

Seminario 7: Introducción a ESB Sistemas Distribuidos

Apuntes creados por Juan Boubeta y modificados por Antonio Balderas,
Pablo García y Salvador Gutiérrez

Departamento de Ingeniería Informática
Universidad de Cádiz



Escuela Superior de Ingeniería
Dpto. de Ingeniería Informática



Curso 2018 – 2019

- 1 Introducción
- 2 Funcionalidades de un ESB
- 3 Algunos ESB de código abierto

Sección 1 | Introducción

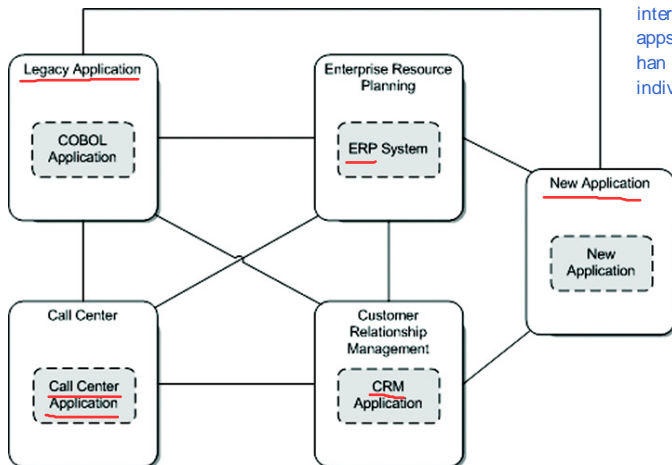
Definición de *Enterprise Service Bus* (ESB)

- Elemento de integración en las arquitecturas orientadas a servicios o *Service-Oriented Architecture* (SOA).
- Al igual que con WS-BPEL, ESB se puede utilizar como una plataforma de integración que permite exponer recursos y aplicaciones como servicios.
- Permite la interacción entre aplicaciones heterogéneas desde las más modernas hasta las más convencionales (*legacy*).
- En definitiva, ESB ofrece soluciones a problemas de integración cuando se quieren proporcionar servicios a los clientes.

¿Necesitamos un ESB? (I)

Arquitectura punto-a-punto: ¿qué ocurre al incorporar una nueva aplicación?

Apps y servicios se comunican entre sí.

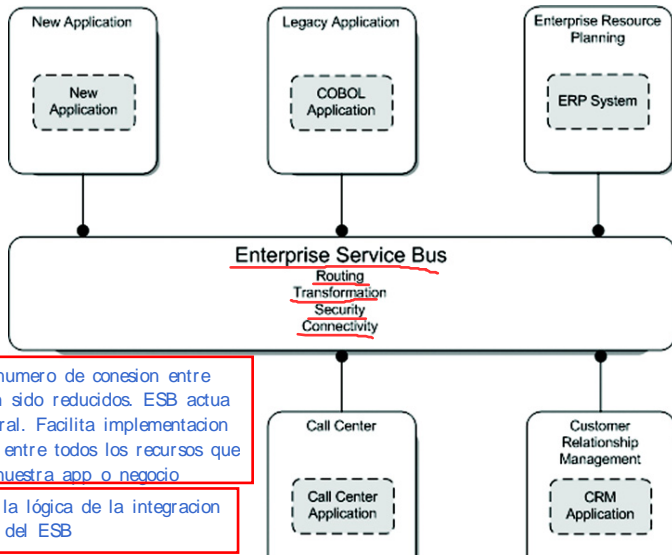


interacciones entre apps son las que se han definido de forma individualizada.

Fuente: [Rademakers & Dirksen]

¿Necesitamos un ESB? (II)

SÍ: Arquitectura que utiliza un ESB para integrar las aplicaciones



con el ESB el numero de conesion entre aplicaciones han sido reducidos. ESB actua como nexo central. Facilita implementacion y comunicacion entre todos los recursos que intervienen en nuestra app o negocio

Es compleja en la lógica de la integracion que está dentro del ESB

¿Necesitamos un ESB? (III)

Ventajas de un ESB

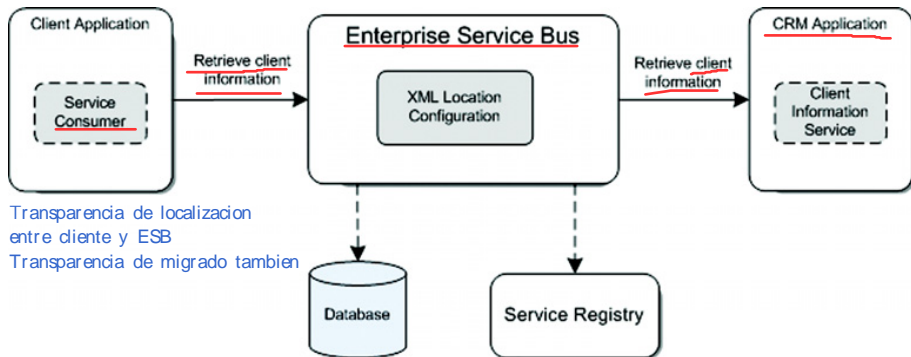
- Facilita la integración de aplicaciones.
- Ideal para trabajar en entornos heterogéneos: diferentes tecnologías y protocolos.
- Reduce el coste total de la gestión y el mantenimiento.

Sección 2 | Funcionalidades de un ESB

Funcionalidades de un ESB (I)

Transparencia de localizacion

Desacoplamiento entre el consumidor y el proveedor de servicio.

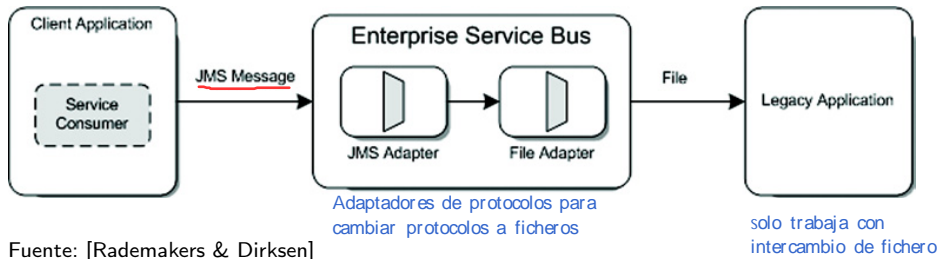


Fuente: [Rademakers & Dirksen]

Funcionalidades de un ESB (II)

Conversión de protocolos de transporte

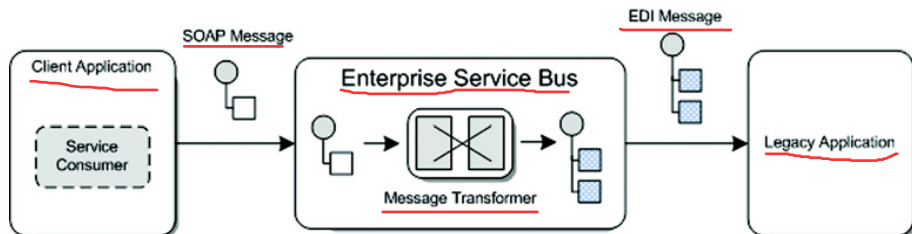
Integración de aplicaciones con diferentes protocolos de transporte:
HTTP(S) a JMS, SMTP a TCP...



Funcionalidades de un ESB (III)

Transformación de mensajes

De un formato a otro, utilizando estándares abiertos como XSLT y XPath.

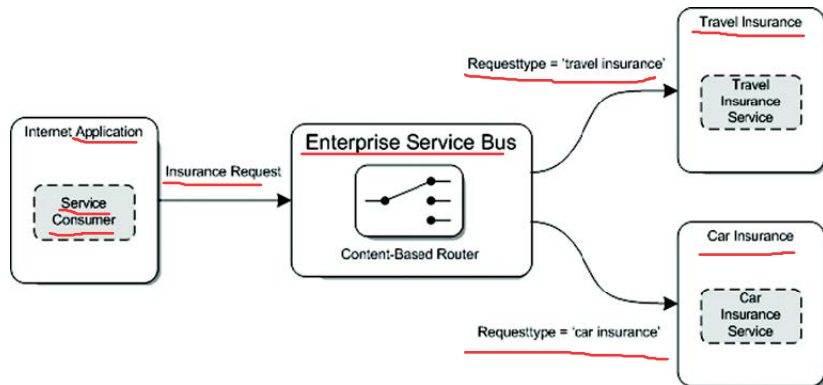


Fuente: [Rademakers & Dirksen]

Funcionalidades de un ESB (IV)

Encaminamiento de mensajes

Elección del destino de cada mensaje.

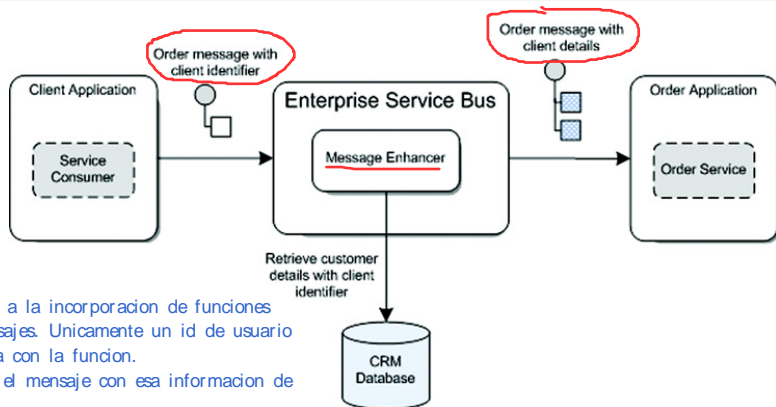


Fuente: [Rademakers & Dirksen]

Funcionalidades de un ESB (V)

Enriquecimiento de mensajes

Incorporación de información extra a los mensajes.



Referencia a la incorporación de funciones a los mensajes. Únicamente un id de usuario se asocia con la función.

Enriquece el mensaje con esa información de la función

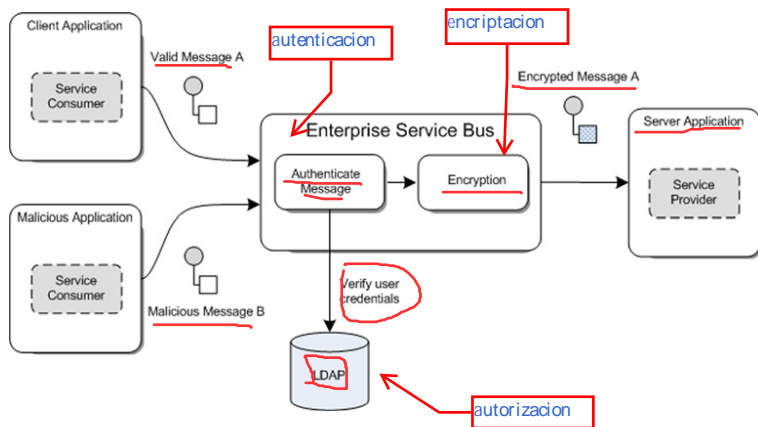
A partir del id pasado, proporciona la información al mensaje

Fuente: [Rademakers & Dirksen]

Funcionalidades de un ESB (VI)

Seguridad

Autenticación, autorización y encriptación (con clave pública del 'receptor').

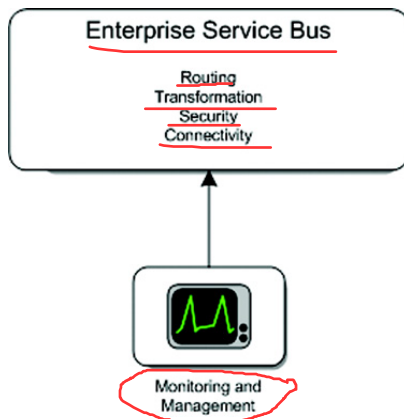


Fuente: [Rademakers & Dirksen]

Funcionalidades de un ESB (VII)

Administración y monitorización

Configuración del ESB y monitorización de los mensajes en tiempo de ejecución.



Sección 3 | Algunos ESB de código abierto

Algunos ESB de código abierto (I)


- Apache ServiceMix: <http://servicemix.apache.org>
- Apache Tuscany: <http://tuscany.apache.org/>
- JBoss ESB: <http://www.jboss.org/jbossesb/>
- Mule: <http://www.mulesoft.org/>
- Open ESB: <http://www.open-esb.net/>
- Petals ESB: <http://petals.ow2.org/>
- Spring Integration:
<http://www.springsource.org/spring-integration>
- WSO2 ESB:
<http://wso2.com/products/enterprise-service-bus/>

Algunos ESB de código abierto (II)


Rademakers y Dirksen consideran Mule como el mejor ESB atendiendo a los siguientes criterios:

- Incorporación de todas las funcionalidades relevantes de un ESB.
- Bien documentado.
- Visibilidad en el mercado.
- Desarrollo activo y respaldado por una comunidad de software.
- Flexibilidad y extensibilidad mediante una lógica personalizada.
- Soporte para una gran cantidad de protocolos de transporte y opciones de conectividad.
- Integración con otros proyectos de código abierto.
- Proporción de un IDE: MuleStudio.

Referencias bibliográficas I

 M.P. Papazoglou
Web Services & SOA: Principles and Technology.
Pearson – Prentice Hall, 2012.

 T. Rademakers & J. Dirksen
Open Source ESBs In Action
Manning, 2009.

 J. Boubeta Puig, G. Ortiz e I. Medina Bulo
Procesamiento de Eventos Complejos en Entornos SOA: Caso de
Estudio para la Detección Temprana de Epidemias
Jornadas de Ciencia e Ingeniería de Servicios (JCIS), 2011.

Referencias bibliográficas II



M. Edwards et al.

Un modelo conceptual para los sistemas de procesamiento de eventos (2010)

www.ibm.com/developerworks/ssa/webservices/library/ws-eventprocessing



J. L. Maréchaux

Combining Service-Oriented Architecture and Event-Driven Architecture using an Enterprise Service Bus (2006)

www.ibm.com/developerworks/webservices/library/ws-soa-eda-esb/index.html