JavaOne℠

Sun's 2002 Worldwide Java Developer Conference

A Standard Tag Library for JavaServer™ Pages (JSTL)

Authoring Web Applications Without Pain

Pierre Delisle
Staff Engineer
Sun Microsystems

Dan Mandell
Research Student
Stanford University

Speakers Qualifications

Pierre Delisle

- Specification Lead for JSR-052 (JSP Standard Tag Library)
- Committer at jakarta-taglibs, an open-source repository for tag libraries and tools

Dan Mandell

- Committer at jakarta-taglibs
- Author of tools such as the Custom Tag Library Extension for UltraDev (CTLX), and Tag Library Client Generator for Web Services

Overall Presentation Goal

Learn the benefits of the JavaServer Pages™ (JSP™) Standard Tag Library as well as tools to improve the process of developing dynamic web applications

Learning Objectives

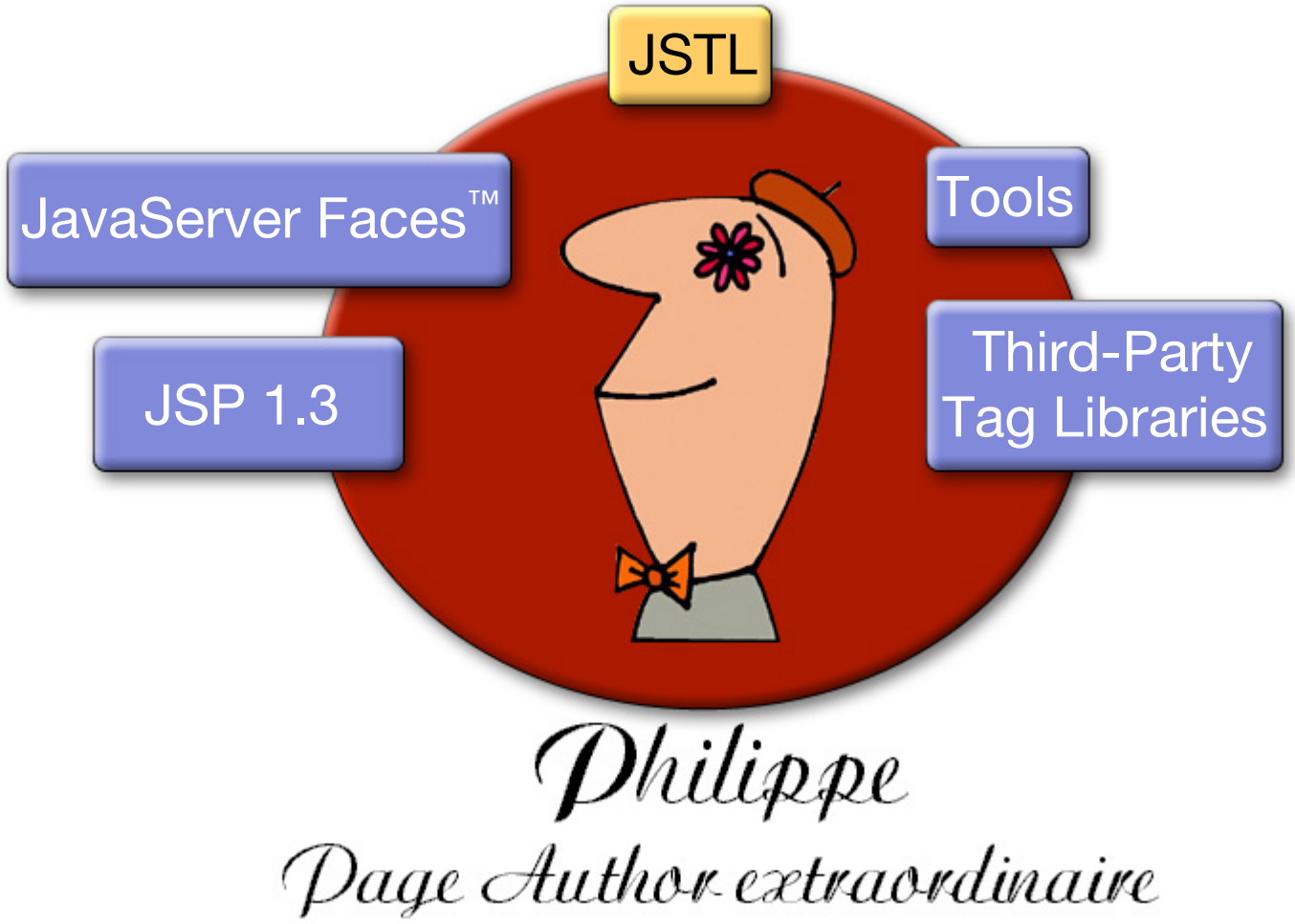
As a result of this presentation, you will be able to:

- Describe the key components of the JSP™ Standard Tag Library (JSTL)
- Understand their motivation and how they should be used
- Learn the benefits of tools and tag libraries for the development of dynamic web applications

Presentation Agenda

- JSTL
  - A language adapted for page authors
  - Standard actions for common needs
- Demo
  - Web Service Tag Library Client Generator
  - Custom Tag Library Extension for Dreamweaver UltraDev

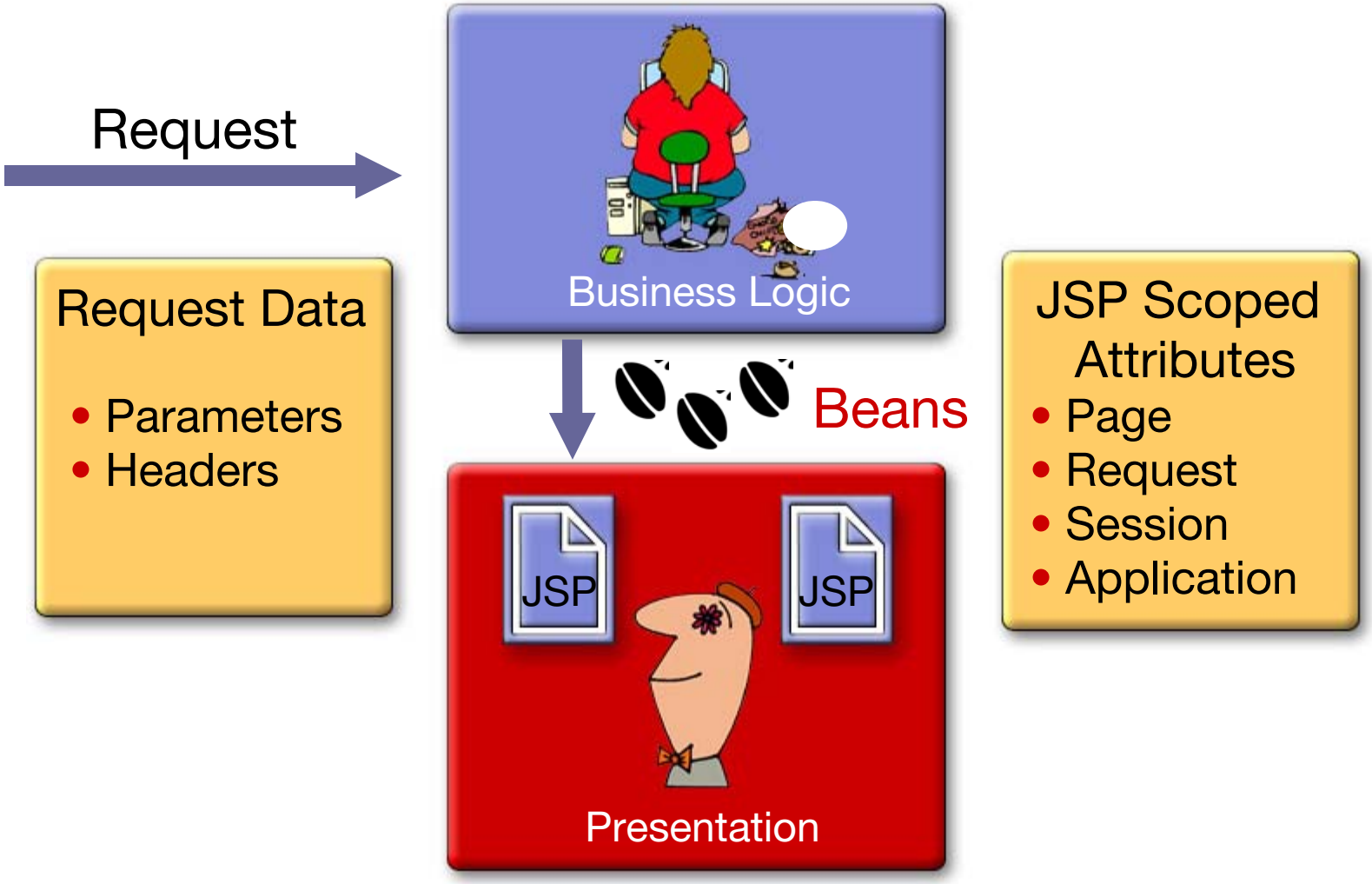
Simplify the Life of Page Authors



JavaOne

A Language Adapted for Page Authors

Easy Access/Manipulation of Application Data



Before…

1. Must declare

2. Must know type

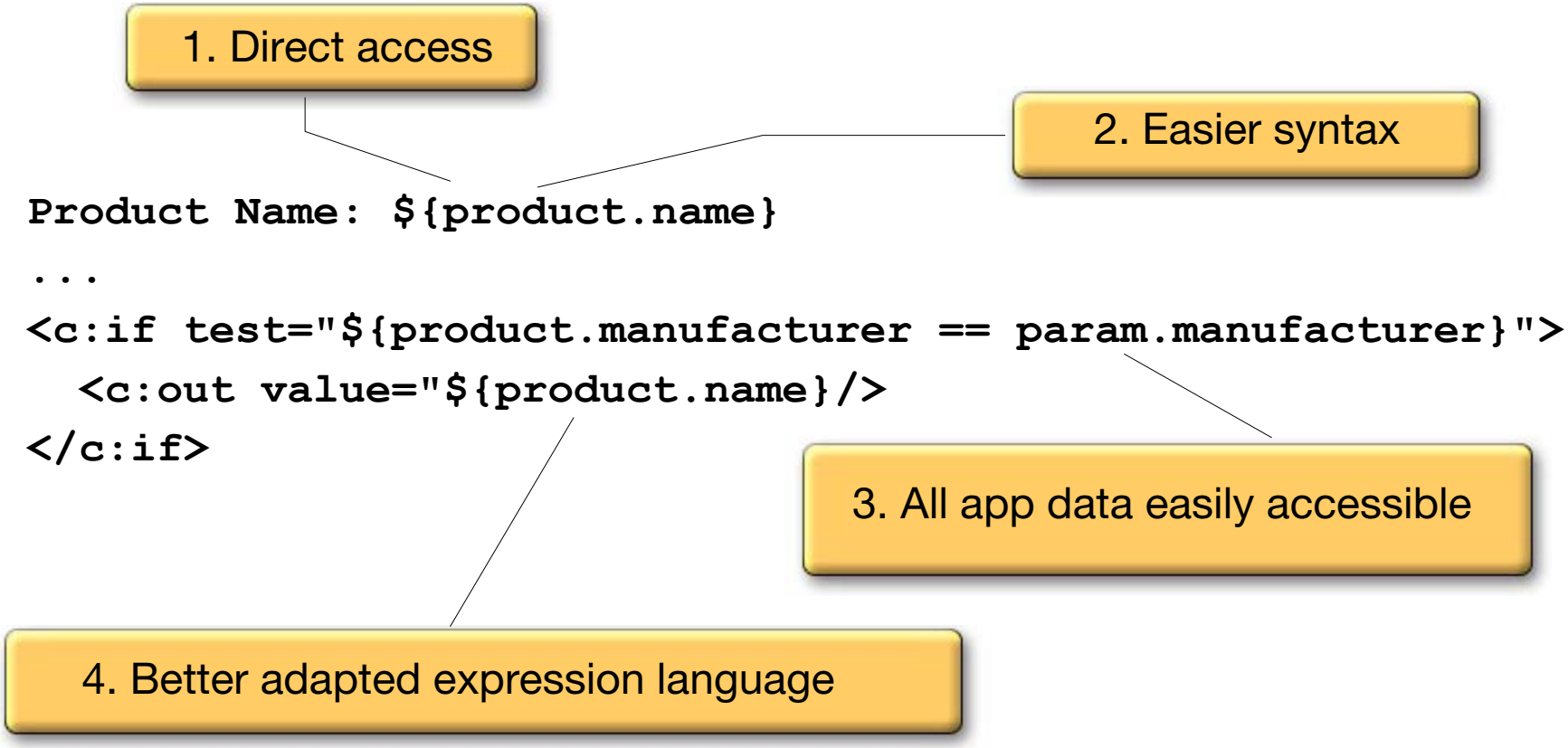
```
<jsp:useBean id="product"
    type="acme.com.Product" scope="request"/>
...
Product Name: <%= product.getName() %>
...
<% if (product.getManufacturer().equals("ACME")) { %>
   ...
<% } %>
```

3. Awkward syntax

4. Knowledge of scripting language required even for simple manipulations

JavaOne

After



JavaOne

Expression Language

A key requirement for JSTL, owned by JSR-152, designed in collaboration with the 2 expert groups

Just 6 key concepts:

Expression Language

A key requirement for JSTL, owned by JSR-152, designed in collaboration with the 2 expert groups

Just 6 key concepts:

1. Syntax

```
${expression}

value="product_${name}.${type}"
```

Expression Language

A key requirement for JSTL, owned by JSR-152, designed in collaboration with the 2 expert groups

Just 6 key concepts:

2. Application data directly accessible

```
${foo} -> pageContext.findAttribute("foo")
```

Implicit objects:

- `page, request, session, application`
- `param/params`
- `pageContext`

Expression Language

A key requirement for JSTL, owned by JSR-152, designed in collaboration with the 2 expert groups

Just 6 key concepts:

3. Beans and collections rule

```
${user.address.city}
${products[product.id]}
${products["DCR-PC100"]}
```

Expression Language

A key requirement for JSTL, owned by JSR-152, designed in collaboration with the 2 expert groups

Just 6 key concepts:

4. Operators

```
relational:
  ==, !=, <, >, <=, >=
arithmetic:
  +, -, *, /, %
binary:
  and, or, not
```

Expression Language

A key requirement for JSTL, owned by JSR-152, designed in collaboration with the 2 expert groups

Just 6 key concepts:

5. Automatic type conversions


```
begin="${request.beginValue}"
```

Expression Language

A key requirement for JSTL, owned by JSR-152, designed in collaboration with the 2 expert groups

Just 6 key concepts:

6. Default values

```
<c:set var="city"
       value="${user.address.city}"
       default="N/A"/>
```

EL Support Actions

<c:out> – Display EL expression

```
<c:out value="${customer.address.city}"
       default="unknown"/>
```

<c:set> – Set scoped variable

```
<acme:foo>
  <acme:bar>
    <x:atag>...</x:atag/>
  </acme:bar>
</acme:foo>
```

```
<c:set var="bar">
  <x:atag>...</x:atag/>
</c:set>
<acme:foo bar="${bar}"/>
```

<c:remove> – Remove scoped variable

Conditional Actions

Simple conditional execution

```
<c:if test="${user.visitCount == 1}">
    This is your first visit!
</c:if>
```

Mutually exclusive conditional execution

```
<c:choose>
  <c:when test="${verbosityLevel == 'short'}>
    <c:out value="${product.shortDescription}"/>
  </c:when>
  <c:otherwise>
    <c:out value="${product.longDescription}"/>
  </c:otherwise>
</c:choose>
```

Iteration Actions

```
<table>
  <c:forEach items="${customers}"
             var="customer"
             varStatus="status">
    <tr>
      <td><c:out value="${status.count}"/></td>
      <td><c:out value="${customer}"/></td>
    </tr>
  </c:forEach>
</table>
```

1. All J2SE collection types supported

2. Current item

3. Iteration status

- begin, end, step
- `<c:forTokens delims="...">`

Standard Actions for Common Needs

Multiple TLDs

One jar file with multiple TLDs to better group the tags within their own taglib

(since JSTL covers a broad range of topics)

Functional Area	URL	Prefix	Example
Core	http://java.sun.com/jstl/ea/core	c	`<c:tagname ...>`
XML processing	http://java.sun.com/jstl/ea/xml	x	`<x:tagname ...>`
I18N-capable formatting	http://java.sun.com/jstl/ea/fmt	fmt	`<fmt:tagname ...>`
Database access	http://java.sun.com/jstl/ea/sql	sql	`<sql:tagname ...>`

URL Related Actions

Hypertext Links

URL rewriting

```
<c:url url="http://acme.com/exec/register"
       var="myUrl">
  <c:param name="name" value="${param.name}"/>
  <c:param name="country" value="${param.country}"/>
</c:url>

<a href='<c:out value="${myUrl}"/>'>Register</a>
```

Parameter name and value automatically URL encoded

JavaOne

<jsp:include> —Webapp Centric

Web Application
Same Context

Relative URL

```
/foo/bar.jsp
    bar.jsp
```

```
<jsp:include>
```

<c:import> —URL Centric

Internet Resources

Absolute URL

```
http://acme.com/foo
ftp://xyz.org/README
```

Web Application
Same Context

Relative URL

```
/foo/bar.jsp
bar.jsp
```

Web Application
Foreign Context

Relative URL

```
context="/global"
/copyright.html
```

```
<c:import url="...">
```

Resource Content Included Inline: Unnecessary Buffering

1. Read resource content
2. Write content to JspWriter

```
<acme:transform>
  <jsp:include page="/exec/employeesList"/>
</acme:transform>
```

3. Read body content
4. Process data

Resource Content Included Inline, or Exported as String or Reader

Included inline

```
<c:import url="/xml/doc.xml" var="in"/>
```

String: cached and reusable

```
<c:import url="/xml/doc.xml" var="in"/>
<acme:process in="$in"/>
```

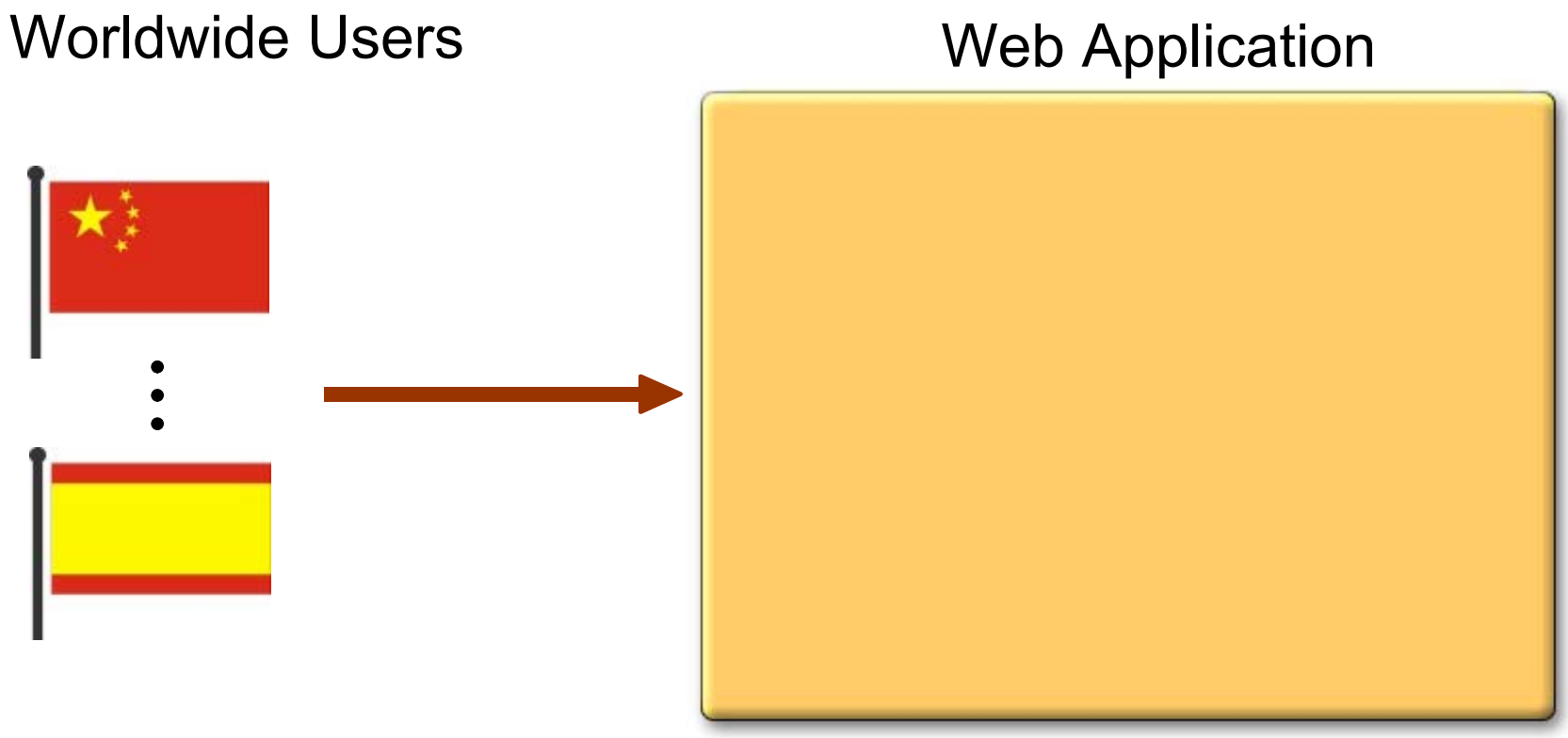
Reader: no buffering

```
<c:import url="http://acme.com/customers"
          varReader="in"/>
  <acme:process customers="$in"/>
</c:import>
```

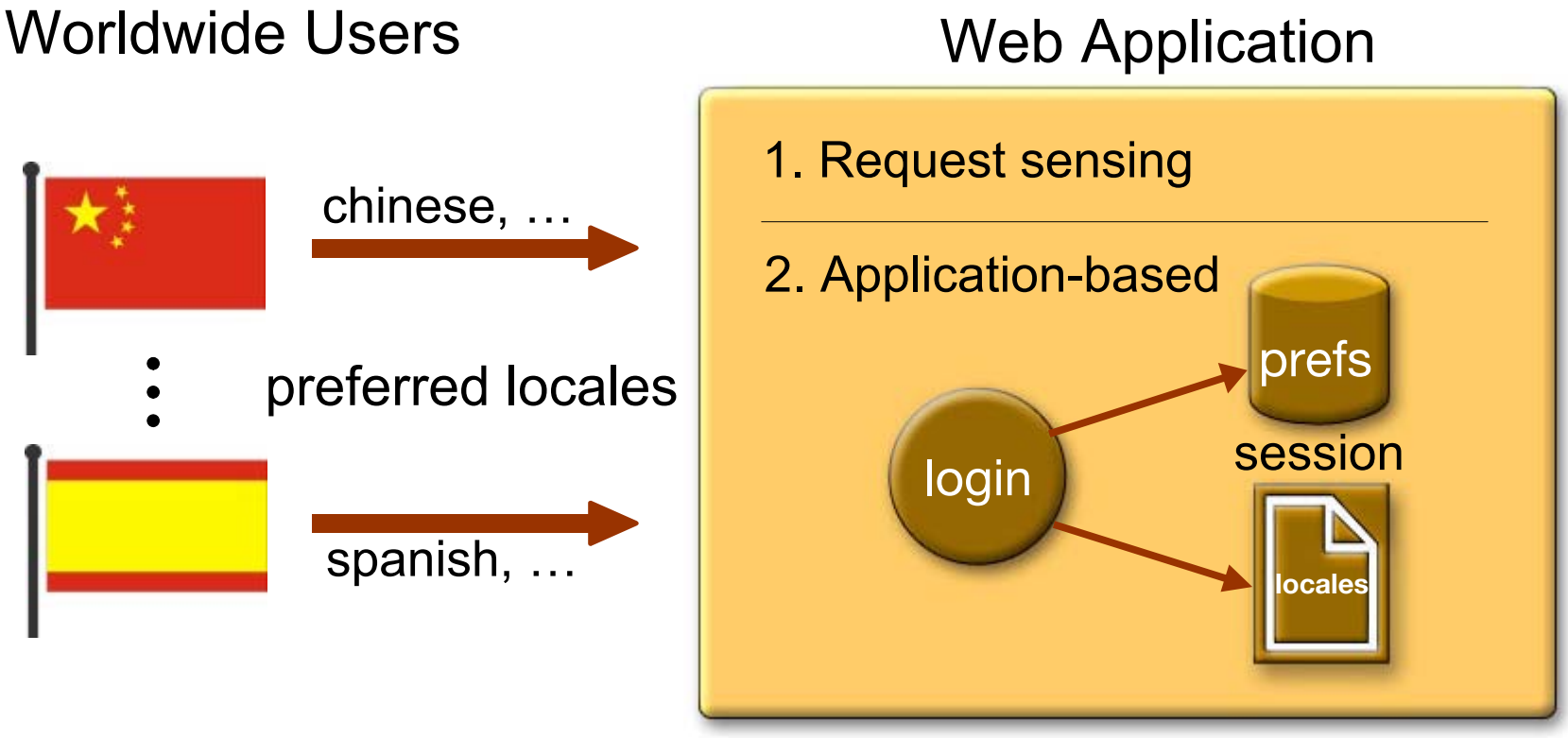

JavaOne

Internationalization (I18N) and Text Formatting

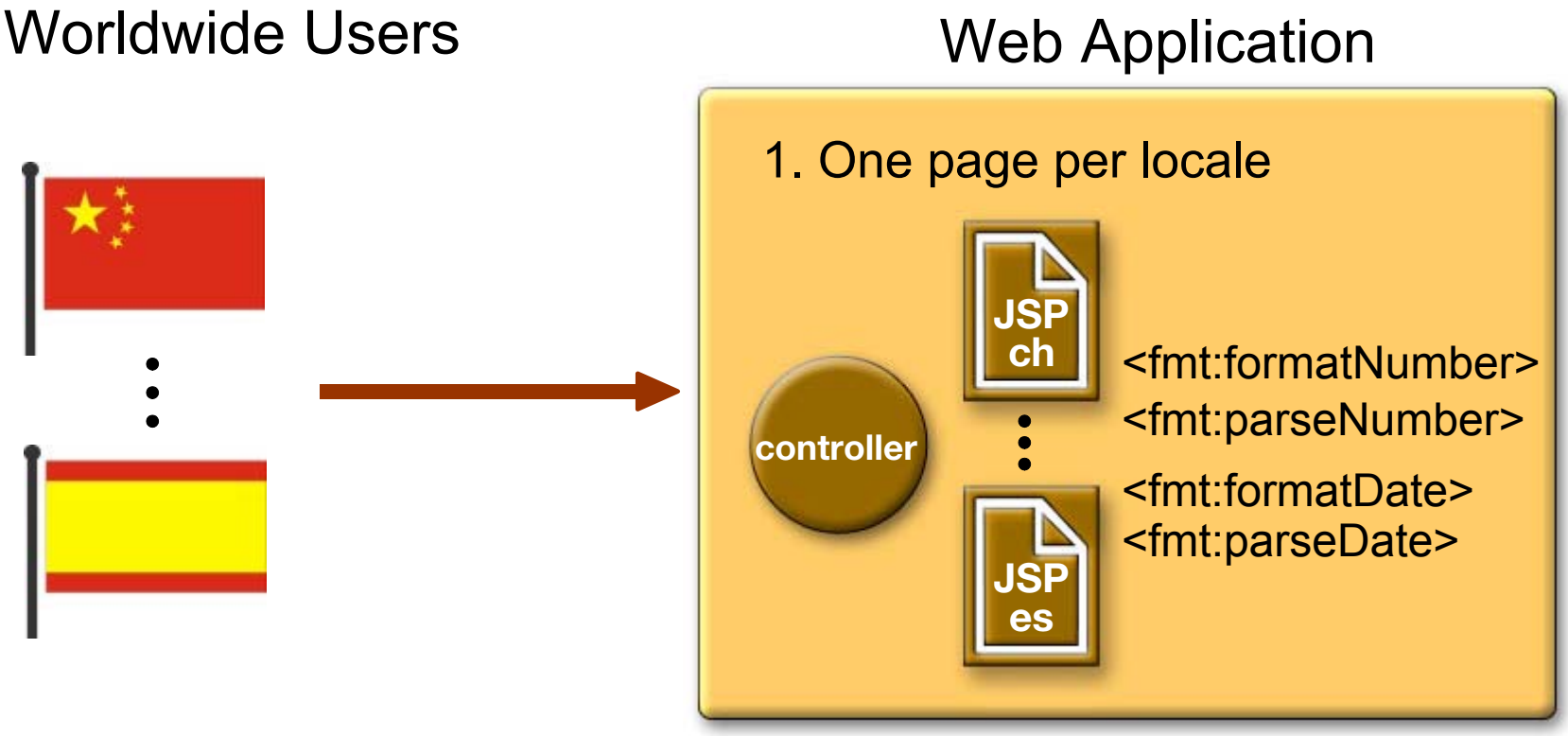
Webapp for Global Markets



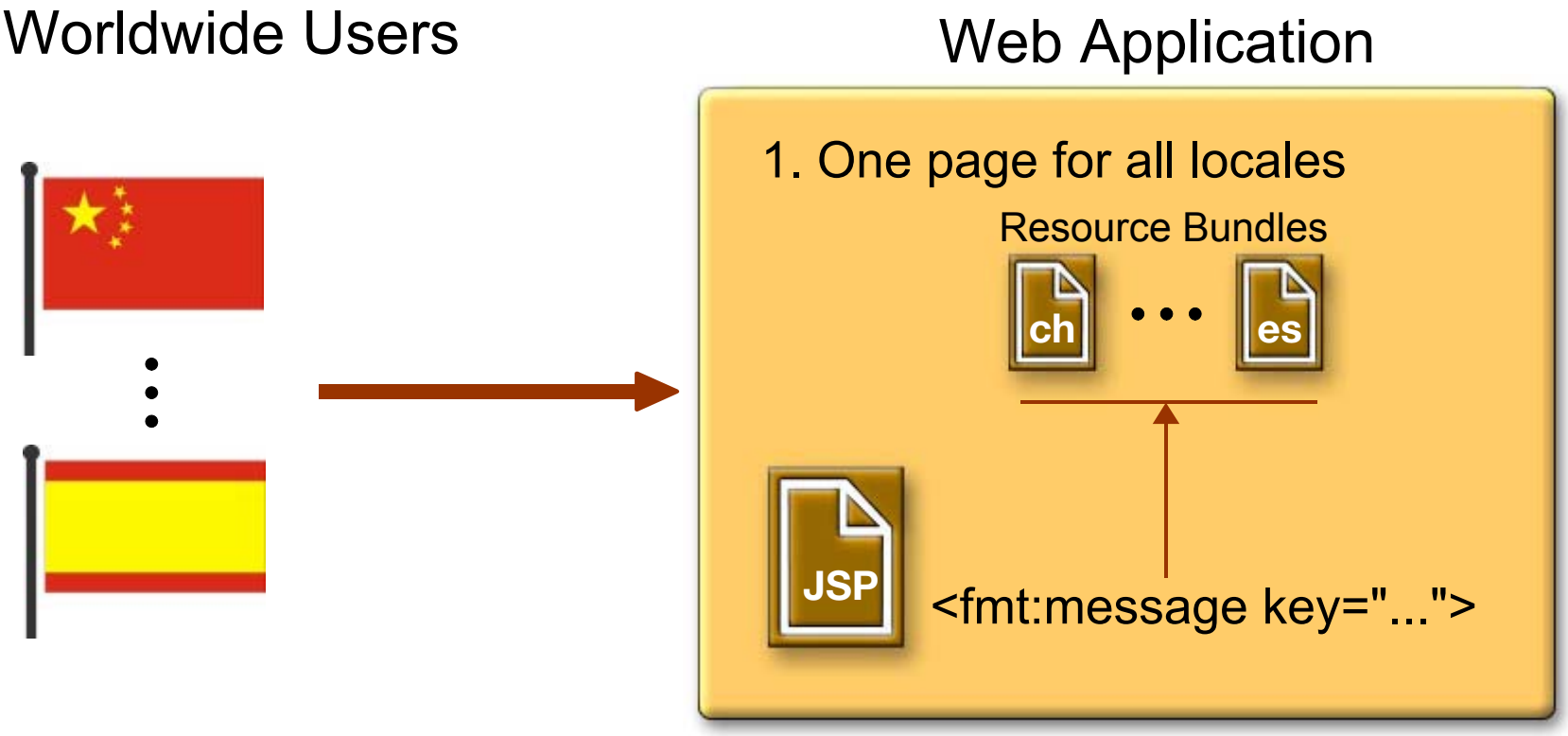
Locale


Worldwide Users
chinese, …
preferred locales
spanish, …
Web Application
1. Request sensing
2. Application-based
prefs
session
login
locales

I18N Architecture



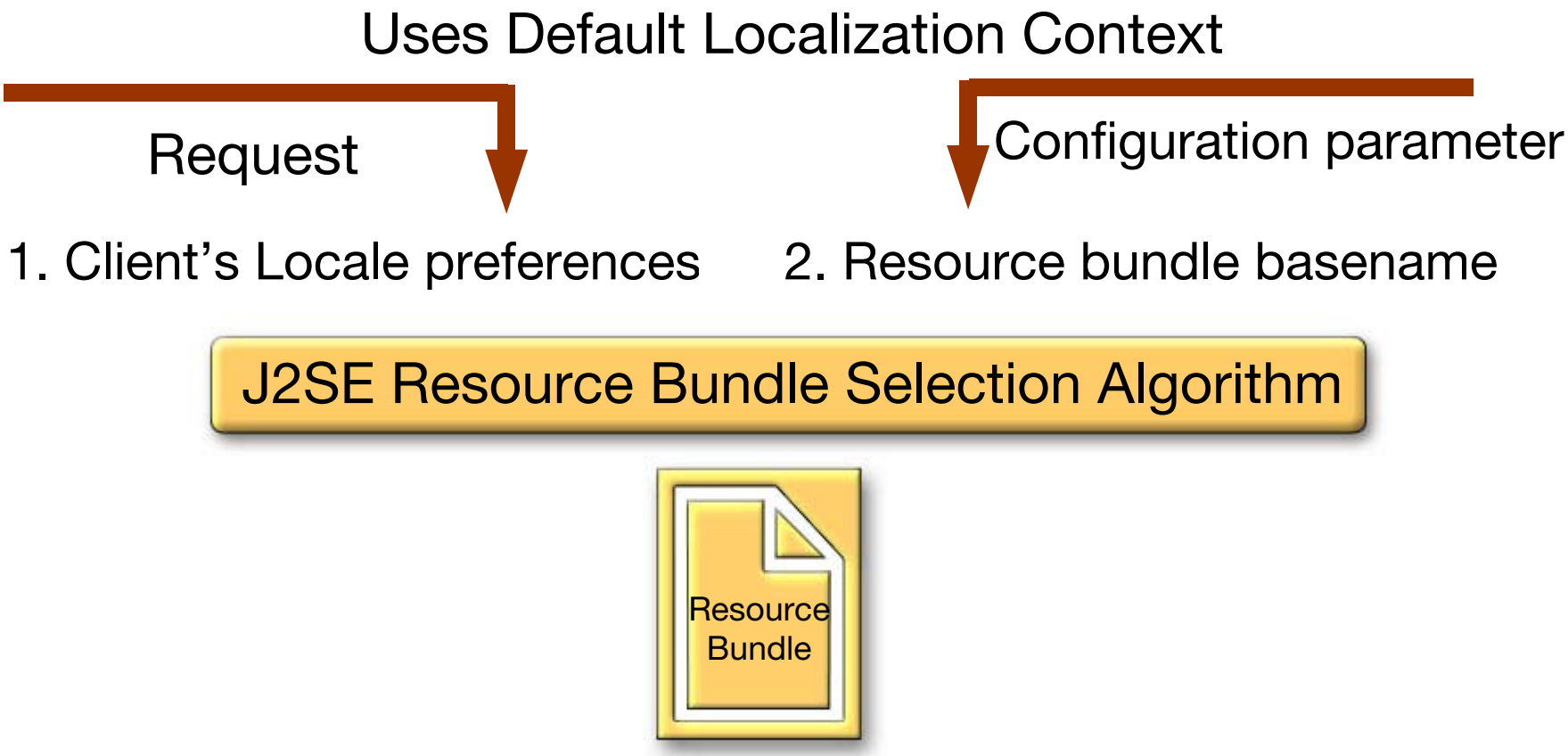
I18N Architecture



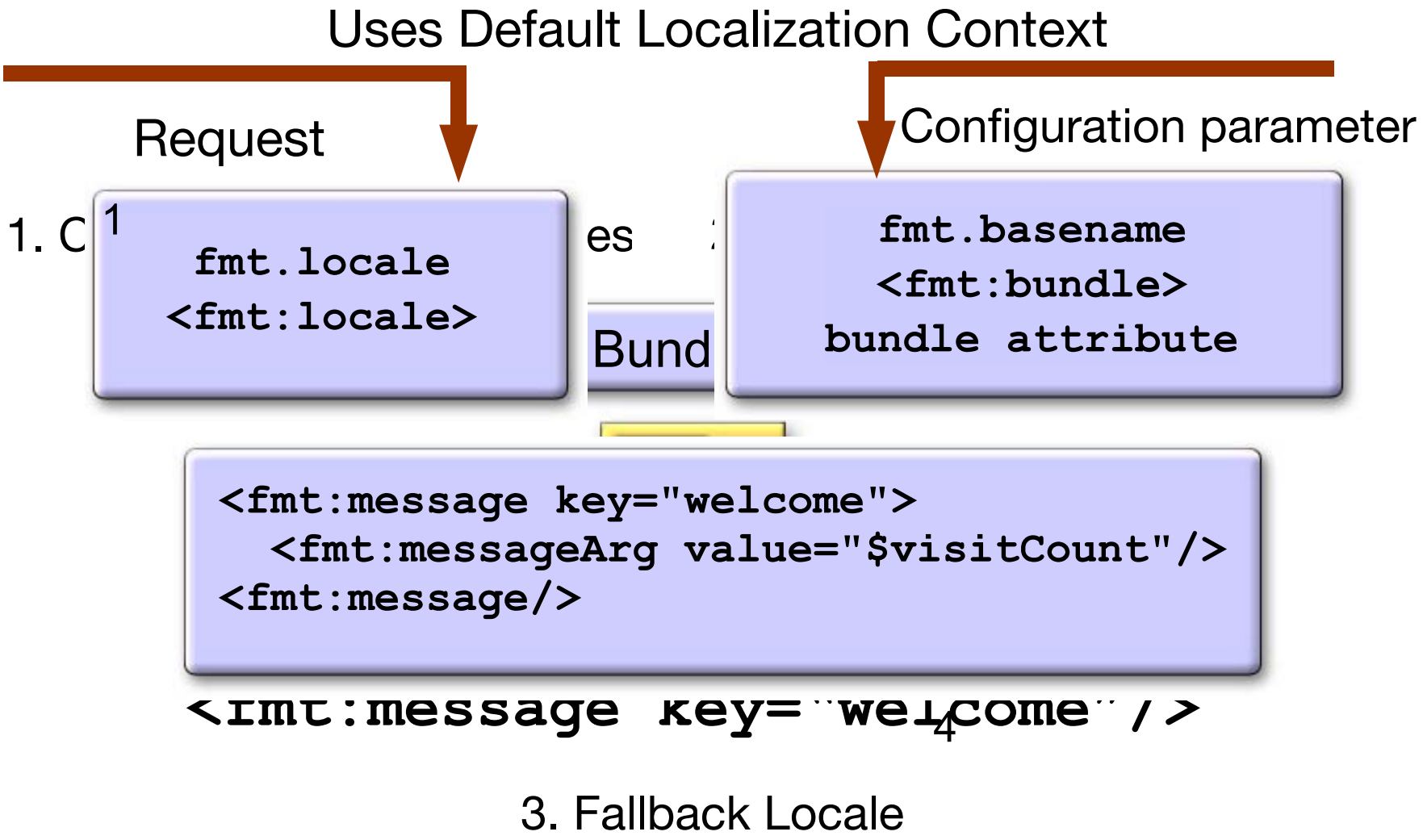
How Easy to I18N a Webapp?

```
<fmt:message key="welcome"/>
```

How Easy to I18N a Webapp?

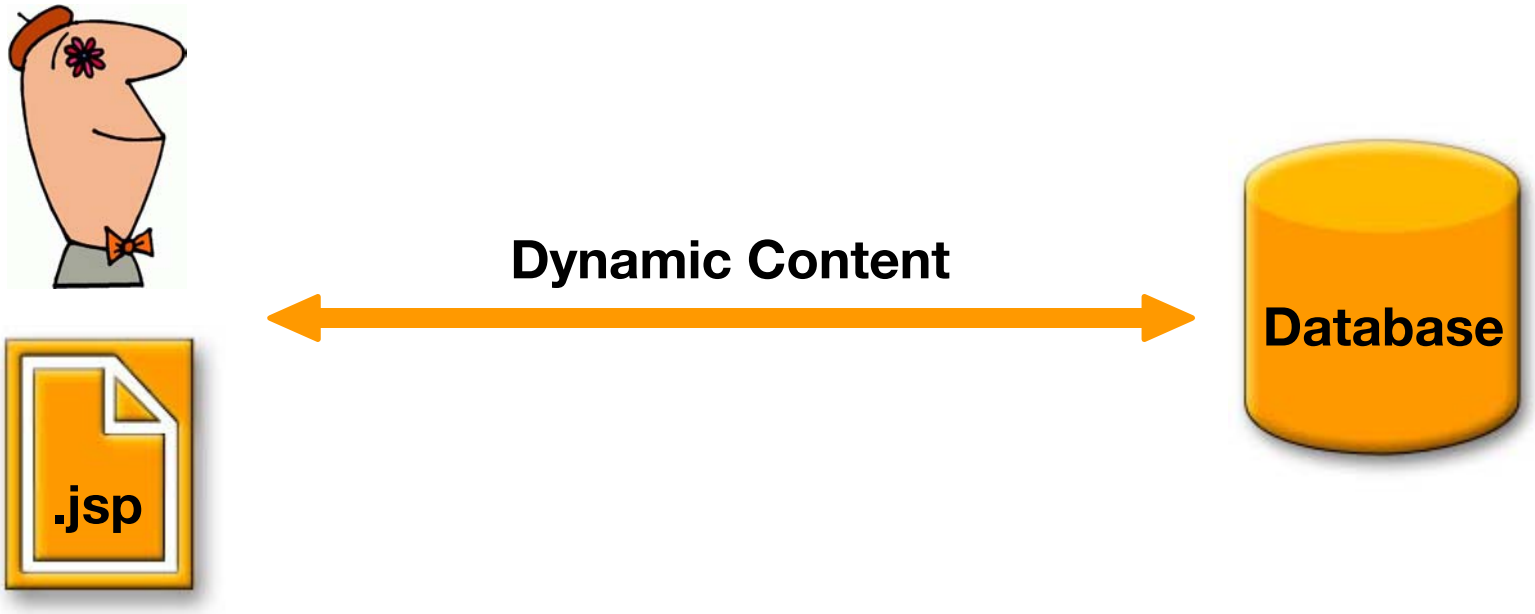


I18N Flexibility

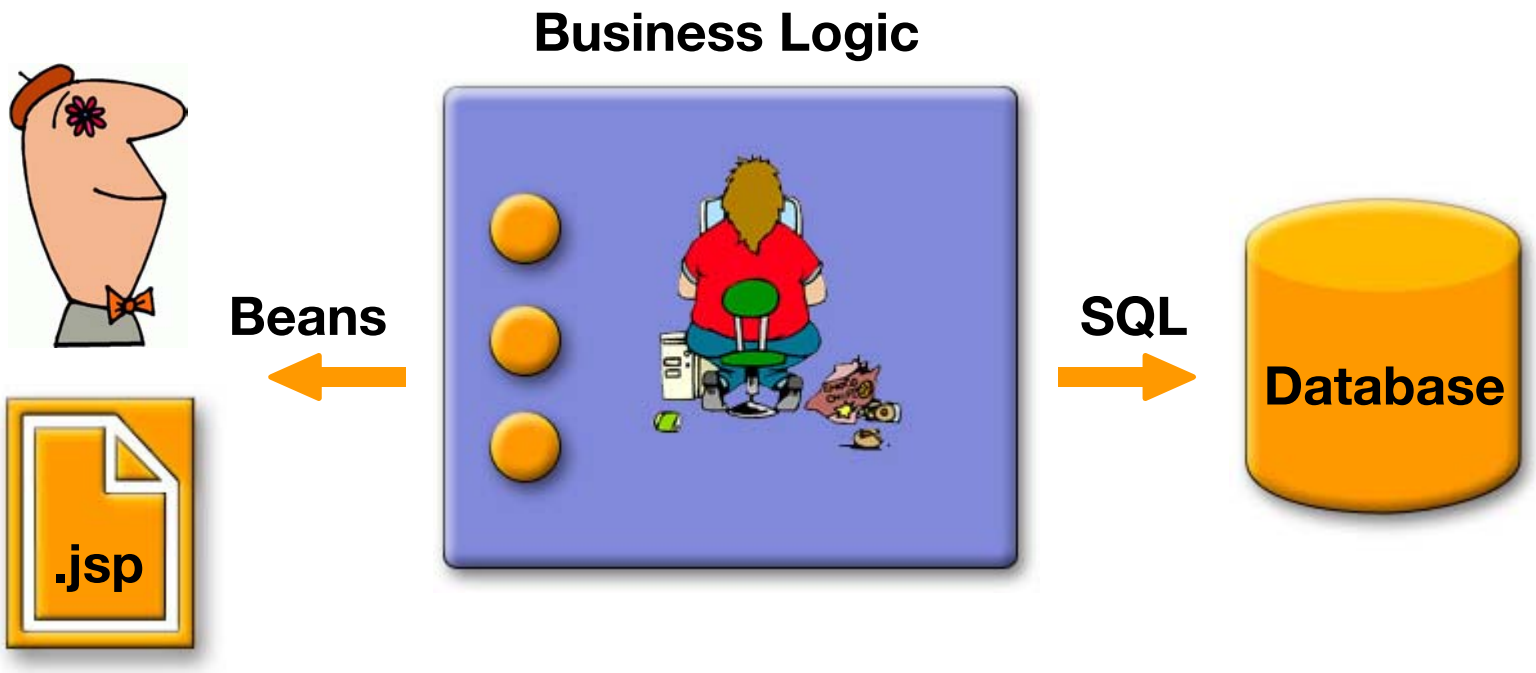


Database Access

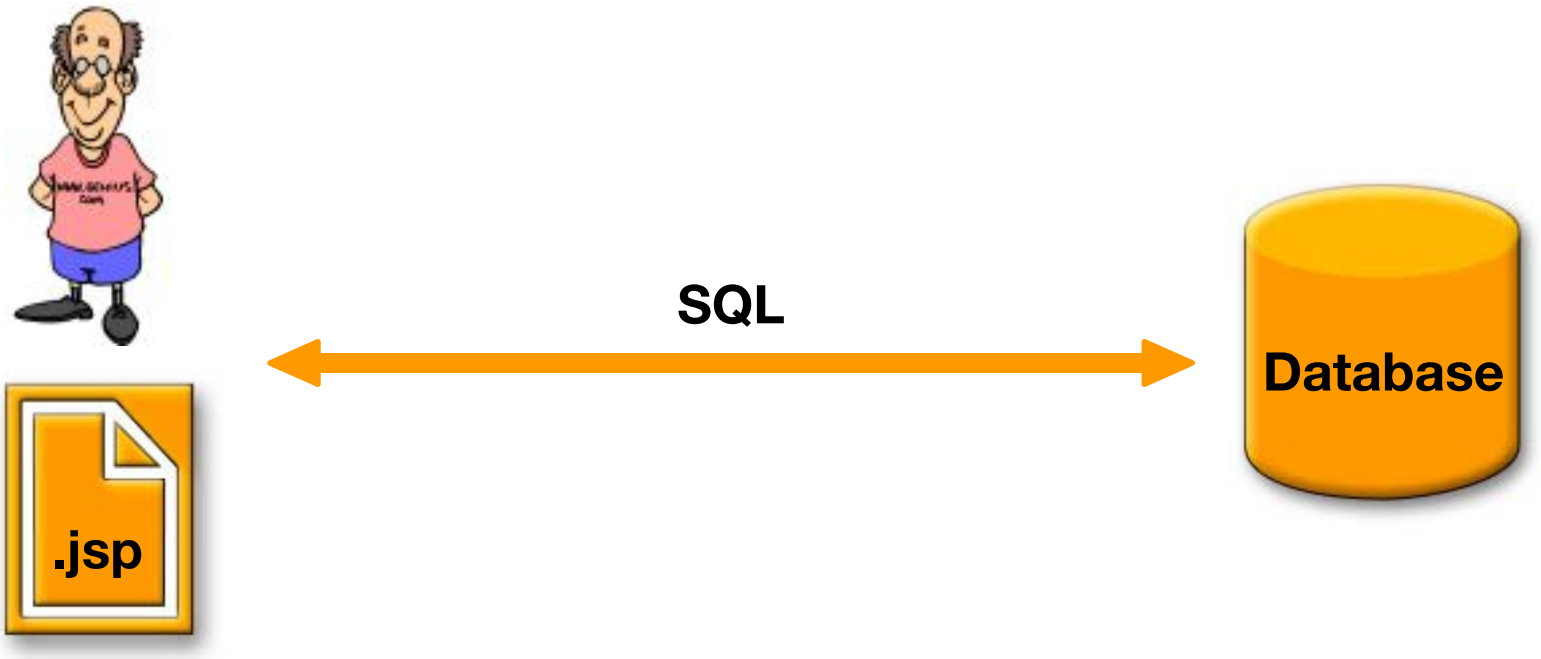
Database Access



MVC Architecture



RAD/Prototyping/Simple Apps



SQL Actions


```
Query the database
  <sql:query>

Easy access to
result set
  Result
  ResultSupport


Update the database
  <sql:update>
  <sql:transaction>
```

DataSource

All DB actions operate on a DataSource

Different ways to access a DataSource

- Transparent collaboration
  - via configuration parameter **`sql.dataSource`**

```
<db:query query="..." />
```

- Explicit collaboration
  - Object provided by application logic
  - Object provided by <sql:driver> action

```
<sql:driver var="dataSource"
     driver="org.gjt.mm.mysql.Driver"
     url="jdbc:..."/>
<sql:query dataSource="${dataSource}" .../>
```

Querying a Database

```
<sql:query var="customers" dataSource="${dataSource}">
    SELECT * FROM customers
    WHERE country = 'China'
    ORDER BY lastname
</sql:query>

<table>
<c:forEach var="row" items="${customers.rows}">
  <tr>
    <td><c:out value="${row.lastName}"/></td>
    <td><c:out value="${row.firstName}"/></td>
    <td><c:out value="${row.address}"/></td>
  </tr>
</c:forEach>
</table>
```

Updating a Database

```
<sql:transaction dataSource="${dataSource}">
  <sql:update>
    UPDATE account
    SET Balance = Balance - ?
    WHERE accountNo = ?
    <sql:param value="${transferAmount}">
    <sql:param value="${accountFrom}">
  </sql:update>
  <sql:update>
    UPDATE account
    SET Balance = Balance + ?
    WHERE accountNo = ?
    <sql:param value="${transferAmount}">
    <sql:param value="${accountTo}">
  </sql:update>
</sql:transaction>
```

XML

XML Everywhere

1. Easy Access to XML Data

XPath

```
$doc/employee[@name=$param:name]
```

XPath variable bindings

1. Easy Access to XML Data

XPath – expression language for XML actions

```
<x:out>
<x:set>
```

```
<x:if>
<x:choose>
<x:forEach>
```

`select` attribute to specify XPath expression

2. Get the XML Data

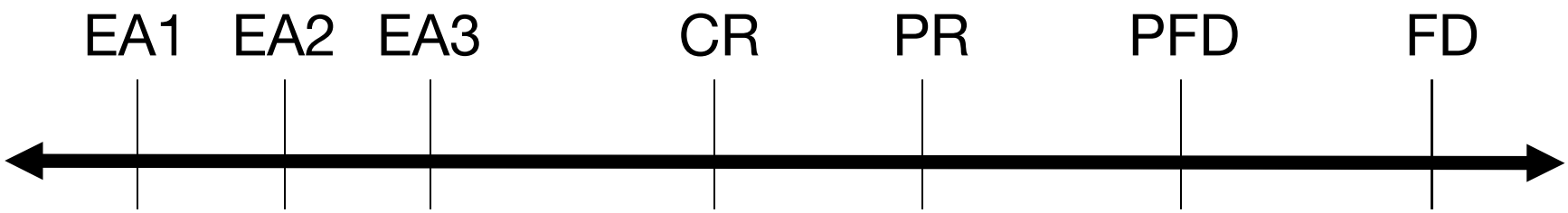

```
<x:parse>
in    xmlUrl,xmlText,body content
out   var, varDom
perf filter (org.xml.sax.XMLFilter)
```

3. XSLT Transformation


```
<x:transform>
in    xmlUrl,xmlText,body content
in    xsltUrl,xsltText
out   var,result,page
perf  transparent caching of Transformer objects possible
```

Status

Schedule



7/3/01
EL support
TLVs
Flow Control

11/21/01
Import
I18N
XML
EcmaScript

12/12/01
Database

2/4/02

3/12/02
Beta 1

4/15/02
Beta 2

5/6/02

FCS—5/21/02

Demo

Specification and Resources

JSTL Specification—Public Review

http://jcp.org/aboutJava/communityprocess/review/jsr052

JSTL Reference Implementation

http://jakarta.apache.org/taglibs/doc/standard-doc/intro.html

Tag Libraries

http://java.sun.com/products/jsp/taglibraries.html

CTLX

http://jakarta.apache.org/taglibs/doc/ultradev4-doc/intro.html

BOF 1714—JSTL Community Input

Wed 7:00 PM, Ralston, Sheraton Palace

Summary

The JavaServer Pages™ Standard Tag Library (JSTL) simplifies the life of a page author

- Language better adapted to the needs of page authors
- Standard actions for common needs

Now in Public Review: Use it and send comments to jsr-52-comments@jcp.org

Tools are getting there…give them a try!

Q&A

JavaOne℠

Sun's 2002 Worldwide Java Developer Conference™

BEYOND BOUNDARIES

Session 1694