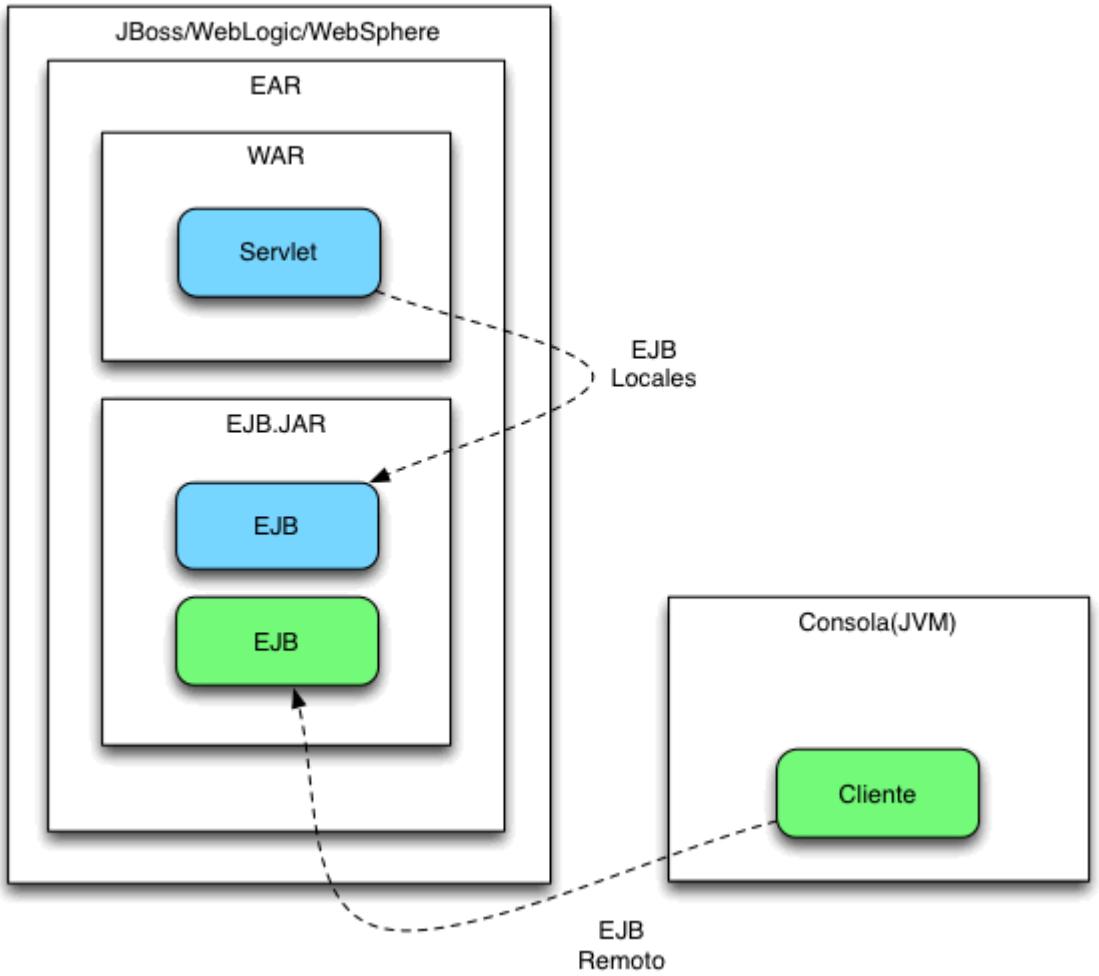
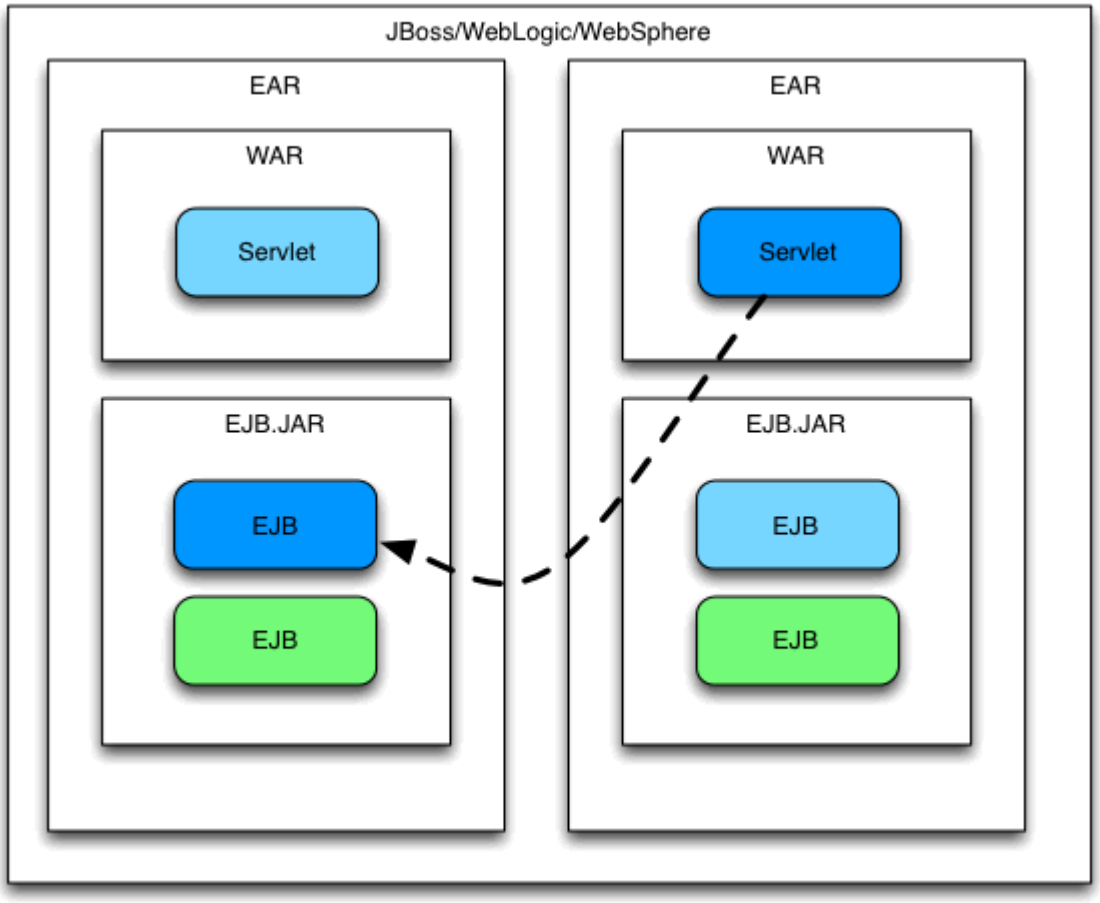
Ya hemos hablado en otros post sobre EJBs remotos y locales recalcando que los ejbs locales se acceden desde la misma maquina virtual y los remotos desde otra maquina virtual.



Es cierto que esta situación se produce habitualmente .Sin embargo no es la situación mas típica con la que nos podamos encontrar a nivel de uso de EJBs .Quizas la situación mas habitual es la de tener dos aplicaciones (EARs) y que desde la aplicación A haya que acceder a un EJB de la aplicación B.



Ahora bien este acceso se produce desde la misma maquina virtual pero en distintos EARs . ¿Se trata de un acceso local o de un acceso remoto?. La respuesta no es evidente .En principio se trataría de un acceso local ya que nos encontramos en la misma maquina virtual .

EJB Remove vs Local

Lamentablemente esto no tiene porque ser así siempre ya que la especificación unicamente obliga a que el interface de acceso local este disponible para los clientes que se encuentran ubicados en el mismo EAR . No especifica el acceso desde otros EARs . Por lo tanto la respuesta dependerá de cada servidor de aplicaciones. Si queremos asegurarnos de que el acceso funciona en cualquier servidor de aplicaciones deberemos construir un EJB con acceso remoto.

```
package com.arquitecturajava.ejb;

import javax.ejb.Remote;

@Remote
public interface HolaEJBRemote {
 public String mensaje();
}
```

```
package com.arquitecturajava.ejb;

import javax.ejb.Stateless;
@Stateless
public class HolaEJB implements HolaEJBRemote {
 public HolaEJB() {
 // TODO Auto-generated constructor stub
 }

@Override
 public String mensaje() {
 // TODO Auto-generated method stub
 return "hola peticion remota a un ejb";
 }

}
```

Creado el EJB remoto podremos acceder a el de una forma sencilla a traves de por ejemplo

un servlet y utilizando la anotación de @EJB .Eso si deberemos apoyarnos en los conocimientos adquiridos en el post anterior de "Portable JNDI" y utilizar la anotación @EJB con el atributo de lookup. Vamos a ver un servlet que accede al EJB.

```
package com.arquitecturajava;

import java.io.IOException;
import java.io.PrintWriter;

import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.arquitecturajava.ejb.HolaEJBRemote;
@WebServlet("/ServletAccesoRemoto")
public class ServletAccesoRemoto extends HttpServlet {
 private static final long serialVersionUID = 1L;

 @EJB(lookup="java:global/EARHolaMundoRemoto/EJBRemote/HolaEJB")
 HolaEJBRemote miejb;

 public ServletAccesoRemoto() {
 super();
 // TODO Auto-generated constructor stub
 }
 protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
```

```
 PrintWriter pw= response.getWriter();
 pw.println(miejb.mensaje());
 pw.close();

 }

/**
 * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
 */
 protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
 // TODO Auto-generated method stub
 }

}
```

Podemos ver como la anotación referencia al EJB que queremos acceder .

```
@EJB(lookup="java:global/EARHolaMundoRemoto/EJBRemote/HolaEJB")
 HolaEJBRemote miejb;
```

Solo nos queda invocar el servlet y ver el resultado .




```
hola peticion remota a un ejb
```