



Arquitecturas Distribuidas

TEMA 5. Otras arquitecturas distribuidas

III. Otros entornos de objetos distribuidos

III. Otros entornos de objetos distribuidos

1. Problemas de CORBA
2. Java Enterprise Edition
 1. EJB
 2. Servidor de aplicaciones
 3. Arquitectura Java EE
 4. Ejemplo
3. Otros sistemas RMI
4. Sistemas basados en eventos
5. Otra definición de middleware

Problemas de CORBA

- Problemas en la especificación: las primeras versiones dejaron muchas cuestiones sin especificar. Se solventaron con las nuevas versiones pero ya era demasiado tarde para que la industria lo adoptara.
- Se especificó mal: complejo, muy caro de implementar completamente y ambiguo
- Los vendedores nunca proporcionaron productos completamente compatibles
- Sun lanzó los EJB, competencia directa y mucho más simples
- Microsoft nunca apoyó CORBA
 - Apostó por XML y lanzó SOAP como protocolo de invocación de procedimientos remotos
- La fragmentación del mercado hizo que muchos vendedores decidieran adoptar el modelo de Web Services, incluso aunque es menos eficiente.
- CORBA es una tecnología “de nicho” en la práctica

Java Enterprise Edition

- Antes denominada J2EE, ahora Java EE.
- Conjunto de librerías para el desarrollo de aplicaciones empresariales en múltiples capas.
- Basada en un modelo de componentes software
 - Nivel de abstracción superior a un objeto. Sería el equivalente software a un componente hardware: el componente define una interfaz y el conjunto de mensajes que admite y resultados que produce, que son siempre iguales.
- Las aplicaciones se construyen a partir de componentes “de negocio” (Enterprise Java Beans, EJB)
 - EJB = objeto remoto en el servidor
 - Los clientes invocan remotamente métodos de los EJB
- Se integra con una capa web, que utiliza JSP (equivalente Java a PHP, aprox.). Permite el uso de Web Services.
- Proporciona múltiples librerías auxiliares

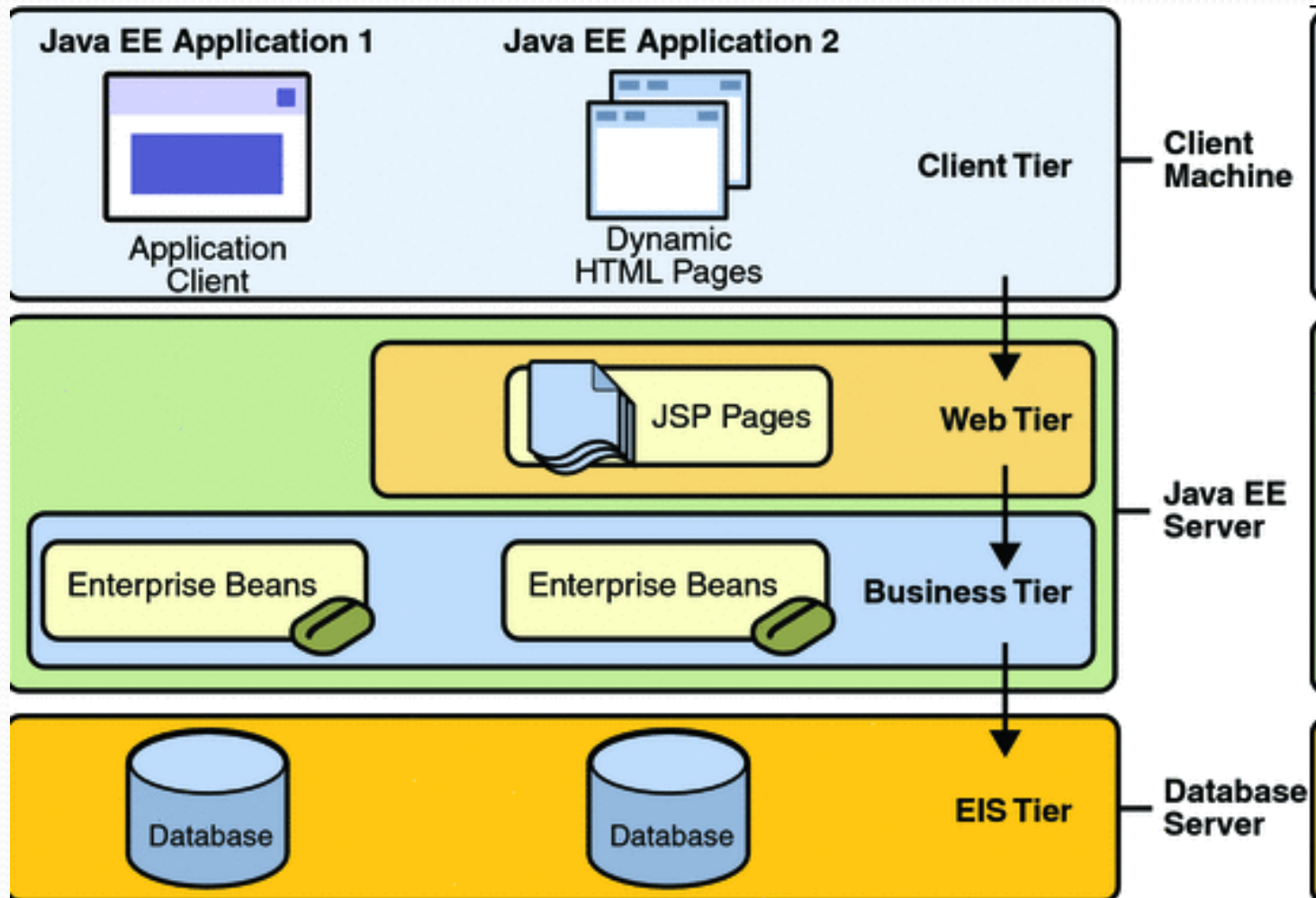
EJB

- Objeto/componente que proporciona métodos para ser invocados remotamente por los clientes
- Dos tipos
 - Session Bean: objeto transitorio, se instancia para responder a una invocación y se borra al terminar
 - Message-driven Bean: igual que el anterior pero además permite recibir mensajes asíncronamente de otros componentes
- Persistencia automática
 - Los campos de un componente *persistente* se guardan automáticamente en una BBDD.

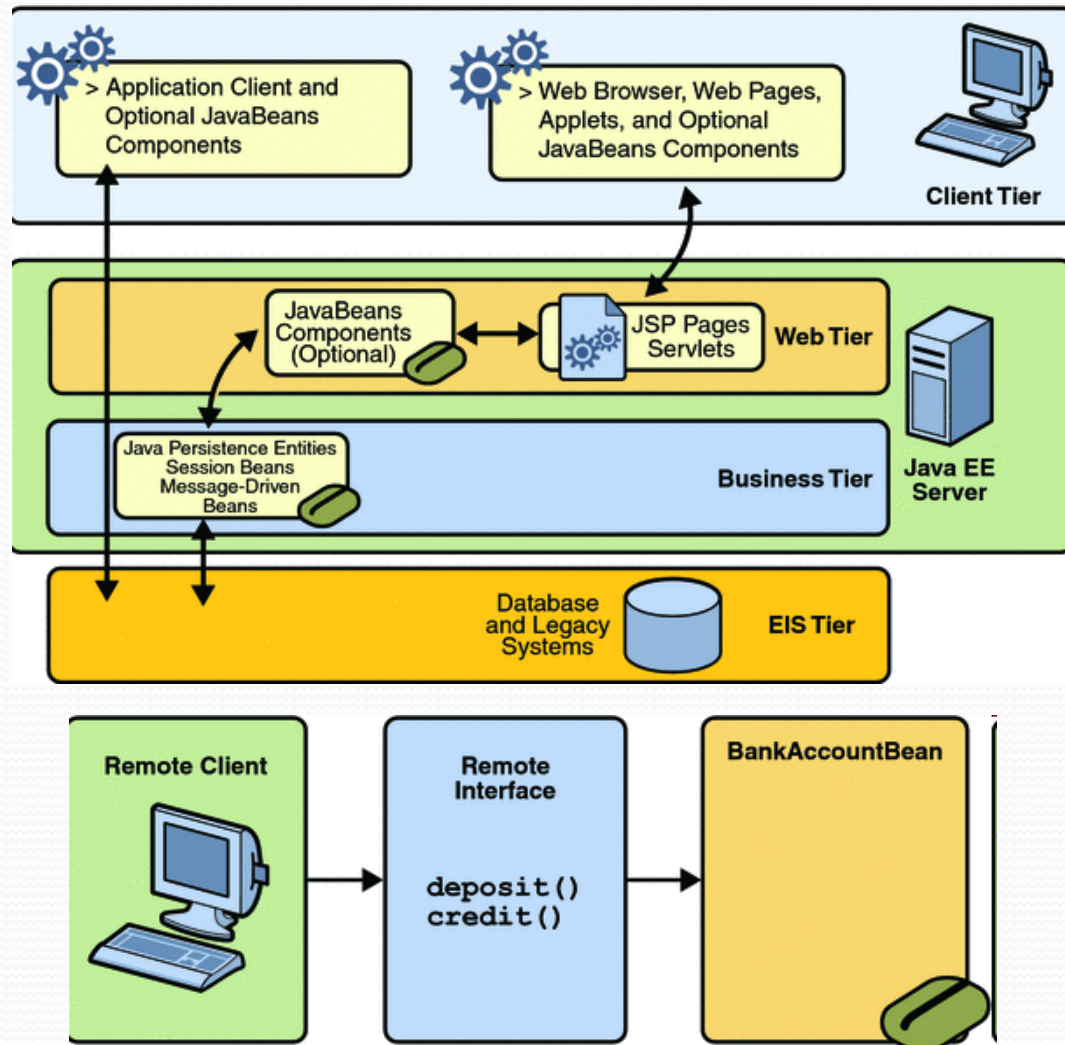
Servidor de aplicaciones

- Gestor de todos los elementos de la arquitectura
 - Contenedor de componentes (EJB, entidades persistentes)
 - Contenedor de JSP y páginas web
- Servicios adicionales
 - Transacciones
 - Seguridad
 - Mensajes
 - Conectores con tecnologías externas

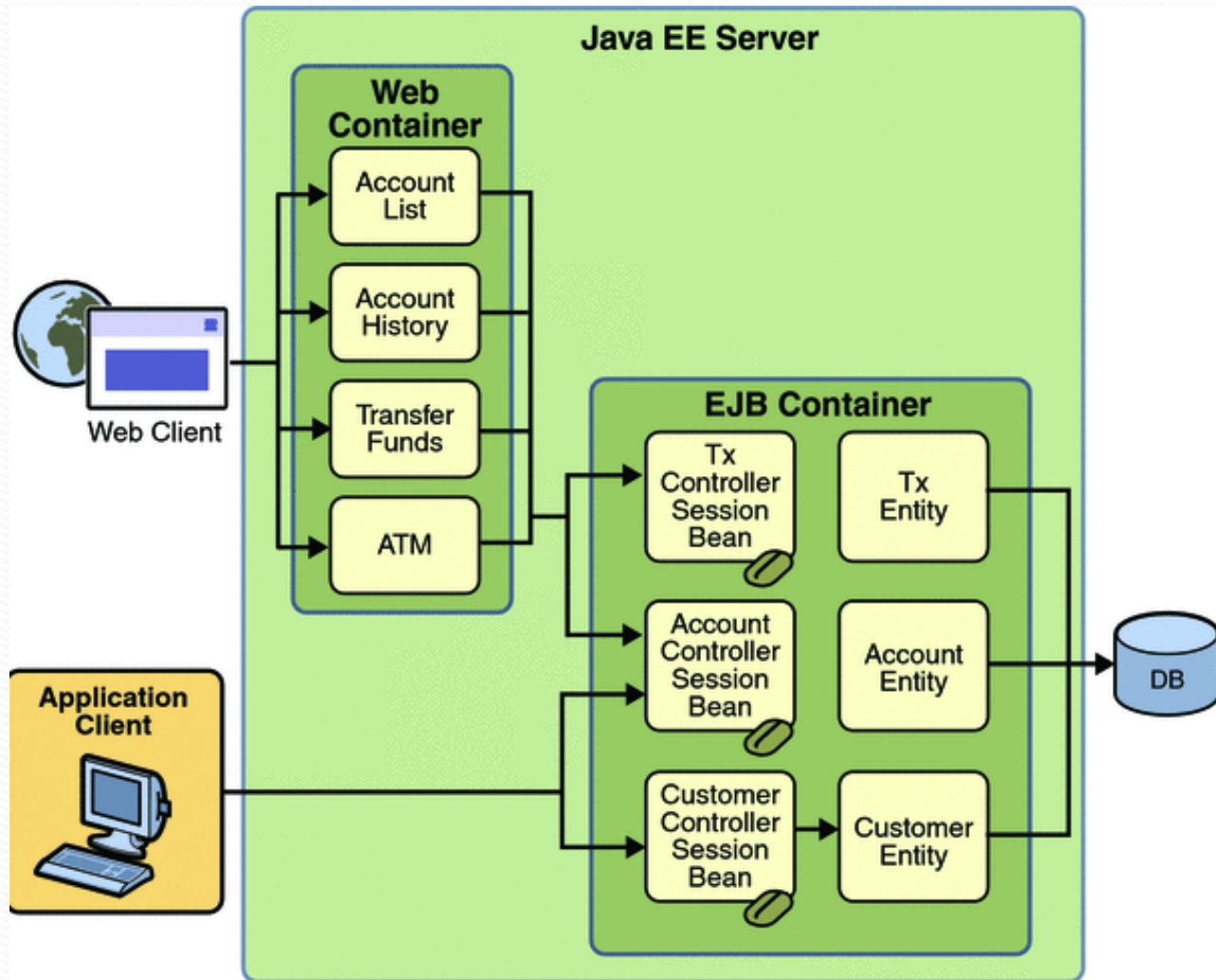
Arquitectura Java EE (I)



Arquitectura Java EE (II)



Ejemplo

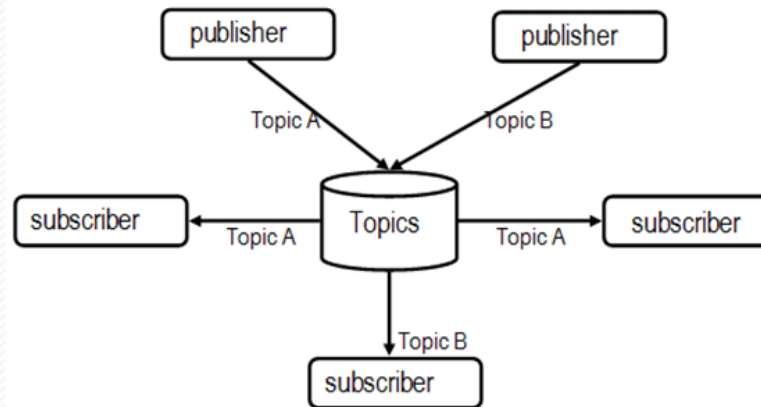


Otros sistemas RMI

- Java RMI y .NET Remoting
 - Exclusivos de la plataforma (no interoperables con otras tecnologías)
 - Más simples que CORBA
 - Servicios adicionales (transacciones, seguridad, etc.) más limitados

Sistemas basados en eventos

- Los componentes software reaccionan ante la ocurrencia de eventos (físicos o lógicos)
- Es necesario un mecanismo de comunicación de eventos asíncrono
- La mayoría de las tecnologías para sistemas distribuidos incorporan estos mecanismos (Java Messaging Service, Microsoft WCF, CORBA)
- Habitualmente basados en el patrón *publisher/consumer*



Otra definición de middleware

- En muchas ocasiones se denomina *middleware* a cualquier capa software que permite la integración e interoperabilidad de diferentes aplicaciones y servicios
- El *middleware* en sentido amplio es, por tanto, el software que adapta formatos y mensajes y permite intercambiar información entre aplicaciones, servicios y sistemas diferentes.
- Las capas *middleware* tienden a ser bloques transitorios en las arquitecturas software: se utilizan en muchas ocasiones para integrar tecnologías novedosas en sistemas más antiguos. A la larga acaban siendo incorporadas como parte integral del sistema.