

# Una Introducción al *Enterprise Service Bus*

## Sistemas Distribuidos

Juan Boubeta Puig

Editado por: Antonio Balderas Alberico y Pablo García Sánchez

Grupo UCASE de Ingeniería del Software  
Departamento de Ingeniería Informática

Marzo de 2018



# Índice

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



# Índice

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

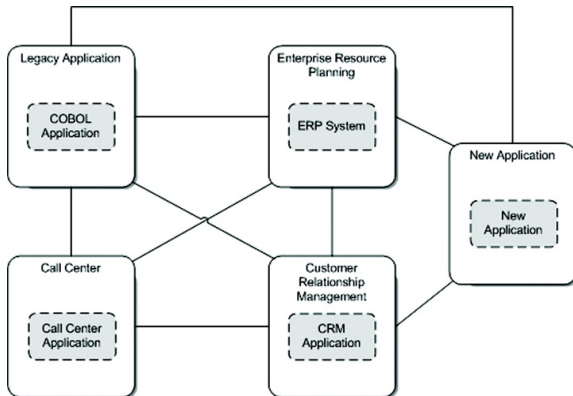


# Definición de *Enterprise Service Bus* (ESB)

- **Elemento de integración** (multiprotocolo y multipropósito) en las arquitecturas orientadas a servicios o *Service-Oriented Architecture* (SOA).
- Combina servicios Web, mensajería, transformación, encaminamiento y enriquecimiento de datos, políticas de seguridad, entre otros.
- Integra los enfoques dirigido por eventos (EDA) y orientado a servicios (SOA).
- Un servicio desplegado en un ESB puede ser lanzado por un consumidor o un evento.
- Permite la interacción entre aplicaciones heterogéneas desde las más modernas hasta las más convencionales (*legacy*).

# ¿Necesitamos un ESB? (I)

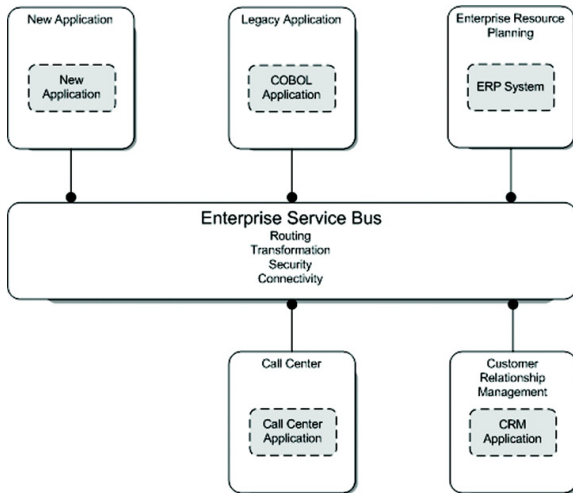
Arquitectura punto-a-punto: ¿qué ocurre al incorporar una nueva aplicación?



Fuente: [Rademakers & Dirksen]

# ¿Necesitamos un ESB? (II)

Sí: Arquitectura que utiliza un ESB para integrar las aplicaciones



Fuente: [Rademakers & Dirksen]

# ¿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.

UCA

Universida

# Índice

1 Introducción

2 Funcionalidades de un ESB

3 Algunos ESB de código abierto

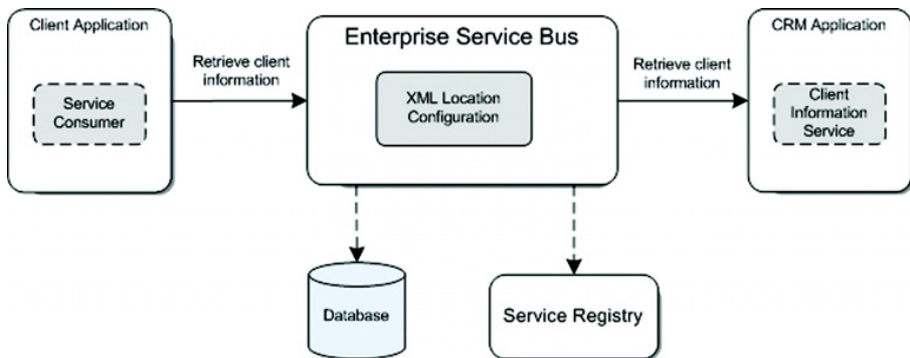




# Funcionalidades de un ESB (I)

## Transparencia de localizacion

Desacoplamiento entre el consumidor y el proveedor de servicio.

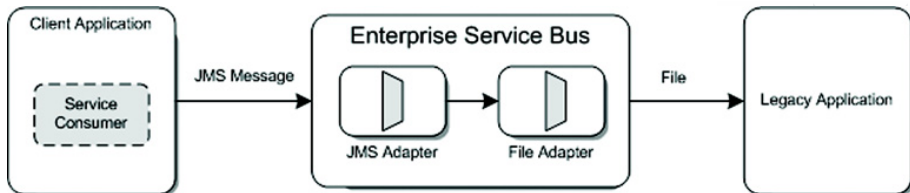


Fuente: [Rademakers & Dirksen]

## 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...

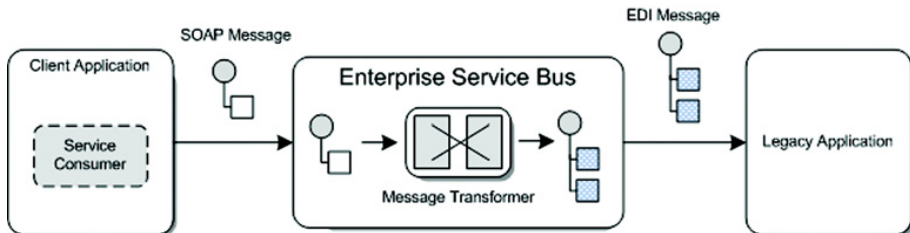


Fuente: [Rademakers & Dirksen]

## Funcionalidades de un ESB (III)

### Transformación de mensajes

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

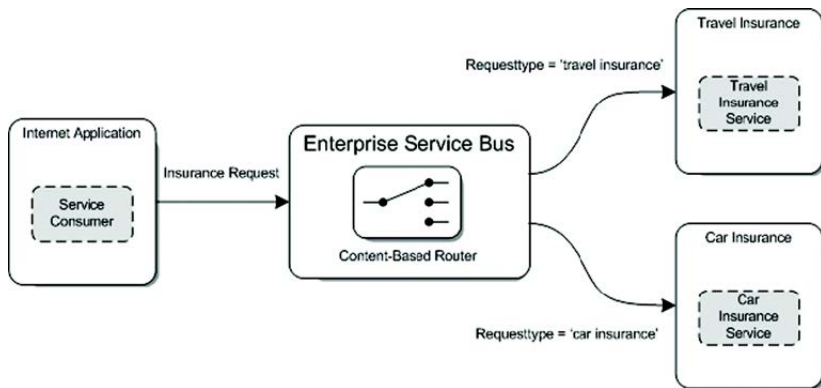


Fuente: [Rademakers & Dirksen]

# Funcionalidades de un ESB (IV)

## Encaminamiento de mensajes

Elección del destino de cada mensaje.

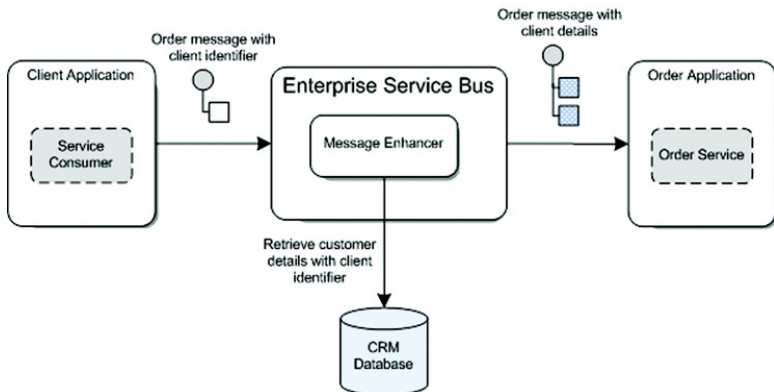


Fuente: [Rademakers & Dirksen]

# Funcionalidades de un ESB (V)

## Enriquecimiento de mensajes

Incorporación de información extra a los mensajes.

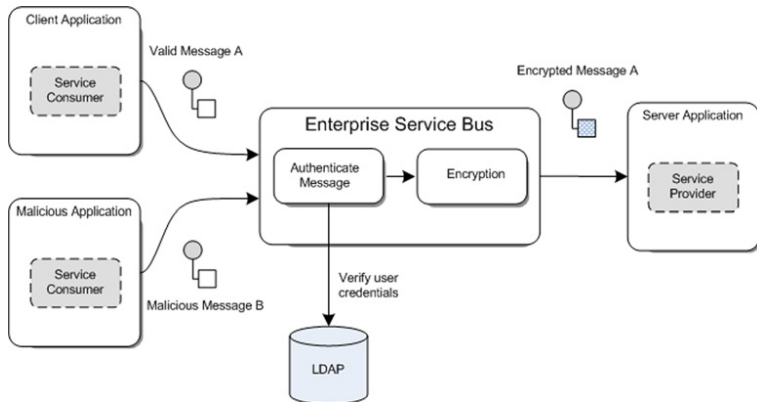


Fuente: [Rademakers & Dirksen]

# Funcionalidades de un ESB (VI)

## Seguridad

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

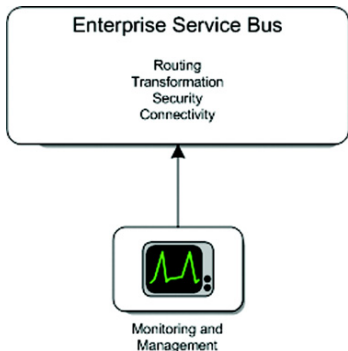


Fuente: [Rademakers & Dirksen]

# Funcionalidades de un ESB (VII)

## Administración y monitorización

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



Fuente: [Rademakers & Dirksen]

# Índice

## 1 Introducción

## 2 Funcionalidades de un ESB

## 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/>

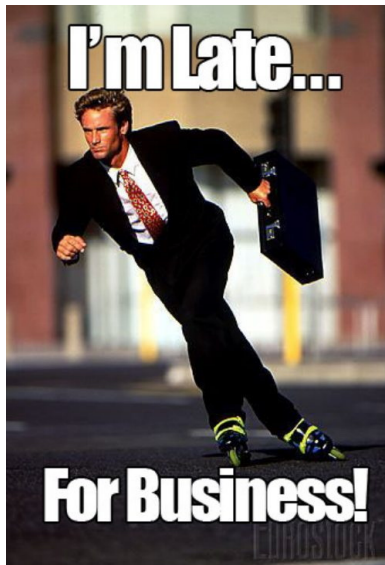


## Algunos ESB de código abierto (II)

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](http://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](http://www.ibm.com/developerworks/webservices/library/ws-soa-eda-esb/index.html)

