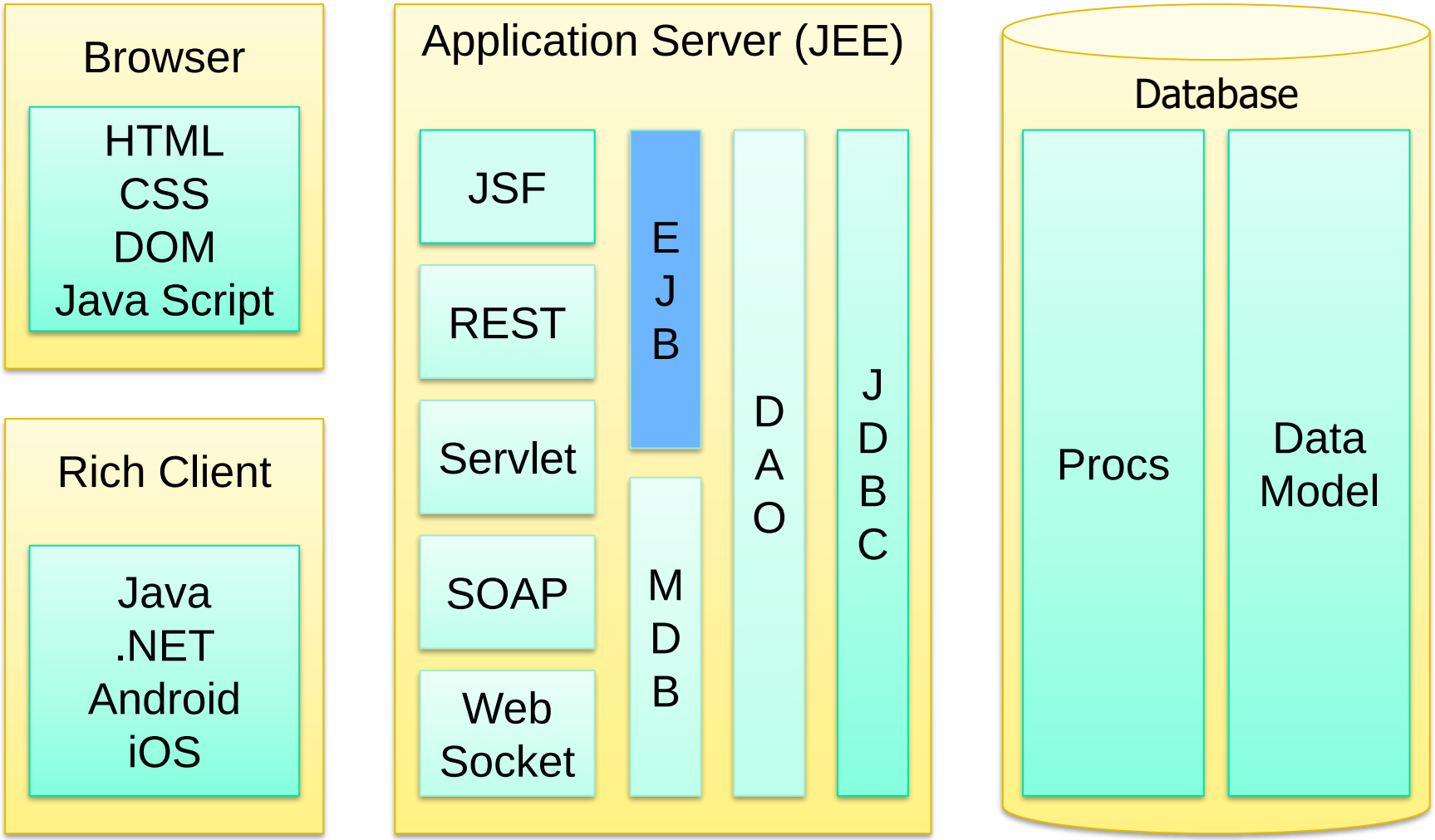
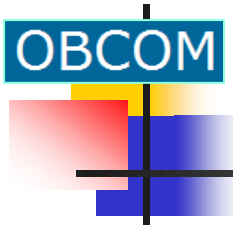

# Aplicaciones Web (parte 5)

Eduardo Ostertag Jenkins, Ph.D.
OBCOM INGENIERIA S.A. (Chile)
Eduardo.Ostertag@obcom.cl

# Enterprise Java Beans: EJB

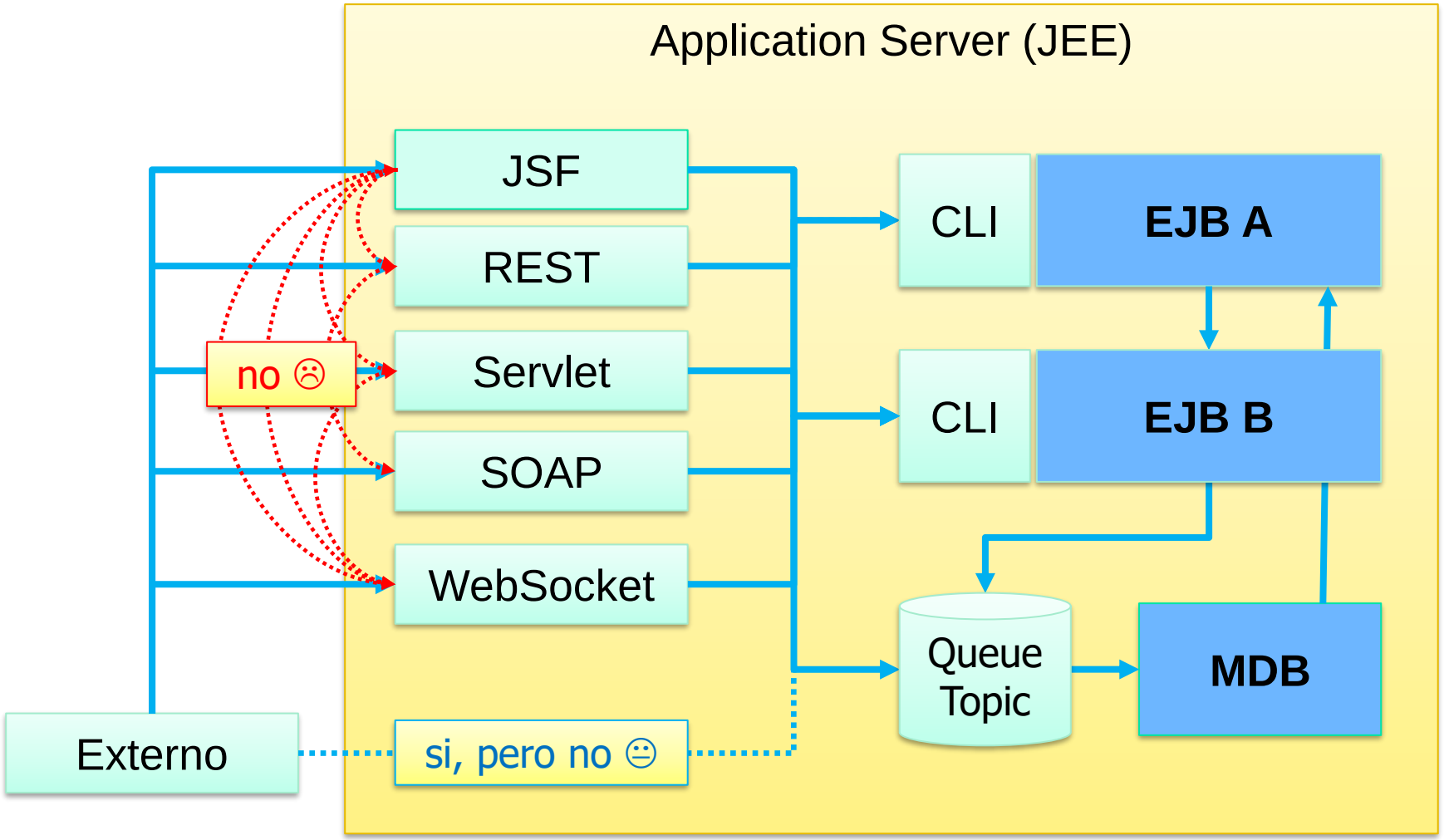


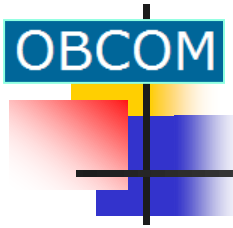



# Tipos de Enterprise Beans

- **Session** Enterprise Beans
  - **Statefull**: implementan diálogos 😐
  - **Stateless**: no mantienen estado ☺
  - **Singleton**: una instancia (timers) ☺
- **Message-driven** Enterprise Beans
  - Procesamiento asíncrono de mensajes
  - Mensajes se despachan vía colas JMS
  - Los MDB no son llamados directamente

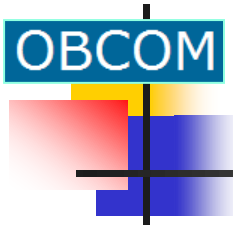
# ¿Quién invoca a quién?





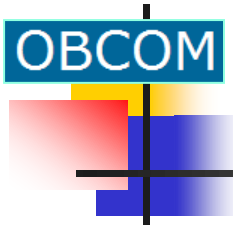

# Stateless Enterprise Beans

- Manejan las transacciones y la seguridad
  - Begin, Commit y Rollback de transacciones
  - Las que incluyen operaciones JDBC, JMS, etc.
- Se debe definir una "Bussiness Interface"
  - @javax.ejb.**Local** (sólo dentro de la app)
  - @javax.ejb.**Remote** (invocable por todos)
  - `public interface SampleRemote {…}`
- Requiere una clase "Stateless EJB"
  - @javax.ejb.**Stateless**
  - `public class MyBean implements MyAPI {…}`



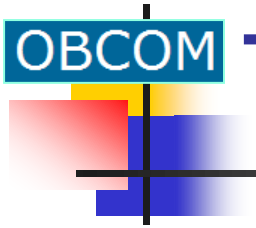

# Interfaz de un Stateless EJB

```
@Remote
public interface SampleEjbRemote
{
    BuscarPorIdResult buscarPorId(BuscarPorIdRequest args)
        throws SampleEjbException;

    BuscarPorNombreResult buscarPorNombre(BuscarPorNombreRequest args)
        throws SampleEjbException;

    BuscarPorRutResult buscarPorRut(BuscarPorRutRequest args)
        throws SampleEjbException;

    ...
}
```




# Clase de un Stateless EJB

```
@Stateless(name="SampleEjb")
@TransactionAttribute(TransactionAttributeType.REQUIRED)
public class SampleEjb implements SampleEjbRemote
{
    @Resource(name="SampleDataSource")
    private DataSource dataSource;

    @Override
    public BuscarPorIdResult buscarPorId(BuscarPorIdRequest args)
        throws SampleEjbException
    {
        try {
            return BuscarPorId.execute(dataSource, args);
        } catch (Exception ex) {
            throw new SampleEjbException(ex);
        }
    }

    ...
}
```




# Tipo transaccional de un EJB

- **REQUIRED**: utiliza la transacción vigente, o crea una transacción nueva.
- **REQUIRES_NEW**: siempre crea una transacción nueva, independiente si ya existe una vigente.
- **MANDATORY**: debe existir una transacción vigente. Es una error invocar sin una transacción.
- **NOT_SUPPORTED**: ejecuta fuera del contexto de la transacción vigente, si existe alguna.
- **SUPPORTS**: ejecuta con la transacción vigente, o sin una transacción, si no hay una vigente.
- **NEVER**: siempre ejecuta fuera de una transacción. Es un error invocar con una transacción.

# Excepción de un Stateless EJB

```
@ApplicationException(rollback = true)
public class SampleEjbException extends Exception
{
    private static final long serialVersionUID = 1L;

    public SampleEjbException()
    {
        super();
    }

    public SampleEjbException(String message)
    {
        super(message);
    }

    public SampleEjbException(Throwable cause)
    {
        super(cause);
    }

    public SampleEjbException(String message, Throwable cause)
    {
        super(message, cause);
    }
}
```

# Archivo ejb-jar.xml estándar

META-INF/ejb-jar.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ejb-jar ...>

  <module-name>SampleEjb</module-name>

  <enterprise-beans>
    <session>
      <ejb-name>SampleEjb</ejb-name>
      <resource-ref>
        <res-ref-name>SampleDataSource</res-ref-name>
        <res-type>javax.sql.DataSource</res-type>
        <res-auth>Container</res-auth>
        <res-sharing-scope>Shareable</res-sharing-scope>
      </resource-ref>
    </session>
  </enterprise-beans>

  <ejb-client-jar>SampleEjbClient.jar</ejb-client-jar>

</ejb-jar>
```

# Glassfish: glassfish-ejb-jar.xml

META-INF/glassfish-ejb-jar.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<glassfish-ejb-jar>

  <enterprise-beans>
    <ejb>
      <ejb-name>SampleEjb</ejb-name>
      <resource-ref>
        <res-ref-name>SampleDataSource</res-ref-name>
        <jndi-name>jdbc/NombreDeAlgunDataSource</jndi-name>
      </resource-ref>
    </ejb>
  </enterprise-beans>

</glassfish-ejb-jar>
```

# Wildfly: jboss-ejb3.xml

META-INF/jboss-ejb3.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss:ejb-jar ...>

  <enterprise-beans>
    <session>
      <ejb-name>SampleEjb</ejb-name>
      <resource-ref>
        <res-ref-name>SampleDataSource</res-ref-name>
        <lookup-name>java:/jdbc/NombreDeAlgunDataSource</lookup-name>
      </resource-ref>
    </session>
  </enterprise-beans>

</jboss:ejb-jar>
```

# WebLogic: weblogic-ejb-jar.xml

META-INF/weblogic-ejb-jar.xml

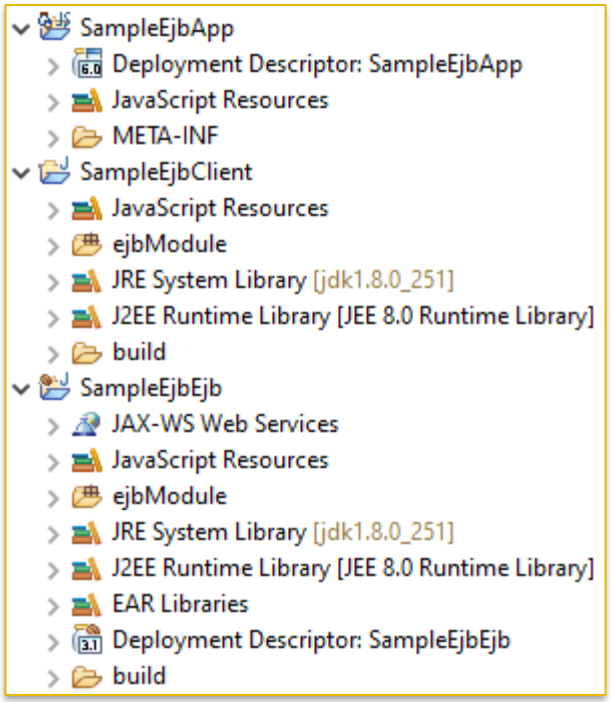
```
<?xml version="1.0" encoding="UTF-8"?>
<weblogic-ejb-jar ...>

  <weblogic-enterprise-bean>
    <ejb-name>SampleEjb</ejb-name>
    <resource-description>
      <res-ref-name>SampleDataSource</res-ref-name>
      <jndi-name>jdbc/NombreDeAlgunDataSource</jndi-name>
    </resource-description>
    <enable-call-by-reference>true</enable-call-by-reference>
  </weblogic-enterprise-bean>

</weblogic-ejb-jar>
```

# Projecto cliente de un EJB

```
<ejb-client-jar>SampleEjbClient.jar</ejb-client-jar>
```

ejb-jar.xml

- EAR (**Enterprise Application Proyect**)
  - Empaqueta el JAR del Stateless EJB
  - Empaqueta el JAR para clientes del EJB
- Client JAR (**Java Proyect**)
  - Contiene las clase que necesita el EJB…
  - …y que necesitan los que invocan al EJB
  - Como la **Interfaz** y la **Excepción** del EJB…
  - …y otras clases requeridas (como **DTO**s)
- EJB JAR (**EJB Proyect**)
  - Contiene la lógica del Stateless EJB



# Invocar a un Stateless EJB

```
@WebService
public class SampleEjbWeb
{
    @EJB
    private SampleEjbRemote ejb;

    @WebMethod
    public BuscarPorIdResult buscarPorId(BuscarPorIdRequest args)
        throws SampleEjbException
    {
        ...
        BuscarPorIdResult result = ejb.buscarPorId(args);
        ...
    }

    ...
}
```

Injección

```
        ...
        SampleEjbRemote ejb = InitialContext.doLookup("…");
        BuscarPorIdResult result = ejb.buscarPorId(args);
        ...
```

Dinámico

# Muchas gracias

Muchas gracias