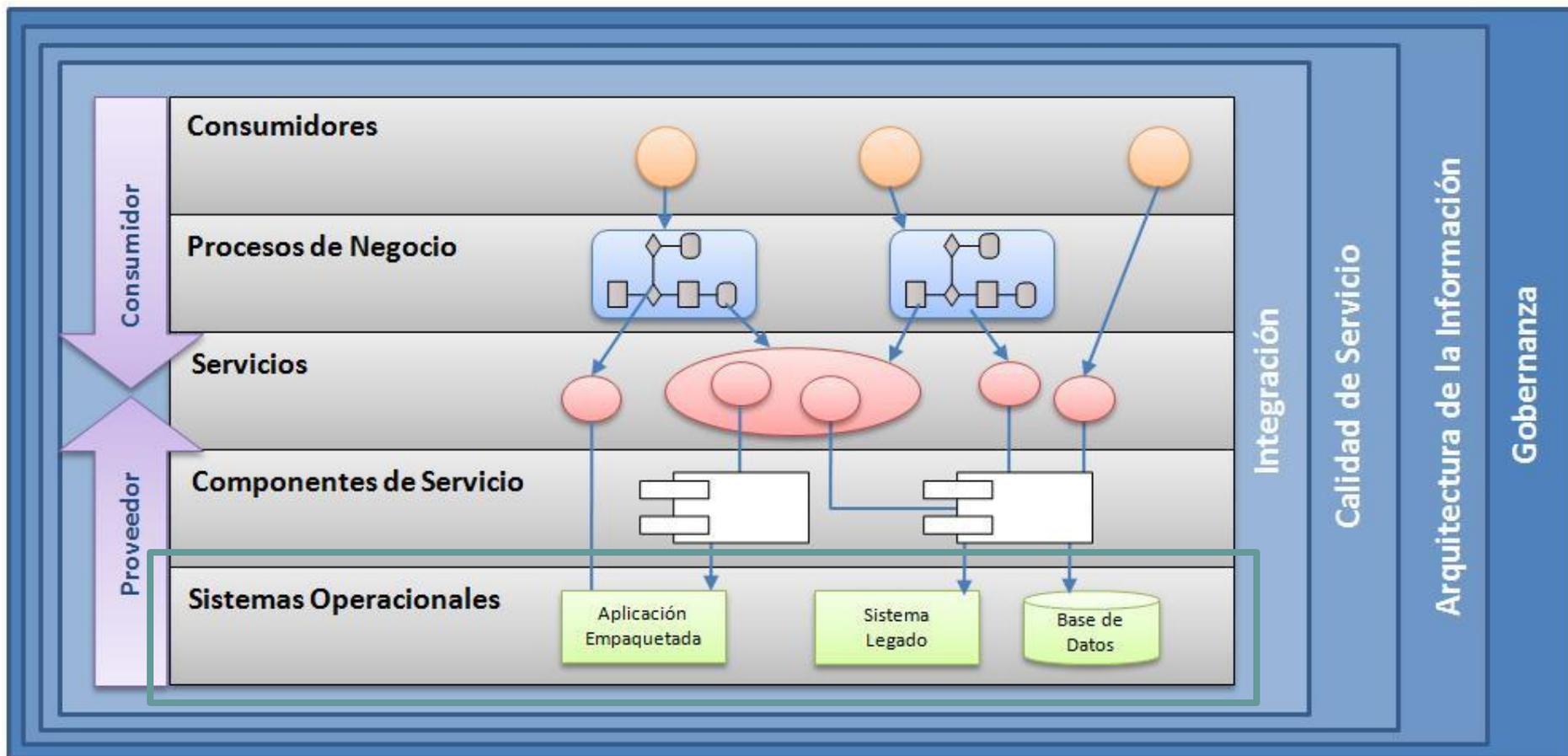
- ❑ Los beneficios reales de SOA se obtienen en general a mediano o largo plazo
- ❑ Con SOA se introducen nuevos roles, productos y procesos
- ❑ Identificación de Servicios y Definición de Contratos de Servicios
- ❑ Gobernanza de SOA

(Alluri, 2009)



Arquitectura Referencia para SOA

Capas de una SOA



(Arsaniani, 2004)

Arquitectura Referencia para SOA

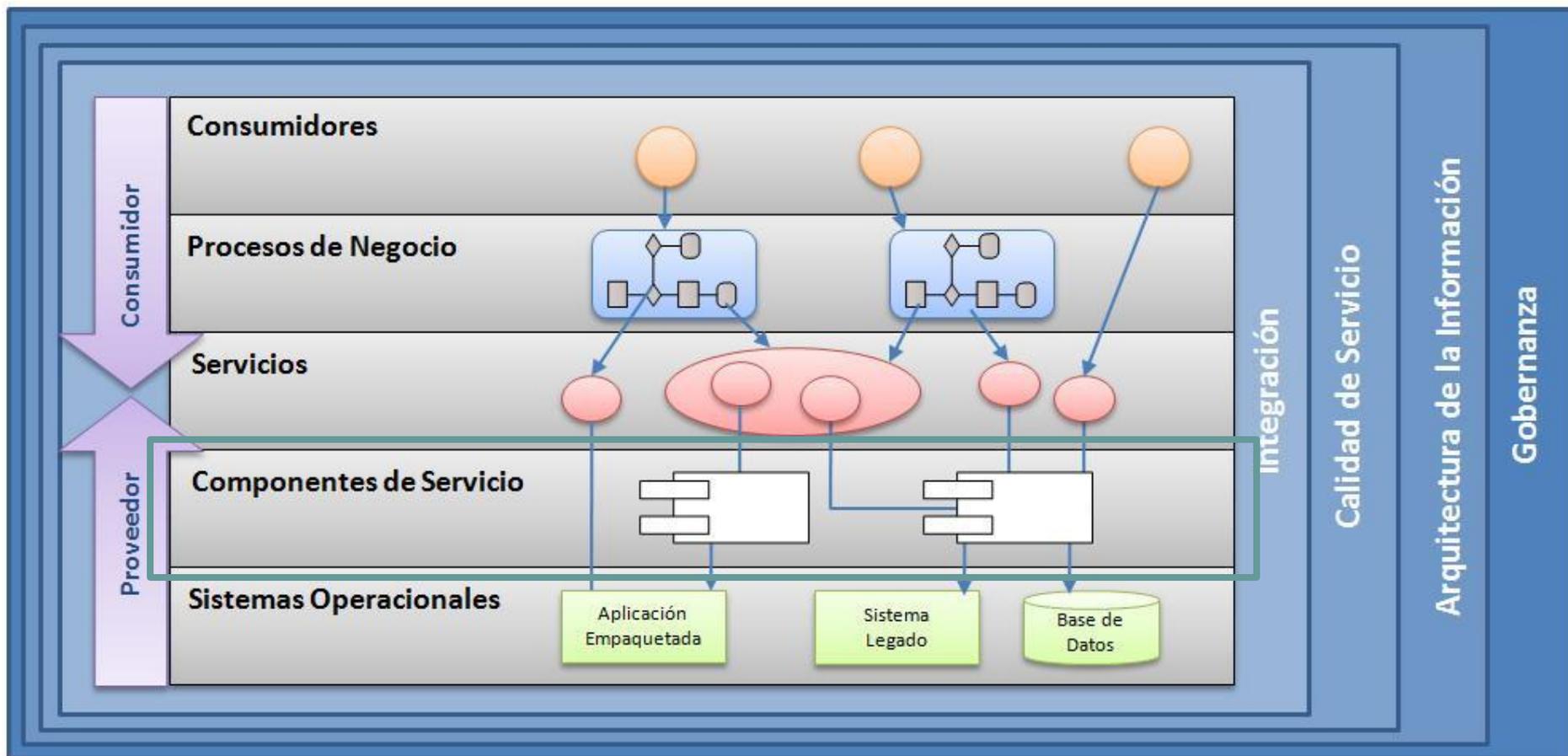
Capa: Sistemas Operacionales

- ❑ 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

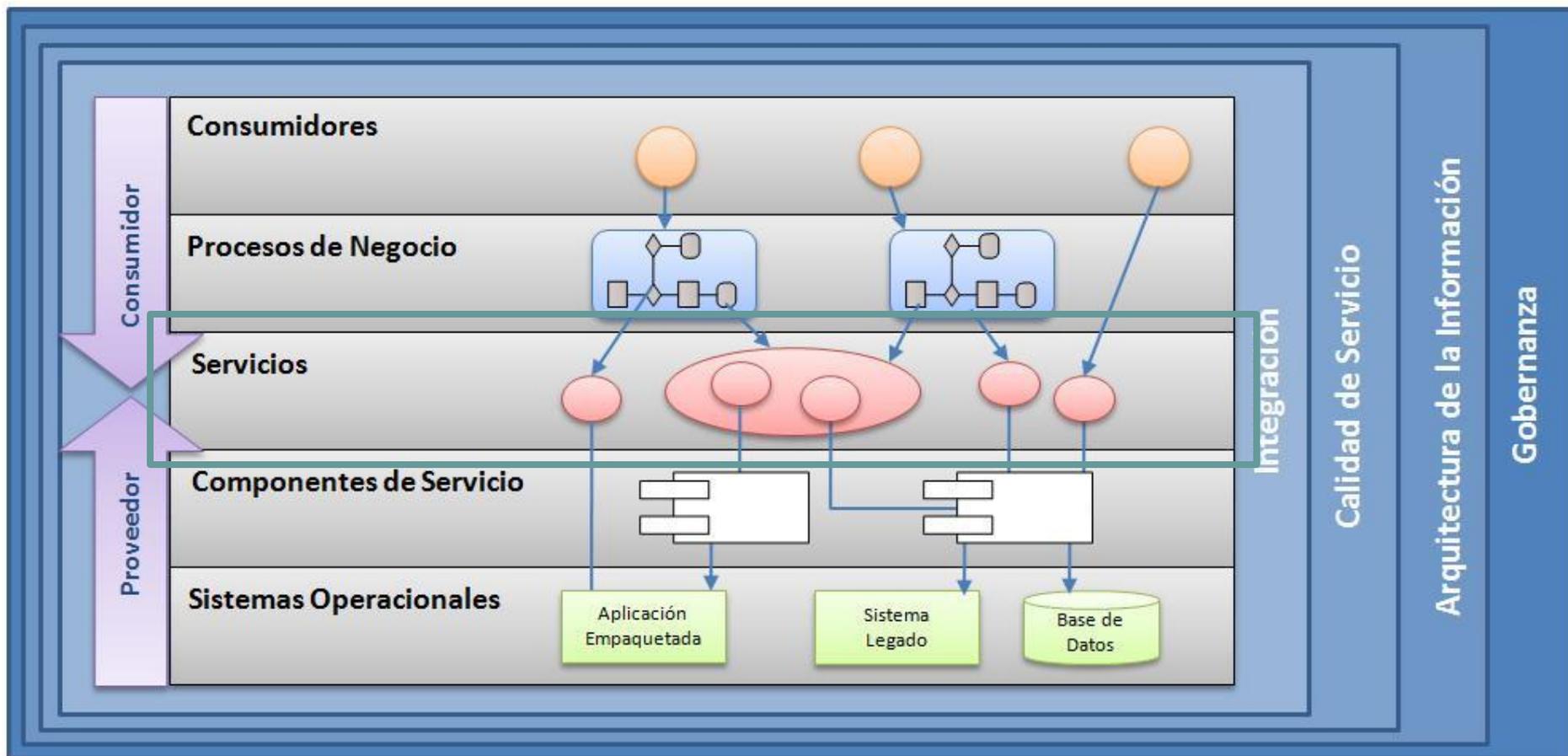
Capa: Componentes de Servicios

- ❑ 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

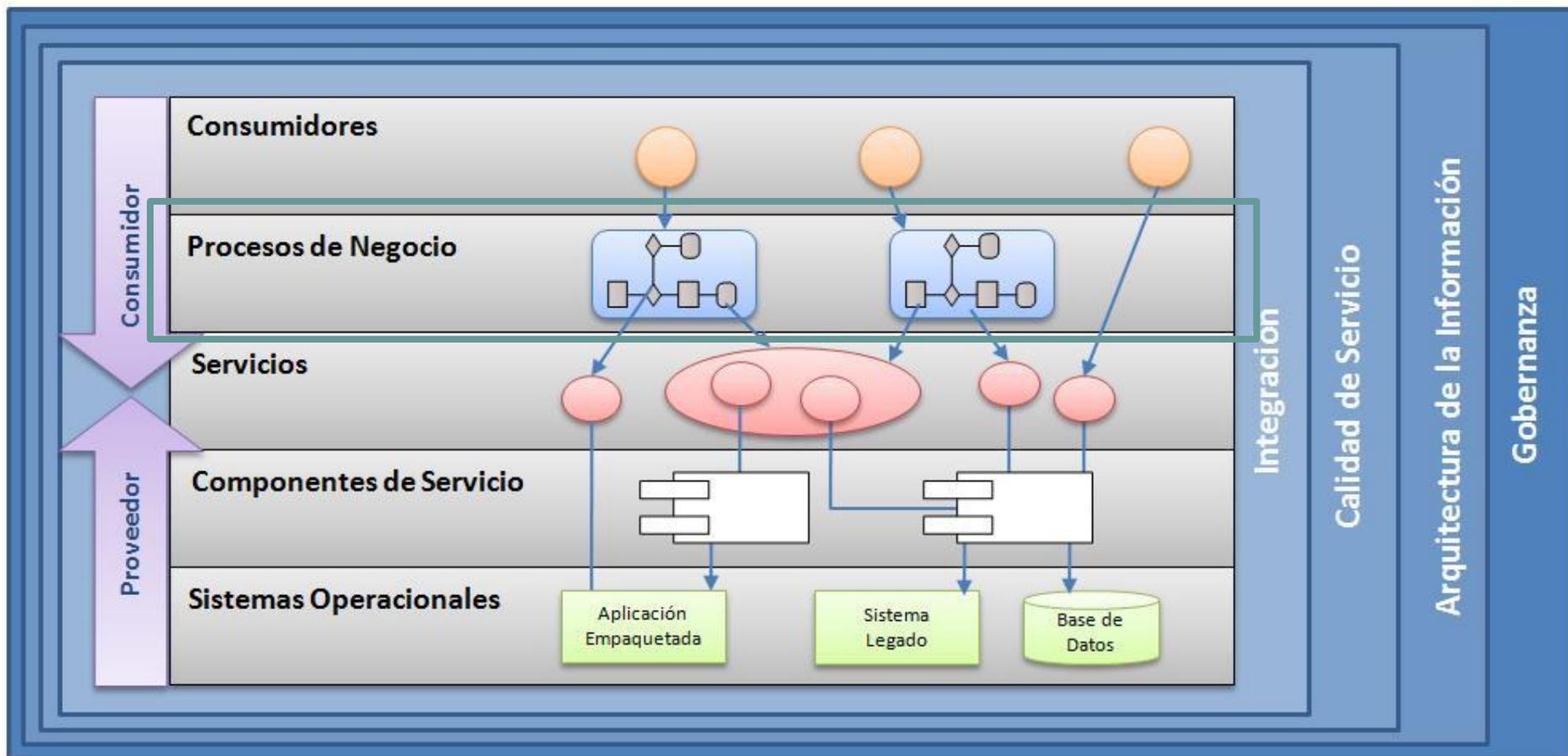
Capa Servicios

- ❑ 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

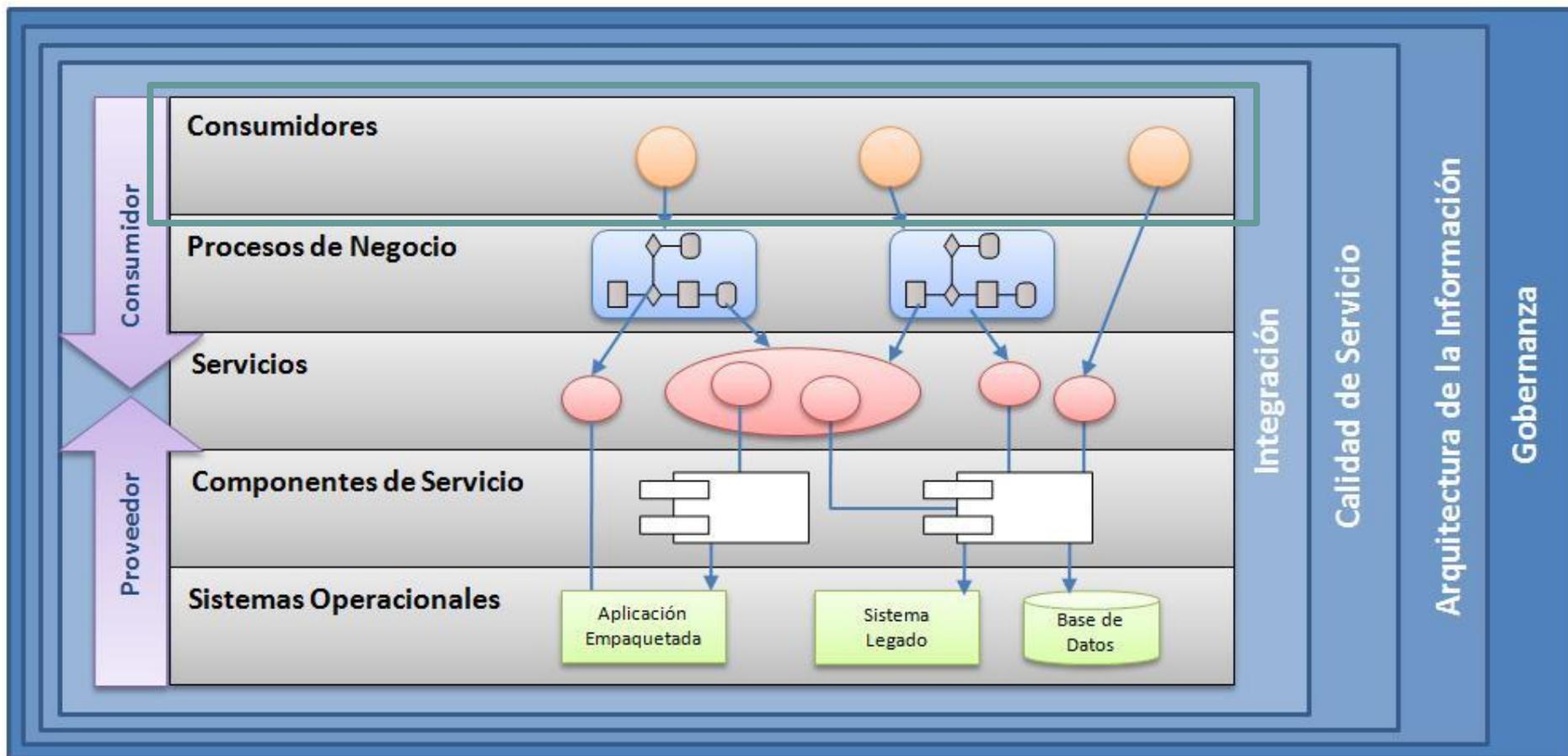
Capa: Procesos de Negocio

- ❑ 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 redundante, generalmente, en una mayor reutilización y agilidad de negocio



Arquitectura Referencia para SOA

Capas de una SOA



Arquitectura Referencia para SOA

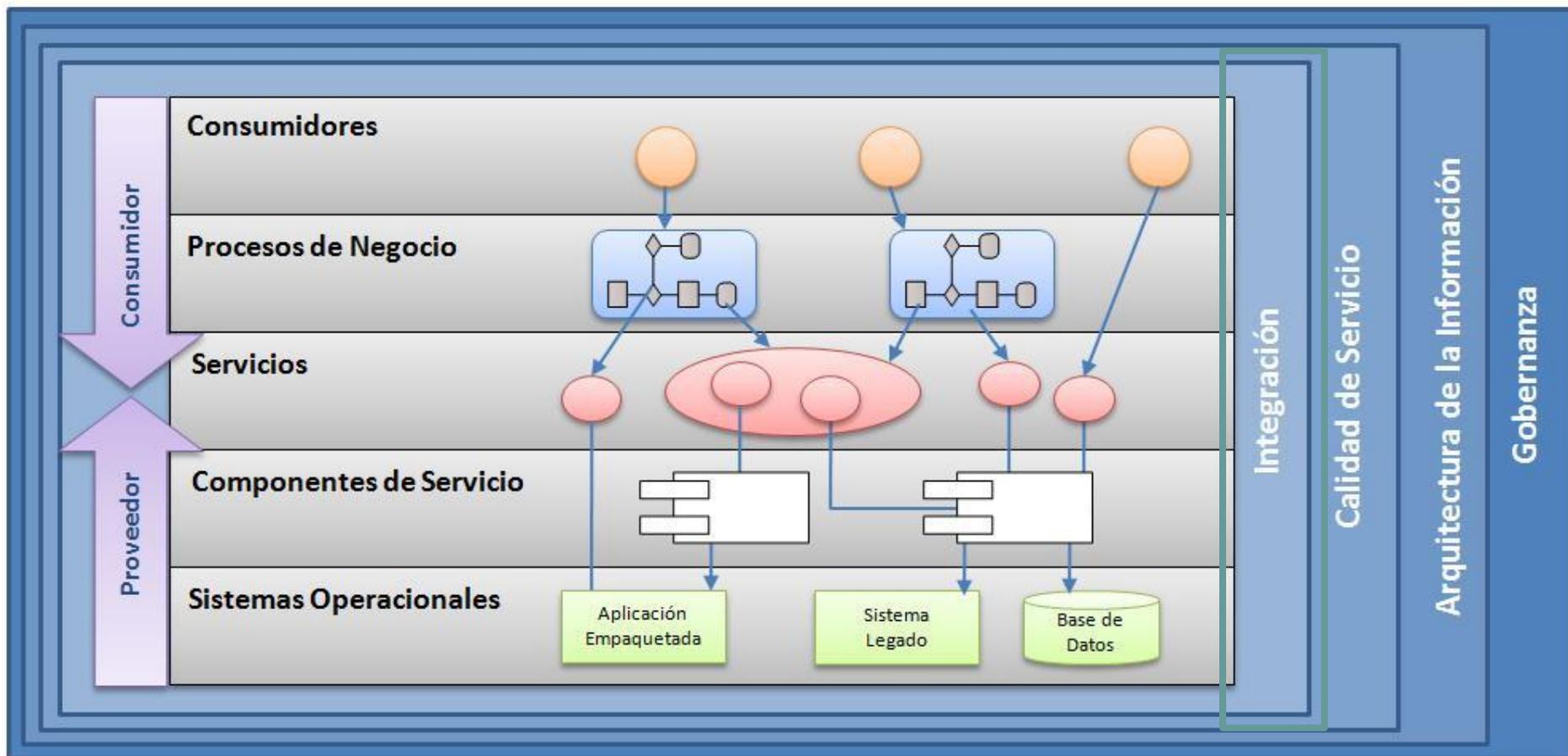
Capa: Consumidores

- ❑ 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

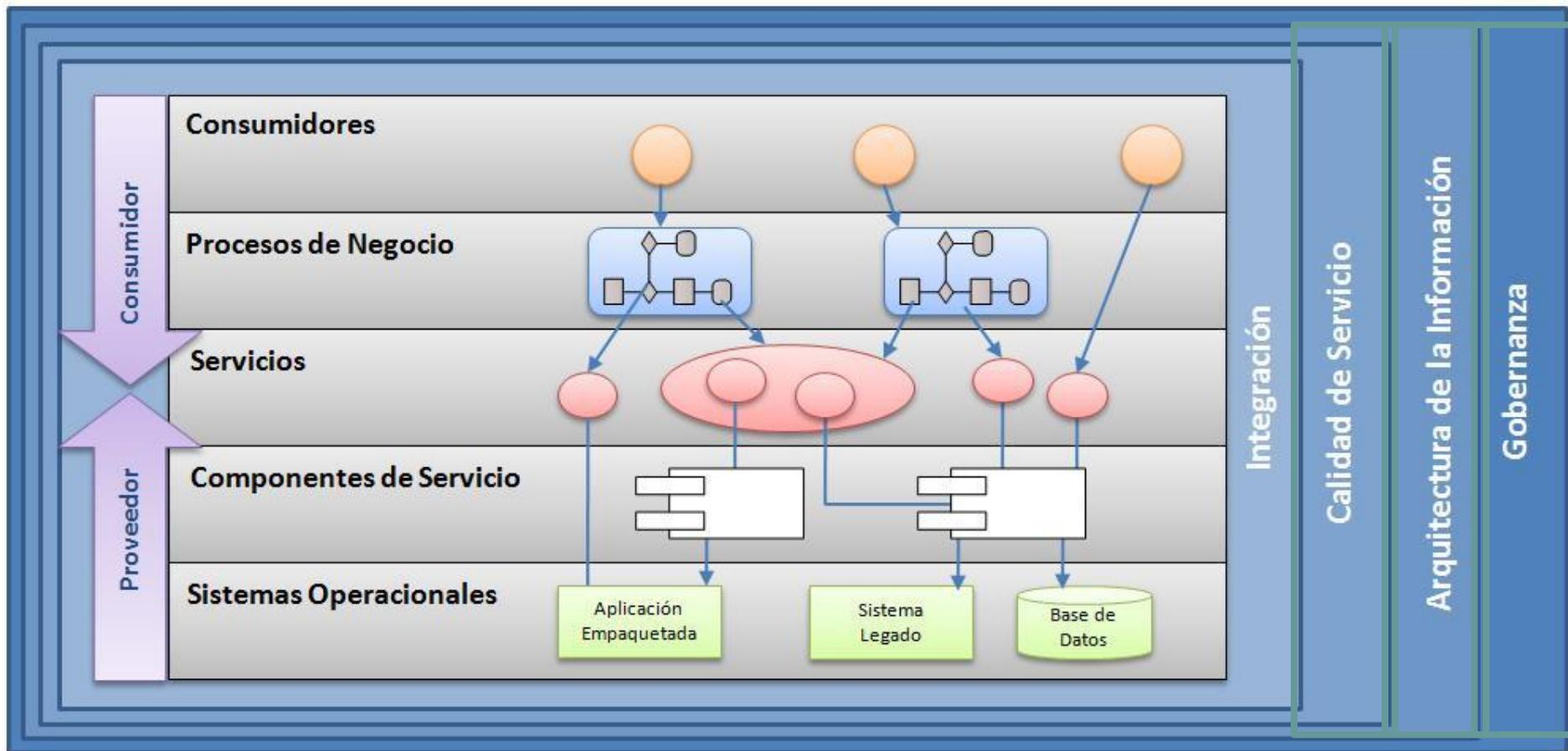
Capa: Integración

- ❑ 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)



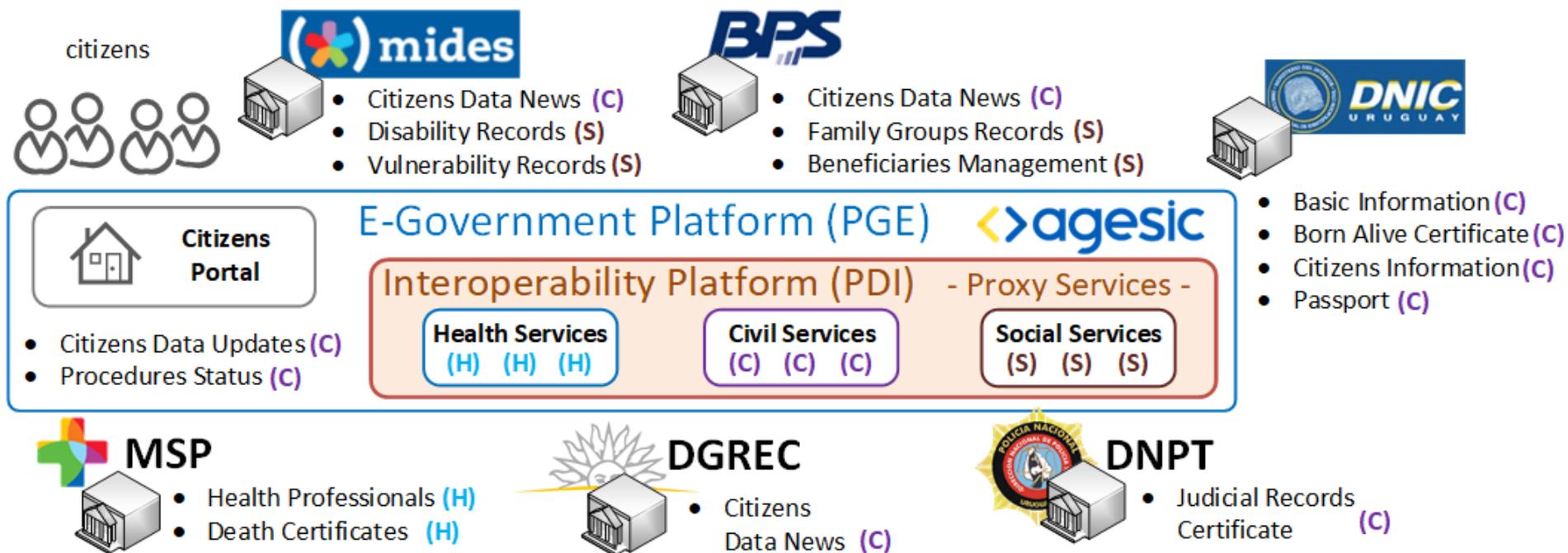
Arquitectura Referencia para SOA

Capas de una SOA



Caso de Estudio

Plataforma de Gobierno Electrónico

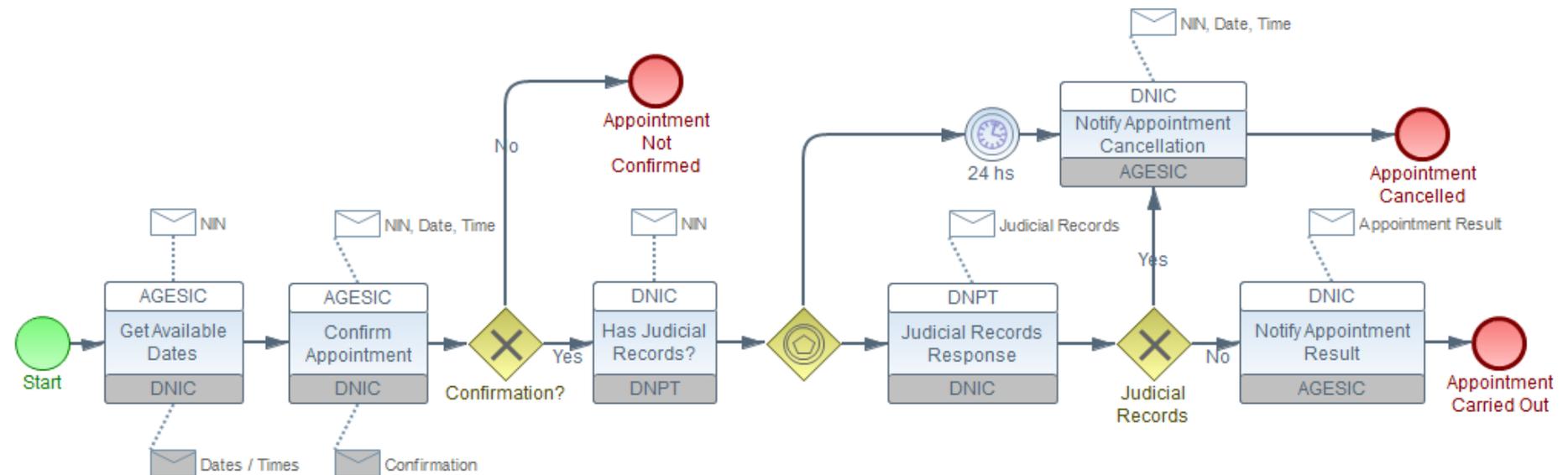


<https://www.gub.uy/agencia-gobierno-electronico-sociedad-informacion-conocimiento/tematica/catalogo-plataforma-interoperabilidad>



Caso de Estudio

Plataforma de Gobierno Electrónico



Proceso Colaborativo: Solicitud de Pasaporte



Middleware para SOA

- ❑ Web Services
- ❑ Enterprise Service Bus (ESB)



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

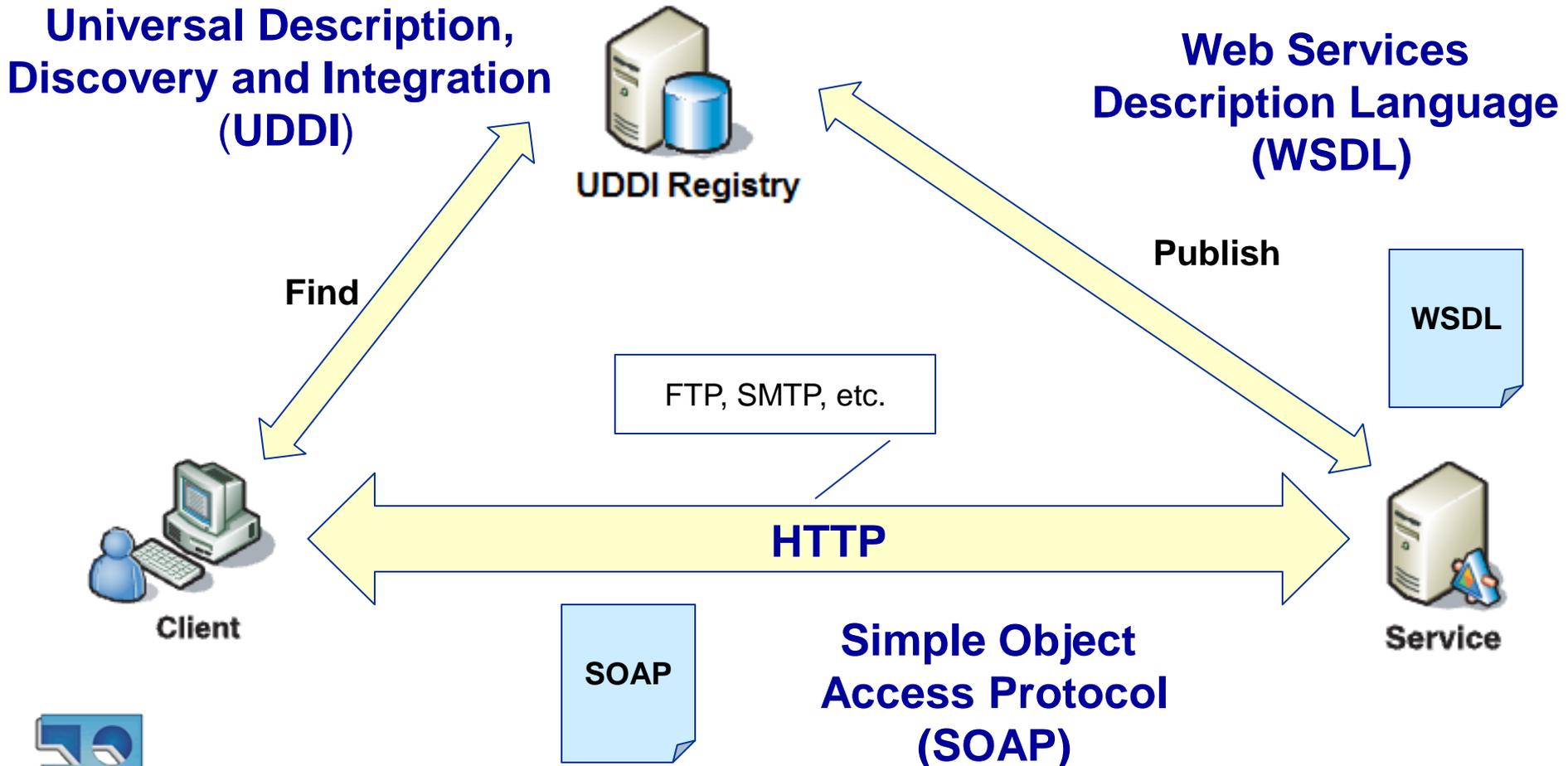
- Un Web Service es una **aplicación de software identificada por una URI**, cuyas interfaces y formas de acceso pueden ser **definidas, descriptas y descubiertas como artefactos XML**, y soporta la interacción directa con otros componentes de software utilizando **mensajes basados en XML**, intercambiados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

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



Middleware para SOA

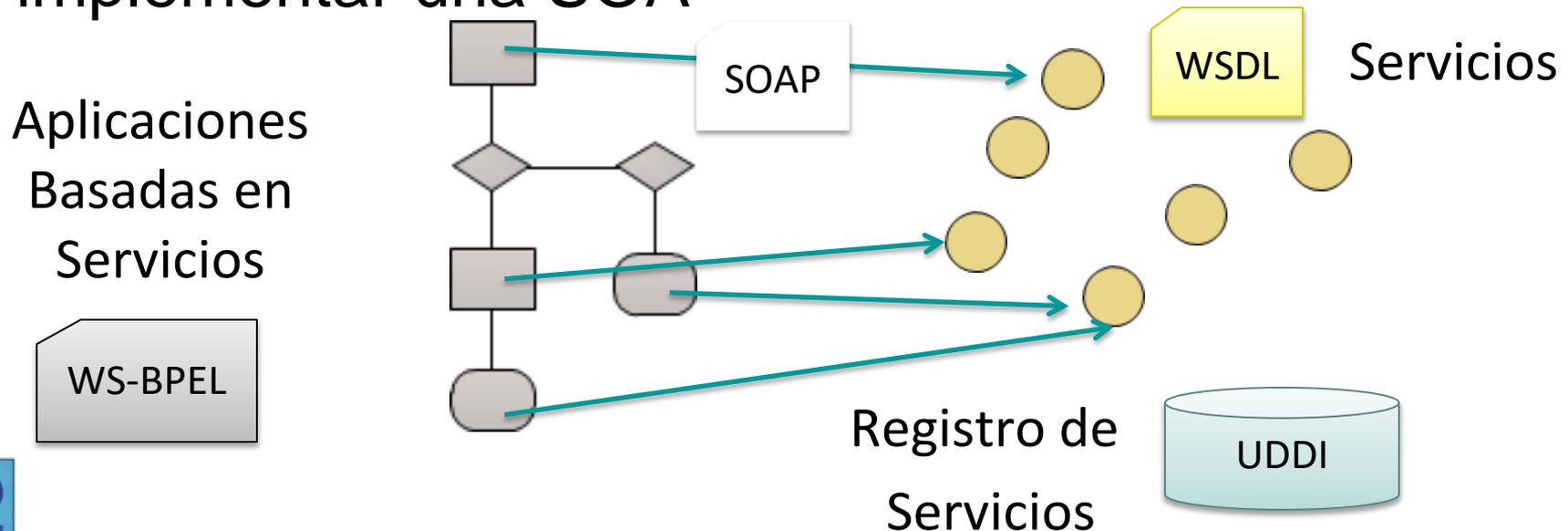
Web Services: Estándares Básicos



Middleware para SOA

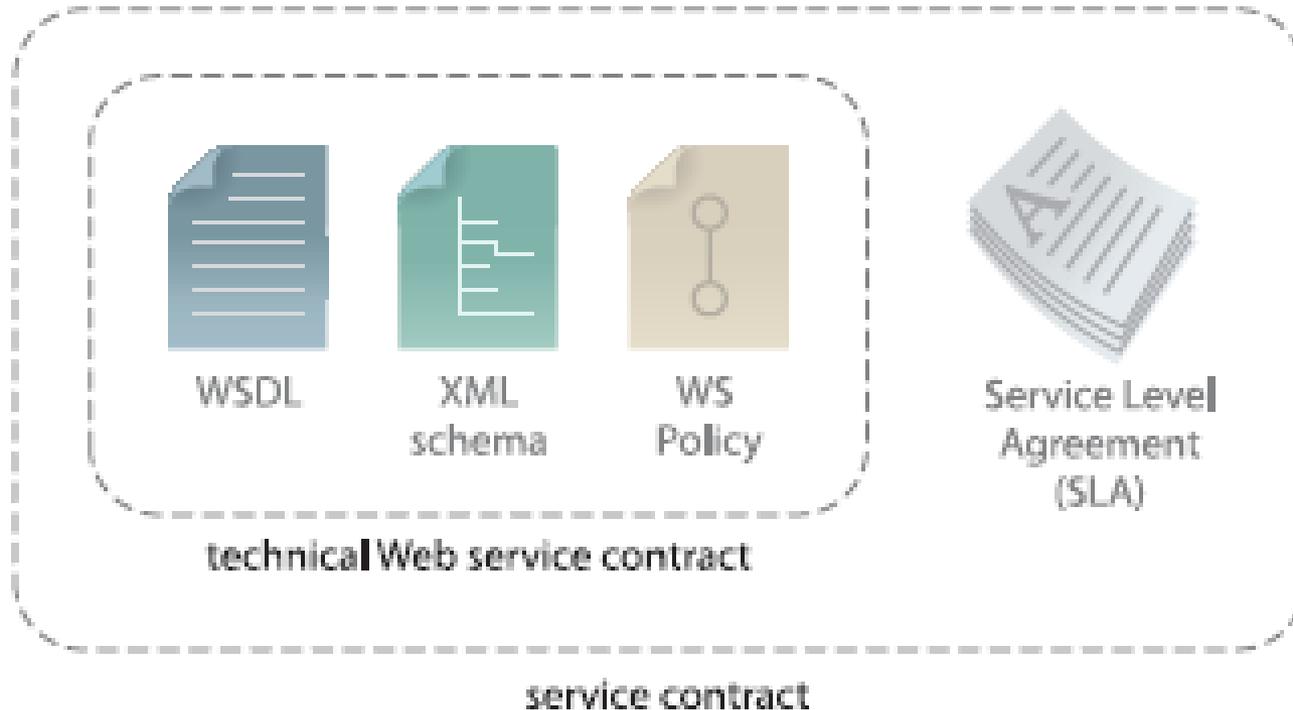
Web Services

- 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



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://j2earchitec.blogspot.com/2013/01/SOA-Basics-III.html>



Middleware para SOA

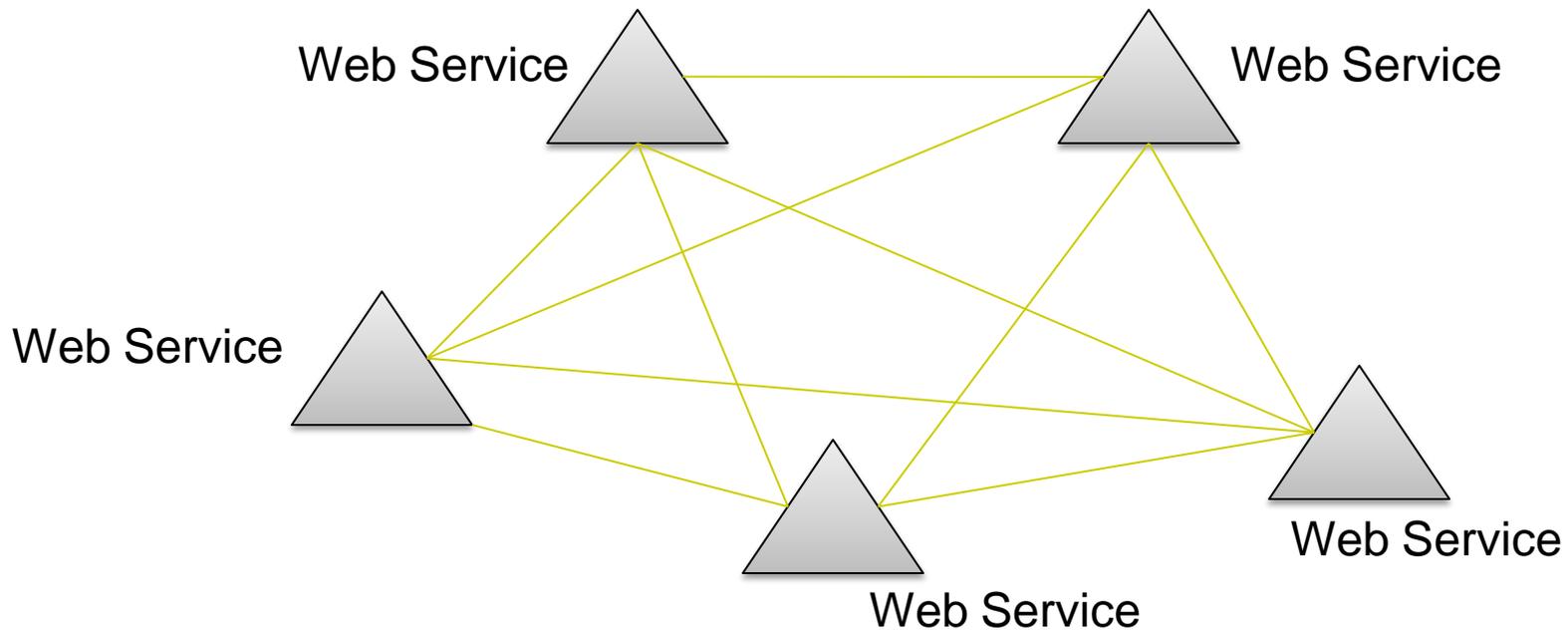
Web Services: Limitaciones

- 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)



- 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



Middleware para SOA

Web Services: Limitaciones

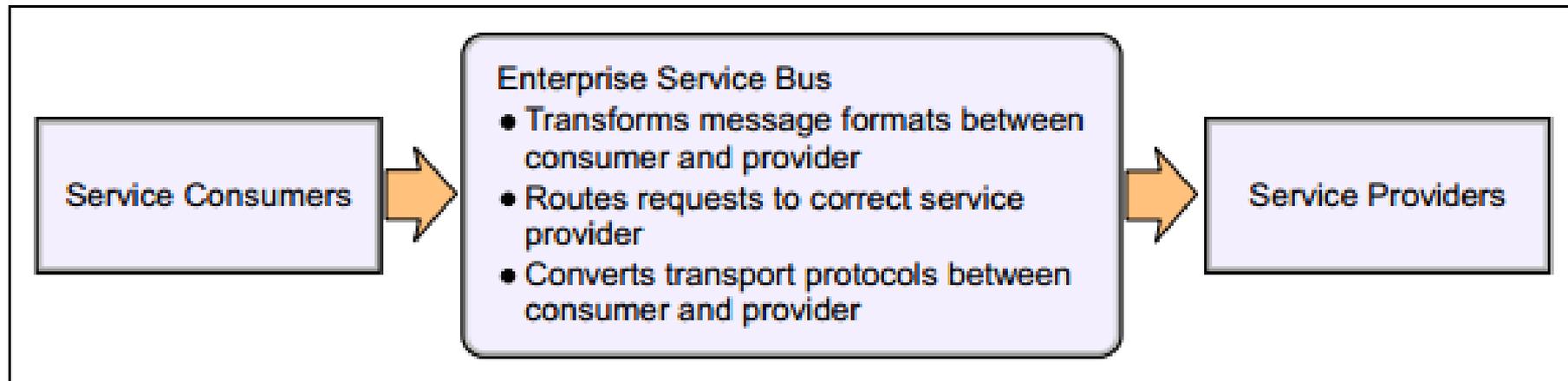
- ❑ Estas interacciones punto a punto también requieren que cada consumidor maneje un protocolo adecuado para cada proveedor que necesita utilizar.
- ❑ El tener que instalar distintos adaptadores de protocolos en varias aplicaciones cliente agrega costos y complica la mantenibilidad.

(Endrei et al., 2004)



Middleware para SOA

Enterprise Service Bus



(Endrei et al., 2004)



Middleware para SOA

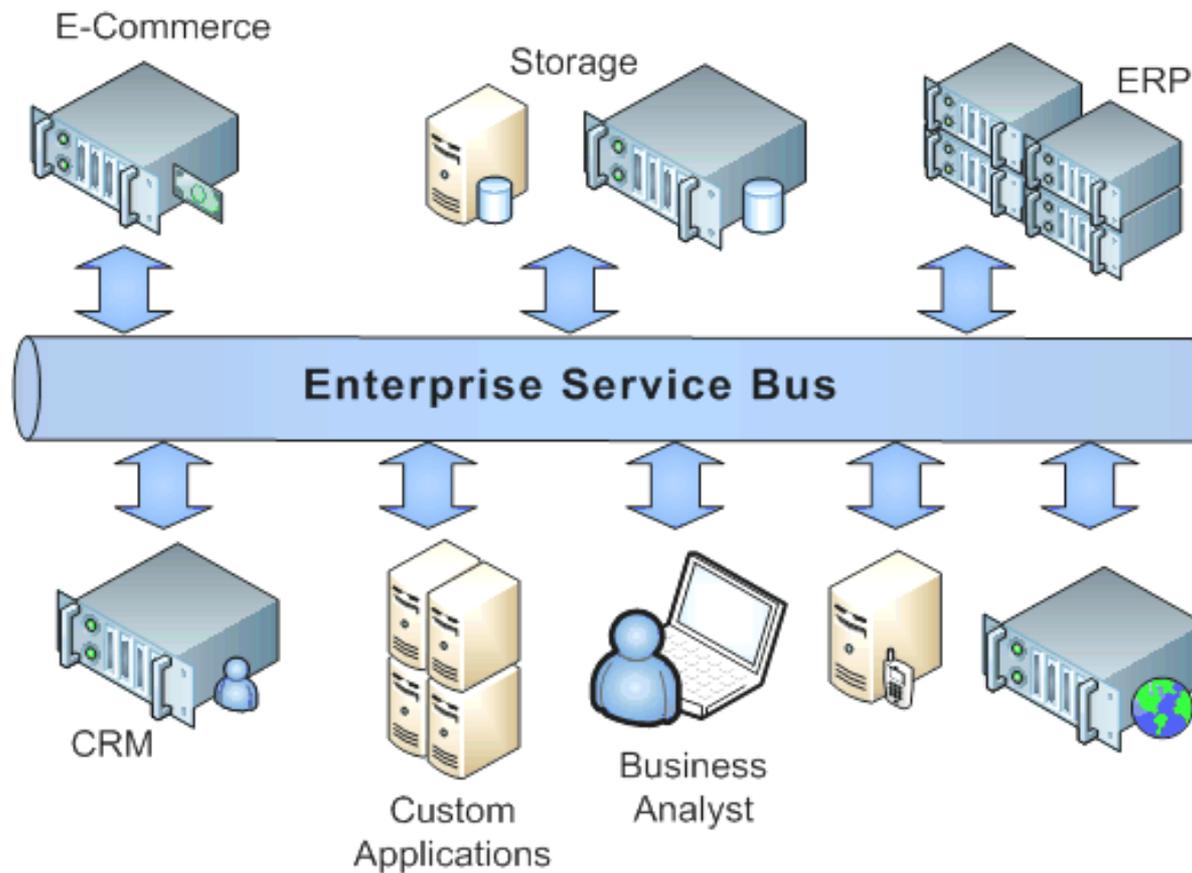
Enterprise Service Bus

- 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)



Integraciones NxN



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



- ❑ 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 introducen también desafíos
- ❑ Arquitectura Referencia SOA
 - Capas funcionales y no funcionales

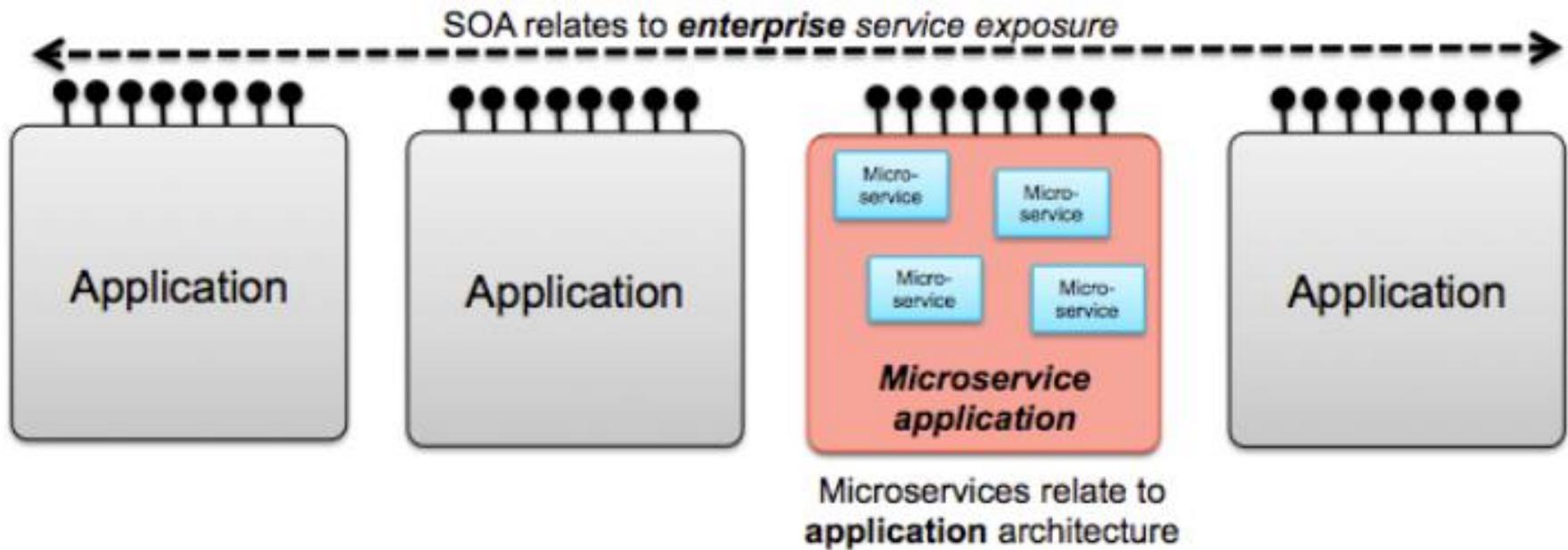


- ❑ Web Services
 - Dadas sus características constituyen la tecnología preferida para implementar una SOA (SOA != WS)
- ❑ Enterprise Service Bus
 - Provee distintas capacidades que facilitan la implementación de una SOA (SOA != ESB)



Resumen

¿y Microservicios?



- ❑ Arquitectura para el desarrollo de aplicaciones distribuidas
 - Aplicación dentro de la organización
 - Aplicación entre organizaciones
- ❑ Varias tecnologías vistas en el curso pueden ser utilizadas para implementar una SOA
- ❑ No solo entender las tecnologías, si no, los principios que las guían y cómo utilizarlas
 - Principios SOC



Preguntas



- ❑ 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.



- ❑ 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.



- ❑ 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.



- ❑ 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.



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

