Project Phobos: Server-Side Scripting for the Java™ Platform

Roberto Chinnici

Ludovic Champenois

Senior Staff Engineers
Sun Microsystems, Inc.
http://phobos.dev.java.net/

TS-6957

Goal of This Talk

Learn how to build modern web applications the quick and easy way using Phobos

Agenda

What Is Phobos?

Programming Model (With Demo)

Ajax Using jMaki (With Demo)

Extensibility, Persistence (With Demo)

Conclusions

Agenda

What Is Phobos?

Programming Model (With Demo)

Ajax Using jMaki (With Demo)

Extensibility, Persistence (With Demo)

Conclusions

What Is Phobos?

- Lightweight application framework
- Running on the Java platform
- Supporting multiple scripting languages
- Current focus is on JavaScript™ technology
- Deploy to any Servlet container

What Problem Does It Address?

- Scripting languages growing in popularity
- Ajax places new emphasis on interactive development—avoid the compile/deploy cycle
- A scripting engine by itself is not enough
- Tooling is an important aspect

Key Functionality

- URL mapping
- Java Specification Request (JSR)-223 scripting engine integration
- Context management (scopes)
- Container independence
- Server-side JavaScript technology support

Goal: Productivity and Performance

- Be more productive by developing selected parts of your web application in a scripting language
- Remove the impedance mismatch from Ajax
- Glue together Java libraries and components
- Deploy to a proven platform

JavaScript Technology?

"JavaScript (technology) on servers will emerge as one of several programming models popularized by Web platforms by 2009 (0.7 probability)"

Gartner Report, November 21, 2006

Quick Guide

- Language: JavaScript programming language, others
- URL dispatching: ordered regexps
- Templates: EJS, FreeMarker, anything
- ORM: Java Persistence API
- Ajax: jMaki integration, other toolkits
- Extras: all Java libraries

Installing Phobos

- Set of NetBeans™ software plug-ins
- Bootstrapped using the Ajax update center
- Or use the Sun™ Web Developer Pack

Sun Web Developer Pack
Ride the next generation technologies for web application development.
Get It Now »

Agenda

What Is Phobos?

Programming Model (With Demo)

Ajax Using jMaki (With Demo)

Extensibility, Persistence (With Demo)

Conclusions

Development Process

1. Start your IDE
2. Create skeleton application using wizard
3. Run it in debug mode
4. Map out the URLs for pages, services, Ajax
5. Attach logic to them
6. Test out interactively
7. Go back to step 4, repeat
8. Stop the application, generate a war file
9. Done!

Application Layout

```
/application
    /controller
        main.js
    /dynamic
        sample.ejsp
    /module
        application.js
        resource.js
    /script
        index.js
    /template
    /view
        main.ejs
```

```
/static
    /resources
        ...jMaki...
    /css
        main.css
    faq.html
    release_notes.html
```

```
/environment
    development.js
    startup-webapp.js
```

URL Design

- External “appearance” of your application
- Keep URLs clean
- Recognize certain patterns

Plain script:	/doSomething.js
Qualified operation:	/store/display_cart
Resource:	/catalog/isbn/1234-5678-90

- All can take query arguments ?view=html
- Natural mapping to implementation logic

Plain Scripts

/application/script

- Servlet-like, but written in any language

```
response.status = 200;
response.contentType = "text/html";
var writer = response.writer;
writer.println("<html><head><title>Hello from
                              Javascript</title></head><body>”);
for (var i = 0; i < 10; ++i) {
    writer.println(“Hello from Javascript!<br>”);
}
writer.println(“</body></html>");
writer.flush();
```

Controllers

/application/controller - /application/view - model

- MVC pattern, in JavaScript programming language

```
library.common.define(controller, “main”, function() {

  // constructor
  this.Main = function() {}

  // action method
  this.Main.prototype.show = function() {
    library.view.render(“main.ejs”);
  }
}
```

- /main/show parsed as /@controller/@action

Views—Embedded JavaScript Technology Files

/application/view - .ejs extension

- Always rendered by controllers
- Simple templating system, PHP-like
- Embedded JavaScript technology statements

  <% ... statements ... %>
- Embedded JavaScript technology expressions

  <%= ... expression ... %>

System Apps

- http://myserver:8888/system
- In-browser development helpers
  - Code generation, URL mapping, CRUD, ...
- Part of the running application
- “Eat your own dog food”
- IDE in a browser?

DEMO

Sample application using NetBeans IDE

JavaOne

javaone

JavaOne

JavaOne

JavaOne

'javaone

JavaOne

'javaone

JavaOne

'javaone

Agenda

What Is Phobos?

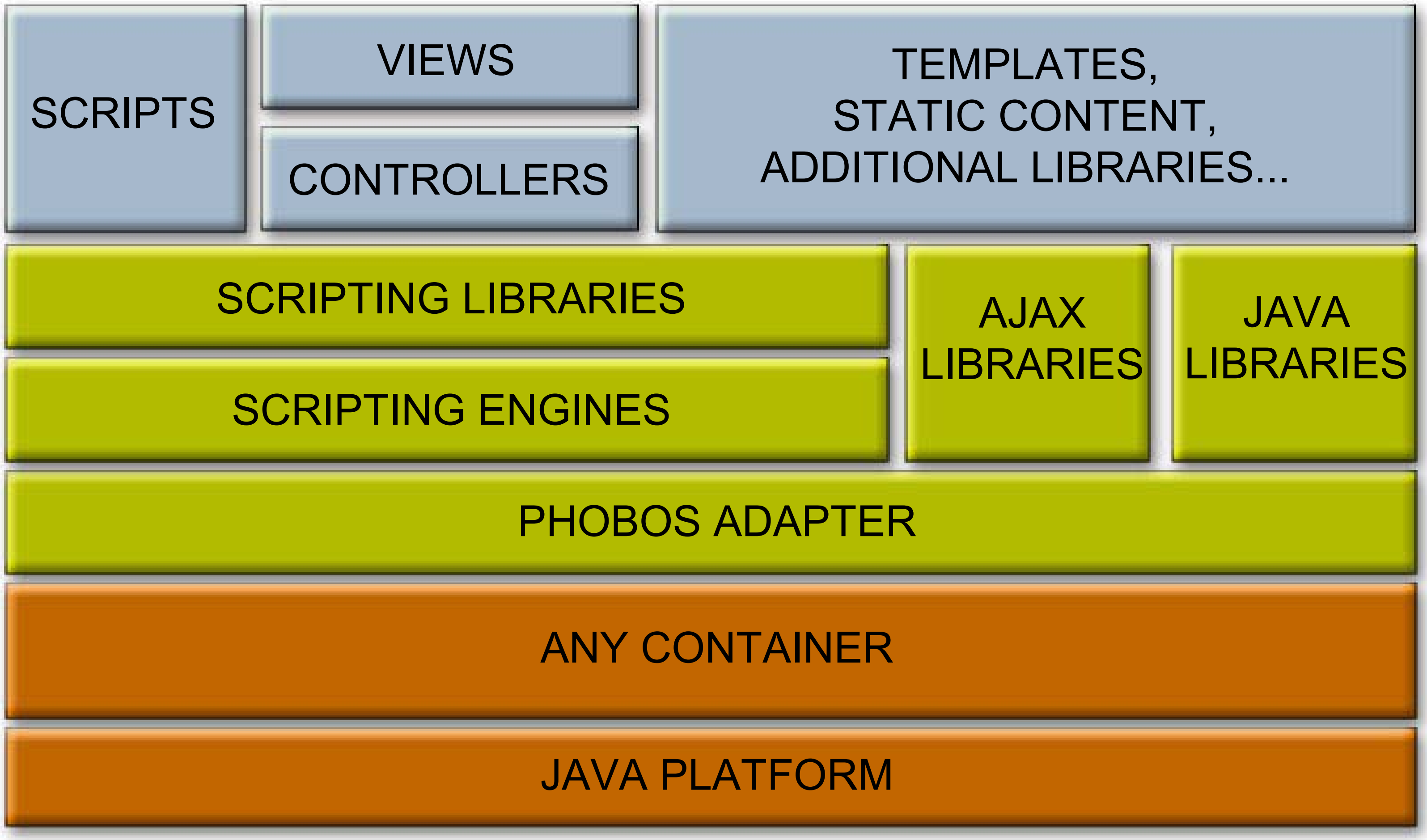
Programming Model (With Demo)

Ajax Using jMaki (With Demo)

Extensibility, Persistence (With Demo)

Conclusions

Architecture



Multiple Extension Points

- Adapter to swap in a new container
- JSR 223 for scripting engines
- Java libraries
- Ajax/client libraries
- Extensions at the JavaScript technology level
- Fully customizable URL mappings

Default URL Mappings

- Several predefined patterns
- No configuration needed

Index page	`/`
Static content	`/[`*`path/`*`]`*`static_content`*
Script	`/[`*`path/`*`]`*`scriptname`*
Controller	`/`*`controller`*`[/`*`action`*`][/`*`id`*`]`
PHP-like content	`/[`*`path/`*`]`*`dynamic_content`*

Resources

/application/module

- REST framework
- Resources are classes
- Methods are HTTP methods: GET, PUT, ...
- Code deals with HTTP entities
  - Content type, payload, extension headers
- Many HTTP aspects offloaded to framework

Declaring New URL Mappings

/application/module/application.js - onStartup

- Add a new rule at startup

```
application.mapping.rules.push({
    url: "/collection/@id",
    factory: "module.atom.createCollectionResource",
    fn: "library.mapping.maybeREST"
});


application.mapping.rules.push({
    url: "/",
    script: "index.rb"
});
```

Phobos on GlassFish™ Build v.3

- New, modular application server runtime
- Phobos as a lightweight container
- No dependency on the Servlet container
- Fast startup, small memory footprint

JSR 223 Scripting Engines

- Automated engine discovery
- Just drop a new engine in the classpath
- Engine selected based on the file extension
  - .js .rb .py .groovy .xslt .scm ...
- Many engines available on java.net
  http://scripting.dev.java.net/
- Practically all of them have the ability to call from scripting into Java code

JavaScript Technology in Phobos

- Mozilla Rhino 1.6R4
- Robust, fast implementation
- Optional compilation to bytecode
- Built-in debugging support
- Powerful interface to Java code
- Many language extensions

Accessing Java Libraries

- Integrated JavaScript-Java programming language bridge
- Bean properties become JavaScript technology properties
- Often can copy and paste Java source code

```
var builder = new Packages.org.jdom.input.SAXBuilder();
var doc = builder.build(
                   new java.io.FileInputStream("a.xml"));

// response is a javax.servlet.http.HttpServletResponse
response.setStatus(200);
response.status = 200; // equivalent
```

JavaScript Technology Extensions in Phobos

- Continuations
- Dynamic objects
- Allow many advanced constructs:
  - Multiple inheritance
  - Autoloaded modules
  - Builders
  - DoesNotUnderstand: / missing_method
- E4X

E4X

- XML support at the language level
- XPath like search syntax

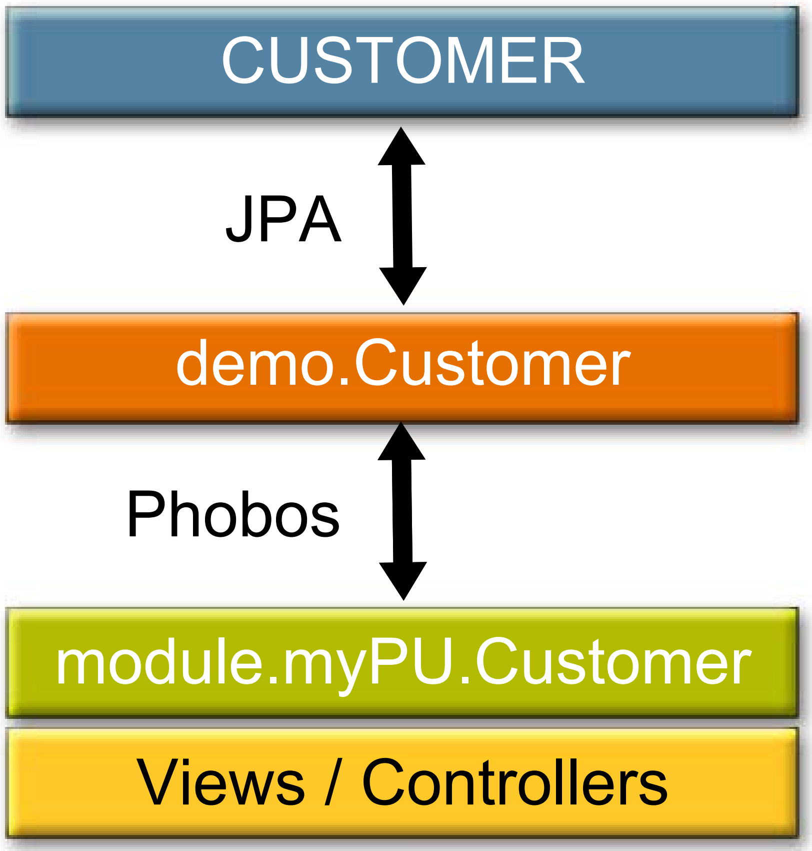
```
// HTML example
var doc = <html/>;
doc.head.title = “Hello, world!”;
doc.body.@bgcolor = “#224466”;
doc.body.p = “This is all the text on this page.”;

// Atom example
default xml namespace = new Namespace("atom",
"http://www.w3.org/2005/Atom");
var feed =
<feed><title>{title}</title><author><name>{author}</name></
author></feed>;
```

Persistence in Phobos

- Thin wrapper around Java Persistence API
- JavaScript technology model, view, controller generated based on JPA entity classes

1. Create a JPA library project
2. Add it to the Phobos classpath
3. Run the generator
4. Customize ad lib

Persistence Mapping



Database table

Entity class in a persistence unit

JavaScript technology module

DEMO

Persistence

Agenda

What Is Phobos?

Programming Model (With Demo)

Ajax Using jMaki (With Demo)

Extensibility, Persistence (With Demo)

Conclusions

Phobos Summary

- Fast, interactive development model
- Targeted at rich web applications (Ajax)
- Complementary to existing Java technologies
  - Persistence, web services, JavaServer Faces™ platform, Enterprise JavaBeans™ (EJB™), ...
- Full IDE support in NetBeans IDE

For More Information

- Phobos
  http://phobos.dev.java.net/
- Project jMaki
  http://ajax.dev.java.net/
  - Sessions TS-6375, TS-9516
- Project GlassFish
  http://glassfish.dev.java.net/
- Sun Web Developer Pack
  http://developers.sun.com/web/swdp/

Q&A

Roberto Chinnici
Ludovic Champenois

Project Phobos: Server-Side Scripting for the Java™ Platform

Roberto Chinnici

Ludovic Champenois

Senior Staff Engineers
Sun Microsystems, Inc.
http://phobos.dev.java.net/

TS-6957

Shortcut—Self-Rendering Views

/application/dynamic - .ejsp extension

- Views that don’t need a controller
- Useful to add dynamic behavior to existing, static HTML pages
- Complete analogy with PHP
- Unlike PHP, you don't have to use them all the time!