Procesamiento de Eventos Complejos en Arquitecturas Orientadas a Servicios 2.0

Itinerario Formativo de Doctorado 7009

Juan Boubeta Puig

Grupo UCASE de Ingeniería del Software
Departamento de Ingeniería Informática

14 de mayo de 2013

Índice

Índice

Índice

Arquitecturas orientadas a servicios (I)

Servicio

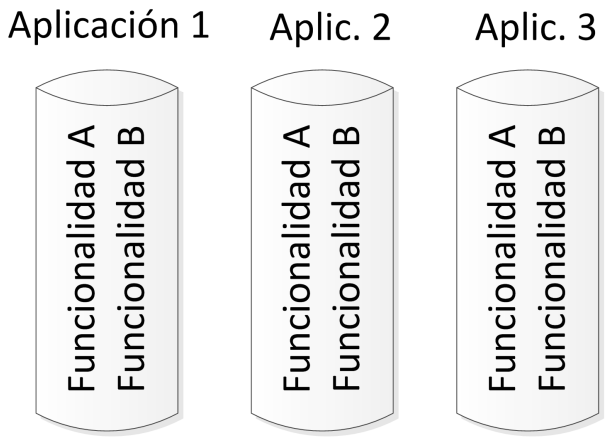
- Un “contrato” (prestaciones al usuario).
- “Funcionalidad concreta” que puede descubrirse y que describe qué puede hacer y cómo interactuar con él.

Arquitectura Orientada a Servicios o *Service-Oriented Architecture* (SOA)

- Arquitectura software que define un modelo desacoplado de los servicios para soportar los requisitos de los procesos de negocio.
- Proporcionan “funciones” que pueden ser reutilizables por distintos clientes (sólo necesitan conocer la descripción del servicio).

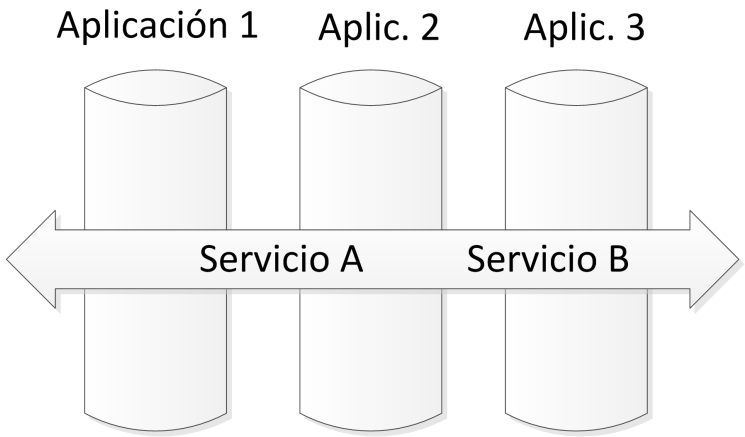
Arquitecturas orientadas a servicios (II)

El enfoque silo



Funcionalidades comunes duplicadas

El enfoque SOA



Funcionalidades comunes compartidas a través de servicios

Índice

Servicios Web (I)

Definición (W3C, 2004)

*Un **sistema software** diseñado para ofrecer **interacción máquina-a-máquina** sobre una red. Tiene una **interfaz** descrita en un formato procesable por la máquina: **WSDL**. Otros sistemas interactúan con el servicio Web en una forma prescrita por su descripción utilizando **mensajes SOAP**, normalmente transmitidos usando HTTP con una serialización XML y con otros estándares Web relacionados.*

- **Publicar los WS**: estándares WSDL (*Web Services Description Language*), XML (*eXtensible Markup Language*) y SOAP (*Simple Object Access Protocol*).
- **Componer los WS**:
  - Orquestación: estándar WS-BPEL (*Business Process Execution Language*).
  - Coreografía: WS-CDL (*Web Services Choreography Description Language*).

Servicios Web (II)

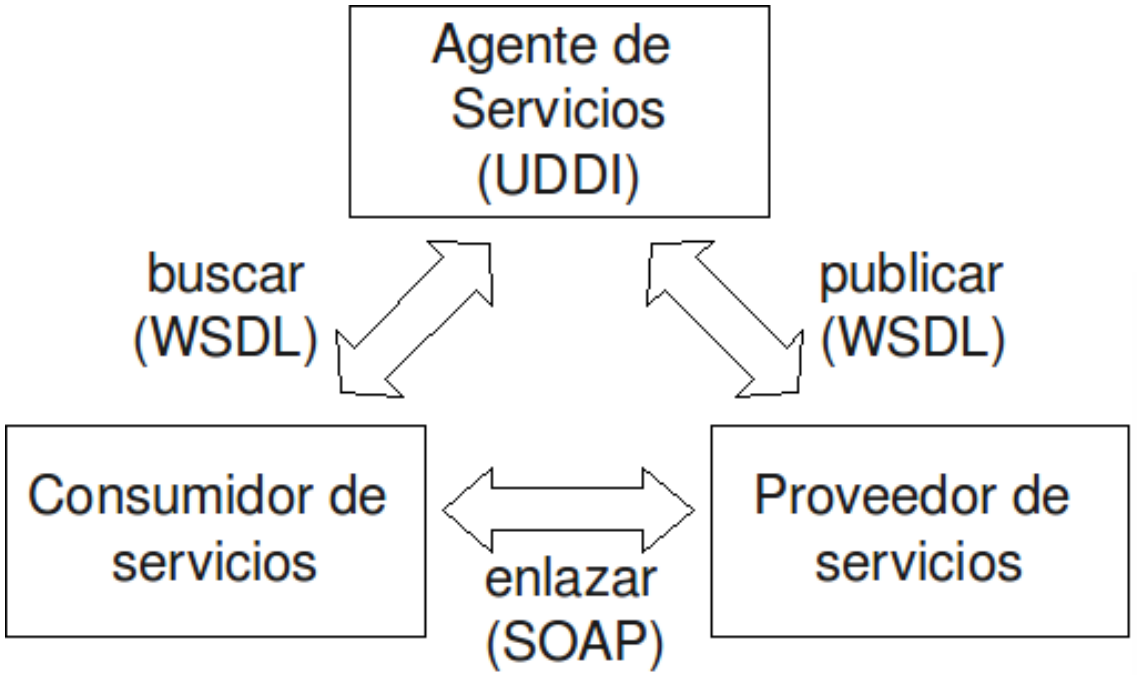
WSDL

- Lenguaje basado en XML utilizado para describir la interfaz de WS.
- Describe qué pueden hacer, dónde se encuentran, qué tipo de datos esperan y en qué formato.
- Sólo se describe el WS, no su implementación $\Longrightarrow$ reduce los problemas de compatibilidad entre WS.

SOAP

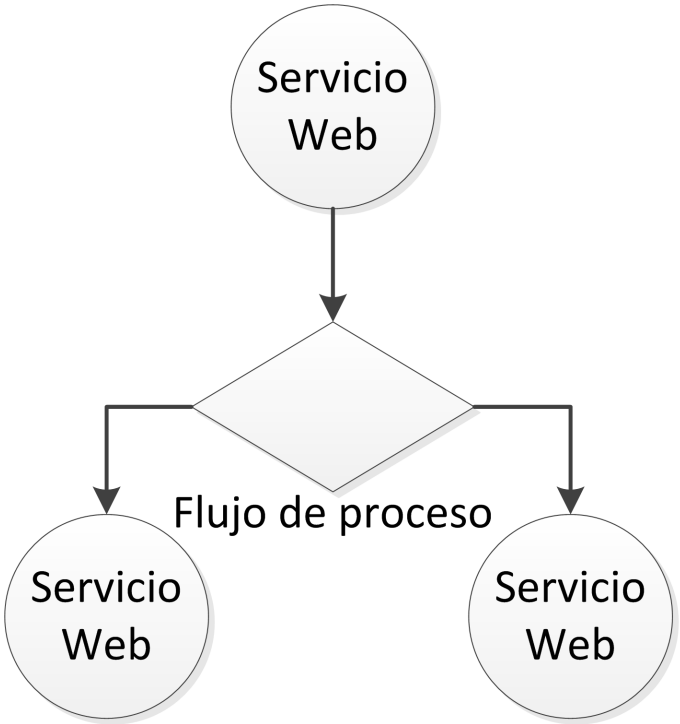
- Protocolo simple y extensible basado en XML, estandarizado por el W3C.
- Utilizado para el intercambio de mensajes en un entorno distribuido.

Servicios Web (III)

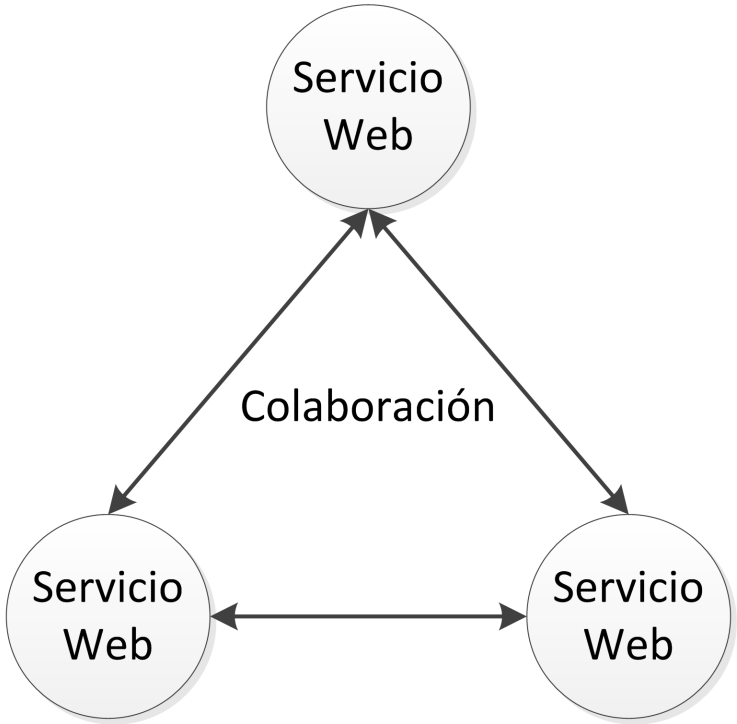


Servicios Web (IV)

Orquestación vs Coreografía



Orquestación



Coreografía

Índice

Eventos

Evento

1. Un **cambio** en el **estado** de algo.
2. **Algo** que **ocurre** (o no ocurre), o se espera que ocurra.
3. Una **condición detectable** que puede lanzar una notificación.

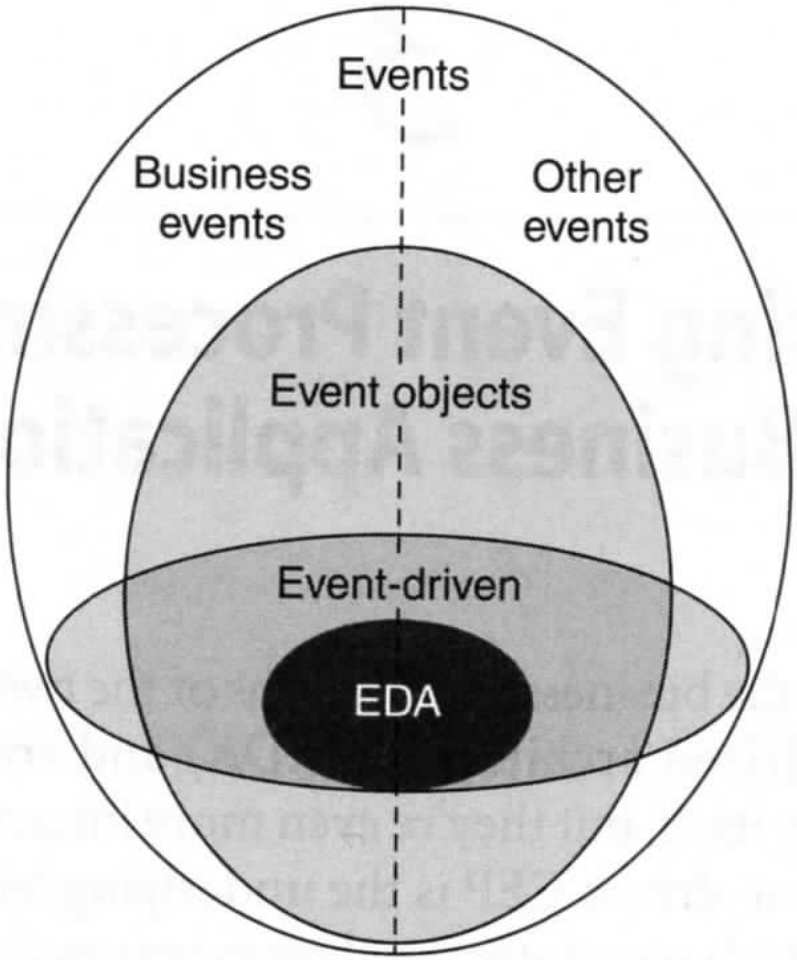
Evento de negocio

Un evento con significado relevante para actividades comerciales, industriales y gubernamentales.

Características de eventos

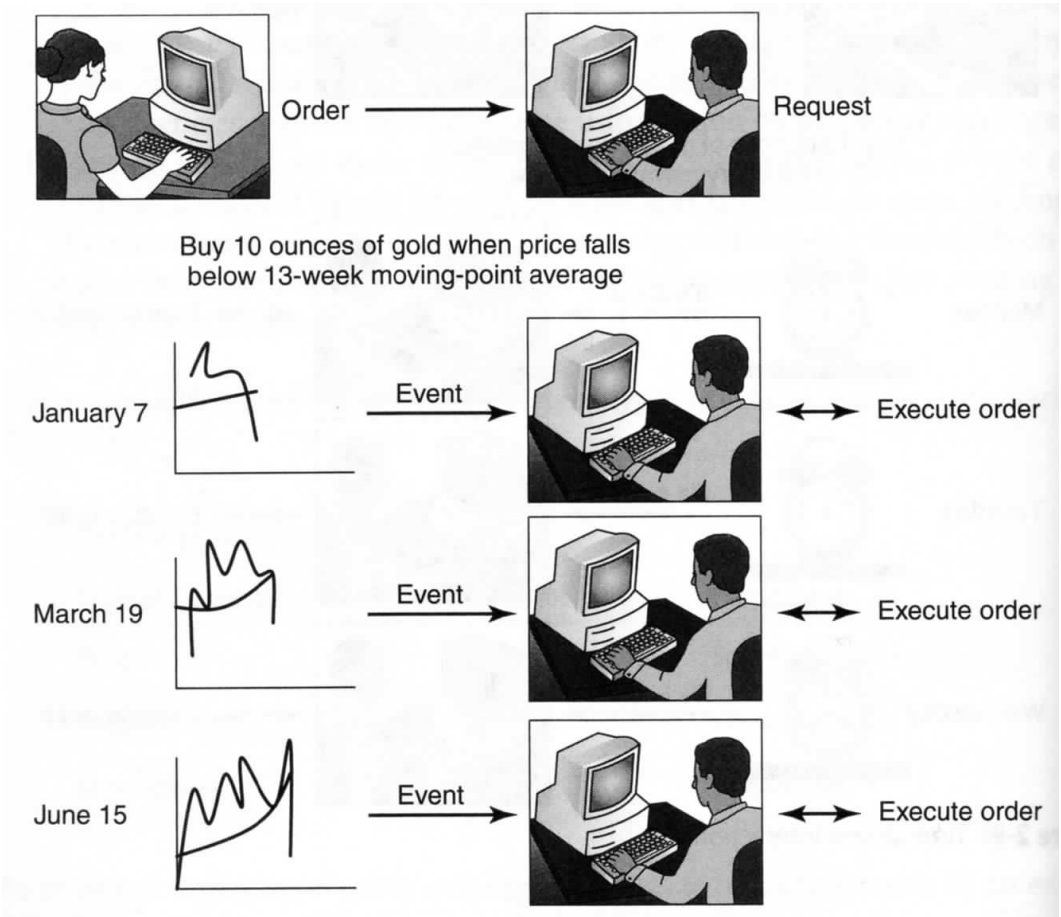
- No pueden ser previstos con exactitud.
- Actúan como “estímulos”.
- Permiten detectar situaciones críticas y oportunidades de negocio.
- Importantes en la sociedad: el mundo real es complejo y dinámico.

Procesamiento de eventos



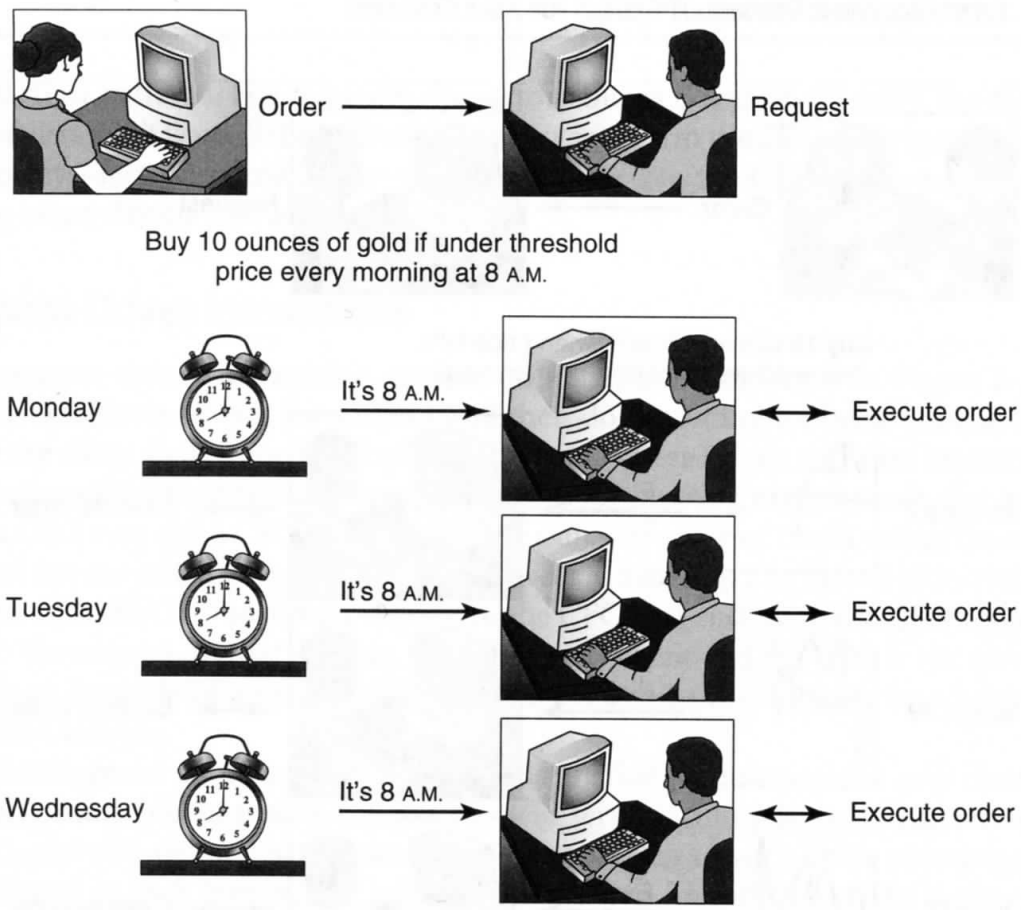
Fuente: [Chandy y Schulte]

Interacciones “dirigidas por eventos”



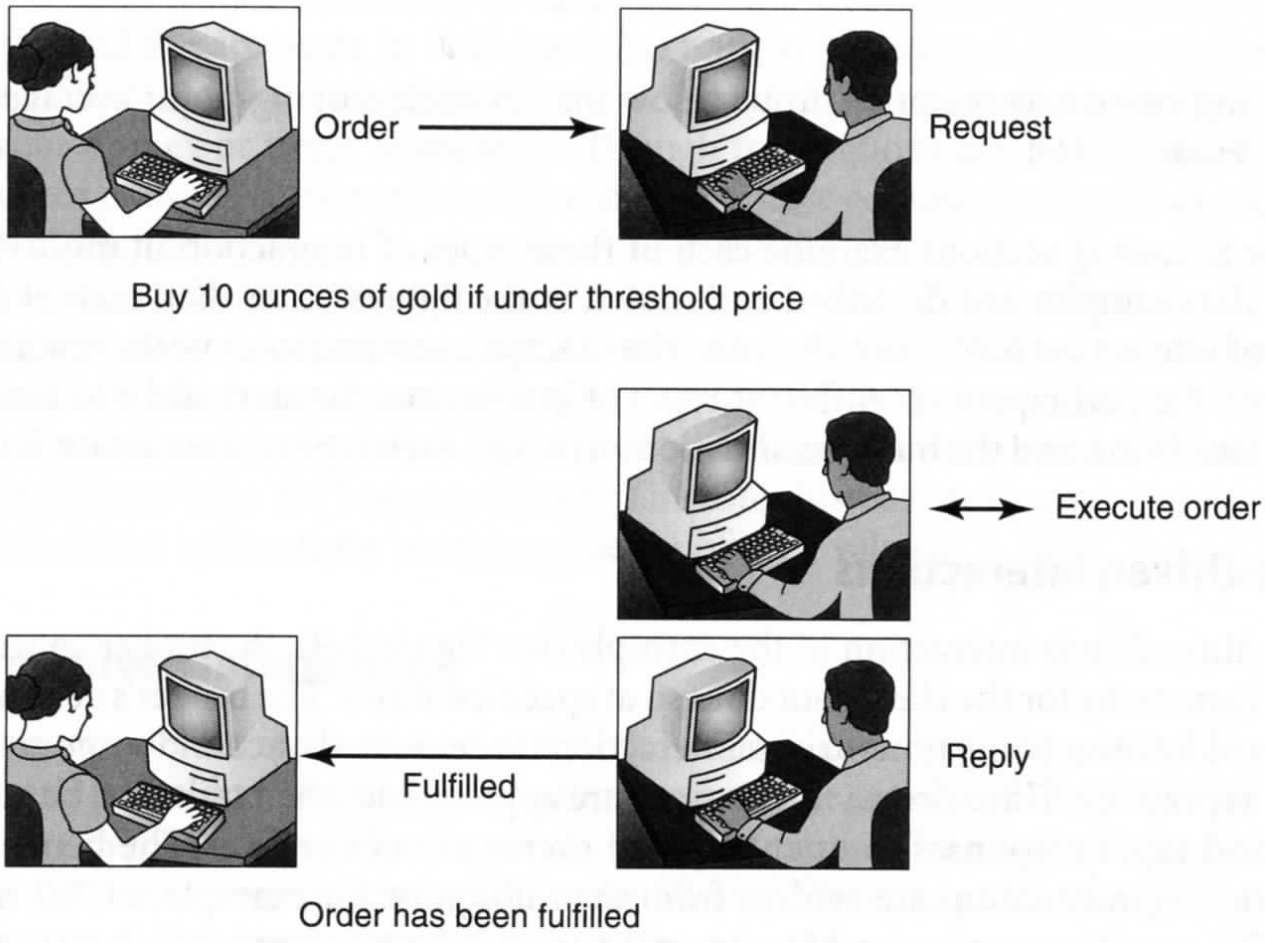
Fuente: [Chandy y Schulte]

Interacciones “dirigidas por tiempo”



Fuente: [Chandy y Schulte]

Interacciones “dirigidas por peticiones”



Fuente: [Chandy y Schulte]

Índice

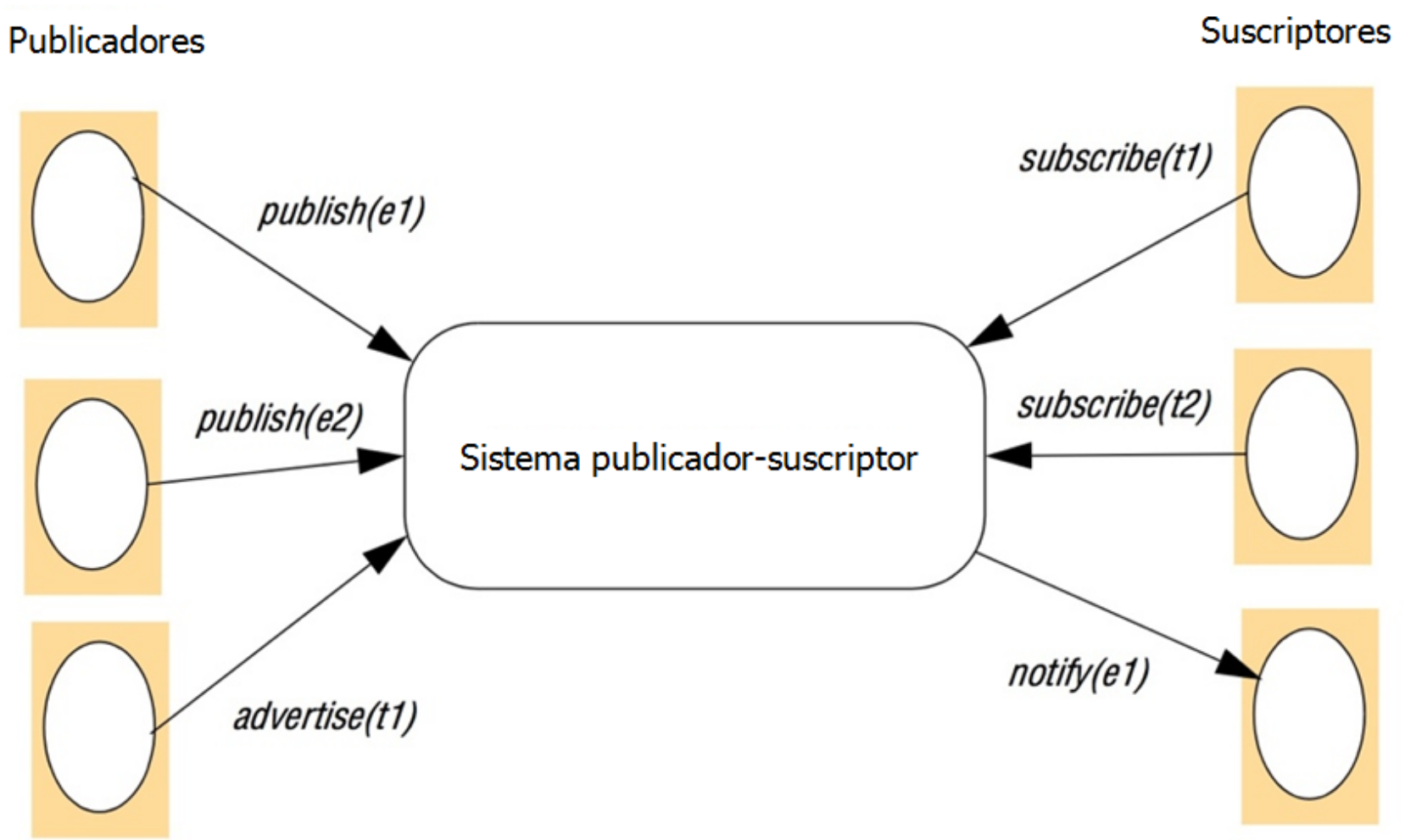
Arquitecturas dirigidas por eventos (I)

Arquitecturas dirigidas por eventos o *Event-Driven Architecture* (EDA)

- No es un término que acapare cada una de las actividades relativas a eventos: estilo particular de procesamiento de eventos.
- Estilo arquitectónico en el que uno o más componentes de un sistema software son dirigidos por eventos y están desacoplados.
- Objetivo: detectar y responder a eventos **simples** tan pronto como sea posible.
- “Complementa” a SOA.

Arquitecturas dirigidas por eventos (II)

Sistema publicador-suscriptor



Fuente: [Coulouris et al.]

Arquitecturas dirigidas por eventos (III)

Sistema publicador-suscriptor

Características de los sistemas publicador-suscriptor

- **Heterogeneidad**:
  - Cuando las notificaciones de eventos son utilizadas como medio de comunicación, los componentes en un sistema distribuido que no fueron diseñados para interoperar pueden trabajar juntos.
  - Se requiere que los objetos generadores de eventos publiquen los tipos de eventos que ofrecen, y que los otros objetos se suscriban a patrones de eventos y ofrezcan una interfaz para recibir y tratar las notificaciones resultantes.
- **Asincronía**:
  - Las notificaciones se envían asíncronamente por publicadores generadores de eventos a todos los suscriptores que están interesados en éstas.
  - Se evita que los publicadores necesiten sincronizarse con los suscriptores (publicadores y suscriptores desacoplados).

Arquitecturas dirigidas por eventos (IV)

Sistema publicador-suscriptor

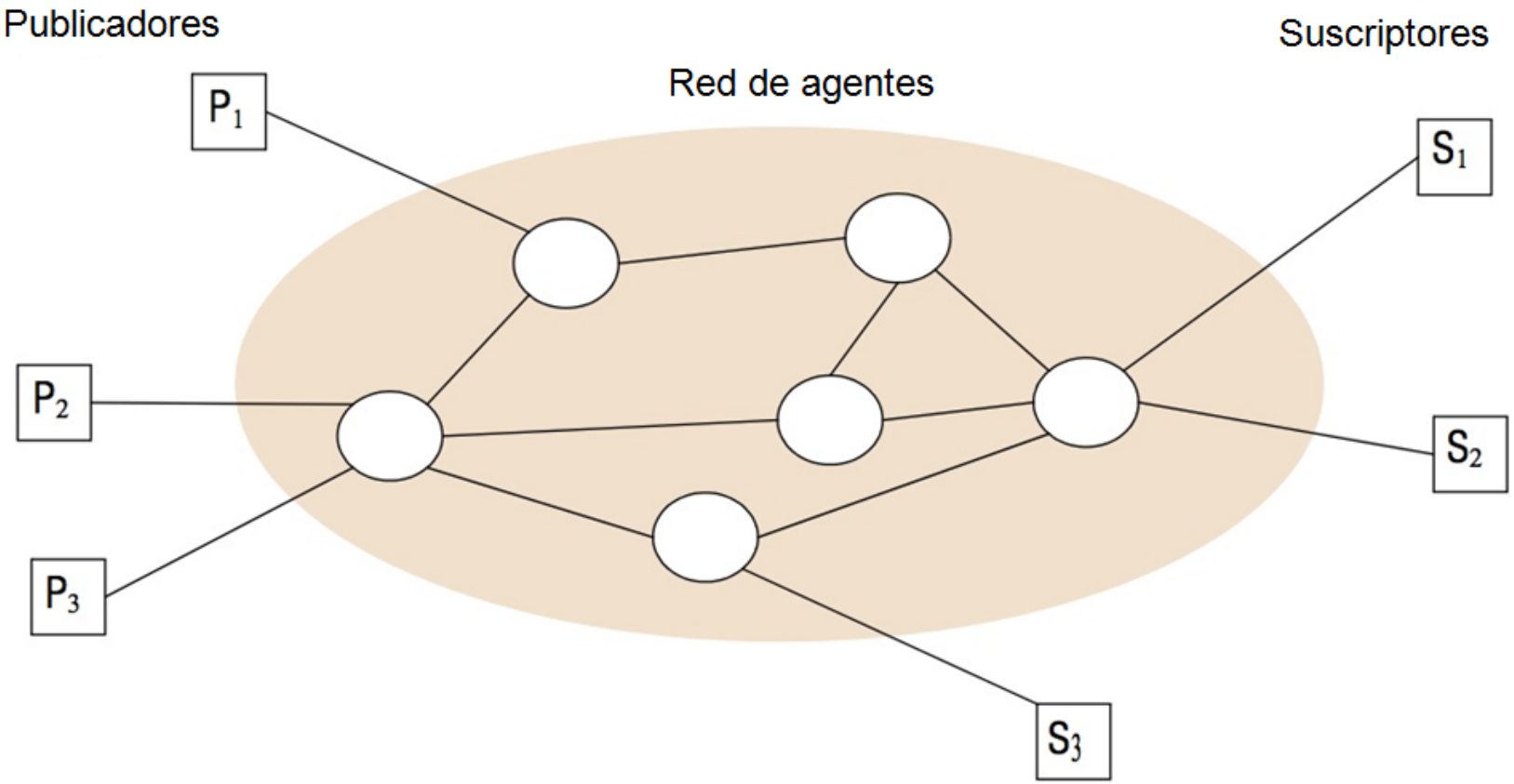
Aplicaciones de sistemas publicador-suscriptor

Gran variedad de dominios de aplicación, especialmente relacionados con diseminación de eventos a gran escala:

- Sistemas de información financieros.
- Flujos de datos en tiempo real: RSS *feeds*, plataformas IoT (Xively)...
- Apoyo al trabajo colaborativo: los participantes necesitan estar informados de eventos de interés compartido.
- Apoyo a la computación ubicua, incluyendo la gestión de eventos que provienen de infraestructuras ubicuas (eventos de localización...).
- Aplicaciones de monitorización, incluyendo monitorización de redes en Internet.
- Componente clave de la infraestructura de Google (diseminación de eventos sobre anuncios, *ad clicks*, a las partes interesadas).

Arquitecturas dirigidas por eventos (V)

Sistema publicador-suscriptor: implementación distribuida



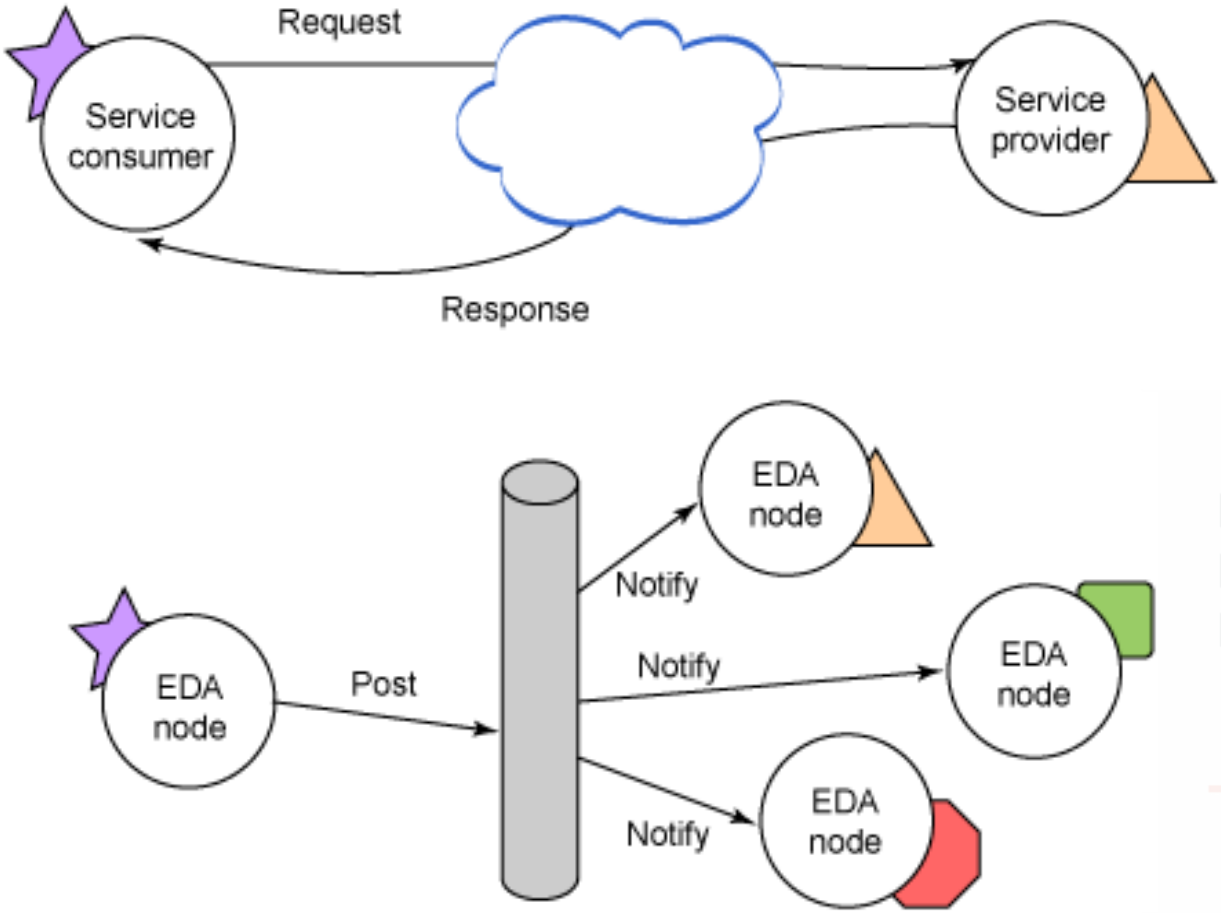
Fuente: [Coulouris et al.]

Índice

SOA vs EDA (I)

	SOA	**EDA**
Interacciones	Débilmente acopladas	Desacopladas
Comunicaciones	Uno a uno	Muchos a muchos (publicación-suscripción)
Reacción frente a	Clientes	Eventos
Tipo de comunicación	Síncrono	Asíncrono

SOA vs EDA (II)



Fuente: [Maréchaux]

Índice

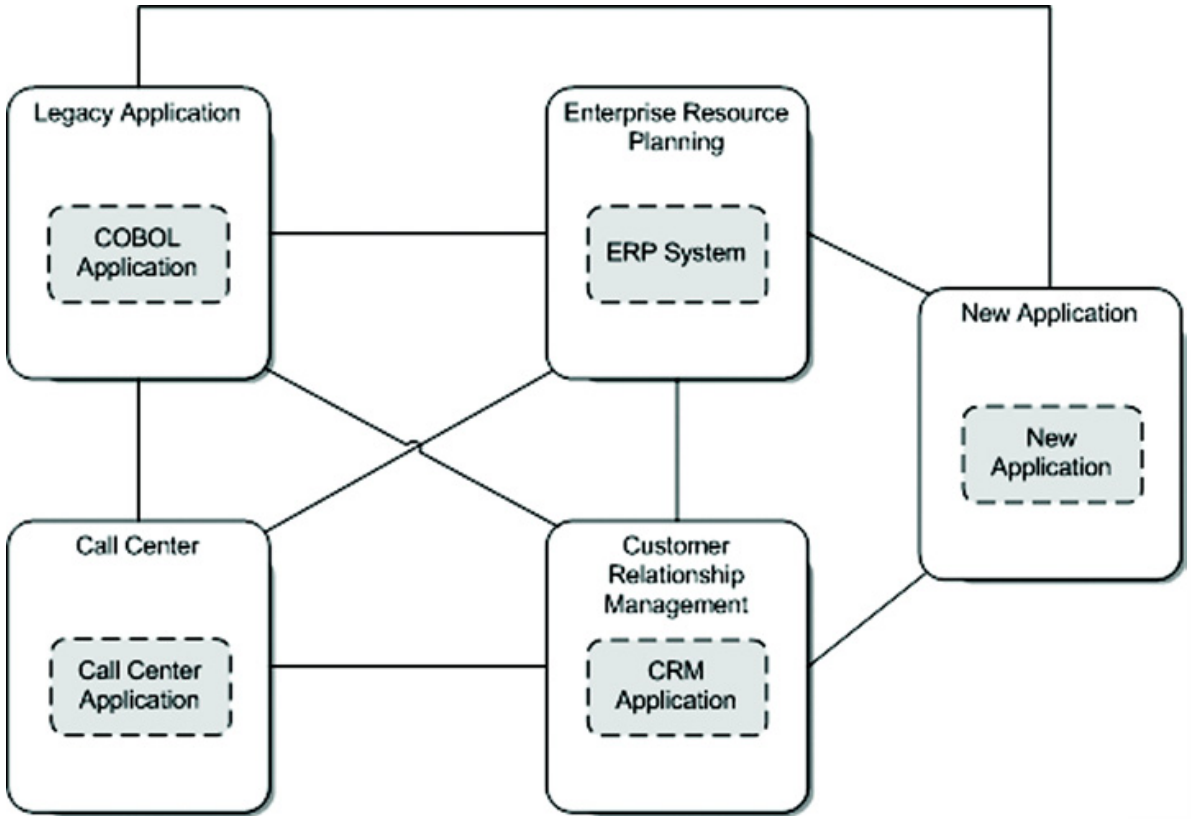
Índice

Definición de *Enterprise Service Bus* (ESB)

- **Elemento de integración** (multiprotocolo y multipropósito) en 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)

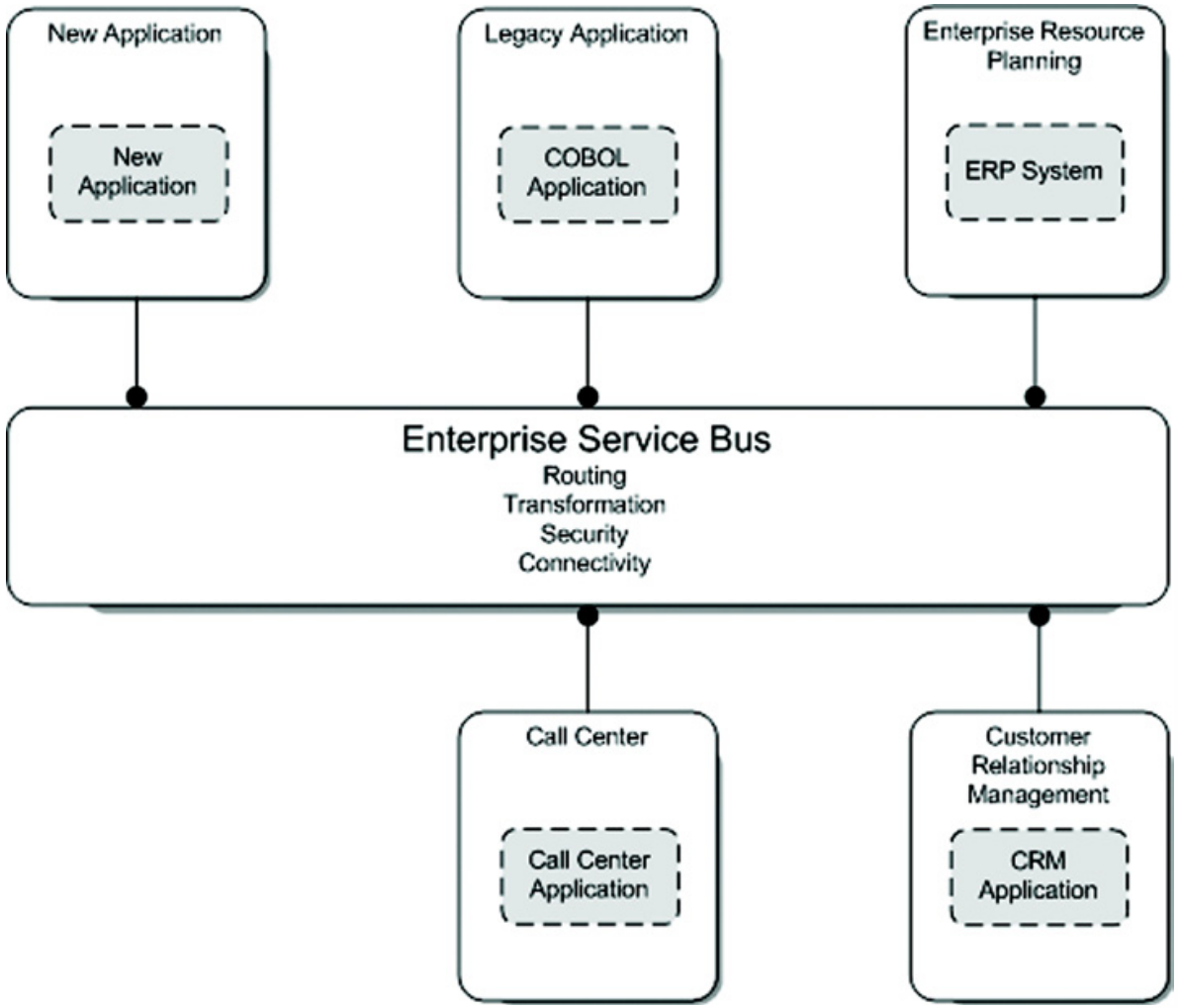
Arquitectura punto-a-punto: ¿qué ocurre al incorporar una nueva aplicación?



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

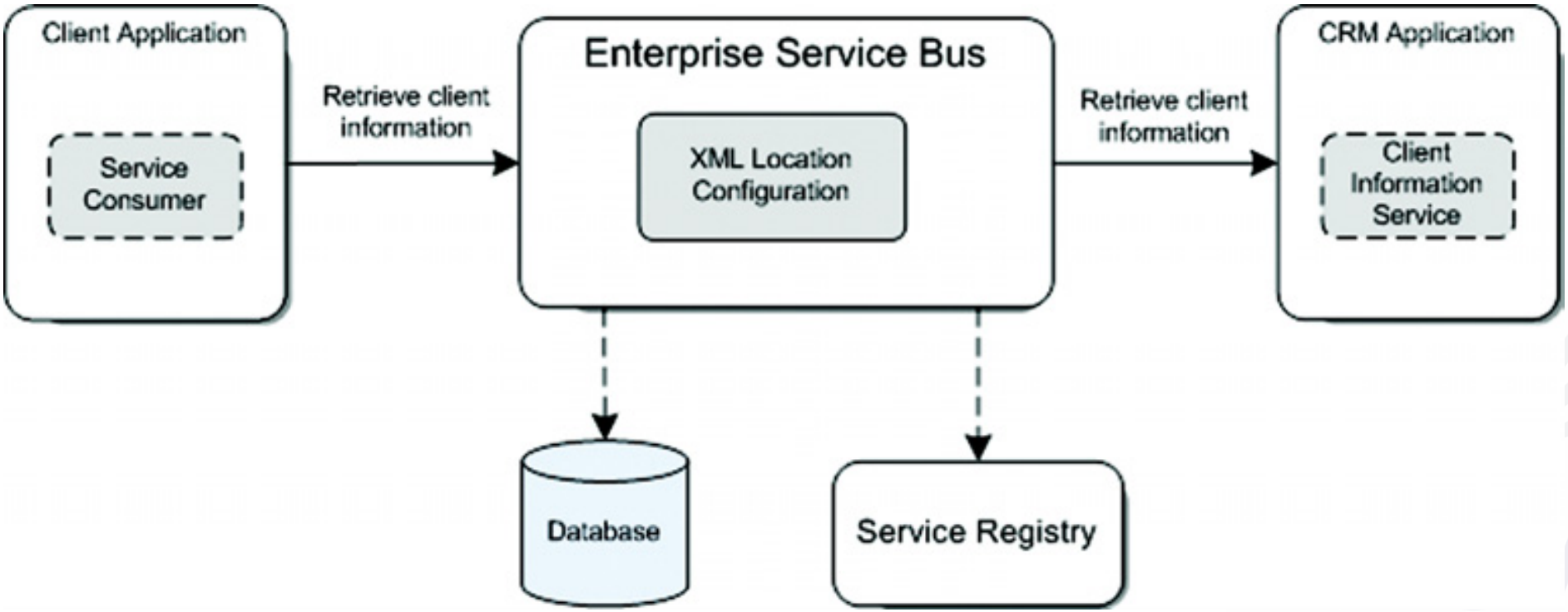
- Facilita la integración de aplicaciones.
- Ideal para trabajar en entornos heterogéneos: diferentes tecnologías y protocolos.
- Reduce el coste total de la gestión y el mantenimiento.

Índice

Funcionalidades de un ESB (I)

Transparencia de localizacion

Desacoplamiento entre el consumidor y el proveedor de servicio.

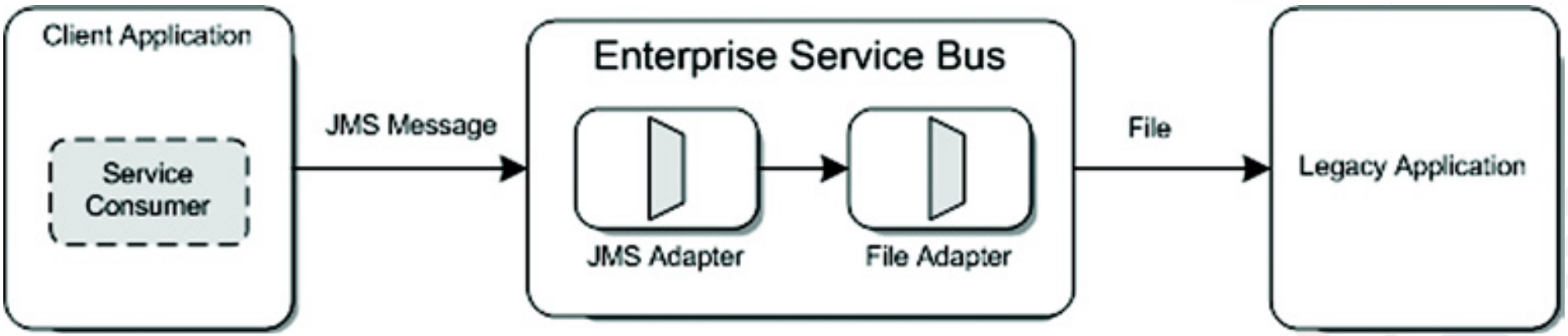


Fuente: [Rademakers & Dirksen]

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...

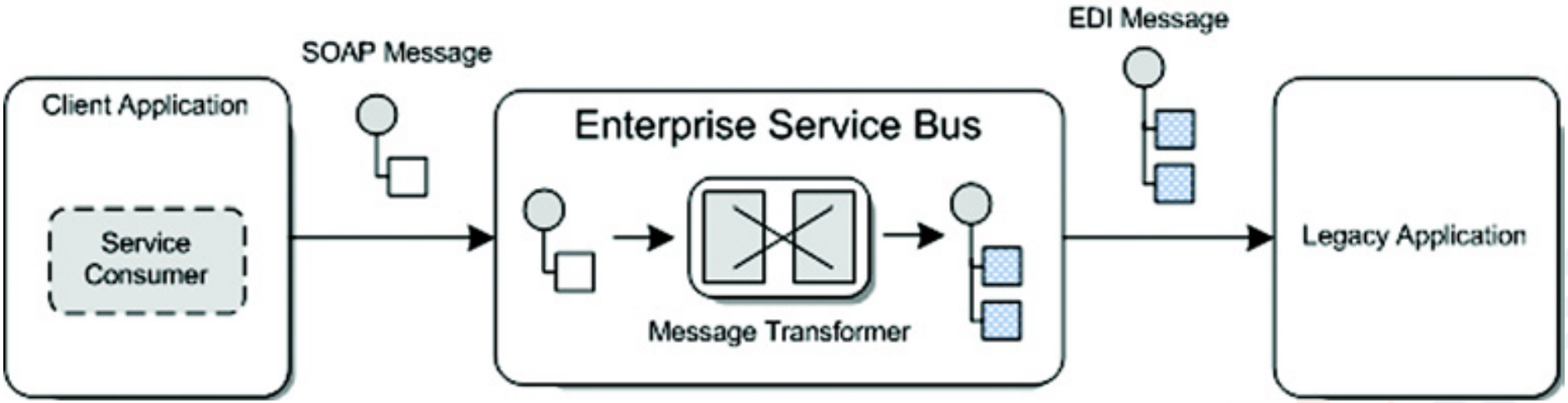


Fuente: [Rademakers & Dirksen]

Funcionalidades de un ESB (III)

Transformación de mensajes

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

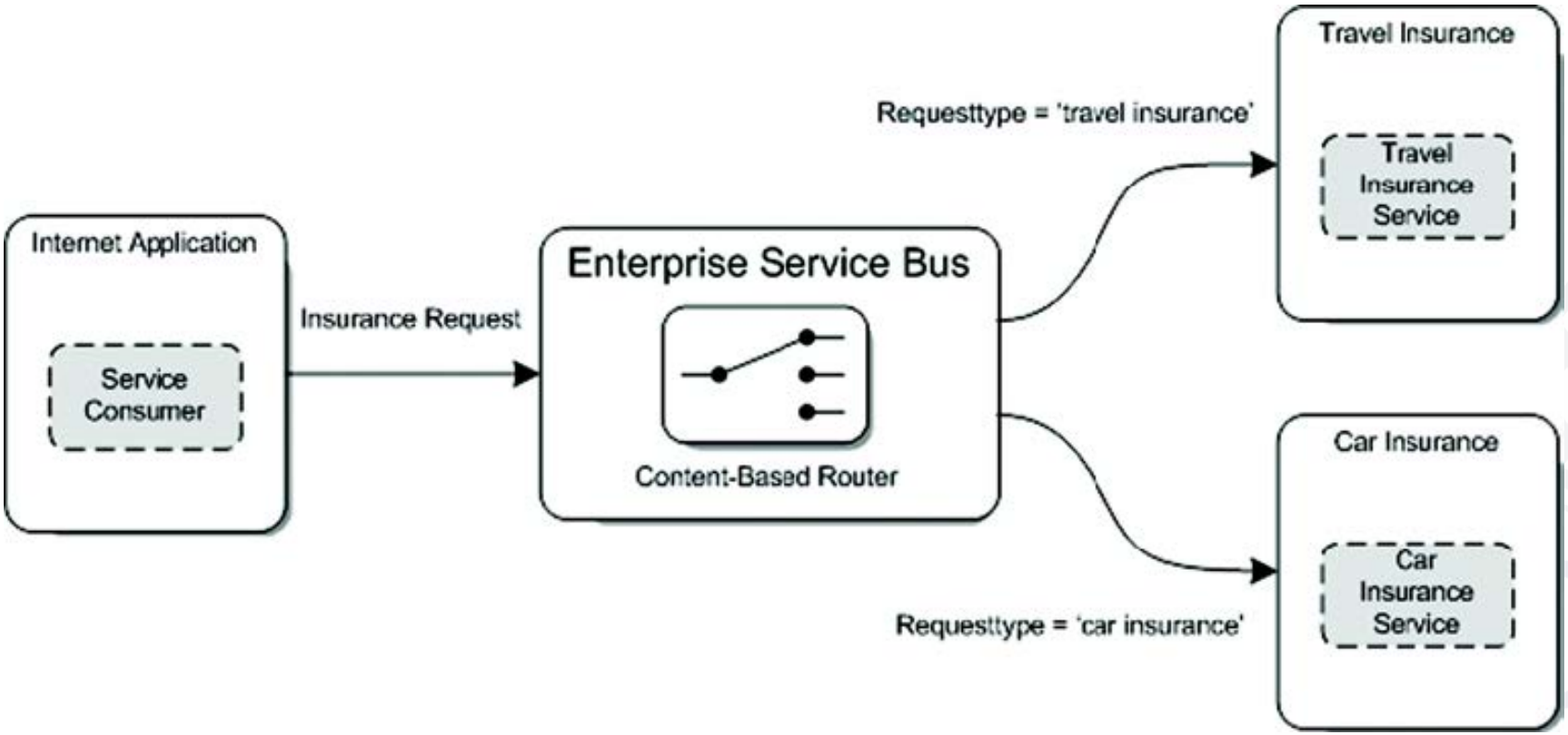


Fuente: [Rademakers & Dirksen]

Funcionalidades de un ESB (IV)

Encaminamiento de mensajes

Elección del destino de cada mensaje.

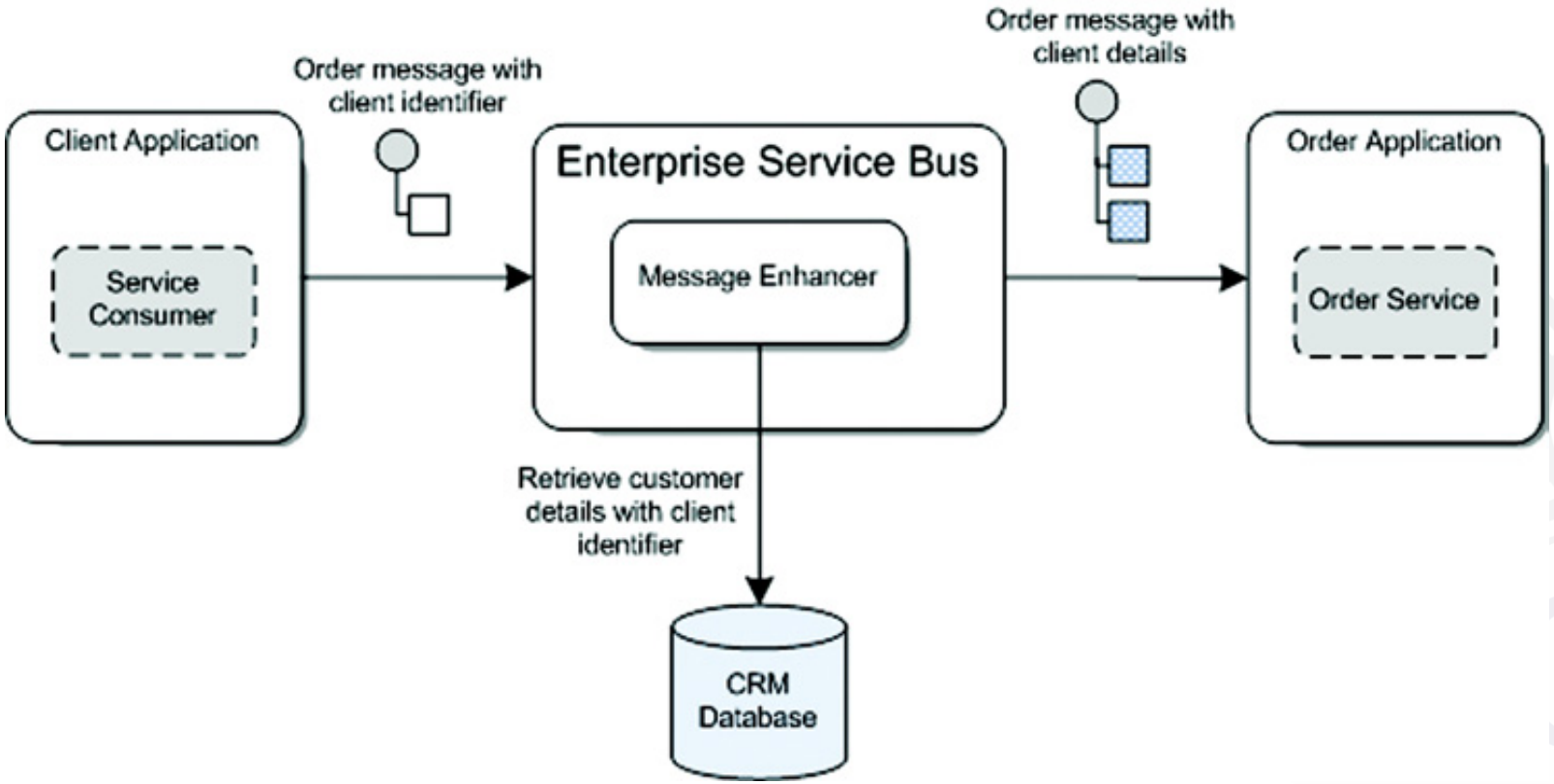


Fuente: [Rademakers & Dirksen]

Funcionalidades de un ESB (V)

Enriquecimiento de mensajes

Incorporación de información extra a los mensajes.

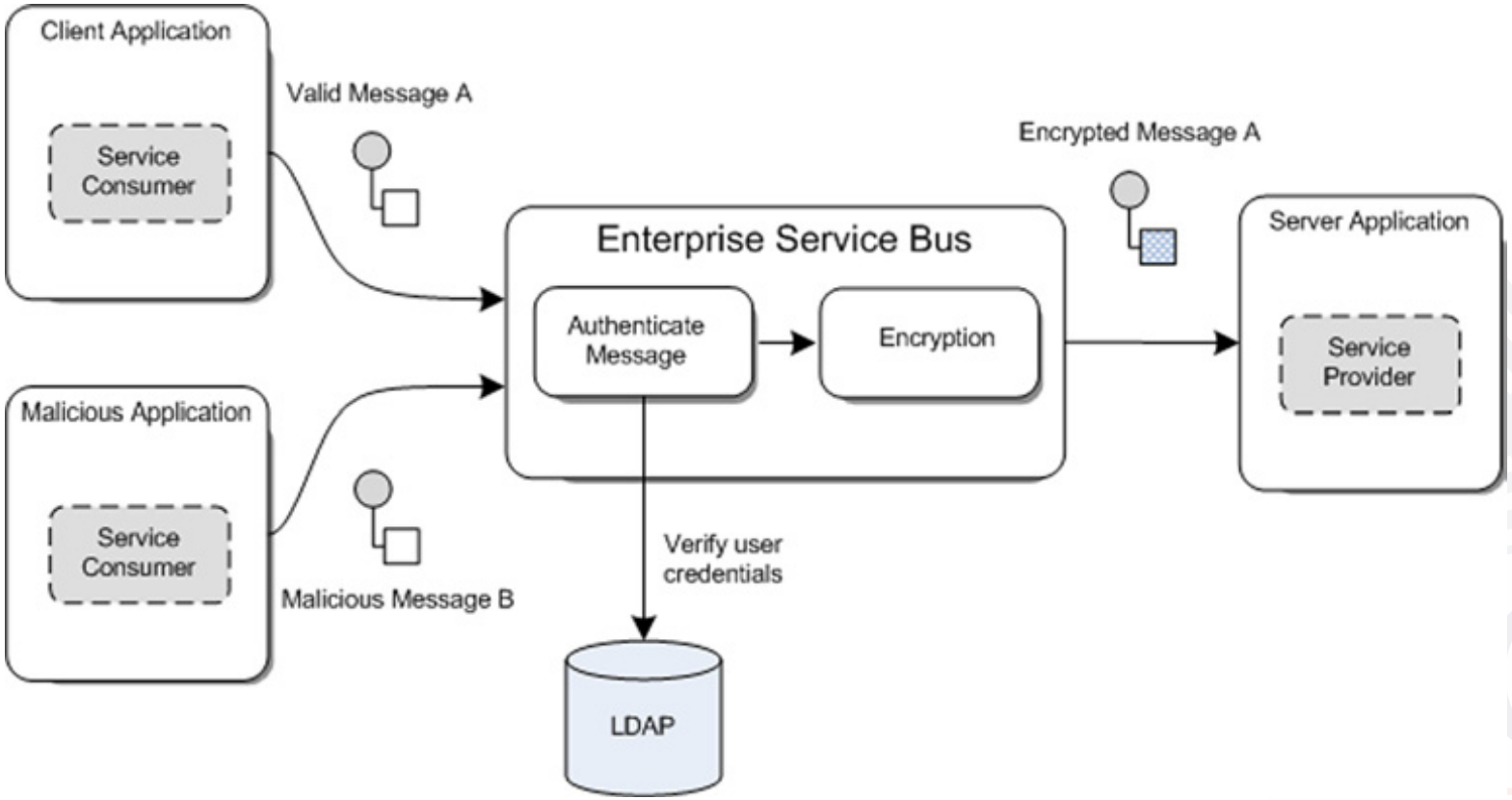


Fuente: [Rademakers & Dirksen]

Funcionalidades de un ESB (VI)

Seguridad

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

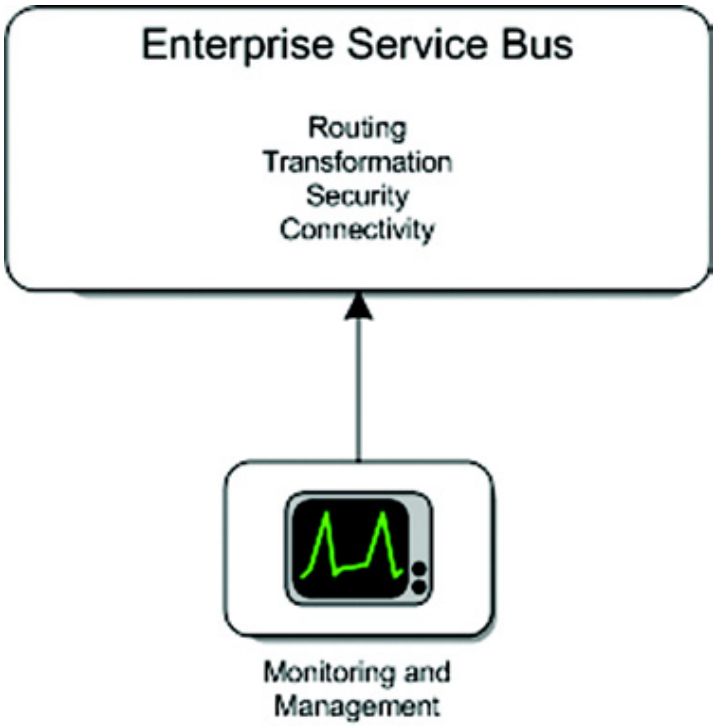


Fuente: [Rademakers & Dirksen]

Funcionalidades de un ESB (VII)

Administración y monitorización

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



Fuente: [Rademakers & Dirksen]

Índice

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.

Índice

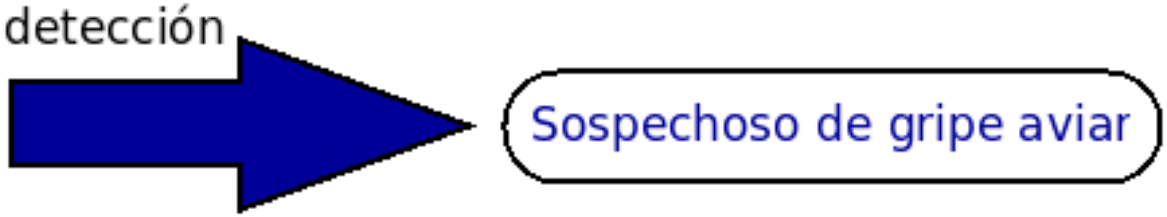
Procesamiento de eventos complejos (I)

- Procesamiento de eventos complejos o *Complex Event Processing* (CEP).
- Tecnología emergente que proporciona un conjunto de técnicas que ayudan a hacer un uso eficiente de EDA.
- Permite procesar, analizar y correlacionar grandes cantidades de eventos.
- Para detectar y responder en tiempo real a situaciones críticas del negocio.
- Se utilizan unos patrones de eventos que inferirán nuevos eventos más complejos y con un mayor significado semántico.
- Ayuda a las empresas a comprender lo que está ocurriendo en cada momento, combinando múltiples eventos simples en complejos.
- Requisitos: motor CEP (p.e. Esper) y lenguaje específico (p.e. EPL).

Procesamiento de eventos complejos (II)

Escenario de detección de un caso sospechoso de gripe aviar

Patrón de evento complejo



Evento complejo

Procesamiento de eventos complejos (III)

Tipos de aplicaciones

- Responden a situaciones que cambian rápidamente y de forma asíncrona, y donde las interacciones no tienen que ser transacciones.
- Gestionan excepciones.
- Reaccionan rápidamente a situaciones inusuales.
- Requieren bajo acoplamiento y adaptabilidad.

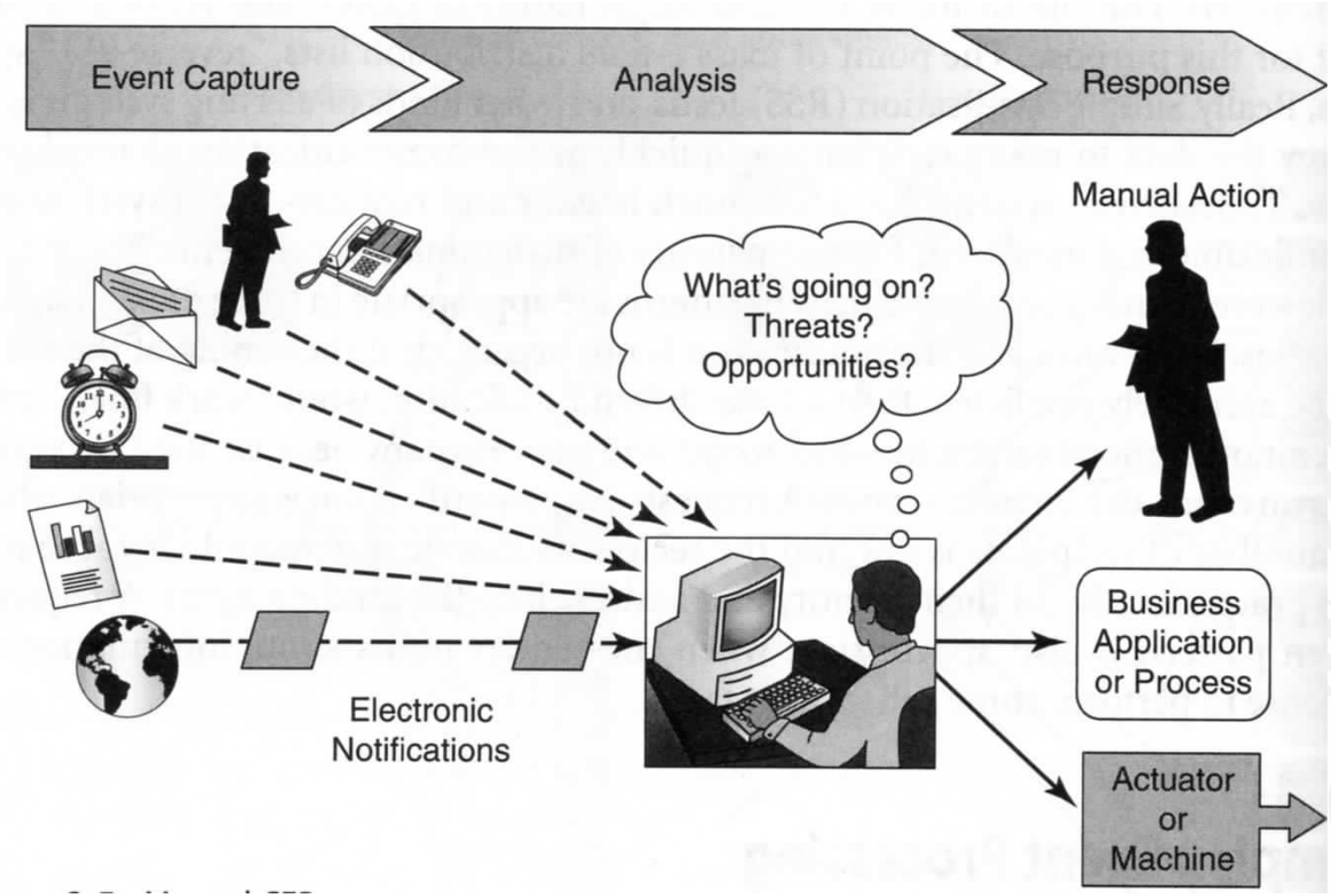
Procesamiento de eventos complejos (IV)

Implementación

- Enfoques:
  - Manual.
  - Parcialmente automatizado.
  - Automatizado.
- Estos enfoques presentan 3 fases:
  1. Captura de eventos.
  2. Análisis.
  3. Respuesta.

Procesamiento de eventos complejos (V)

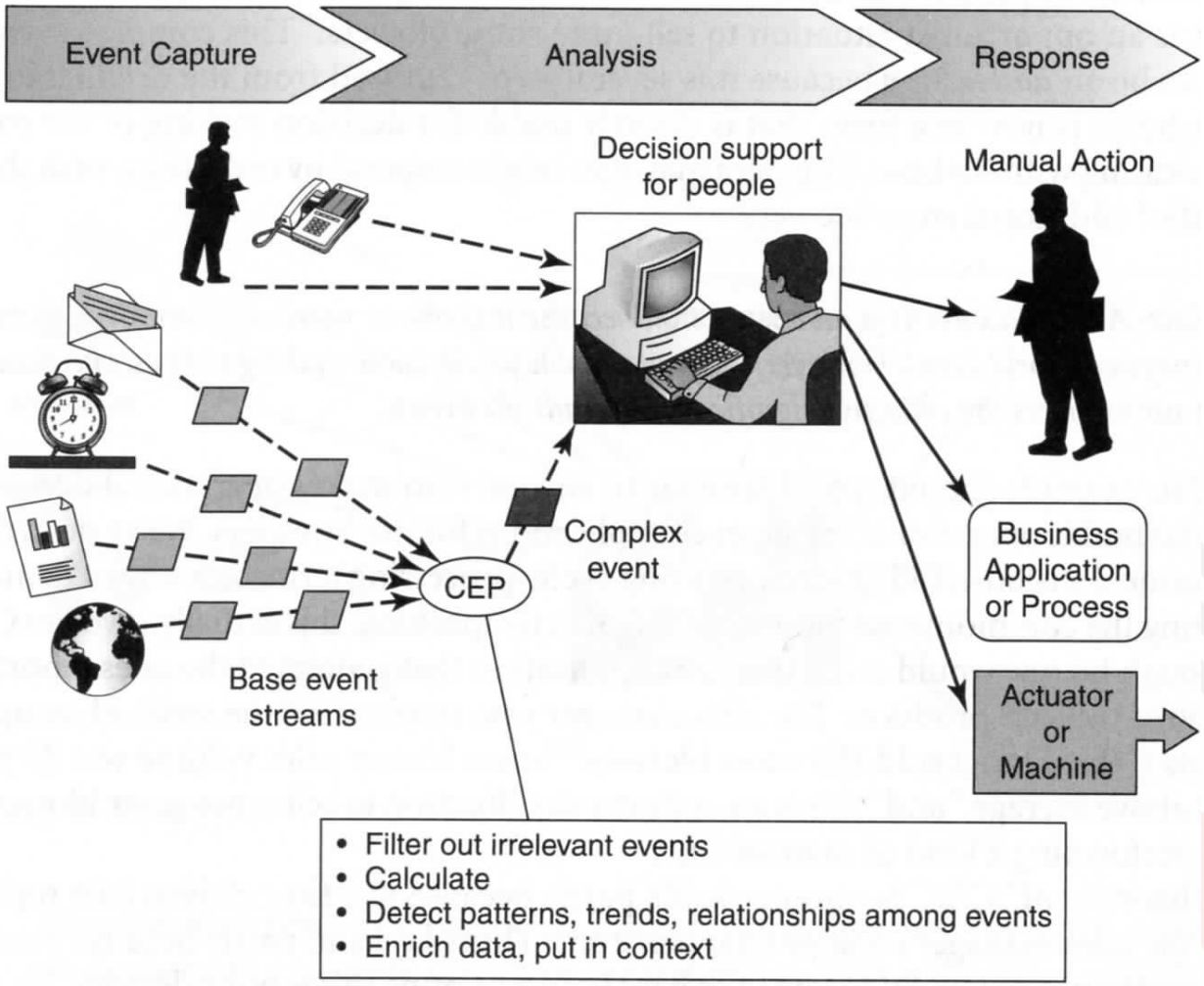
Manual



Fuente: [Chandy y Schulte]

Procesamiento de eventos complejos (VI)

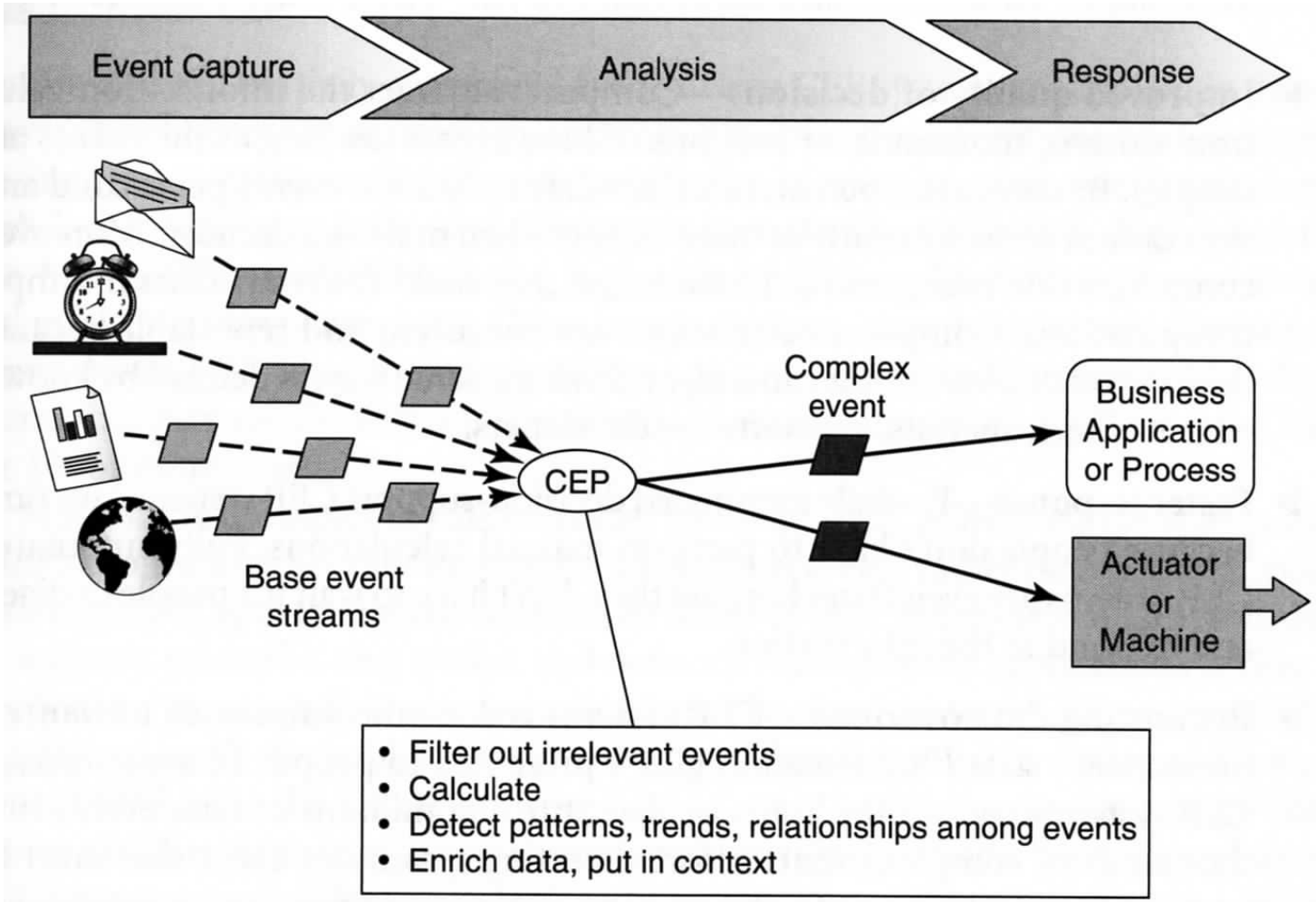
Parcialmente automatizado



Fuente: [Chandy y Schulte]

Procesamiento de eventos complejos (VII)

Automatizado (I)



Fuente: [Chandy y Schulte]

Procesamiento de eventos complejos (VIII)

Automatizado (II)

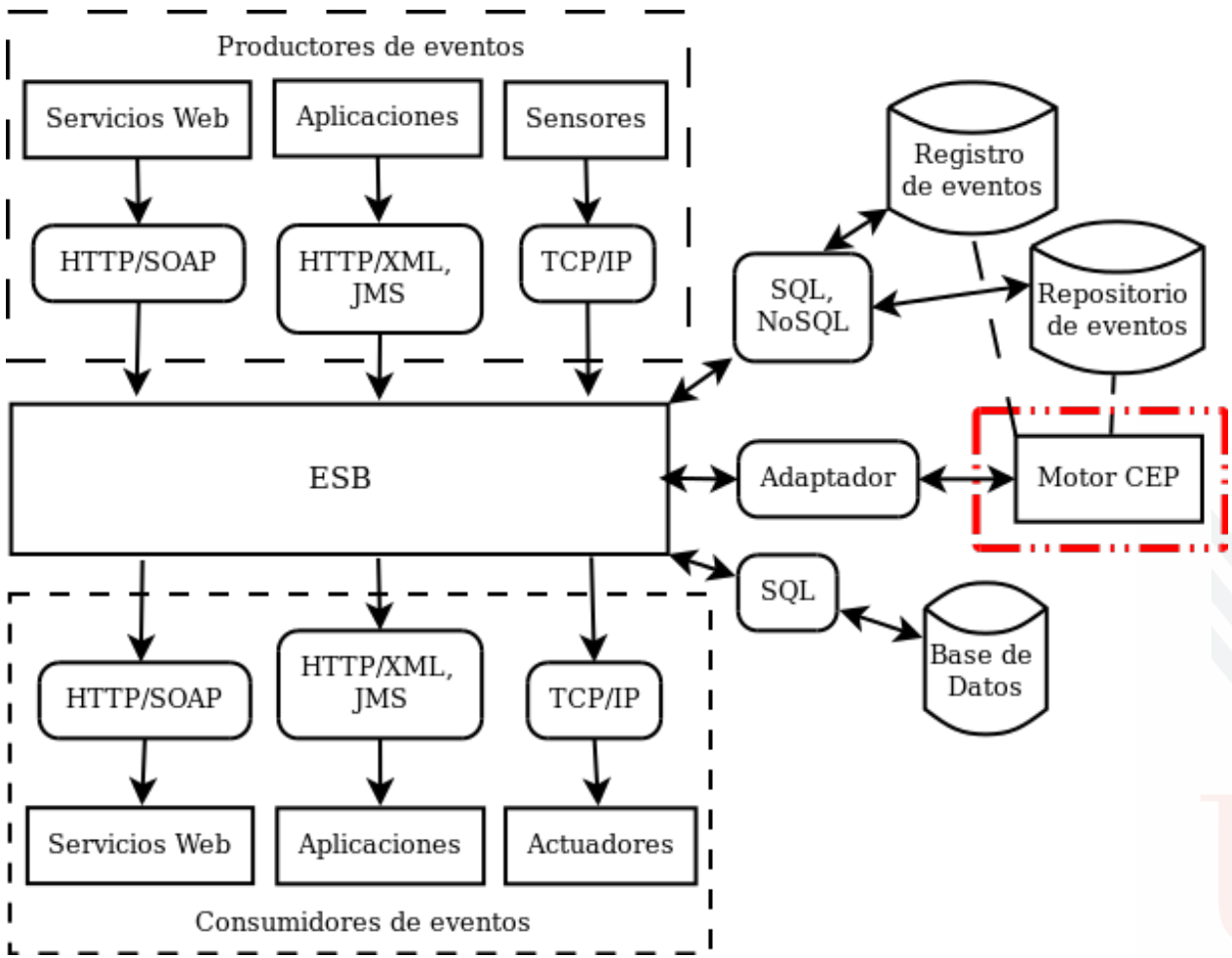
Ventajas del CEP automatizado

- Mejora la calidad de las decisiones.
- Respuesta más rápida.
- Previene la sobrecarga de datos.
- Reduce el coste.

Índice

Arquitectura para la integración de CEP en SOA 2.0

Permite detectar eventos relevantes en sistemas complejos y heterogéneos



Fuente: [Boubeta-Puig et al.]

Referencias bibliográficas I

K. M. Chandy y W. R. Schulte
Event Processing: Designing IT Systems for Agile Companies
McGraw-Hill, 2010

G. Coulouris; J. Dollimore & T. Kindberg
Distributed Systems: Concepts and Design (5ª ed.)
Addison-Wesley, 2012.

O. Etzion and P. Niblett
Event Processing in Action
Manning, 2010

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

Referencias bibliográficas II


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.


M. Edwards et al.
Un modelo conceptual para los sistemas de procesamiento de eventos (2010)
www.ibm.com/developerworks/ssa/webservices/library/ws-eventprocessing

Referencias bibliográficas III

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

OASIS.
Web Services Business Process Execution Language 2.0
http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html, abril 2007.

W3C.
Extensible Markup Language (XML)
http://www.w3.org/XML/, enero 2012.

Referencias bibliográficas IV

W3C.
SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)
http://www.w3.org/TR/soap12-part1/, abril 2007.

W3C.
Web Services Choreography Description Language Version 1.0
http://www.w3.org/TR/ws-cdl-10/, noviembre 2005.

W3C.
Web Services Description Language (WSDL) 1.1
http://www.w3.org/TR/wsdl, marzo 2001.