Ajax and Web 2.0 Related Frameworks and Toolkits

SUN TECH DAYS 2008–2009
A Worldwide Developer Conference

Dennis Chen
Director of Product Engineering / Potix Corporation
dennischen@zkoss.org

Agenda

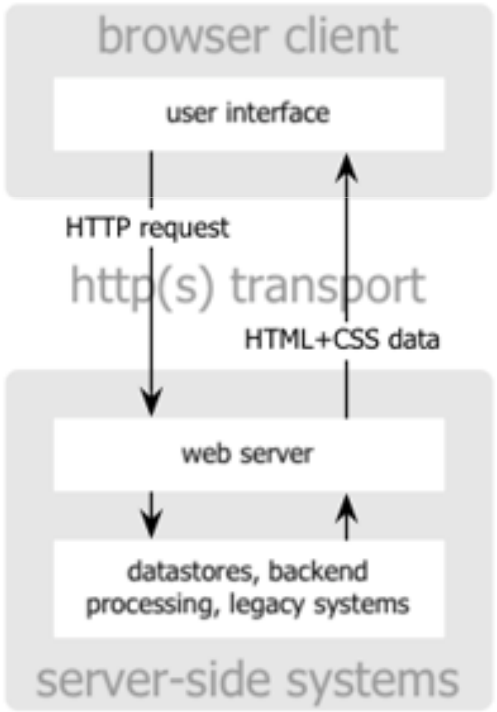
- Ajax Introduction
- Access Server Side (Java) API/Data/Service
  - jQuery + DWR
  - GWT
  - ZK
- Summary

AJAX INTRODUCTION

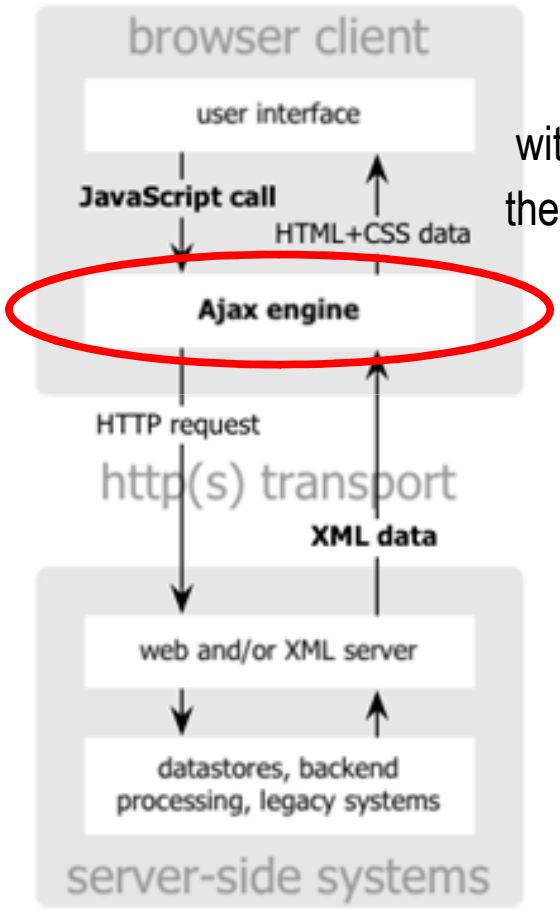
What is Ajax

- Asynchronous JavaScript and XML
- Browser asynchronously get data from a server and update page dynamically without refreshing(reloading) the whole page
- Web Application Development Techniques
  - DHTML, CSS (Cascade Style Sheet)
  - Javascript and HTML DOM
  - Asynchronous XMLHttpRequest

Traditional Web Application vs. Ajax



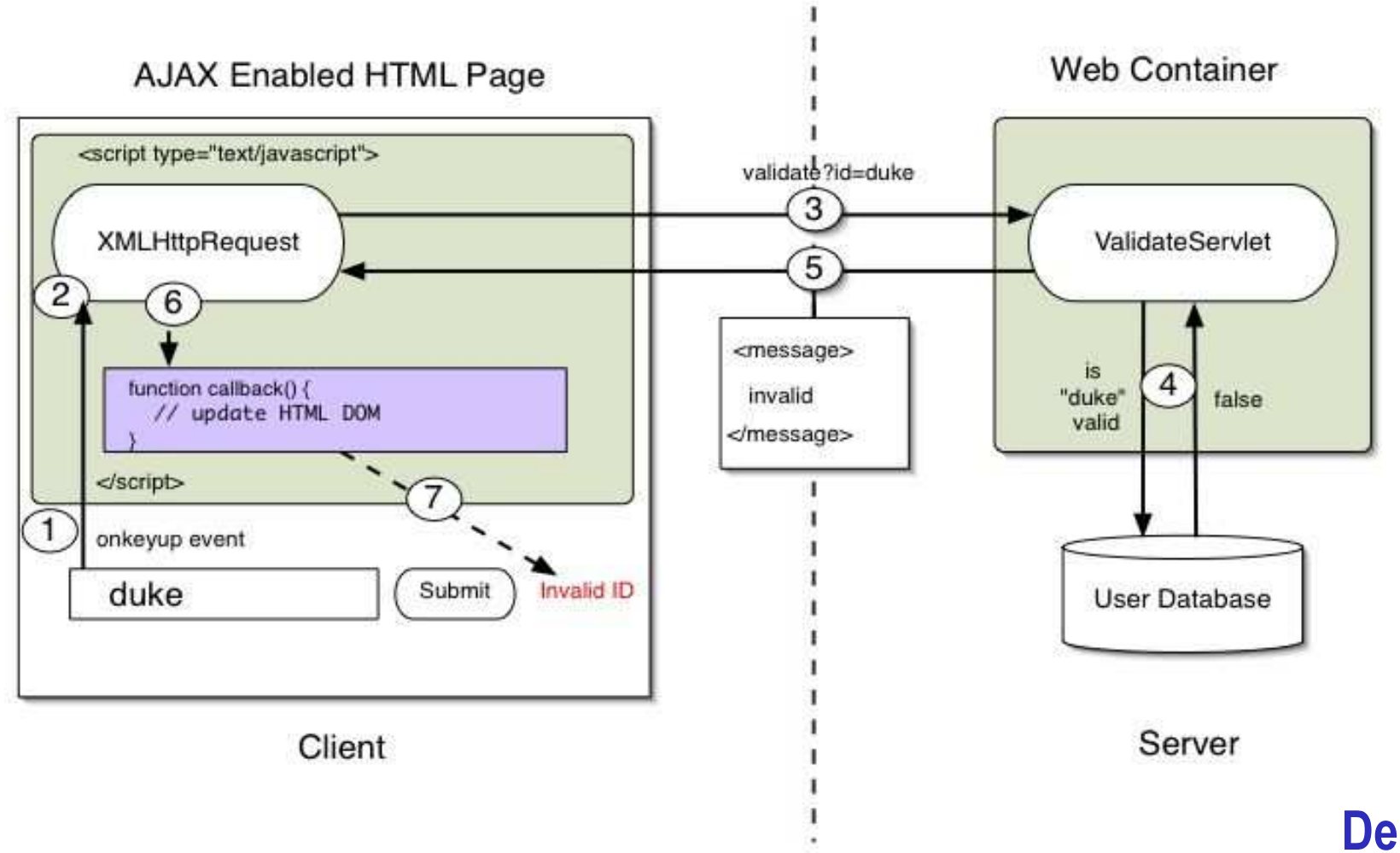
classic
web application model



Ajax
web application model

within a browser,
there is AJAX engine

Ajax Interaction Example



Demo

Challenge of Providing Ajax

- Technique Issue
  - JavaScript , DHTML and CSS
- Cross Browser Issue
  - IE 6,7,8 , Firefox 2,3 , Safari 3, Opera 9, Google Chrome…
- Asynchronous Event and Data
- Business Logic and Security
- Maintenance

Solutions : Ajax Frameworks / Toolkits

- Client Side Toolkit
  - > jQuery, Prototype , …
- Client Side Framework
  - > GWT ,YUI, Ext-JS, Dojo …
- Server Side Toolkit (Java)
  - > DWR…
- Server Side Framwork
  - > ZK, ECHO2, ICEface ..

zk Direct RIA

dojo the java

ext

Google Web Toolkit

DWR

Yahoo! UI Library

BACKBASE

OpenLaszlo

NextApp

ICEfaces.org

APACHE WICKET GRADUATED

JQUERY + DWR

What is jQuery

- http://jquery.com/
- A JavaScript toolkit that simplifies the interaction between HTML and JavaScript.
- Lightweight and Client only
- Provides high level function to
  - Simplify DOM access and alternation
  - Enable multiple handlers per event
  - Add effects/animations
  - Asynchronous interactions with server
- Solve browser incompatibilities

Powerful Selector / Traversing API

- Powerful Selector

```
$(“div”) // by tag
$(“.errorbox”) //by css class
$(“#cur_article”) //by element id
$(“form input”) //all input in form
```

Demo

- Traversing API

```
$(“div”).eq(2) // div which on index 2
$(“div”).not(".green, #blueone“)
$(“div”).filter(fn)
```

Demo

Handle Events

```
//listen click
$(“#cur_article").click(function () {
   $(this).toggleClass("highlight");
});

//listen click
$(“#target”).bind(“click”,fn);

//fire click event
$(“#target”).trigger(“click”);

//unlisten click
$(“#target”).unbind(“click”,fn);
```

Demo

DOM Manipulation

```
//change attribute
val = $(“#target”).attr(“a_attribute”);
$(“#targetX”).attr(“a_attribute”,val);

//change html content
val = $(“#target”).text();
$(“#targetY”).html(val);

//change style class
$(“#target”).addClass(“error”);
$(“#targetZ”).toggleClass(“error”);
```

Demo

Other Features

- Animation
  - show(), hide(), fadein(), fadeout(), slideup(), slidedown()
- Widget
  - Accordion, Datepicker , Dialog, Slider, Tabs
- jQuery.ajax / Ajax Assistance
  - load(url, [data], callback)
  - $.get(url, [data], callback)
  - $.post(url, [data], callback)
  - $.ajax(options)

Demo

What is DWR

- http://directwebremoting.org/
- Direct Web Remoting
- RPC calls from client-side JavaScript to server-side Java object's API
- Generates client stub (Proxy), which is a JavaScript code
- Provides server side runtime to pass client RPC
- Client stub (Proxy) handles marshalling of parameters and return value

Generate JavasSript Stub Code

- Java
  - AjaxService.getOptions():String[]
- Javascript Stub
  - AjaxService.getOptions(callback)
  - function callback(data){}

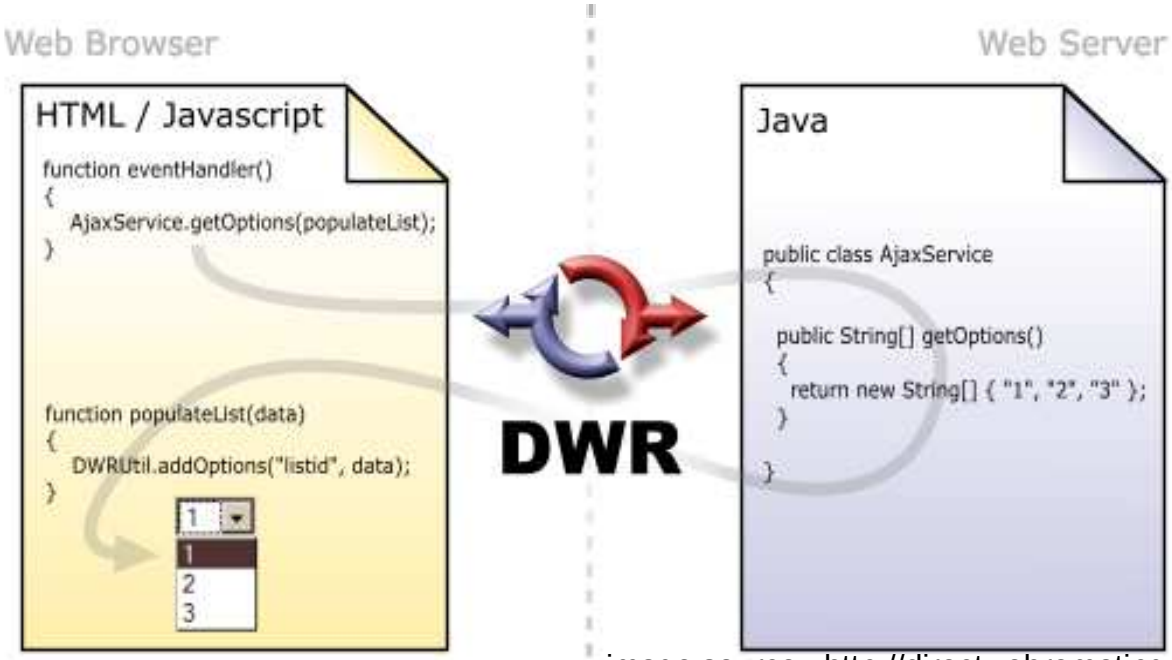
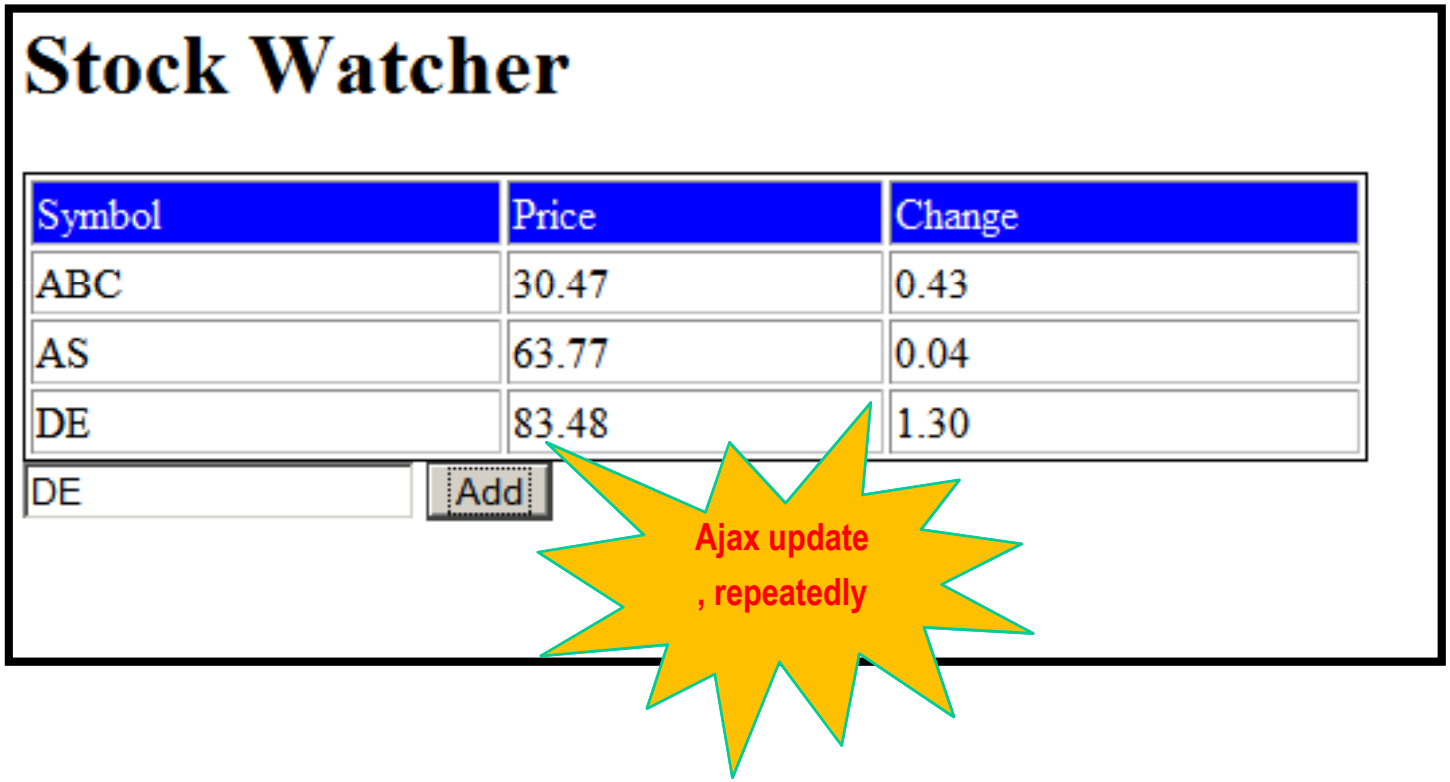


image source : http://directwebremoting.org/

Convert Data Object to JavaScript

- Primitive Type
- Date, Array, Collection
- DOM Objects
- JavaBean

Stock Watcher



[Example] Stock Watcher

- Server Side Java Code

```
public class StockService {
    public StockPrice[] getPrice(String[] symbols){
        …
    }
}

public class StockPrice {
    private String symbol;
    private double price;
    private double change;

    public String getSymbol(){…}
    public double getPrice(){…}
    public double getChange(){…}
    …
}
```

[Example] Stock Watcher (Cont. 1)

- DWR Configuration – dwr.xml

```
<dwr>
  <allow>
    <create creator="new" javascript="StockService">
      <param name="class" value="my.StockService" />
    </create>
    <convert converter="bean" match="my.StockPrice">
      <param name="include" value="symbol, price, change" />
    </convert>
  </allow>
</dwr>
```

[Example] Stock Watcher (Cont. 2)

- Html

```
<script src="jquery-1.2.6.min.js"></script>
<script src='dwr/interface/StockService.js'></script>
<script src='dwr/engine.js'></script>
…
<table >…<tbody id="stockList“ /></table>
<div>
   <input id="symbol" type="text">
   <button id="btn" >Add</button>
</div>
```

- Client Side Java Script

```
…
SW.symbols = []; // a symbol array
$("#btn").click( function(){
    var symbol = $("#symbol").val();
   SW.symbols.push(symbol);
   StockService.getPrice(SW.symbols,SW.update);
});
```

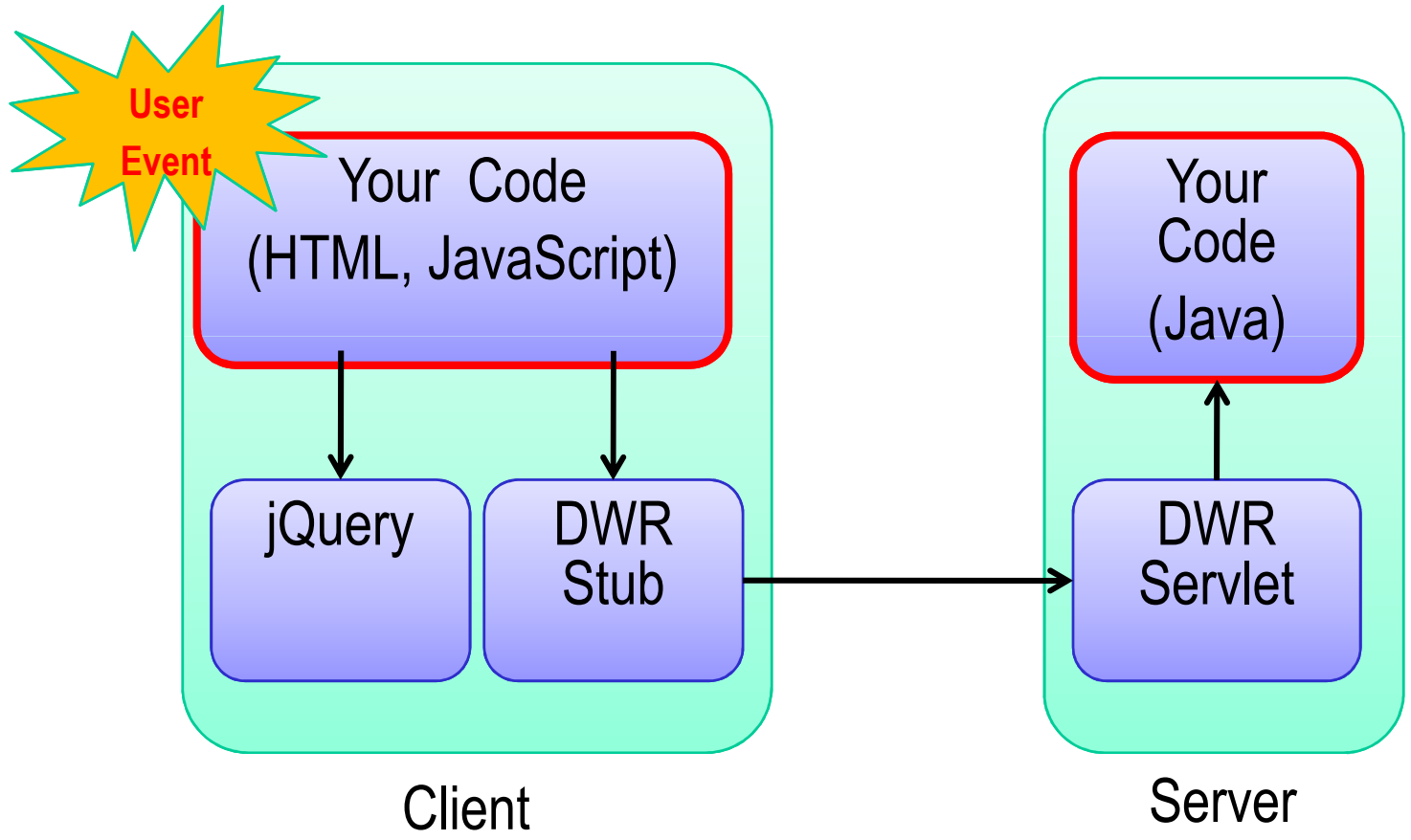
[Example] Stock Watcher (Cont. 3)

- Client Side Java Script

```
//ajax callback
SW.update = function(data) {
        var stockList = $("#stockList");
        stockList.empty();
        for ( var i = 0; i < SW.symbols.length; i++) {
                var row = "<tr>";
                row += "<td>" + SW.symbols[i] + "</td>";
                if (data) {
                        for ( var j = 0; j < data.length; j++) {
                                if (data[j].symbol == SW.symbols[i]) {
                                        row += "<td>" + data[j].price.toFixed(2) + "</td>";
                                        row += "<td>" + data[j].change.toFixed(2) + "</td>";
                                }
                        }
                }
                row += "</tr>";
                stockList.append(row);
        }
}
```

Demo

Code Structure



It Helps, But…

- jQuery + DWR helps you
  - Manipulate HTML more easier
  - Do Ajax request more easier
- You still have to
  - Write Javascript
  - Manipulate HTML DOM
  - Build rich UIs
    - <table/> <input/>…

GWT

What is GWT

- http://code.google.com/webtoolkit/
- Google Web Toolkit
- Write AJAX apps in the Java language
- Component-base, Event-driven
- Compile, Deploy and Run in JavaScript,
  - Host mode , debug in Java
- Communicate with server through RPC
  - Simple/Default RPC
  - JSON

What does the GWT code look like

MyApplication.java

```
1  package org.zkoss.gwt.demo1.client;
2
3  import com.google.gwt.core.client.EntryPoint;
10
11 public class MyApplication implements EntryPoint {
12
13     public void onModuleLoad() {
14         VerticalPanel vPanel = new VerticalPanel();// the layout
15         vPanel.setWidth("100%");// the style
16         vPanel.setHorizontalAlignment(VerticalPanel.ALIGN_CENTER);
17         RootPanel.get().add(vPanel);
18
19         // Add image and button to the RootPanel
20         Image img = new Image(
21                 "http://code.google.com/w                    ;
22         Button button = new Button("Click
23         // We can add style names
24         button.addStyleName("pc-template-
25
26         vPanel.add(img);
27         vPanel.add(button);
28
29         button.addClickListener(new Click
30             public void onClick(Widget se
31                 // handle button click
32             }
33         });
34     }
35 }
36
```

Component Base

- Widgets
  - Button, Textbox, Tree, RichTextArea …
- Panels / Layout
  - DockPanel, HorizontalPanel, TabPanel …
- Custom Widget
  - Composite
- Widget Gallery

Event Driven

- Click Listener Example

```
public void anonClickListenerExample() {
  Button b = new Button("Click Me");
  b.addClickListener(new ClickListener() {
    public void onClick(Widget sender) {
      // handle the click event
    }
  });
}
```

- Keyboard Listener Example

```
public void adapterExample() {
  TextBox t = new TextBox();
  t.addKeyboardListener(new KeyboardListenerAdapter() {
    public void onKeyPress(Widget sender, char keyCode, int modifiers) {
      // handle only this one event
    }
  });
}
```

Simple RPC Call

- Both Client & Server Side
  - An service interface extends **RemoteService**

```
public interface StockPriceService extends RemoteService {
  StockPrice[] getPrices(String[] symbols);
}
```

- Server Side
  - A class extends **RemoteServiceServlet** and implement service interface

```
public class StockPriceServiceImpl extends
RemoteServiceServlet implements StockPriceService {
  public StockPrice[] getPrices(String[] symbols) {
    ...
  }
}
```

Simple RPC Call (Cont.)

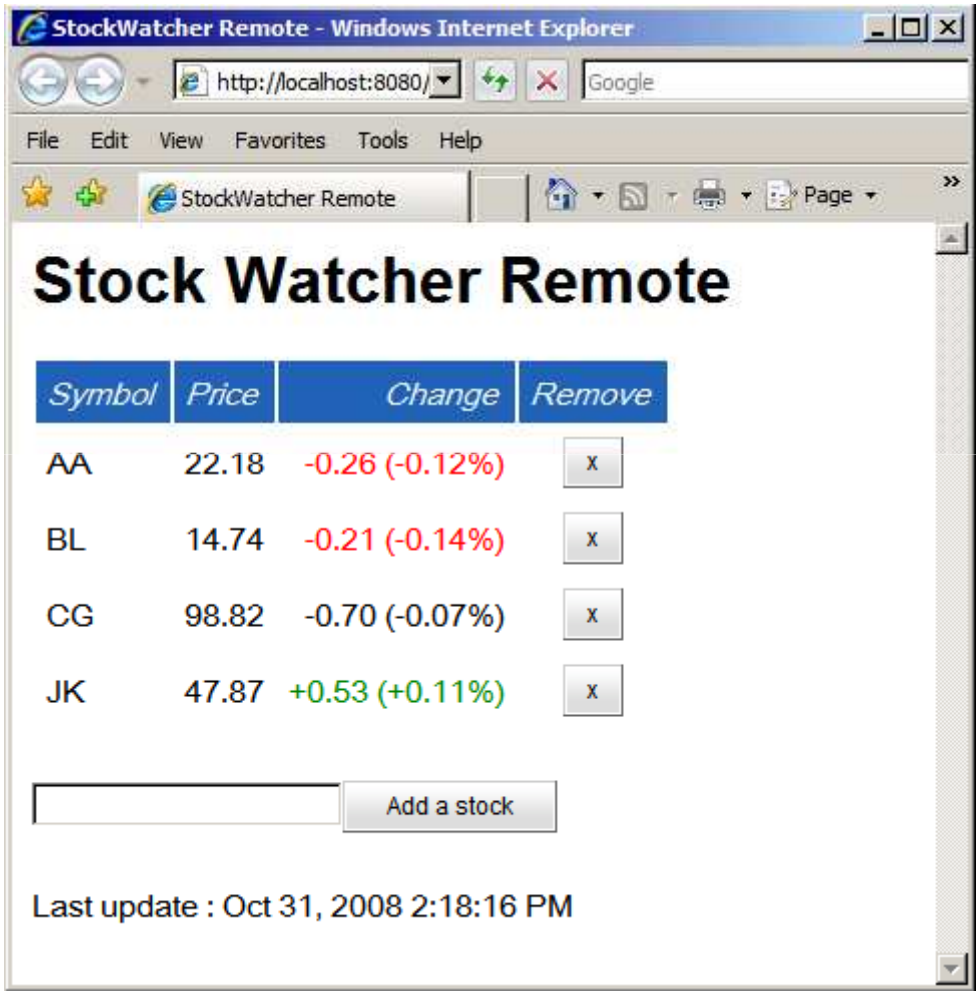
- Client Side (Write in Java, will be compiled into JavaScript)
  - A interface with API which is named according to service interface

```
public interface StockPriceServiceAsync {
  void getPrices(String[] symbols, AsyncCallback<StockPrice[]> callback);
}
```

  - Invoke RPC

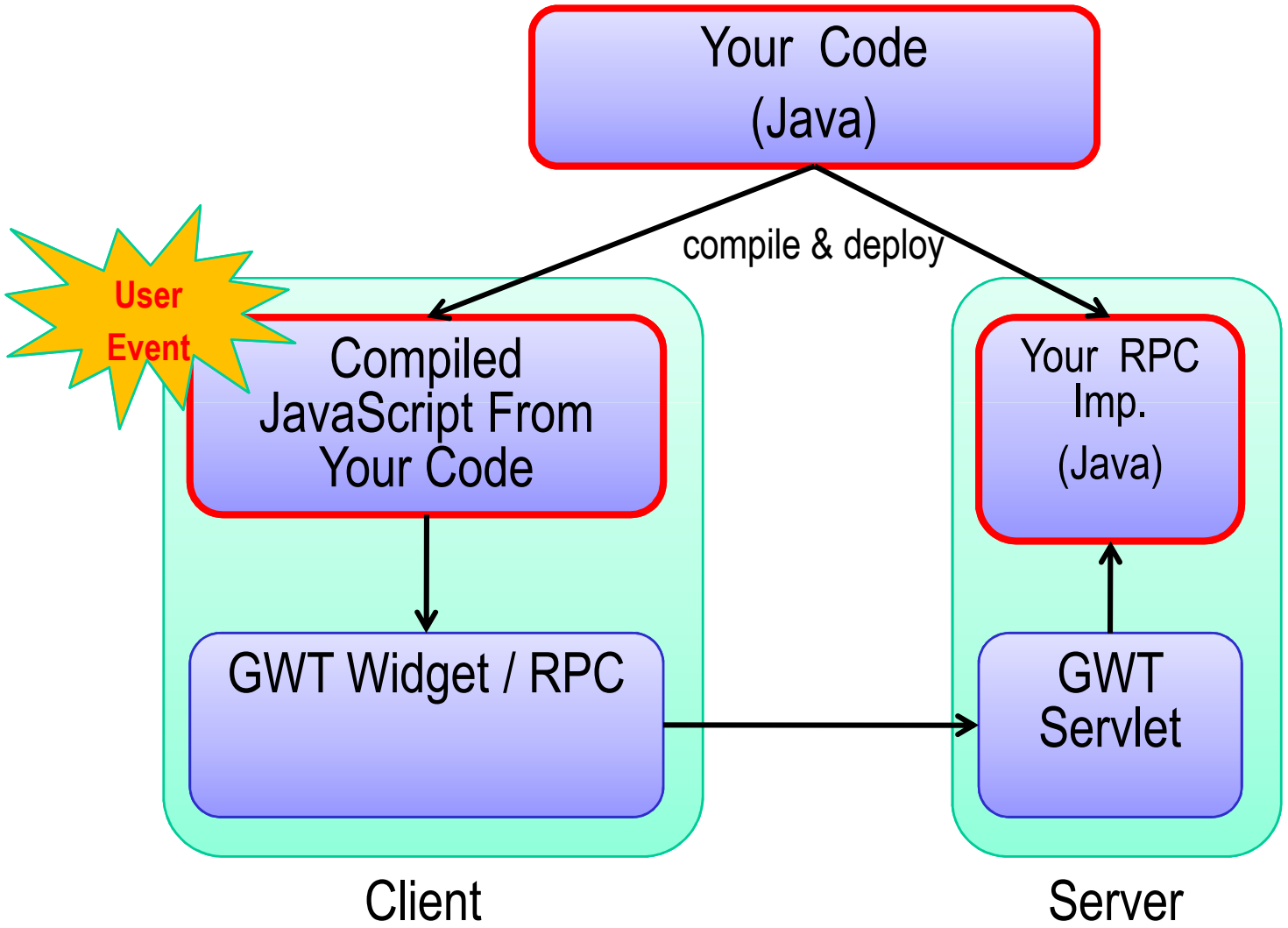
```
StockPriceServiceAsync stockPriceSvc =
  (StockPriceServiceAsync)GWT.create(StockPriceService.class);
AsyncCallback<StockPrice[]> callback = new AsyncCallback<StockPrice[]>() {
  public void onFailure(Throwable caught) {
    // do something with errors
  }
  public void onSuccess(StockPrice[] result) {
    // update the watch list FlexTable
  }
};
//in timer…
stockPriceSvc.getPrices(symbols, callback);
```

[Example] Stock Watcher (GWT)



Demo

Code Structure



It Helps, But…

- GWT helps You
  - Write Ajax Application by Java Code
  - do RPC, not http request
- You still have to
  - Know which code run in client, which run in server
    - Not all Java Code can compile into JavaScript
  - Integrate with your JEE Application
    - With HTML/JSP and other JavaScript.

ZK – Direct RIA

What is ZK

- An Open Source Server Centric Ajax Framework
- Pure Java (No need to write JavaScript)
- Event-driven, Component-base
- Markup & Scriptable
- Standards-based
  - XUL/XHTML
  - Java EE
- Extensible

What does the ZK code look like

- a XML/ZUML file (with .zul extension) handle all things.

hello.zul

```
<zk >
    <window title="My First Window" border="normal" width="300px">
        <hbox>
            Your Name:<textbox id="tb" value="Dennis"/>
        </hbox>
        <button label="Hi" onClick="doClick()" />

        <zscript><![CDATA[//@DECLARATION
            public void doClick(){
                alert("Hello "+tb.getValue());
            }
        ]]></zscript>

    </window>
</zk>
```

Demo

What does the ZK code look like (Cont. 1)

- ZUML as View

```
<window title="My First Window" border="normal" width="300px"
    id="mainWin" apply="demo.MainController">
    <hbox>
        Your Name:
        <textbox id="tb" />
    </hbox>
    <button id="btn" />
</window>
```

- Java Code as Controller

```
public class MainController extends GenericForwardComposer {
    Textbox tb;
    Button btn;

    public void onCreate$mainWin(Event evt){
        tb.setValue("Dennis");
        btn.setLabel("Hi");
    }

    public void onClick$btn(Event evt) throws Exception {
        Messagebox.show(tb.getValue());
    }
}
```

What does the ZK code look like (Cont. 2)

- Java Code handle all the stuff

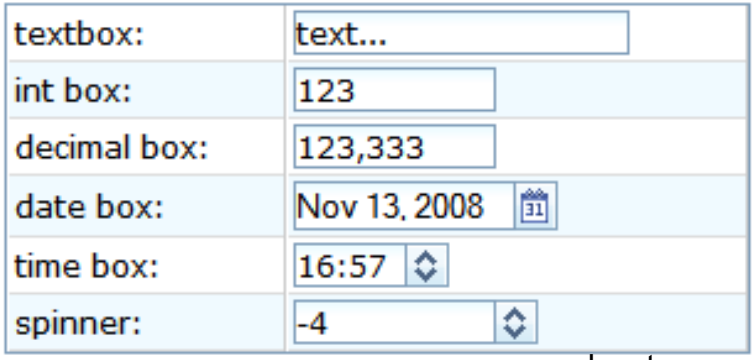
```
public class MyRichlet implements Richlet {

    public void service(Page page) {
        Window mainWin = new Window();
        mainWin.setId("mainWin");
        mainWin.setTitle("My First Window");
        mainWin.setBorder("normal");
        mainWin.setWidth("300px");
        mainWin.setPage(page);

        Button btn = new Button("Click me");
        btn.setParent(mainWin);
        btn.addEventListener("onClick",new EventListener(){
            public void onEvent(Event arg0) throws Exception {
                Messagebox.show("Hello");
            }
        });
    }

    //...
```

Markup Language & Scriptable

- Markup Language
  - <window title="Hello"/>
  - new Window().setTitle("Hello");
- Scriptable
  - <zscript>doSomething()</zscript>
  - <button onClick="doSomething()"/>
  - Language : Java / JavaScript / Ruby/ Groovy
- Easy to build UI template
- Fast for prototyping

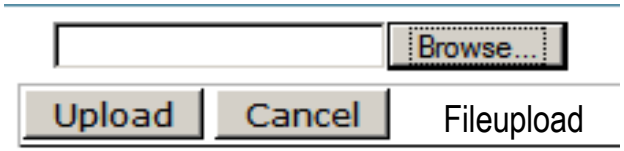
ZK Explorer

Rich Component Set

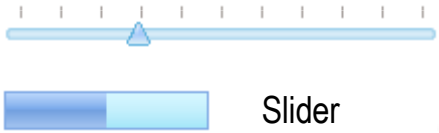


Inputs

Combobox

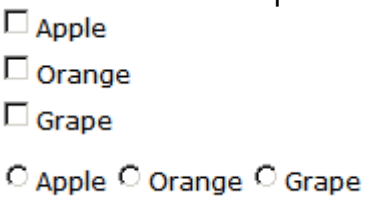


Fileupload

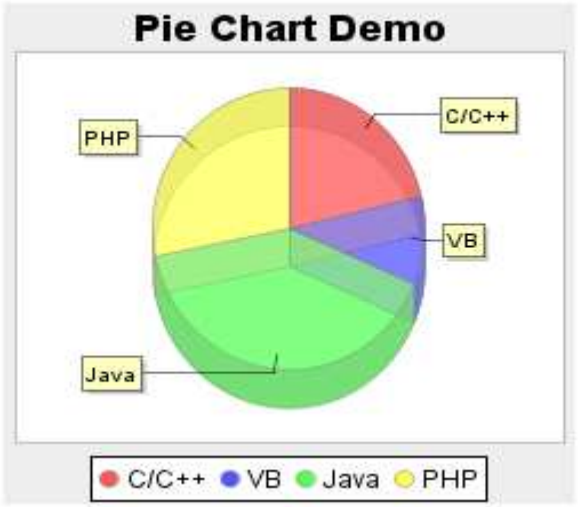


Slider

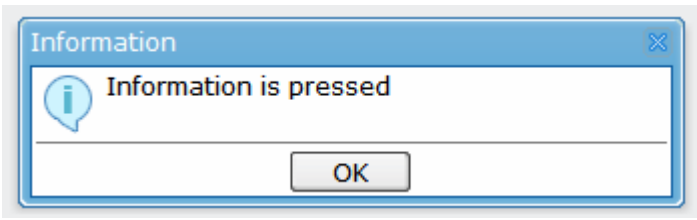
Captcha



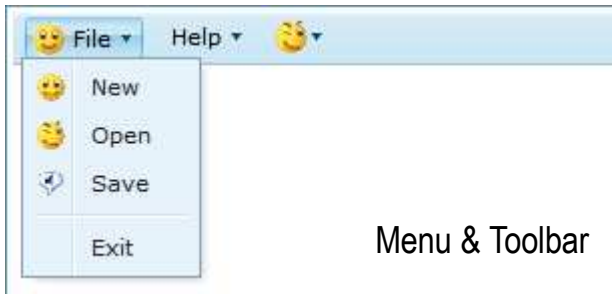
Checkbox & Radio



Chart



Window



Menu & Toolbar

Rich Component Set (Cont. 1)

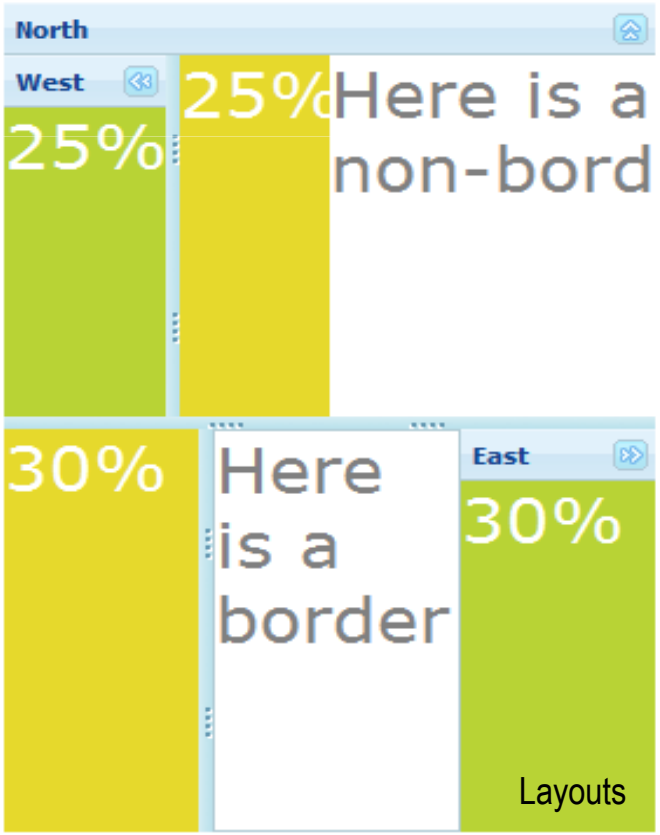
Tabbox

Name	Description
Item 1	Item 1 description
Item 2	Item 2 description
Item 2.1	
Item 2.1.1	
Item 2.1.2	
Item 2.2	Item 2.2 is something who cares
Item 3	

Tree

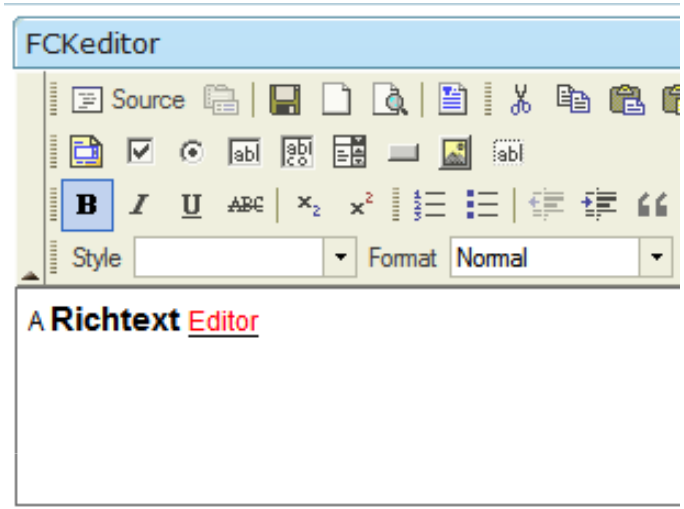
From	Subject	Received	Size
Date:Today			
[From]	[Subject]	[Received]	[Size]
SourceForge.net	[zk1 - Help] RE: ZK problem in dynamic menu	2008/4/29 15:47:27	12KB
SourceForge.net	[zk1 - Help] RE: FileUpload	2008/4/29 15:37:57	14KB
SourceForge.net	[zk1 - Help] RE: Datebox format	2008/4/29 15:27:31	11KB
SourceForge.net	[zk1 - Help] FileUpload	2008/4/29 14:07:25	11KB
SourceForge.net	[zk1 - General] RE: Opening more than one new browser window	2008/4/29 13:44:17	12KB
5 emails	zk1	2008/4/29	60KB
Date:Yesterday			
SourceForge.net	[zk1 - Help] RE: SelectedItemConverter Question	2008/4/28 13:31:12	14KB
SourceForge.net	[zk1 - Help] RE: Times_Series Chart help	2008/4/28 13:26:37	14KB
SourceForge.net	[zk1 - Help] RE: SelectedItemConverter Question	2008/4/28 10:14:27	14KB
3 emails	zk1	2008/4/28	42KB
Date:Sunday			
SourceForge.net	[zk1 - General] RE: Opening more than one new browser window	2008/4/27 13:44:17	12KB

Grid & Listbox



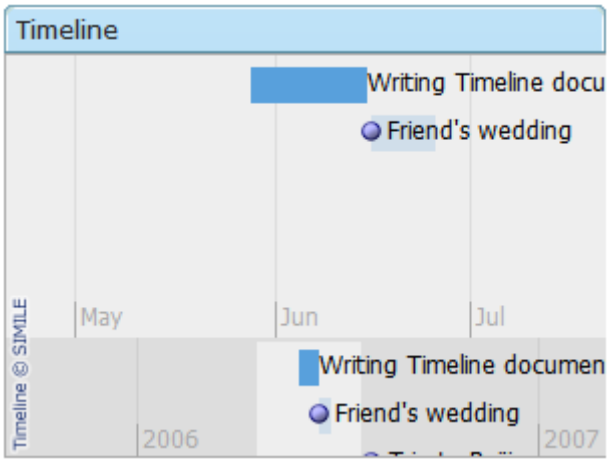
Layouts

Rich Component Set (Cont. 2)



FCKEditor

Google map



Timeline

Rich Component Set (Cont. 3)

Quarter: Quarter 3 Style: Style 1

	A	B	C	D
1	**Year**	**Financial Data of 2007**		
2	**Item**	**Quarter 3**		
3	Liquid Assets	187,967,615		
4	Fund and Investment	27,159,093		
5	Fixed Assets	7,047,513		
6	Intangible Asset	182,348		
7	Other Assets	2,166,805		
8	**Total Assets**	224,523,374		
9	Current Liabilities	142,212,438		
10	Long Term Liabilities	3,000		
11	Other Liabilities	717,980		
12	**Total Liabilities**	142,933,418		
13	Capital Stock	34,752,682		
14	Capital Surplus	7,234,727		
15	Retained Earnings	34,300,237		
16	Other Equity	4,667,500		
17	Treasury Stock	634,810		
18	**Shareholders Equity**	81,589,956		

Spreadsheet

Event Driven

- Component Events (name start with ‘on’)
  - onClick, onSelect, onChange, …
  - In ZMUL

```
<button label="Hi" onClick="doClick()" />
<zscript><![CDATA[//@DECLARATION
    public void doClick(){
        alert("Hello "+tb.getValue());
    }
]]></zscript>
```

  - In Java

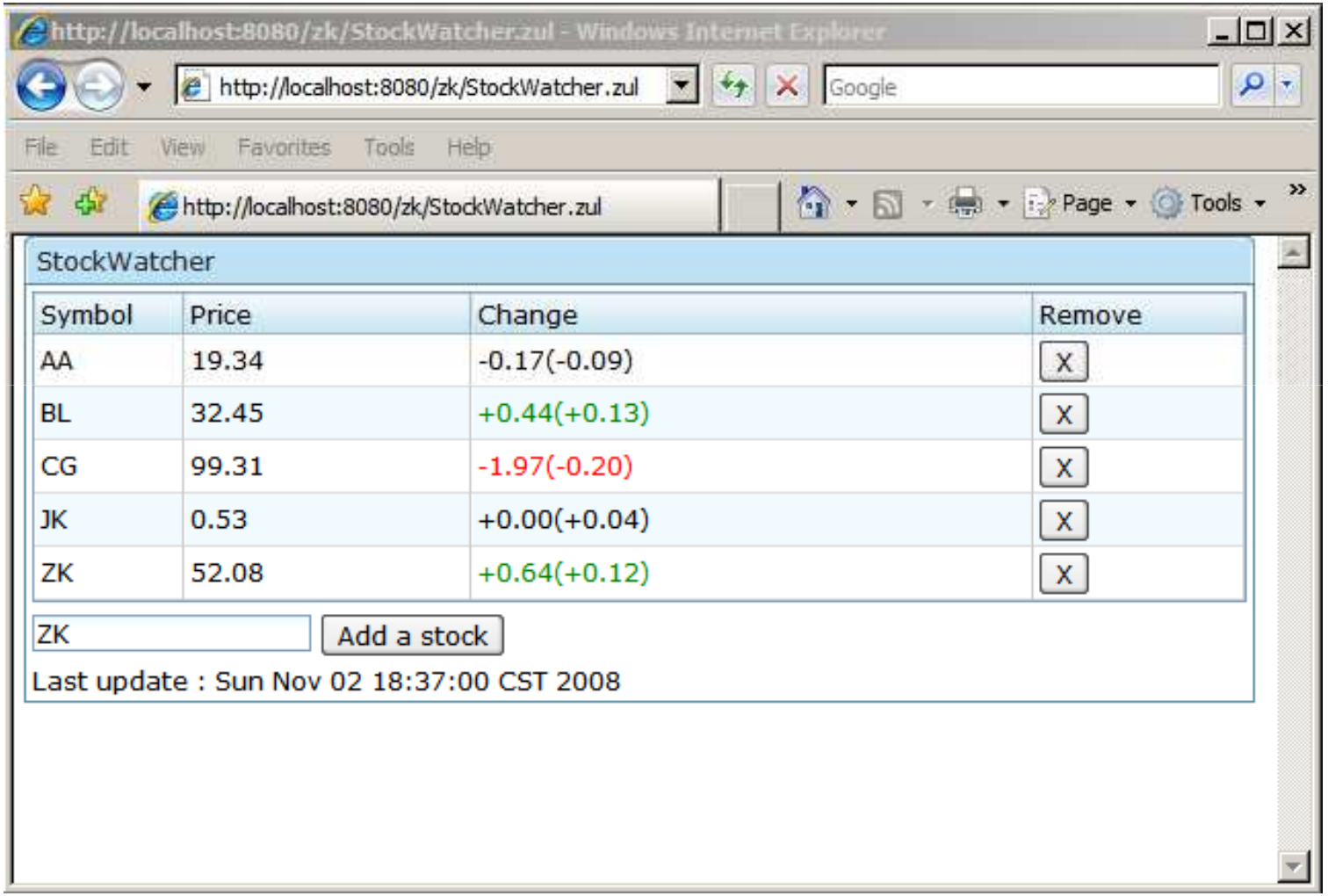
```
btn.addEventListener("onClick",new EventListener(){
    public void onEvent(Event arg0) throws Exception {
        Messagebox.show("Clicked ");
}});
```

- Fire the Event
  - Events.post(“onClick”,target,data);
  - Events.send(“onClick”,target,data);

No need to do RPC

- Accessing Object / API / Resource Directly
- Use JDBC / Hibernate / JPA Directly
- Spring Integration
  - Context Management
  - IOC
  - Security

[Example]Stock Watcher (ZUML)

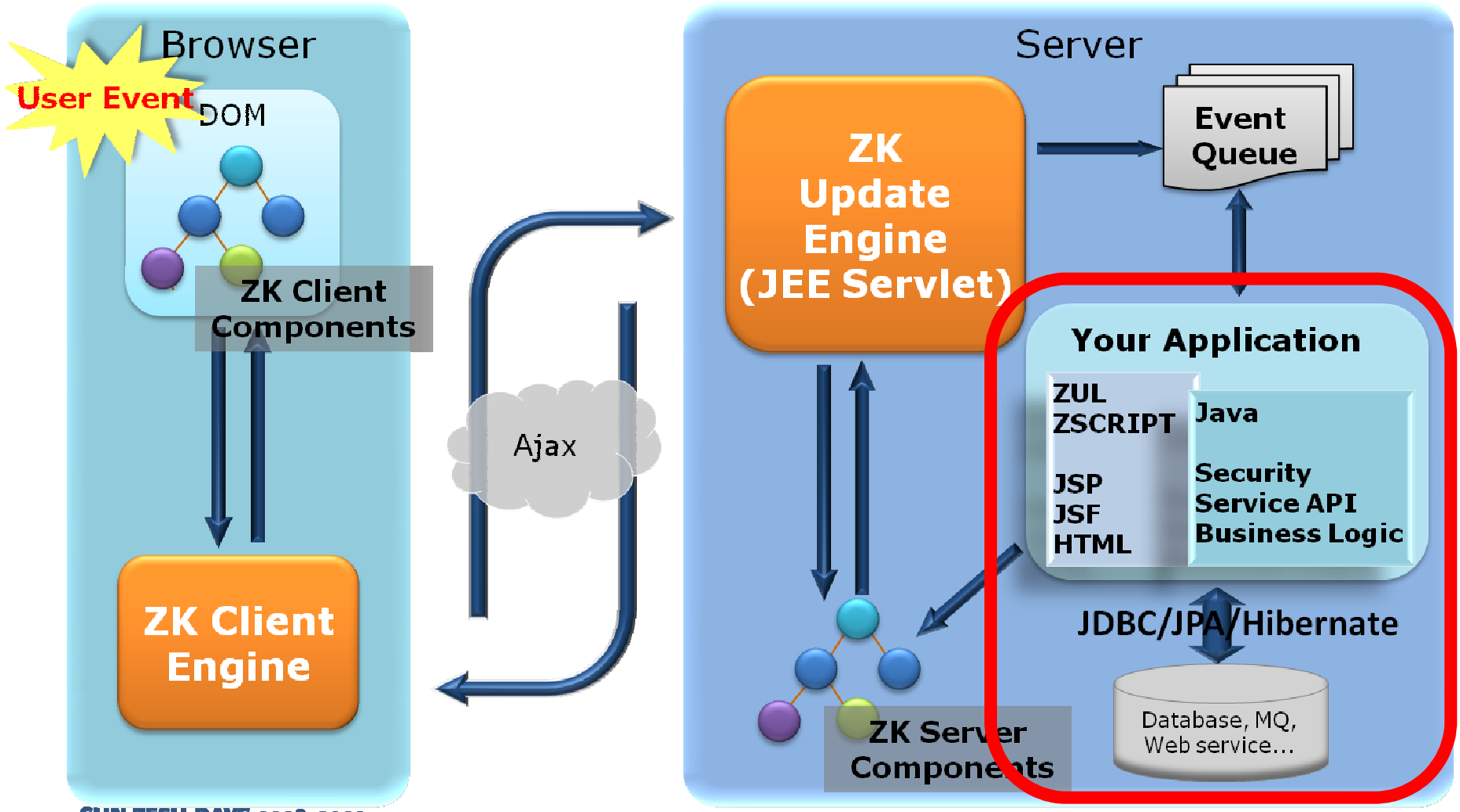


Demo

```
<zk>
    <style>
    .negativeChange{
        color:red;
    }
    .positiveChange{
        color:green;
    }
    </style>
    <window title="StockWatcher" width="600px" border="normal">
        <zscript>
        //...
        </zscript>
        <vbox width="100%">
        <grid>
            <columns>
                <column>Symbol</column>
                <column>Price</column>
                <column>Change</column>
                <column>Remove</column>
            </columns>
            <rows id="rows">

            </rows>
        </grid>
        <hbox><textbox id="symbol"/><button label="Add a stock"
            onClick="symbols.add(symbol.getValue());updateStock()"/></hbox>
        <label id="last"/>
        </vbox>
        <timer delay="1000" repeats="true" onTimer="updateStock()"/>
    </window>
</zk>
```

```
<zscript><![CDATA[//@DECLARATION
DecimalFormat priceFormat = new DecimalFormat("#,##0.00");
DecimalFormat changeFormat = new DecimalFormat("+#,##0.00;-#,##0.00");
StockService service = new StockService();
ArrayList symbols = new ArrayList();
public void updateStock(){
    StockPrice[] sp = service.getPrice((String[])symbols.toArray(new String[0]));
    rows.getChildren().clear();

    for(int i=0;i<sp.length;i++){
        Row row = new Row();
        new Label(sp[i].getSymbol()).setParent(row);

        String priceText = priceFormat.format(sp[i].getPrice());
        String changeText = changeFormat.format(sp[i].getChange());
        String changePercentText = changeFormat.format(sp[i].getChangePercent());

        new Label(priceText).setParent(row);
        Label chg = new Label(changeText+"("+changePercentText+")");
        chg.setParent(row);
        //....
        final Button rm = new Button("X");
        rm.setAttribute("symbol",sp[i].getSymbol());
        rm.addEventListener("onClick", new org.zkoss.zk.ui.event.EventListener(){
            public void onEvent(Event evt){
                symbols.remove(rm.getAttribute("symbol"));
                updateStock();
            }
        });
        rm.setParent(row);
        row.setParent(rows);
    }
    last.setValue("Last update : "+new Date().toString());
    rows.invalidate();
}
]]>
```

Architecture



Server Push

- Reverse-Ajax, Server send content to the client actively.

```
<zscript><![CDATA[
    desktop.enableServerPush(true);
    new Thread(){
        public void run(){
            while(true){
                sleep(1500);
                Executions.activate(desktop);
                updateStock();
                Executions.deactivate(desktop);
            }
        }
    }.start();
]]></zscript>
```

Component Extension - Macro Component

- Any ZUL page can become a Macro Component

mainpanel.zul

```
1  <?component name="stockwatcher" macroURI="StockWatcher.zul"?>
2  <zk>
3      <columnlayout width="900px">
4          <columnchildren width="40%" style="padding: 5px">
5              <panel height="100px" framable="true" title="column1-2"
6                  border="normal" maximizable="true" style="margin-bottom:10px">
7                  <panelchildren>Panel</panelchildren>
8              </panel>
9              <panel height="100px" title="column1-3" border="normal"
10                 closable="true">
11                 <panelchildren>Panel</panelchildren>
12             </panel>
13         </columnchildren>
14         <columnchildren width="60%" style="padding: 10px">
15             <panel style="margin-bottom:10px" title="column1-1"
16                 border="normal" maximizable="true" collapsible="true">
17                 <panelchildren>
18                     <stockwatcher />
19                 </panelchildren>
20             </panel>
21             <panel title="Data" maximizable="true" border="normal"
```

column1-2

Panel

column1-3

Panel

column1-1

StockWatcher

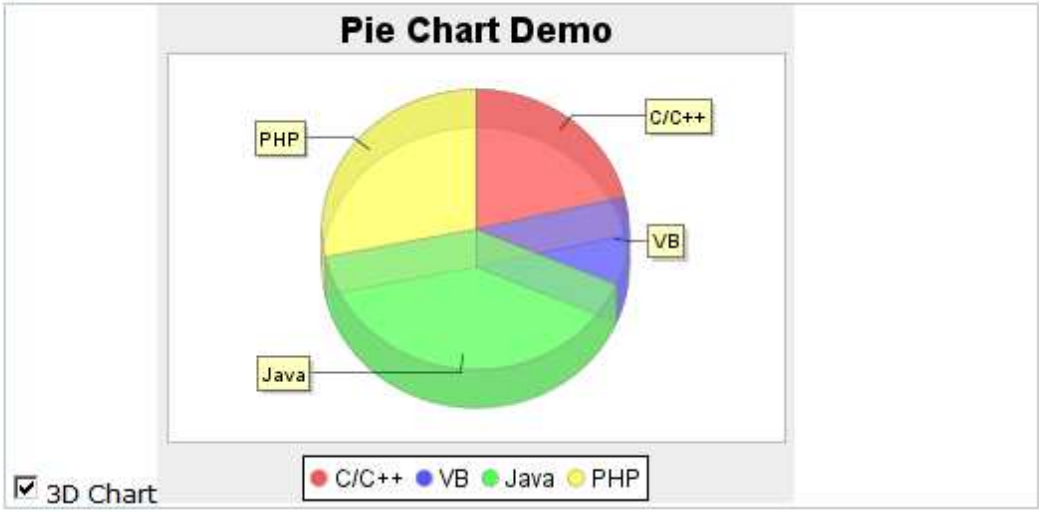
Symbo	Price	Change	Remove
ab	14.99	-0.23(-0.16)	X

cde Add a stock

Last update : Mon Nov 03 20:39:44 CST 2008

Data

category	value
C/C++	21.2
VB	10.2
Java	40.4
PHP	28.2



Demo

Integrate with HTML

- .zhtml
  - use namespace '*http://www.zkoss.org/2005/zul*' to mashup zul

register.zhtml

```
<html xmlns:z="http://www.zkoss.org/2005/zul">
<head></head>
<body>
    Register a account (HTML)
    <form action="registeraction.jsp">
    <table>
        <tr><td>Name</td>
        <td><z:textbox name="username"
            onChange='if("dennis".equals(self.value)) throw new WrongValueException(self,"Account Exist!")'/>
        </td></tr>
        <tr><td>Password</td><td><z:textbox name="password" type="password"/></td></tr>
    </table>
    <div>
        <input type="submit" value="Login"/>
    </div>
    </form>
</body>
</html>
```

Demo

Integrate with JSP

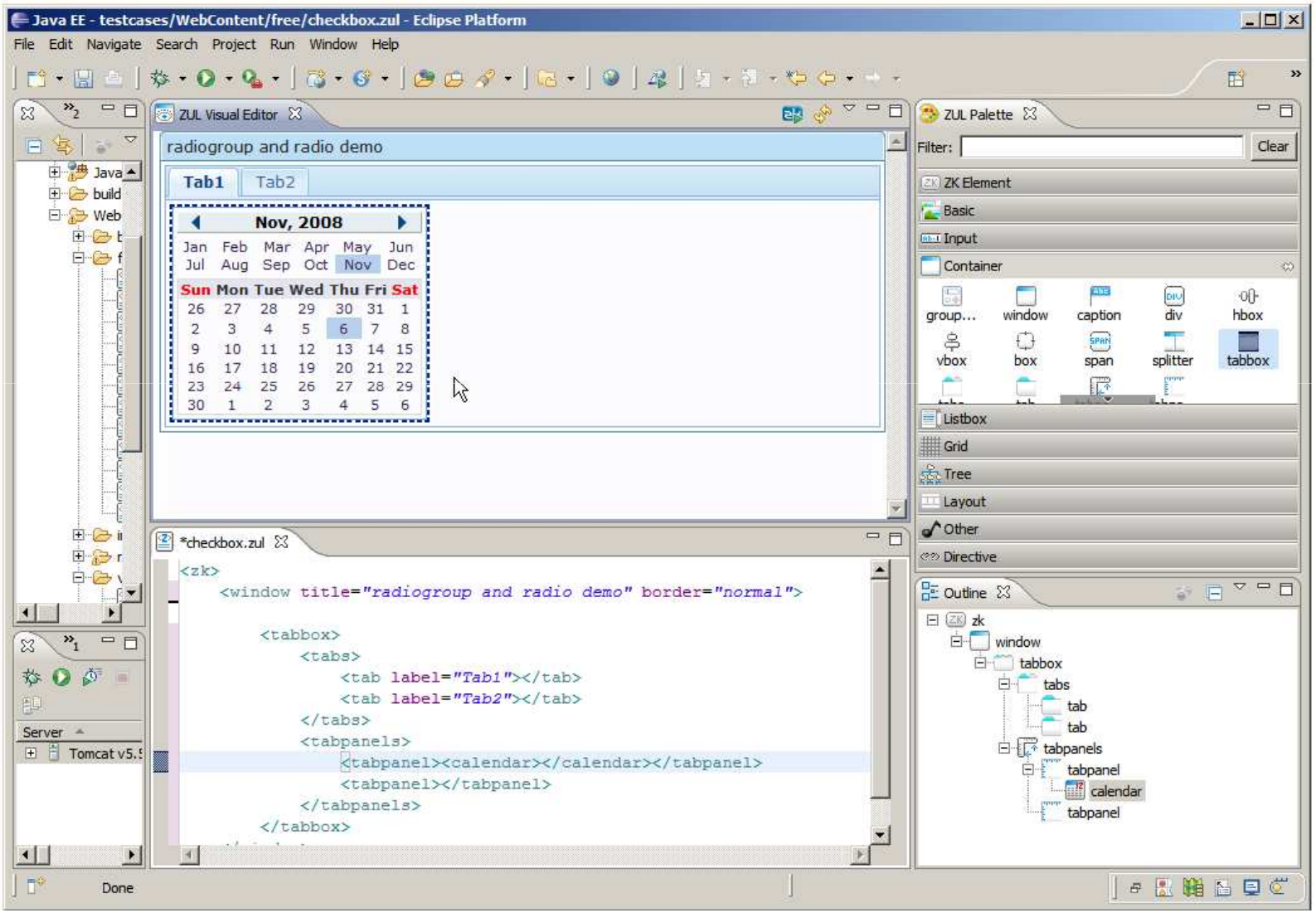
- Use tag lib: http://www.zkoss.org/jsp/zul

*register.jsp

```
<%@ taglib uri="http://www.zkoss.org/jsp/zul" prefix="z" %>
<head></head>
<body>
<%out.println("Register a account (JSP)");%>
<z:page>
    <form action="registeraction.jsp">
    <table>
        <tr><td>Name</td>
        <td><z:textbox name="username"
            onChange='if("dennis".equals(self.value)) throw new WrongValueException(self,"Account Existed!")'/>
        </td></tr>
        <tr><td>Password</td><td><z:textbox name="password" type="password"/></td></tr>
    </table>
    <div>
        <input type="submit" value="Login"/>
    </div>
    </form>
</z:page>
</body>
</html>
```

Demo

ZK Studio – WYSIWYG IDE



Why ZK

- Direct RIA
  - Simple programming model
  - Direct access database, enterprise resource
  - 200+ Ajax components
  - WYSIWYG visual Ajax editor
  - Mobile access
- Enterprise
  - High security
  - Scalability, clustering and failover
  - Enterprise environment integration
  - Customizability
- Open
  - Open source
  - Standards base
  - 800,000+ downloads

SUMMARY

Summary

- jQuery
  - A client side JavaScript toolkit to help developer manipulate HTML
  - Provide a lots of utility function on JavaScript
- DWR
  - A toolkit to wrap server side Java API into a client side JavaScript RPC API
- GWT
  - A client side Ajax framework, developer writes Java, the code is compiled to be running in JavaScript.
  - Component base, use RPC to access Server Side API
- ZK
  - A server centric Ajax framework, pure java, developer no need to write JavaScript.
  - Rich UI component set, markup language, scriptable, fast for prototyping extensible , easy to access any server side API/resource/service.
  - Easy to integrate with other Java /JEE standard.


THANK YOU
Sun microsystems
SUN TECH DAYS 2008–2009
A Worldwide Developer Conference