

Taller de Sistemas de Información 2

Introducción a la plataforma Java Empresarial



Instituto de
Computación



Facultad de
Ingeniería



Universidad de la
República de Uruguay



Java EE 7

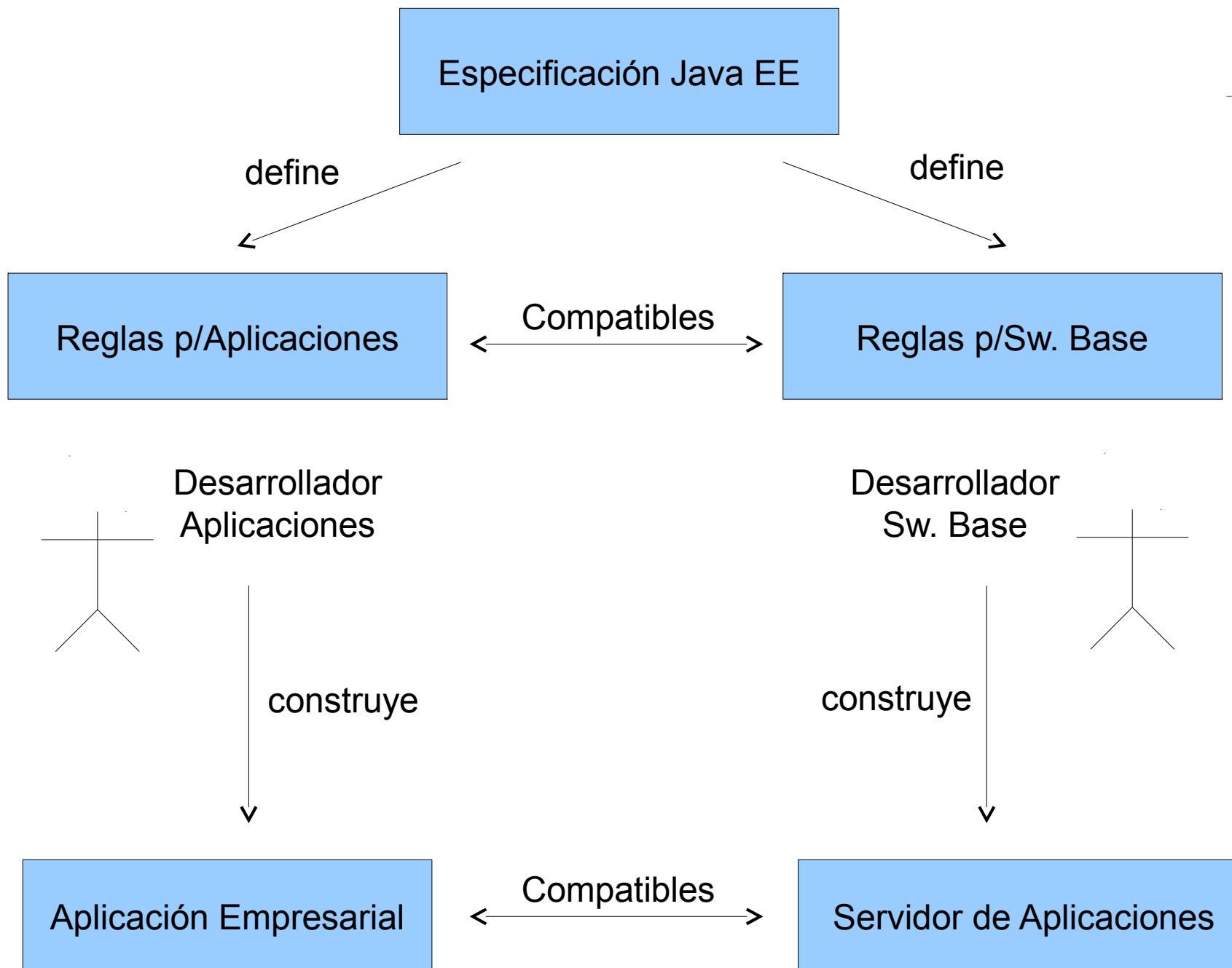
- El objetivo principal de Java EE 7, es el de simplificar el desarrollo de sistemas empresariales, brindando un marco común para la construcción de los diferentes componentes en Java EE
- Los desarrolladores se benefician de incrementos en la productividad
- El uso de mas anotaciones y convenciones, formas de empaquetado, etc., frente a configuraciones basadas en XML, suelen simplificar los procesos de construcción del software

Modelo de aplicación

- El modelo de aplicación de Java EE se basa en el lenguaje Java y la máquina virtual Java
 - Portabilidad, seguridad, productividad son algunos de los aspectos heredados de ambos elementos
- Java EE está diseñado para soportar aplicaciones que proveen servicios a los diferentes participantes de los procesos de negocio de una organización
 - Clientes, empleados, socios de negocio, proveedores, etc.
- Son aplicaciones complejas que implican acceso a múltiples fuentes de información, así como distribución de la misma a múltiples destinos

Modelo de aplicación

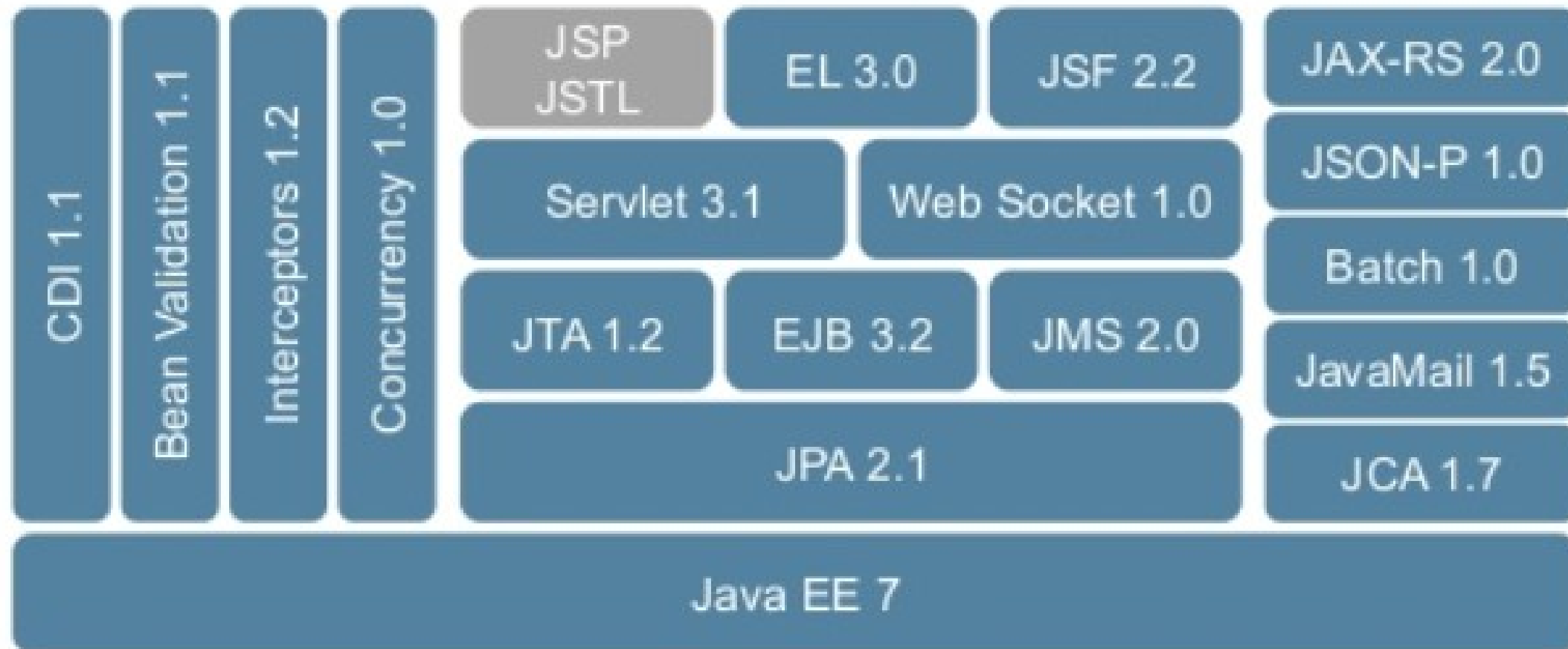
- El modelo de aplicación define una arquitectura para implementar servicios como aplicaciones distribuidas, brindando escalabilidad, accesibilidad y capacidades de administración
- El modelo particiona el trabajo necesario para construir una aplicación o servicio distribuido en
 - La lógica de negocio y presentación que debe implementar el desarrollador
 - Los servicios estándar provistos por la plataforma Java EE
- El desarrollador puede confiar en la plataforma para proveer soluciones a los problemas “difíciles” en la construcción de servicios empresariales



Principios centrales

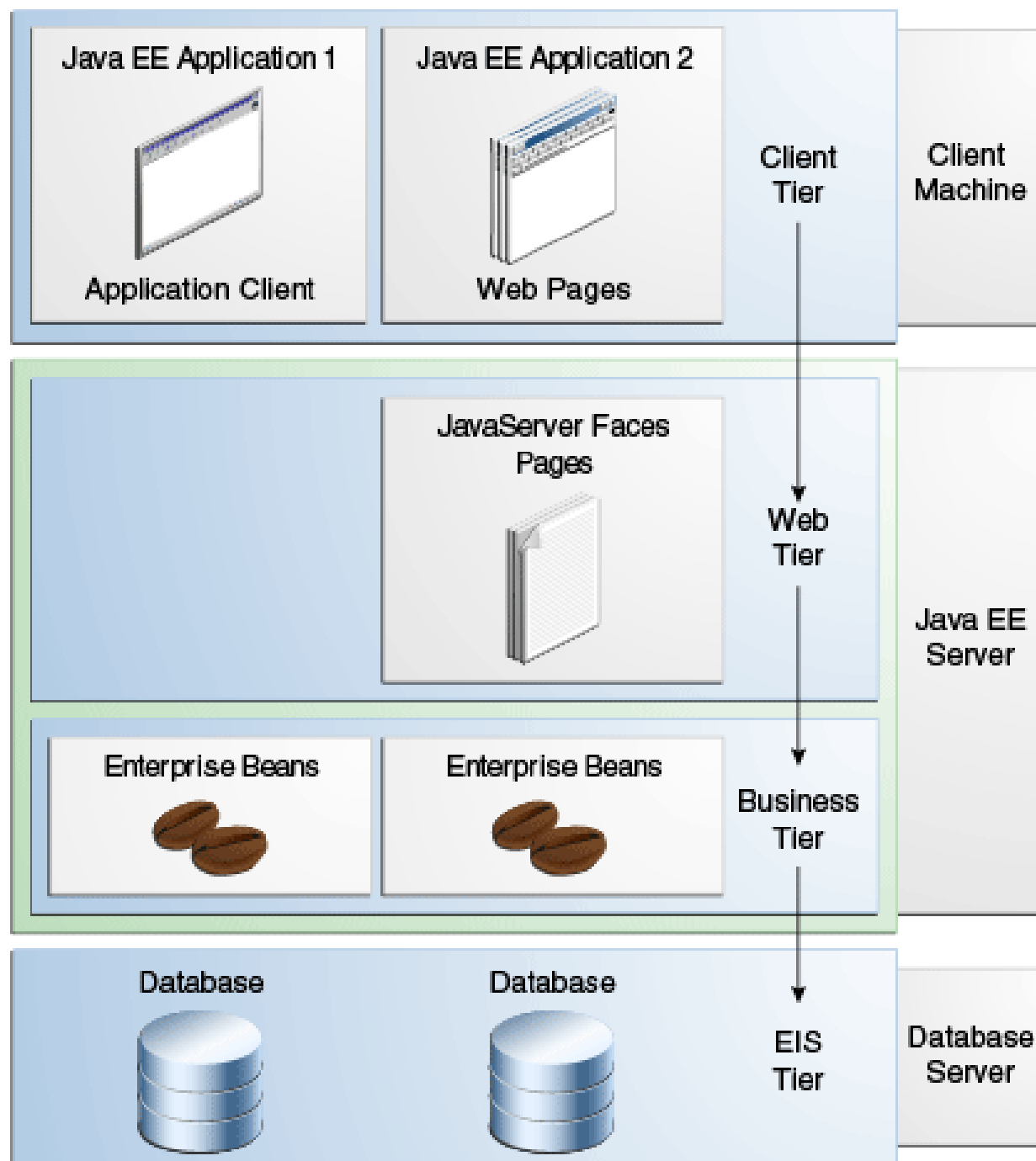
- Neutral con respecto al proveedor
- Núcleo estable / Ecosistema fuerte
- Runtime completamente integrado
- Convention-Over-Configuration / Intelligent Defaults
- APIs no redundantes, con roles especializados
- Enfoque en minimalismo, simplicidad y productividad
- Centrado en Java
 - Tipado fuerte, orientado a objetos, compatible hacia atrás

Java EE 7 – APIs



Aplicaciones distribuidas basadas en componentes

- La lógica de la aplicación es dividida en componentes de acuerdo a la función que deben cumplir
- Los componentes que forman la aplicación, son instalados en diferentes locaciones (maquinas), según el lugar físico al que corresponde el componente en el ambiente Java EE
- Tenemos los siguientes lugares físicos
 - Client-tier, ejecuta en la maquina cliente
 - Web-tier y Business-tier, ejecutan en el servidor Java EE
 - (Enterprise Information System) EIS-tier, ejecutan en el servidor EIS



Seguridad

- La plataforma Java EE permite definir restricciones en tiempo de deploy para las aplicaciones empresariales
- Esto permite que las restricciones de seguridad sean portables a diferentes mecanismos e implementaciones de esas restricciones de seguridad
- El objetivo es aislar al desarrollador de las aplicaciones de las complejidades de implementación de los mecanismos de seguridad

Seguridad

- A su vez, Java EE provee mecanismos de login estandarizados para determinar las identidades de los usuarios
- El objetivo es que la misma aplicación pueda funcionar en diferentes ambientes de seguridad, sin necesidad de cambios en el código

Componentes

- Las aplicaciones Java EE se construyen a partir de componentes
- Un componente es una unidad autocontenida de software, que es ensamblada en una aplicación Java EE, con sus clases y archivos relacionados, que a su vez se puede comunicar con otros componentes
- Tenemos los siguientes tipos de componentes
 - Clientes aplicación y applets, que ejecutan en el cliente
 - Servlets, Java Server Faces y Java Server Pages, componentes web que ejecutan en el server
 - Componentes EJB (Enterprise Beans), son componentes de negocio que ejecutan en el server

Componentes

- Son escritos en el lenguaje Java, siendo compilados como cualquier otra clase en el lenguaje
- La diferencia con una aplicación Java tradicional, es que estos últimos son ensamblados en una aplicación Java EE. Se verifica que los mismos estén bien formados, para luego ser desplegados en el servidor de aplicaciones
- Una vez desplegados, los mismos ejecutan en el contexto del servidor, siendo **administrados y gestionados por el servidor de aplicaciones**

Cientes web

- Un cliente web consiste en dos partes
 - Páginas web dinámicas, las cuales contienen diversos tipos de markup (HTML, XML, etc.), el cual es generado por componentes web que ejecutan en la capa web
 - Un browser que renderiza las páginas recibidas desde el servidor
- Un cliente de este tipo también se denomina cliente fino
- Generalmente estos clientes no consultan bases de datos, no hacen cálculos o transformaciones complejas ni se conectan con aplicaciones legadas

Cientes web

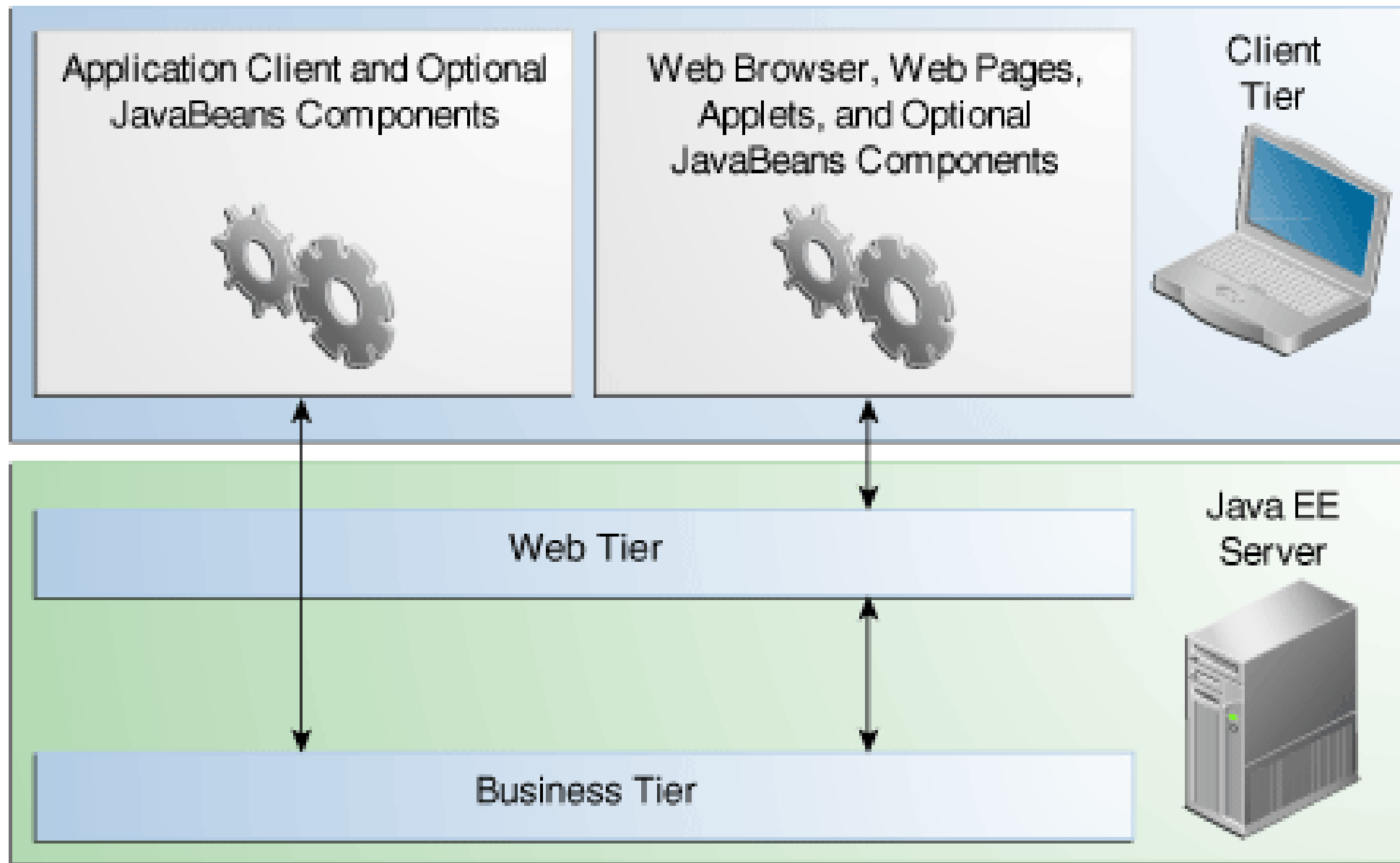
- Cuando se trabaja con este tipo de componentes, las operaciones complejas son descargadas en componentes EJB ejecutando en el servidor
- En este ambiente estos componentes pueden aprovechar mejor aspectos como la seguridad, velocidad y recursos disponibles de la plataforma

Cientes aplicación

- Estos clientes ejecutan en la maquina cliente, brindando un mecanismo para ejecutar aplicaciones ricas que no pueden ser representadas por un lenguaje de markup
- Estas aplicaciones pueden ser aplicaciones Swing, AWT o aplicaciones de consola
- Pueden acceder a los EJB en la capa de negocio, así como consumir servicios REST/SOAP provistos en la capa web de la aplicación
- Asimismo, estos clientes pueden ser escritos en otros lenguajes diferentes de Java, si es que la interacción se realiza a través de servicios web

Cientes Applet

- Es una pequeña aplicación Java que ejecuta en el contexto de una maquina virtual Java, embebida en un web browser
- Generalmente los clientes requieren además el plugin de Java en el browser y una política de seguridad para ejecutar exitosamente
- Por este motivo, y por la forma en que se deben programar las interfaces en este caso, es que se prefieren los clientes finos a la hora de ejecutar en el contexto de un browser

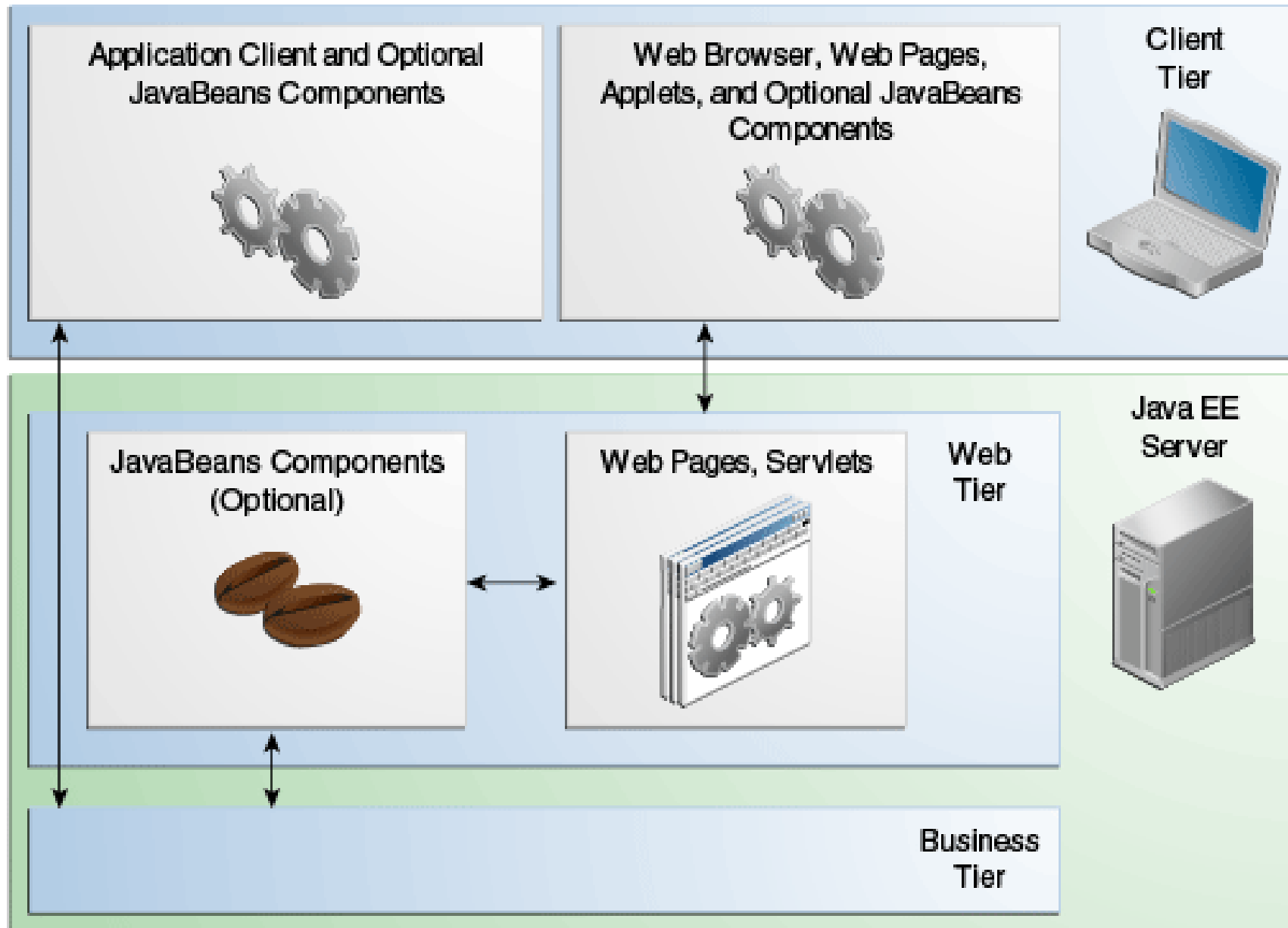


Componentes web

- Estos pueden ser Servlet o paginas creadas con Java Server Faces o Java Server Pages
- Un Servlet es una clase Java que dinámicamente procesa solicitudes HTTP para generar respuestas HTTP
- Una pagina JSP es un documento HTML con código Java embebido que permite generar contenido dinámico
- Una pagina JSF es un componente web que permite construir una interfaz de usuario en base a componentes gráficos y eventos

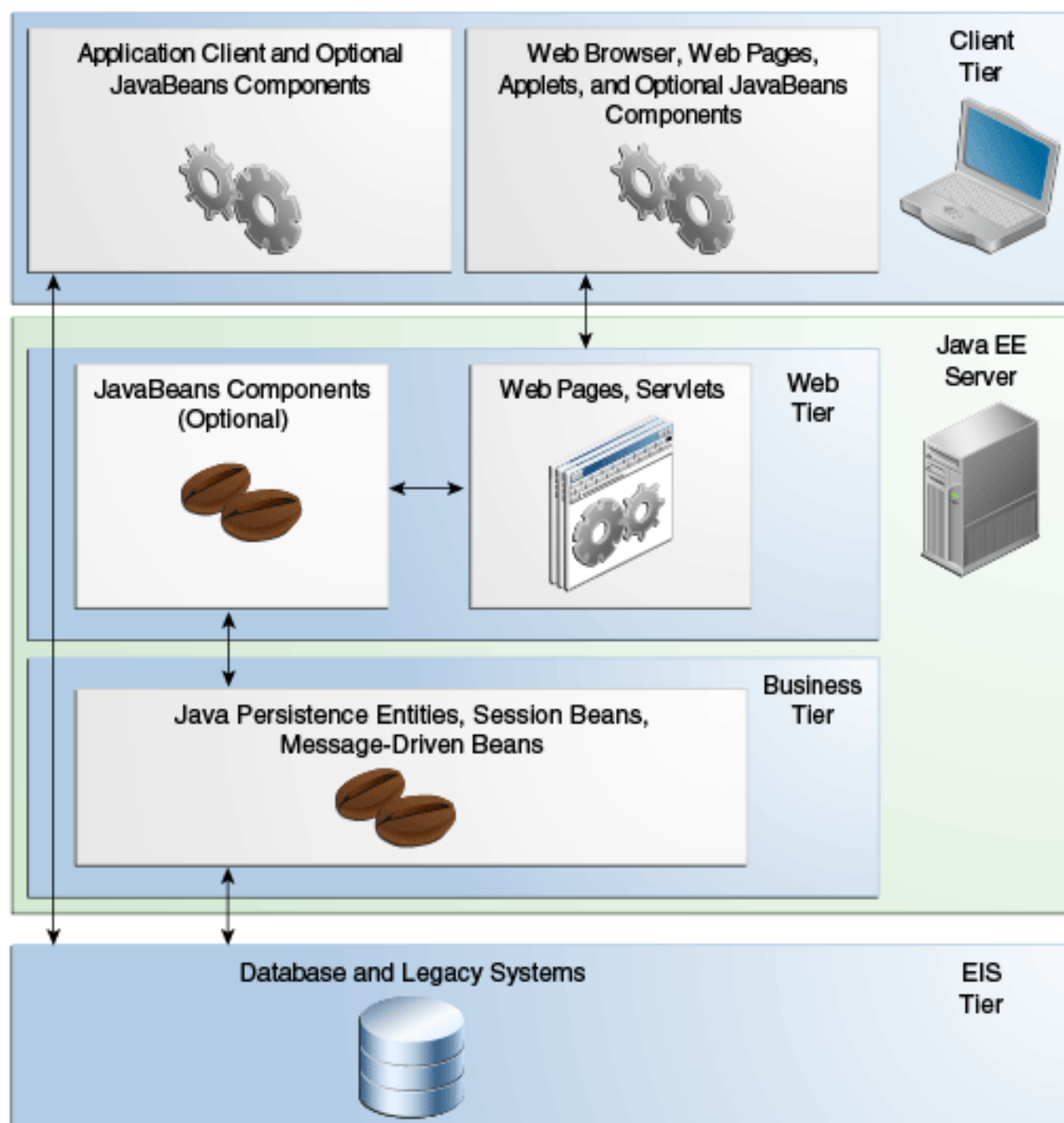
Componentes web

- A su vez se puede contar con paginas HTML estáticas y recursos auxiliares (imágenes, scripts, etc.)
- Estos pueden ser usados por los componentes, pero no son componentes en si mismo
- De la misma forma, los componentes web pueden ser acompañados por clases utilitarias, pero las mismas no son componentes en si mismo



Componentes de negocio

- La lógica que resuelve el problema de un determinado aspecto del negocio, suele implementarse en los componentes de negocio EJB, los cuales ejecutan en la capa de negocio o en la propia capa de negocio
- Los componentes de negocio generalmente reciben datos de las capas superiores, procesándolos y enviándolos luego a las capas inferiores para persistencia
- Asimismo, el proceso inverso también es ejecutado por estos componentes, recuperando datos de la capa de persistencia, para devolverlos a capas superiores



EIS – tier

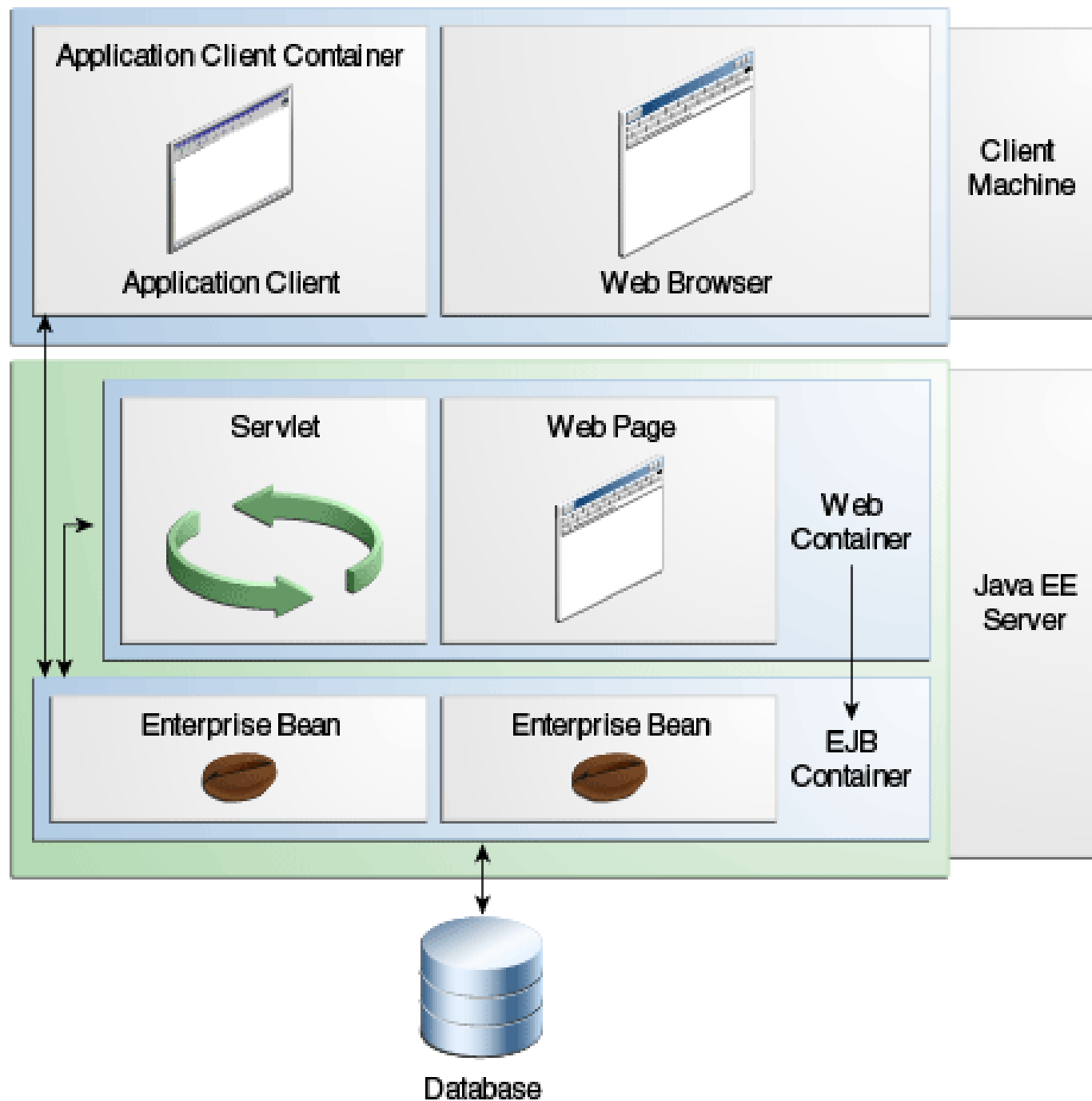
- Es una capa que engloba sistemas y servicios externos en los que una solución Java EE se apoya
- Puede incluir
 - Enterprise Infrastructure Systems
 - Enterprise Resource Planning
 - Mainframe Transaction Processing
 - Database Systems
 - Otros sistemas legados

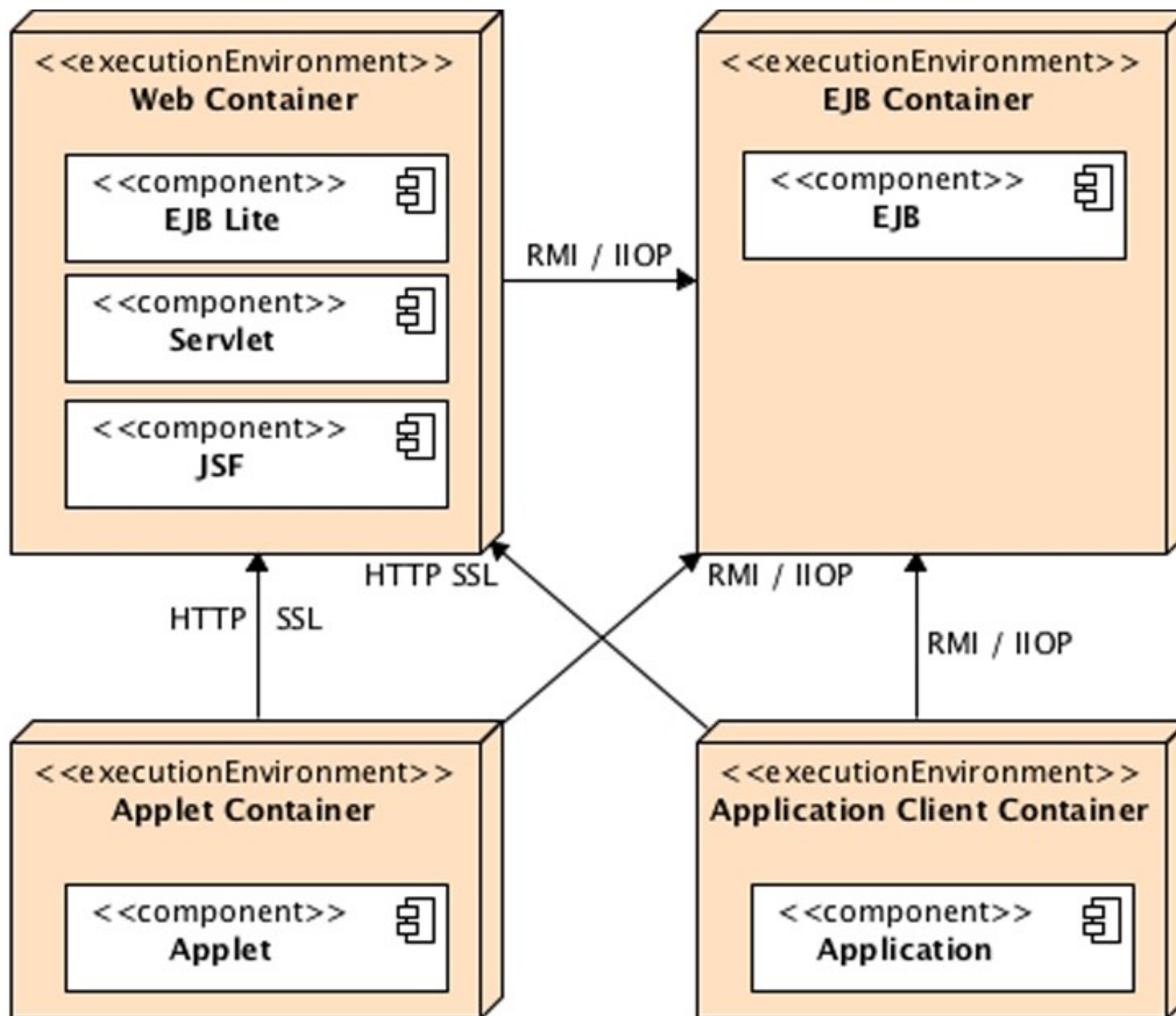
Arquitectura

- Java EE es una especificación implementada por una serie de contenedores
- Un contenedor es un runtime para las aplicaciones Java EE
 - Provee servicios para los componentes allí instalados
 - Manejo del ciclo de vida, inyección de dependencias, acceso a recursos, concurrencia, etc.
- Se agrupan en una entidad denominada “Servidor de aplicaciones”

Servidor de aplicaciones

- Es un software que provee un ambiente de ejecución para las aplicaciones
- Le brindan a los componentes, la facilidad para poder acceder a los recursos necesarios para funcionar
- Es un host para los diferentes contenedores que utiliza la aplicación





Contenedores

- La infraestructura de Java EE esta particionada en contenedores
- Cada contenedor:
 - Tiene un rol especifico
 - Soporta una serie de APIs
 - Ofrece una serie de servicios
- Esconden complejidad técnica y aumentan la portabilidad

Contenedores

- Dependiendo del tipo de aplicación que vayamos a armar, deberemos entender las capacidades y limitaciones de los contenedores involucrados
- Por ejemplo, si armamos una aplicación web:
 - Si no necesitamos lógica transaccional remota, podemos utilizar un solo contenedor
 - Pero si tenemos que invocar lógica remota o utilizar mensajería, necesitaremos mas de un contenedor

Contenedores

- Applet Containers
 - Provistos por el browser para ejecutar applets
 - Este contenedor utiliza un modelo de sandbox para garantizar la seguridad del ambiente de ejecución
- Application Client Containers
 - Consisten en las clases y bibliotecas necesarias para desarrollar aplicaciones que dialoguen con los contenedores Web y EJB

Contenedores

- Web Container
 - Provee los servicios necesarios para ejecutar componentes web
 - Servlets, JSP, Filtros, JSF, Listeners, Web Services y EJB lite
 - Es responsable de soportar el protocolo HTTP y HTTPS entre el cliente y el servidor
 - Las paginas web accedidas por el cliente, son generadas en este contenedor

Contenedores

- EJB Container
 - Es el responsable por manejar la ejecución de los componentes EJB
 - Session Beans y Message Driven Beans
 - Contienen la lógica de negocio transaccional (potencialmente remota) de la aplicación
 - Provee servicios como la gestión del ciclo de vida, transacciones, mensajería, seguridad, concurrencia, distribución, nombrado, etc.

Servicios

- Los contenedores proveen servicios a los componentes instalados
- Como desarrollador, permiten que nos concentremos en el desarrollo de los aspectos relacionados con el negocio
- Los problemas de base relacionados a las aplicaciones empresariales, son resueltos por estos servicios

Servicios

- Java Transaction API
 - Provee un servicio para el demarcado de transacciones, para ser usado desde las aplicaciones
 - También provee una interfaz entre el gestor de la transacción y el gestor de recursos (que en última instancia provee la implementación de la transacción)

Servicios

- Java Persistence API
 - Es el API estándar para mapeos objeto – relacional (ORM)
 - Brinda un lenguaje de consulta (JPQL) para que se pueda acceder en forma orientada a objetos a la información almacenada en la base
- Bean Validation
 - Provee mecanismos para realizar validaciones a nivel de métodos y clases

Servicios

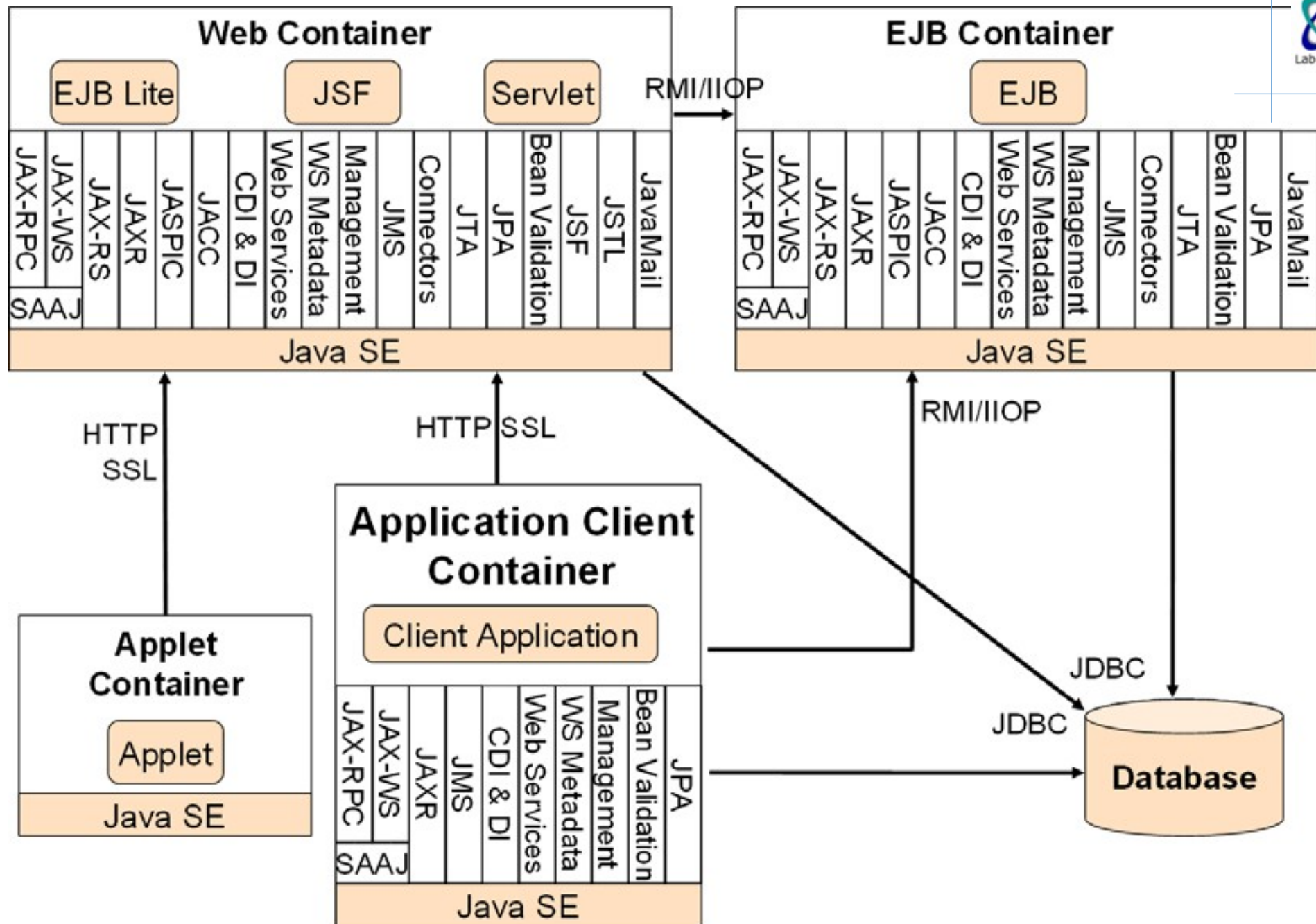
- Java Message Service API
 - Permite que los componentes se comuniquen en forma asíncrona a través de mensajes
 - Soporta comunicación confiable usando un modelo P2P o Pub-Sub
- Java Naming and Directory Interface API
 - Es usada para acceder a servicios de nombrado y directorio
 - Permite asociar nombres a objetos, así como localizar los mismos a través de sus nombres

Servicios

- Seguridad
 - Java Authentication and Authorization Service (JAAS) provee los mecanismos para resolver la autenticación y autorización necesarias para acceder a los diferentes componentes de una aplicación
- Web Services
 - Java EE brinda soporte para servicios web SOAP (JAX-WS) y servicios web RESTful (JAX-RS)

Servicios

- Inyección de dependencias
 - Desde Java EE 5, algunos recursos (data sources, JMS factories, unidades de persistencia, EJBs) podían ser inyectados en componentes administrador
 - En Java EE 7 se va un paso mas al incorporar CDI (Context and Dependency Injection) como parte de la especificación



Packaging

- Para poder ser instalados en un container, los componentes deben primero ser empaquetados siguiendo un formato predefinido
- Java define el formato JAR (Java ARchive)
 - Agregan múltiples archivos, como ser clases, descriptores, recursos, bibliotecas externas, etc
 - Se comprimen usando el formato ZIP
- Java EE extiende esta idea, definiendo diferentes tipos de archivos comprimidos

