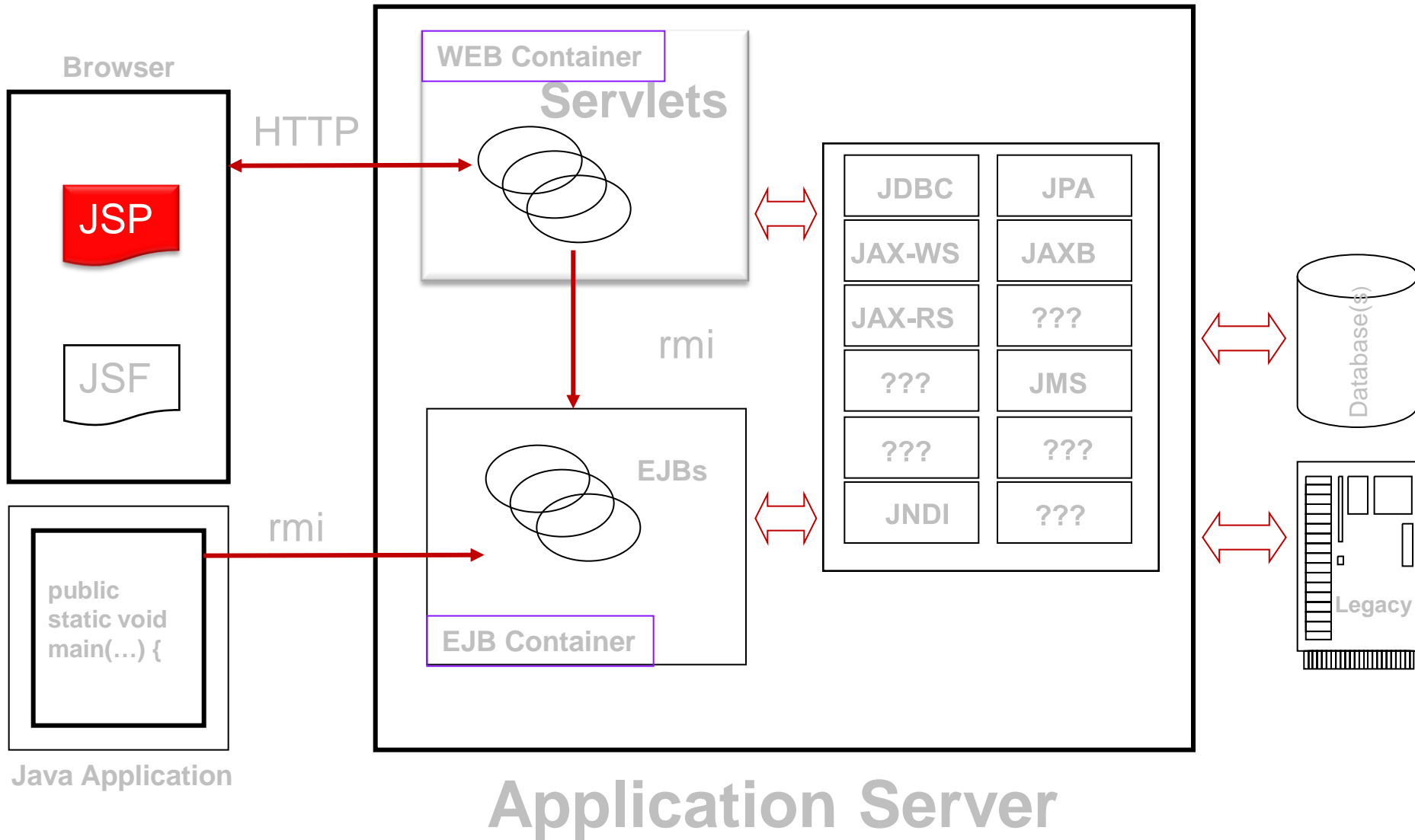


# ***Java Enterprise Edition***

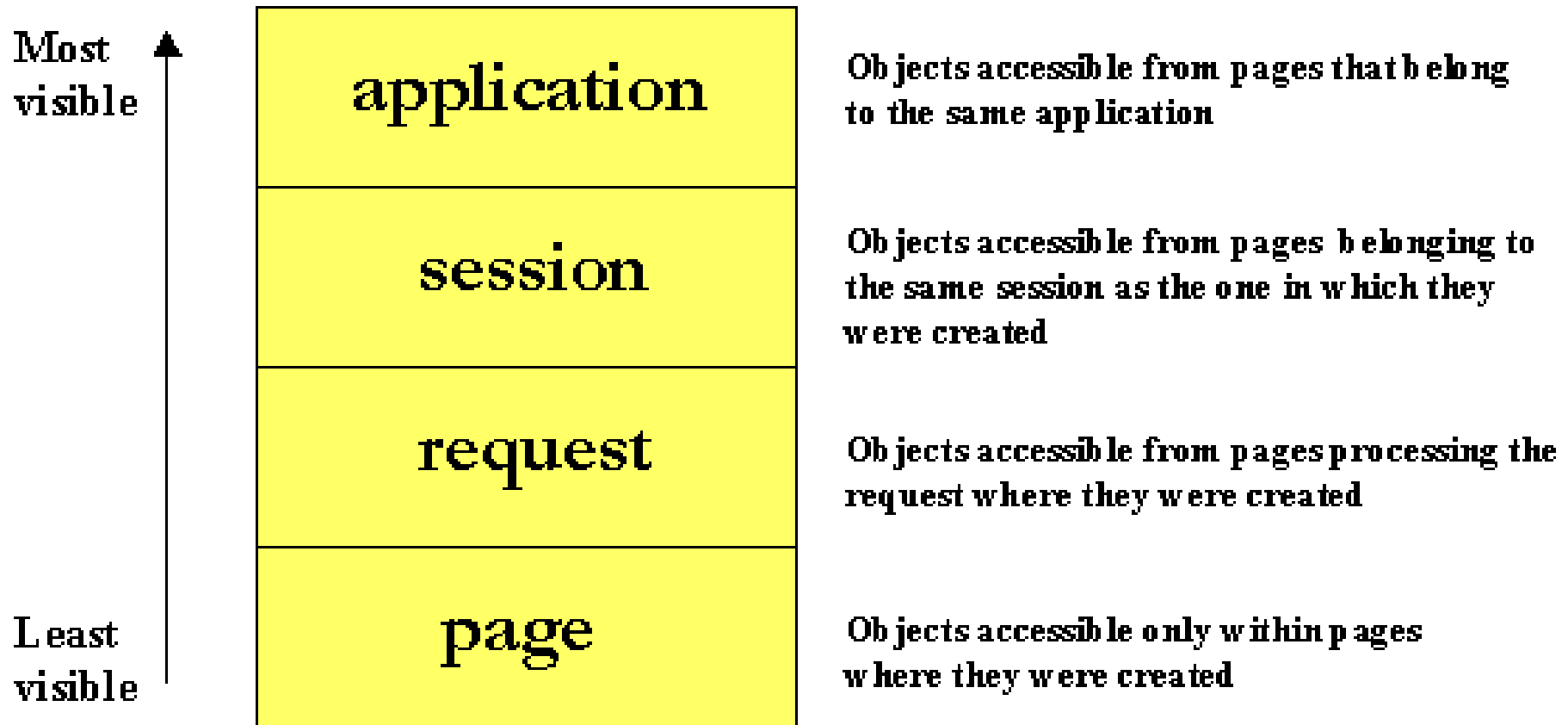


# Java EE APIs – The big picture

Focus on JSPs



# Scopes (First look)



▶ <Your Project>/WEB-INF/jsp/hello.jsp

```
<%@page import="java.util.Date"%>
```

```
<html>
```

```
  <head><title>Hello</title></head>
```

```
  <body>
```

```
    <p>Hello World!</p>
```

```
    <p>Current time: <%= new Date() %></p>
```

```
  </body>
```

```
</html>
```

## ▶ Expression

```
<p>Current time: <%= new Date() %></p>
```

## ▶ Scriptlet

```
<p>Current time: <% out.println(new Date()); %></p>
```

## ► Declaration

<%!

```
private Date currentDate(){  
    return new Date();  
}
```

%>

<p>Current time: <%= currentDate() %></p>

- ▶ request – HttpServletRequest
- ▶ response – HttpServletResponse
- ▶ out – Writer
- ▶ session – HttpSession
- ▶ application – ServletContext
- ▶ pageContext – PageContext

- ▶ `jsp:include`  
Includes a file at the time the page is requested
- ▶ `jsp:forward`  
Forwards the requester to a new page
- ▶ `jsp:getProperty`  
Inserts the property of a JavaBean into the output
- ▶ `jsp:setProperty`  
Sets the property of a JavaBean
- ▶ `jsp:useBean`  
Finds or instantiates a JavaBean



# *Expression Language (EL)*

- ▶ Easy way to access JavaBeans in different scopes
- ▶ Total Sum:  $\${row.price * row.amount}$

# Basic Operators in EL

Operator	Description
.	Access a bean property or Map entry
[]	Access an array or List element
( )	Group a subexpression to change the evaluation order
+	Addition
-	Subtraction or negation of a value
*	Multiplication
/ or div	Division
% or mod	Modulo (remainder)
== or eq	Test for equality
!= or ne	Test for inequality
< or lt	Test for less than
> or gt	Test for greater than
<= or le	Test for less than or equal
>= or ge	Test for greater than or equal
&& or and	Test for logical AND
or or	Test for logical OR
! or not	Unary Boolean complement
empty	Test for empty variable values

# Scopes (revisited)

```
<%  
application.setAttribute("subject", "Web information systems");  
session.setAttribute("topic", "Servlets");  
request.setAttribute("lector", "Anti");  
%>
```

```
Subject: ${subject}  
Topic: ${topic}  
Lector: ${lector}
```

```
<%  
application.setAttribute("subject", "Web information  
    systems");  
session.setAttribute("topic", "Servlets");  
request.setAttribute("lector", "Anti");  
pageContext.setAttribute("subject", "The new topic");  
application.setAttribute("subject", "The newest topic");  
%>
```

Subject: \${subject}

Topic: \${topic}

Lector: \${lector}

```
<%  
application.setAttribute("subject", "Web information  
    systems");  
session.setAttribute("topic", "Servlets");  
request.setAttribute("lector", "Anti");  
pageContext.setAttribute("subject", "The new topic");  
application.setAttribute("subject", "The newest topic");  
%>
```

```
Subject: ${subject}  
Topic: ${topic}  
Lector: ${lector}
```

# JavaBeans & EL (1/2)

```
public class Person implements Serializable {  
  
    private String name;  
  
    public Person() {}  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
}
```

```
Person person = new Person();  
person.setName("Alan Turing");  
request.setAttribute("person",  
    person);
```

```
<p>Person: ${person.name}</p>
```

# Java Standard Tag Library (JSTL)

## ▶ Set of standard tools for JSP

<%

```
▶ List<String> lectors = Arrays.asList("Jack", "Jill", "Anti");  
pageContext.setAttribute("lectors", lectors);
```

%>

```
<c:set var="guestLector" value="Anti" />
```

```
<c:forEach var="Lector" items="{lectors}">
```

```
  Name: ${lector}
```

```
  <c:if test="{lector eq guestLector}">(guest)</c:if>
```

```
  <br />
```

```
</c:forEach>
```