

# Fundamentos de la integración de sistemas software

Profesor: **Alejandro Sirvent Llamas**

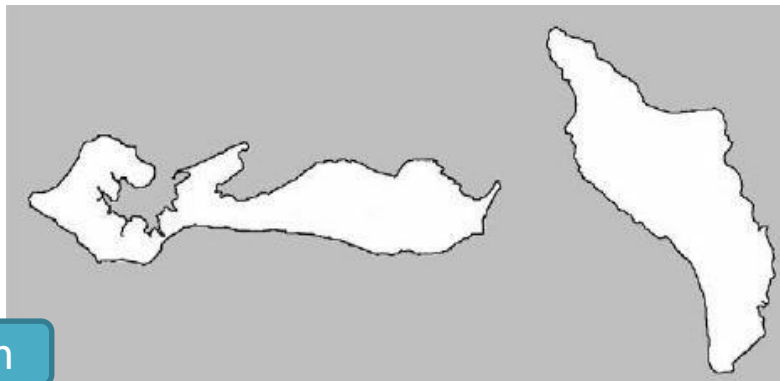
- Situación actual:
  - Liberalización comercio internacional
    - **Globalización**
  - Continuo desarrollo de tecnologías
    - **Moore's Law** (Se dobla la capacidad de las máquinas cada 18 meses más o menos)
  - Explosión de redes
    - **Internet**

- Situación actual:
  - Empresas
    - Sistemas de información extensos, dispersos y muy interconectados.
    - Globalización: competencia global.
    - Fusiones y absorciones: tamaño y peso necesario para competir.
    - Intercambio de información: redes.

- Situación actual:
  - Empresas
    - Reducido **time-to-market**
      - Se refiere al **tiempo que transcurre entre que un producto es creado y su lanzamiento al mercado.**
      - Ejemplo: Google que consigue lanzar productos al mercado que todavía están en fase de desarrollo y esos productos son adquiridos por millones de personas mientras ellos continúan con el desarrollo y las mejoras.
      - Ejemplo Apple con iPhone:
        - » Impensable esperar 3-4 años por un nuevo iPhone.
        - » La evolución de las tecnologías haría que el proceso de gestación del dispositivo no acabará nunca → Siempre habría mejoras tecnológicas.
        - » Por ello, **el Time to Market, cada vez es más corto y precipitado y conlleva la aparición constante de productos** que están muy lejos de ser perfectos y, en realidad, **cada novedad no es más que la versión mejorada del anterior.**

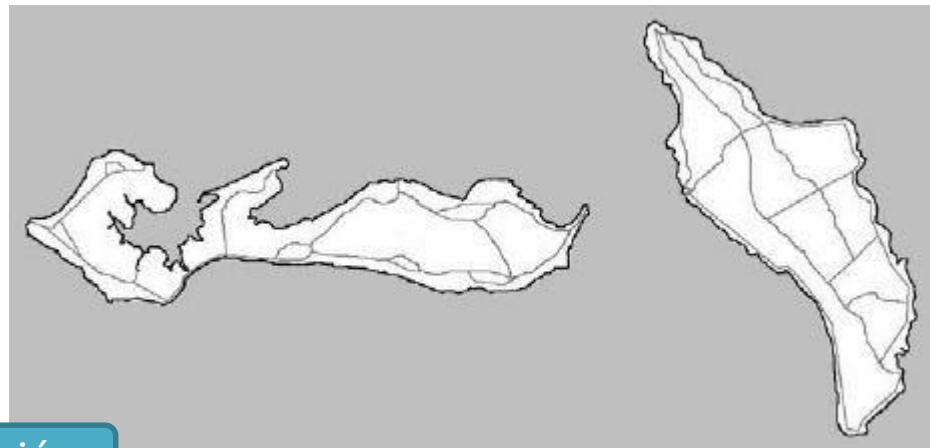
- Situación actual:
  - Empresas
    - Mejora de la reactividad y adaptabilidad **Zero Latency Enterprise (ZLE)** .
      - Concepto de *Gartner Group* que consiste en minimizar el periodo de tiempo que transcurre entre la captura de la información y la utilización de la misma.
      - ZLE hace referencia al **acercamiento en tiempo real de la información** y la administración de la relación con el cliente **permitiendo a los usuarios acceder a esta en tan sólo unos segundos**.
      - La entidad que adopta esta estrategia, hace que el **acceso a la información** y los **procesos de intercambio** de la misma se produzcan en **tiempo real** → ventaja competitiva para las **empresas**, que pueden **acortar sus tiempos de decisión, reacción y adaptación**
      - La latencia **no puede ser** literalmente **cero** en ningún sistema real porque las computadoras necesitan tiempo para "pensar".

- Historia de integración:
  - Islas (IT sin comunicar), a principios de 1960.
    - Cada empresa o departamento que había sido informatizado vivía tranquilamente en su propio universo, con sus aplicaciones específicas y sin interacción con el exterior.
    - “El océano era vasto e ilimitado, y los barcos nunca conocieron a otros buques”.
    - La información llegaba a los oídos de los habitantes, a cargo de los pájaros o las botellas a la deriva en el mar.



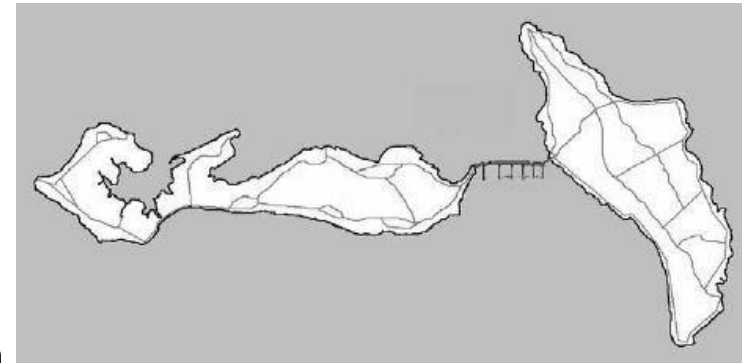
- Historia de integración:
  - Con el aumento de los intercambios económicos y el progreso continuo de tecnologías de computación
    - A partir de la informatización de la nómina o de contabilidad , aparecieron aplicaciones en la isla de gestión de clientes o de producción.
    - “se añadieron fortalezas , mazmorras y castillos”.
    - Surgió una necesidad de comunicación entre aplicaciones:
      - » **EAI**: Enterprise Application Integration.
      - » **A2A**: Application to (two) Application.

- Historia de integración:
  - Los caminos entre diferentes aplicaciones fueron “excavados”.
  - No había ningún mapa
  - “número de vehículos que circulaban fue en aumento”
  - “Para evitar colapso, se fue urbanizando” las aplicaciones.

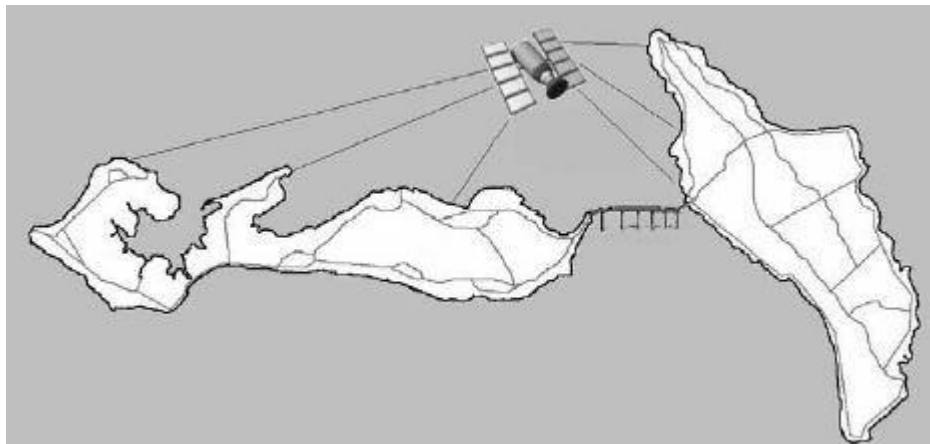




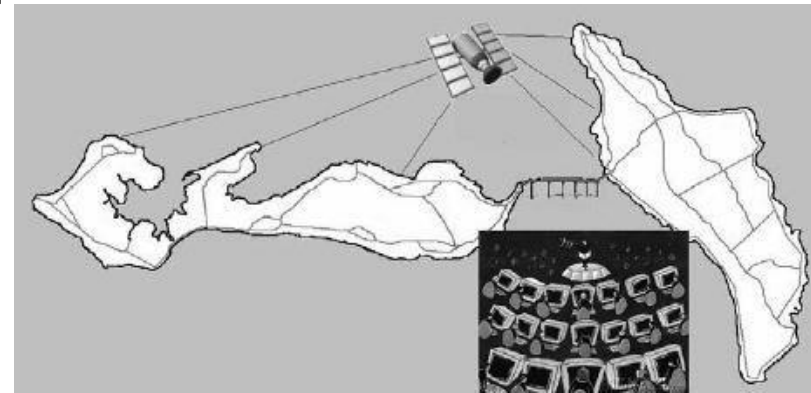
- Historia de integración:
  - El océano hizo su parte al traer información desde el exterior directamente a la playa
    - Sólo unos pocos botes llegaban a las playa - en forma de **disquetes** o **cintas magnéticas**.
    - El contenido era transportado a lomos de mulas a los dominios indicados.
    - La frecuencia de llegada de los barcos se aceleró y se necesitaron puertos, nació el **B2B**: Business to Business.
    - Las rutas marítimas prosperaron , y con ellos, los riesgos de colisión o la piratería → se crearon puentes
    - Nació **MOM**: Message-Oriented Middleware.



- Historia de integración:
  - Se tenían que comprobar que la información que estaban produciendo se ajustaba a lo que se esperaba para avanzar en el negocio.
  - Para controlar el tráfico fueron necesarios “los satélites de observación”.
  - Nació **BAM**: Business Activity Monitoring.



- Historia de integración:
  - Un conjunto completo de tecnologías refuerzan la actividad y la armonía que debe reinar en las islas
  - Surgen dos elementos complementarios :
    - **SOA**
    - **BPM**
    - “los servicios (SOA ) son como un grupo de músicos en una orquesta, cada uno toca sus partituras musicales, pero bajo la dirección de un director de orquesta ( BPM )

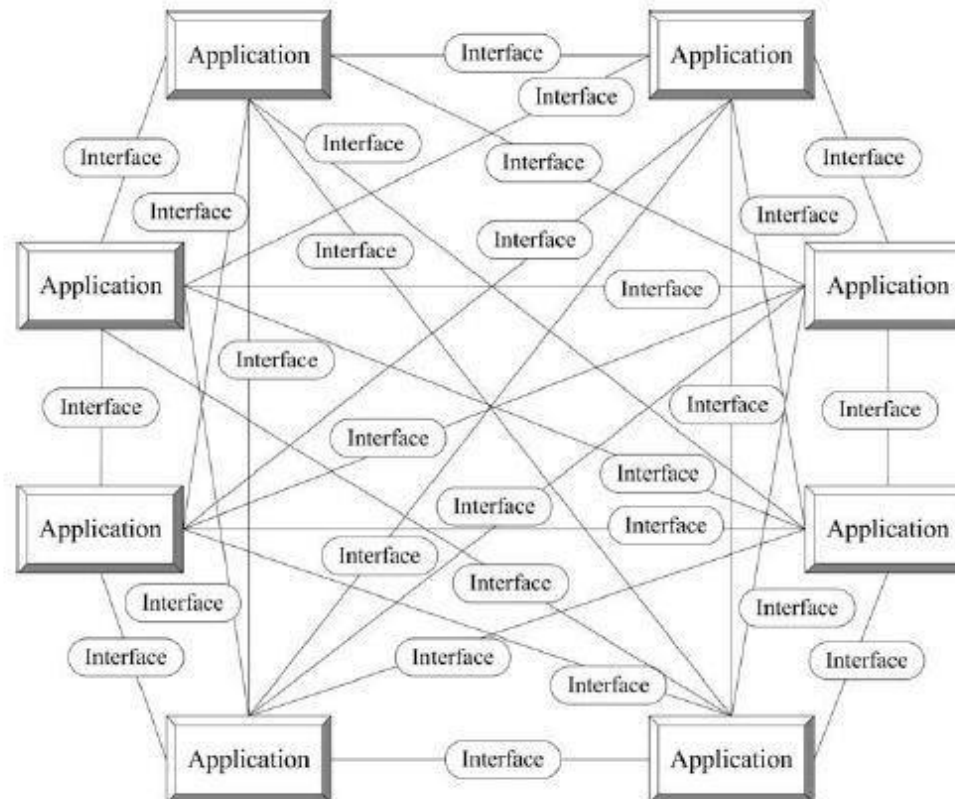


- Historia de integración:
  - Complejidad
    - Todas las aplicaciones con todas
    - Número de interfaces bidireccionales

$$n(n-1)/2$$

- 6 aplicaciones → 15 interfaces
- 150 aplicaciones → 11175 interfaces

- Con esta complejidad estamos creando un “spaghetti system”



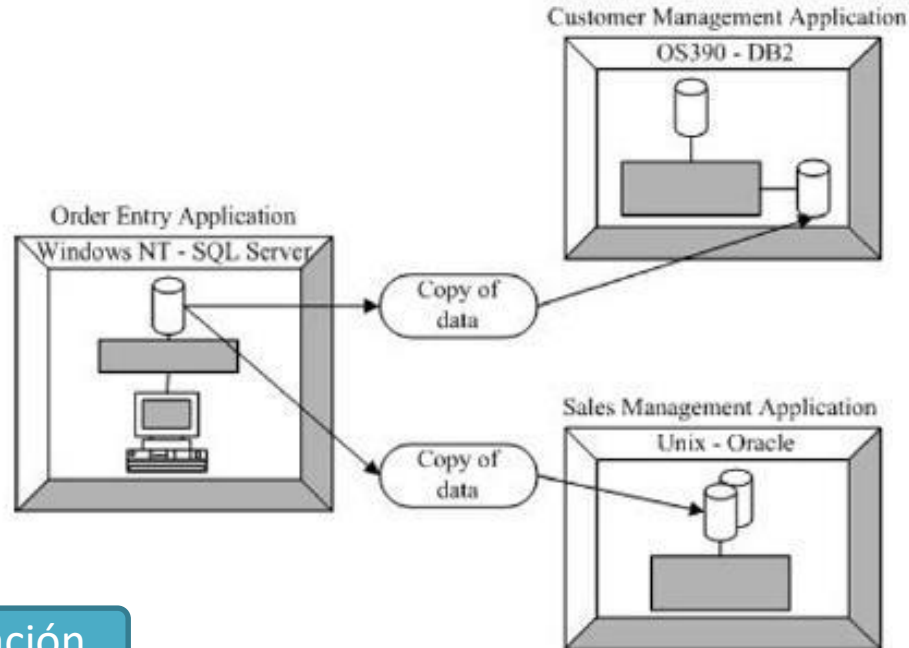
- Problemas de integración:
  - Propagación y consistencia de datos.
  - Gestión de procesos multifase.
  - Creación de aplicaciones compuestas.
  - Ejemplo:
    - Consideramos tres aplicaciones, una para tomar un pedido, una para la gestión de clientes, y uno para la gestión de ventas.
    - Cada aplicación funciona en diferentes plataformas y utiliza tecnologías apropiadas para esas plataformas.
    - Cada uno utiliza información acerca de los clientes empresariales. La consistencia de la información entre estas tres aplicaciones deben garantizarse.

- Problemas de integración:
  - Propagación y consistencia de datos
    - La razón principal para implementar una solución de integración de aplicaciones, suele ser propagar datos e imponer consistencia en los datos.
    - Esto implica asegurar la consistencia de los datos almacenados de forma redundante por diferentes aplicaciones distribuidas geográficamente.
    - Se pueden utilizar 2 métodos:
      - Repository approach (copia de datos)
      - Event-based approach

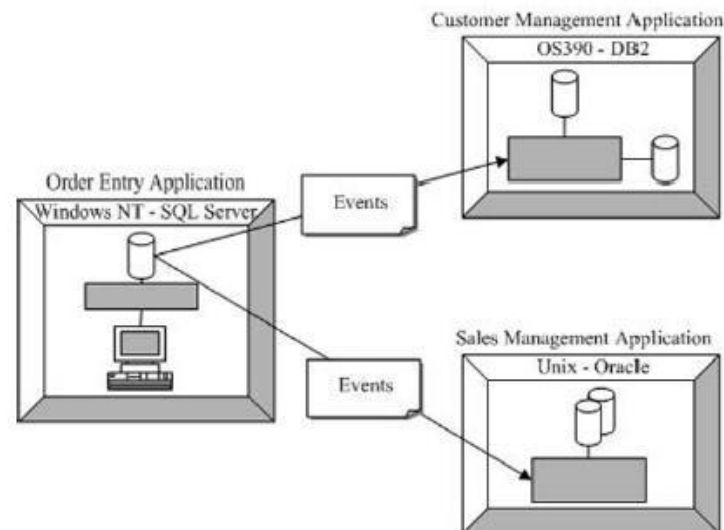
- Problemas de integración:
  - Propagación y consistencia de datos
    - Repository approach (copia de datos)
      - El enfoque repositorio: copia los datos directamente de la BD de la toma de pedidos) , con las BD que se utilizan por las otras dos aplicaciones.
      - Supone modelos compatibles.
      - No preserva integridad, ya que no hace posible la aplicación de la lógica de actualización y normas de gestión de esas otras aplicaciones.



- Problemas de integración:
  - Propagación y consistencia de datos.
    - Repository approach .



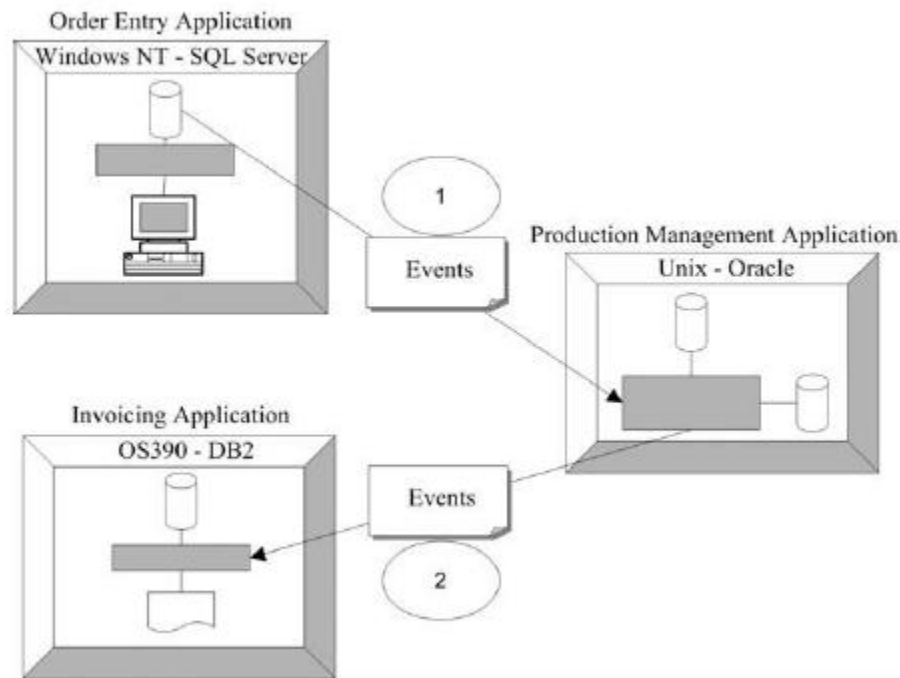
- Problemas de integración:
  - Propagación y consistencia de datos
    - Event-based approach
      - Transmisión de un evento con cambios a las otras aplicaciones, que garanticen su tratamiento, la actualización de sus propios recursos por sí mismos.



- Problemas de integración:
  - Gestión de procesos multifase
    - Multiproceso no necesariamente lineal
      - La integración, trata un proceso de negocio que comprende de varios pasos con las interacciones asíncronas y unidireccionales, (lógicamente interdependientes)
      - El procesamiento realizado por cada paso depende del procesamiento realizado por los otros pasos.
  - Implicación humanos

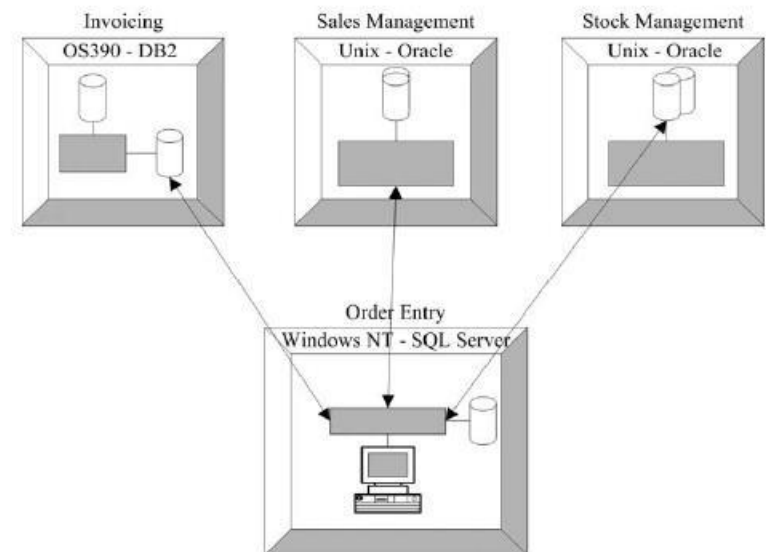
- Problemas de integración:
  - Gestión de procesos multifase, Ejemplo:
    - Consideramos tres aplicaciones: una para tomar pedidos, uno para gestionar la producción, y otra para manejar la facturación.
    - El objetivo es la integración de todo el proceso en un enfoque tipo: tomar la orden - fabricar el artículo - facturar.
    - Esta integración requerirá la secuenciación de las tres aplicaciones.

- Problemas de integración:
  - Gestión de procesos multifase, Ejemplo:



- Problemas de integración:

- Creación de aplicaciones compuestas.  
“composite application”
  - Aquí se trata de crear una nueva aplicación mediante el uso de los datos o de los servicios disponibles en las aplicaciones existentes.
  - Como ejemplo:
    - Aplicación para el manejo de las ventas en la Web.
    - Llamadas a 3 servicios: facturación, gestión de ventas y gestión de inventario



- Tipos de aplicaciones:
  - *Batch applications.*
  - Transactional applications.
  - *Client/server applications.*
  - *Web applications.*
  - *Real-time applications.*
  - *Software packages.*

- Tipos de aplicaciones:
  - Batch applications
    - Ritmo de proceso (Pace). Periódicamente, por grupos. Dependiente de:
      - Volumen de datos.
      - Dependencias con otras aplicaciones.
      - Organización interna de los procesos empresariales.
    - Formato: flat format, creadas con lenguajes antiguos tipo COBOL.
    - Volumen: millones de eventos en un batch
    - Conectividad: uso de ficheros o bases de datos



- Flat format – longitud y posición fija  
[dato1][dato2]...[dato n]
- Flat format – longitud y posición variable (datos secuenciales)  
[ctr1][dato1][ctr2][dato2]...[ctr n][dato n]
- Flat format – con delimitadores (datos secuenciales)  
[dato1],[dato2],...,[dato n]
- Flat format – con delimitadores (datos no secuenciales)  
Key2 = dato2, key n = dato n, ..., key1 = dato1
- Hierarchical or tree-structure format  
Key1 = (dato1, dato2, key7=(dato14, dato15)), key5=dato5, ...
- XML format

- Tipos de aplicaciones:
  - Aplicaciones Transaccionales
    - Ritmo de proceso (Pace), procesan los eventos a medida que ocurren en producción.
      - Interacción de usuarios
      - Coexistencia con otras aplicaciones (posible problema de acceso)
    - Formato: dependiente de las tecnologías (más complejo que Batch)
    - Volumen: gran número de eventos de forma simultánea
    - Conectividad: uso de bases de datos y la interfaz de pantalla.

- Tipos de aplicaciones:
  - Aplicaciones Cliente/servidor
    - Ritmo de proceso (Pace). Conforme ocurren en producción
    - Formato: Estructuras de datos complejas
    - Volumen: prima la simultaneidad
    - Conectividad: se usa base de datos y la interfaz de pantalla.

- Tipos de aplicaciones:
  - Aplicaciones WEB
    - Ritmo de proceso (Pace). Los eventos se procesan como y cuando se producen en la producción, como características principales:
      - Disponibilidad.
      - Ergonomía, adaptándose a las condiciones del usuario.
    - Formato: Estructuras de datos complejas (XML,JSON)
    - Volumen: Simultaneidad es crítica. Volumen de usuarios puede ser desconocido.
    - Conectividad: usan tecnología web (HTTP, SMTP, etc.)

- Tipos de aplicaciones:
  - Aplicaciones en tiempo real.
    - Funcionan según lógicas que puede operar tanto en lotes como en modo transaccional (Híbridos).
    - Ritmo de proceso (Pace). Tiempo real.
    - Formato: Estructuras de datos complejas (XML,JSON)
    - Volumen: volumen elevado de proceso
    - Conectividad: a través de API's

- Tipos de aplicaciones:
  - Paquetes de software
    - En función de su destino, los paquetes de software comerciales exhiben características de un tipo u otro.
    - Los ERP, por definición abarcan numerosas funciones en el negocio de TI.

## EAI – Enterprise Application Integration

- Colección de **métodos, herramientas y servicios** que **operan** unidos para **intercomunicar aplicaciones** heterogéneas, que forman parte de una **empresa** tradicional o distribuida.
- El **concepto de EAI no** es para nada **nuevo**.
  - Es tan antiguo como la necesidad de intercambiar datos entre dos aplicaciones separadas y las empresas han tenido este problema casi desde el mismo momento en que empezaron a usar sistemas de software.
- **Dificultades** de EAI
  - Las dificultades más inmediatamente visibles, aunque no necesariamente las más importantes, son principalmente **técnicas** (seguridad, escalabilidad).

## EAI – Enterprise Application Integration

- EAI, ocupa el dominio más antiguo de la integración: el dominio de los negocios Application to Application "A2A".
- Este suele ser el ámbito de la integración de:
  - ERP
  - Integración de front-office
  - Aplicaciones con las aplicaciones de back-office
  - Integración de software para Customer Relationship Management (CRM)
  - Supply Chain Management (SCM)
  - ...



## B2B – Business to business (integración)

- Colección de **métodos, herramientas y servicios** que **operan** unidos para **intercomunicar aplicaciones** TIC heterogéneas en múltiples **empresas** con el objetivo de realizar transacciones comerciales.
- Se dice de los **procesos** cuyo intercambio de información entre ellos se produce **sin intervención** de las **personas**.

## B2B – Business to business (integración)

- Ejemplo:
  - Se utiliza como terminología para hablar de la transmisión de información entre fabricantes y distribuidores de un producto.
- Permite a las empresas digitalizar, integrar, automatizar y optimizar los flujos de información y los procesos empresariales clave que se extienden a través de los ecosistemas empresariales internos y externos.

## **B2B – Business to business (integración)**

- Podemos decir que la integración B2B corresponde a la integración A2A entre empresas.
- Las diferencias derivan fundamentalmente de lo que la empresa no puede controlar:
  - La red de comunicación.
  - El formato de intercambio.
  - El proceso de intercambio.

## B2C – Business to Customer

- Conjunto de **métodos, herramientas y servicios** que **trabajan unidos** para **intercomunicar** las **aplicaciones** heterogéneas de una **empresa** con el objetivo de proporcionar a los **clientes** un acceso directo a los servicios ofrecidos por la empresa.

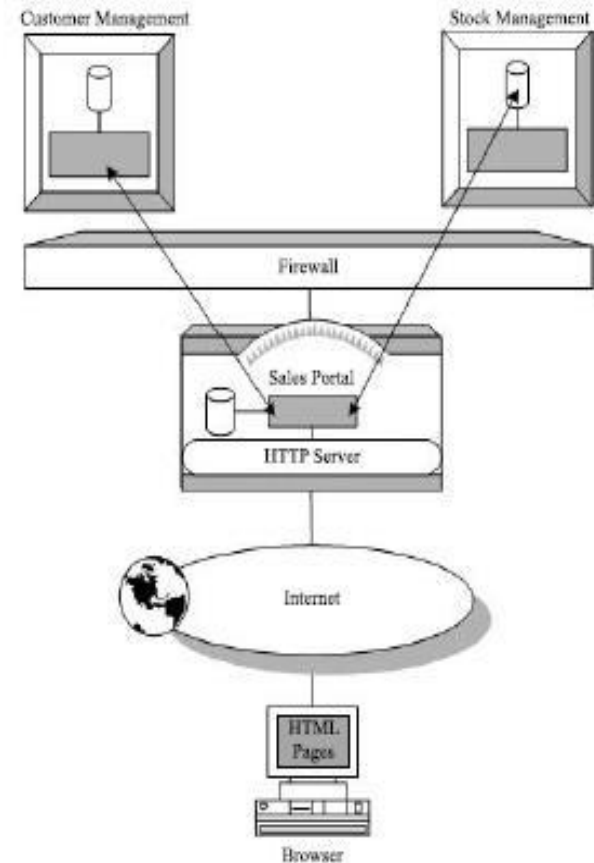


Figure 2.11. Example of integrating exchanges with the consumer

## BPM – Business Process Management

- Conjunto de **métodos, herramientas y servicios** para **modelar, ejecutar y optimizar procesos de negocios** en la empresa.
- **Enfoque centrado en los procesos** para **mejorar el rendimiento** que **combina** las **tecnologías** de la información con **metodologías** de **proceso y gobierno**.
- **BPM** es una colaboración entre personas de negocio y tecnólogos para fomentar procesos de negocio efectivos, ágiles y transparentes.

- **BPM – Business Process Management**
- Se incluyen en esta definición, los procesos en la cadena de valor que son horizontales para las organizaciones en el empresa, y que se extienden más allá de los socios de la empresa, incluyendo proveedores, clientes, bancos, etc...
- Ventajas:
  - Transparencia en todas las etapas del proceso.
  - Un mayor control administrativo.
  - Aumento de productividad.
  - Reducción de costos.
  - Automatización de procesos y generación de evidencias.

## BAM – Business Activity Monitoring

- El objetivo, es proporcionar acceso en tiempo real a los indicadores críticos de rendimiento de negocio para mejorar la velocidad y la eficacia de las operaciones de negocio.
- BAM **analiza** los datos, crea informes y alertas relacionadas con los eventos importantes y relevantes para el buen funcionamiento del negocio → Toma de **decisiones** sea **rápida** y **eficaz**.

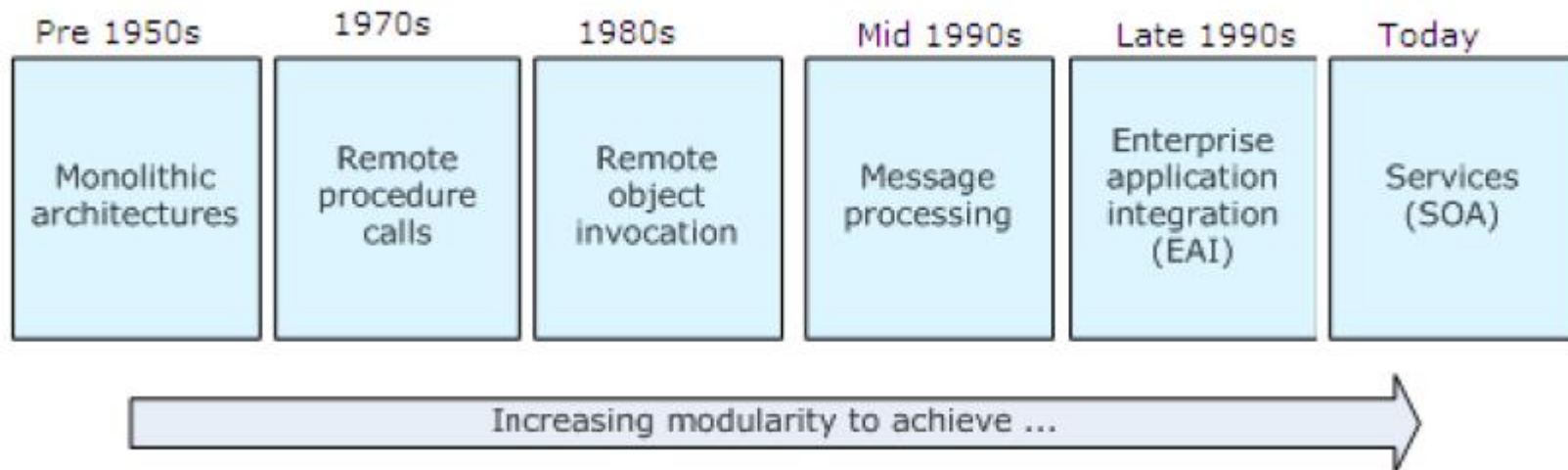
## BAM – Business Activity Monitoring

- Una herramienta BAM **no es una fuente de datos** pues la herramienta sólo recopila y muestra los datos de otros sistemas o fuentes de datos.
- El **uso adecuado** de un sistema BAM depende de la **correcta definición de indicadores** clave para el análisis del negocio.
- Utilizar un sistema BAM **permitirá conocer en tiempo real** los temas importantes que estás sucediendo en el negocio, **sin tener que buscar información en varios lugares**, ya que estas herramientas son capaces de consolidar toda la información y presentarla de forma práctica y fácil de entender.



## SOA – Service-Oriented Architectures

- Es un **paradigma** para **organizar y usar medios distribuidos** que pueden ser propios o estar bajo el control de dominios diferentes.
- SOA es un estilo arquitectónico para la construcción de aplicaciones de software en base a servicios disponibles.  
Características:
  - **Reutilización.**
  - Aumenta la eficiencia en los procesos.
  - Amortiza la inversión realizada en sistemas.
  - Reduce costes de mantenimiento.
  - Fomenta la innovación orientada al desarrollo de servicios.
  - Simplifica el diseño, optimizando la capacidad de organización.

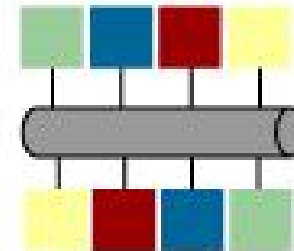


**Messaging Backbone**

- Point-to-point connection between applications
- Simple, basic connectivity

**Enterprise Application Integration (EAI)**

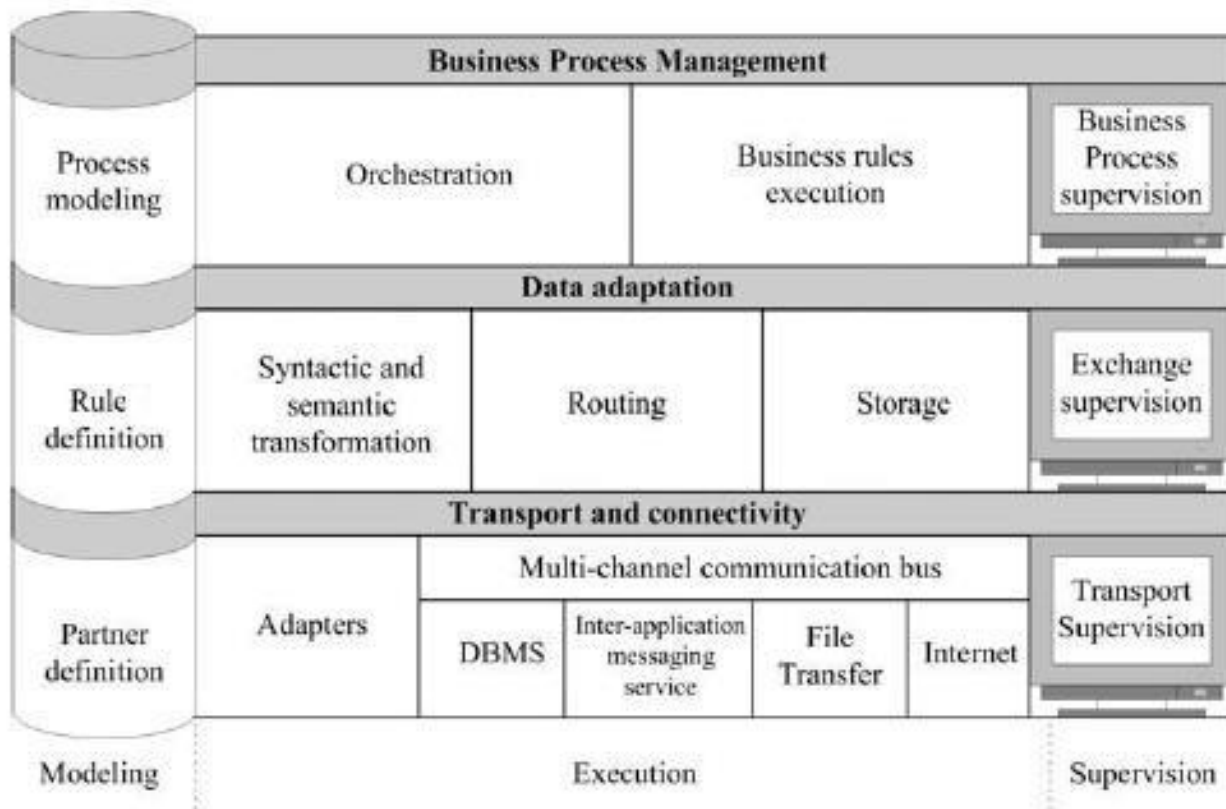
- EAI connects applications via a centralized hub
- Easier to manage larger number of connections

**Service Orientated Integration**

- Integration and choreography of services through an Enterprise Service Bus
- Flexible connections with well defined, standards-based interfaces

Flexibility

- **Niveles de servicio**
  - Transporte y conectividad
  - Adaptación de la información
  - Procesos de negocio
  - Cada nivel presenta 3 visiones
    - Modelado
    - Ejecución
    - Supervisión



**Figure 3.1.** Components in the integration infrastructure

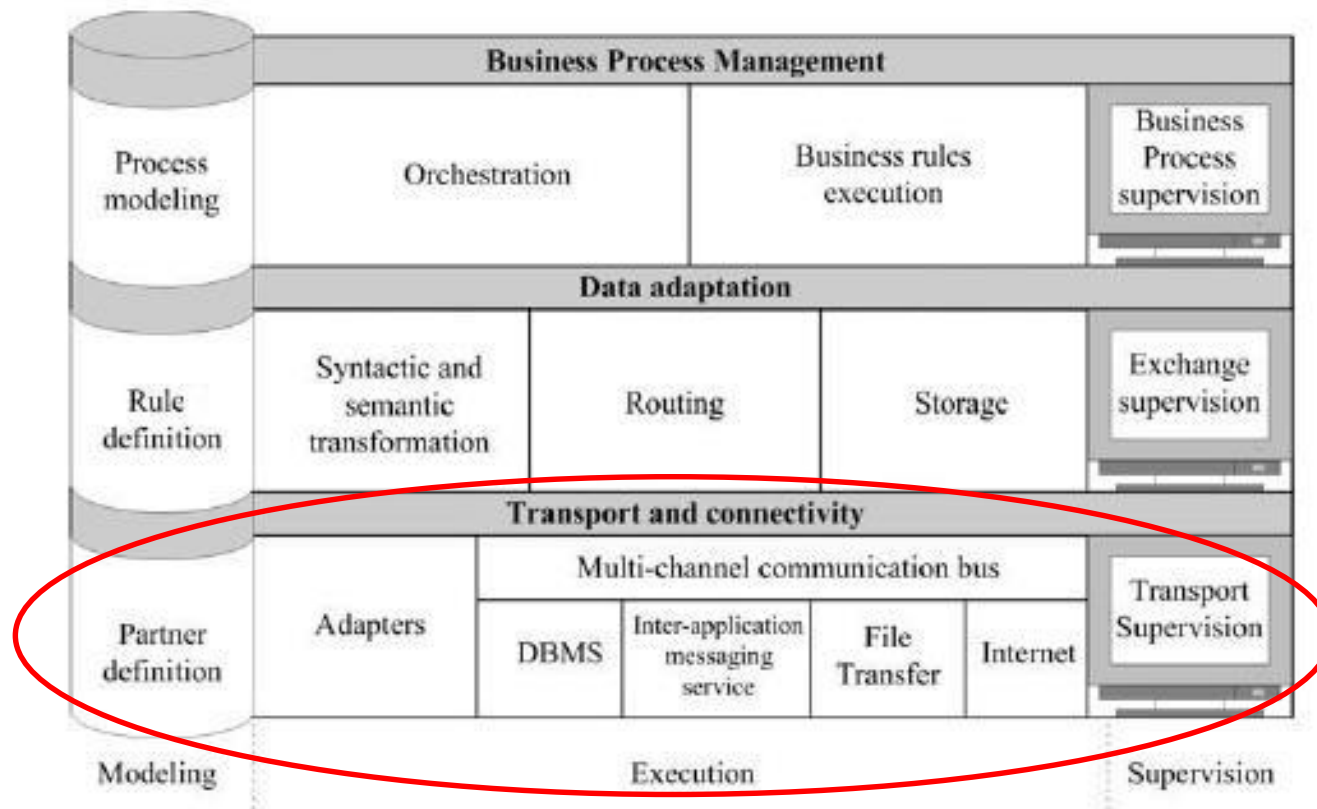


Figure 3.1. Components in the integration infrastructure

- Modelado
  - Define los participantes en el intercambio de datos y las reglas para este intercambio.
  - Repositorio (externo y centralizado) , contiene la descripción de los elementos que permiten que datos y/o eventos pasen de una aplicación a otra.
    - Topografía de las aplicaciones integradas
      - Localización, plataforma, canal de comunicación, dirección de red, protocolo a usar, identificador del gestor de base de datos, identificador de tablas, etc.
    - Condiciones para captar/ enviar datos o eventos
      - Timeslot, trigger de aplicación, etc.
    - Seguridad
      - Autenticación, encriptado, etc.

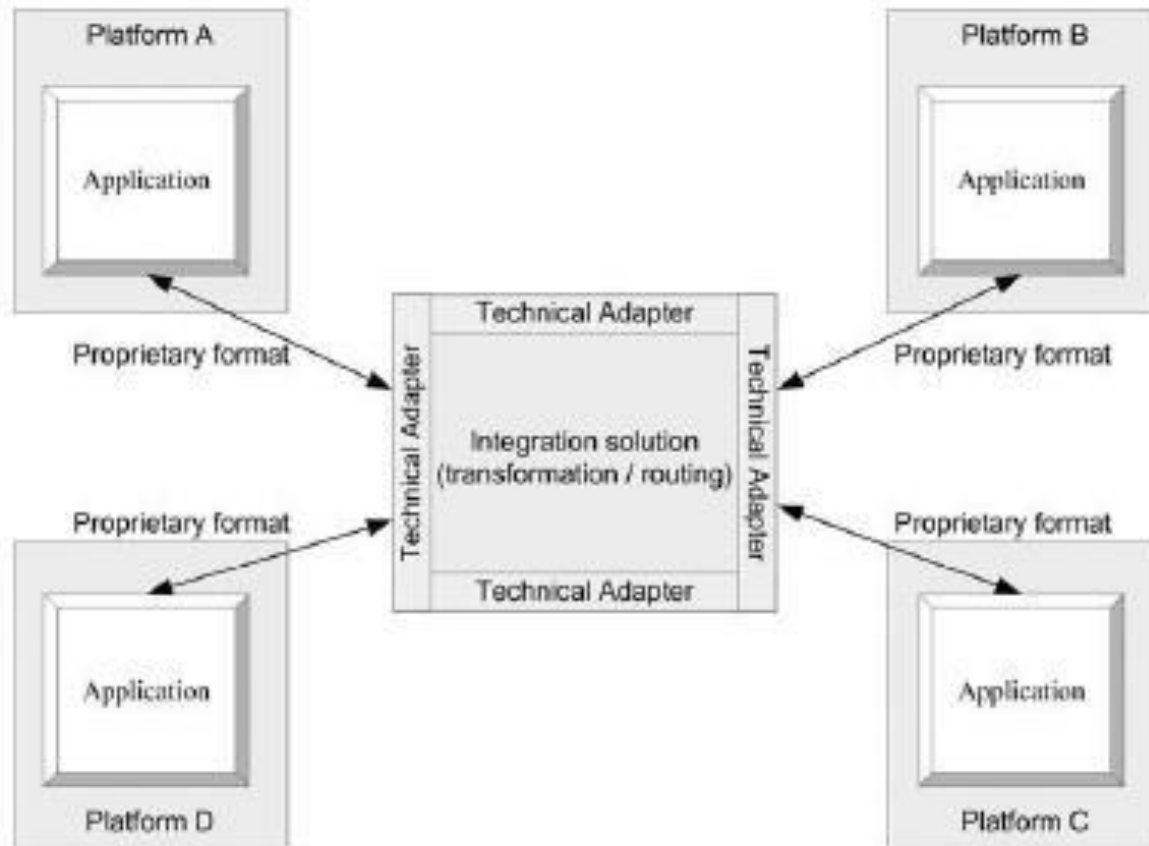
- Ejecución – Canales principales para transporte de datos
  - Sistemas de gestión de base de datos
    - Ofrece replicación
  - Transferencia de ficheros
    - Antiguo, sencillo de implementar
    - Ej: PeSIT y ETEBAC en banca, OFTP en la industria del automóvil, etc.
  - Sistemas de mensajería entre aplicaciones
    - MOM – Message-Oriented Middleware



- Ejecución – Canales principales para transporte de datos
  - Internet
    - HTTP – Hypertext Transfer Protocol
    - SMTP – Simple Mail Transfer Protocol
    - SOAP – Simple Object Access Protocol
    - ebMS – ebXML Messaging Service
    - RNIF – RosettaNet Implementation Framework
    - EDIINT – EDI over INternet

- Ejecución – Conectividad
  - Importante: **no intrusismo** (en la aplicación del sistema de información)
  - Adaptadores
    - Enlazan físicamente la aplicación a la infraestructura de intercambio
    - Deben permitir recuperar los metadatos.
    - Los adaptadores favorecen el no-intrusismo.
    - Por ejemplo, en una aplicación que crea un fichero, el adaptador debería percatarse de esto y generar un evento en la infraestructura de intercambio.
    - Tipos
      - Thin or technical adapters
      - Thick or business adapters

## Thin or technical adapters

Figure 3.10. *Light (technical) adapter*

## Thin or technical adapters

- **Ventajas**

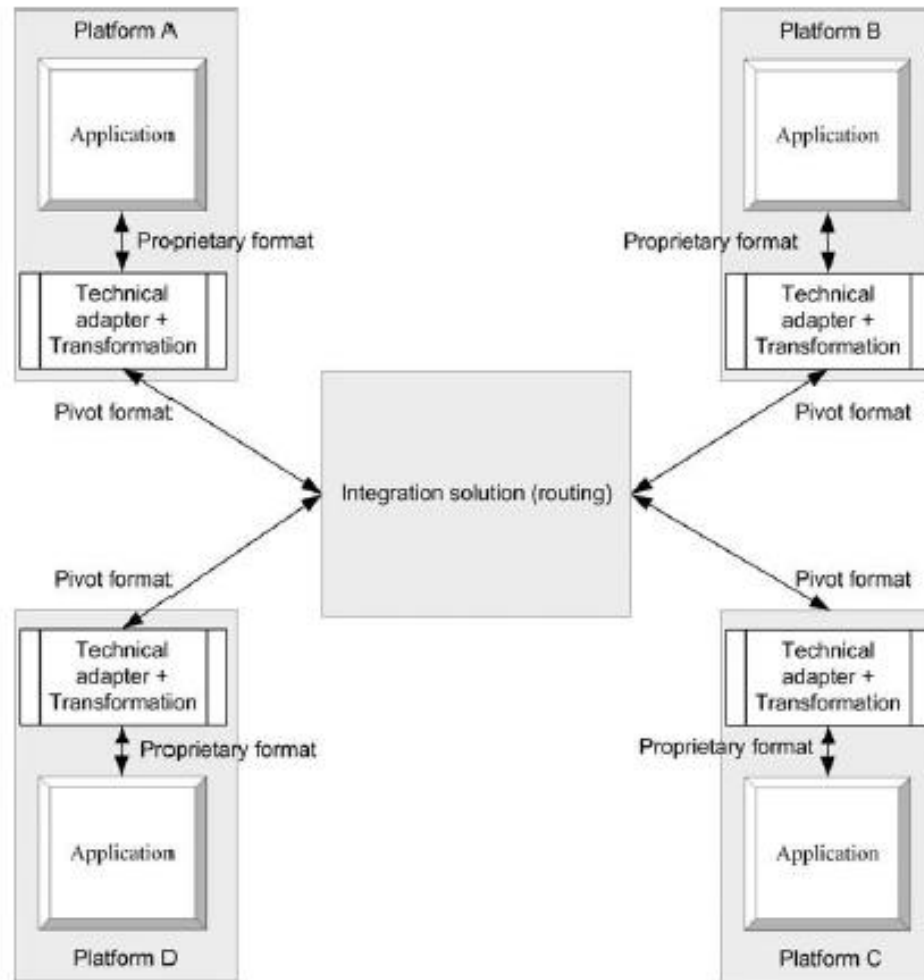
- **Transformación** únicamente en la **plataforma** que **aloja** el núcleo de la solución de integración.
- **Fácil despliegue** , ya que ningún componente especial es necesario para ser instalado en las plataformas que alojan las aplicaciones que se integren .
- Adaptador **independiente** del **formato** de los **datos** y ligado únicamente al tipo de comunicación utilizado por los paquetes de software que se integra.
- Solución totalmente adaptable , sea cual sea el contexto ( **acoplamiento débil** con aplicaciones).
- Capacidad para gestionar reglas de transformación complejos , ya que están **centralizadas** en el **núcleo** de la solución de integración .
- **Precio Asequible**.

### Thin or technical adapters

- **Inconvenientes**

- No se cargan directamente las estructuras definidas en el sistema ERP → **dificultad** en la **entrega** y las **reglas** de transformación que se pueden definir entre los formatos propietarios .
- Sin tratamiento automatizado de los estados para el cómputo de los flujos de datos por el ERP.
- Enlaces punto a punto entre las aplicaciones a través de reglas de transformación (a menudo recreando una **arquitectura de espagueti** en la solución de integración ) .
- La **complejidad** de ciertas **reglas** de transformación .
- **Potencial** de **cuello de botella** , porque todas las transformaciones se realizan en el núcleo de la solución .

## Thick or business adapters



## Thick or business adapters

- **Ventajas**

- El uso sistemático de un **formato canónico** o de pivote para constituir un repositorio de eventos de negocios estandarizados.
- La posibilidad de un juego completo de parámetros entregados con la solución de integración ( **plug-and-play** ) para el uso estándar de determinados paquetes de software , de nuevo a través del uso de un formato canónico .
- La asignación de la carga de trabajo de transformación entre las plataformas que apoyan los adaptadores .

## Thick or business adapters

- **Inconvenientes**
  - Necesidad de **dobles transformaciones**, lo que aumenta la carga de trabajo de procesamiento global y por lo tanto potencialmente afectara el rendimiento solución.
  - No se pueden utilizar Parámetros predefinidos para la conexión de los paquetes de software si la empresa no utiliza formatos estándar , si se los adapta, que es el caso la mayor parte del tiempo.
  - Aumento de **la sensibilidad desde el desarrollo** de paquetes de software, ya que el adaptador está muy estrechamente relacionado con la versión del paquete de software utilizado.
  - Necesidad de implementar los componentes de las plataformas de aplicaciones con frecuentes actualizaciones de los ajustes principales.
  - Que las aplicaciones heredadas en la empresa no tengan en cuenta estos adaptadores.
  - El **alto precio** de estos adaptadores, ya que pueden incorporar avanzadas funciones de transformación.



- **¿Cuál elegir?**
  - Como tantas otras veces, cuando comparamos diferentes posibilidades, no hay ninguno que sea absolutamente mejor que el otro.
  - En la mayoría de los casos, la mejor solución consiste en encontrar un equilibrio entre el uso de thin adapters and thick adapters.

- **Supervisión:**

- Los eventos o los datos transmitidos entre las aplicaciones pueden comprender información crítica para la empresa.
- En todo momento, debe ser posible determinar donde se pueden encontrar.
- Esto es muy importante porque los mecanismos de intercambio son a menudo asíncronos.
- Herramientas de transporte (Los adaptadores), deben de hacer posible el envío de información sobre el progreso de las operaciones a un sistema de supervisión.

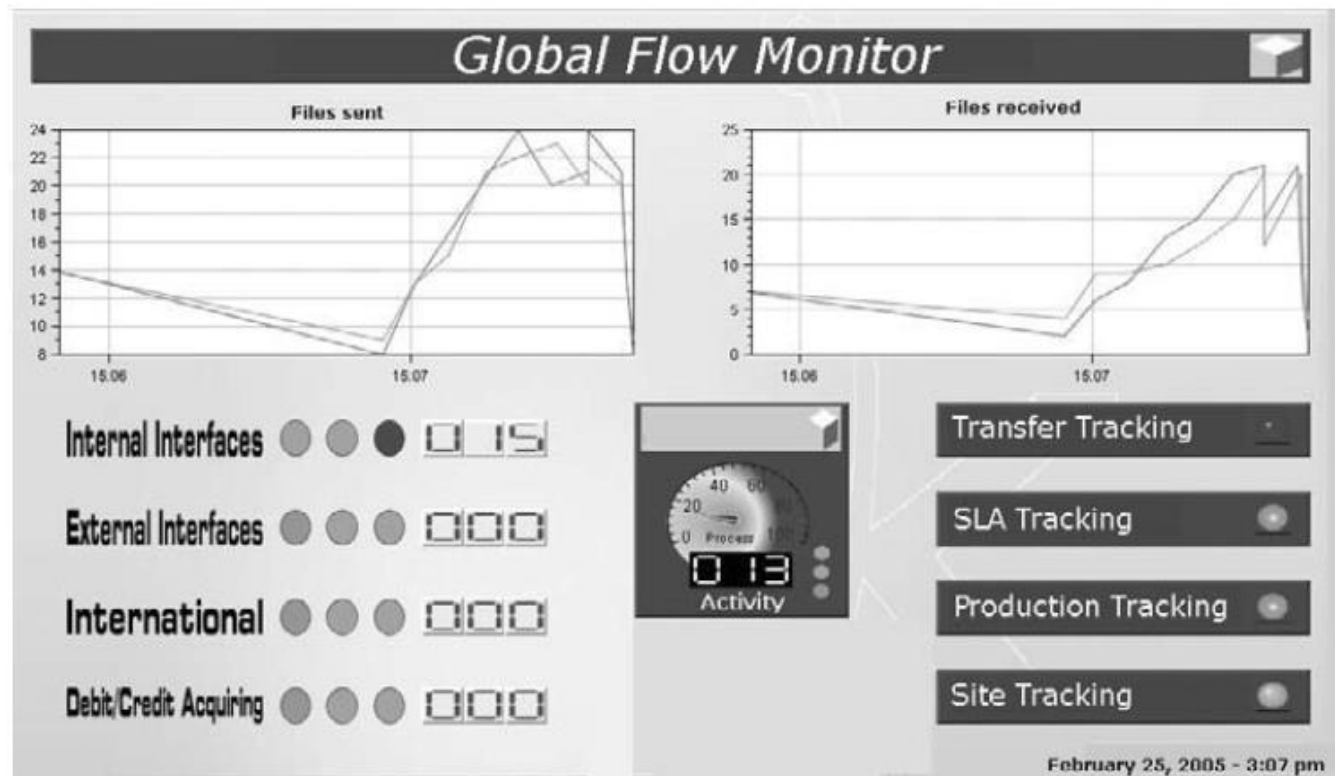
- **Supervisión:**

- Estas herramientas proporcionarán en particular las indicaciones en cuanto se produzca un incidente:
  - Los problemas de conexión entre los gestores de transferencia de archivos , entre las MOM's con la aplicación.
  - Reinicios que podrían llevarse a cabo para transmitir la información.
  - Problemas, tales como : la falta de espacio en disco para la creación de un archivo , la ausencia de una cola o su saturación.
  - Etc.

- **Supervisión**

- Se requiere un sistema de supervisión global que centralizará el conjunto de información de eventos y supervisión .
- Proporcionará una interfaz gráfica donde los operadores pueden consultar la información de seguimiento y realizar intervenciones.

## Supervisión:



Niveles

Nivel de transporte y conectividad

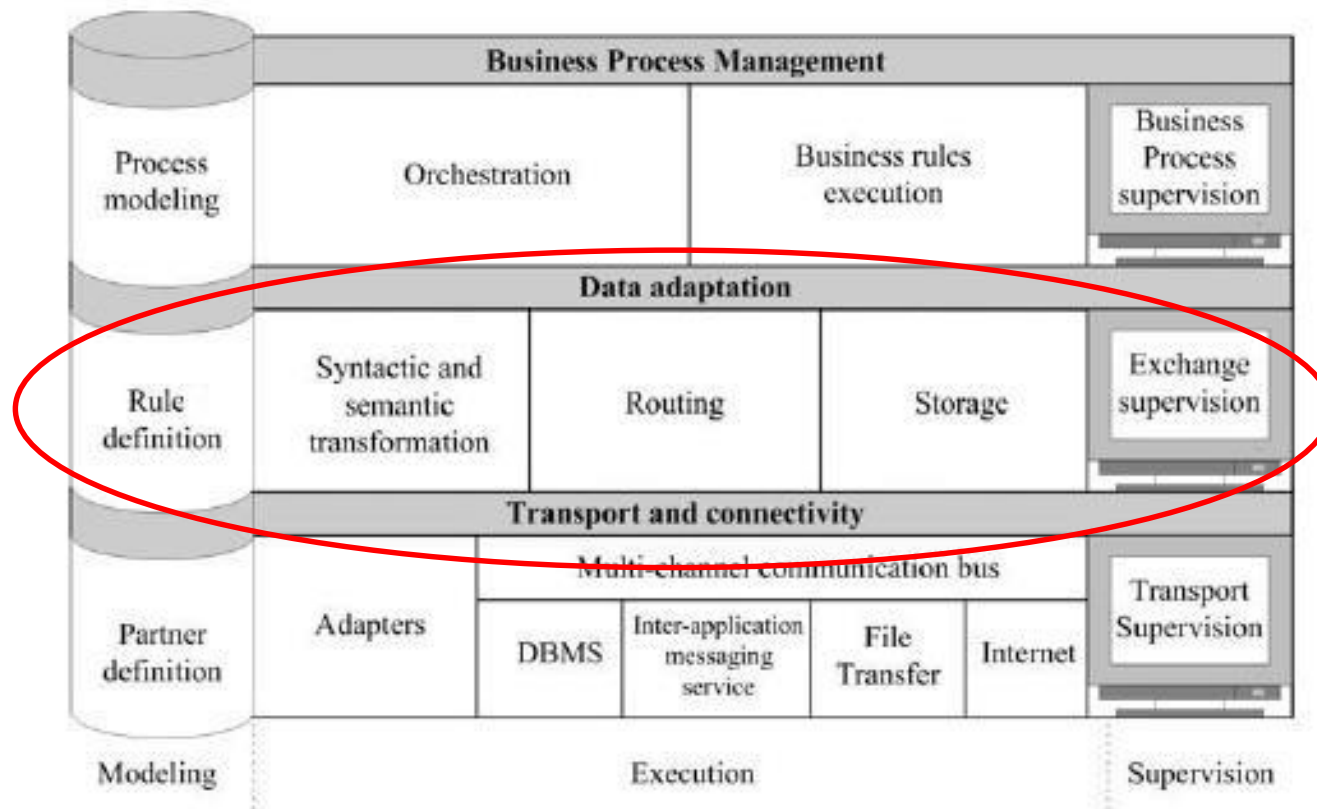


Figure 3.1. Components in the integration infrastructure

- Modelado:
  - La captura, transporte y entrega de información a los destinatarios no es suficiente para hacer frente a todo el conjunto de problemas de integración.
  - También debe ser posible **adaptar los eventos producidos por las aplicaciones originales a lo que esperan las aplicaciones receptoras.**
  - Debe ser posible determinar que los receptores y las aplicaciones cumplen lo que esperan en el momento oportuno.
  - Este es el objetivo del nivel de adaptación de la información.

- Modelado
  - Definición de reglas
    - Transformación
      - Un evento debe ser transformado cuando se expresa en un formato o utilizando una sintaxis que no es directamente comprensible por la aplicación receptora.
    - Enrutamiento
      - Consiste en determinar el destinatario de los eventos o datos generados por una aplicación.
    - Almacenamiento
      - Los diferentes tipos de aplicaciones tienen diferentes ritmos de trabajo.
      - Debe ser posible adaptar el ritmo de explotación de cada aplicación.
      - Para ello, la infraestructura de integración debe proporcionar mecanismos para el almacenamiento temporal de estos eventos con el fin de ponerlos a disposición de las aplicaciones receptoras.



- Modelado:
  - La adaptación de los eventos a las operaciones de las aplicaciones requiere de la definición de reglas para la transformación, almacenamiento y enrutamiento.
  - Estas reglas deben estar centralizadas en un diccionario global
  - Este diccionario incluirá en particular:
    - Definición de eventos en el sistema
    - Identificar reglas para dichos eventos
    - Reglas de transformación a aplicar
    - Reglas de enrutamiento
    - Etc.

- Ejecución
  - Transformación
    - Reglas de transformación de datos
    - Transformación sintáctica
      - Objetivo: modificar la representación de un evento para hacerlo usable por la aplicación responsable de procesarlo
        - » XML → flat format (para COBOL)
    - Transformación semántica
      - Objetivo: modificar el significado de todo o parte de la información de un evento para deducir otros eventos que serán utilizados por el destino
        - » XML con datos bancarios → mensaje en texto para un cliente

```

XML Schema
<elementType id= "Book" >
<element type= "#title" occurs= "OPTIONAL" />
<element type= "#author" occurs= "ONEORMORE" />
</elementType>

XML Document generated
<Book>
<author>Henry Ford</author>
<author>Samuel Crowther</author>
<title>My Life and Work</title>
</Book>

```

Figure 3.17. Example XML Schema

```

XML Order

<Order>
<CurrencyCode>EUR</CurrencyCode>
<PaymentMode>transfer</PaymentMode>
<OrderDate>20000310</OrderDate>
<CustomerReference>
  <Name>Mr BEAN</Name>
  <CustomerCode>007</CustomerCode>
  <Company>Editions HERMES</Company>
  <Address>
    <Address1>Editions HERMES</Address1>
    <Address2>8, quai du Marché-Neuf</Address2>
    <PostalCode>75004</PostalCode>
    <City>Paris</City>
    <Country>FRANCE</Country>
  </Address>
</CustomerReference>
  <OrderLine>
    <Quantity>25</Quantity>
    <ProductDescription>PC, Pentium III 500 MHz,
      8 processors, RAM 256 Mb</ProductDescription>
  </OrderLine>
</Order>

Order for application written in COBOL

.....10.....20.....30.....40.....50.....60
EURtransfer 20000310Mr BEAN                      007HERMES
.....70.....80.....90.....100.....110.....120
Editions HERMES      8, quai du Marché-Neuf 75004Paris
.....130.....140.....150.....160.....170.....180.....190
France      25PC, Pentium III 500 MHz,8 processors,
.....200
RAM 256 Mb

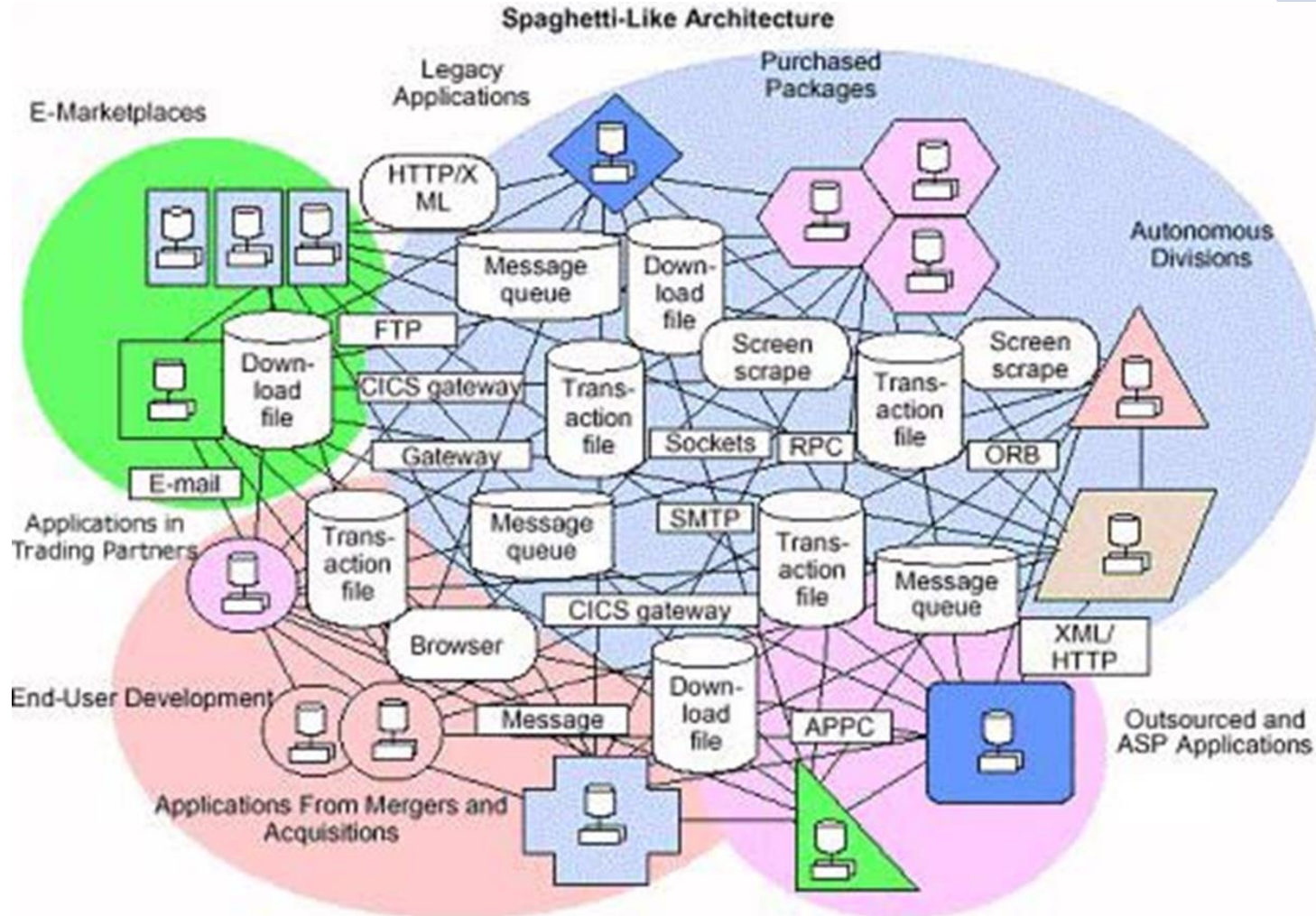
```

Figure 3.18. Syntactical transformation of an order

- Ejecución

- Enrutamiento

- Determinar el destinatario de los eventos o datos generados por una aplicación:
      - Spaghetti system. La aplicación conoce los destinatarios → direccionamiento directo.
      - Mecanismo de publicación/suscripción.



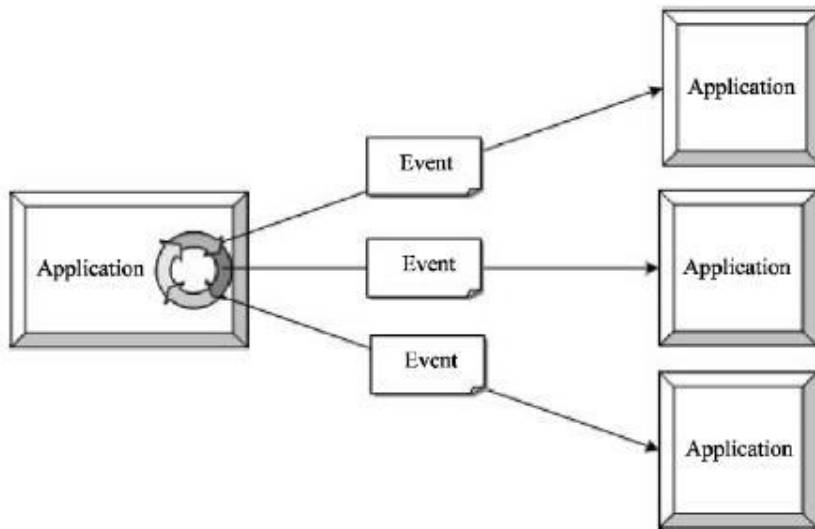


Figure 3.21. Routing by application

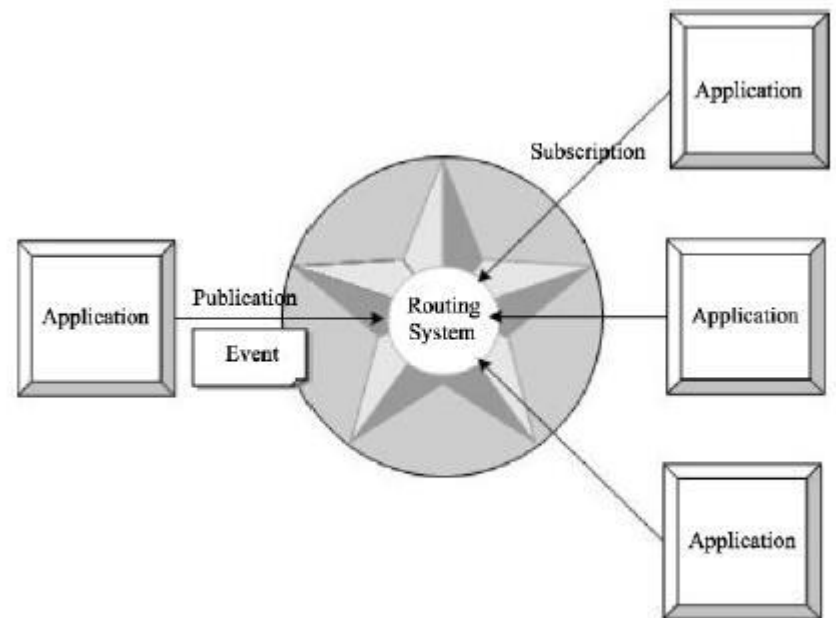


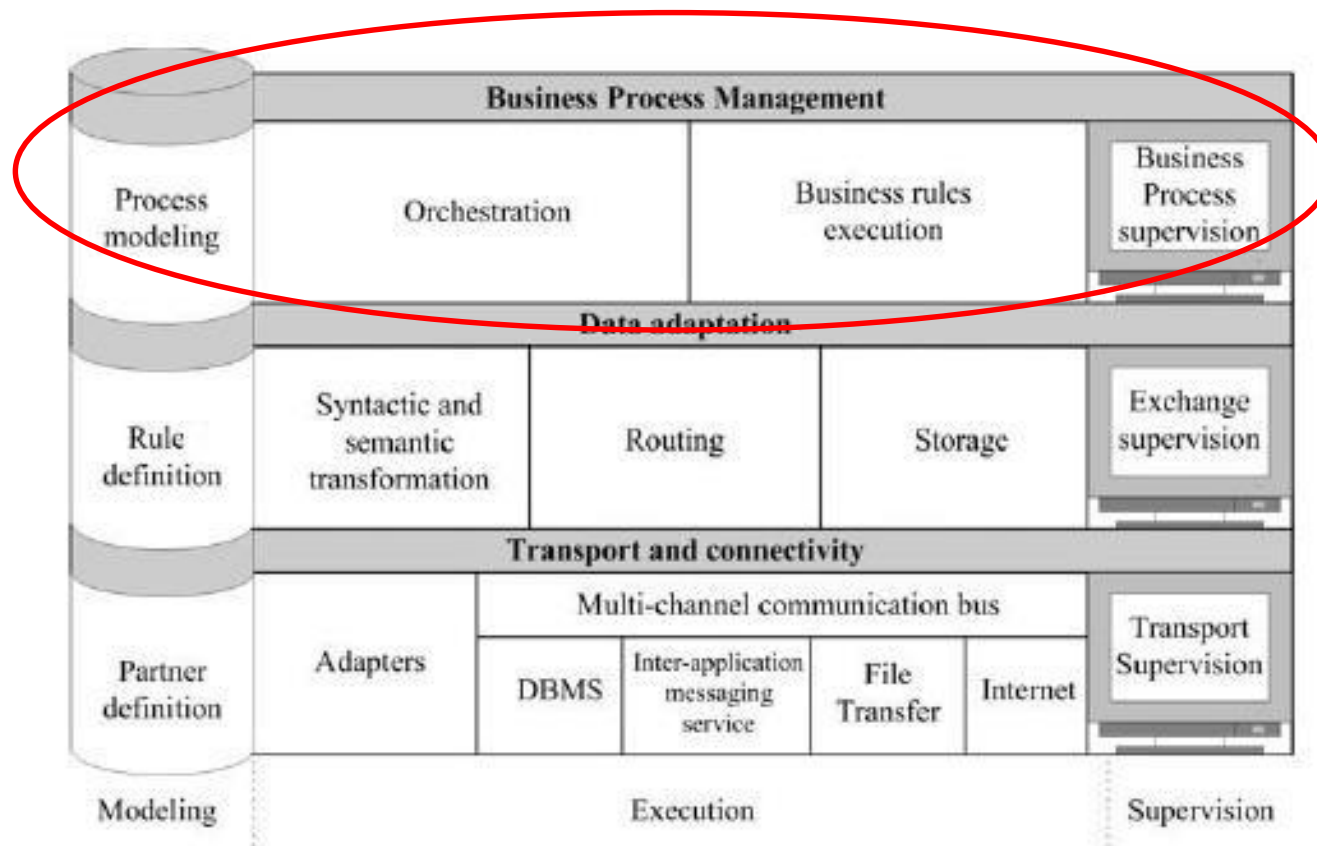
Figure 3.22. Mechanism of publication/subscription

- Ejecución
  - Almacenamiento
    - Dependiendo de cómo se soporte un evento, este se llevará a cabo en el ambiente adecuado:
      - Sistemas de ficheros.
      - Bases de datos.
      - Colas.



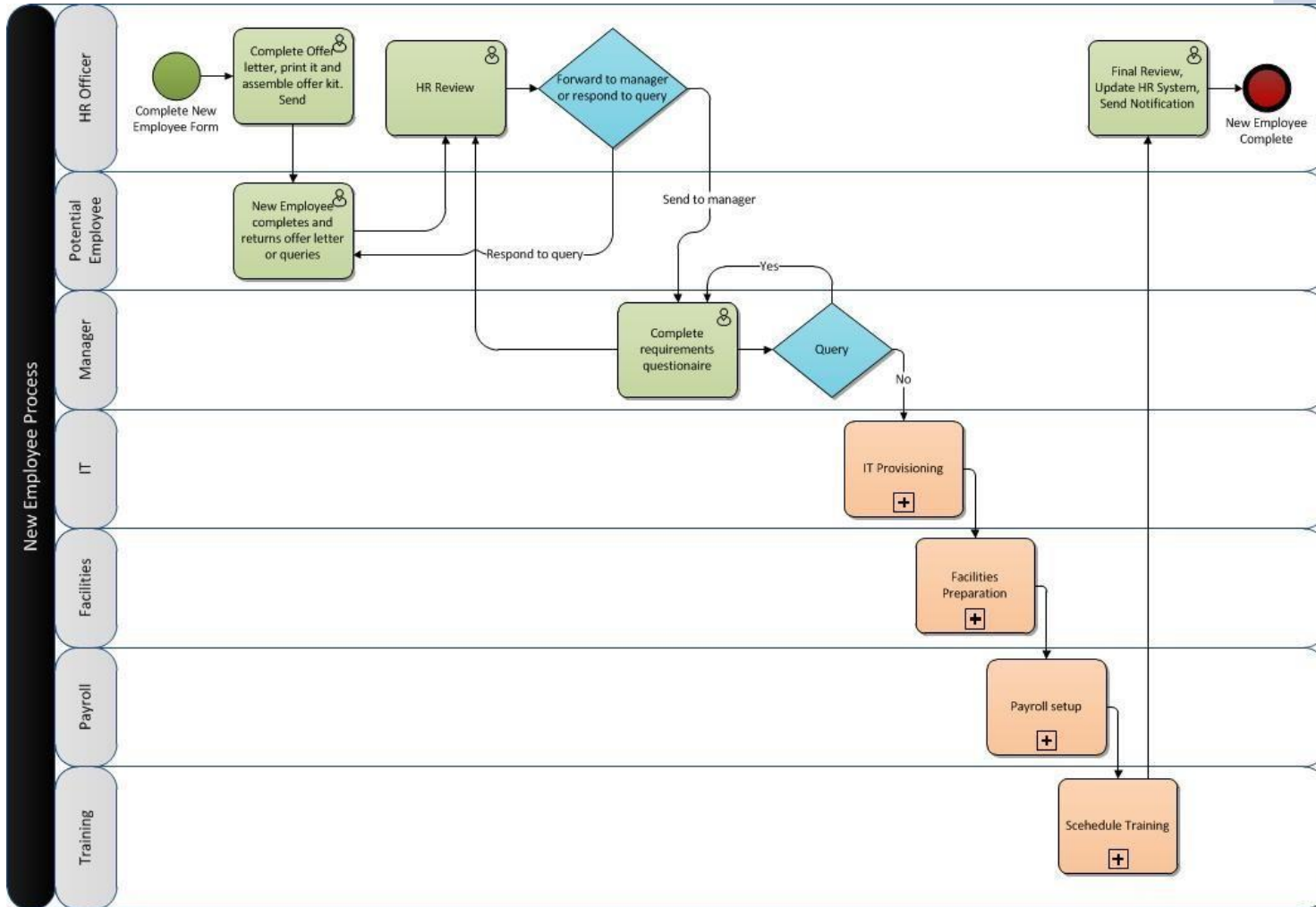
- Supervisión
  - Debe ser posible realizar el seguimiento del proceso de integración de extremo a extremo
  - Se debe mostrar el estado de un evento en el proceso de intercambio entre aplicaciones en cualquier punto.
  - Posibilidad de ampliar pasos complejos
  - Alertas
    - Directas ante errores
    - Derivadas de reglas de chequeo (p.e. “en orden adecuado”).

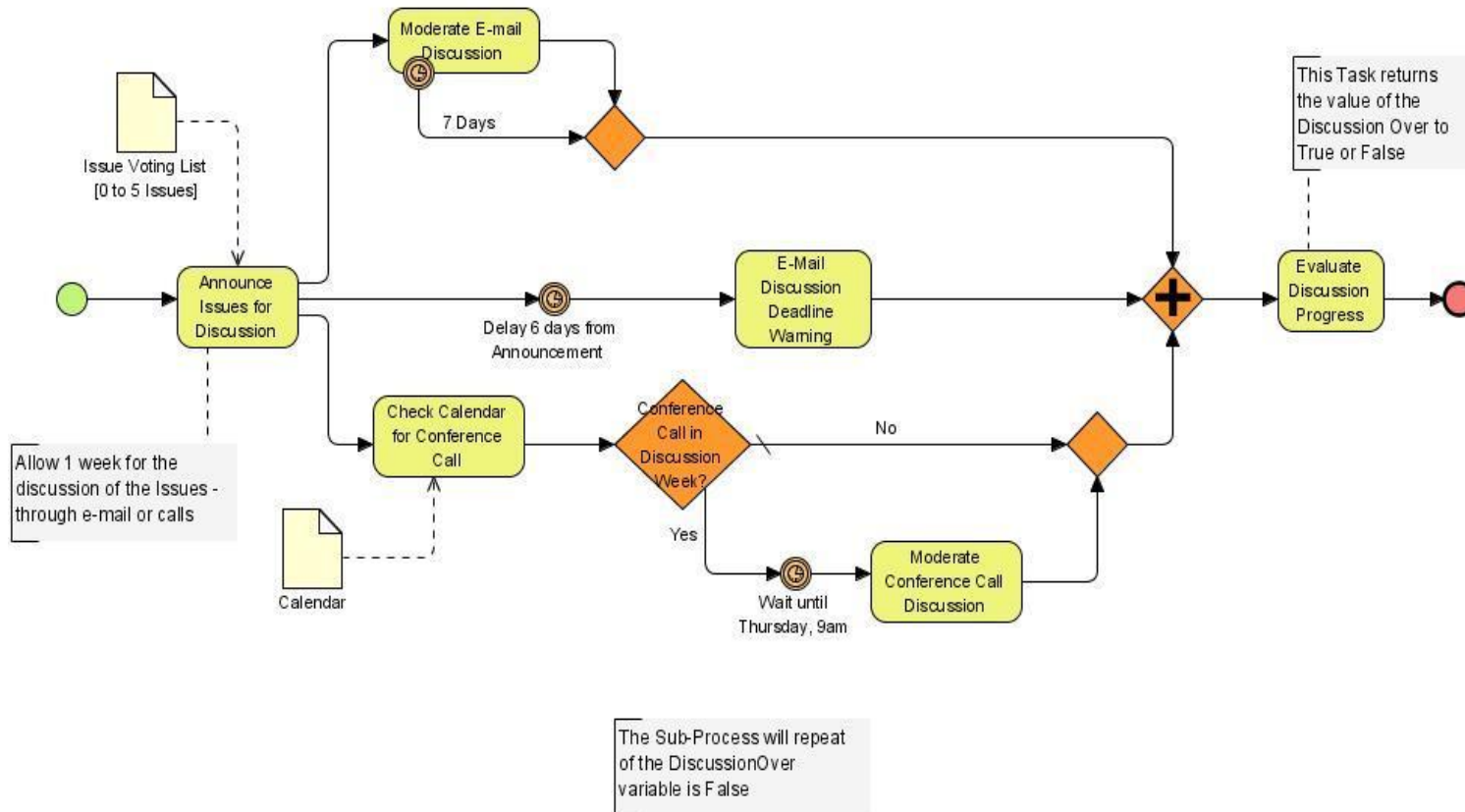




**Figure 3.1.** Components in the integration infrastructure

- Modelado
  - Representa el proceso para que pueda ser implementado automáticamente
    - Actividades, relaciones entre ellas, criterios de inicio y fin, participantes, datos, etc.
  - BPMN – Business Process Management Notation
    - Conexiones
    - Acciones
    - Grupos de elementos
    - Artefactos (información adicional)





- Ejecución
  - Organización de actividades
    - Ejecución de las secuencias modeladas (lanzar actividades en orden, aplicar reglas, enviar información al sistema de supervisión, etc.)
    - Motor de ejecución de procesos
    - Lenguaje de ejecución de procesos: BPEL (standard OASIS)
  - Ejecución de reglas de negocio
    - Business Rules Engine (BRE)
    - “¿Qué se hace ante una falta de stock?”

- Supervisión
  - Asegurar la correcta secuencia de las aplicaciones o tareas que forman el proceso
    - BAM proporciona indicadores en tiempo real para evaluar la eficiencia de los procesos
    - Alertas en caso de actividad anormal

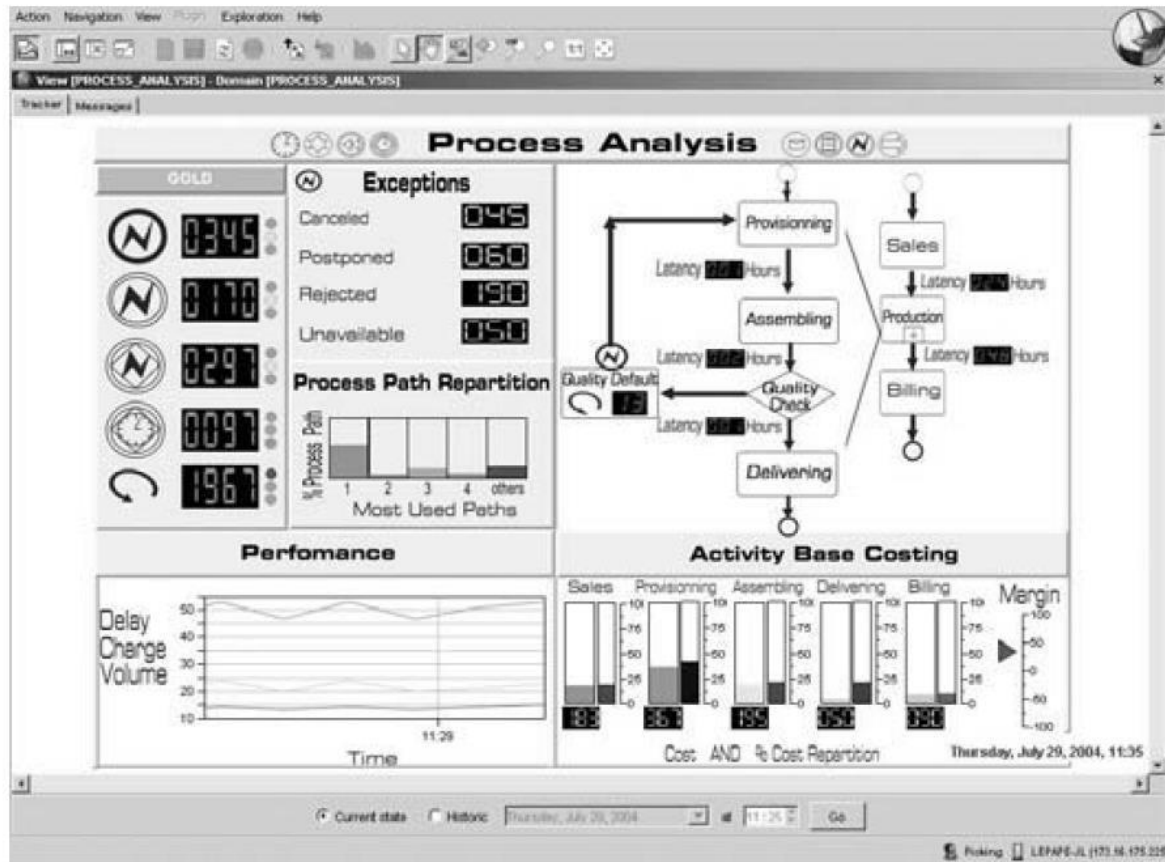


Figure 3.42. Supervision of the processes (© Axway, 2006)

# Fundamentos de la integración de sistemas software



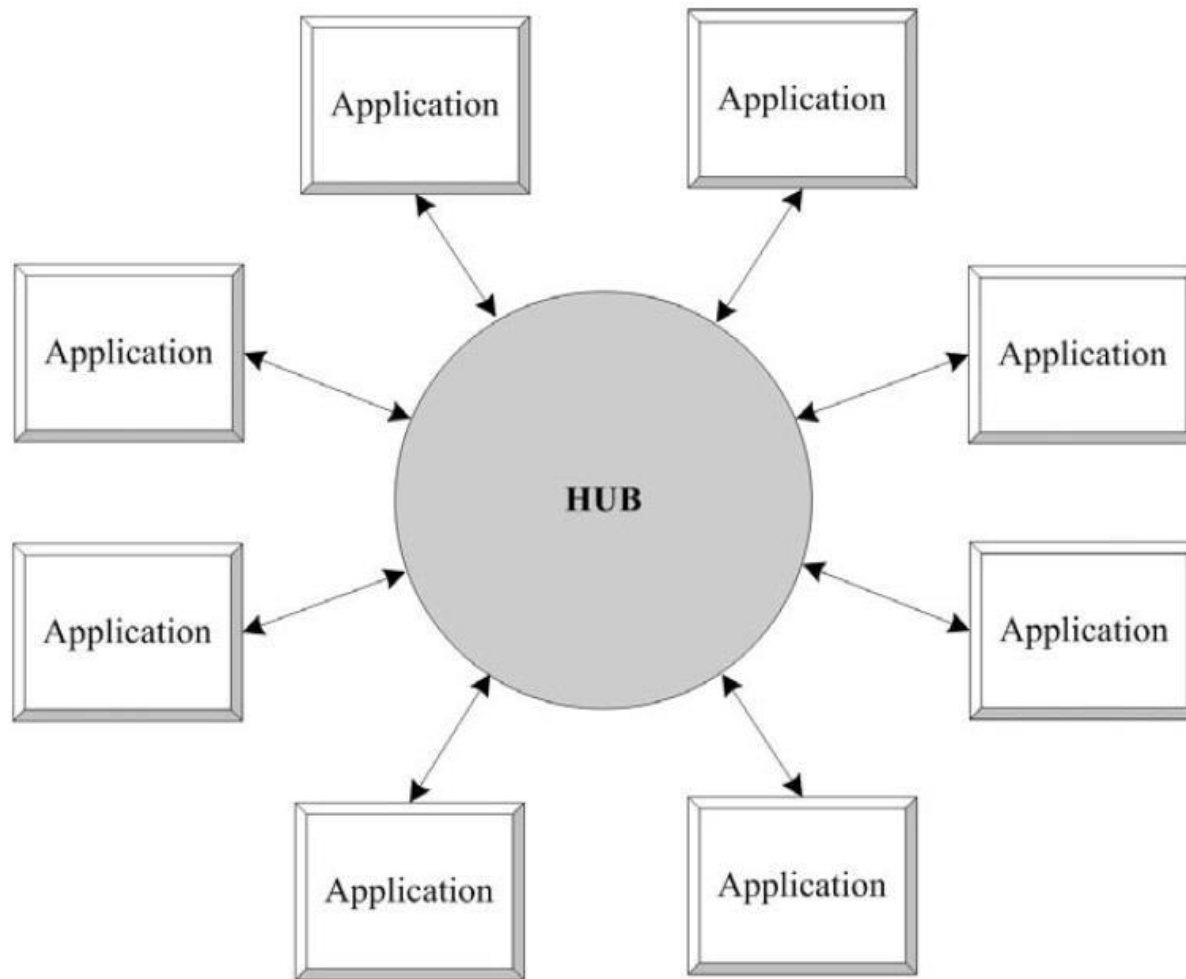
Niveles

Nivel de procesos de negocio



- Otras consideraciones
  - Comunicación síncrona/asíncrona
    - El resultado de la integración entre aplicaciones no debe volver a crear un sistema espagueti dentro de la solución de integración.
    - El **acoplamiento** debe ser **débil** entre estas aplicaciones.
    - Integración mediante propagación de datos o interacciones unidireccionales → asíncrona (**MOMs**)
    - Aplicaciones compuestas → síncrona (RMI, **Web services**)

- Otras consideraciones
  - Arquitectura centralizada/distribuida
    - Una vez que el tipo de comunicación se ha determinado, la cuestión ahora que hay que preguntarse :
      - Que arquitectura de intercambio deseamos, esto influye directamente en la elección de la herramientas a utilizar.
    - Centralizada (SPOF)
    - Distribuida (snowflake). Grandes empresas con infraestructuras a gran escala



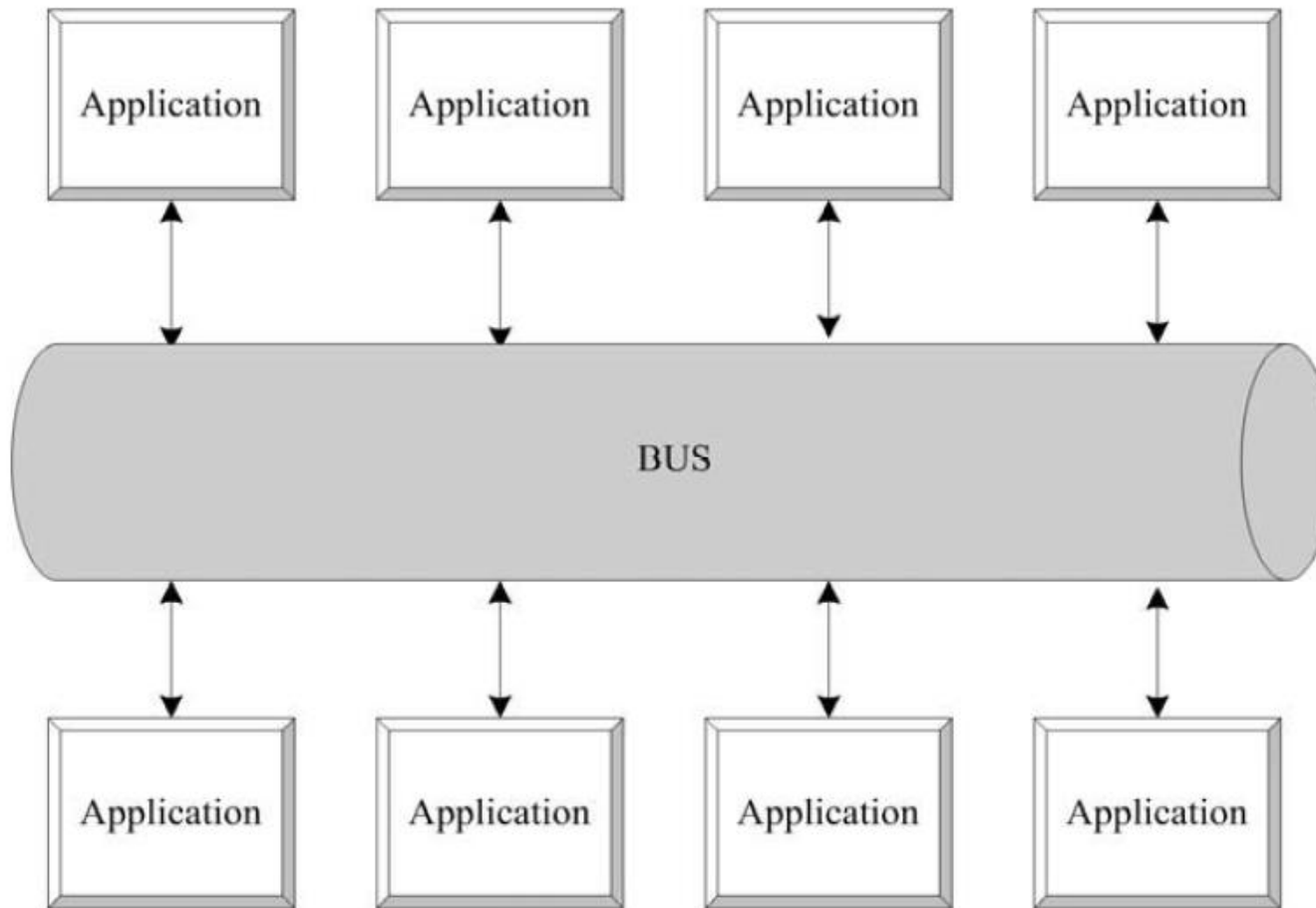
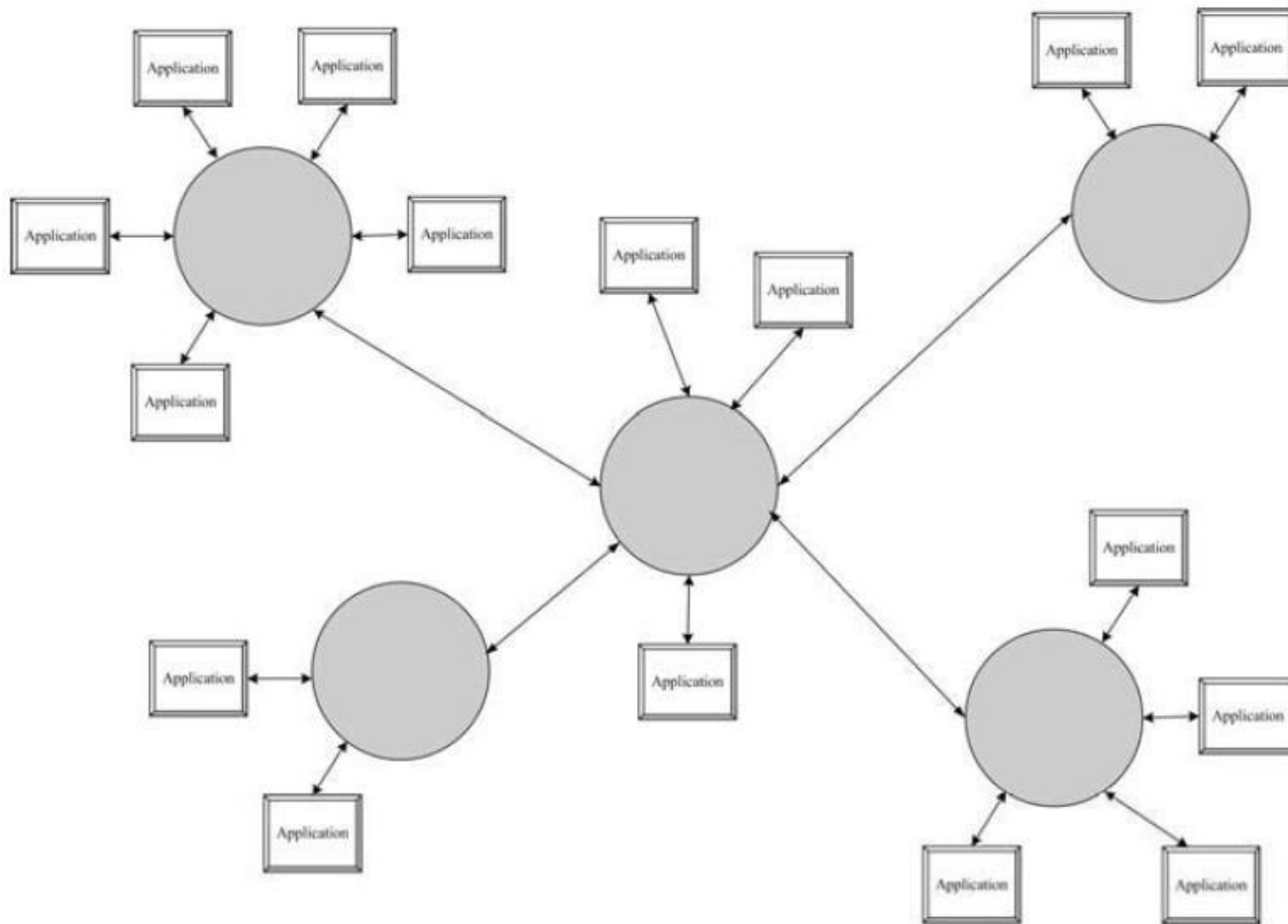


Figure 3.50. *Bus architecture*



- XML vs JSON:
  - XML y JSON son los dos formatos más comunes para el intercambio de datos en la Web hoy en día.
    - XML fue creado por el W3C en 1996.
    - JSON fue especificado públicamente por Douglas Crockford en 2002.
    - Aunque sus propósitos no son idénticos, con frecuencia se usan para realizar la misma tarea, que es el intercambio de datos.
    - Ambos tienen estándares abiertos bien documentados en la Web ( [RFC 7159](#) , [RFC 4825](#) )
    - Ambos son legibles por humanos y por máquinas.
    - Ninguno de ellos es absolutamente superior al otro, ya que cada uno es más adecuado para diferentes casos de uso.

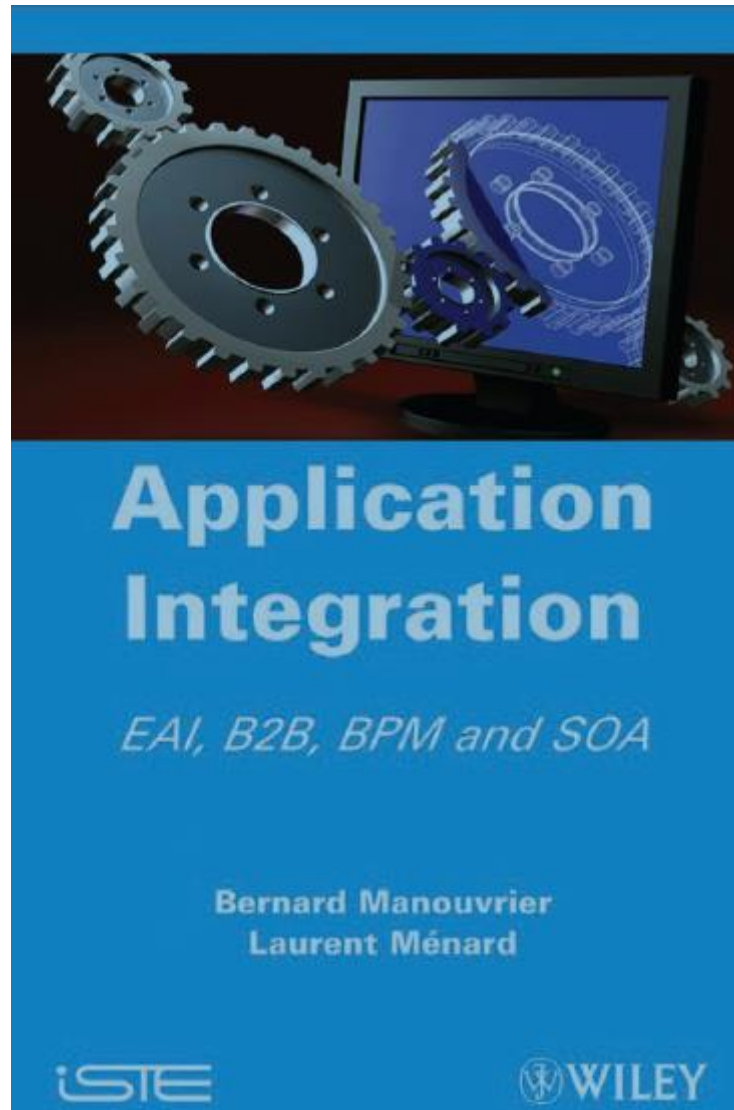
- XML vs JSON:
  - Ventajas XML
    - XML puede colocar metadatos en las etiquetas en forma de atributos.
    - Una de las ventajas más importantes que tiene XML sobre JSON es su capacidad para comunicar contenido mixto, es decir, cadenas que contienen marcas estructuradas.
    - Manejo por parte de los navegadores.

- XML vs JSON:
  - Ventajas JSON
    - JSON es significativamente menos detallado que XML.
    - Los archivos JSON que contienen la misma información que sus homólogos XML son más pequeños → transmisión y procesamiento más rápidos.
    - JSON se serializa y deserializa más rápido que XML.
    - JSON usa menos recursos totales que XML durante el procesamiento.



- XML vs JSON:
  - El propósito de XML es el marcado de documentos.
  - El propósito de JSON es el intercambio de datos estructurados.
  - Futuro IoT: ¿JSON o XML?
    - [https://www.cs.tufts.edu/comp/150IDS/final\\_papers/tstras01.1/FinalReport/FinalReport.html](https://www.cs.tufts.edu/comp/150IDS/final_papers/tstras01.1/FinalReport/FinalReport.html)

- Adaptadores de negocio
  - ¡Cuanto más adaptadores, más sencilla la integración! ¿?
    - Adaptadores muy específicos
    - Versiones
- Ficheros: los “primos pobres” de la integración
  - Diversos problemas
    - En caso de error se bloquea todo o parte del fichero?
    - Volumen innecesario?
- Procesos y servicios lo son todo
  - BPM and SOA: top-down approach
    - From business to IT
  - EAI and B2B: bottom-up approach
    - From IT to business



# Fundamentos de la integración de sistemas software