

Introducción al Middleware



Enterprise Service Bus



Agenda

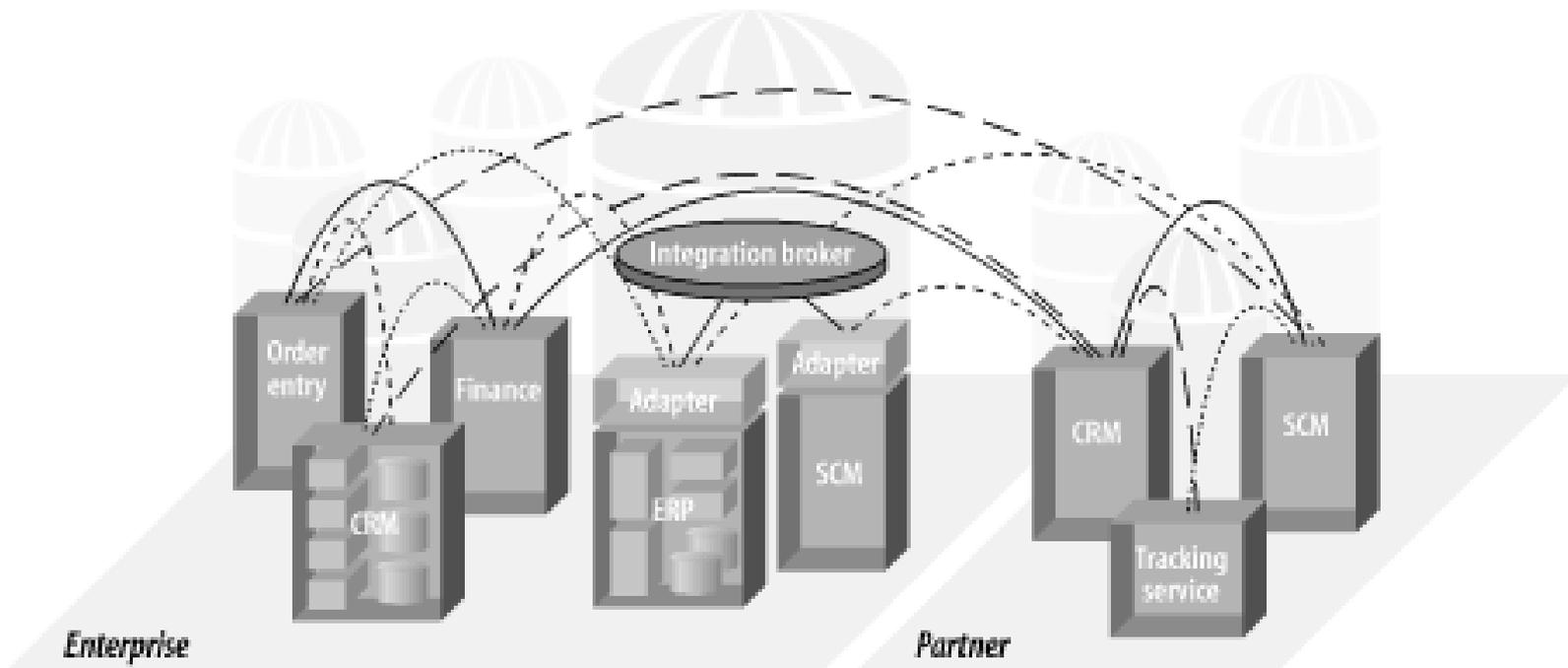
- ❑ Introducción

- ❑ Capacidades de un ESB

- ❑ Patrones ESB
 - Mediación, Conectividad y Despliegue



Arquitectura accidental



Arquitectura accidental

- ❑ Es un tipo de arquitectura que nadie definió
- ❑ Es el resultado de sucesivas integraciones punto a punto “tecnológicamente únicas” durante varios años
 - FTP, comunicación por sockets, MOMs propietarios, CORBA, SOAP u otras tecnologías RPC



Arquitectura accidental: Resultado

- ❑ Alto acoplamiento entre aplicaciones
 - Comunicaciones “tecnológicamente cableadas”
- ❑ Infraestructura rígida
- ❑ Dificultad de adaptarse rápidamente a nuevos requerimientos
 - Un cambio en una aplicación puede impactar en N aplicaciones
- ❑ Frágil arquitectura empresarial



Arquitectura accidental: Resultado

- ❑ No hay un modelo de seguridad corporativo
 - Seguridad adhoc punto a punto

- ❑ No hay una API común

- ❑ No hay protocolos de comunicaciones comunes



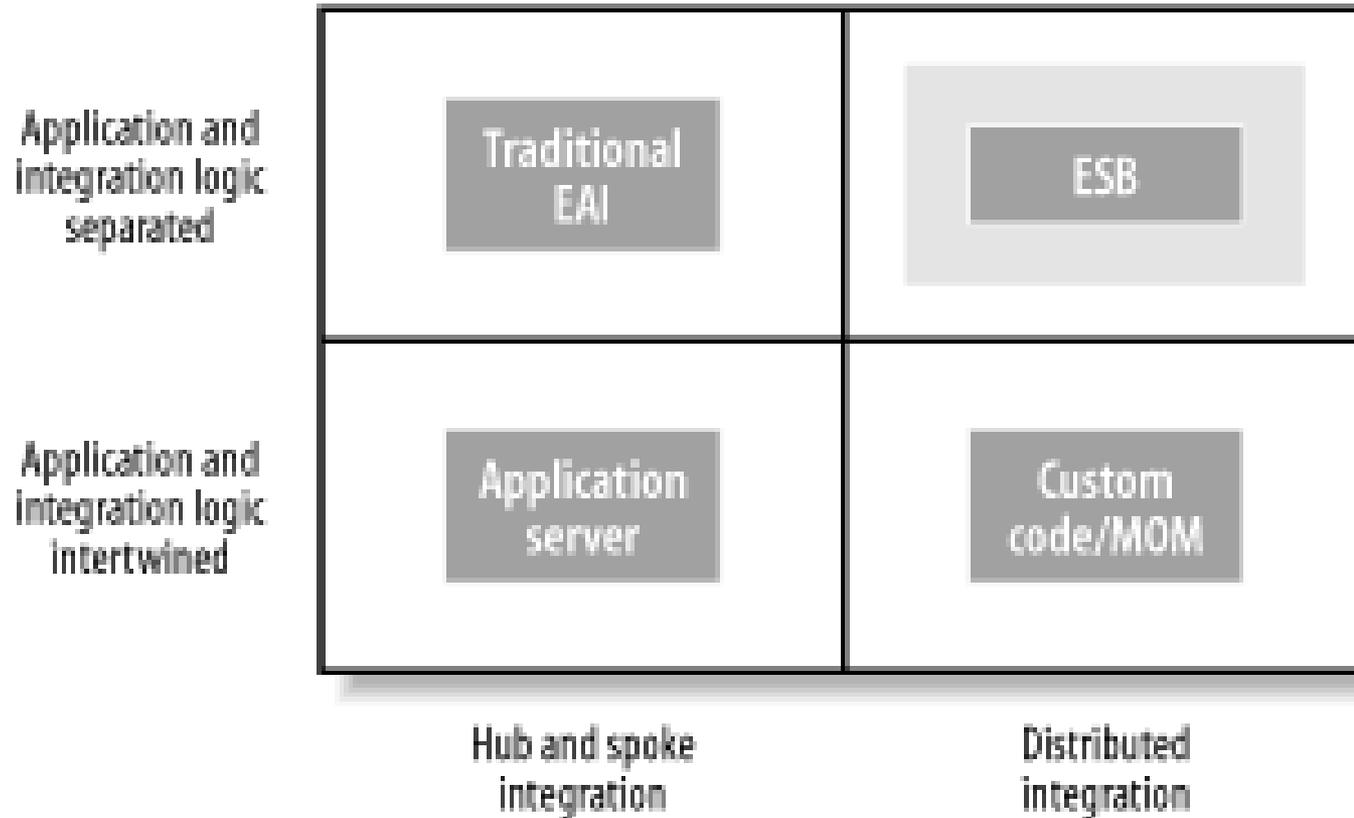
Comunicaciones poco confiables

Necesitamos...

- ❑ Nuevos enfoques para la integración de aplicaciones empresariales que permitan evitar la arquitectura accidental
- ❑ Necesitamos un nuevo tipo de arquitectura empresarial.

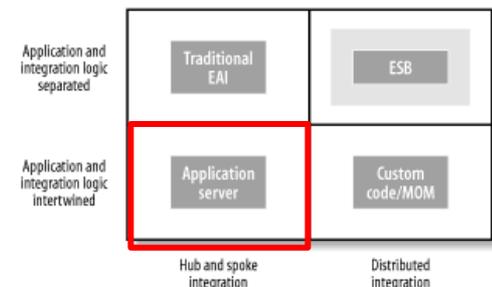


Enfoques de integración



Enfoques de integración

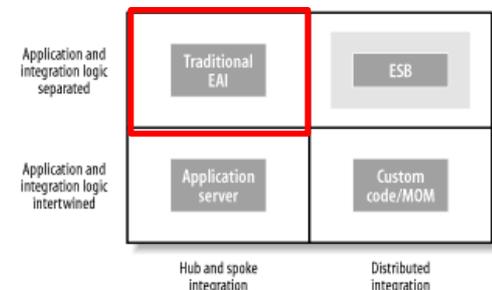
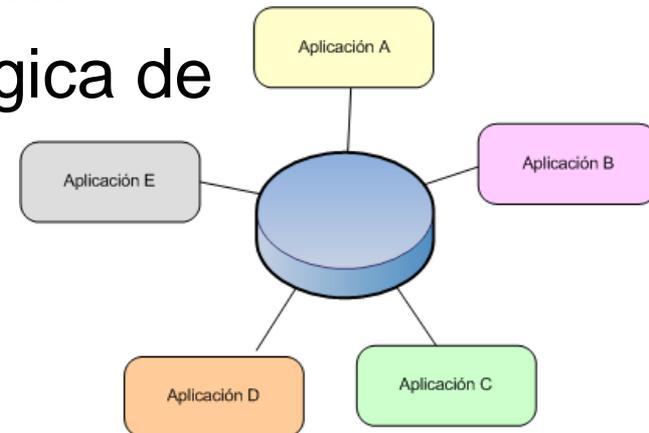
- Servidores de aplicación:
 - Interoperabilidad mediante protocolos estándares
 - p.ej: CORBA, SOAP, etc
 - Alto acoplamiento entre lógica de negocio y lógica de integración



Enfoques de integración

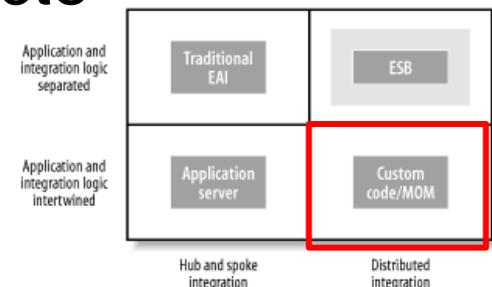
□ Brokers EAI:

- Arquitectura de tipo hub-and-spoke
- Separan lógica de negocio de lógica de integración
- Plataformas monolíticas
 - Instalación todo o nada
 - En general muy costosas
- Falta de estándares
 - Vendor lock-in
 - Mayores costos



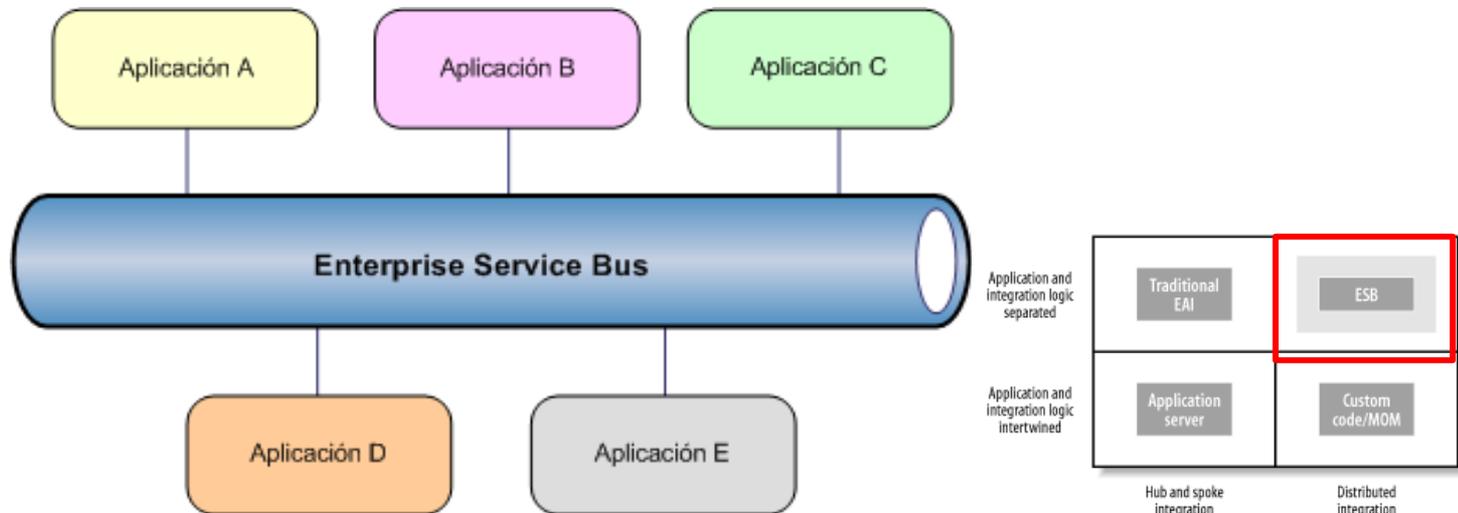
Enfoques de integración

- MOM (Message Oriented Middleware)
 - Permiten conectar aplicaciones de forma desacoplada y asíncrona
 - Alto acoplamiento entre lógica de negocio y lógica de integración
 - P. ej: ruteo y transformación de mensajes en la aplicación
 - Escalabilidad dependiente del producto

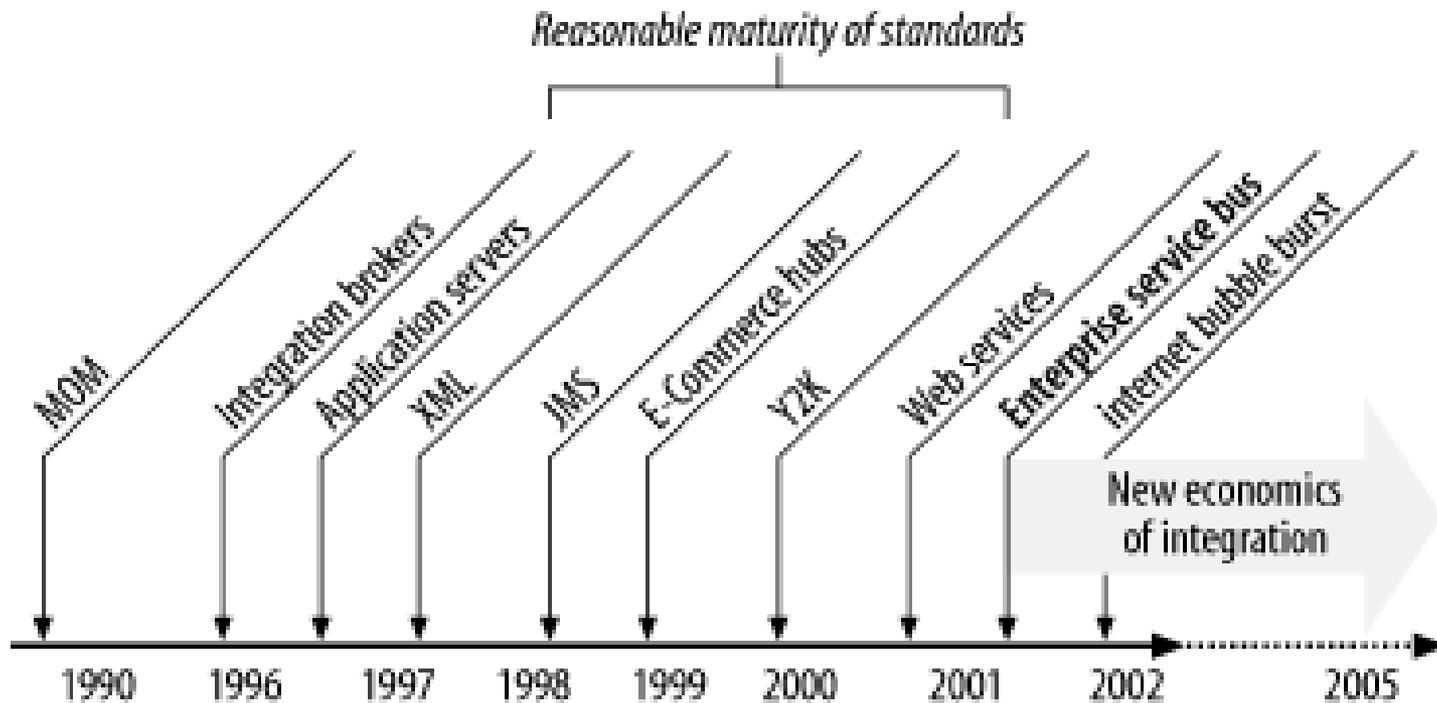


Enfoques de integración

- Enterprise Service Bus (ESB)
 - Configuración en lugar de programación
 - Ambiente de integración altamente distribuido
 - Clara separación entre lógica de negocio e integración



Evolución hacia el ESB



Algunas definiciones...

- 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)

Algunas definiciones...

- El ESB es una columna vertebral de **mensajería** basada en **estándares abiertos** y diseñada para posibilitar la implementación, despliegue y administración de soluciones **SOA**. Es un conjunto de capacidades de infraestructura implementadas vía **tecnologías de middleware** que posibilitan una SOA y alivianan problemas de **disparidad entre aplicaciones** que se ejecutan en **plataformas heterogéneas** y usan **diversos formatos de datos**.



(Mike P. Papazoglou 2007)

Algunas definiciones...

- Un ESB representa un ambiente para fomentar la **inter-conectividad sofisticada** entre **servicios**. Provee una **capa de procesamiento intermedio** que puede ayudar a solucionar problemas comunes asociados a la **confiabilidad, escalabilidad y disparidad en la comunicación**.



Resumiendo las definiciones...

- ❑ Plataforma de integración
- ❑ Tecnologías de middleware
- ❑ Estándares abiertos
- ❑ Mensajería
- ❑ Web Services
- ❑ Transformación de formatos
- ❑ Ruteo inteligente
- ❑ Conectividad
- ❑ Integridad transaccional
- ❑ Escalabilidad
- ❑ Confiabilidad
- ❑ Aplicaciones heterogéneas
- ❑ Soporte a SOA



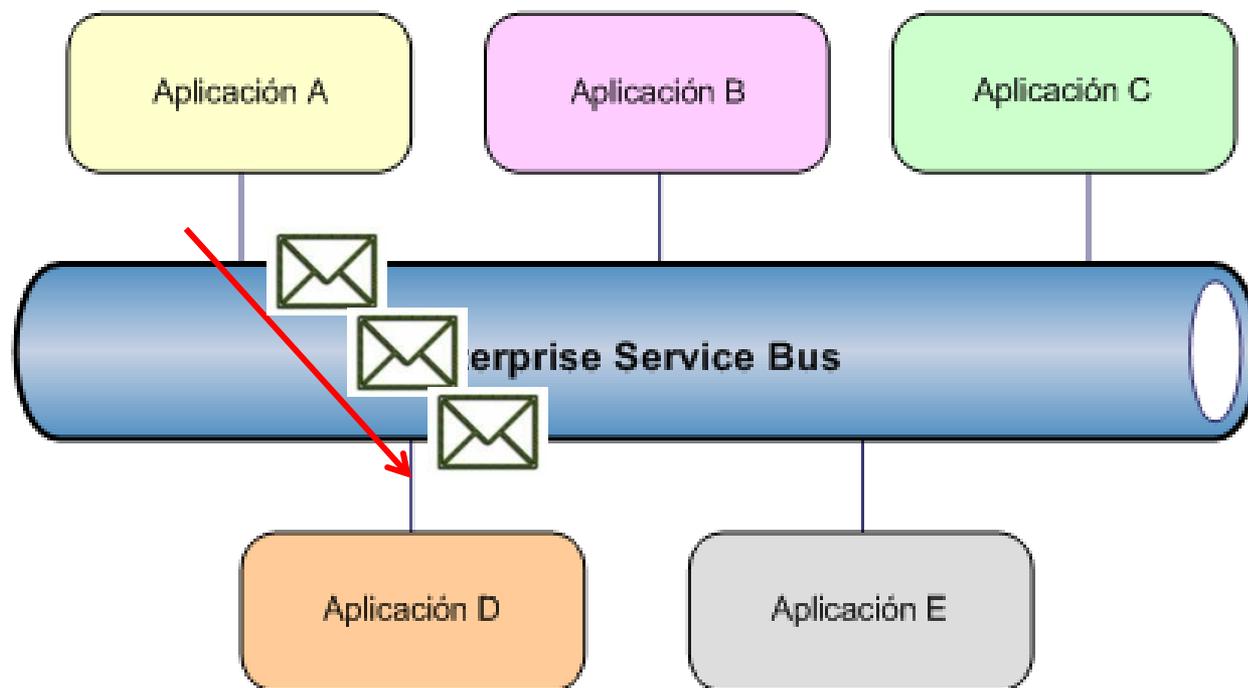
Estilos de Integración en ESBs

- Un ESB soporta tres principales estilos de integración:
 - Service-oriented architectures (SOA)
 - las aplicaciones se comunican via servicios reutilizables con interfaces explícitas y bien definidas
 - Message-driven architectures
 - las aplicaciones envían mensajes a otras aplicaciones
 - Event-driven architectures (EDA)
 - las aplicaciones generan y consumen mensajes (para notificar eventos) independientemente unas de otras

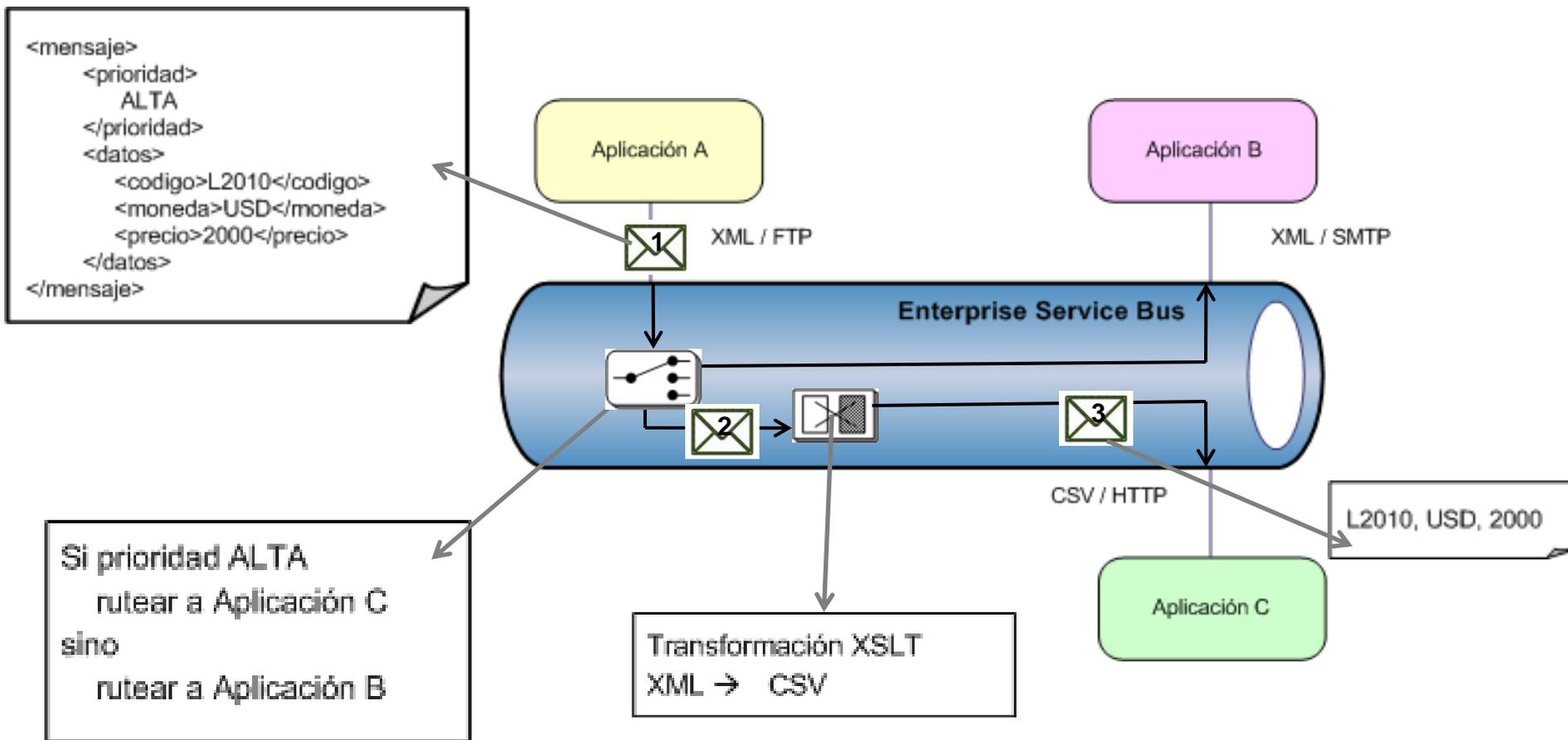
(Keen et al. 2004)



Integración con un ESB



Integración con un ESB



Integración con un ESB

- ❑ En lugar de interactuar directamente las aplicaciones se comunican enviando mensajes a través del ESB
- ❑ Los mensajes que fluyen a través del ESB son en general mensajes XML



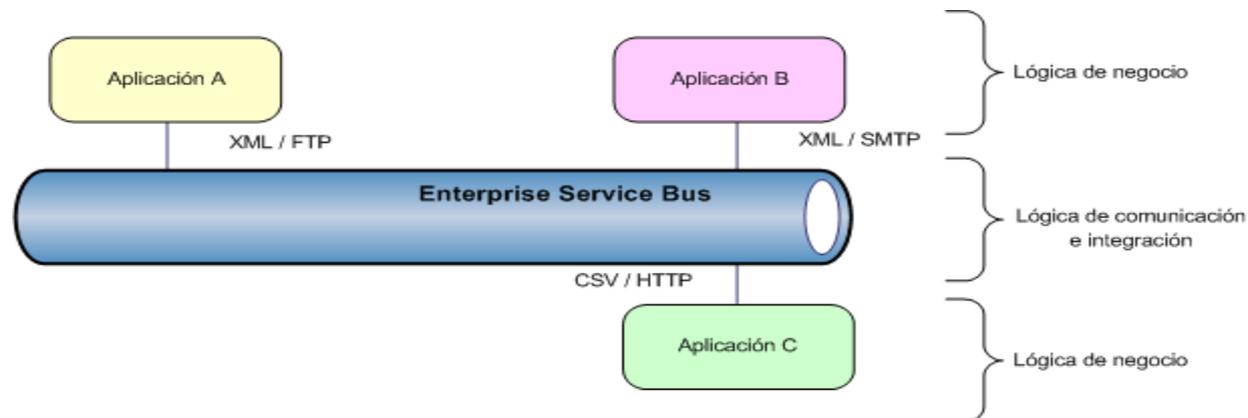
Integración con un ESB

- ❑ El ESB puede realizar distintas operaciones de mediación sobre los mensajes
 - transformaciones, ruteo, etc
- ❑ Estas operaciones permiten asegurar que las aplicaciones se puedan conectar correctamente

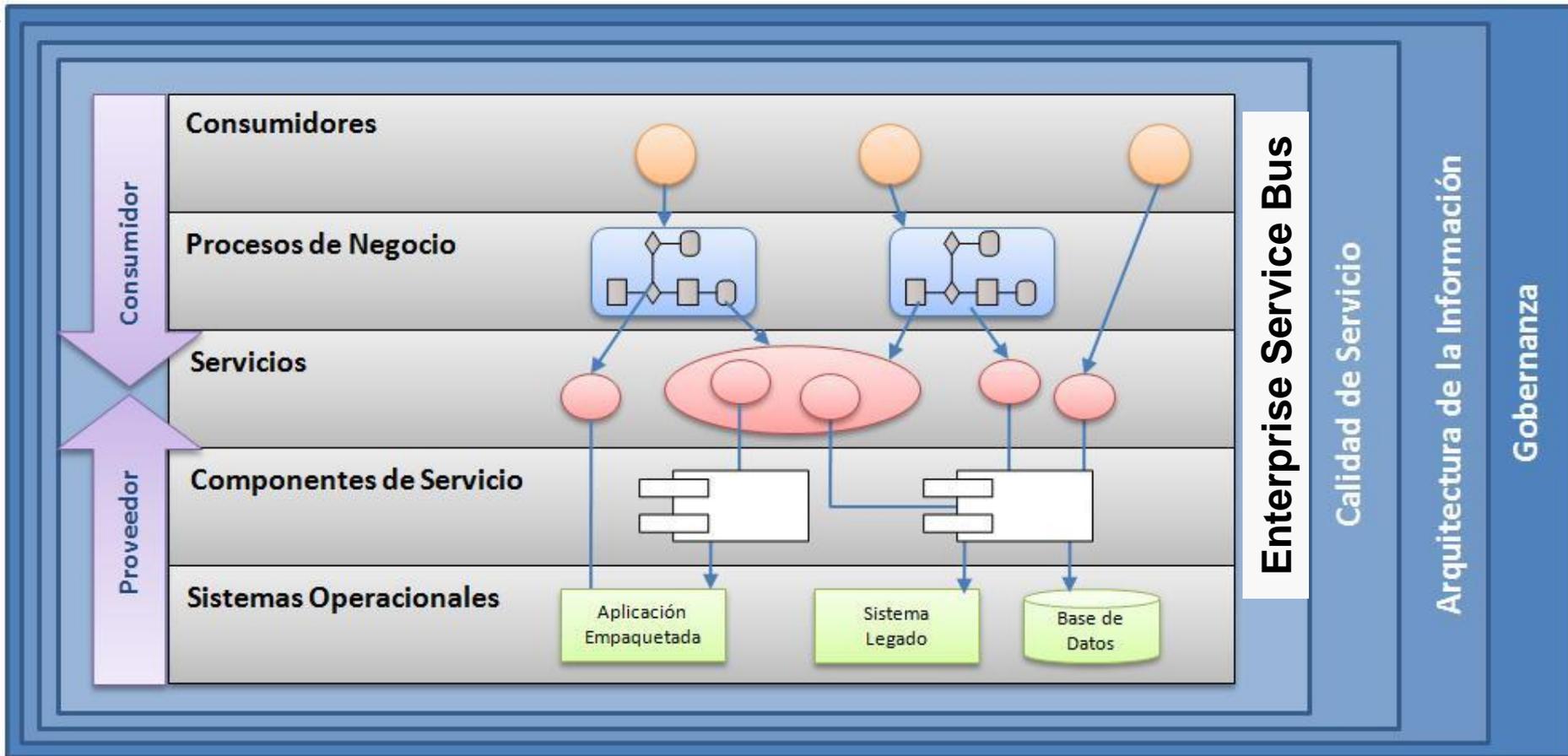


Integración con un ESB

- ❑ El ESB promueve el bajo acoplamiento entre las aplicaciones y divide la lógica de comunicación e integración en pequeños módulos reutilizables
- ❑ Permite además una clara separación entre:
 - lógica de integración y comunicación
 - lógica de negocio implementada por las aplicaciones



ESB en una SOA



Agenda

- Introducción

- Capacidades de un ESB

- Patrones ESB
 - Mediación, Conectividad y Despliegue



Principales capacidades

- ❑ Conectividad / Adaptadores
- ❑ Transformación de Mensajes
- ❑ Ruteo Inteligente
- ❑ Flujos de Mediación
- ❑ Mensajería Asincrónica
- ❑ Monitoreo y Administración
- ❑ Otras características...



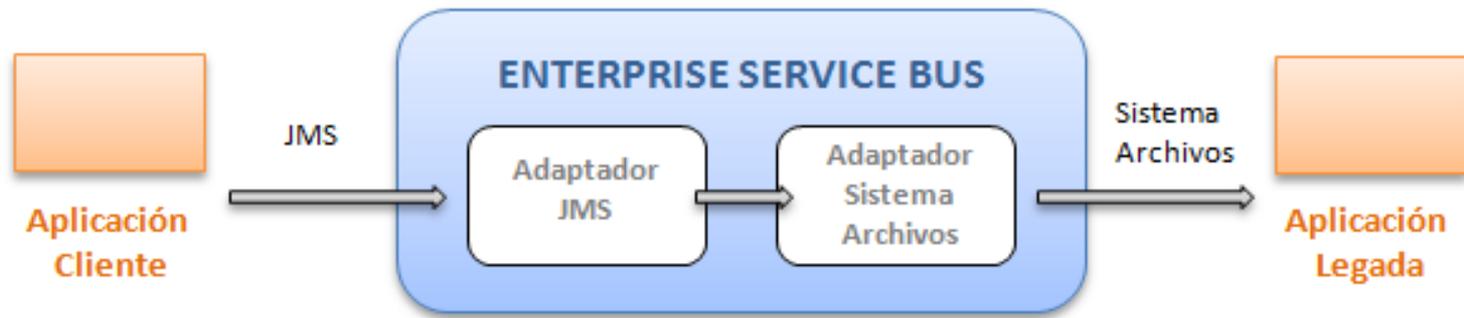
Conectividad /Adaptadores

- ❑ Un ESB típicamente incluye varios adaptadores que le permiten interactuar con múltiples protocolos de comunicación o transporte
 - HTTP, FTP, POP3/SMTP, sistemas de archivos, etc
- ❑ En general se configuran utilizando algún archivo de configuración o asistente
- ❑ Un adaptador funciona en general como servidor o cliente
 - Por ejemplo, un HTTP listener o HTTP consumer



Conectividad / Adaptadores

- ❑ Permiten satisfacer un requerimiento de integración común conocido como conversión de protocolo (protocol switch).
 - Dos aplicaciones que necesitan integrarse y no manejan un protocolo de comunicación común.

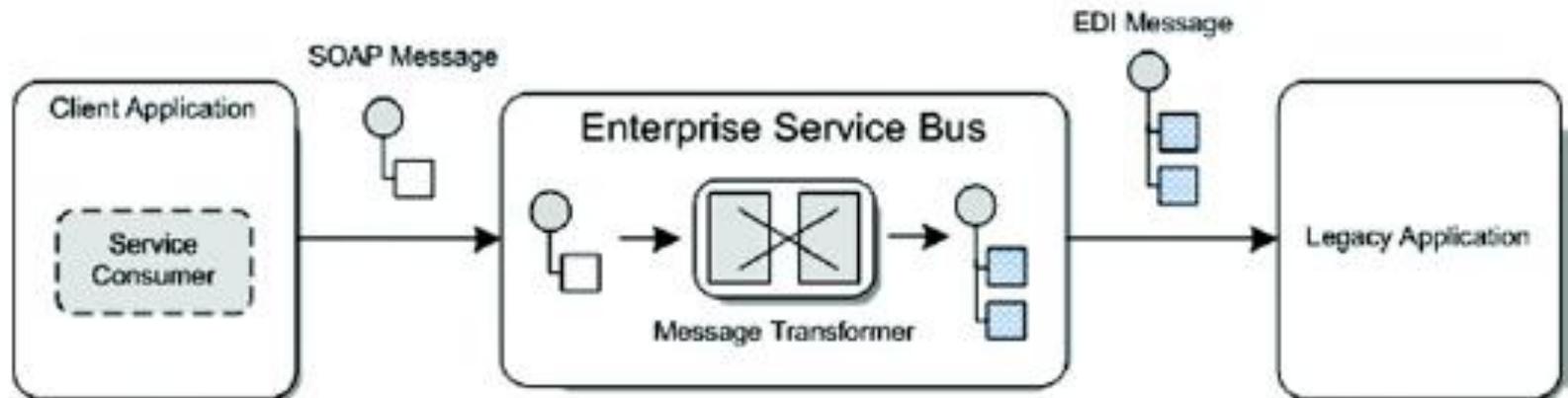


(Rademakers and Dirksen 2008)



Transformación de Mensajes

- ❑ Los ESBs también incluyen capacidades de transformación de mensajes.
- ❑ Estas capacidades posibilitan, por ejemplo, que aplicaciones que utilizan distintos formatos o modelos de datos puedan comunicarse.

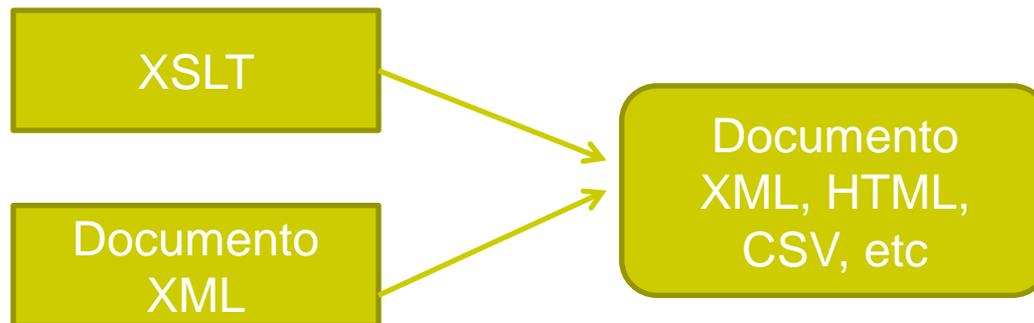


(Rademakers and Dirksen 2008)



Transformación de Mensajes

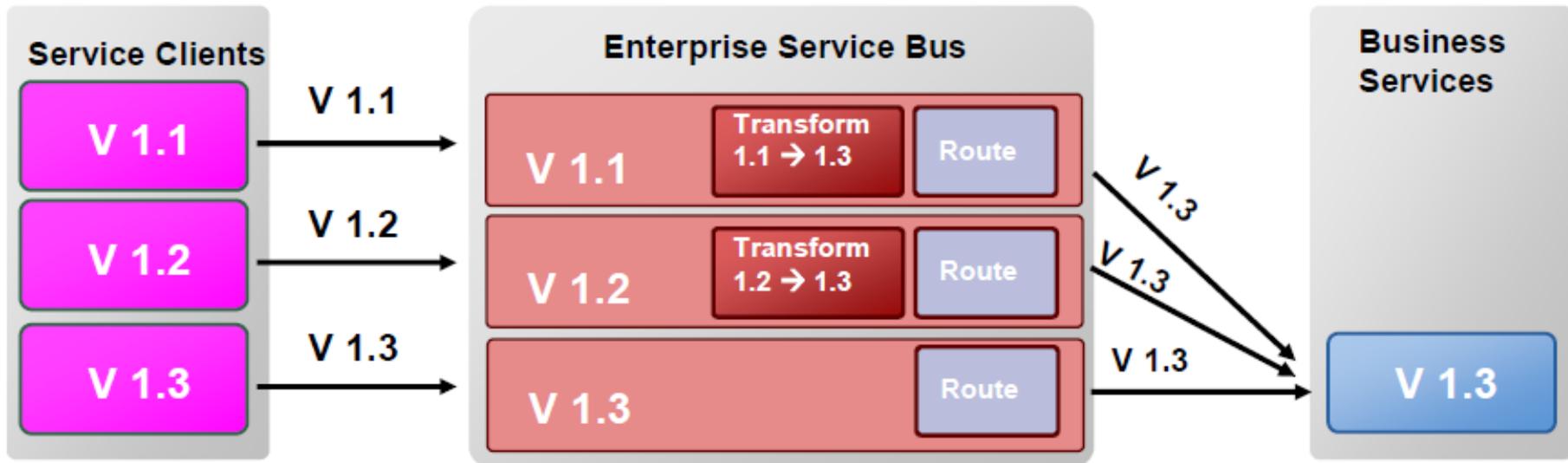
- En general, los ESB soportan transformaciones a través de XSLT



- También incluyen herramientas para transformar mensajes no XML (ej. Smooks)



Transformación de Mensajes



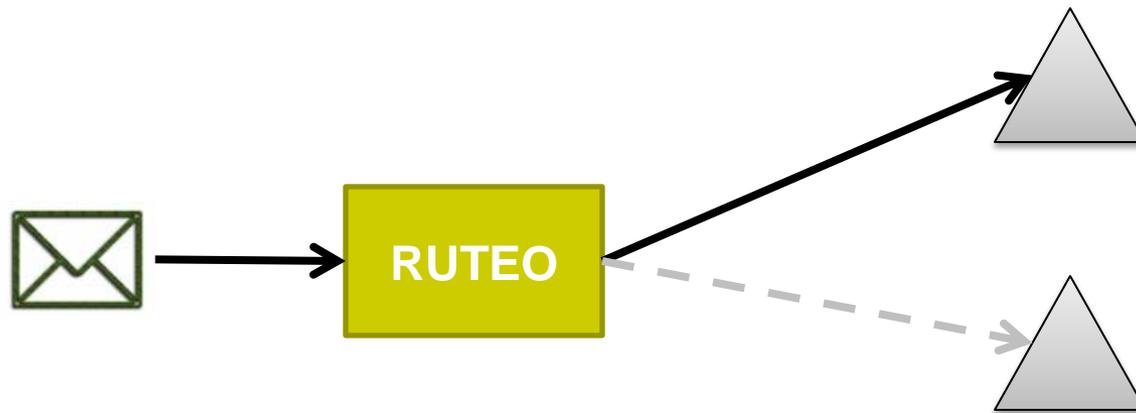
□ Ejemplos:

- Dar soporte a diferentes versiones de un servicio
- SOAP 1.1 \leftrightarrow SOAP 1.2

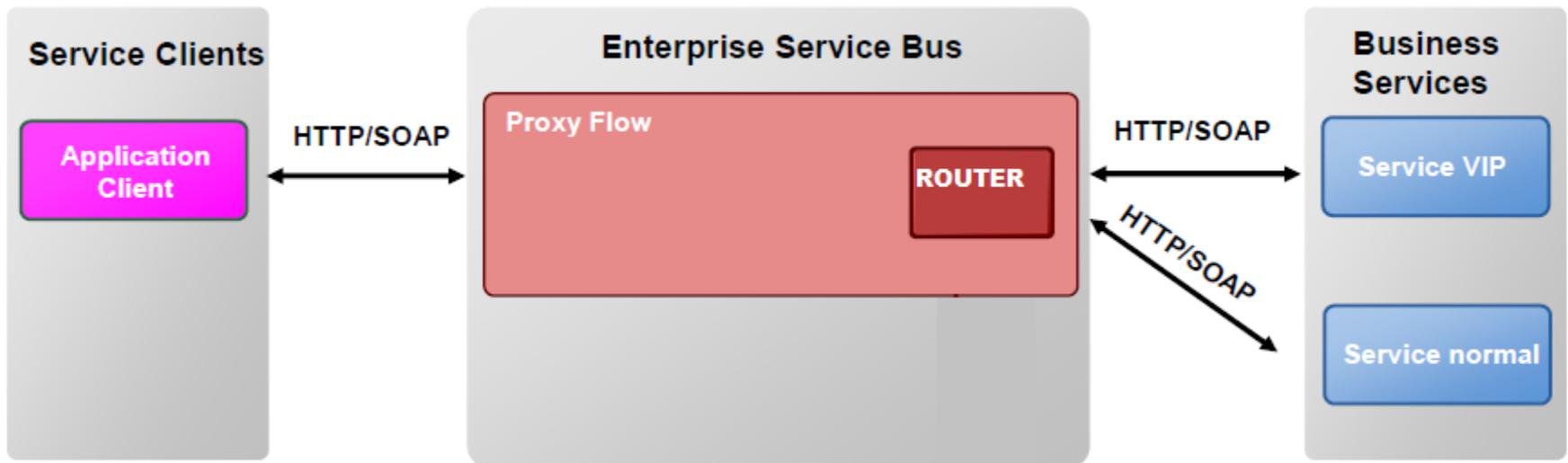


Ruteo Inteligente

- El ruteo inteligente permite determinar dinámicamente el destino de un mensaje (seleccionando de varios destinos posibles)

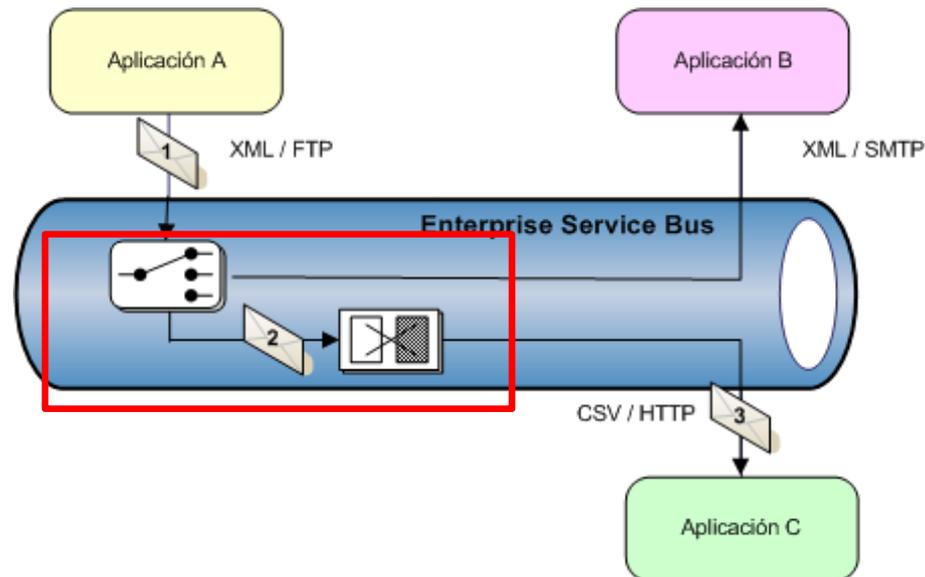


Ruteo Inteligente



Flujos de Mediación

- Los Flujos de Mediación (mediation flows, microflows) especifican una secuencia de operaciones de mediación (transformación, ruteo, etc) a ejecutar en un mensaje



Flujos de Mediación

- ❑ Están pensados para procesos simples y de corta duración, abordando principalmente problemáticas de integración y comunicación
- ❑ Los ESB permiten especificar estos flujos utilizando herramientas gráficas, archivos de configuración, DSLs



Sistema de Mensajería (MOM)

- Los ESBs en general se apoyan o incluyen sistemas de mensajería (MOMs)

- Posibles usos de un MOM en un ESB:
 - Garantía de Entrega
 - Mapeo entre estilos de interacción
 - Control de carga
 - Balanceo de carga



Monitoreo y Administración

- ❑ Los ESB incluyen funcionalidades de monitoreo que proveen valores para distintas métricas
 - tiempos de respuesta,
 - cantidad de mensajes procesados por servicio,
 - errores en la invocación de servicios, etc
- ❑ Esto permite detectar:
 - cuellos de botella,
 - incumplimiento de requerimientos de calidad de servicio, etc

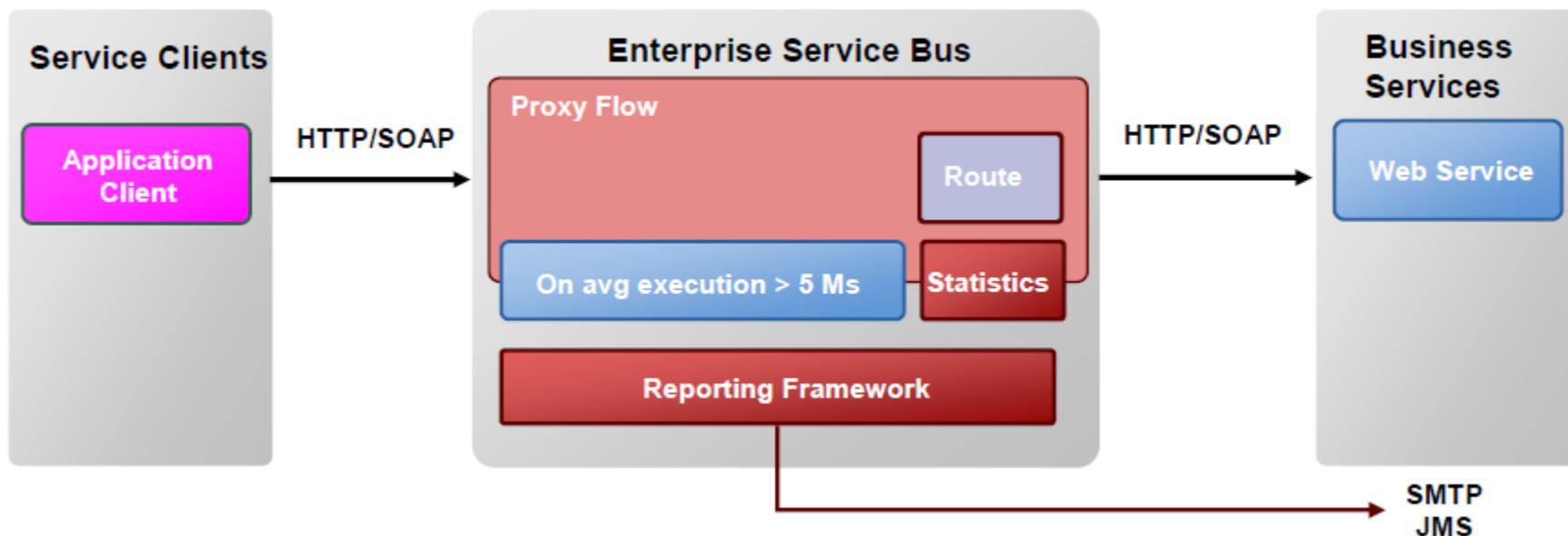


Monitoreo y Administración

- ❑ También pueden proveer la capacidad de enviar alertas si ciertos valores de esas métricas no son adecuados, por ejemplo:
 - si el tiempo de respuesta de un servicio es mayor a un determinado valor
 - si la cantidad de invocaciones a un servicio supera un determinado valor



Monitoreo y Administración



(Rischbeck, 2008)



Otras Características

- ❑ Seguridad
 - Autenticación, autorización, confidencialidad, soporte a estándares (WS-Security, etc)
- ❑ Calidad de Servicios
 - Transacciones
 - Confiabilidad
- ❑ Registro de Servicios
 - Publicación, descubrimiento, etc...
- ❑ Otras ver (Keen et al. 2004)(Chappell 2004)



Agenda

- Introducción

- Capacidades de un ESB

- Patrones ESB
 - **Mediación**, Conectividad y Despliegue



Patrones de Mediación

- ❑ Los mensajes que fluyen a través del ESB son generalmente procesados por flujos de mediación
- ❑ Si bien las operaciones de mediación no están restringidas en lo que pueden hacer, existen un conjunto de operaciones, y combinaciones que son comúnmente utilizadas y han sido documentadas como **Patrones de Mediación**



Patrones de Mediación

- Patrones de Mediación Simples
 - Patrones de Ruteo
 - Patrones de Transformación
 - Patrones de Administración
 - Otros Patrones

- Patrones de Mediación Compuestos



Patrones de Mediación: Ruteo

- ❑ Content-based router
- ❑ Context-based router
- ❑ Load-balancing router
- ❑ Splitter
- ❑ Recipient List
- ❑ Aggregator
- ❑ Message Filter



Patrones de Mediación: Transformación

- ❑ Transformación de Contenido
- ❑ Transformación de Modelos de Datos
- ❑ Transformación de Formato de Datos
- ❑ Enriquecimiento de Contenido
- ❑ Filtrado de Contenido

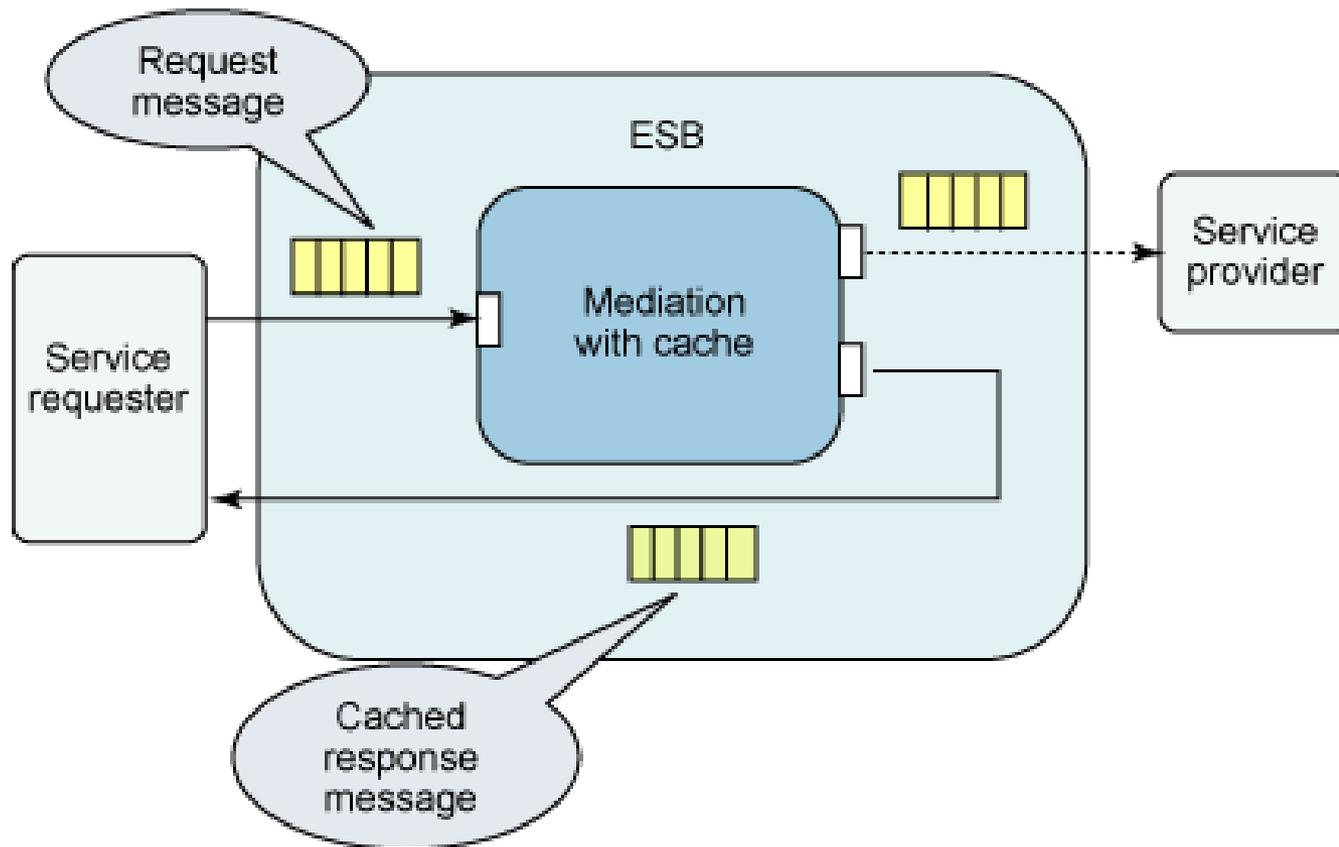


Otros Patrones: Cache

- ❑ El patrón Cache recibe un mensaje y retorna otro que fue previamente almacenado y enviado como respuesta al mensaje de solicitud.
- ❑ Constituye una solución reusable para mejorar la performance en un ambiente SOA



Cache Mediation Pattern



(Yan Fang et al. 2006)

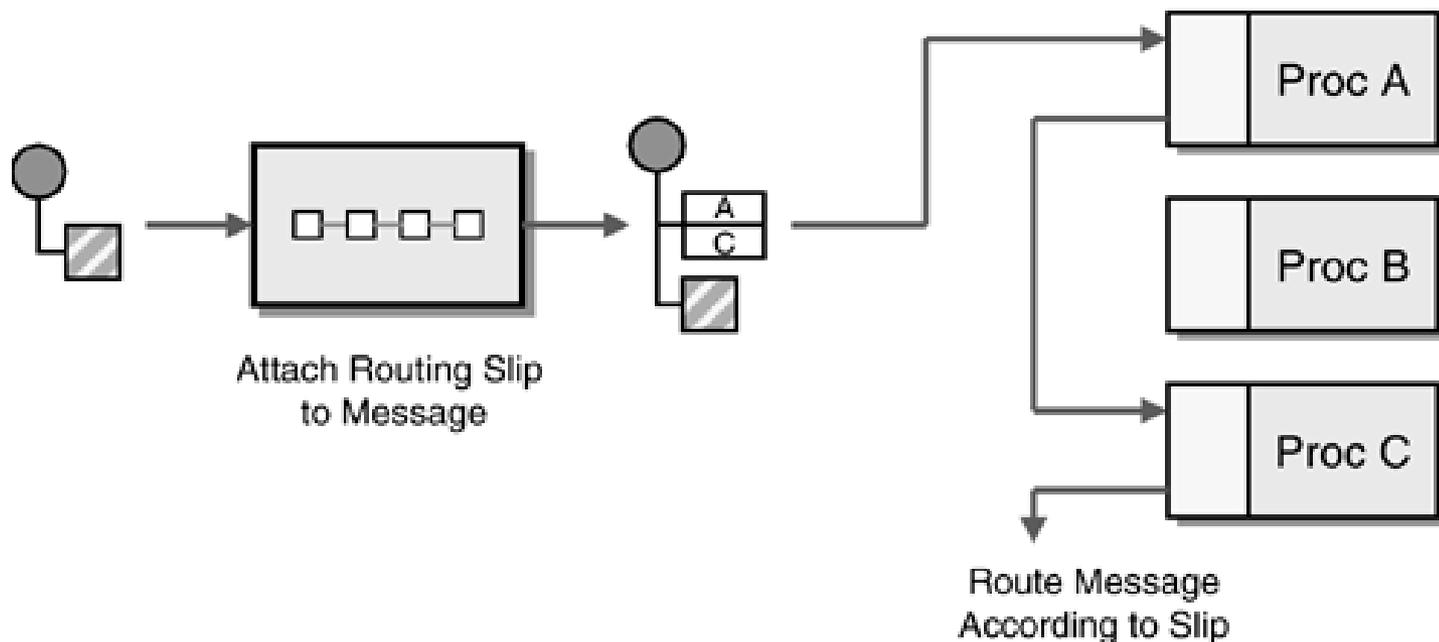
Patrones de mediación compuestos

- Ruteo basado en Itinerario
- Verify Enrich Transform Operate



Ruteo Basado en Itinerario

- El ruteo basado en itinerario determina el destino del mensaje en base a un itinerario.



(Hohpe and Woolf 2003)



Ruteo Basado en Itinerario

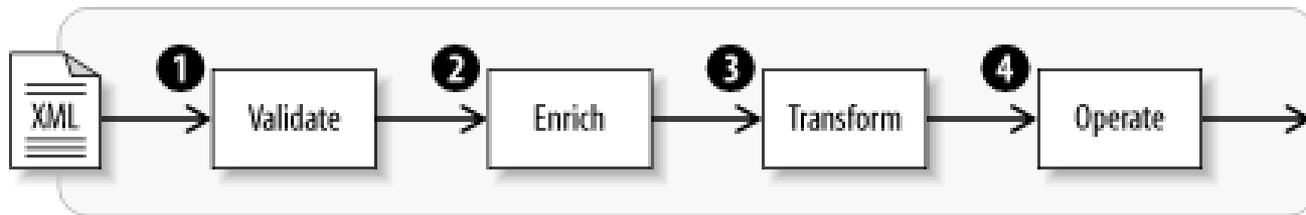
- ❑ El itinerario es metadata que se incluye en el propio mensaje y especifica al ESB el camino que debe seguir el mismo.
- ❑ Los itinerarios contribuyen a la naturaleza distribuida del ESB eliminando la dependencia de un motor de ruteo centralizado.
 - Mejora sobre los Broker EAI!



También se conoce como **routing slip**

VETO Pattern

- El patrón VETO (Validate, Enrich, Transform, Operate) es un patrón de integración que permite que se ruteen datos validados y consistentes a través del ESB



VETO Pattern

- ❑ Validate (Validar)
 - Valida que el mensaje cumpla con ciertos requerimientos (ej. usando un XML Schema o WSDL)
- ❑ Enrich (Enriquecer)
 - Implica agregar datos adicionales al mensaje para hacerlo más entendible por el servicio destino
- ❑ Transform (Transformar)
 - Transforma el mensaje al formato destino
- ❑ Operate (Operar)
 - Invoca la aplicación o servicio destino



Agenda

- Introducción

- Capacidades de un ESB

- Patrones ESB
 - Mediación, **Conectividad** y Despliegue



Patrones de Conectividad

- ❑ A través de la experiencia adquirida en varias implementaciones reales, se han identificado un conjunto de patrones de conectividad basados en ESB
- ❑ Estos apuntan a proveer un marco que:
 - asista en la toma de decisiones relativas a arquitectura e implementación
 - facilite la selección y utilización de productos
 - brinde un conjunto de enfoques de integración recomendados y probados

(Wylie, 2009)



Patrones de Conectividad

- ❑ Virtualización de Servicios
- ❑ Habilitación de Servicios
- ❑ Gateway
- ❑ Procesamiento de Archivos



Virtualización de Servicios

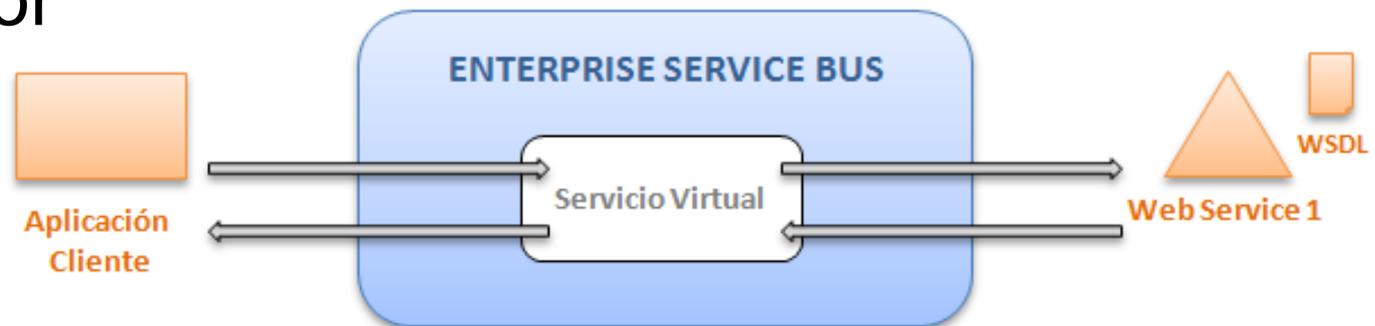
- ❑ Proveen una capa de procesamiento intermedio entre los servicios
- ❑ Esto permite abordar requerimientos de mediación (ruteo, transformación, etc) que surjan por necesidades de conectividad
- ❑ Los patrones en este grupo son:
 - Simple Service Proxy
 - Service Selector
 - Service Translator
 - Service Normalizer

(Wylie, 2009)



Simple Service Proxy

- ❑ El patrón **Simple Service Proxy** toma un servicio existente y crea un nuevo servicio virtual en el ESB
- ❑ De esta forma el servicio original se accede a través de un punto controlado y posiblemente utilizando un protocolo diferente al del proveedor



(Wylie, 2009)



Simple Service Proxy

- ❑ Este patrón introduce un punto de mediación en el ESB y es la base para patrones más elaborados
- ❑ Si bien en este caso simple las operaciones de mediación están limitadas a operaciones de monitoreo, es posible introducir funcionalidades de gestión:
 - manejo de errores, alertas,
 - gestión de tráfico, cargo por utilización de servicio
 - métricas de performance

(Wylie, 2009)



Simple Service Proxy

- ❑ Este patrón simple también da soporte a la Gobernanza de SOA
- ❑ En general se utiliza en conjunto con el requerimiento de que todos los servicios de una organización sean accedidos únicamente a través del ESB
- ❑ Este patrón también provee una forma de asegurar el acceso y ocultar la estructura interna de una organización
 - Provee un punto de partida para ofrecer acceso a los servicios desde fuera de la zona de seguridad de una organización

(Wylie, 2009)



Service Selector, Translator y Normalizer

- Estos patrones extienden el patrón anterior introduciendo:
 - Ruteos
 - Transformaciones
 - Ruteos y Transformaciones conjuntas



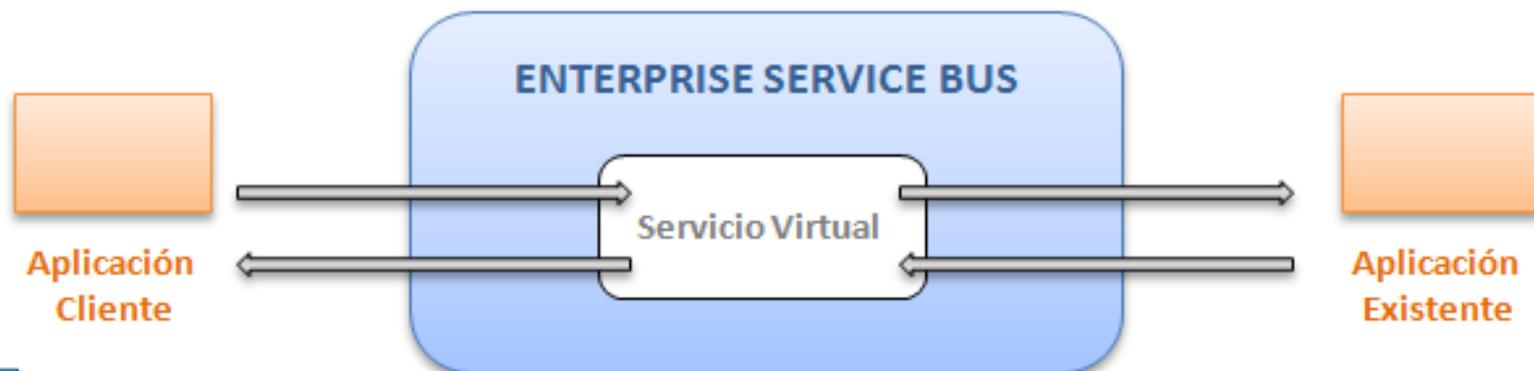
Patrones de Conectividad

- ❑ Virtualización de Servicios
- ❑ **Habilitación de Servicios**
- ❑ Gateway
- ❑ Procesamiento de Archivos



Habilitación de Servicios

- Abordan el problema de encapsular funcionalidades que no presentan una interfaz de servicio, y exponerlas en el ESB a través de una interfaz orientada a servicios.
 - Facilitan la migración hacia una SOA!



(Wylie, 2009)



Habilitación de Servicios

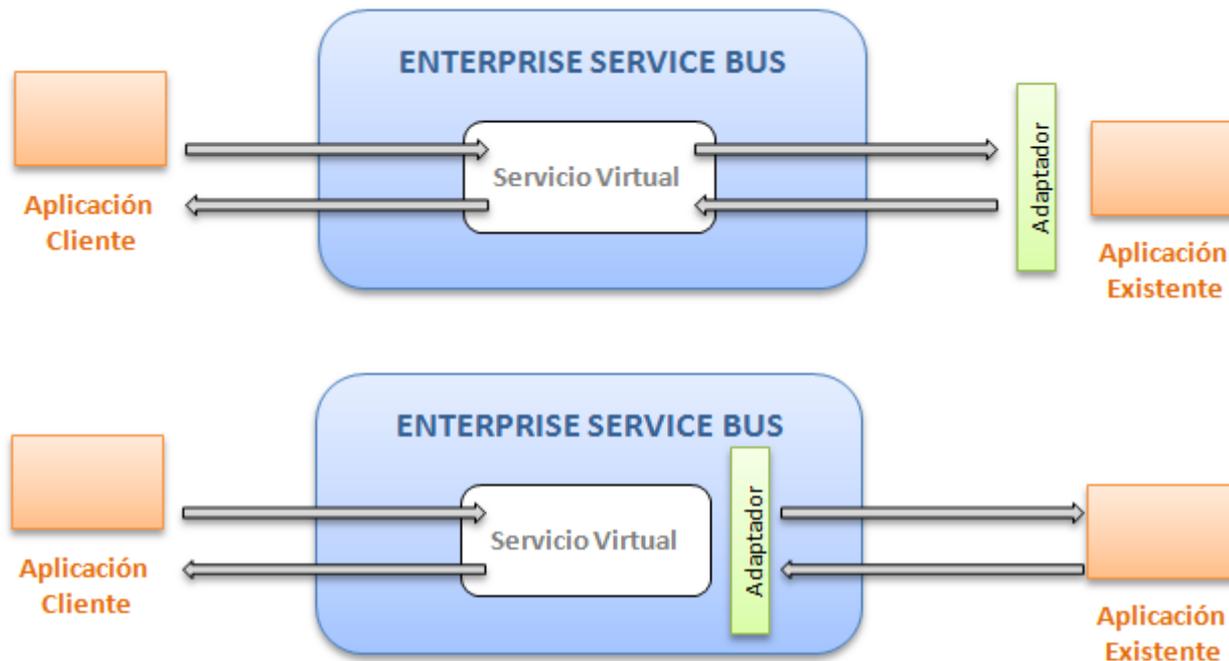
- ❑ Estos patrones permiten reutilizar los activos existentes de las organizaciones sin requerir cambios radicales
- ❑ Posibilitan además el acceso a servicios que se construyen en base a funcionalidades implementadas en diversas tecnologías
- ❑ En este grupo de patrones se distinguen:
 - Simple Service Facade
 - Complex Service Facade
 - Client Access

(Wylie, 2009)



Simple Service Facade

- Los adaptadores pueden ubicarse del lado de la aplicación o (más comúnmente en el ESB)



(Wylie, 2009)



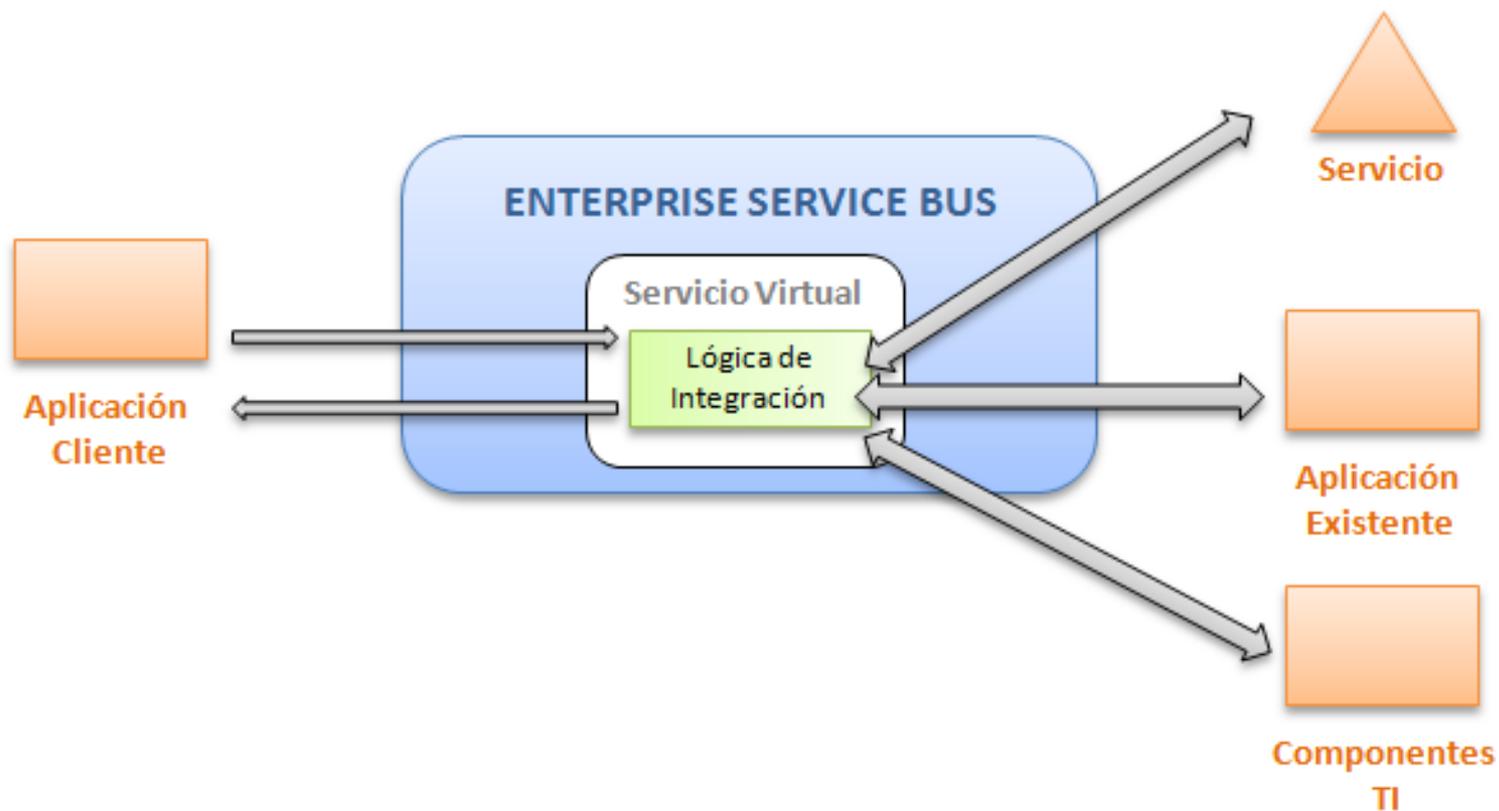
Simple Service Facade

- ❑ Este patrón es similar al patrón Servicio Proxy Simple, pero en este caso las solicitudes no son satisfechas por servicios, sino por aplicaciones existentes
- ❑ Las aplicaciones se acceden a través de mensajes, adaptadores u otras APIs
- ❑ Este patrón puede ser extendido a través de ruteos, transformaciones o normalizaciones.



(Wylie, 2009)

Complex Service Facade



(Wylie, 2009)



Complex Service Facade

- ❑ El patrón Complex Service Facade incorpora lógica de integración, en base a algunas de las capacidades nativas de los ESB
- ❑ Se utiliza generalmente para construir servicios de más alto nivel a partir de servicios, componentes o funcionalidades de granularidad más fina.

(Wylie, 2009)

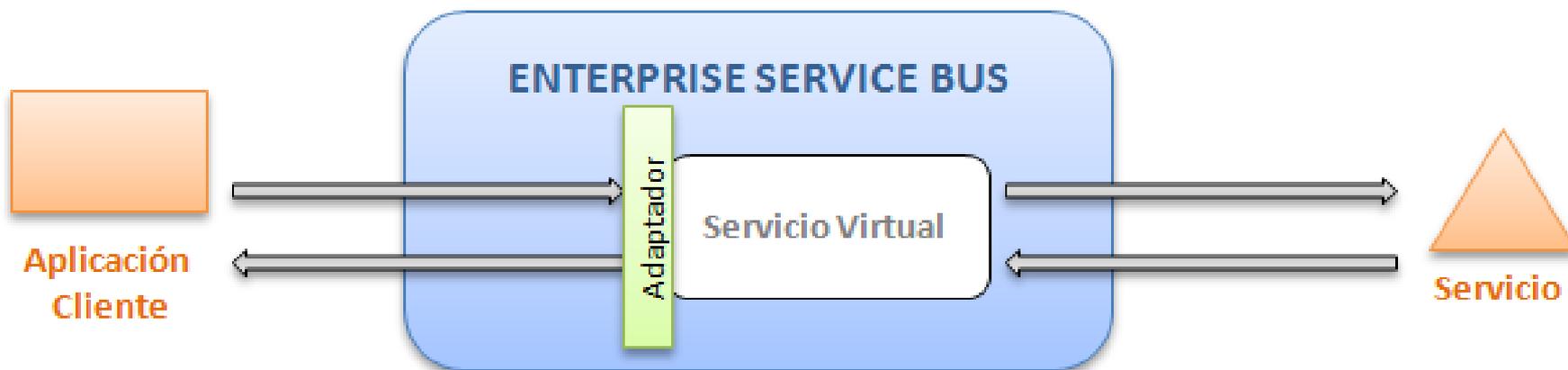


Complex Service Facade

- La lógica de integración se puede construir de varias formas, pero generalmente consiste en:
 - una secuencia de llamadas
 - el envío de múltiples solicitudes con la posterior agregación de resultados
 - la distribución de información.



Client Access



Client Access

- ❑ El patrón Client Access provee puntos de entrada para clientes que no tienen la capacidad de acceder a interfaces orientadas a servicios
- ❑ Esto en general se realiza a través de adaptadores
- ❑ Este patrón permite a aplicaciones empaquetadas existentes, participar en una arquitectura empresarial que incorpora el enfoque SOA

(Wylie, 2009)

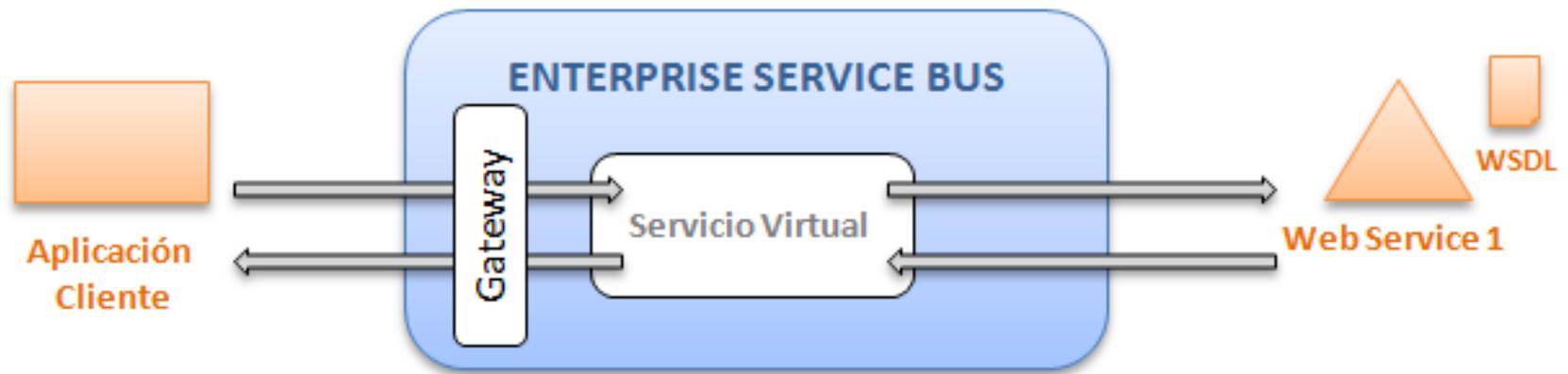


Patrones de Conectividad

- ❑ Virtualización de Servicios
- ❑ Habilitación de Servicios
- ❑ Gateway
- ❑ Procesamiento de Archivos



Gateway



Gateway

- Un Gateway es una parte del ESB que provee funciones de borde, las cuales se aplican a todos los mensajes entrantes al ESB.
- Luego de aplicarse un patrón Gateway, se puede ejecutar otro de los patrones o llamar directamente a un servicio.



(Wylie, 2009)

Security Gateway

- ❑ Este tipo de gateway separa todo el procesamiento “no estándar” de seguridad de la infraestructura principal
- ❑ Puede ejecutar tareas de autenticación, autorización y posiblemente auditoría
- ❑ En este caso siempre existe una comunicación confiable entre el Gateway y el resto de los componentes del ESB
- ❑ Permite simplificar los temas de seguridad en



el ESB

(Wylie, 2009)

Patrones de Conectividad

- ❑ Virtualización de Servicios
- ❑ Habilitación de Servicios
- ❑ Gateway
- ❑ **Procesamiento de Archivos**



Procesamiento de Archivos

- ❑ Un ESB puede proveer un ambiente de ejecución administrado para el procesamiento de archivos (locales o vía FTP)
- ❑ Los patrones para el procesamiento de archivos involucran actividades como:
 - transformación de datos de los archivos
 - división de un archivo en registros individuales
 - ruteo de registros
 - acumulación de registros en archivos
 - ruteo de archivos a determinados procesadores

(Wyllie, 2009)



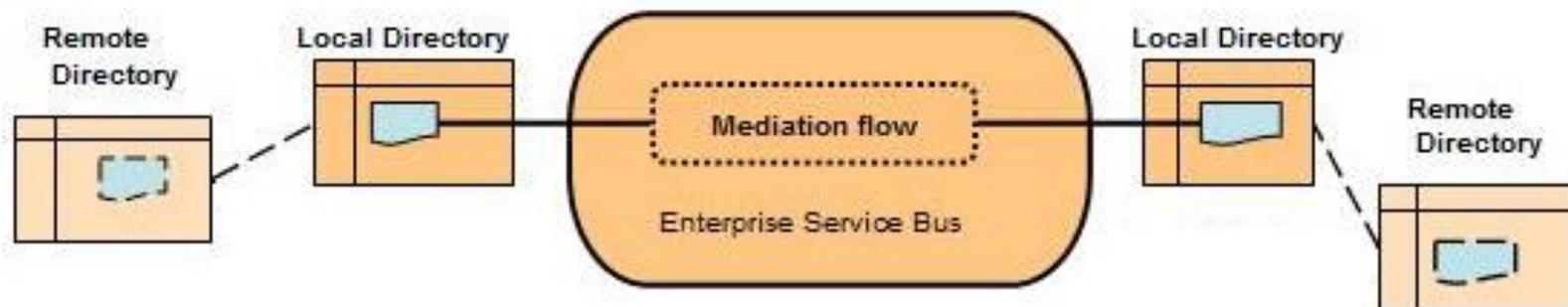
Procesamiento de Archivos

- ❑ File Record Processor
- ❑ Record Batcher
- ❑ File Record Distributor

(Wylie, 2009)



File Record Processor



(Wylie, 2009)



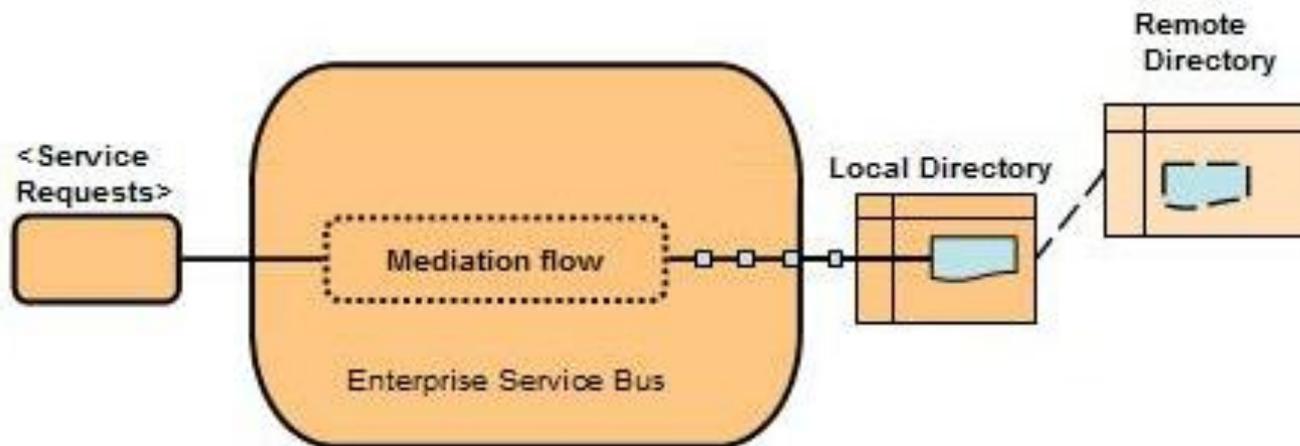
File Record Processor

- ❑ Se procesa cada registro de un archivo y acumula los resultados en uno o más archivos transitorios
- ❑ Luego de completada la operación, estos archivos serán movidos a directorios destino
- ❑ Soporte a procesamiento batch

(Wylie, 2009)



Record Batcher



(Wylie, 2009)



Record Batcher

- ❑ Este patrón recibe datos como solicitudes de servicios (p.ej: SOAP), mensajes o a través de adaptadores y luego de procesarlos genera archivos temporales que, una vez completos, son movidos a directorios destinos

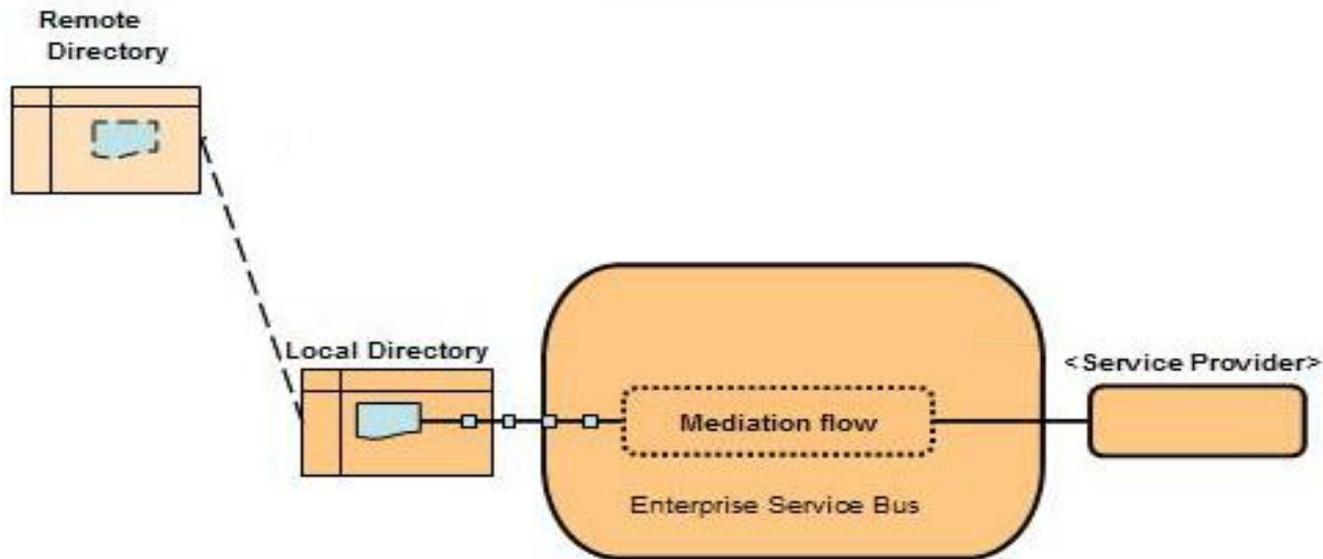


Record Batcher

- ❑ Este patrón da soporte a interacciones entre sistema más nuevos y sistemas legados que únicamente soportan interfaces tipo batch
- ❑ Este patrón se puede utilizar como un primer paso para integrar sistemas legados a una arquitectura orientada a servicios



File record Distributor



(Wylie, 2009)

File record Distributor

- Este patrón divide un archivo en registros y utiliza cada registro para realizar llamadas a servicios o para pasar un mensaje



Agenda

- Introducción

- Capacidades de un ESB

- Patrones ESB
 - Mediación, Conectividad y **Despliegue**



Patrones de Despliegue

- ❑ Principal diferencia con Brokers EAI
- ❑ Posibilidad de despliegue de varios ESB en una misma organización
 - Por sector, región, fusión de empresas, etc.



Patrones de Despliegue

- ❑ Para empresas pequeñas, un único ESB con un único registro podría ser suficiente
- ❑ Para empresas de gran tamaño es necesario considerar patrones de despliegue más complejos que involucren varios ESBs con varios registros



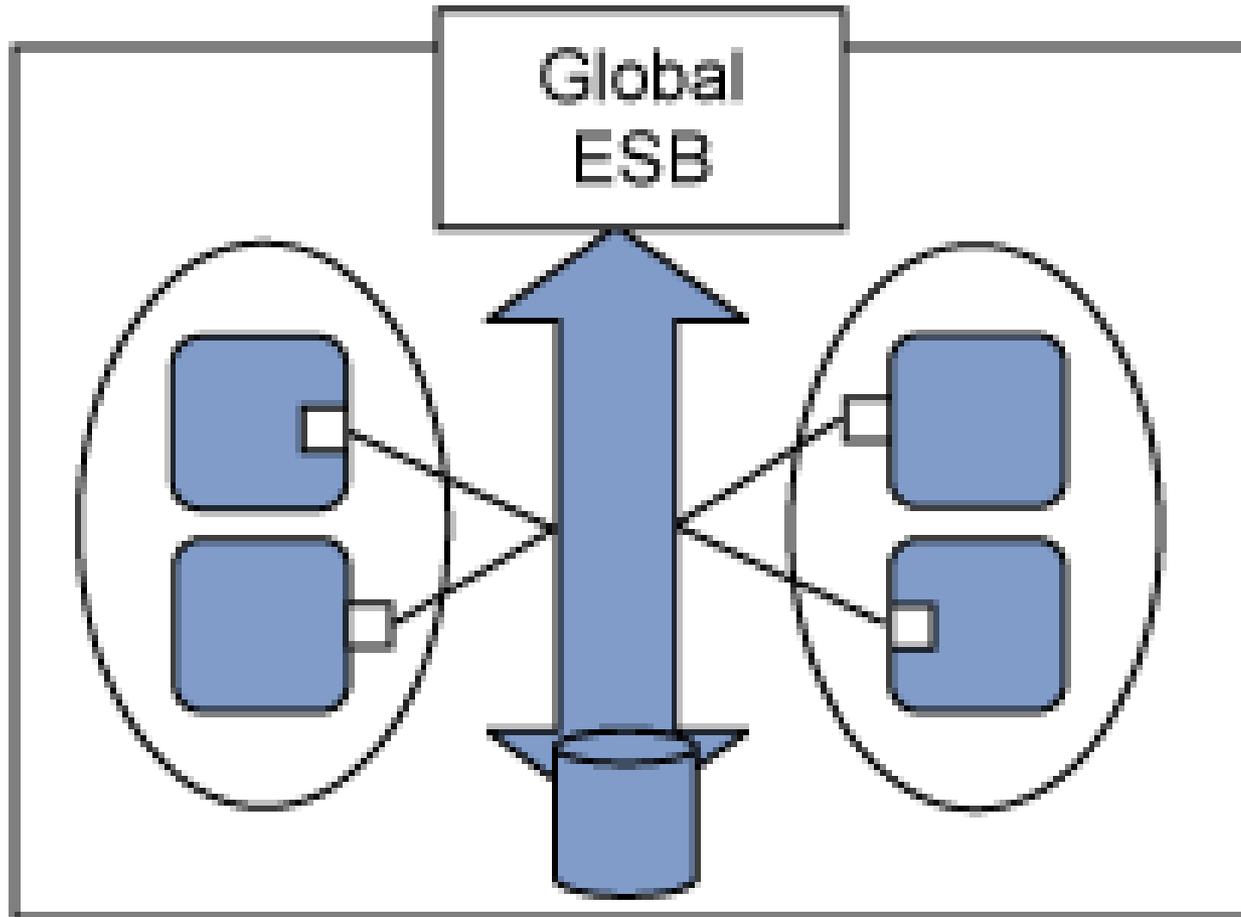
Patrones de Despliegue

- Existen básicamente cuatro configuraciones conocidas para desplegar ESBs:
 - Global ESB
 - Directly Connected ESBs
 - Brokered ESBs
 - Federated ESBs

(Keen et al. 2005) (Hutchison et al. 2005) (Nott and Stockton 2006)



Global ESB



(Keen et al. 2005) (Hutchison et al. 2005) (Nott and Stockton 2006)



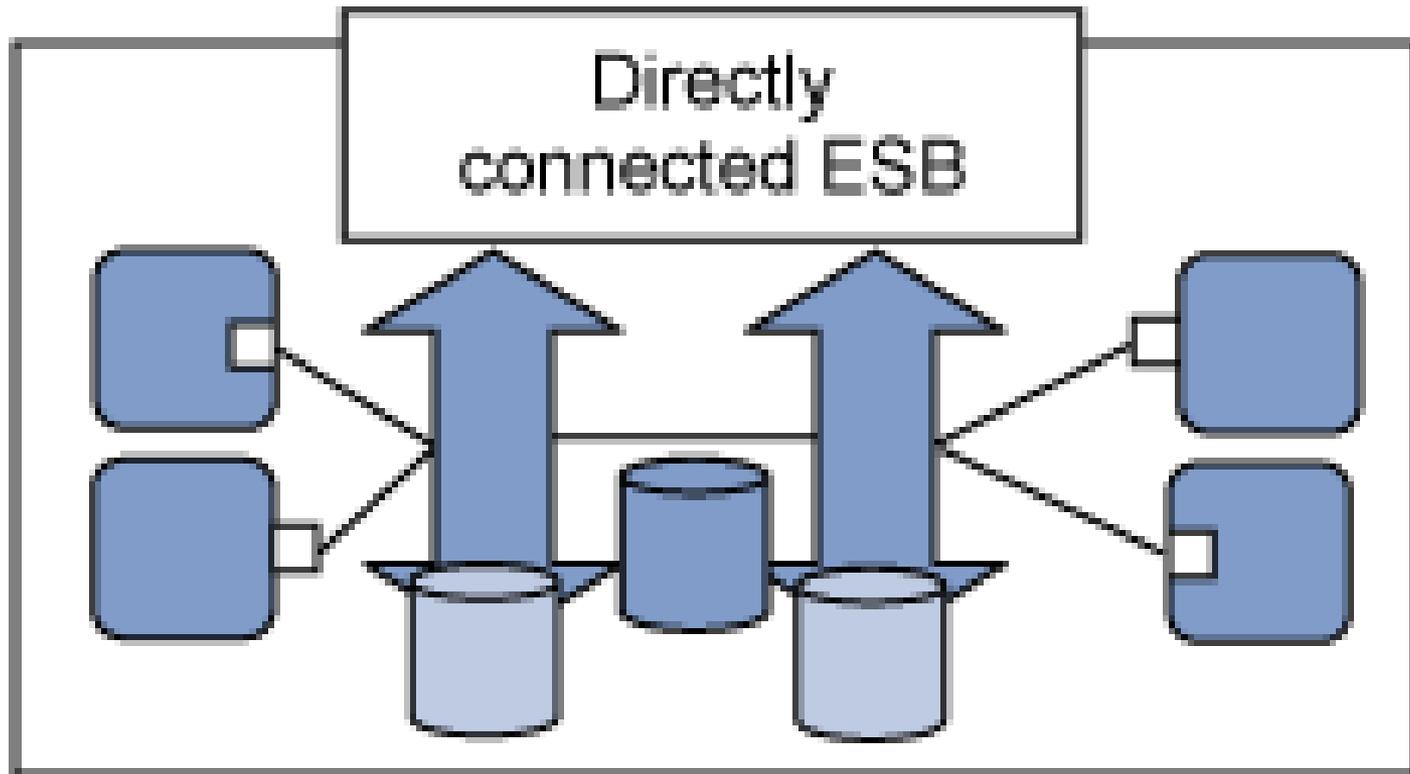
Global ESB

- ❑ En este patrón hay un único ESB, con un único registro, para toda la organización
- ❑ Esta configuración es adecuada para una empresas o departamentos pequeños
- ❑ Los servicios están disponibles y son visibles para todos los participantes

(Keen et al. 2005) (Hutchison et al. 2005) (Nott and Stockton 2006)



Directly Connected ESBs



(Keen et al. 2005) (Hutchison et al. 2005) (Nott and Stockton 2006)



Directly Connected ESBs

- ❑ Todos los servicios de los ESB interconectados están visibles a través de un registro común de servicios.
 - En este registro se encuentra la relación de cada servicio con uno de los ESBs
- ❑ Un cliente llama a su ESB local, que consulta el registro común y pasa la solicitud al ESB destino
- ❑ Sin importar el número de ESBs, esta topología involucra interacciones entre dos ESBs que están directamente conectados



(Keen et al. 2005) (Hutchison et al. 2005) (Nott and Stockton 2006)

Directly Connected ESBs

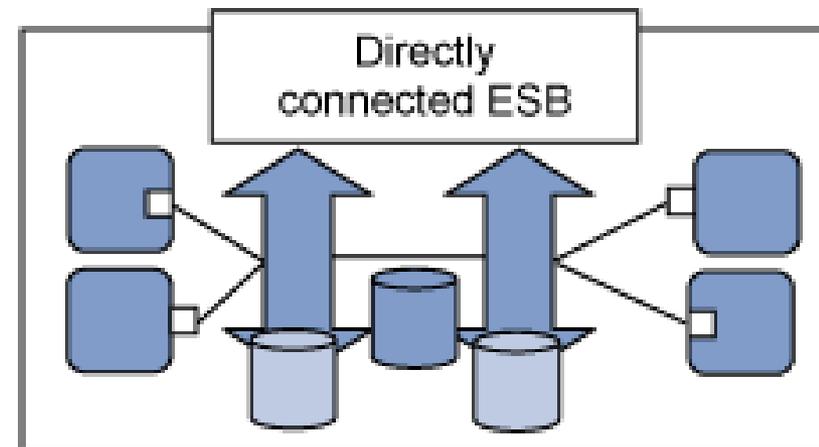
- ❑ Este patrón es adecuado para organizaciones con múltiples divisiones
- ❑ Los servicios que provee cada división se dejan disponibles a toda la organización listándolos en un registro común
- ❑ Cada división se encarga de la seguridad de su ESB y regula el acceso a sus servicios
- ❑ También se debería controlar el acceso al registro de servicios

(Keen et al. 2005) (Hutchison et al. 2005) (Nott and Stockton 2006)



Directly Connected ESBs

- Este patrón soporta la virtualización de servicios de la siguiente manera:
 - Para invocar servicios de otra unidad de negocio, se define un servicio virtual en el ESB local el cual encapsula la búsqueda en el registro global y el ruteo al ESB correspondiente
 - De esta forma, una solicitud de un cliente en una división puede invocar servicios de otra división conectando de forma directa los dos ESBs

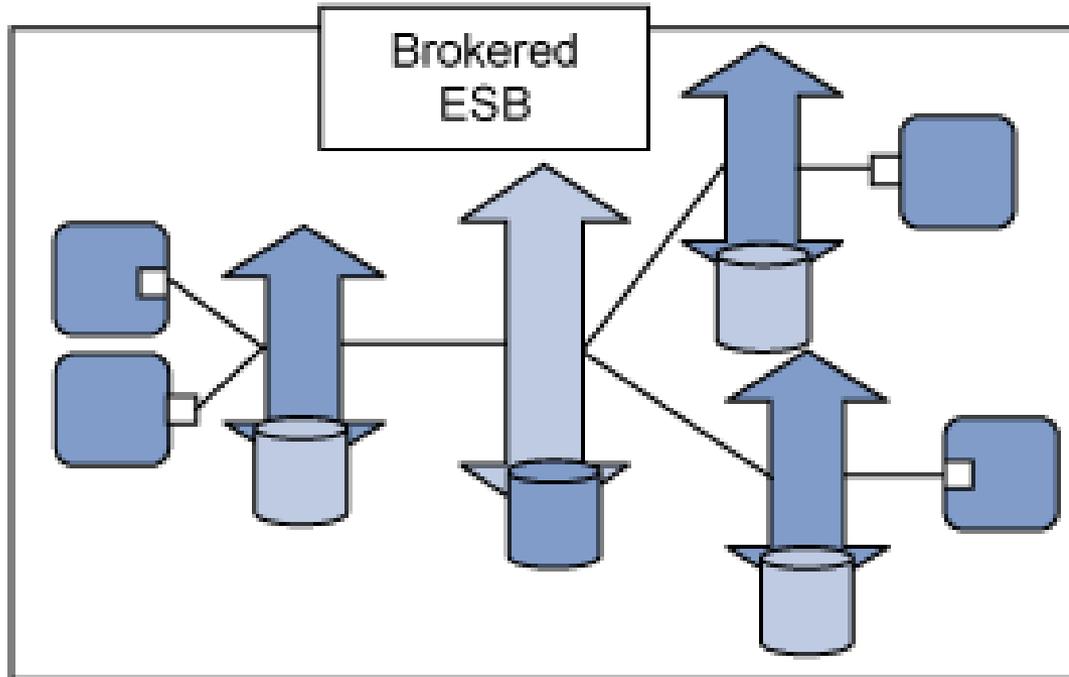


(Keen et al. 2005) (Hutchison et al. 2005) (Nott and Stockton 2006)



Brokered ESBs

- Este patrón utiliza un ESB broker para mediar entre los ESBs conectados

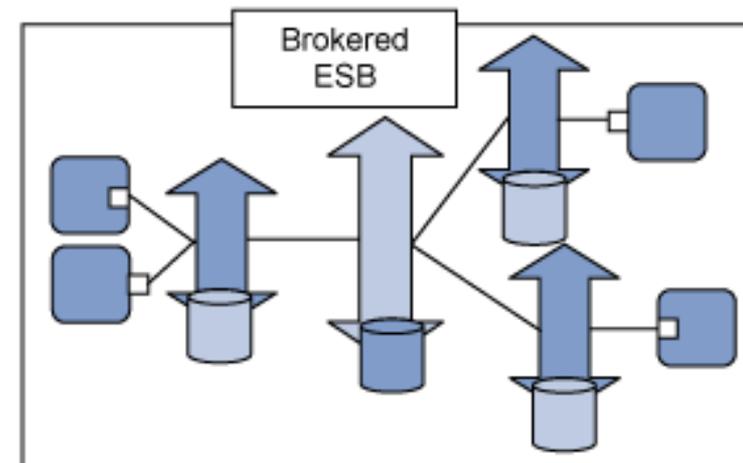


(Keen et al. 2005) (Hutchison et al. 2005) (Nott and Stockton 2006)



Brokered ESBs

- ❑ Es adecuado para organizaciones con múltiples divisiones
- ❑ Permite que servicios locales estén disponibles de forma global publicándolos centralmente
- ❑ Sin embargo en este caso, la lógica de intermediación entre los ESB está centralizada

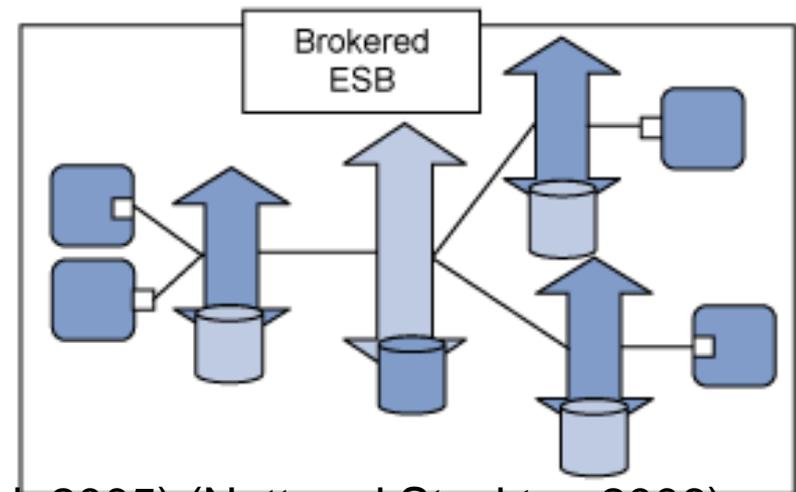


(Keen et al. 2005) (Hutchison et al. 2005) (Nott and Stockton 2006)



Brokered ESBs

- En este caso el flujo de invocación de un servicio de otro ESB es el siguiente:
 1. un cliente solicita un servicio a su ESB local
 2. la solicitud se rutea al ESB broker
 3. el broker efectúa una segunda mediación enviando la solicitud al ESB destino

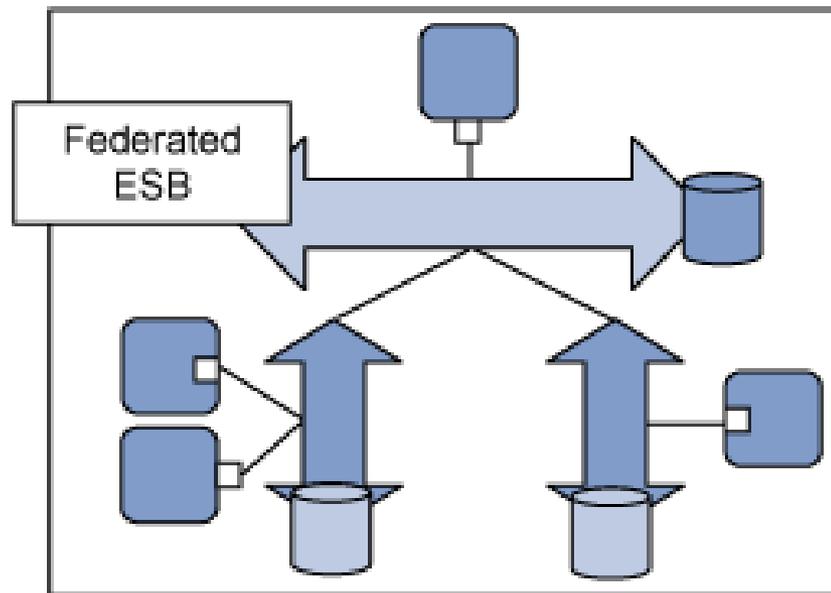


(Keen et al. 2005) (Hutchison et al. 2005) (Nott and Stockton 2006)



Federated ESBs

- En este patrón existe un “ESB maestro” (master ESB) al cual se conectan varios “ESBs esclavos” (slaves ESBs)



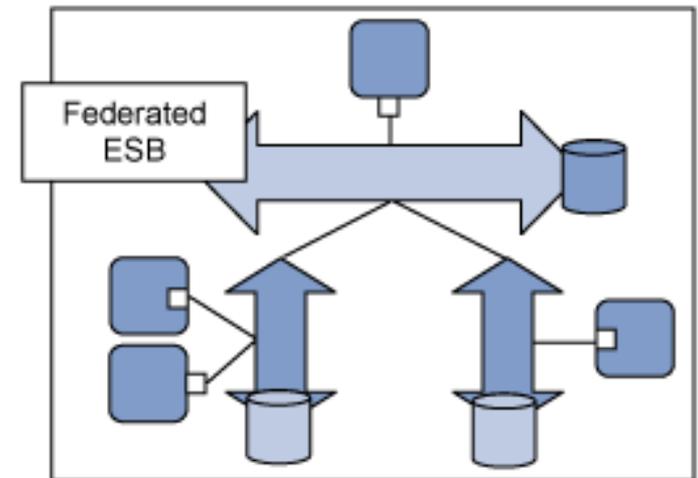
(Keen et al. 2005) (Hutchison et al. 2005) (Nott and Stockton 2006)



Federated ESBs

- ❑ Los consumidores y proveedores de servicios se conectan tanto al ESB maestro como a los ESBs esclavos
- ❑ Este patrón es utilizado por organizaciones con departamentos moderadamente autónomos que están bajo la supervisión de otro departamento

(Keen et al. 2005) (Hutchison et al. 2005)
(Nott and Stockton 2006)



Resumen

- ❑ Motivación de uso de un ESB
 - Arquitectura accidental
- ❑ Repasamos principales capacidades de mediación
 - Ruteo, transformación, conectores
- ❑ Patrones de ESB
 - Mediación, Conectividad y despliegue



Referencias

- ❑ Chappell, David. 2004. Enterprise Service Bus: Theory in Practice. O'Reilly Media.
- ❑ Davis, Jeff. 2009. Open Source Soa. 1st ed. Manning Publications.
- ❑ Endrei, Mark et al. 2004. “IBM Redbooks. Patterns: Service-Oriented Architecture and Web Services.”
- ❑ Erl, Thomas. 2009. SOA Design Patterns. 1st ed. Prentice Hall PTR.



Referencias

- ❑ Hohpe, Gregor, and Bobby Woolf. 2003. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Professional.
- ❑ Hutchison, B., M.-T. Schmidt, Dan Wolfson, and Marcia Stockton. 2005. “SOA programming model for implementing Web services, Part 4: An introduction to the IBM Enterprise Service Bus.”
- ❑ Keen, Martin et al. 2004. “IBM Redbooks. Patterns: Implementing an SOA using an Enterprise Service Bus.”



Referencias

- ❑ Keen, Martin et al. 2005. “IBM Redbooks | Patterns: Integrating Enterprise Service Buses in a Service-Oriented Architecture.”
- ❑ Nott, Chris, and Marcia Stockton. 2006. Choose an ESB topology to fit your business model.
- ❑ Papazoglou, Michael. 2007. Web Services: Principles and Technology. 1st ed. Prentice Hall.



Referencias

- ❑ Rademakers, Tijs, and Jos Dirksen. 2008. Open-Source ESBs in Action: Example Impl. in Mule and ServiceMix. 1st ed. Manning Publications.
- ❑ Rischbeck Thomas, Real Life ESB. 2008.
<http://www.slideshare.net/soasymposium/thomas-rischbeck-real-life-e-s-b>
- ❑ Roshen, SOA-Based Enterprise Integration: A Step-by-Step Guide to Services-based Application. McGraw-Hill Osborne Media, 2009.



Referencias

- ❑ Wylie, H., & Lambros, P. (2009, March 10). Enterprise Connectivity Patterns: Implementing integration solutions with IBM's Enterprise Service Bus products. CT316.
- ❑ Yan Fang, Rao et al. 2006. "Cache mediation pattern specification: an overview."

