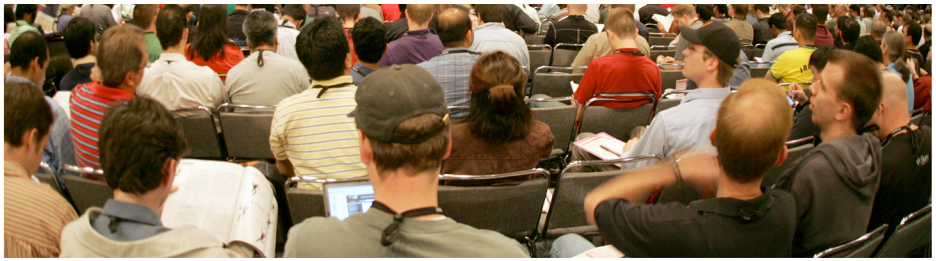


JavaOne

Project jMaki: Web 2.0 App Building Made Easy

Carla Mott

Greg Murray

Sun Microsystems
http://jmaki.com

TS-6375

Goal

Learn how to build AJAX-enabled Web 2.0 applications using Project jMaki framework.

Agenda

What Is Project jMaki?

- Project jMaki Core
- Project jMaki Widgets

Project jMaki Recipe

Demo

What Is Project jMaki?

- Lightweight framework
- Make JavaScript™ technology accessible to Java technology, PHP, Ruby
- Widgets from popular toolkits
  - Dojo, Yahoo, Script.aculo.us, Spry, Google
- Tool support
- Future Protection

Why Use jMaki

- Convention over configuration
- Generic Component libraries
  - Share with project, organization or company
- Tooling Support
- Standardizes Data Model
- Normalized JavaScript technology toolkit APIs
- Server/Client Integration—Standardize Java™ code/PHP to JavaScript programming language on client

Multi-Server Support

JSP technology: index.jsp

```
<%@ taglib prefix="a" uri="http://jmaki/v1.0/jsp" %>
<a:widget name="hello" args="{name: 'Greg'}" />
```

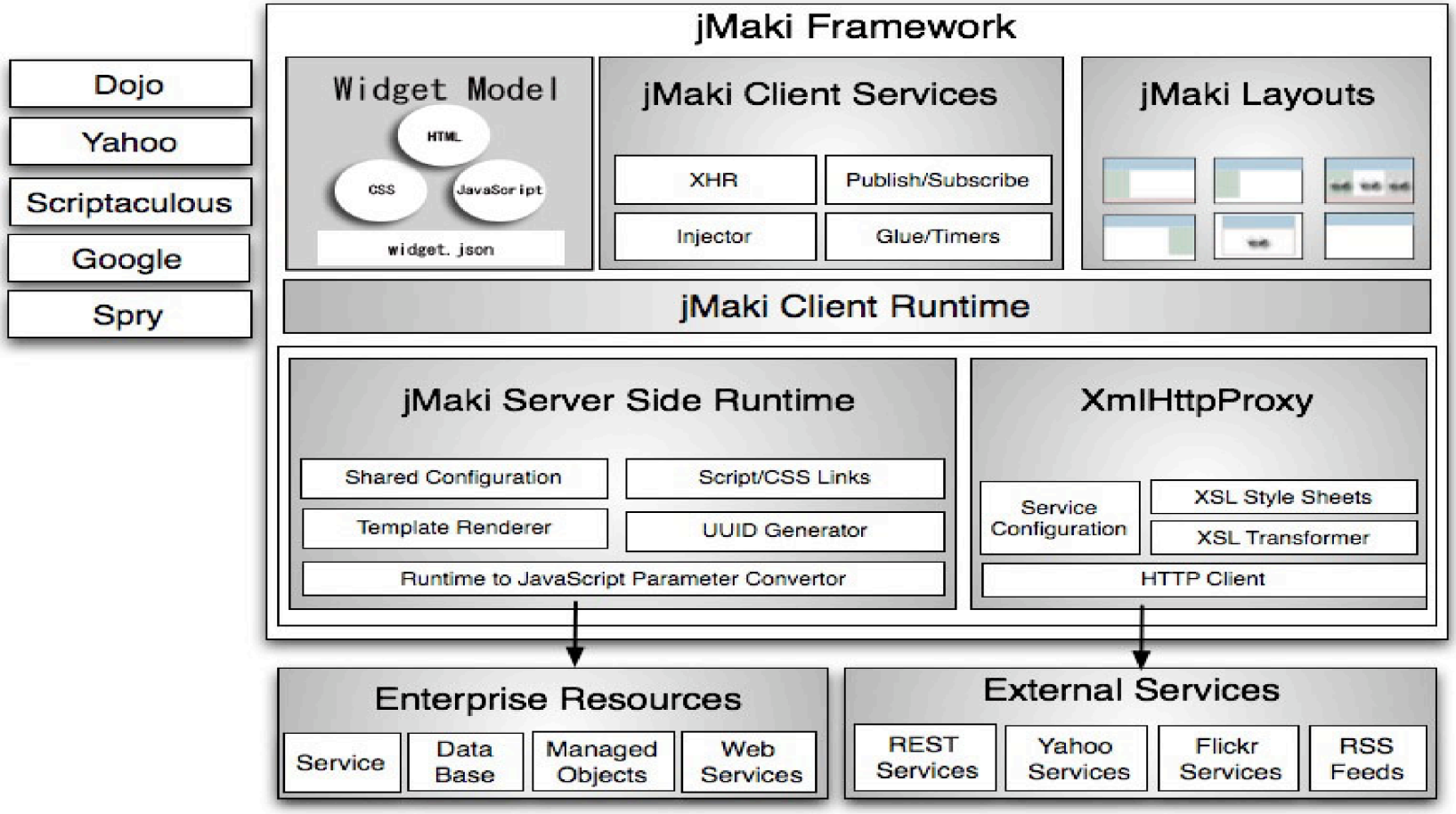
PHP: index.php

```
<?php require_once 'Jmaki.php'; ?>
<?php addWidget("hello", null, "{name: 'Greg'}"); ?>
```

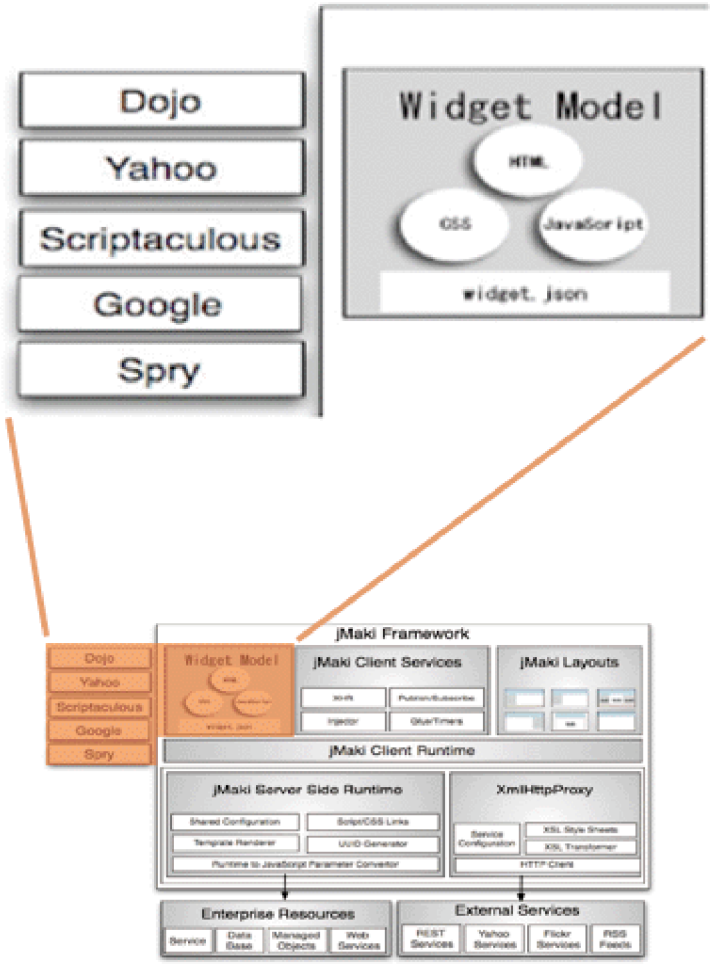
Phobos index.ejs

```
<% library.jmaki.insert({
   component:"hello",
   args: {name: 'Greg'}
}); %>
```

Project jMaki Architecture

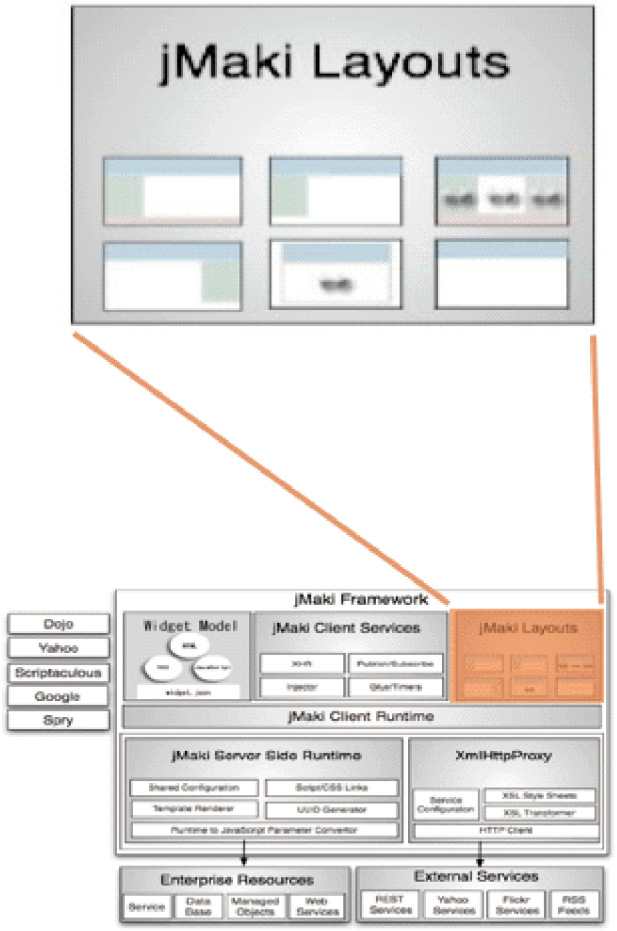


Widget Model



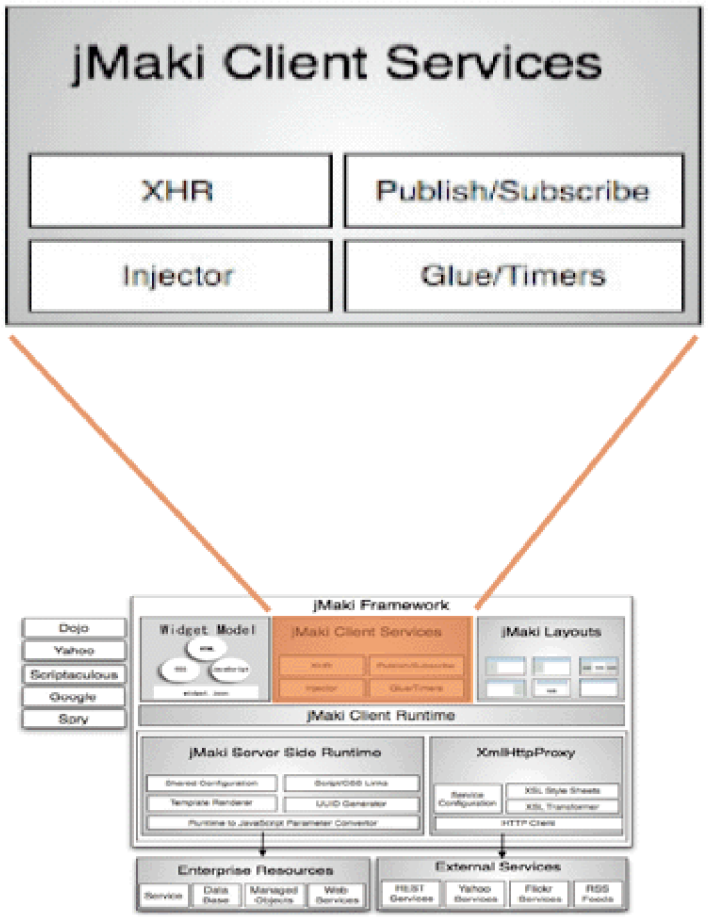
- Reusable JavaScript technology component
- Common API for all widgets
- Described by HTML, CSS, JavaScript technology and JSON files
- Widget dependency information in widget.json

Layout



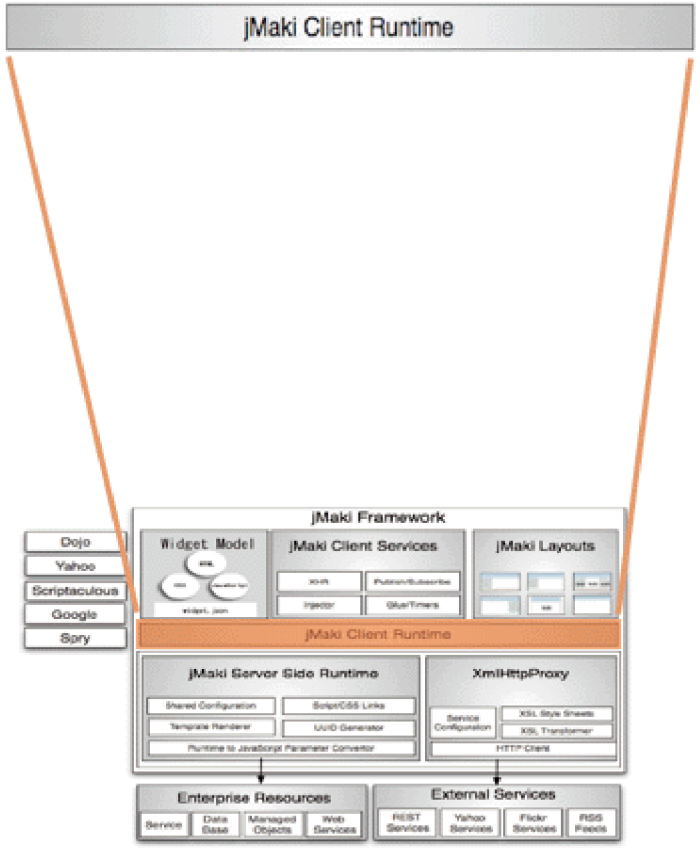
- Standard based CSS and HTML
- Templates provided

Client Services



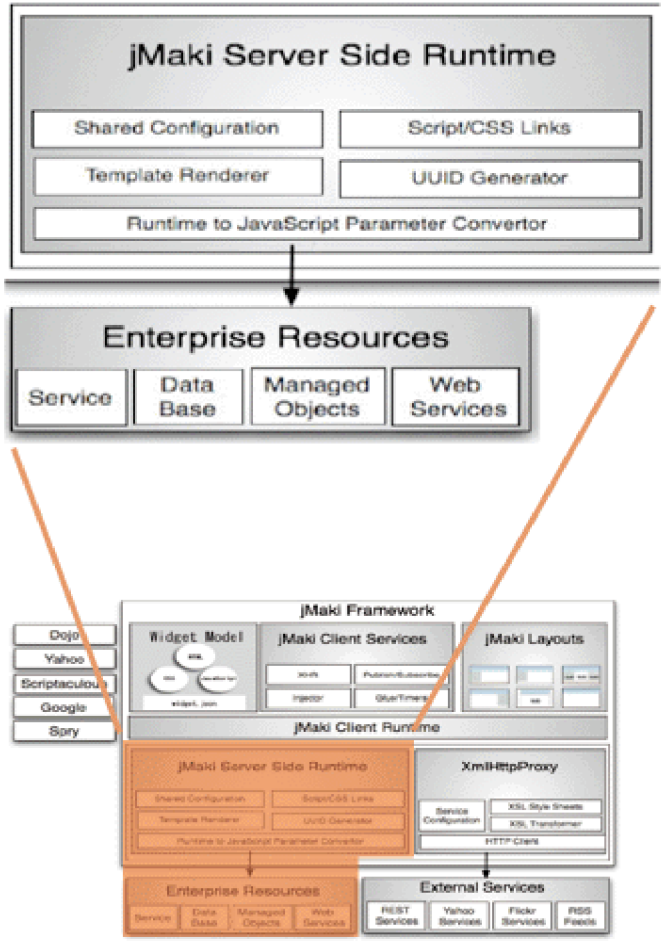
- Glue wires widgets together
  - Based on publish and subscribe mechanism
  - Event bus on client
- Injector loads the contents of external page in current page
- Filter RSS data from web services

Client Runtime



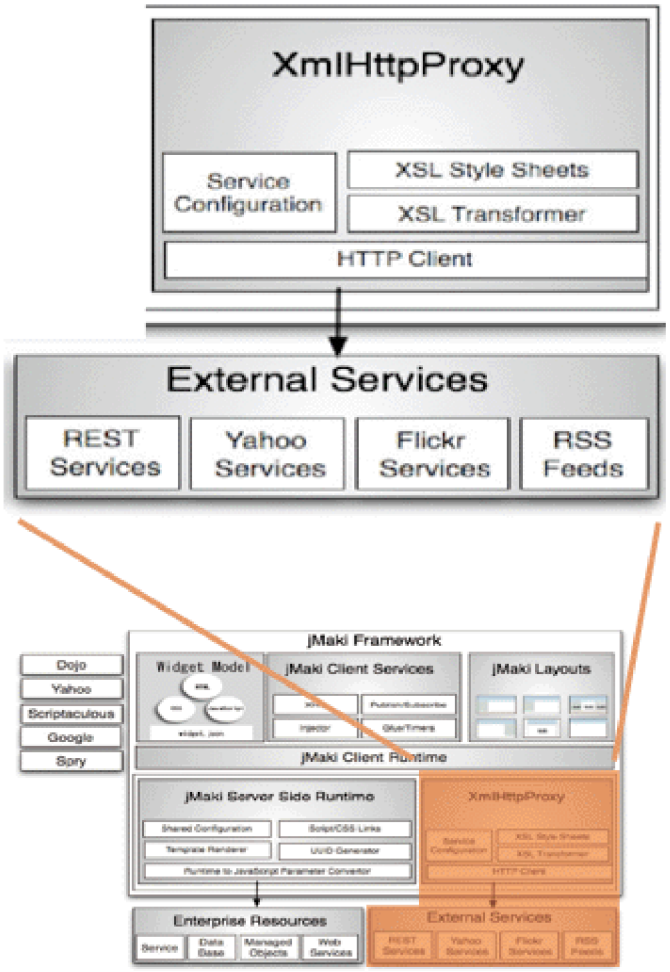
- Bootstrap widgets
- Track widgets on page
- Pass parameters to appropriate widgets from server side

Server Side Runtime



- Ties client runtime to server side technology
- Renders script/CSS based on library types
- Handles data conversion as needed

XmlHttpProxy



- Access to RESTful web services not in the web app domain
- Provides customizable XSL-to-JSON transformations
- RSS Data – *Lingua Franca*

Project jMaki Models

- Tabbed Views
- Trees
- Accordions
- Menus
- Data Grids/Tables

Learn one model and use any widget implementation

Tabbed Views Model

```
{
  'tabs' :[
            {'label' :'My Tab', 'content' : 'Some Content'},
            {'label' :'My Tab 2', 'url' : 'tab1.jsp'}
          ]
}
```

Trees Model

```
{
        'root' : {
           'title' : 'Yahoo Tree Root Node',
           'expanded' : true,
           'children' : [
             { 'title' : 'Node 1.1'},
             { 'title' : 'Node 1.2',
                 'children' : [
                   { 'title' : 'Node 3.1',
                     'onclick' : {'url' :'foo'}}
                 ]
             }
           ]
         }
        }
```

Accordion Model

```
{'rows' :
    [
     {'label':'Books','content':'Book content'},
     {'label':'Magazines','content':'Magazines here'},
     {'label':'Newspaper','url' : 'foo.jsp'}
    ]
}
```

Menu Model

```
{
 columns: [
   { 'label': 'Images',
     'menuItems' : [
       {label:'Birds', url: 'jsonibrowse.jsp'},
       {label:'Cat', url: 'ibrowse.jsp'}
     ]
    },
    {label: 'Bookmarks',
      'menuItems' : [
        {label:'Digg', url:'digg.jsp'},
        {label:'Delicious', url:'delicious.jsp'}
      ]
    }
  ]
}
```

Data Grids/Table Model

```
{
   'columns' :[
        {'title' : 'Company', 'width' : 200, locked:false},
        {'title' : 'Price', 'width' : 75, 'renderer': 'usMoney'},
        {'title' : 'Change', 'width' : 75, 'renderer': 'change'},
        {'title' : '% Change', 'width' : 75, 'renderer': 'pctChange'},
        {'title' : 'Last Updated', 'width' : 85, 'renderer': 'italic'}
        ],
       rows : [
        ['3m Co',71.72,0.02,0.03,'9/1 12:00am'],
        ['Alcoa Inc',29.01,0.42,1.47,'9/1 12:00am'],
        ['Altria Group Inc',83.81,0.28,0.34,'9/1 12:00am']
    ]
}
```

Layout and Templates

- We are a cut-and-paste generation
- Gives you a place to put your widgets
- Uses standard approach
- Web Designer friendly
- Easy to customize

CSS Templates POCH

(Plain Old CSS + HTML)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1transitional.dtd">
<link rel="stylesheet" href="jmaki-standard-footer.css"
type="text/css"></link>
<html>
    <div class="header">
      <div class="banner">Application Name</div>
    </div> <!-- header -->
 </body>
</html>
```

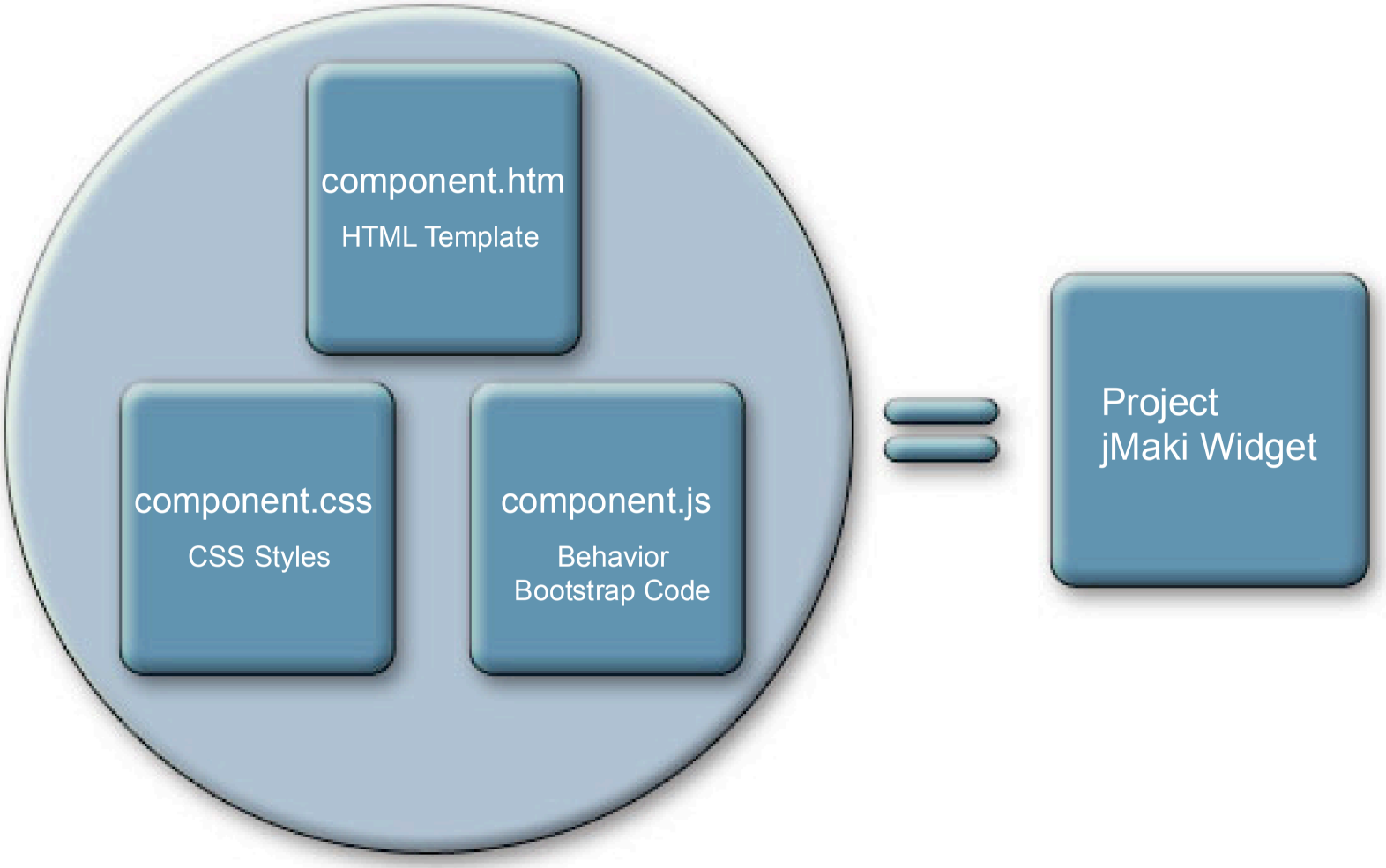
CSS

```
.header {
  height:100px;
  border: 1px solid #000000;
}

.main {
  position: relative;
  width: 100%;
  height:auto;
  margin-top: 5px;
}

.content {
  margin: 0 0 0 200px;
  height: auto;
  border: 1px solid #000000;
}
...
```

Widgets


component.htm
HTML Template
component.css
CSS Styles
component.js
Behavior
Bootstrap Code
Project
jMaki Widget

Hello World Widget

component.htm

```
<div id="${uuid}"></div>
```

component.js

```
jmaki.namespace("jmaki.widgets.hello");
jmaki.widgets.hello.Widget = function(wargs) {
  var mydiv = document.getElementById(wargs.uuid);
  mydiv.innerHTML = "Hello " + wargs.args.name;
}
```

index.jsp

```
<%@ taglib prefix="a"
        uri="http://jmaki/v1.0/jsp" %>
<a:widget name="hello" args="{name: 'world'}" />
```

Wrapping a Dojo Widget

```
dojo.require("dojo.widget.Editor");
// define namespace jmaki.dojo.editor
jmaki.dojo.editor.Widget = function(wargs) {
    var container =  document.getElementById(wargs.uuid);
    this.editor = dojo.widget.createWidget("Editor2",
        { shareToolbar: false, toolbarAlwaysVisible: true,
          focusOnLoad: false }, container);
    this.getValue = function() {
      return this.editor.getEditorContent();
    }
}
```

Configure Widgets

- id: You can override the automatic numbering
- value: In-line a value for simple cases
- service: Requires a corresponding server-side component
- args: JavaScript programming language Object Literal/Associative Array
- selected: For combo boxes or other widgets where a default selected items is needed

config.json or widget.json Files

- A central place for defining Project jMaki configuration information
  - JavaScript programming language library dependencies
  - API keys
  - CSS dependencies
  - Project jMaki Glue handler mappings

External Libraries

```
{"config": {
    "version": "1.2",
    "types": [
        {"id": "dojo",
         "version": ".4.1",
         "libs": ["/resources/libs/dojo/version.4.1/dojo.js" ],
         "preload" : "if (typeof djConfig == 'undefined')
          djConfig = { parseWidgets: false, searchIds: [] };"
        }
    ]
}
```

External Services (Google)

```
{"config": {"version": "1.2",
    "types": [
        {"id": "google.map",
         "libs": ["http://maps.google.com/maps?
                        file=api&amp;amp;v=2&amp;amp;key="],
         "apikey" : "google"
        ], "apikeys" : [
                {"id": "google", "keys": [
                        {"url": "http://localhost:8080/jmaki/",
                         "key": "xoxoxoxo"
                        }
                    ]
                }
         ]
    }
}
```

Publish/Subscribe

- A means for one or more Project jMaki widgets to communicate with each other in a page using topics
- Publish/Subscribe is much like a server-side messaging system by it runs in the scope of an HTML page

https://ajax.dev.java.net/publishsubscribe.html

Publish Example

```
icon.onClick = function (){
   jmaki.publish("/dojo/fisheye", this);
}
```

Subscribe Example

```
<script>
  function fisheyeListener(item) {
     var targetDiv = document.getElementById("newpage");
    var responseText = "";
    var index = Number(item.index);
     switch(index){
       case 1:  // Set responseText equal to Jayashri's bio
     break;
       case 2:  // Set responseText equal to Roberto's bio
     default: responseText += 'Click on one of the photos above';
     break;
     }
     targetDiv.innerHTML = responseText;
  }
  jmaki.subscribe("/dojo/fisheye", fisheyeListener);
</script>
```

Project jMaki Glue

- Allows you to provide un-obtrusive application behavior in a single place
- Based on publish/subscribe
- Ties the widgets together (loosely)
- Very similar to struts actions

Glue Example

```
jmaki.listeners.sendMessage = function(message) {
    jmaki.doAjax({
       method : "post",
       url : "words",
       content : {
          callback : "jmaki.CometClient.callback,
          action : "post",
          message: "{message : \"" + escape(message) + "\"
       }
}

jmaki.addGlueListener("/grizzly/message",
                              "jmaki.listeners.sendMessage");
```

Project jMaki Recipe

- Choose a Layout
- Add Widgets
- Configure services
- Add Glue
- Done!

DEMO

Community

- External commitors – global interest
  - core framework
  - widgets project
- Norbert wins GlassFish Champion award
- Widget Gallery with live examples
- Numerous blogs, articles, screencasts and tutorials
- Sample applications for Java and PHP
- Widget documentation online

Summary

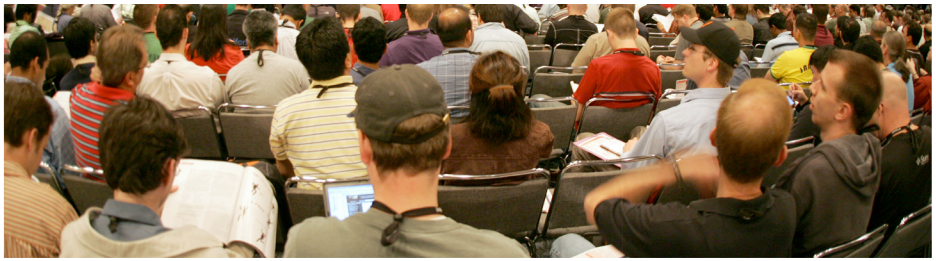
- Lightweight framework for building AJAX-enabled Web 2.0 applications
- Support for Java code, PHP, JavaScript programming language, and Ruby
- Tool support with NetBeans™ architecture and Eclipse
- Simple recipe

For More Information

See Project jMaki website

- http://jmaki.com
- TS-6676 Blueprints for Mashups: Particles Strategies Tips and Code for Designing and Building

Q&A



JavaOne

Projct jMaki: Web 2.0 App Building Made Easy

Carla Mott

Greg Murray

Sun Microssytems
http://jmaki.com

TS-6375