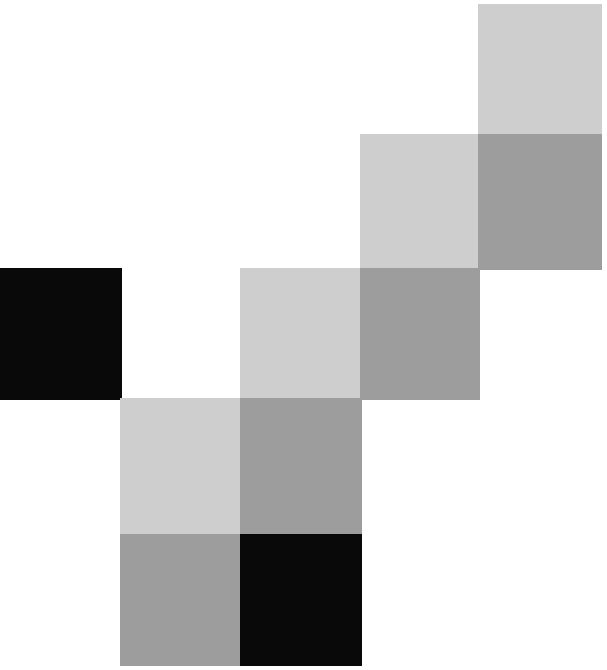


J2EE

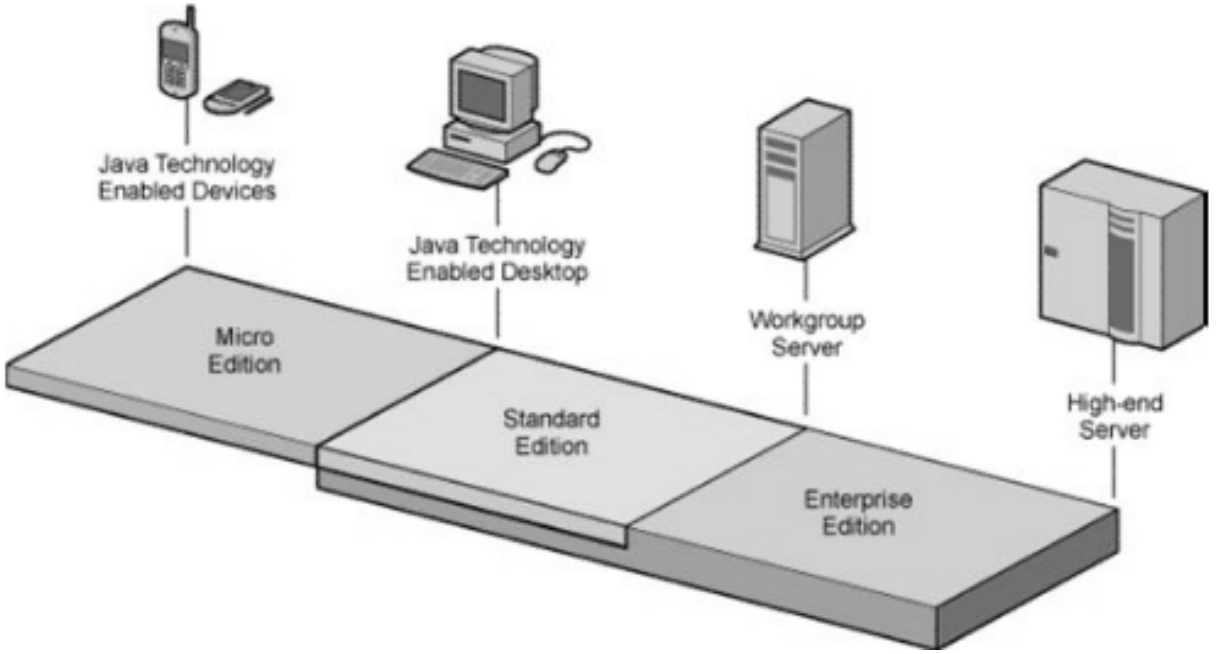
Apresentado por Nuno Nunes

J2EE

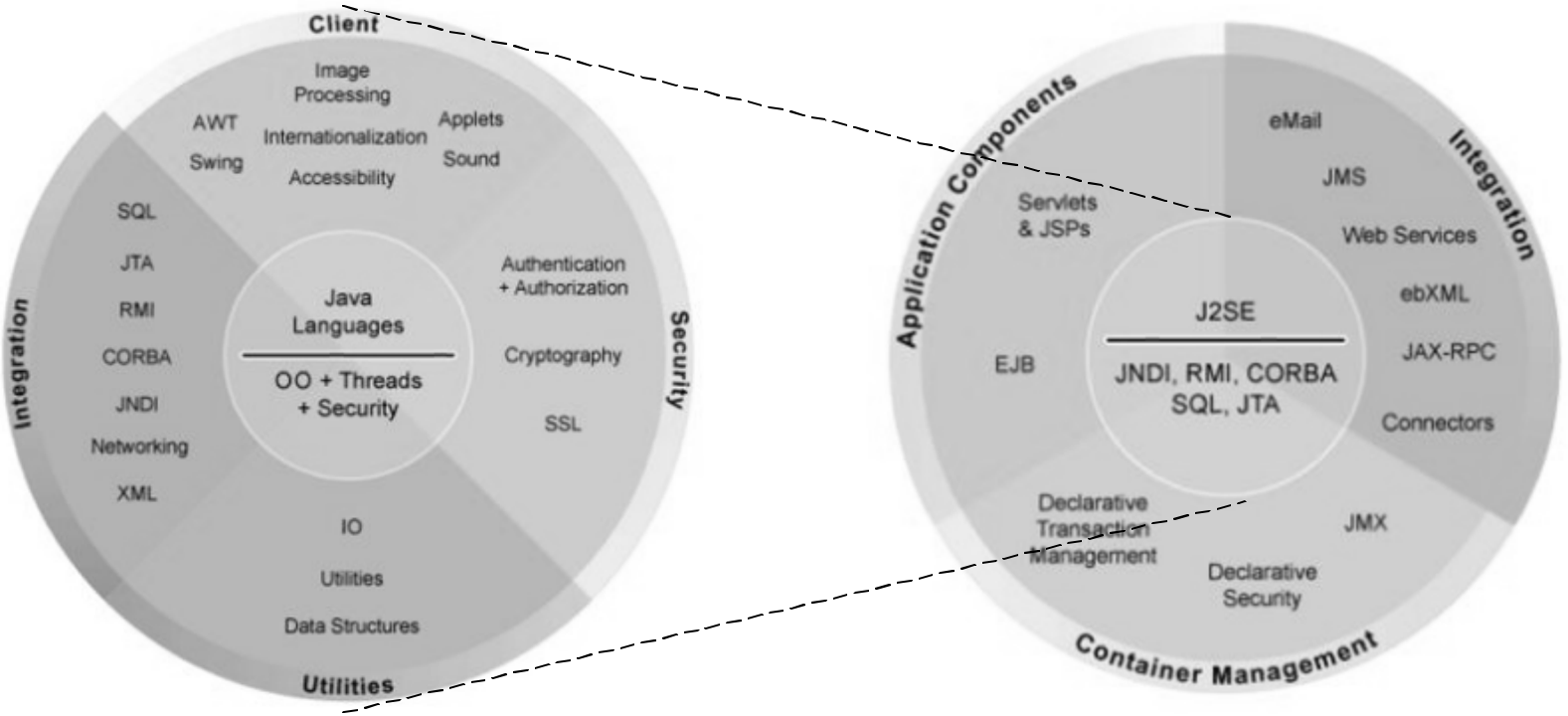
- Tecnologia Java, “A visão”
  - O J2EE
  - Resumo da tecnologia J2EE
- Os componentes do J2EE
  - Do lado do cliente...
  - Do lado do servidor – Tecnologias de apresentação
  - Do lado do servidor – Lógica do negócio
  - Integração
- J2EE versus .net

Tecnologia Java, “A visão”

- J2SE – Java Standard Edition
- J2ME – Java Micro Edition
- J2EE – Java Enterprise Edition



Tecnologia Java, “A visão”



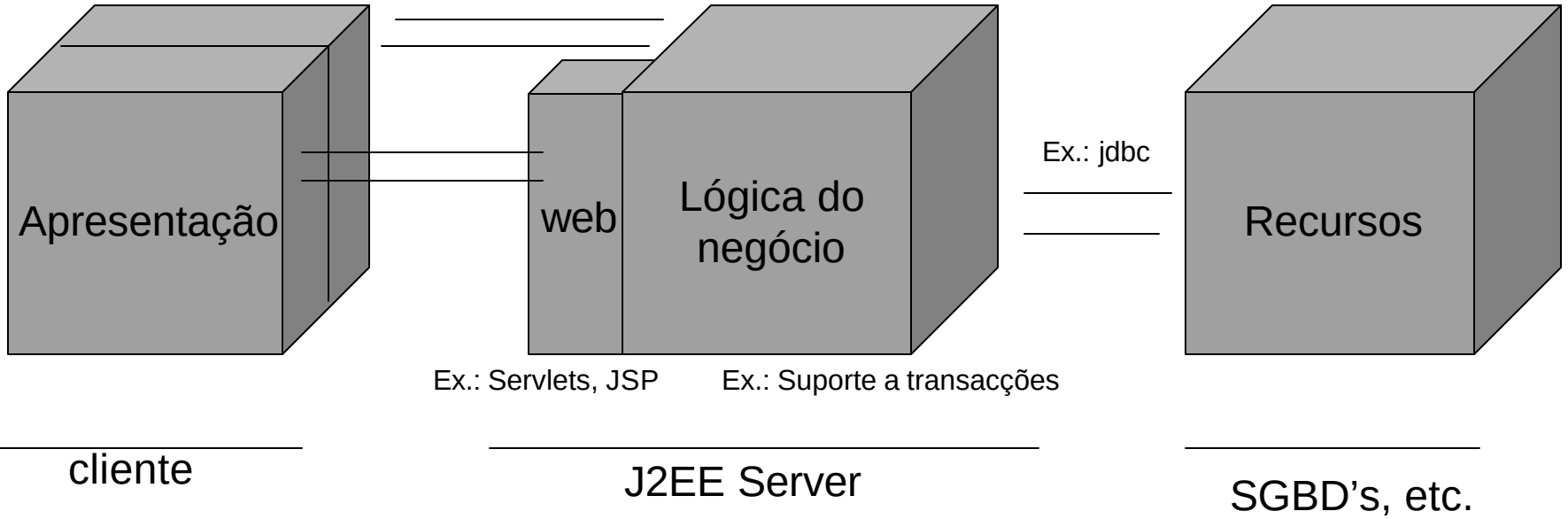
J2SE J2EE

Tecnologia Java, “A visão”

É importante perceber...

...O J2EE não é um produto, mas uma especificação....

Tecnologia Java, “A visão” J2EE



- O J2EE é um enriquecimento do J2SE visando oferecer serviços complementares, importantes, nos ambientes aplicacionais existentes actualmente

Tecnologia Java, “A visão” Resumo (APIs J2EE v1.4)

- **Enterprise Java Beans**
- **Java Servlet / Java Server Pages**
- **JMS**
- **Java Transaccion API**
- JavaMail
- Java API for XML processing (JAXP)
- Java API – XML Based RPC
- SOAP with Atachments API
- JAVA API for XML Registries
- J2EE connector Architecture
- **JDBC API**
- Java Naming and Directory Interface
- Java Authentication and Authorization service

Os componentes do J2EE

Do lado do cliente...

- Interacção através de uma aplicação cliente
- Interacção via equipamentos móveis (telemóveis, PDA’s)
- **Interacção via Web**

Os componentes do J2EE

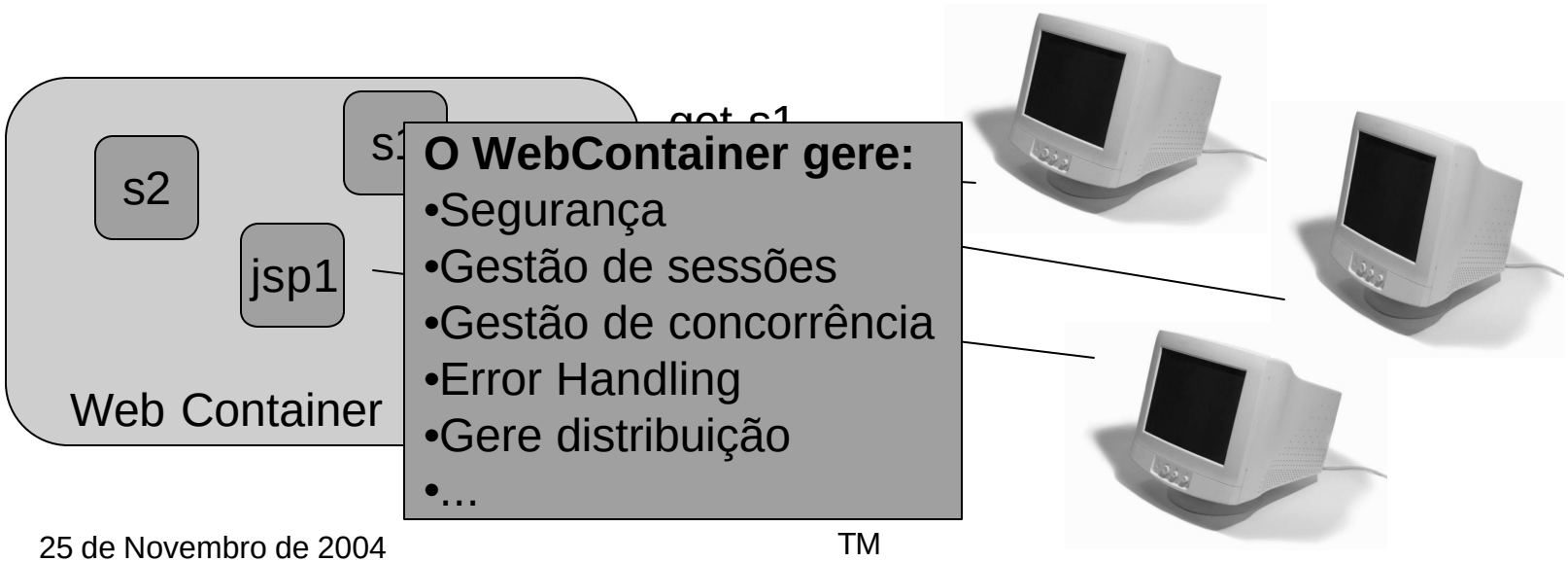
Do lado do servidor - Tecnologias de apresentação

- Porquê dar importância à Web?
  - Os web browsers são ubíquos
  - É muito simples fazer o upgrade de uma aplicação
  - Os servidores web são extremamente escaláveis
- As tecnologias existentes (CGI's, ASP / PHP, etc.) não chegam?
  - São computacionalmente dispendiosos
  - São "sujos" na forma com lidam com os dados (misturam apresentação com lógica do negócio)

Os componentes do J2EE

Do lado do servidor - Tecnologias de apresentação

- A resposta apresentada no J2EE é constítuida por 2 tecnologias distintas:
  - Servlets
  - JSP's



Os componentes do J2EE

Do lado do servidor - Tecnologias de apresentação

```
//Servlet HelloWorld

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {

        public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {

                response.setContentType("text/html");
                PrintWriter out = response.getWriter();
                out.println("<html>");
                out.println("<body>"); out.println("<head>");        out.println("<title>HelloWorld!</title>");
                out.println("</head>"); out.println("<body>");
                out.println("<h1>Hello World!</h1>");
                out.println("</body>");
                out.println("</html>");
        }
}
```

Os componentes do J2EE

Do lado do servidor - Tecnologias de apresentação

```
<HTML>
<HEAD>
    <TITLE>hello jsp</TITLE>
    <%! String message = "Hello, World, from JSP"; %>
</HEAD>
<BODY>
<h2><font color="#AA0000"><%= message%></font></h2>
<h3>
    <font color="#AA0000">
        <%= new java.util.Date() %>
    </font>
</h3>
</BODY>
</HTML>
```

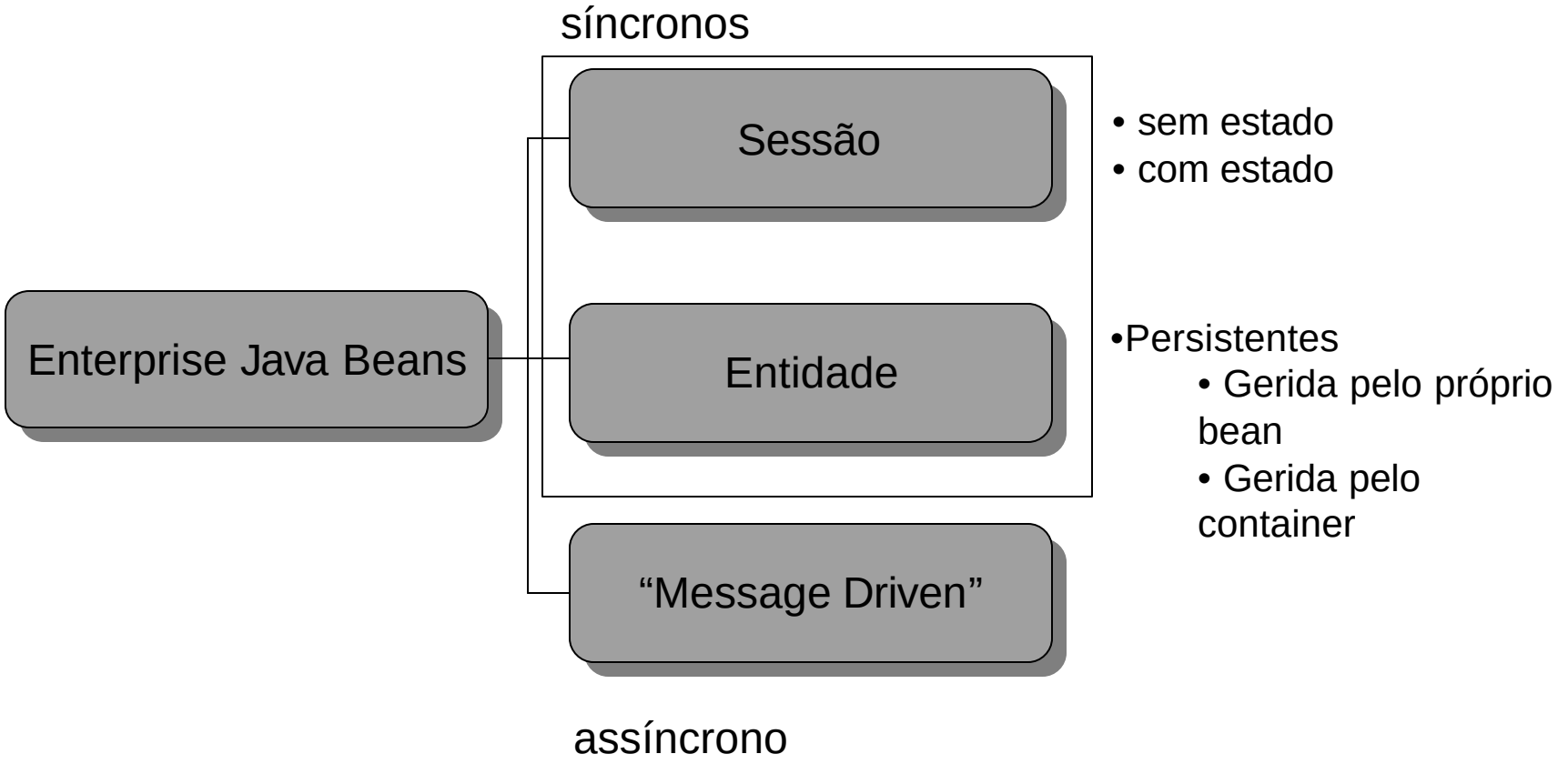
Os componentes do J2EE

Do lado do servidor – Lógica de negócio

- Lógica do negócio
  - EJB (Enterprise Java Beans)
    - Gestão de transacções
    - Segurança
    - Gestão de recursos

Os componentes do J2EE

Do lado do servidor – Lógica de negócio



Os componentes do J2EE

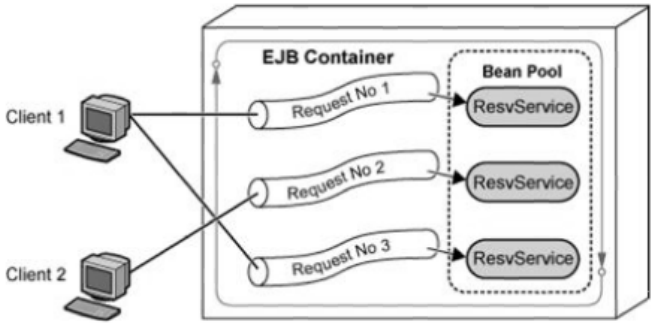
Do lado do servidor – Lógica de negócio

- Session beans (sessão)
  - Representar processos de negócio
    - Ex.: como receber um pagamento (interagir com)

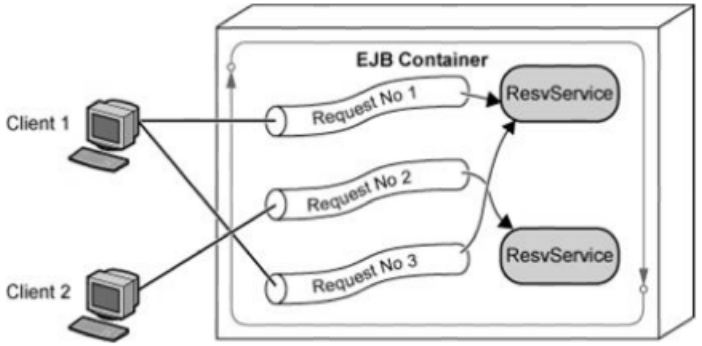
Os componentes do J2EE

Do lado do servidor – Lógica de negócio

- Com estado ou sem estado?
  - Stateless session beans
    - Cada pedido é independente
    - É por inerência mais escalável



  - Statefull session beans
    - Usado qundo é necessário responder a sequências de pedidos
    - Cada session bean, apenas responde a um cliente durante uma sessão (implica que o cliente informe quando termina a sessão)



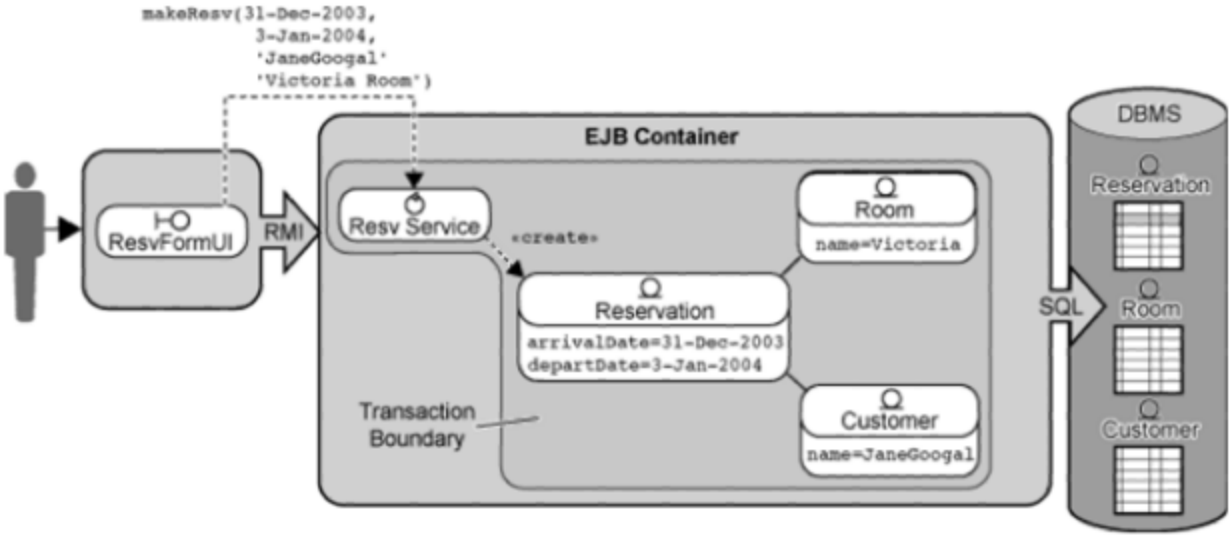
Os componentes do J2EE

Do lado do servidor – Lógica de negócio

- Entity beans (entidades)
  - Representar os dados do negócio (ex.: factura, cliente, items em stock)
- Porquê os Entity beans, não podemos usar um SGBD?
  - A ideia por trás dos Entity beans é simular um sistema “perfeito”
    - Fundamentalmente esconde os SGBD’s,
    - O sistema nunca perde dados e simula memória infinita
    - O programador só tem de usar dados em memória
    - Encontra-se sob um ambiente transaccional
  - Os beneficios são:
    - O “EJB container” controla todo o ambiente transaccional
    - O “EJB container” pode tratar de toda a preservação de dados no SGBD
    - O SGBD utilizado é completamente transparente para o programador / portabilidade no meio de armazenamento
    - Optimizações geridas pelo EJB

Os componentes do J2EE

Do lado do servidor – Lógica de negócio



Os componentes do J2EE

Do lado do servidor – Lógica de negócio

- Message driven bean
  - Criados para responder a sistemas legados
  - Criados para responder a sistemas assíncronos (ex.: JMS)
    - Respondem a mensagens/não gera mensagens

Os componentes do J2EE

Do lado do servidor – Lógica de negócio

```
[user@host app]# mkdir org
[user@host app]# mkdir org/acme
```

HelloBean.java

```
package org.acme;

import java.rmi.RemoteException;
import javax.ejb.*;

public class HelloBean implements SessionBean {
  private SessionContext sessionContext;
  public void ejbCreate() {
  }
  public void ejbRemove() {
  }
  public void ejbActivate() {
  }
  public void ejbPassivate() {
  }
  public void setSessionContext(SessionContext sessionContext) {
    this.sessionContext = sessionContext;
  }
  public String sayHello() throws java.rmi.RemoteException {
    return "Hello World!!!!!";
  }
}
```

Os componentes do J2EE

Do lado do servidor – Lógica de negócio

HelloHome.java

```
package org.acme;

import java.rmi.*;
import javax.ejb.*;
import java.util.*;

public interface HelloHome extends EJBHome {
  public HelloObject create() throws RemoteException, CreateException;
}
```

Os componentes do J2EE

Do lado do servidor – Lógica de negócio

HelloObject.java

```
package org.acme;

import java.rmi.*;
import javax.ejb.*;
import java.util.*;

public interface HelloObject extends EJBObject {
  public String sayHello() throws RemoteException;
}
```

Os componentes do J2EE

Do lado do servidor – Lógica de negócio

META-INF\ejb-jar.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ejb-jar>
  <enterprise-beans>
    <session>
      <ejb-name>Hello</ejb-name>
      <home>org.acme.HelloHome</home>
      <remote>org.acme.HelloObject</remote>
      <ejb-class>org.acme.HelloBean</ejb-class>
      <session-type>Stateless</session-type>
      <transaction-type>Container</transaction-type>
    </session>
  </enterprise-beans>
  <assembly-descriptor>
    <container-transaction>
      <method>
        <ejb-name>Hello</ejb-name>
        <method-name>*</method-name>
      </method>
      <trans-attribute>Required</trans-attribute>
    </container-transaction>
  </assembly-descriptor>
</ejb-jar>
```

Os componentes do J2EE

Do lado do servidor – Lógica de negócio

HelloWorld.java

```
package org.acme;

import javax.rmi.*;
import javax.naming.*;
import java.util.*;

public class HelloWorld {

 public static void main( String args[]) {
  try{

    Properties p = new Properties();

    //The JNDI properties you set depend
    //on which server you are using.
    //These properties are for the Remote Server.
    p.put("java.naming.factory.initial", "org.openejb.client.RemoteInitialContextFactory");
    p.put("java.naming.provider.url", "127.0.0.1:4201");
    p.put("java.naming.security.principal", "myuser");
    p.put("java.naming.security.credentials", "mypass");

    //Now use those properties to create
    //a JNDI InitialContext with the server.
    InitialContext ctx = new InitialContext( p );
```

Os componentes do J2EE

Do lado do servidor – Lógica de negócio

```
    //Lookup the bean using it's deployment id
    Object obj = ctx.lookup("/Hello");

    //Be good and use RMI remote object narrowing
    //as required by the EJB specification.
    HelloHome ejbHome = (HelloHome)
        PortableRemoteObject.narrow(obj,HelloHome.class);

    //Use the HelloHome to create a HelloObject
    HelloObject ejbObject = ejbHome.create();

    //The part we've all been wainting for...
    String message = ejbObject.sayHello();

    //A drum roll please.
    System.out.println( message );

  } catch (Exception e){
    e.printStackTrace();
  }
 }
}
```

J2EE, Integração

- Como integrar com um SGBD?
- Como integrar com um sistema legado?
- Como integrar sistemas entre negócios (B2B)

J2EE, Integração

- Como integrar com SGBD’s
  - JDBC
    - API para Integrar com SGBD’s
  - JTA
    - API para gerir transacções

J2EE, Integração

- Sistemas legados
  - Java Connector Integration
    - Permite fazer chamadas a procedimentos a aplicações noutras linguagens (tipicamente C ou Cobol)
  - JNI (Java Connector)
    - Especifica uma interface. É possível comprar adaptadores para integrar com outras aplicações
  - CORBA

J2EE, Integração

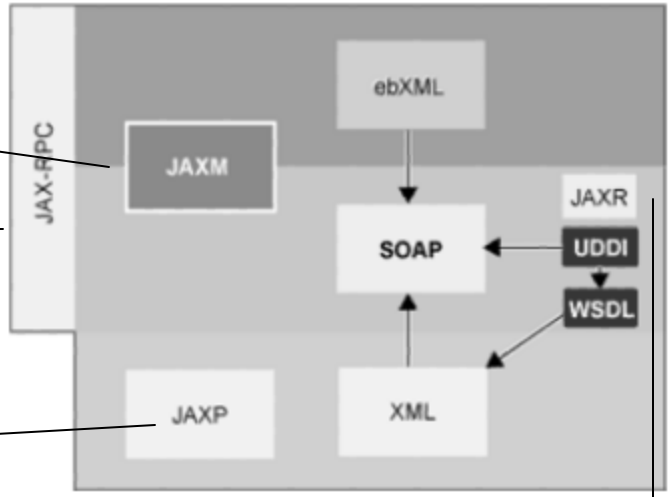
- Business integration (B2B)
  - WebServices, RMI ou CORBA

J2EE, Integração WebServices

Composição e Parsing de mensagens SOAP

“Compile time utilities”
Cria os stubs usados pelos webservices

Processador de XML
via SAX ou DOM

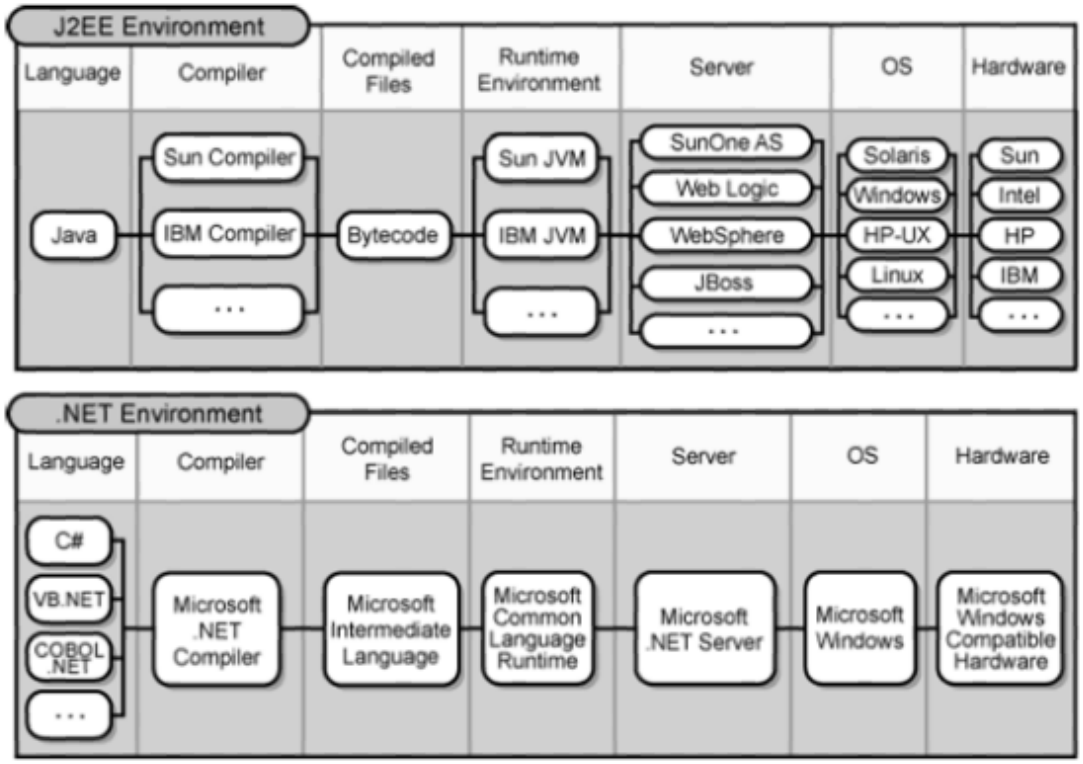


Interface com o sistema UDDI
(publicar e pesquisar webservices)

J2EE versus .net

	J2EE	.NET
Especificação ou produto	Especificação	Produto
Portabilidade do vendedor	30 vendedores	Microsoft ?
Portabilidade no S.O.	Qualquer S.O. que suporte a JVM	Apenas em O.S. Microsoft ?
Linguagens	JAVA	C#, VB, VC++, etc

J2EE versus .net



J2EE versus .net

	J2EE	.net
Acesso a base de dados	JDBC	ADO.net
Serviços de directoria	JNDI	ADO.net
XML Parsing	JAXP (SAX & DOM)	MSXML (SAX & DOM)
WebServices	JAXM (SOAP, ebXML) JAXR (WSDL, UDDI) JAX-RPC	SOAP, WSDL, UDDI
Integração com software legado	JNI, CORBA, JMS e Java Connector Architecture	MS Host Integration Service
Segurança	J2SE (core security), JAAS, J2EE – specific	Passport .net