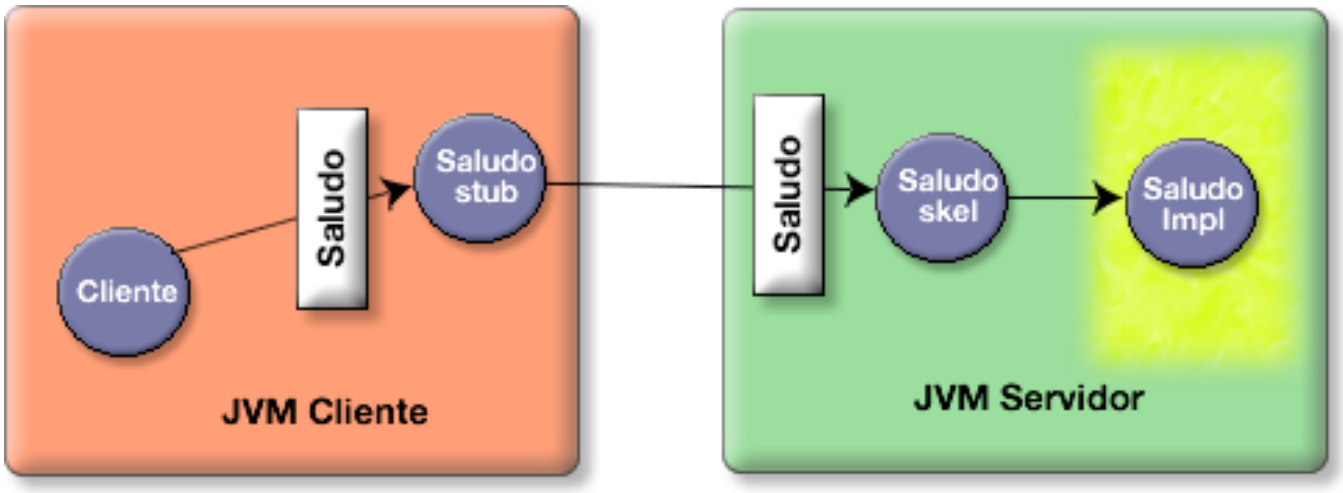
Especialista en Aplicaciones y Servicios Web con Java Enterprise

Enterprise JavaBeans

Sesión 3:

La arquitectura EJB en detalle

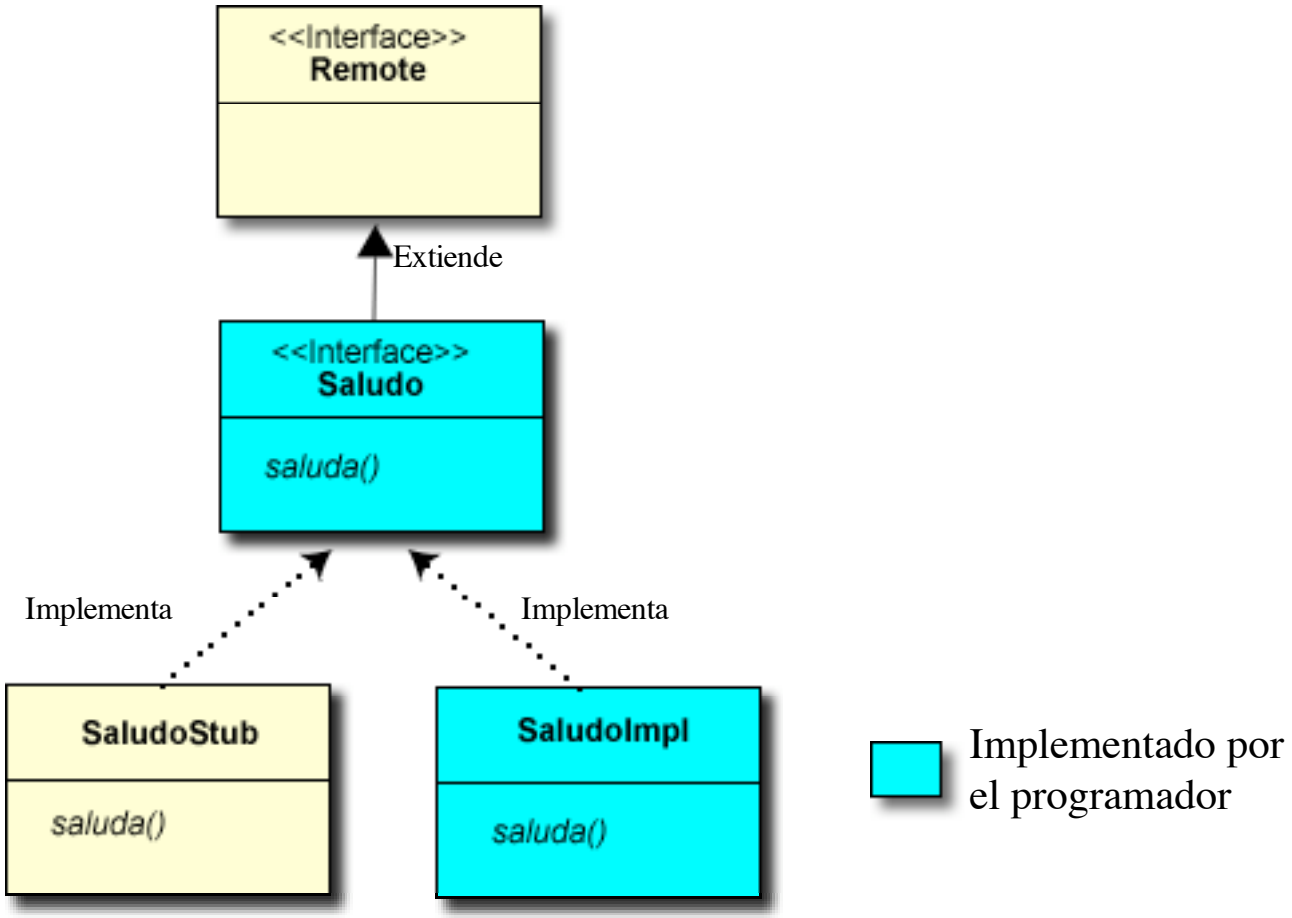
Repaso de RMI



Clases remotas

- Para construir y usar una clase `Saludo` remota con el método `saluda()` debemos:
  - Definir una interfaz `Saludo` con el método `saluda()` que extienda la interfaz `java.rmi.Remote` y hacer que este método declare la interfaz `RemoteException`
  - Definir una clase `SaludoImpl` que implemente la interfaz `Saludo`
  - Llamar al compilador RMI (rmic) para que cree las clases `SaludoStub` y `SaludoSkeleton`
  - Crear al menos un objeto de la clase SaludoImpl y darle un nombre
  - Un cliente debe localizar el objeto remoto, obtener el stub y realizar las llamadas al stub. El cliente debe tener en su máquina virtual la clase `SaludoStub.class`

Diagrama UML

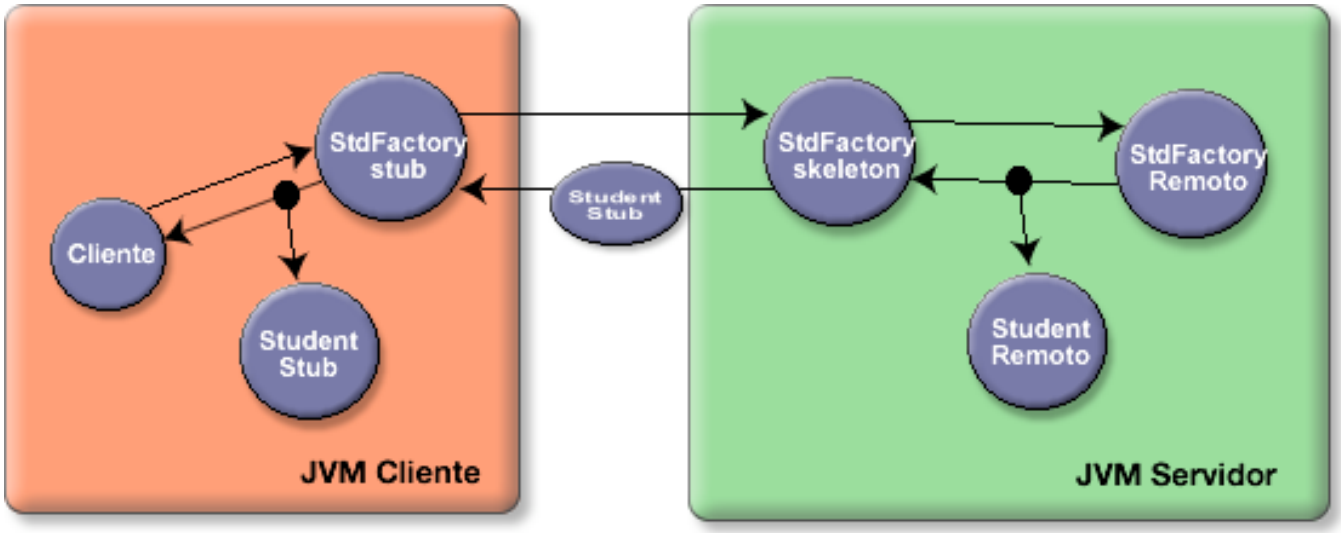


Paso de argumentos

- Los objetos que se pasan como argumentos en las llamadas a métodos remotos y los objetos devueltos deben ser de uno de los siguientes tipos:
  - Objetos primitivos Java
  - Objetos serializables
  - Un array o una colección de objetos primitivos o serializables
  - Un objeto Remote

Devolución de un objeto remoto

- Un caso especial, muy frecuente en EJB, es cuando un objeto remoto devuelve una referencia (stub) a otro objeto remoto



RMI y EJB

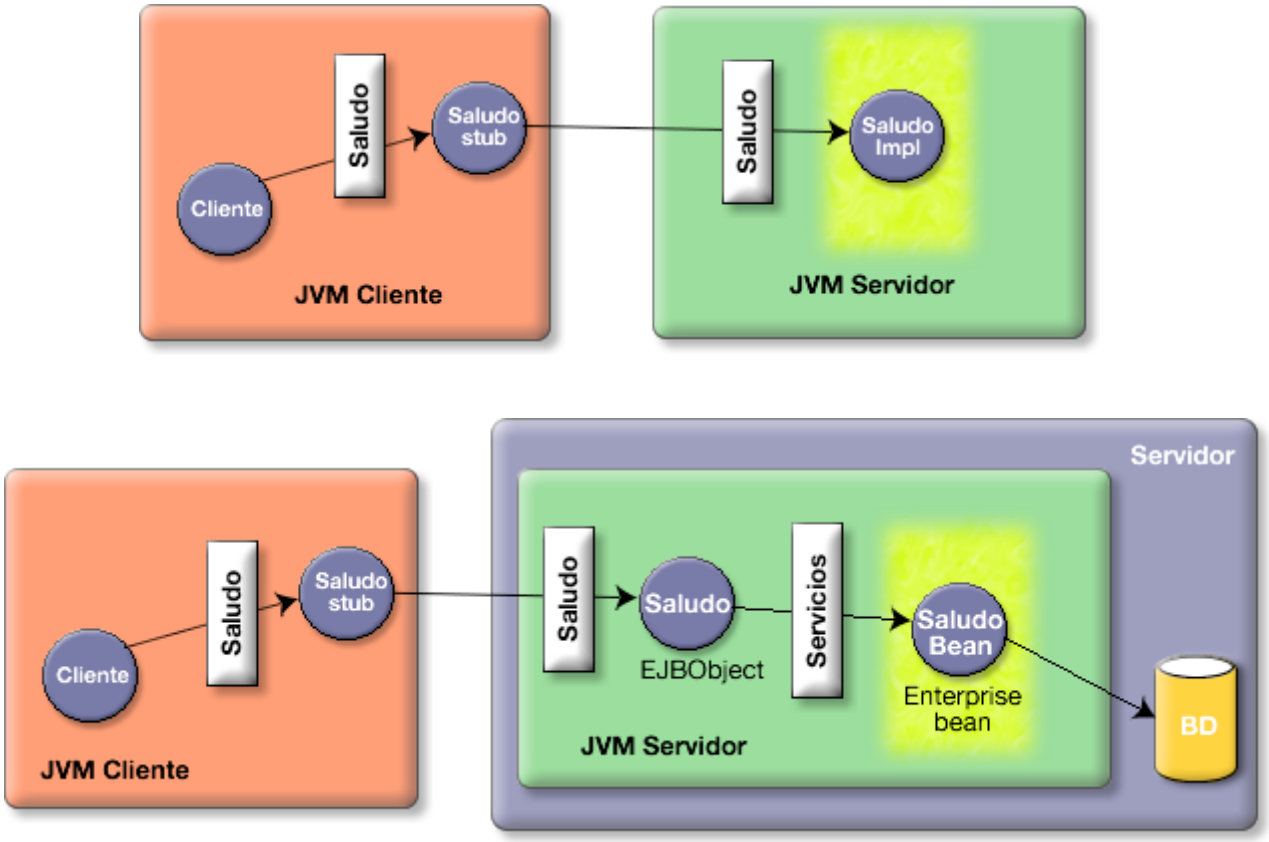
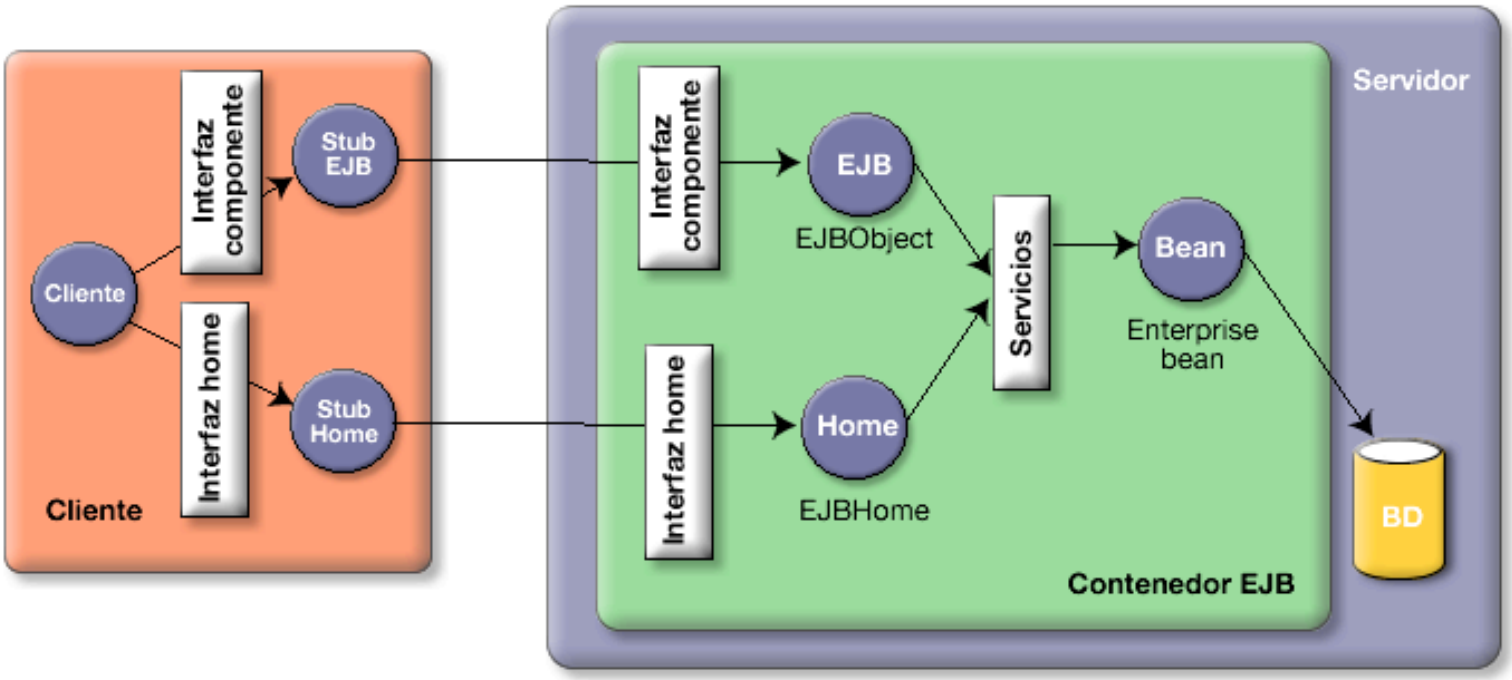
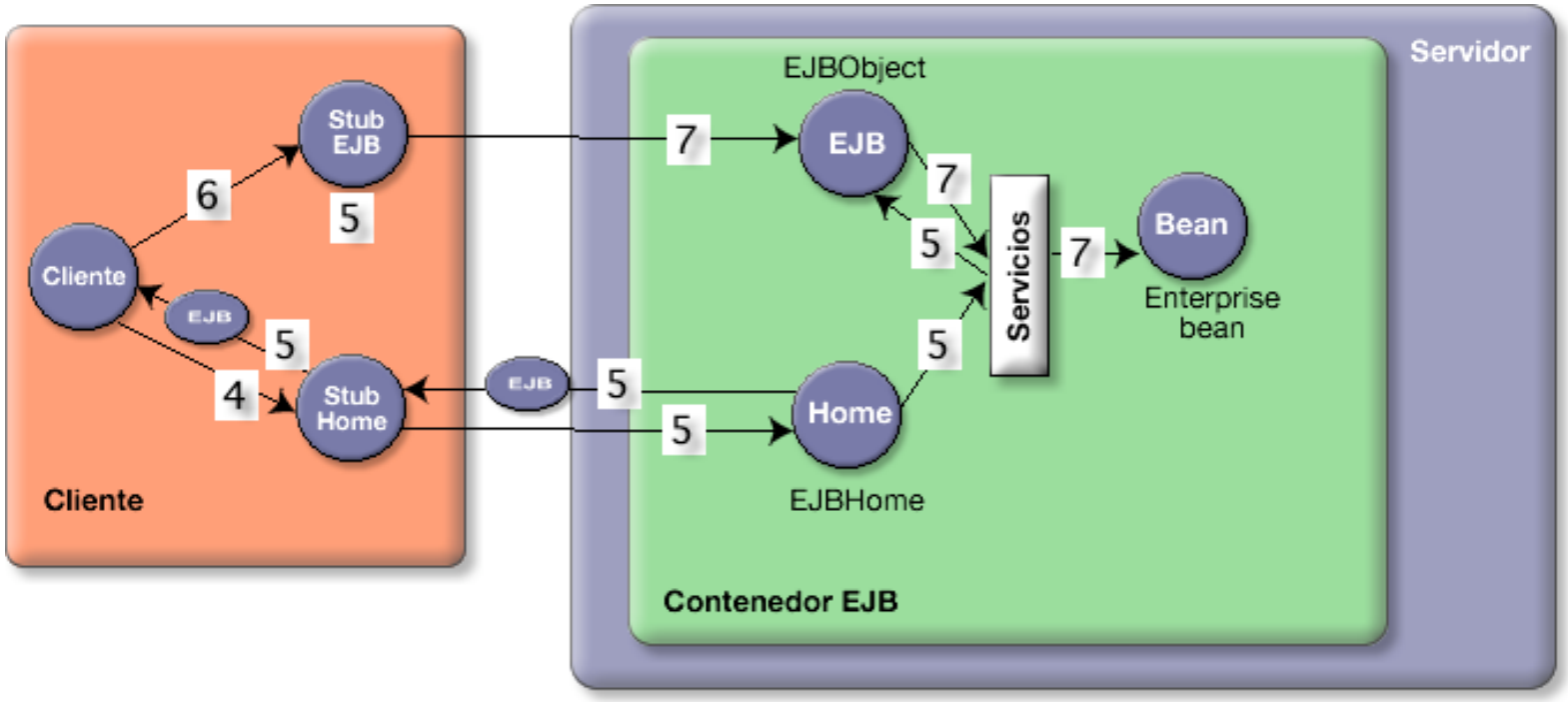

Cliente
Saludo
Saludo stub
JVM Cliente
Saludo
Saludo Impl
JVM Servidor
Cliente
Saludo
Saludo stub
JVM Cliente
Servidor
Saludo
Saludo
EJBObject
Servicios
Saludo Bean
Enterprise bean
JVM Servidor
BD

Figura completa de un enterprise bean



Funcionamiento de la clase Home



Beans de sesión

- Proporcionan un conjunto de funcionalidades y servicios a clientes
- No son persistentes, no representan datos existentes en un almacén de datos, aunque pueden acceder a ellos
- Modelan procesos de negocio, como solicitar un listado, enviar una notificación, ...
- Suelen recibir nombres como `ServicioBroker` o `GestorContratos`

Beans de sesión sin estado

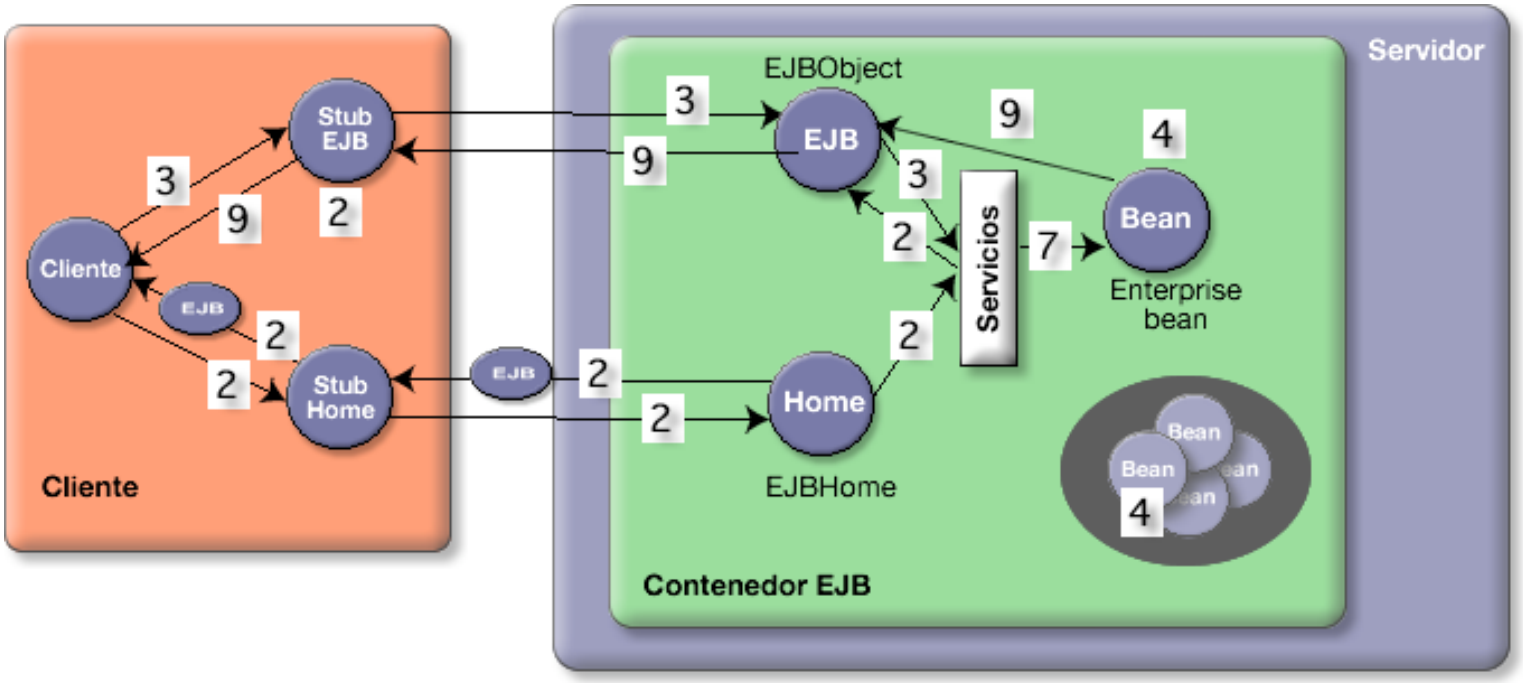
- Ejecutan una petición del cliente sin guardar ninguna información de la misma
- Ejemplos:

```
broker.compraAccion(accion)
calculator.calcPrestamo(prestamo)
```

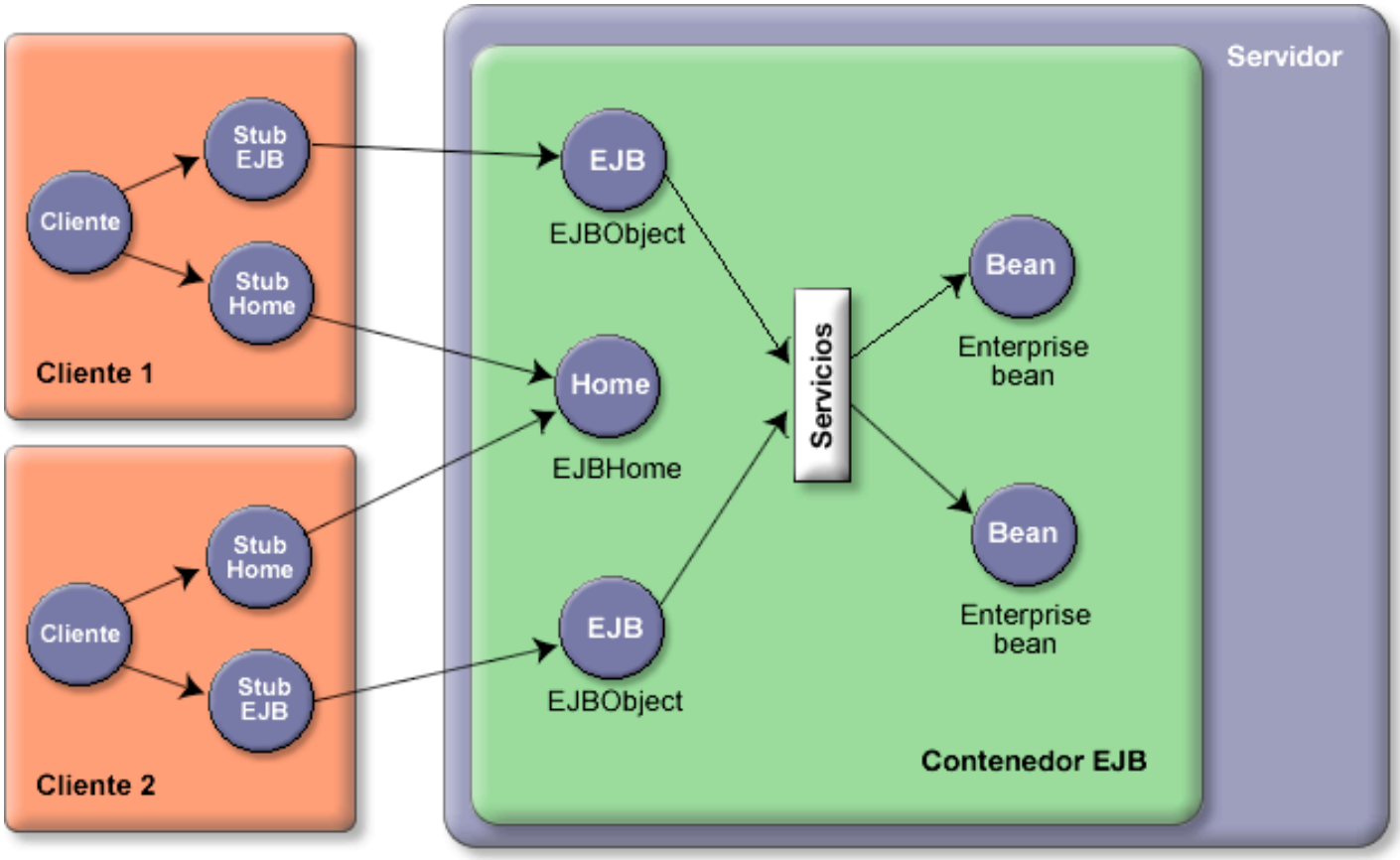
Beans de sesión en el contenedor

- Los beans de sesión son muy eficientes y ligeros
- El contenedor tiene una reserva (pool) de beans de sesión que va reusando según es necesario
- Un bean de sesión sin estado puede ser compartido por más de un cliente

Ciclo de vida de beans de sesión sin estado



Arquitectura de beans de sesión sin estado



Uso de los beans de sesión

- Para usar un bean de sesión un cliente debe:
  - Conseguir una instancia del bean llamando al método `create` de la interfaz home del bean
  - Llamar a los métodos de negocio de la interfaz componente del bean
  - Terminar el uso del bean llamando al método `remove` del bean

Uso de las anotaciones WebLogic

- WebLogic 9.2 proporciona una herramienta que simplifica mucho el desarrollo de los beans: EJBGen
- El programador define una única clase java (la clase de implementación) con anotaciones
- A partir de las anotaciones se generan automáticamente:
  - Las interfaces *home* y *componente* locales y remotas
  - Los descriptores de despliegue `ejb-jar.xml` y `weblogic-ejb-jar.xml`

Ejemplo: SaludoBean.java (1)

```
@Session(ejbName = "SaludoBean")
@JndiName(remote = "SaludoBean", local = "SaludoBeanLocal")
@FileGeneration(remoteClass = Constants.Bool.TRUE,
      remoteHome = Constants.Bool.TRUE,
      localClass = Constants.Bool.TRUE,
      localHome = Constants.Bool.TRUE,
      remoteClassName = "Saludo",
      remoteHomeName = "SaludoHome",
      localHomeName = "SaludoLocalHome",
      localClassName = "SaludoLocal")
public class SaludoBean extends GenericSessionBean implements SessionBean {

   private static final long serialVersionUID = 1L;
   private String[] saludos = { "Hola, que tal?", "Cuanto tiempo sin verte",
         "Que te cuentas?", "Me alegro de volver a verte" };

   public void ejbCreate() {
   }

   ... (sigue)
```

Ejemplo: SaludaBean.java (2)

```
    @RemoteMethod()
    @LocalMethod()
    public String saluda() {
       int random = (int) (Math.random() * saludos.length);
       return saludos[random];
    }

    @RemoteMethod()
    @LocalMethod()
    public SaludoTO getSaludo(int numDatos) {
       int random = (int) (Math.random() * saludos.length);
       String saludo = saludos[random];

       Date fecha = new Date();
       ArrayList<Integer> datos = new ArrayList<Integer>();

       for (int i = 0; i < numDatos; i++) {
          datos.add(i);
       }

       SaludoTO miSaludo = new SaludoTO(saludo, fecha, datos);
       return miSaludo;
    }
}
```

¿Preguntas?