



# Integración de aplicaciones con SOA

## Sesión 1: Integración de aplicaciones



# Puntos a tratar

- Introducción
- EAI y las aplicaciones existentes
- Retos que plantea la EAI
- Capas en la integración de aplicaciones
- Tecnologías *middleware* para EAI
- Servicios de infraestructura de integración
- Estrategias de integración
- Arquitectura de integración



# Introducción

- Necesidad creciente de las empresas de un fácil acceso a la información por parte de las aplicaciones
- Objetivo: **maximizar** los beneficios de cada aplicación y del sistema de información (SI) como un todo
- Alternativas:
  - Reemplazar el SI por uno nuevo
  - Introducir soluciones comerciales
  - **Integración de las aplicaciones existentes**



# Definición EAI

- EAI: *Enterprise Application Integration*
- Desde el punto de vista del negocio:
  - Ventaja competitiva que supone la integración de las aplicaciones en un proceso de negocio
- Desde el punto de vista técnico:
  - Integración de aplicaciones y datos, sin modificar demasiado los sistemas existentes, en poco tiempo, y con bajo coste
- EAI proporciona una metodología estándar para la comunicación entre aplicaciones y fuentes de datos



# Sistemas de Información de Empresas

- EIS: *Enterprise Information System*
  - Procesos de negocio +
  - Infraestructura para tecnologías de la información (IT)
- Un EIS proporciona la estructura de información para la empresa
- Una empresa puede crear varias EIS según sus necesidades
- Cuando una empresa opera durante mucho tiempo, puede estar usando EIS desarrollados sobre diferentes plataformas y arquitecturas



# Arquitectura de aplicaciones existentes

- Sistemas *legacy*: basadas en tecnologías en desuso
  - Sistemas monolíticos (*mainframe*)
  - Sistemas cliente/servidor
- Década de los 70
  - Integración de sistemas monolíticos mediante una BD centralizada
- Década de los 80
  - Arquitectura de dos capas cliente/servidor
  - Clientes *fat* y *thin*
  - Permanecen las soluciones monolíticas



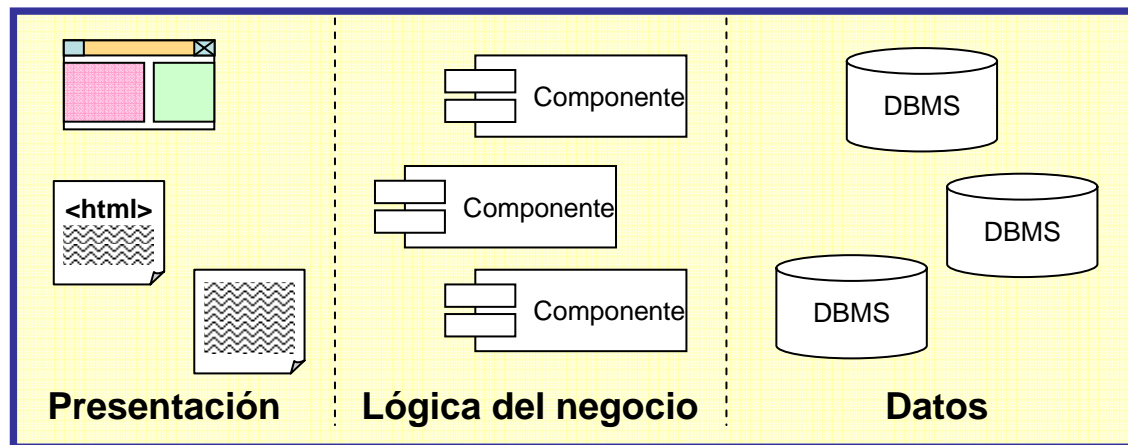
# Problema de análisis de datos

- Década de los 90
  - Solapes funcionales entre las aplicaciones
  - *Datawarehouse*
    - Análisis de grandes cantidades de datos de aplicaciones diferentes
    - Comparte funcionalidades entre las aplicaciones
  - *Enterprise Resource Planning* (ERP)
- Situación actual
  - Mezcla disparatada de sistemas existentes de los cuales dependen los negocios de las empresas
  - ¿Cómo cambiar esta situación sin comprometer las operaciones?



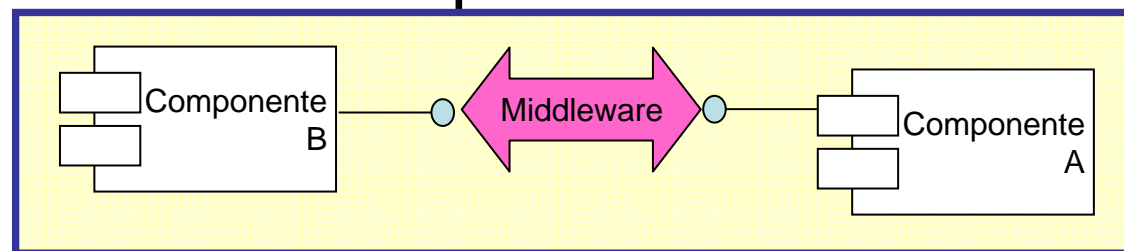
# Arquitecturas modernas

- Arquitecturas multi-capa



Ejemplo  
arquitectura 3  
capas

- Componente**: empaqueta código ejecutable
- Middleware**: facilita la interoperabilidad entre componentes







## Retos que plantea la EAI

- Nivel de soporte tecnológico de los EIS
- Restricciones administrativas y tecnológicas
- Capacidad de integración de los EIS
- Grado de exposición de detalles de bajo nivel
- Transacciones y seguridad
- Escalabilidad



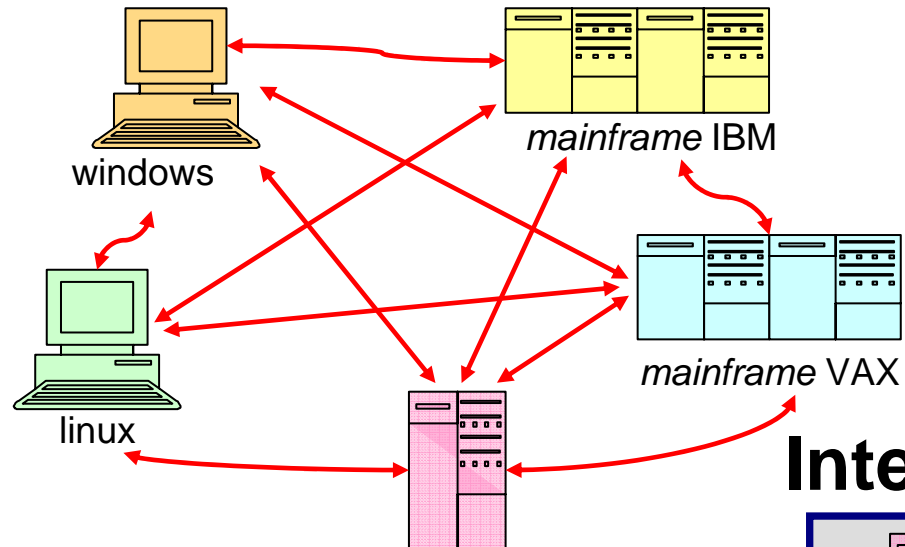
# Capas en la integración de aplicaciones

- Integración de la plataforma
- Integración a nivel de datos
- Integración de la interfaz de la aplicación
- Integración de métodos del negocio
- Integración de la presentación
- Integración inter-EAI o B2B

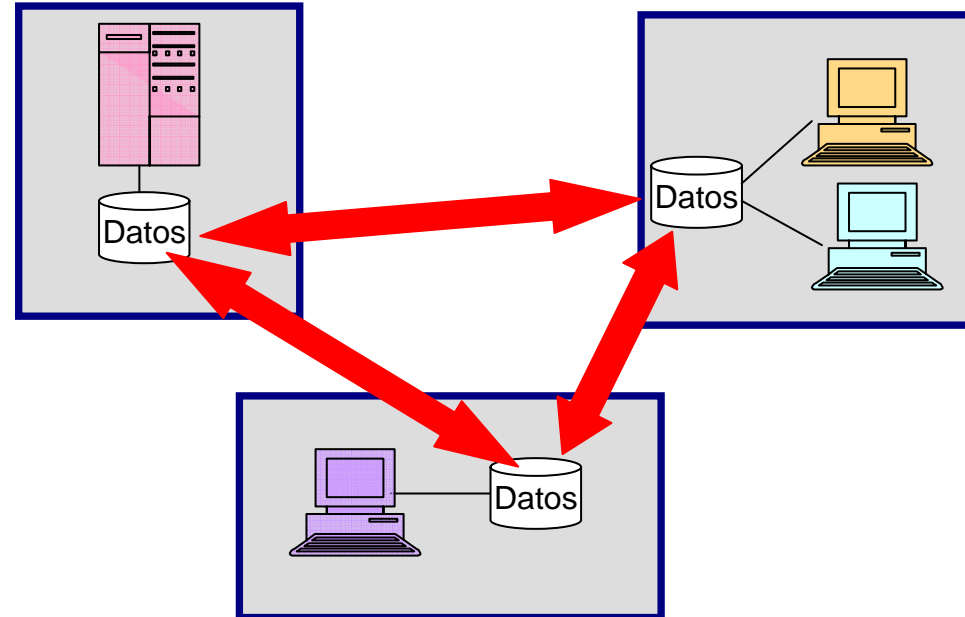
Omitir alguna capa acelera el proceso de integración a corto plazo, pero "pagaremos por ello" a largo plazo



# Integración de la plataforma

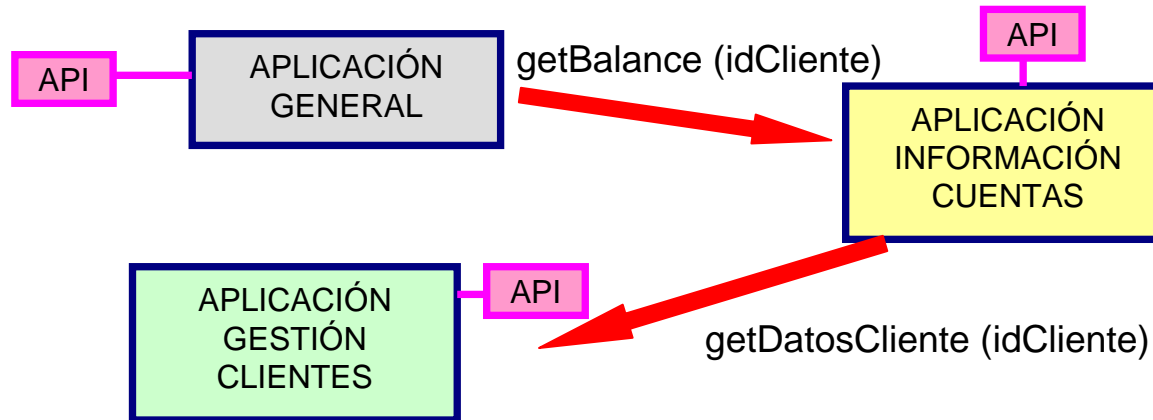


# Integración a nivel de datos

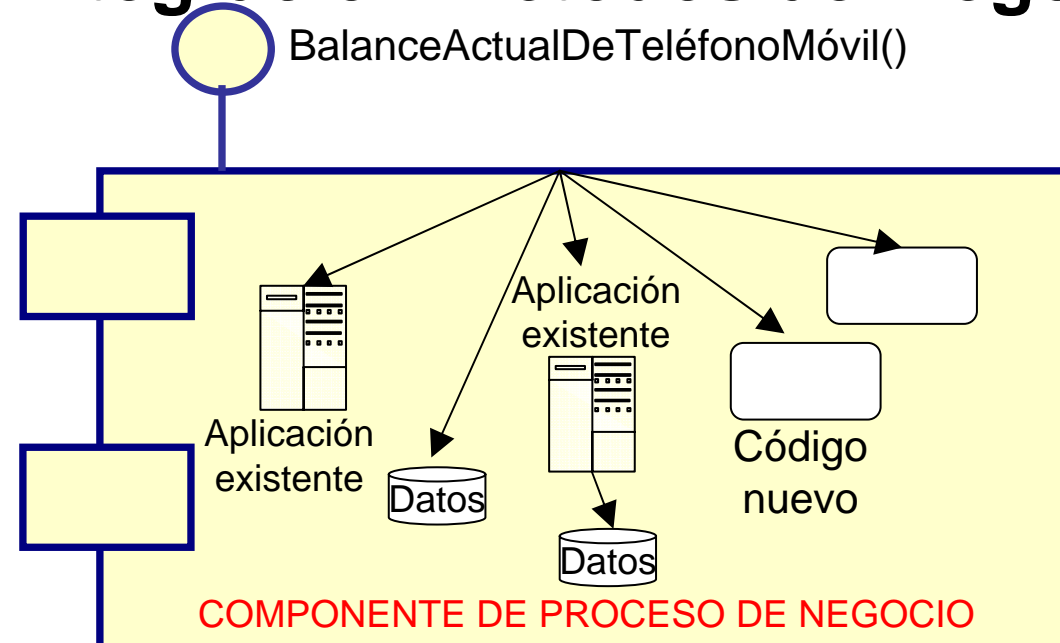




# Integración interfaz de la aplicación

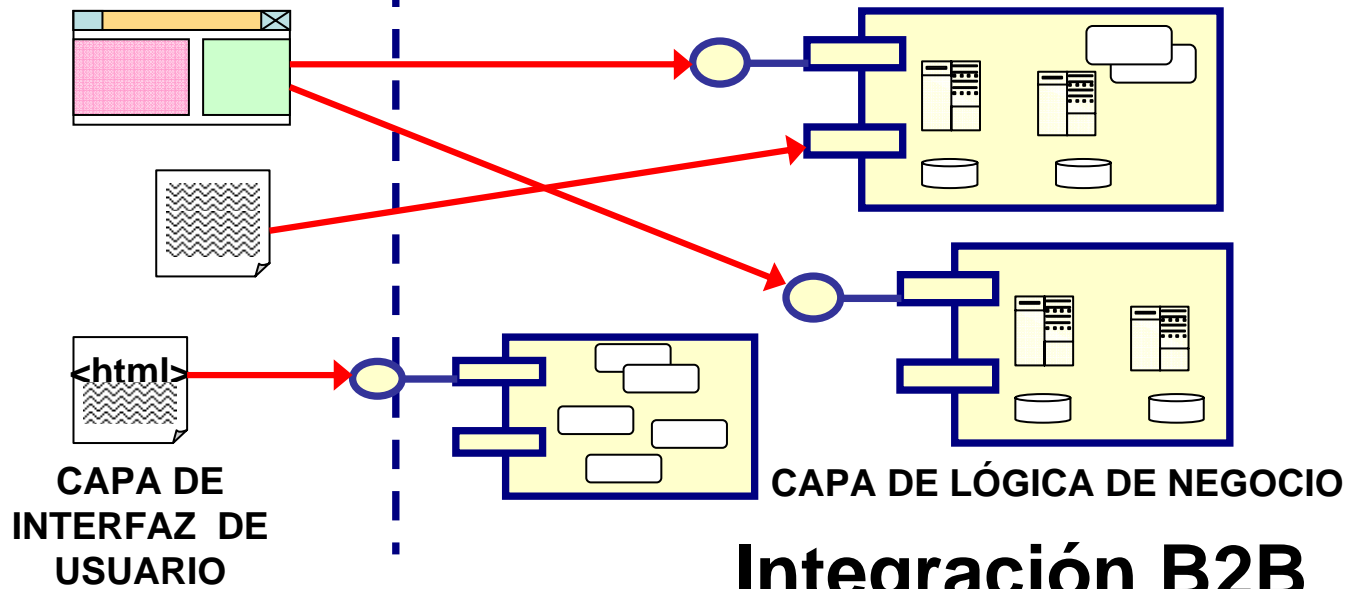


# Integración métodos del negocio

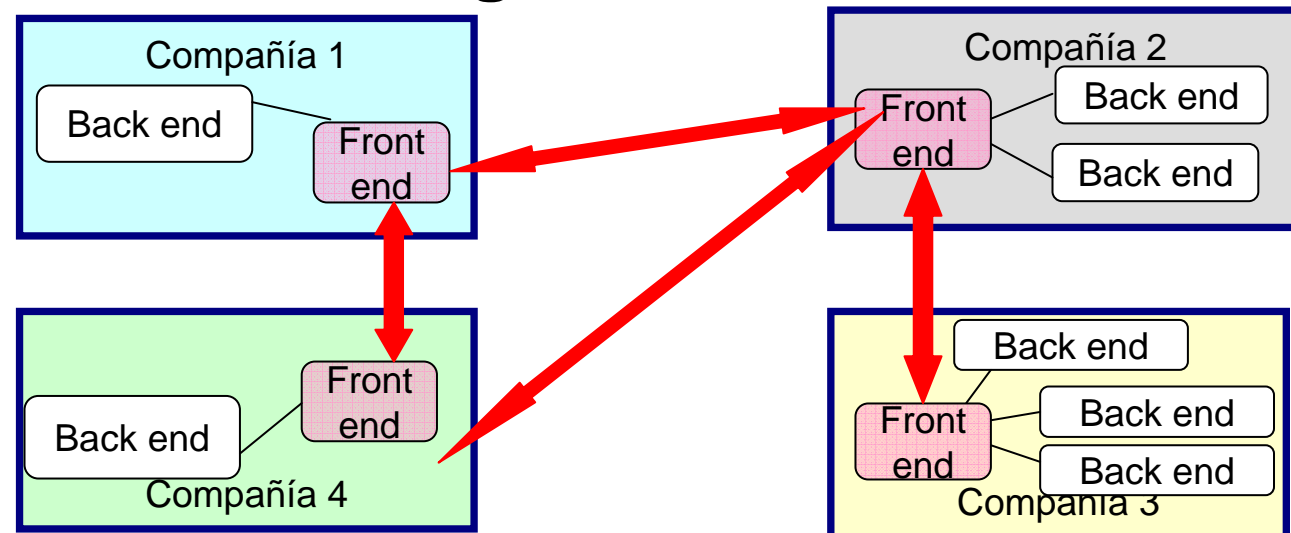




# Integración de la presentación



# Integración B2B





# Técnicas *middleware* para EAI

- Software de los servicios del sistema que se ejecuta entre la capa del sistema operativo y la capa de aplicación
- Facilitan la comunicación entre diferentes aplicaciones software
- Introducen un nivel de abstracción en la arquitectura del sistema
- Introducen una cierta sobrecarga en el sistema con respecto a la comunicación

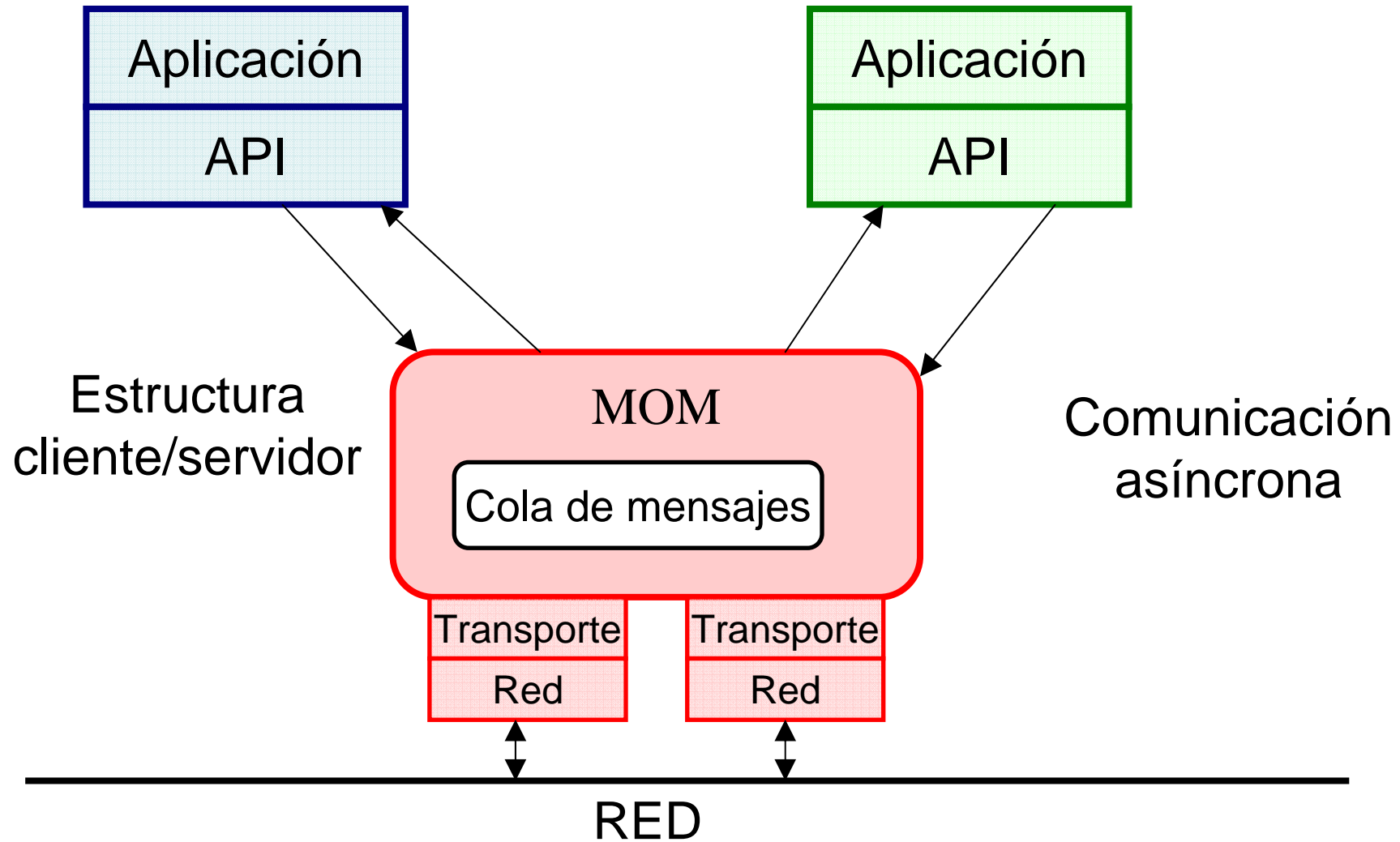


## Productos *middleware*

- Tecnologías de acceso a BD
  - JDBC, JDO, ODBC, ADO
- *Middleware* orientado a mensajes (MOM)
- Llamadas a procedimientos remotos (RPC)
- Monitores de procesamiento de transacciones (TP)
- Intermediarios de peticiones de objetos (ORB)
- Servidores de aplicaciones



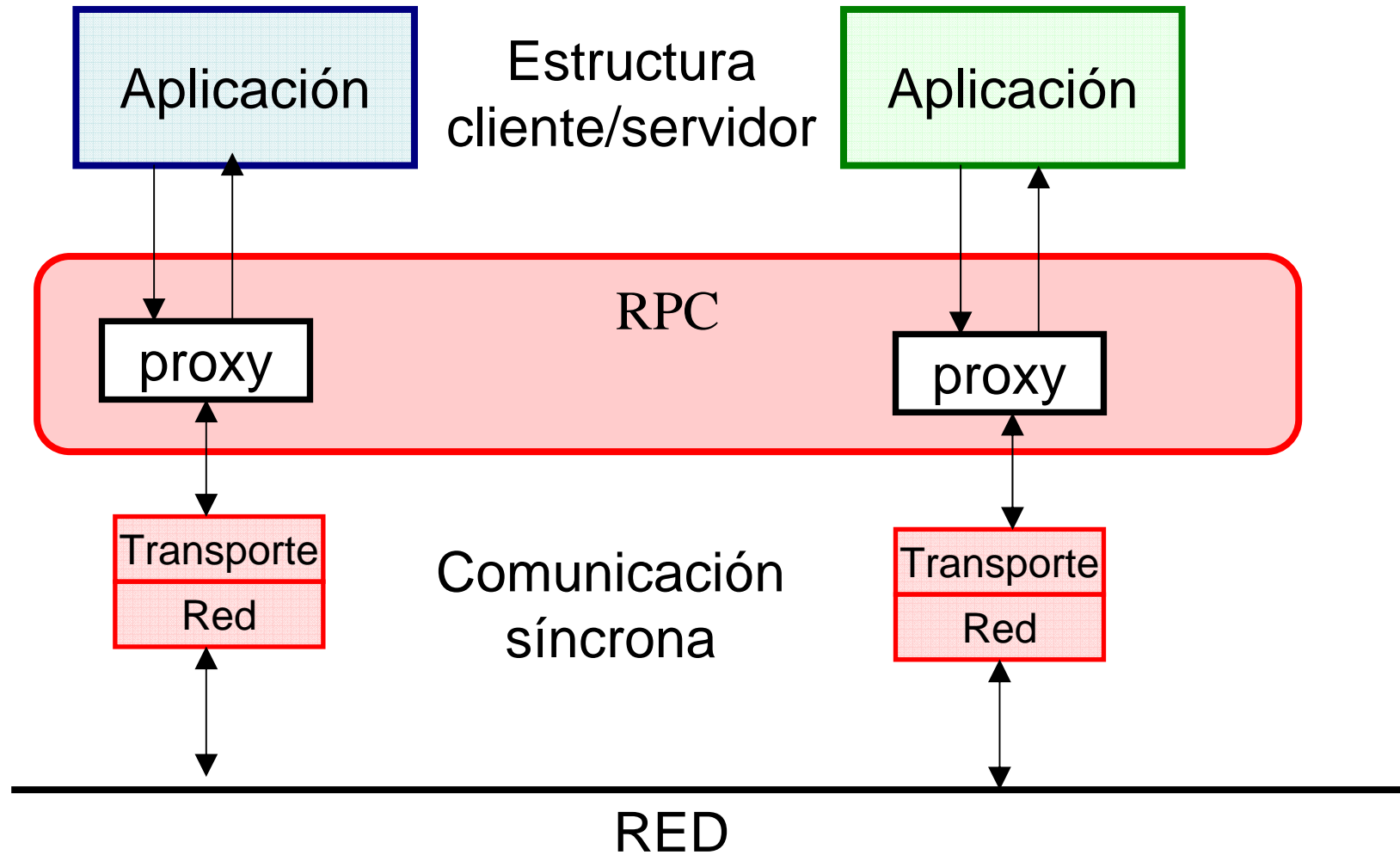
# MOM





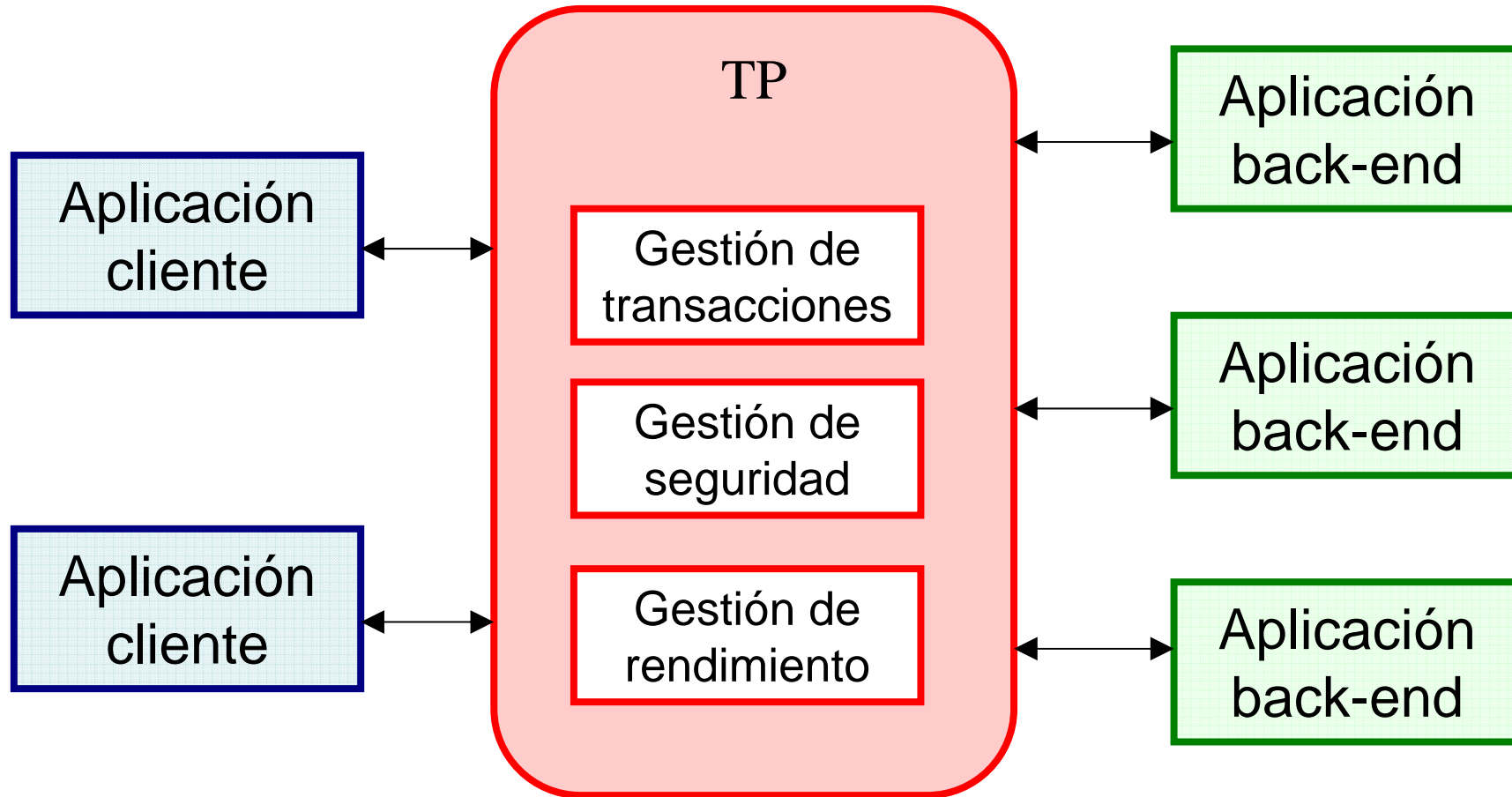


# RPC



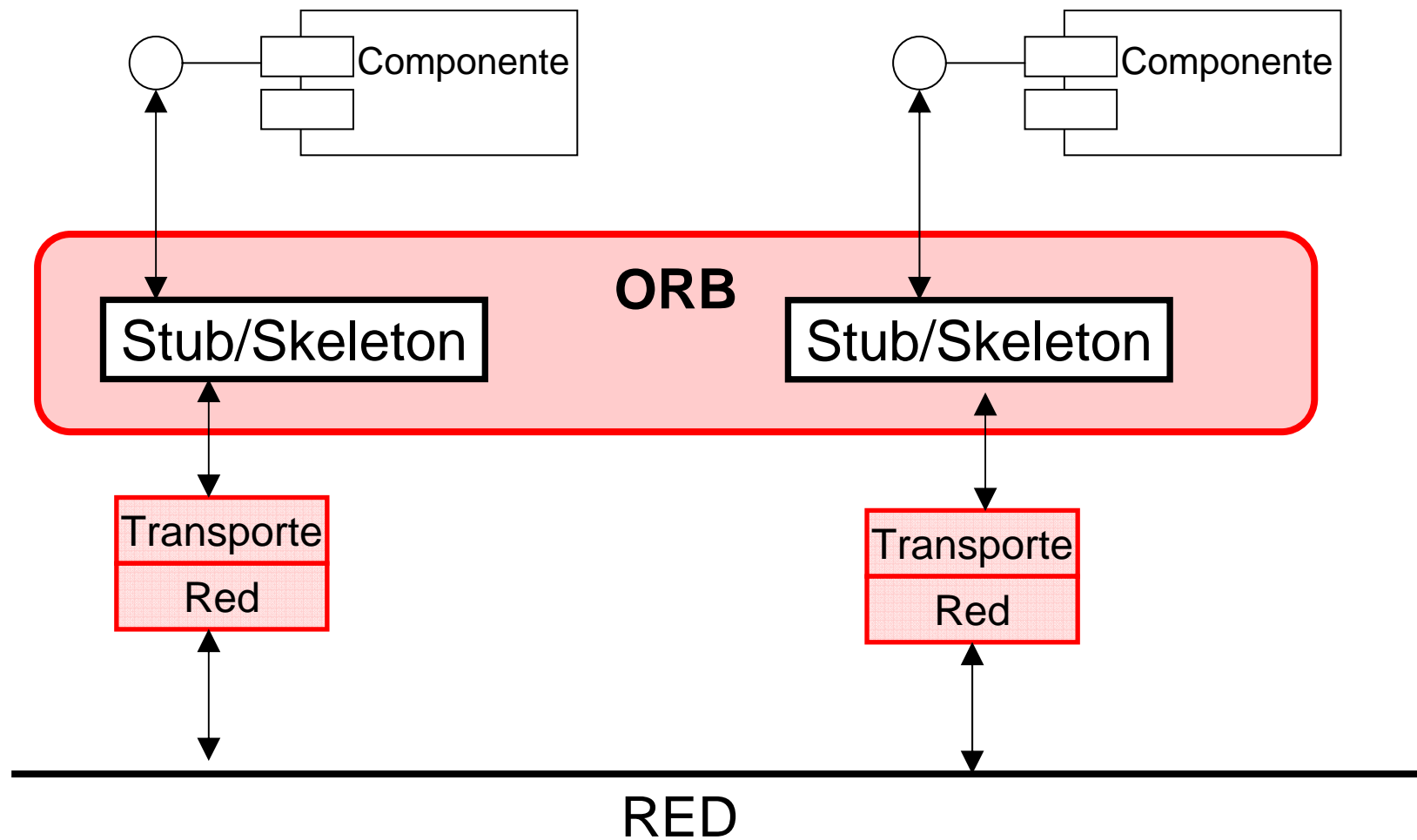


# TP





# ORB



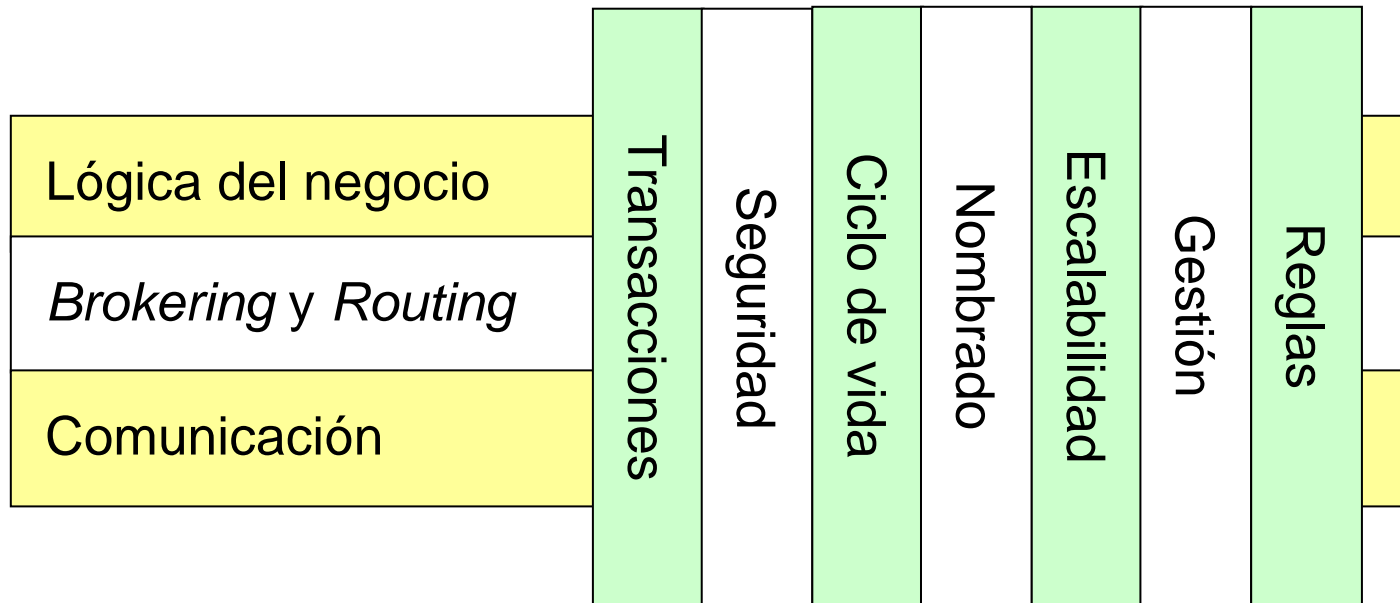


# Servidores de aplicaciones

- Gestionan las interacciones entre la capas cliente y datos
- Gestionan el entorno en el que se despliegan los componentes lógicos del negocio (contenedor)
- Proporcionan servicios MOM, ORB, transacciones, seguridad, balanceo de carga y gestión de recursos
- Aspectos a tener en cuenta:
  - Aspectos técnicos, caracter abierto, interoperabilidad, coste, madurez



# Servicios de infraestructura necesarios para la integración



Los servicios de infraestructura pueden ser realizados por diferentes tecnologías *middleware*.

Cuestión clave: **INTEROPERABILIDAD**



# Interoperabilidad

- Cuestiones:
  - ¿Es posible conseguir la interoperabilidad?
  - ¿Cuánto esfuerzo tenemos que invertir para ello?
- Plataforma software:
  - Combinación de tecnologías software necesarias para ejecutar aplicaciones
- Plataformas actuales
  - J2EE ← plataformas abiertas
  - CORBA ← plataformas abiertas
  - Microsoft.NET ← plataforma propietaria



# *Integration broker*

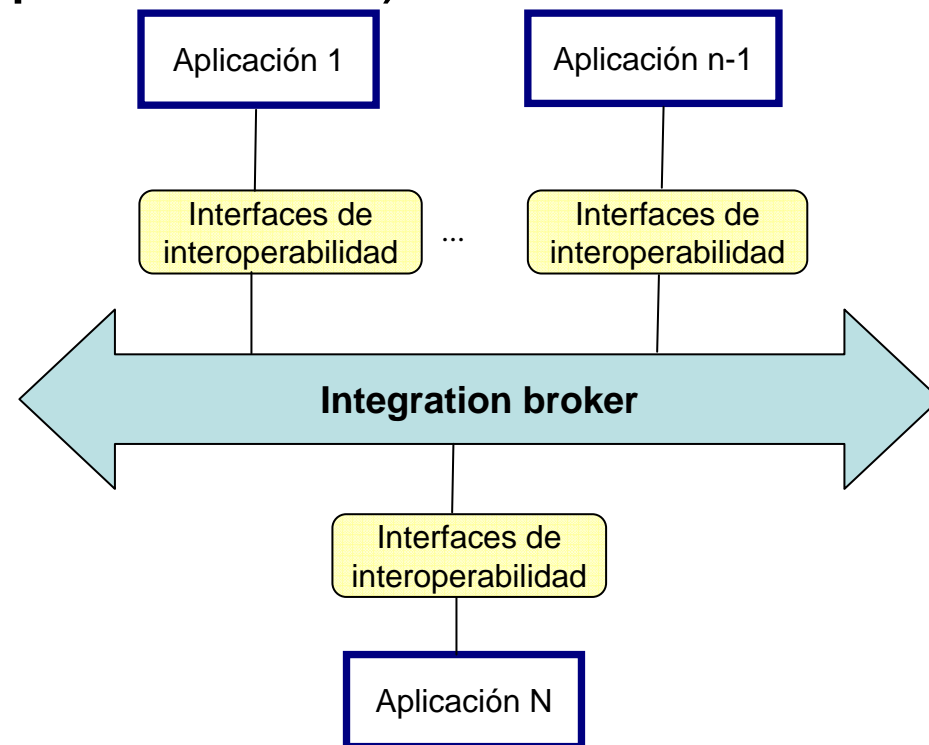
- Todos los productos EAI proporcionan un ***integration broker***
  - Es una abstracción utilizada para las tecnologías que componen la infraestructura de integración
  - Proporciona los servicios horizontales y verticales y centraliza la gestión de dichos servicios
- El *integration broker* se usa en todos los niveles de integración
- Evita los problemas de la integración punto a punto
  - Reduce la integración de n-a-n a 1-a-n



# Interfaces de interoperabilidad

- Definen qué servicios pueden pedir las aplicaciones clientes a las aplicaciones servidor (contratos entre las aplicaciones)

Si las interfaces permanecen inalterables, podemos reemplazar partes o toda la aplicación servidor sin influenciar para nada al cliente







# Técnicas de integración

- Aproximación **descendente** (*top-down*)
  - Estrategia de integración definida
  - Define en primer lugar la arquitectura del sistema integrado
- Aproximación **ascendente** (*bottom-up*)
  - Se centra en problemas individuales
  - Soluciones rápidas que pueden ocultar problemas



# Arquitectura de integración

- Especifica la estructura en su totalidad, los componentes lógicos y las relaciones lógicas entre las aplicaciones a integrar
- Objetivos esenciales de la integración:
  - Entradas de datos únicas
  - Acceso a datos con latencia cero
- SI basado en componentes virtuales:
  - Es un sistema que "parece" un sistema desarrollado nuevamente

**Un SI basado en componentes es el objetivo último de la EAI**



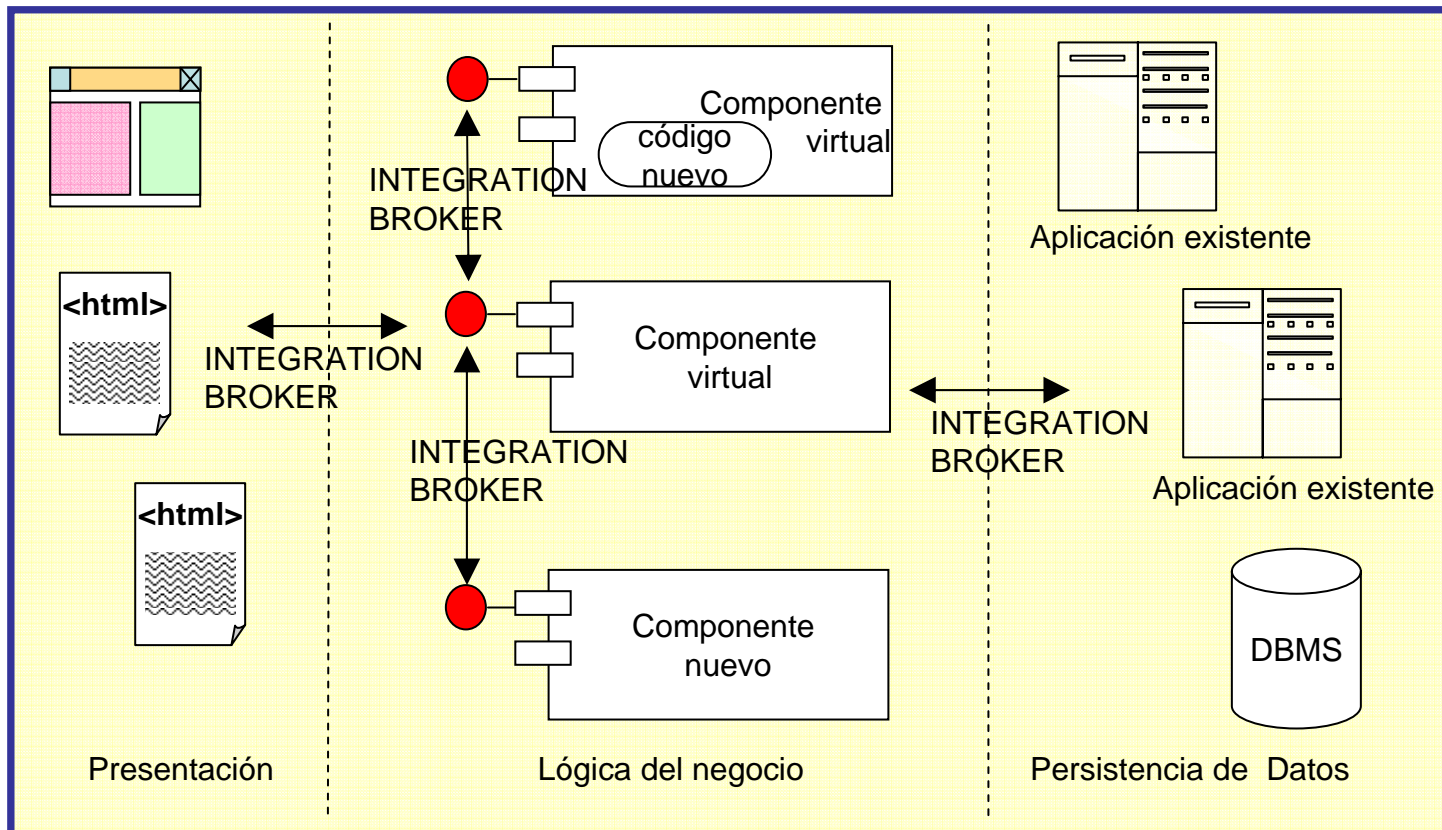
# Componentes virtuales

- Son bloques de construcción que encapsulan los detalles de las aplicaciones existentes
- Muestran la aplicación existente a través de unos interfaces
- La interoperabilidad a través de componentes virtuales se consigue mediante un *integration broker*
- En cada nivel de integración se construyen componentes de integración más abstractos y complejos



# Arquitectura de integración multi-capa

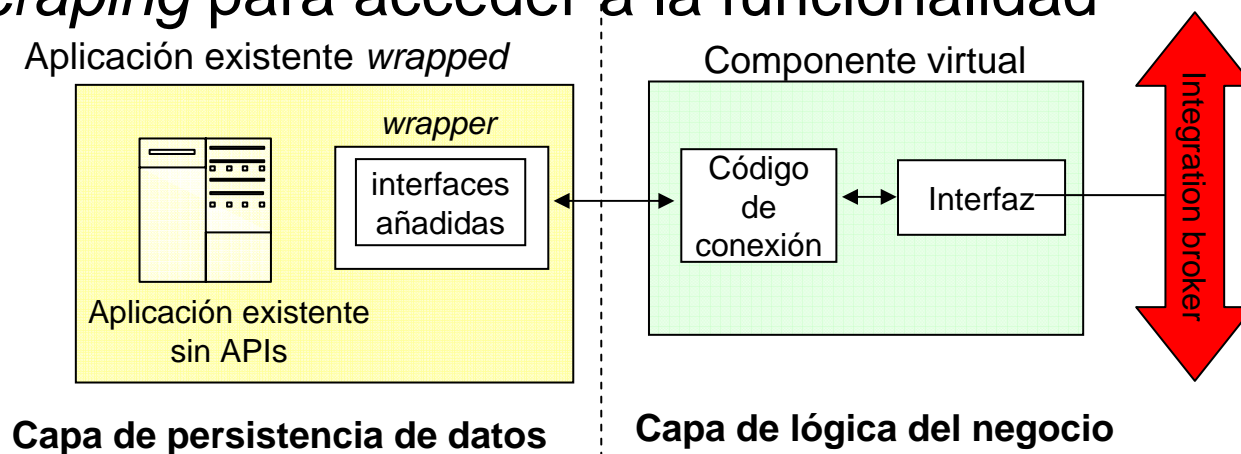
- Integración multi-capa basada en componentes





# Wrappers

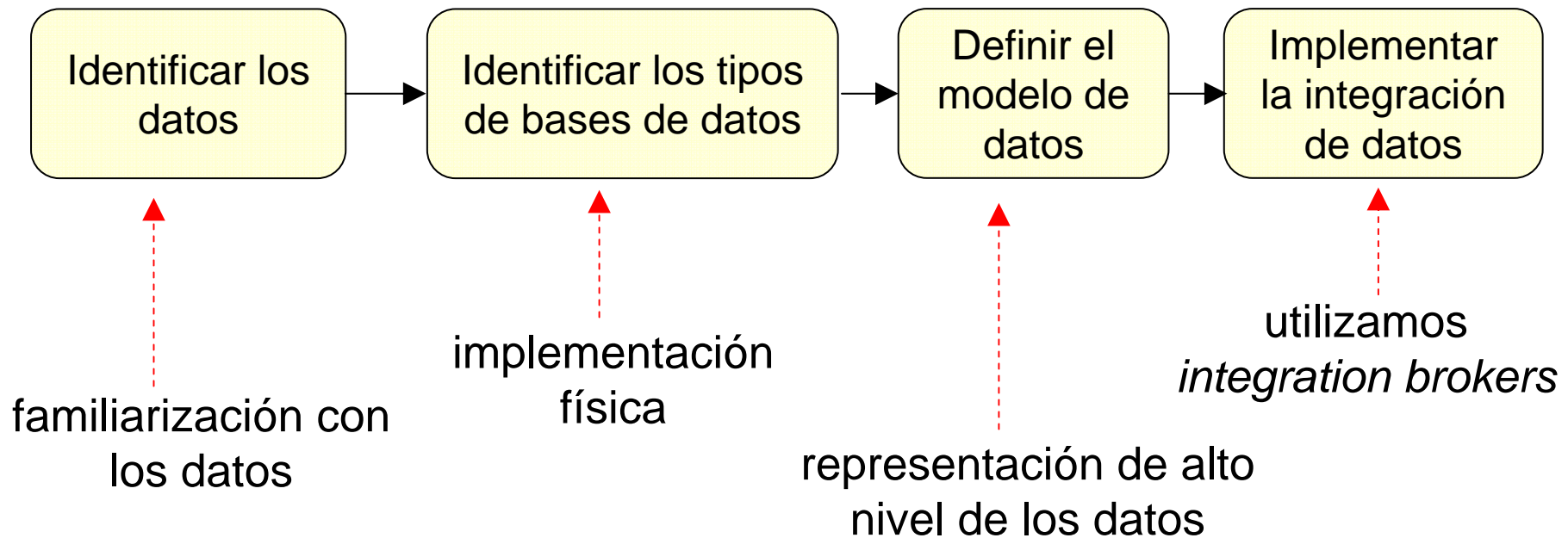
- Si las aplicaciones existentes no proporcionan ningún tipo de interfaces, necesitaremos desarrollarlas o comprarlas
- **Wrapper**: interfaz de aplicación que vamos a añadir
- El desarrollo de *wrappers* implica modificar código
- Si no se dispone del código se puede utilizar *Screen scraping* para acceder a la funcionalidad





# Integración a nivel de datos

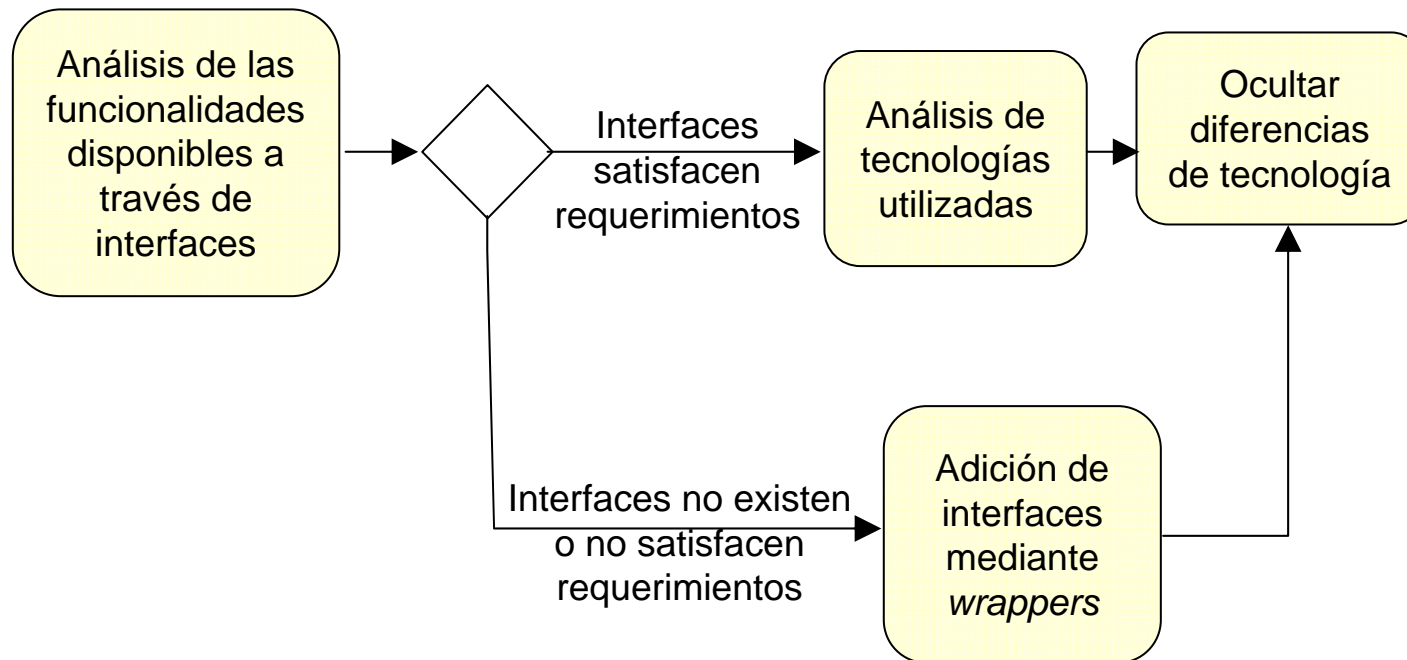
- No requiere cambios en las aplicaciones





# Integración a nivel de interfaz de aplicación

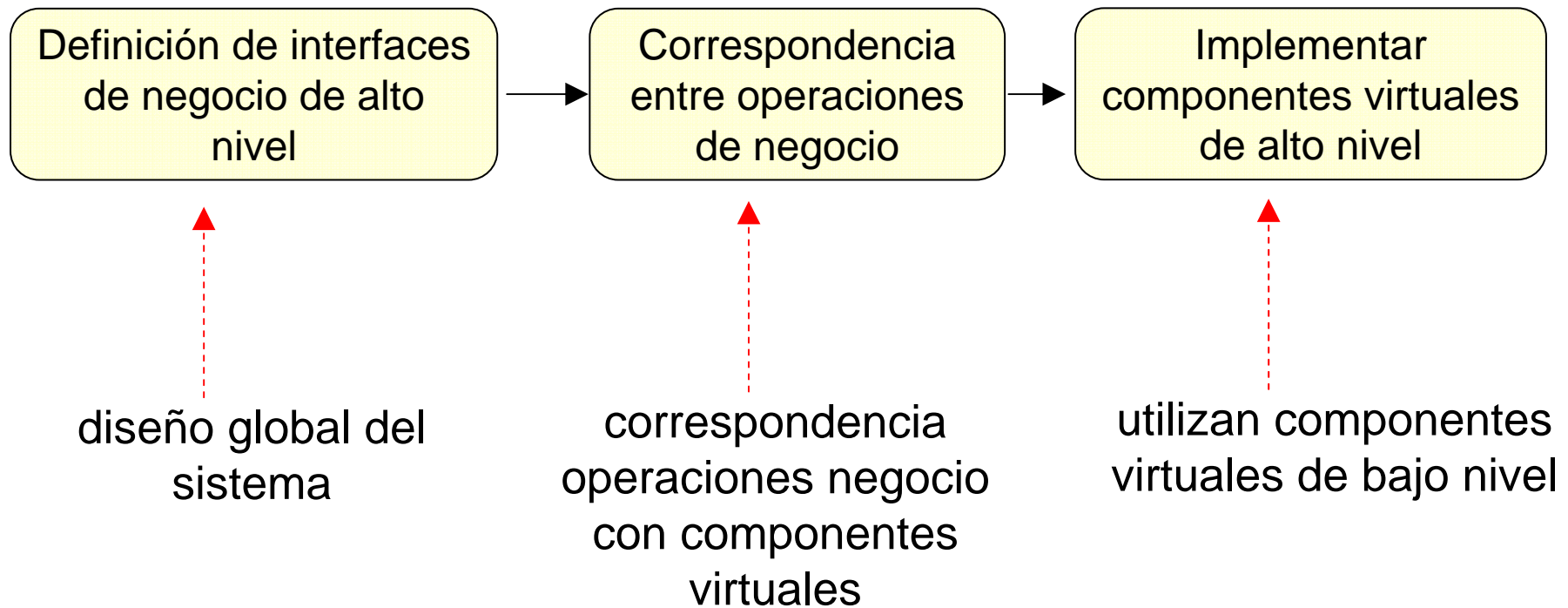
- Se consigue mediante APIs (utiliza *wrappers*)





# Integración a nivel de lógica del negocio

- Se desarrollan componentes virtuales (en varios niveles) que proporcionen las interfaces con los métodos de la lógica del negocio

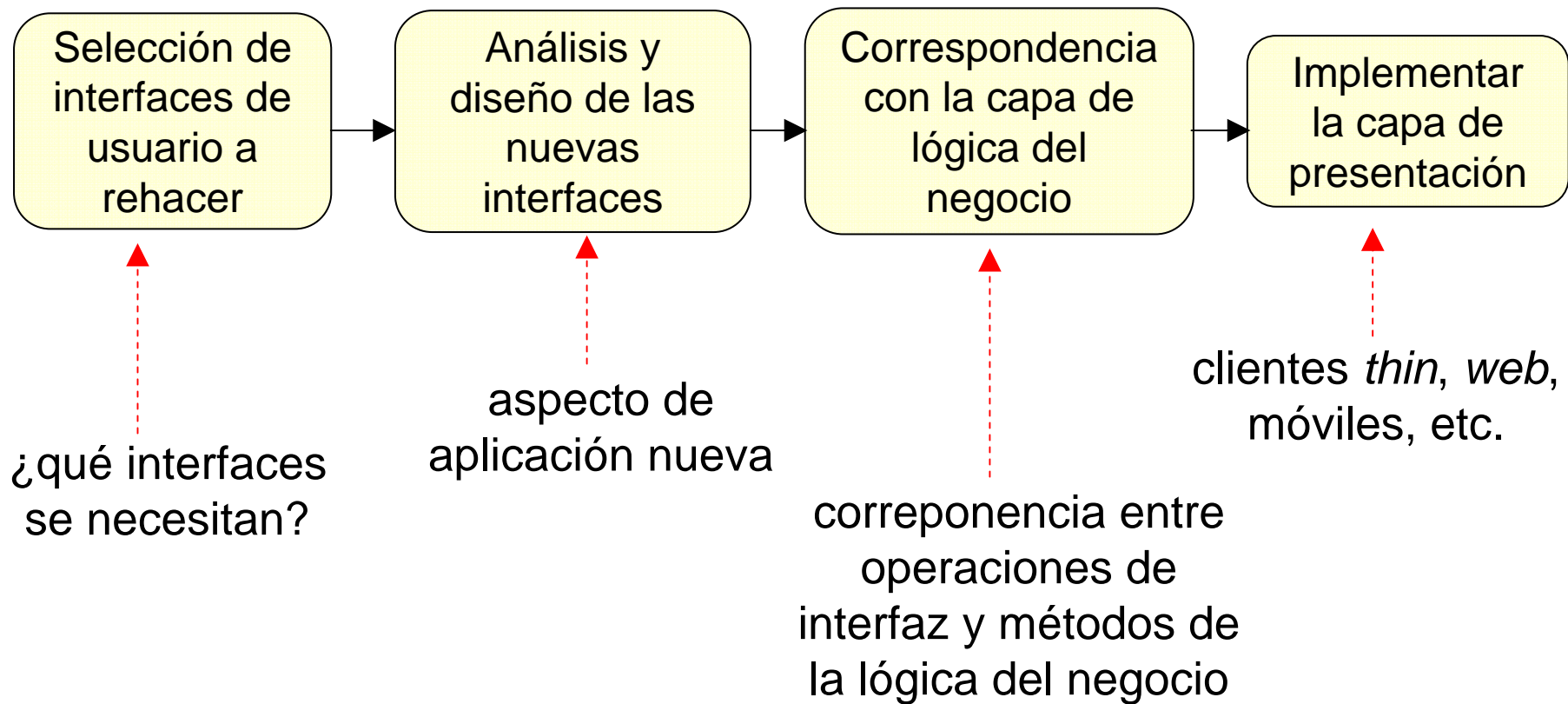






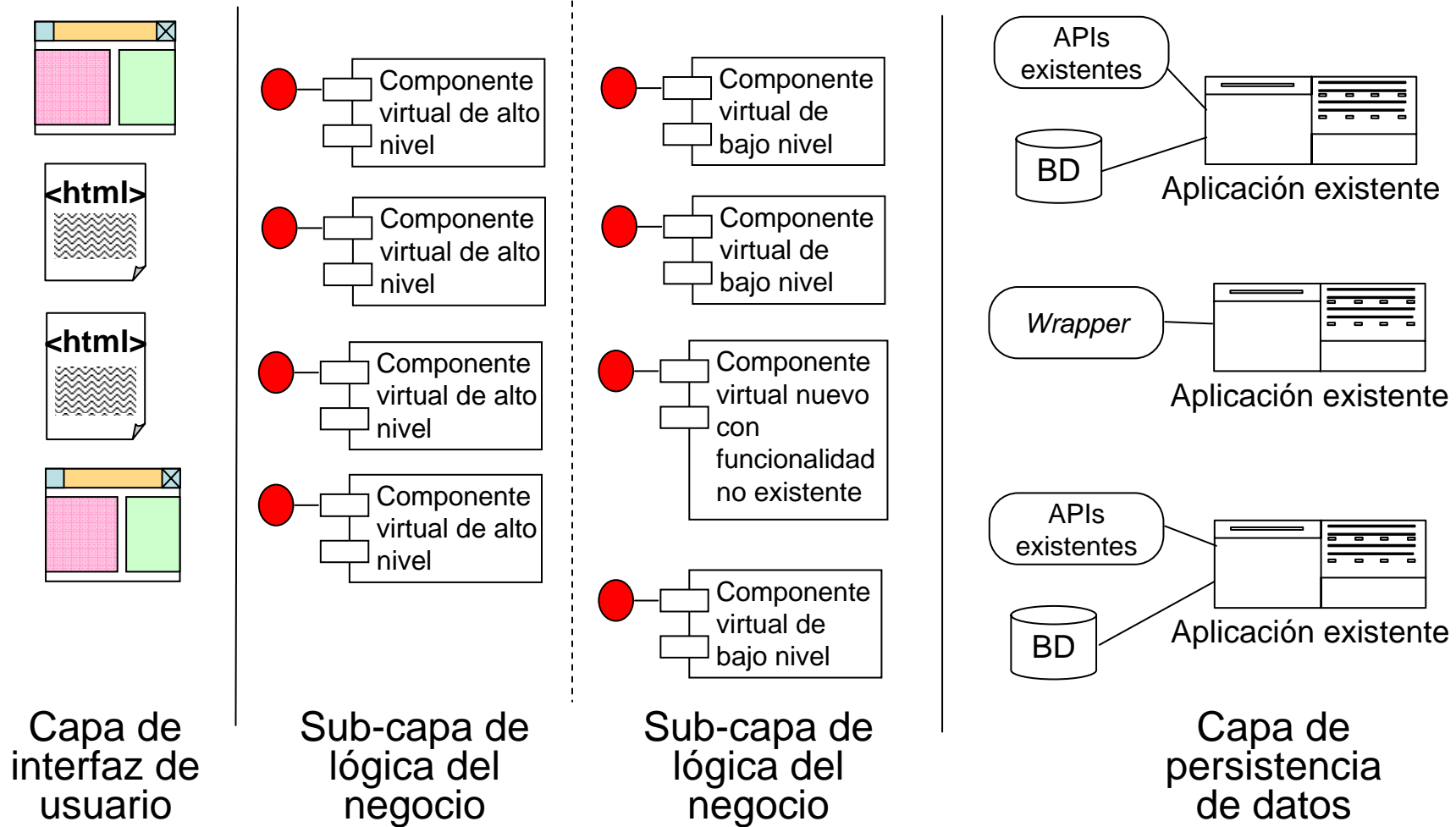
# Integración a nivel de presentación

- Se define una interfaz de usuario común



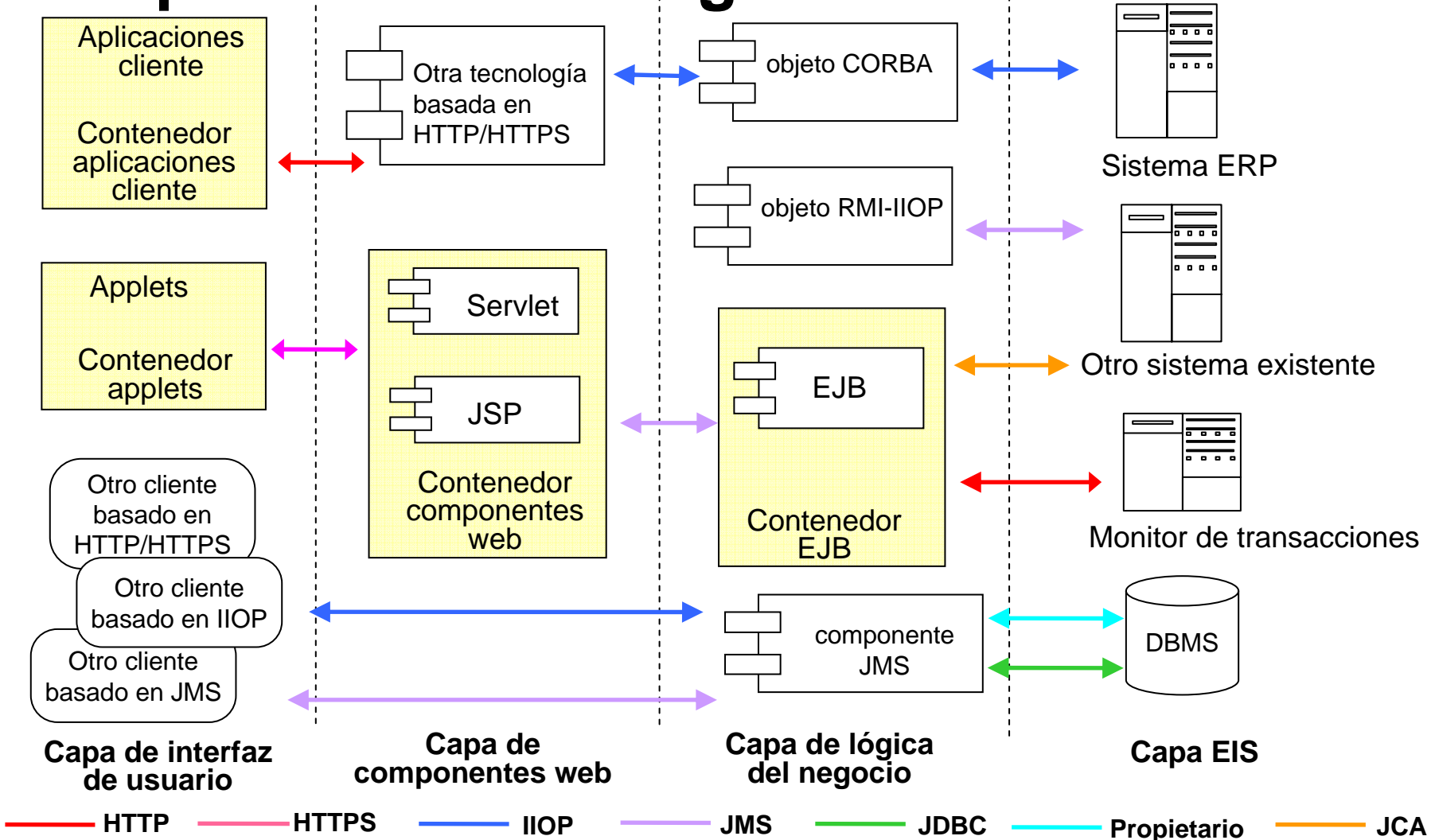


# Arquitectura de integración



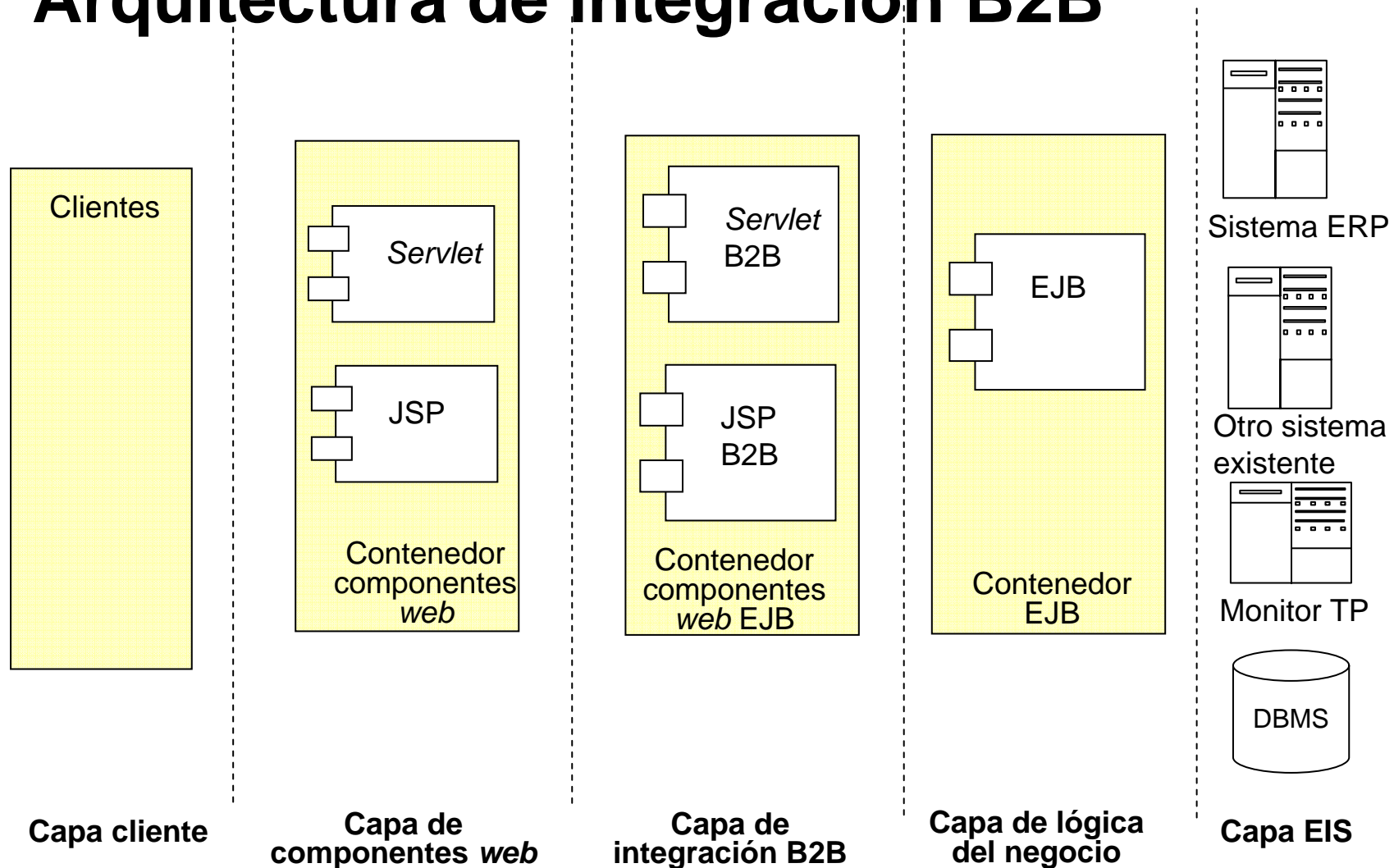


# Arquitectura de integración J2EE





# Arquitectura de integración B2B





# ¿Preguntas...?