



Scripting and the Java™ Platform

Productivity and Performance

Roberto Chinnici

**Senior Staff Engineer
Sun Microsystems, Inc.**

April 18, 2007



Lots of activity around scripting
on the Java™ platform

JRuby on Rails

Groovy + Spring + Hibernate
=
Grails

***ails**
(Insert your own prefix)

Project Phobos

A minimal framework
to bridge the two worlds

What is Phobos?

- Lightweight application framework
- Running on the Java™ platform
- Support for multiple scripting languages
- Current focus is on JavaScript
- Deploy to any servlet container

Plugging the Gaps

- URL mapping
- Scripting engine integration
- Context management
- Container independence

Keywords

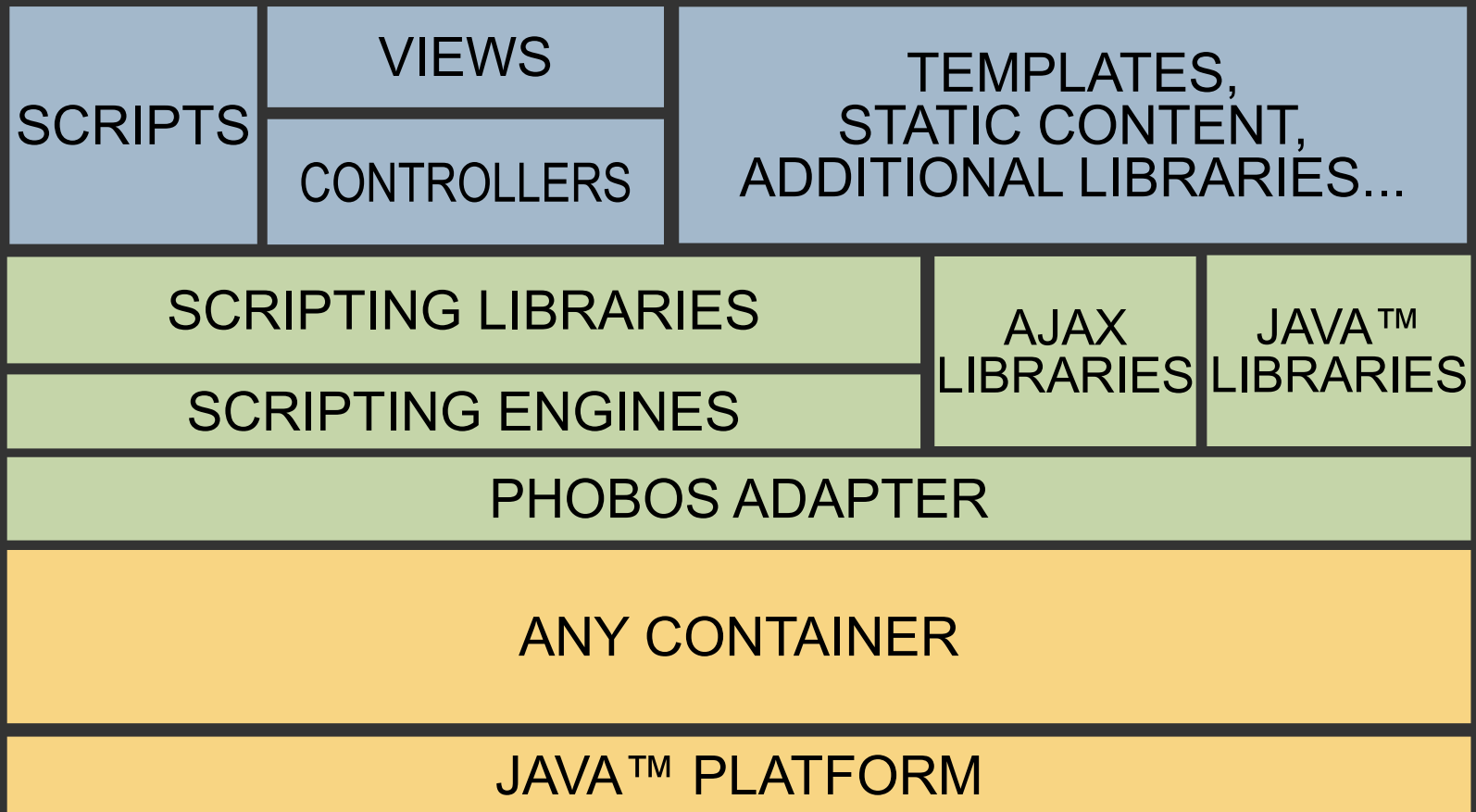
- Complementary
- Focused
- Multilingual
- Toolable
- Un-opinionated?

Capture the glue

Goal

***DELIVER A GREAT
DEVELOPER EXPERIENCE***

Architecture



The Java™ Platform as Runtime

- Multithreaded
- State of the art garbage collector
- No Global Interpreter Lock
- Zillions of libraries
- Proven deployment platform
- Bytecode as compilation target
- More dynamic language-friendly every day
invokedynamic bytecode in JDK 7

Programming model

Development Process

1. Start NetBeans
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, you're done

URL Design

- Plain scripts
/doSomething.js
- Controller/action pattern
/store/display_cart
- Resource (REST)
/catalog/isbn/1234-5678-90
- User-defined
...anything else...

How Does Phobos Stack Up?

- URL dispatching: ordered regexps
- Templates: EJS, FreeMarker, anything
- ORM: Java™ Persistence API + wrapper
- Form handling: nothing
- Extras: all Java™ libraries

Application Layout

```
/application
```

```
  /controller
```

```
    test.js
```

```
  /dynamic
```

```
    sample.ejsp
```

```
  /module
```

```
    application.js
```

```
  /script
```

```
    index.js
```

```
  /template
```

```
  /view
```

```
    test.ejs
```

```
/static
```

```
  /resources
```

```
    ...jMaki...
```

```
  /css
```

```
    main.css
```

```
  /faq.html
```

```
  /release_notes.html
```

```
/environment
```

```
  /development.js
```

```
  /startup-webapp.js
```

Multiple “Platforms”

- Servlet container, single jar, embedded
- Concrete *PhobosAdapter* subclass
- Applications are platform-agnostic
- Layout is virtual
- Allows optimizations, repackaging, versioning, masking

Embedded JavaScript Files

- PHP/ASP-like syntax

```
<html>
  <head>
    <title><%= model.title %></title>
  </head>
  <body>
    <%= model.greeting %> from Javascript!
  </body>
</html>
```

IDE Integration

- Phobos ships as NetBeans plugins
- Eclipse support is on our to-do list
- Full development environment:
 - wizards, component palette, language-aware editor, debugger, variable inspector, breakpoints, watches, multiple threads, ...
- Application runs inside the IDE at development time

Grizzly

- NIO-based HTTP connector
- Small, fast, scalable, embeddable
- Asynchronous request processing (ARP)
- COMET support

System Apps

<http://myserver:8888/system/>

- In-browser development helpers
 - Code generation, URL mapping, ...
- Part of the running application
- “Eat your own dogfood”
- IDE in a browser?

Cheat Sheet

- Controllers are JavaScript
- Models are JavaScript or Java™ objects
- Views contain embedded JavaScript
- Libraries are JavaScript or Java™ code
- Client logic in JavaScript
- Server is client framework-agnostic
- Data serialized in JSON format

A quiz

C and Bourne Shell

COM and Visual Basic

C and Perl/Python/Ruby

C++ and Office macros

XCMD and HyperCard

“Soft vs Hard” Principle

- Two layers: soft and hard
- Application and system
- Lots of successful systems follow this rule
- Java™ platform close to being the exception

No other platform broke it and made it big, ever

Java™ language and...

... we can't pick the winner!

Phobos is Multilingual

- JSR-223 -based
- Engine discovery
 - META-INF/services/javax.script.ScriptEngineFactory*
- Capabilities described by interfaces
 - javax.script.Invocable*
 - javax.script.Compilable*
- File extension used to find the engine

“Soft” Engines

- Non-embedded DSL facility
- Translate to favorite target language
- Plug new mini-languages
- Inherit debugging support from lower layer

A Plea to Language Designers

- Make your language open
- Semantics not based on C interpreter only
- Emulation has limits
- Especially across reflective boundaries
- CPython / Jython / IronPython / Pypy

Phobos uses JavaScript

Why JavaScript?

- “The world most misunderstood programming language”, yet...
 - Ubiquitous
 - Known by millions of (non?)developers
 - “The assembly language of the Internet”
 - Functional, dynamic, prototype-based

Flawed?

The Ugly

- Unusual OO model
- Coercion
- Few built-in libraries
- No iterators
- Scheme-like, yet no macros

Mozilla Rhino + Extensions

- Optionally compiled to bytecode
- Continuations, dynamic objects, E4X
- Suddenly, lots of constructs are possible:
 - Autoloaded modules
 - Builders
 - Multiple inheritance
 - DoesNotUnderstand:
- Not your browser's JavaScript

The good

Meant for Script-ing

- Glueing things together
- Accessing existing libraries/objects
- Writing functions
- Trick is to stay in the soft spot

Native JSON

- Really, why use XML?
- No impedance mismatch
- Same data structures on client and server
- No translation layer needed

JavaScript On All Tiers

- No more hiding JavaScript in the browser
- No translations
- Code can become mobile too
- Controllers can migrate to the client

The future

JavaScript 2.0

- ECMAScript 4 initiative
- Type system
- Package system
- *private-public* qualifiers
- Interfaces
- Iterators, generators, comprehensions

Cool!

- More type information =
 - Better match for the Java™ Virtual Machine
 - Better translation to bytecode
 - Better performance

The wildcard

Scala

- Multi-paradigm language
- Unified type system
- Closures
- Actors

Ajax



jMaki

jMaki



- Lightweight JavaScript framework
- Spans multiple widget libraries
- Unified interface to widgets
- Event-based component “glueing”
- Server-agnostic: JSP, JSF, Phobos, PHP

jMaki Value-Add



- Standardized data model
- Common resource layout
- CSS-based layouts
- Declarative or programmatic “glue”
- Widget libraries – with descriptions!
- NetBeans and Eclipse plugins



No library-specific
knowledge needed

jMaki Futures



- Separate core/extensions
- Support Ruby on Rails, WebObjects
- Themes
- 1.0 release

Where Are We?

- Dynamic, extensible server-side solution
- Dynamic, extensible Ajax solution
- Integrated development environment

Now we need a deployment platform that matches all that

GlassFish v3

GlassFish v3

- Next iteration of the GlassFish appserver
- Modular infrastructure
- Extensible
- Multiple containers
 - Web applications (servlet 2.5)
 - Phobos
 - JRuby on Rails
 - PHP using Quercus

Startup time: < 1 second

Memory usage:
Pay for what you need

In case you missed it...

Everything is Open Source

Everything is Open Source

- OpenJDK
- Phobos
- jMaki
- NetBeans
- GlassFish
- Grizzly
- Scala
- ...

Conclusion

There is no fork in the road

Scripting and the Java™ Platform

- Openness
- Choice
- Productivity
- Performance

Links

- Phobos - <http://phobos.dev.java.net/>
- jMaki - <http://ajax.dev.java.net/>
- GlassFish - <http://glassfish.dev.java.net/>
- Sun Web Developer Pack - <http://developers.sun.com/web/swdp/>

Sun Web Developer Pack

Ride the next generation technologies for web application development.

Get It Now »





Scripting and the Java™ Platform

Productivity and Performance

Roberto Chinnici

roberto.chinnici@sun.com