



Jornadas de Informática

## **Integración de Aplicaciones**

# Quiénes somos: un gran proyecto en crecimiento



1.275,9 M€ de facturación en 2012

Bienes de consumo, manufacturas y servicios



40 M€  
connectis  
1.000  
Profesionales

90 M€  
connectis  
1.800  
Profesionales

140 M€  
connectis  
2.600  
Profesionales

+  
cGS

Abril 2012

Diciembre 2012

Septiembre 2013

Diciembre 2013

Aurelius adquiere  
Thales IS



Aurelius adquiere  
Steria ibérica



Aurelius adquiere  
NEC E.S (RU, Suiza,  
España y Portugal)



Aurelius adquiere  
TELVENT Global Services  
(España, Brasil y Chile)



## Integración



## ***Integración.***

Construir un todo.

Completar un todo con las partes que faltaban.

Hacer que alguien o algo pase a formar parte de un todo.

## *Integración sensorial*

La integración sensorial es la función del sistema nervioso que permite asimilar la información del entorno y obtener la percepción correcta de la situación que nos rodea.

Cuando falla una de las piezas que componen ese mecanismo la persona se enfrenta a la **Disfunción de Integración Sensorial (DIS) o Desorden en el Procesamiento Sensorial (DPS)**.

La integración de datos y aplicaciones en las compañías es inherente a la implementación de cualquier negocio. Cuando esta se implementa ad-hoc o en un contexto de crecimiento desordenado esta se convierte en un problema en si misma de difícil solución.

## *Integración empresarial de aplicaciones*

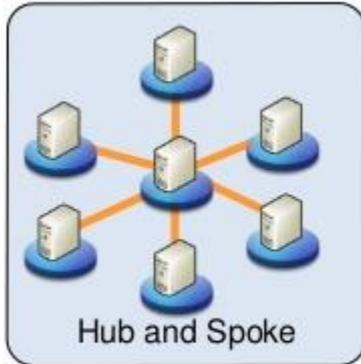
**EAI (*Enterprise Application Integration*) es una disciplina que facilita la interoperabilidad y la comunicación de información entre sistemas de información heterogéneos a través de la implementación de patrones de integración.**

Cuando dichos sistemas no pueden compartir su información, se crean **cuellos de botella que requieren intervención humana en la forma de toma de decisiones o en el ingreso mismo de la información:**

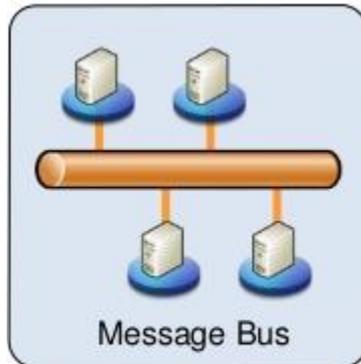
***Islas de Información***

- **Integración de información:** se favorece el flujo de información entre sistemas en escenarios donde múltiples sistemas heterogéneos comparten dicha información (parcial o totalmente).
- **Independencia de soluciones específicas:** al proponer mecanismos de intercambio de información basados en mensajería se favorece la independencia e interoperabilidad.
- **Proporciona una fachada común** ante la necesidad de facilitar el acceso a información por sistemas internos o externos. **Orientación a servicios.**
- **Evita la modificación de los sistemas integrados y la implementación de estrategias ad-hoc.**

- **Mediación:** el sistema EAI se utiliza como sistema intercomunicador para difundir información entre los distintos sistemas.
- **Federación:** el sistema EAI actúa como punto de acceso entre sistemas. Toda la interacción con respecto al mundo exterior pasa por el sistema EAI, que filtra y adapta la información al contexto de los sistemas conectados.



managed / centralized



managed / decentralized

- **Hub-and-spoke:** el sistema EAI actúa como un concentrador, los mensajes tienen que pasar por el sistema para que éste los derive a los destinatarios pertinentes.
- **Bus:** el sistema EAI se utiliza como bus para entregar y recibir los mensajes. El bus posee la configuración apropiada para que ocurra el intercambio de información entre sistemas.

- **Compartir información entre sistemas utilizando un mecanismo común de almacenamiento de datos:**
  - Archivos: varios sistemas utilizan archivos como mecanismo para compartir información. La estructura y el contenido de estos archivos actúa como un contrato entre los sistemas que lo utilizan, siendo necesario que conozcan el formato y el medio de acceso.
  - Bases de datos compartidas: los distintos sistemas acceden a la misma base de datos donde pueden actuar como productores o consumidores de información. Antipatrón: BBDD como mecanismo RPC.
  - No existe el formato de representación de información universal.

## EAI – Antecedentes (II)

### ■ Invocación de Procedimientos Remotos

- Los sistemas participantes en la integración exponen una interfaz pública sobre la que el resto de sistemas pueden invocar remotamente operaciones e intercambiar datos. Encapsulamiento de funcionalidades.

### ■ Ventajas:

- Programación orientada a componentes. Encapsulamiento.
- El modelo de información para el intercambio de datos está bien definido a través de los prototipos de las operaciones expuestas. Acoplado a la implementación.
- Comunicación remota.

### ■ Inconvenientes:

- Los parámetros de las operaciones (entrada/salida) se especifican en las invocaciones y respuestas. Estas estructuras de datos pueden ser muy complejas.
- Sencillo implementar comunicación síncrona, pero muy complejo en comunicaciones asíncronas.
- Altamente dependientes del estado de la red. Las redes fallan.

## ■ Integración basada en Middleware de Mensajería

- Se presenta el mensaje como unidad de intercambio de información e interacción, permitiendo así un mayor nivel de abstracción. Los sistemas integrados son productores/consumidores de mensajes.
- Los datos son encapsulados en los mensajes, que son intercambiados entre las aplicaciones utilizando canales de mensajería.
- Las aplicaciones depositan y reciben los mensajes a través de canales de mensajería.
- El formato del mensaje puede ( debe ) adaptarse en función del interlocutor. Abraza la heterogeneidad de los mecanismos de representación de información en los sistemas integrados.

## ■ **Ventajas:**

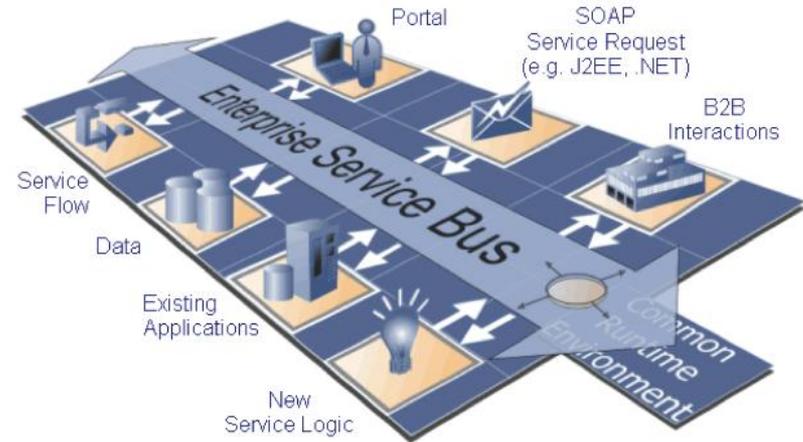
- Desacoplamiento entre sistemas. Interoperabilidad
- Capacidades extra-funcionales: persistencia, tolerancia a fallos, entrega garantizada, priorización de mensajes, balanceo de carga, alta disponibilidad ...
- Los sistemas sólo deben tratar los mensajes y su contenido.

## ■ **Desventajas** (por decir algo):

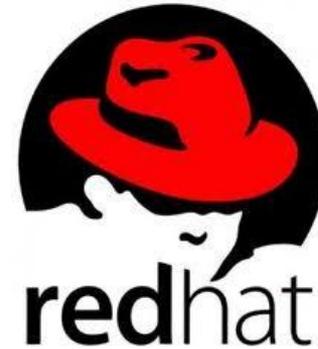
- Administración compleja.
- Consumo de recursos.

# ESB - Definición

- Es un middleware que permite a aplicaciones heterogéneas compartir información utilizando mensajería.
- Actúa como un concentrador entre los distintos sistemas de información y como una interfaz de comunicación entre aplicaciones.
- La principal abstracción es el *mensaje*. Las aplicaciones se mantienen razonablemente desacopladas entre sí.
- Soluciona el problema de la integración de aplicaciones en la misma medida que un sistema relacional de bases de datos soluciona el problema del almacenamiento de información relacional.



# Implementaciones Enterprise Service Bus ( ESB )



fabric8

RED HAT® JBOSS®  
FUSE

## Patrones de Integración

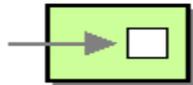
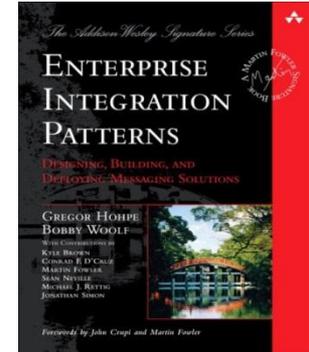
- **Un patrón es una solución general para problemas frecuentes y recurrentes. Los patrones aparecen de experiencias aprendidas por prueba y error.**
- **Los desarrolladores experimentados infieren sus propios patrones cuando reconocen un problema superado al que aplican la mejor solución experimentada.**
- **La integración de aplicaciones mediante mensajería dispone de sus propios patrones.**

## ■ Características

- Un patrón es una solución generalizada a problemas de ingeniería.
- Las soluciones de mensajería tienen sus propios patrones para ayudar a resolver los problemas de integraciones.

## ■ Clasificación de patrones

- Patrones de entrada/salida
- Patrones de construcción de mensajes
- Canales de mensajería
- Enrutado de mensajes
- Transformación de mensajes



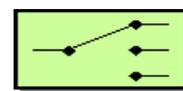
Message  
Endpoint



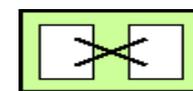
Event  
Message



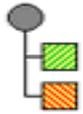
Message  
Channel



Message  
Router

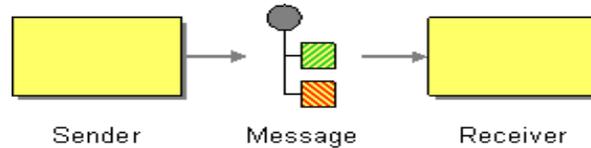


Message  
Translator



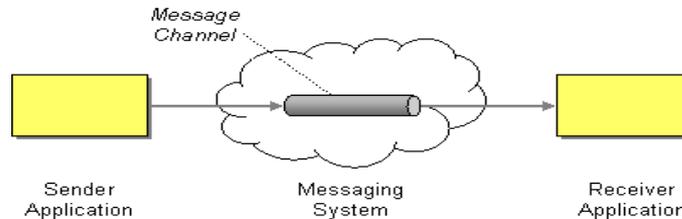
Message

- Un mensaje es una unidad atómica de información que las aplicaciones utilizan para compartir información.

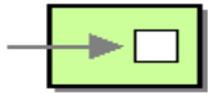


Message Channel

- Las canales de mensajería son rutas virtuales para el intercambio de mensajes entre aplicaciones.

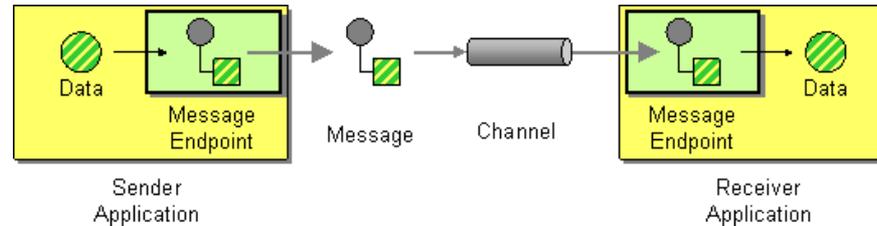


## Message Endpoint

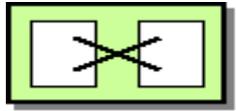


Message  
Endpoint

- Es un punto de acceso de entrada/salida al que los participantes de una comunicación pueden enviar/recibir mensajes.
- Normalmente son proporcionados por el sistema de integración.
- Iniciativa de estandarización: AMQP (Advanced Messaging Queuing Protocol).

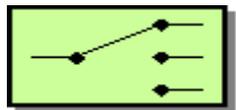


## ■ Message Translator



Message  
Translator

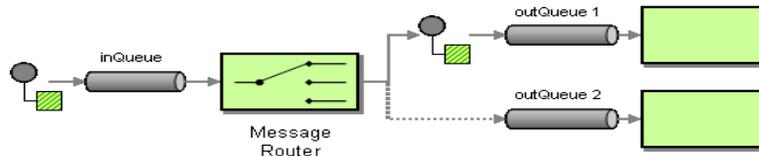
- No todas las aplicaciones utilizan el mismo formato para la representación de información, por lo tanto es necesario traducir entre los distintos formatos.
- Realiza la transformación del formato de los mensajes para su adaptación con respecto a los destinatarios.
- Puede extraer porciones del mensaje original para generar varios mensajes.
- Si los mensajes de intercambio son de tipo texto/XML existen herramientas muy desarrolladas para su transformación y/o extracción de información:
  - XSLT, XPath, XQuery



Message Router

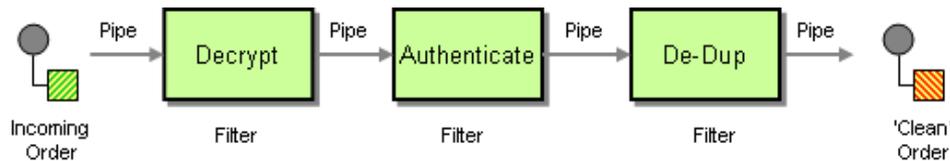
## ■ Message Router

- Permite procesar mensajes en base a su contenido. Implementa la lógica de integración. Rutas de integración.
- Un enrutador de mensajes consume mensajes de un determinado canal de mensajería, y los vuelve a publicar en diferentes canales de mensajería según un conjunto de condiciones definidas sobre el propio mensaje.

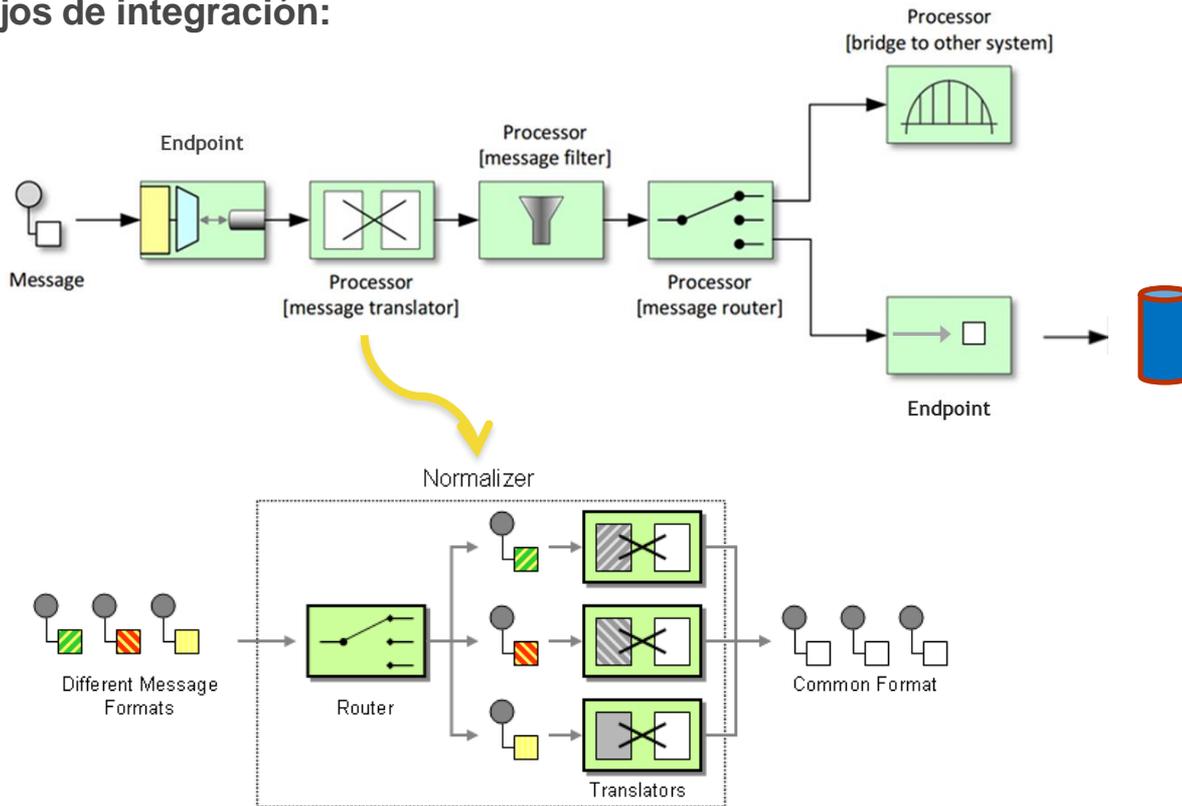


## ■ Message Pipeline

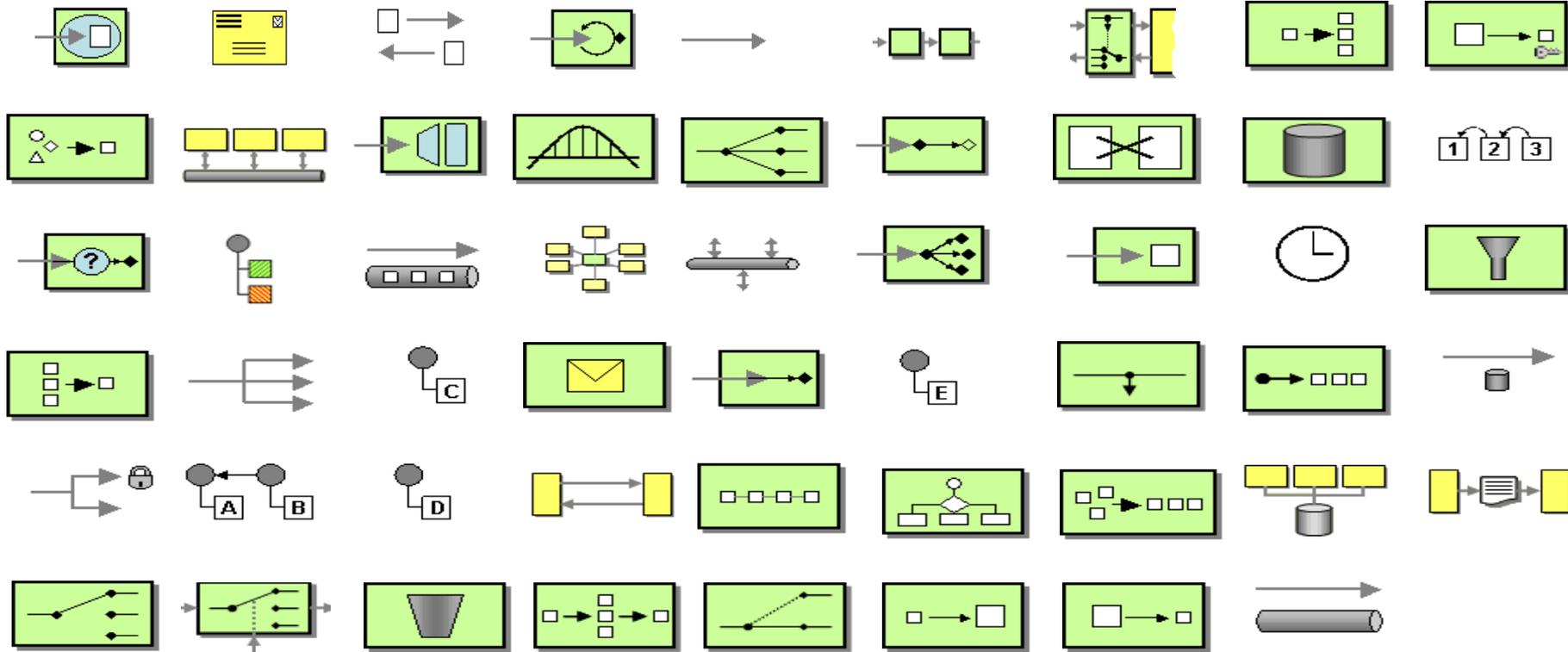
- Permite una mayor complejidad en el tratamiento de los mensajes.
- Modela el flujo de información entre sistemas como una secuencia de actividades sobre el proceso de mensajes.
- El mensaje es tratado por distintos componentes con tareas específicas, siendo el resultado la implementación del flujo de información.
- La aplicación de éstos procesados pueden ocurrir de manera secuencial o en paralelo.



- Flujos de integración:



# Enterprise Integration Patterns

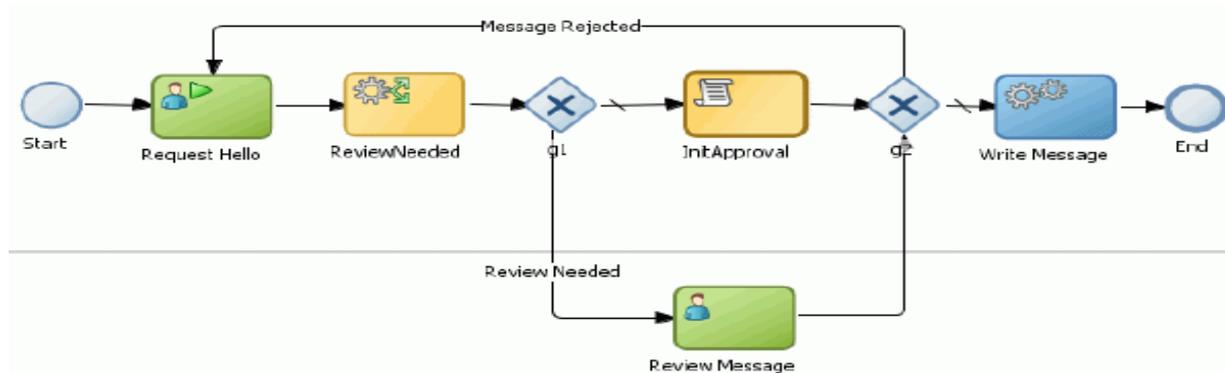


Y muchos mas...

## Otras Tecnologías Relacionadas

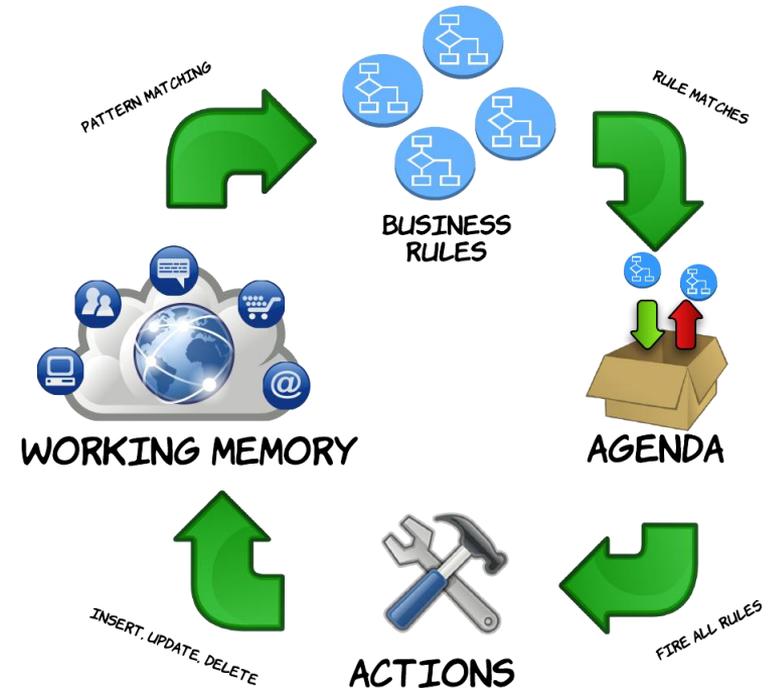
- **BPM: Business Process Management**

- Modela de manera abstracta procesos de negocio empresarial.
- Proporciona un lenguaje independiente para la especificación de estos procesos que será ejecutado por el motor de ejecución de procesos de negocio. Independiente de la implementación.
- Modela tanto la interacción automática entre sistemas como la participación de humanos en el proceso. El humano pasa a ser un participante mas del proceso de negocio.



## ■ BRM: Business Rules Management

- La implementación de procesos de decisión se modela como reglas de negocio.
- El conjunto de reglas de negocio se aplican al flujo de la comunicación para la identificación de patrones ( cumplimiento de reglas ) que permitan la ejecución de procesos de negocio.
- Las reglas de negocio se modelan en un escenario causa/efecto.



## ■ CEP: Complex Event Processing

- La disponibilidad de rutas de integración basadas en patrones, la existencia flujos múltiples de eventos desde/hacia los sistemas integrados, la aplicación de tecnologías de gestión de procesos de negocio y reglas de negocio favorecen la implementación de sistemas de procesamiento complejo de eventos.
- La integración de aplicaciones se modela como un flujo continuo de eventos entre sistemas en forma de mensajes, la toma de decisiones se modela como un conjunto de reglas de negocio que desencadenan como consecuencia la ejecución de procesos de negocio. Se implementa un ciclo continuo que se retroalimenta.
- Permite la implementación flexible e interoperable de casi cualquier escenario de integración independientemente de su complejidad.

## Experiencias en Connectis

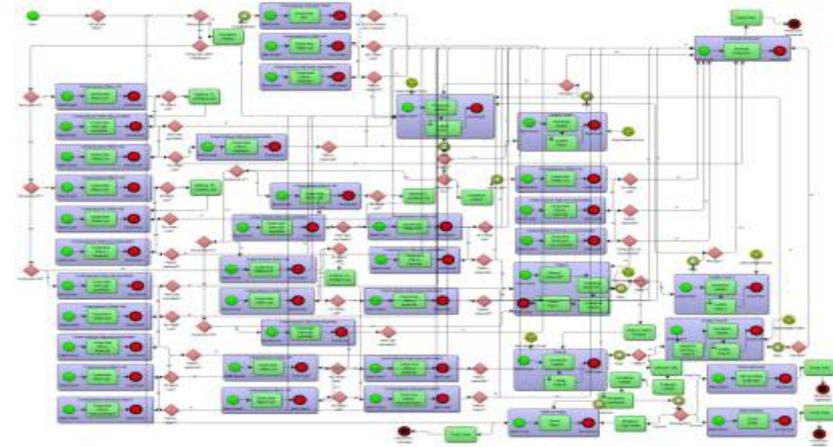
## Perth Children's Hospital ELV Head-End

- **Integración de gestión de edificio para el nuevo hospital para niños en Perth**
  - Sistema de procesamiento complejo de eventos con reglas de negocio, procesos de negocio y patrones de integración EAI
  - Integración con sistemas de información heterogéneos externos e internos
  - Integración con sistemas SCADA de monitorización y control
  - Arquitectura orientada a servicios y modular
  - Gestión centralizada de alarmas
  - Entorno centralizado de administración, monitorización y control basada en portal web.
  
- **Entorno Tecnológico: Java, Red Hat JBoss Fuse, Apache Karaf, Fabric8, JBoss, Liferay, MongoDB, RabbitMQ, InfluxDB, PostgreSQL**



## Sidney Harbour Bridge Visual Image Processing

- **Sistema de tolling free-flow basado en reconocimiento óptico de caracteres**
  - Sistema de procesamiento complejo de eventos con reglas de negocio, procesos de negocio y patrones de integración EAI
  - Integración con pórticos de detección de vehículos, armado de tránsitos compleja mediante reglas con ventanas temporales.
  - Soporte de distintos módulos de reconocimiento óptico de caracteres orientados a reconocimiento de matrículas
  - Arquitectura orientada a servicios y modular
  - Entorno centralizado de administración y gestión de tránsitos basada en aplicación web.
  
- **Entorno Tecnológico: Java, Apache ServiceMix, Adobe Flex, ALPNR, JBoss, Oracle**

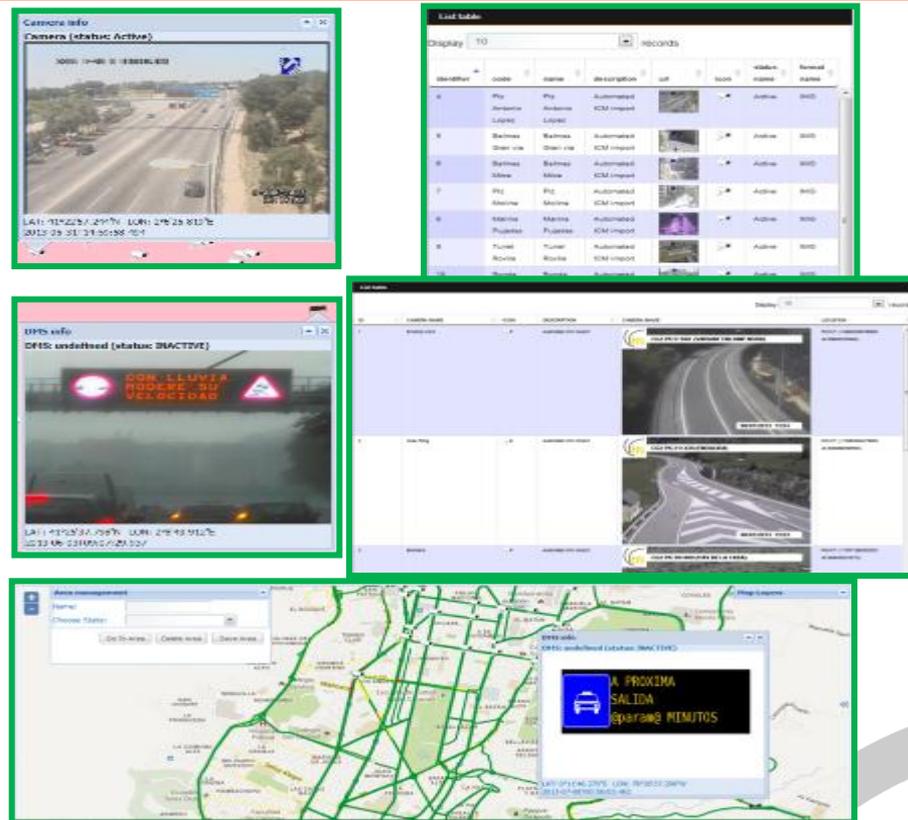


## Dissemination Web Platform for Smart Cities

### ■ Plataforma de publicación de información para smart city

- Plataforma de integración basada en patrones de integración EAI entre las fuentes de datos heterogéneas de una gestión smart city en una ciudad.
- Publicación dinámica de información relevante para distintos actores en la gestión de una smart city en forma de mapas, tablas, listas de eventos, ....
- Arquitectura orientada a servicios y modular
- Entorno centralizado de administración y gestión de información basada en portlets reutilizables desplegados en portal web

### ■ Entorno Tecnológico: Java, Apache ServiceMix, Liferay, ActiveMQ, JBoss, PostgreSQL



## En4Sys – Traffic enforcement

- **Gestión de infracciones de tráfico para ciudades inteligentes**
  - Sistema de gestión de los procesos administrativos de validación, publicación y sanción de infracciones de tráfico.
  - Proceso administrativo gobernado por procesos de negocio configurables y adaptables a las necesidades de la ciudad.
  - Arquitectura orientada a servicios y modular
  - Entorno de procesamiento de infracciones basado en pipelines de integración en contenedores OSGi.
  - Aplicación web de configuración y operación basada en aplicación web configurable altamente cohesionada con el proceso de negocio subyacente.
  - Soporte de comunicación con sistemas de captura automática de matrículas.
  - Soporte para múltiples proveedores de sistemas de captación en forma de módulos como datos de entrada, múltiples formatos de infracción y comunicación con sistemas de terceros ( T-Systems, Corpaire, DGT, ... )



- **Entorno Tecnológico: Java, Apache Felix, JBoss, AI PNR, PostgreSQl**



More than computing, People.

