

# jMaki Web 2.0 Applications Made Easier

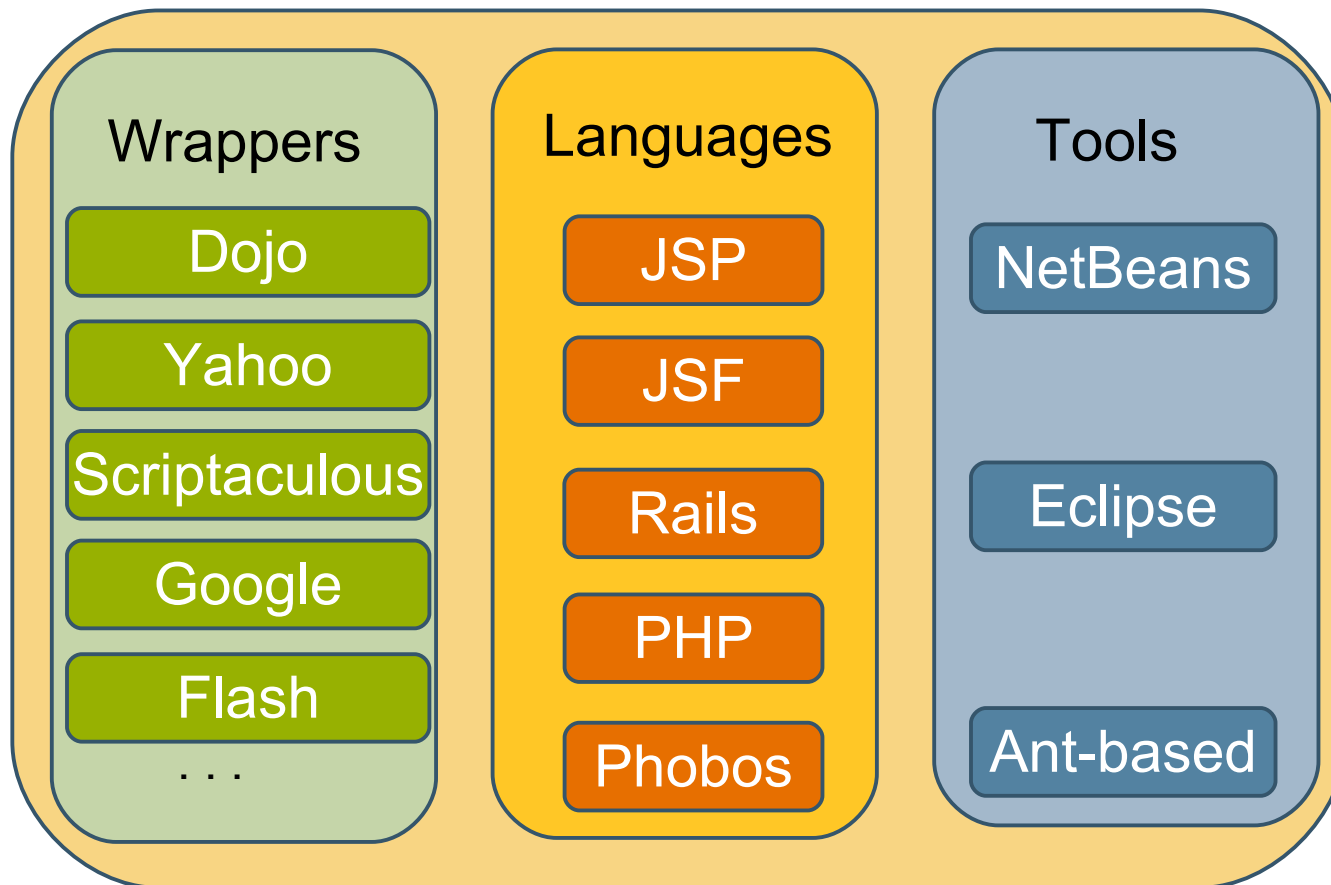


Arun Gupta  
<http://blogs.sun.com/arungupta>

# Goal

Build **Ajax-enabled** web applications following **standard practices** using the best **toolkits and libraries**.

# What Is Project jMaki?

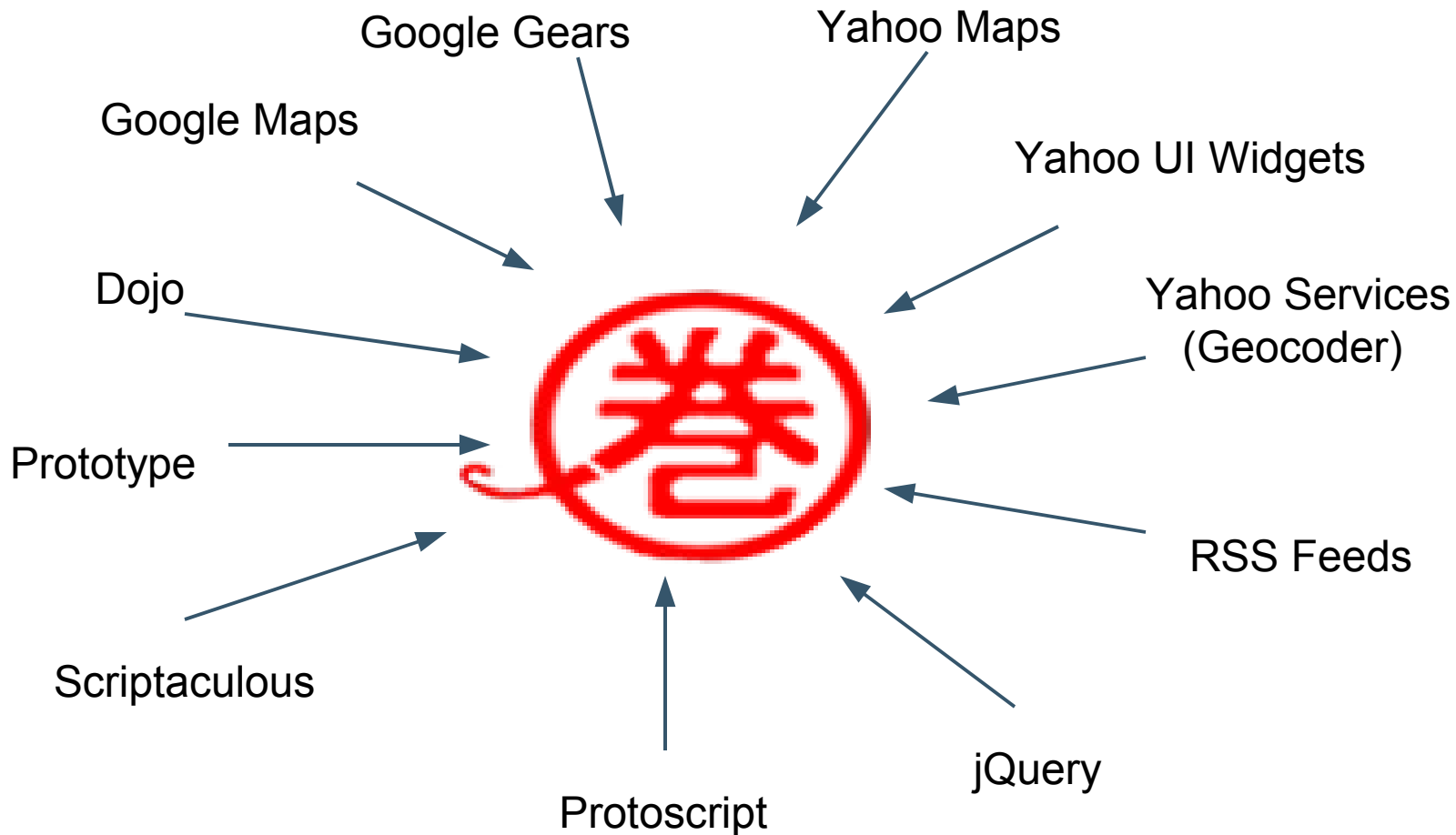


**jMaki – Web 2.0 Apps Your Way  
([jmaki.com](http://jmaki.com))**

# Why use jMaki ?

- Standardizes Event / Data Models
  - > Consistent programming model
- Defaults set out of the box
  - > Convention over configuration
- Portable Widget libraries
  - > Share your extensions / widgets libraries
- Use the tool of your choice
- Theming Support

# jMaki Ecosystem



# Multi-Server Support

## JSP technology: index.jsp

```
<%@ taglib prefix="a" uri="http://jmaki/v1.0/jsp" %>  
<a:widget name="yahoo.button" />
```

## Ruby: index.rhtml

```
<%= javascript_include_tag "jmaki" -%> <%= jmaki_setup -%>  
<%= jmaki_widget 'yahoo_button' %>
```

## PHP: index.php

```
<?php require_once 'Jmaki.php'; ?>  
<?php addWidget( name=>"yahoo_button"); ?>
```

## Phobos index.ejs

```
<% library.jmaki.insert(component: "yahoo_button"); %>
```

# jMaki Framework

- Widget component model
- Data Models
- Events = Actions / Glue
- Layouts / Theming model
- XmlHttpRequest
- Extensions

# Data Models

- Multi-View Containers
  - > Tabbed Views
  - > Accordions
  - > Dynamic Container
- Trees
- Menus
  - > Accordion Menu
  - > Tabbed Menu
- Data Grids/Tables

<http://wiki.java.net/bin/view/Projects/jMakiDataModels>

# Immediate Feedback

- Deploy only once
- Create, Edit JSPs/Ruby/PHP
- Save
- Reload page in browser
- Edit, save, test, edit, save, test
- Debug with FireBug / jMaki debugger as needed

# Demo

## Ease-of-use with NetBeans IDE

# Publish/Subscribe

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

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

# Declarative Actions – Publish

```

{menu : [
  {label: 'Actions', menu: [
    { label : 'Select',
      action :{topic: '/foo/select', message: { targetId : 'tab3'}}
    },
    { label : 'Set Content',
      action :{topic: '/foo/setContent',
        message: { targetId : 'tab3', value : 'Hello There!'}}
    },
    { label : 'Set Include',
      action :{topic: '/foo/setInclude',
        message: { targetId : 'tab3', value : 'main.jsp'}}
    }
  ]
}
] }

```

# Declarative Actions – Subscribe

```
subscribe="/foo"
```

```
{  
  items:[  
    { label : 'My Tab', content : 'Some Content'},  
    { id : 'tab2',  
      label : 'My Tab 2', include : 'tab1.jsp ', lazyLoad : true  
    },  
    { id : 'tab3',  
      label : 'My Tab 3', content : 'More Content', selected : true  
    }  
  ]  
}
```

# Demo

## Publish/Subscribe Demo

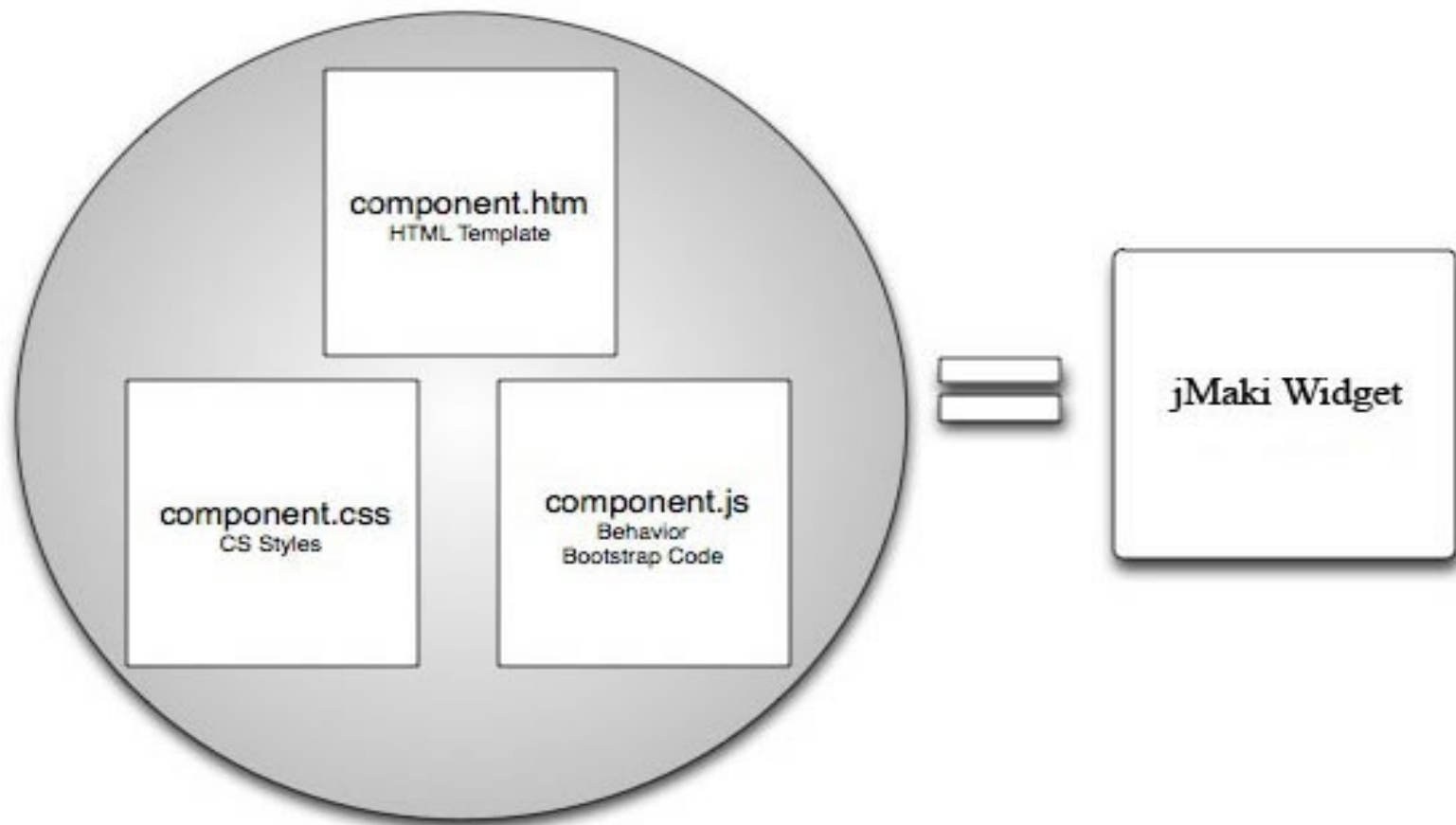
# jMaki Glue – Programmatic Events

- Allows you to provide un-obtrusive application behavior in a single place
- Based on publish/subscribe
- Ties the widgets together (loosely) based on topic names

# Glue Example

```
var node = {  
  label : 'Graft',  
  expanded : true,  
  children : [  
    {label : 'Graft .1'},  
    {label : 'Graft .2'}  
  ]  
};  
  
// publish the update  
jmaki.publish("/dojo/tree/addNodes", {targetId : 'foo', value : node});
```

# Widgets



# 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

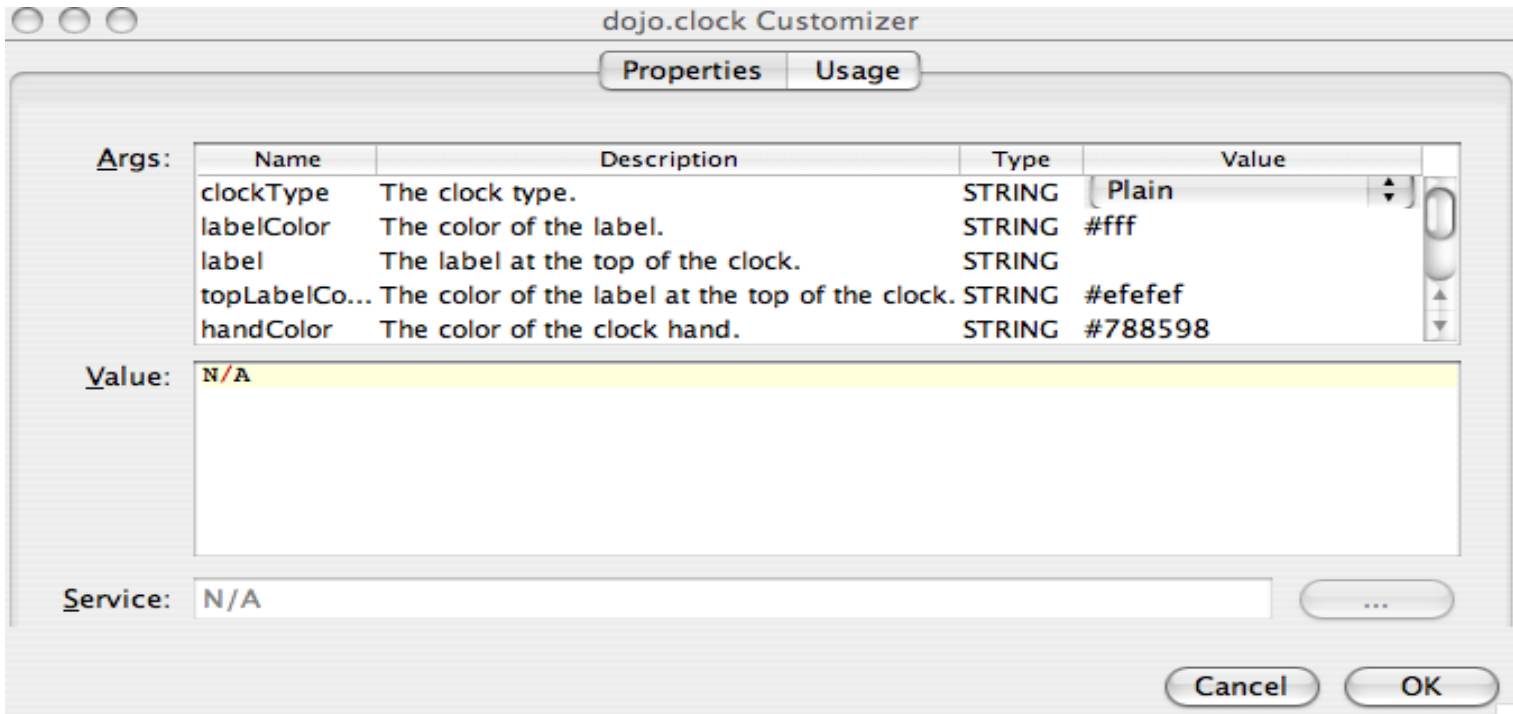
```
<%@ taglib prefix="a"  
    uri="http://java.sun.com/jmaki" %>  
<a:ajax name="hello" args="{name: 'world'}" />
```

# jMaki and Netbeans

- Same source, build, dev team, releases
- Same web site, forums download area
- Total synergy both ways
  - > New jMaki features
  - > New tools features

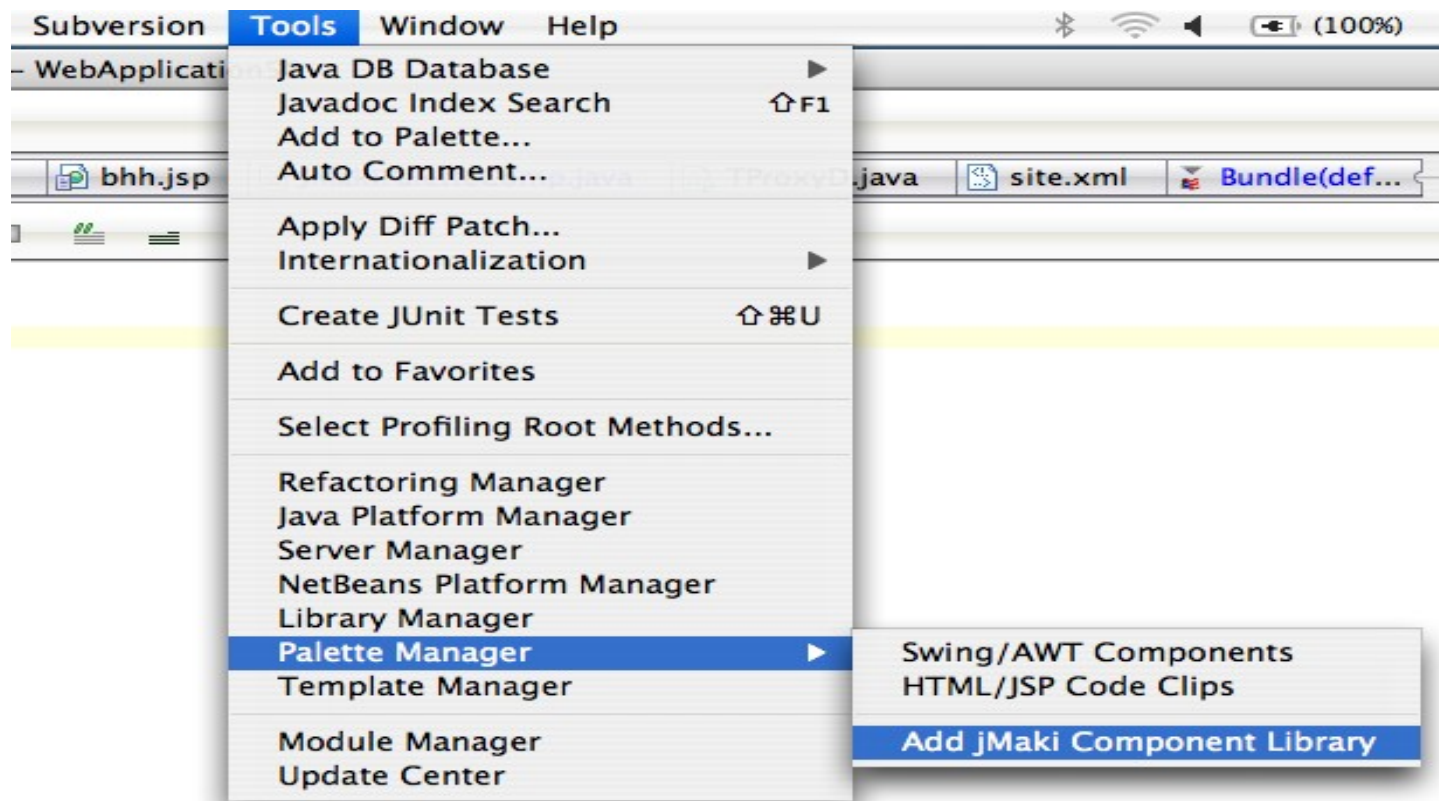
# jMaki Customizer

```
<a:widget name="dojo.clock" />
```



```
<a:widget name="dojo.clock"
args="{clockType:'plain', labelColor:'#fff',
topLabelColor:'#efefef'}/>
```

# Extensible Palette – Create your own

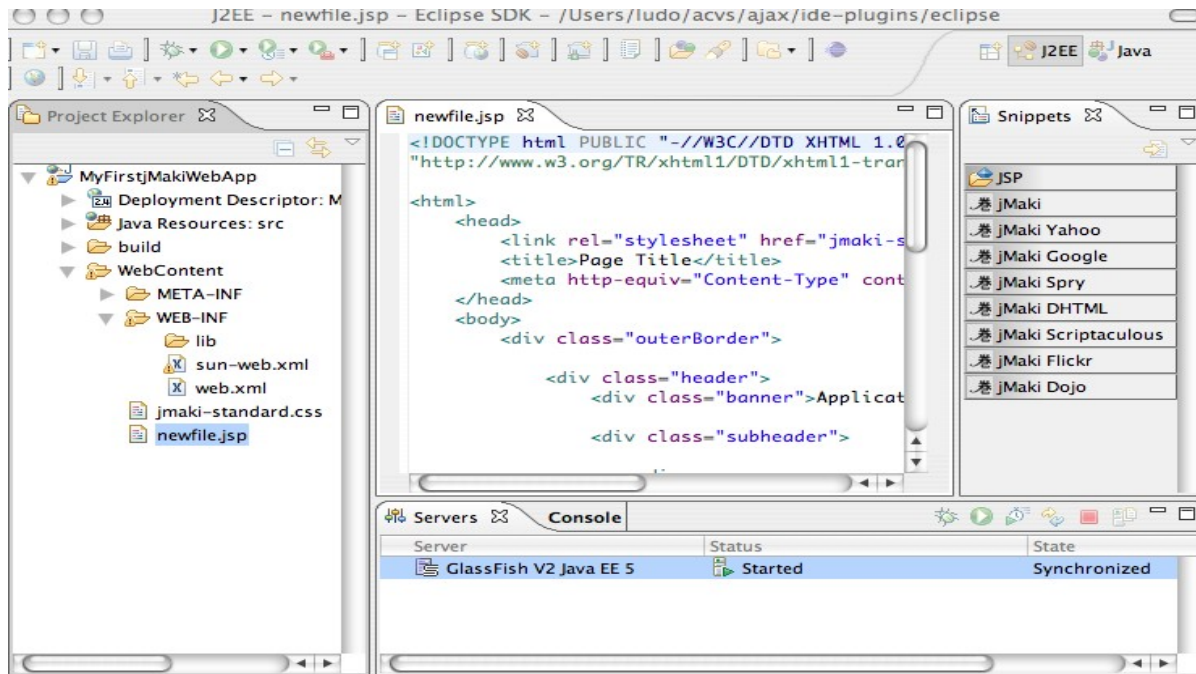


# Demo

## Extensible Widgets (Charting)

# Other Tools

- Ant, Vi, NB 5.5, 5.5.1, 6.0
- Eclipse (<https://ajax.dev.java.net/eclipse>)



# Extensible

- jMaki Charting
  - > <https://jmaki-charting.dev.java.net>
- jMaki Widgets
  - > <https://widgets.dev.java.net>
- jMaki Extensions
  - > <https://jmaki-ext.dev.java.net>
- Your own library

# Demo

## Real-life RIA using jMaki

# Roadmap

- 1.0 Release September 25<sup>th</sup>
- 1.1 Release Discussions have started and branch created
  - > Binding client event bus to server
  - > Server side JavaScript compression
  - > More widgets / libraries

# Summary

- Framework for building Ajax enabled Web 2.0 applications
- Java, PHP, JavaScript, and Ruby Support
- Consistent Data Models
- Event handling with Actions and Glue
- Tool support with NetBeans, Eclipse, and Ant
- Extensible via Extensions and your own Widget libraries

# For More Information

See Project jMaki website

> <http://jmaki.com>



<http://jmaki.com>