Java™ SE: Now and Future

Sang Shin, sang.shin@sun.com
Java Technology Architect
www.javapassion.com
Sun Microsystems, Inc.

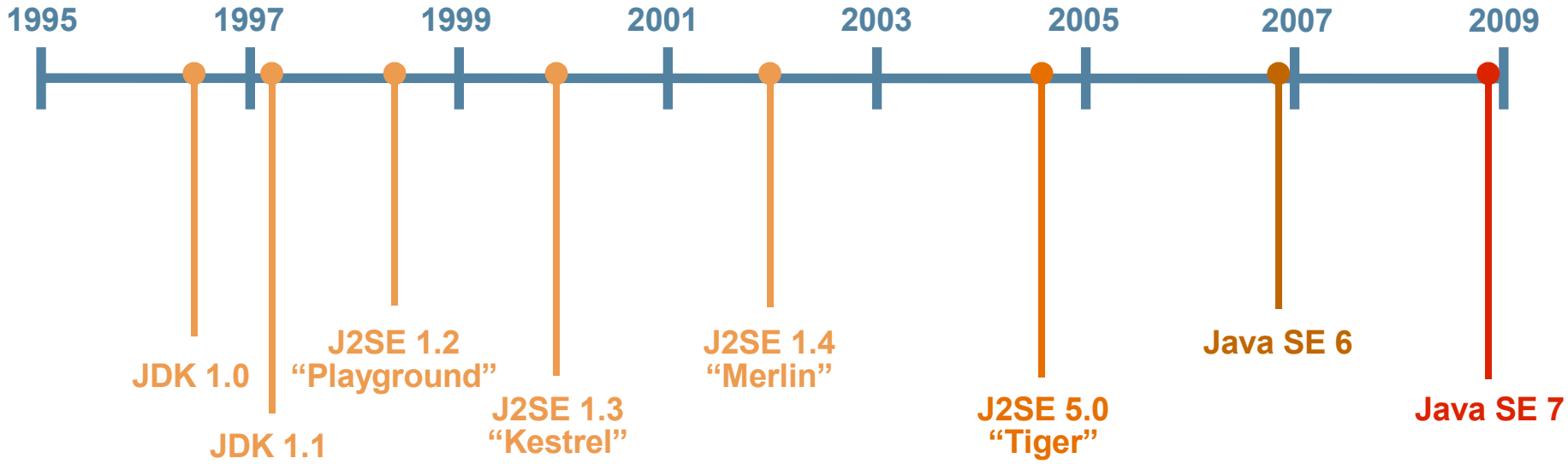
Agenda

- Java SE 6
- The Making of Java SE Platform
- Java Platform Speaks in Many Tongues
- Breaking Up Is Hard to Do
- Java SE Platform on the Desktop
- Some Important Upgrades

Java SE 6

Changing Face of Java

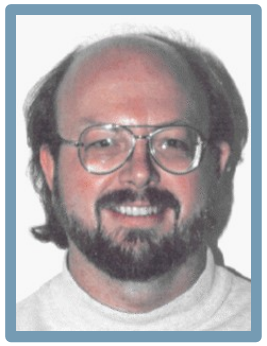
Java technology releases

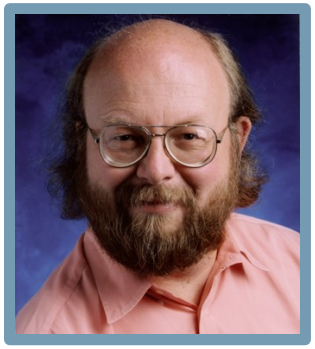


JDK = Java Development Kit (JDK™) | JRE = Java Runtime Environment (JRE™) |
J2SE = Java 2 Platform, Standard Edition (J2SE™ platform)

Changing Face of Java

Java technology releases





1995 1997 1999 2001 2003 2005 2007 2009

JDK 1.0

JDK 1.1

J2SE 1.2 "Playground"

J2SE 1.3 "Kestrel"

J2SE 1.4 "Merlin"

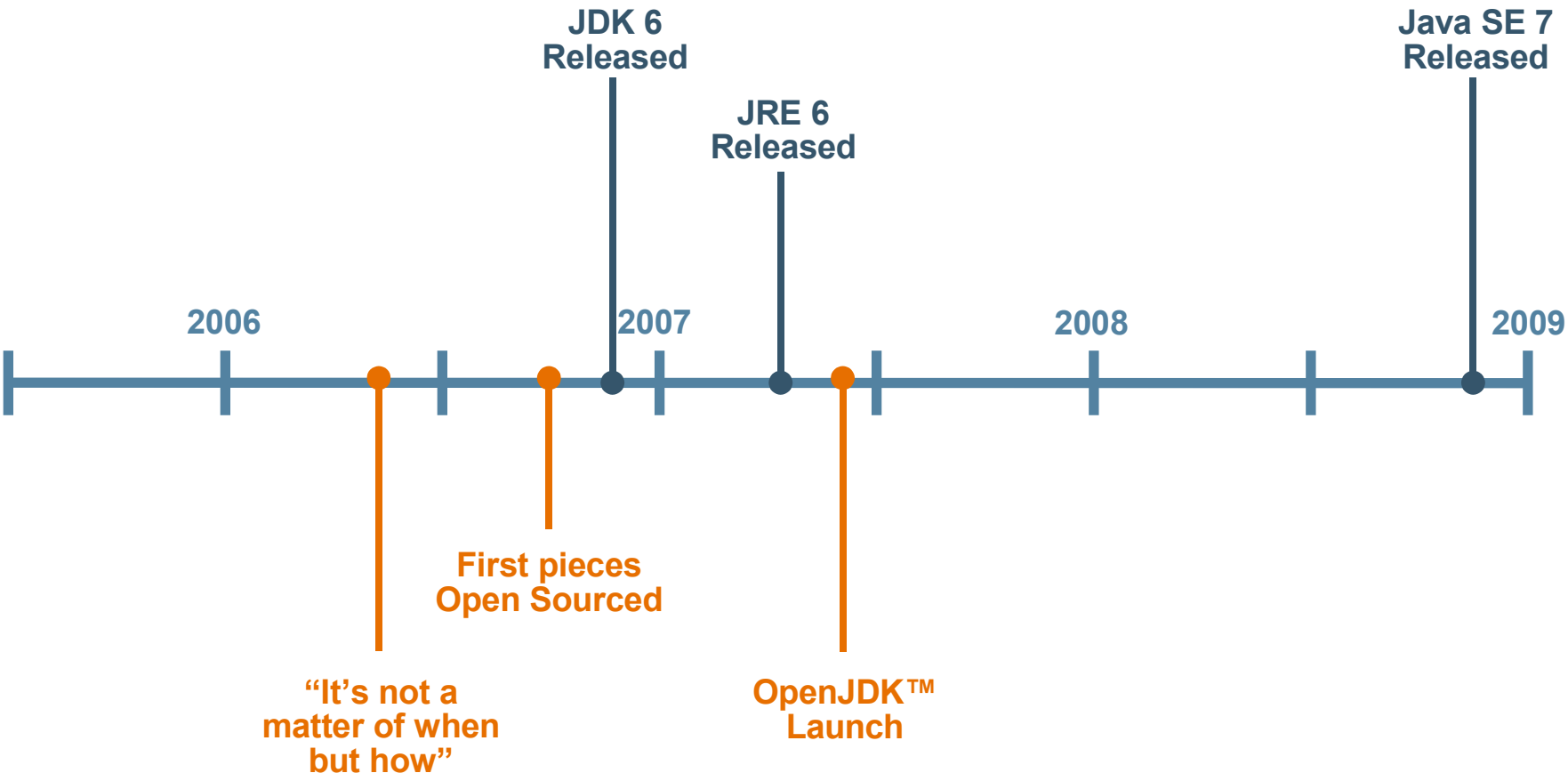
J2SE 5.0 "Tiger"

Java SE 6

Java SE 7

JDK = Java Development Kit (JDK™) | JRE = Java Runtime Environment (JRE™) |
J2SE = Java 2 Platform, Standard Edition (J2SE™ platform)

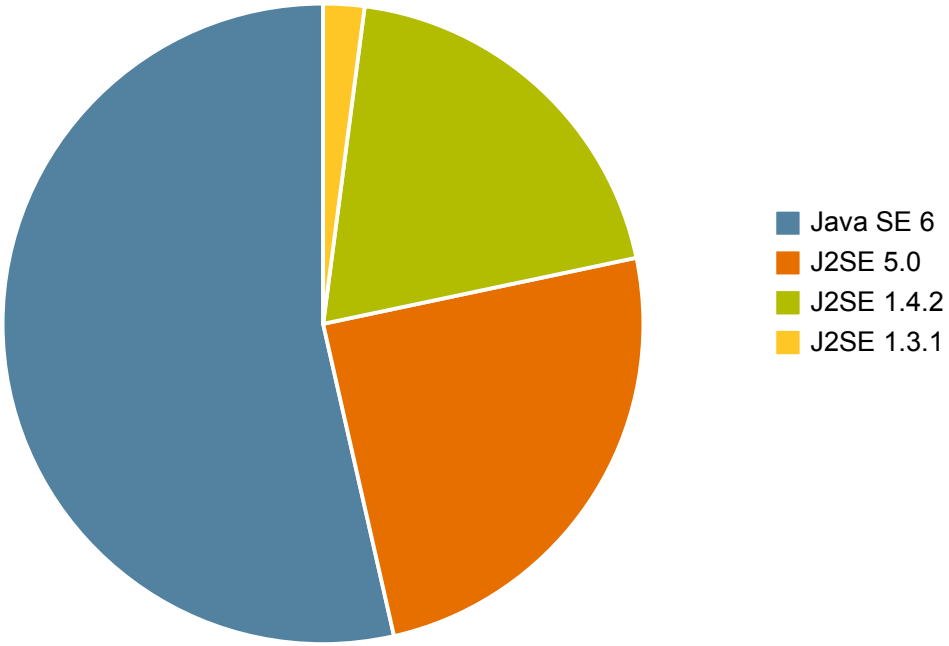
Year in Review



JDK = Java Development Kit (JDK™) | JRE = Java Runtime Environment (JRE™)

JDK Version 6 Adoption

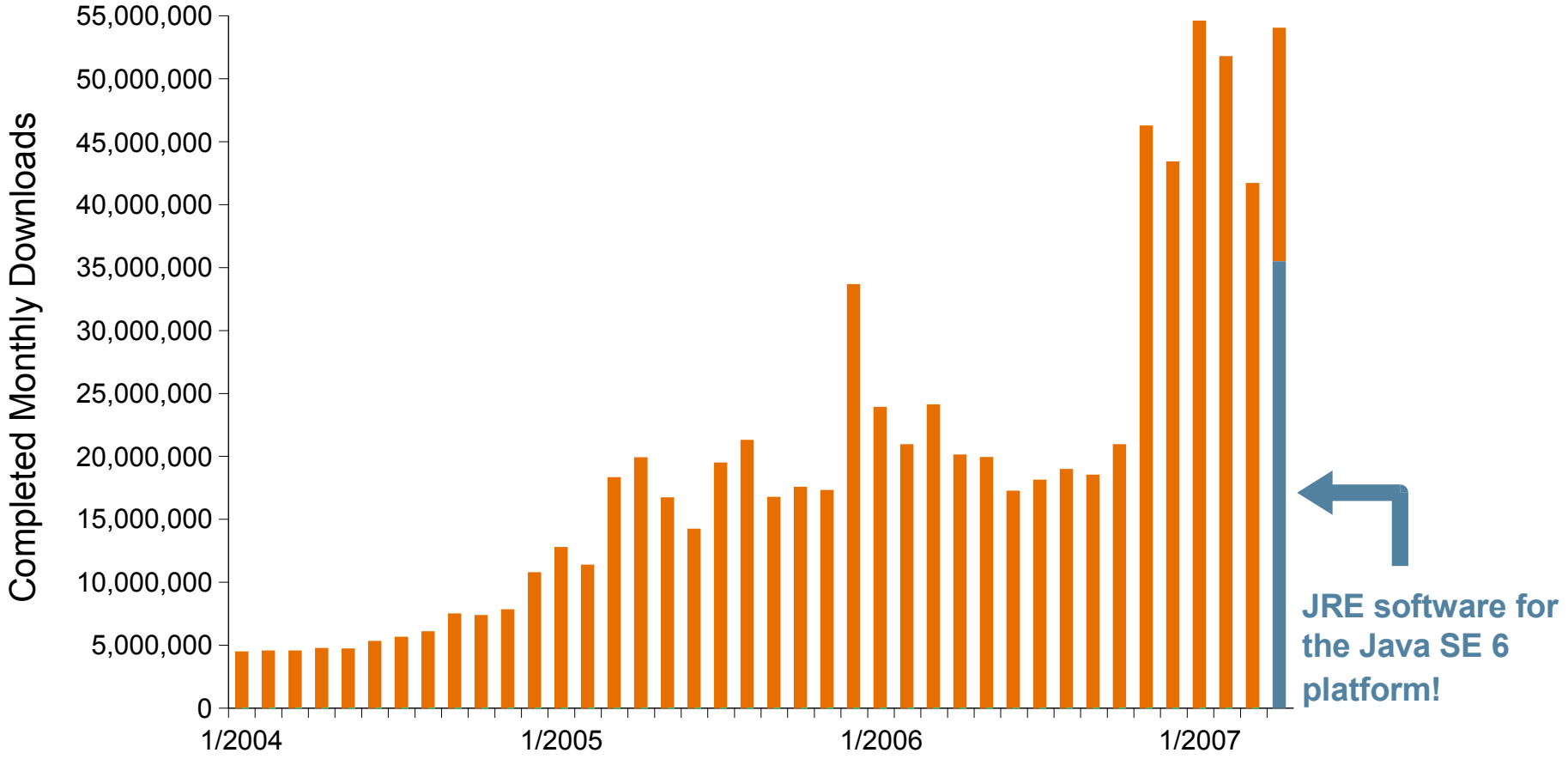
Downloads of the JDK release since December



2,090,155 downloads of the JDK Version 6. Already!

Source: java.sun.com

JRE Software Adoption



Java SE Platform on the Desktop

- 60% of new PCs will have Java SE platform pre-installed
- 91% of all PCs run Java platform*
- That's 540 million PCs worldwide
- 35.5 million of them upgraded to the Java SE 6 platform in the last 20 days in April

Have you upgraded to the Java SE 6 platform yet?

java.sun.com data except *Omniture

Java SE 6 Platform: Top 10 Features

Present

Web Services	Easy to use Web Service APIs
Scripting	Ability to mix JavaScript™technology with Java code
Database	Updated JDBC APIs, all-Java database in JDK
More Desktop APIs	Swingworker, JTable sorting and filtering, GroupLayout and more
Monitoring and Management	Attach on demand
Compiler Access	APIs to control the compiler
Pluggable Annotations	Define your own annotations
Desktop Deployment	Better Swing fidelity, tuned for Vista
Security	Integration with native services
The Ilities: Quality, Compatibility, Stability	Faster more reliable and backwards compatible

http://blogs.sun.com/dannycoward/entry/java_se_6_top_ten

JDBC = Java Database Connectivity (JDBC™)

Java SE Platform Troubleshooting

Java SE 6 platform

Tool name	Purpose
jconsole	GUI for diagnostics
jps	Determine Java technology processes
jmap	Snapshot memory usage
jhat	Analyze memory usage
jstack	Thread status

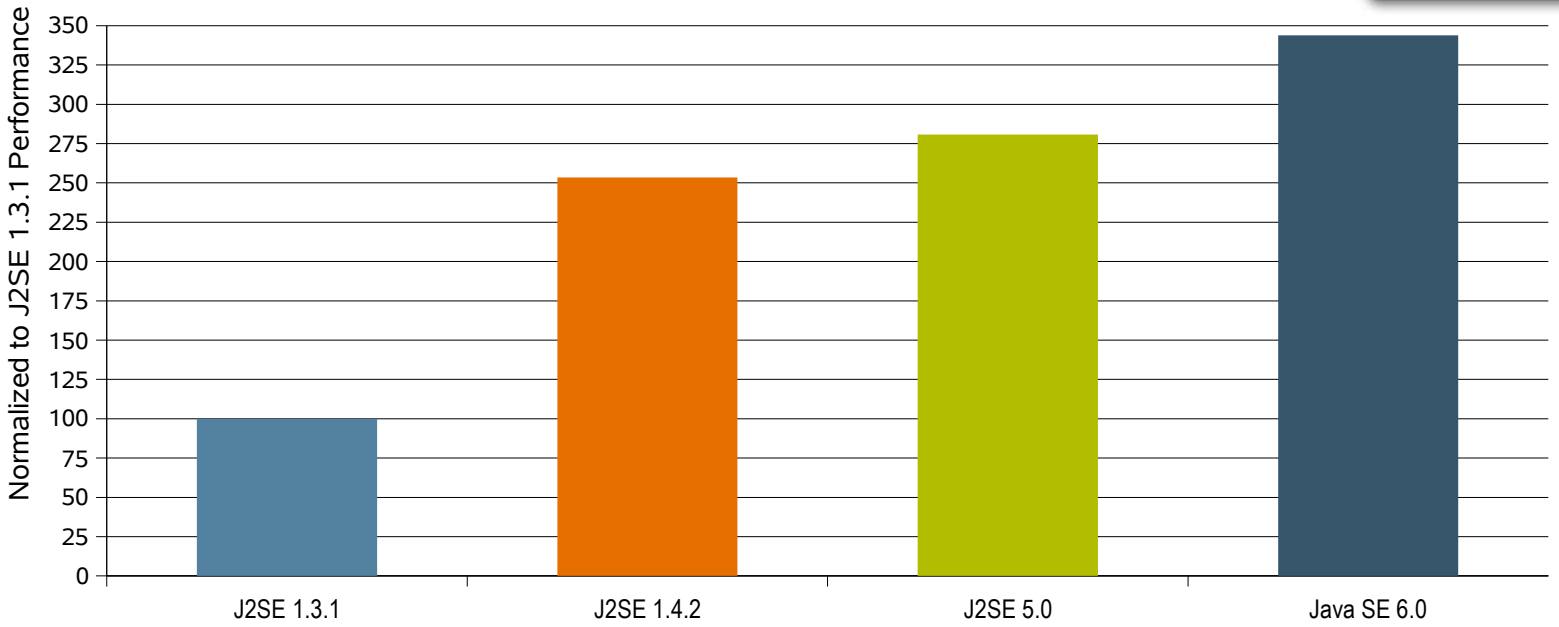
No need to restart the Java Virtual Machine any more!

http://blogs.sun.com/dannycoward/entry/java_se_6_top_ten

Better Performing on the Server

Server benchmark: SPECjbb2000

TS-2885
High-performance
Thur 4.10pm

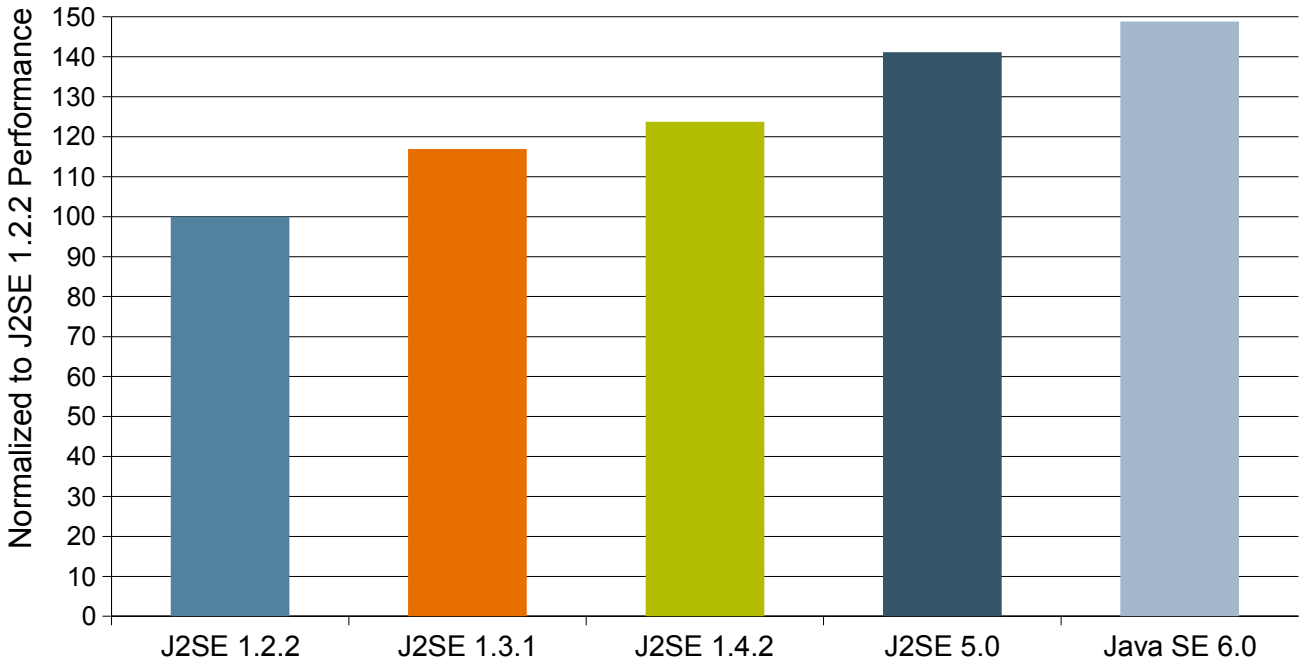


AND Java SE 6 platform holds Single Instance SPECjbb2005 World Record, on Sun Fire™ E25K server with a score of 1,149,100 SPECjbb2005 bops/JVM

JVM = Java Virtual Machine (JVM™)
The terms “Java Virtual Machine” and “JVM” mean a Virtual Machine for the Java™ platform.

Better Performing on the Desktop

Client benchmark: SwingMark



BOF-2366
Performance Q's
Wed 7.55pm

The Making of Java SE Platform

Java SE Platform Is Open Sourced

Today!

- Fully buildable open source JDK software
- Extension of the Peabody experience
- GPLv2 license, encouraging
  - Compatible innovations
  - Wider distribution
- OpenJDK community
  - openjdk.java.net
- Interim governance board
  - Charter to build OpenJDK constitution

How Java SE Platform Gets Developed

Present and future

	Java Community Process	OpenJDK
What	API Specifications	JDK Implementation
Where	jcp.org	openjdk.java.net
How	Expert Groups	OpenJDK projects
When	Now	Now!

The Java Platform Speaks in Many Tongues

Why Go Multi-Lingual

Present and future

- Java programming language is the best general purpose language!
- Many other languages, many other virtues
  - Rapid prototyping and experimentation
  - Particular styles of programming
  - Mixing different types of developers
  - Or just for fun

Java SE Platform Supports Scripting

Java SE 6 platform

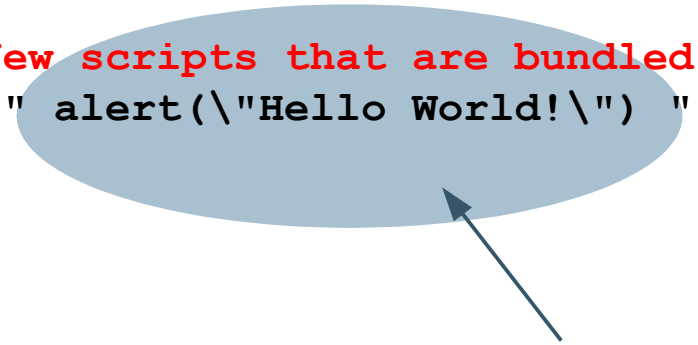
- Scripting for the Java Platform (JSR 223)
  - Developer APIs to mix script fragments in
  - Framework APIs for adding script engines
- Collecting conforming scripting engines
  - See scripting.java.net
- We added a JavaScript technology engine
  - JavaScript technology works out of the box

JSR = Java Specification Request

Scripting on Java SE 6 Platform

Java SE 6 platform

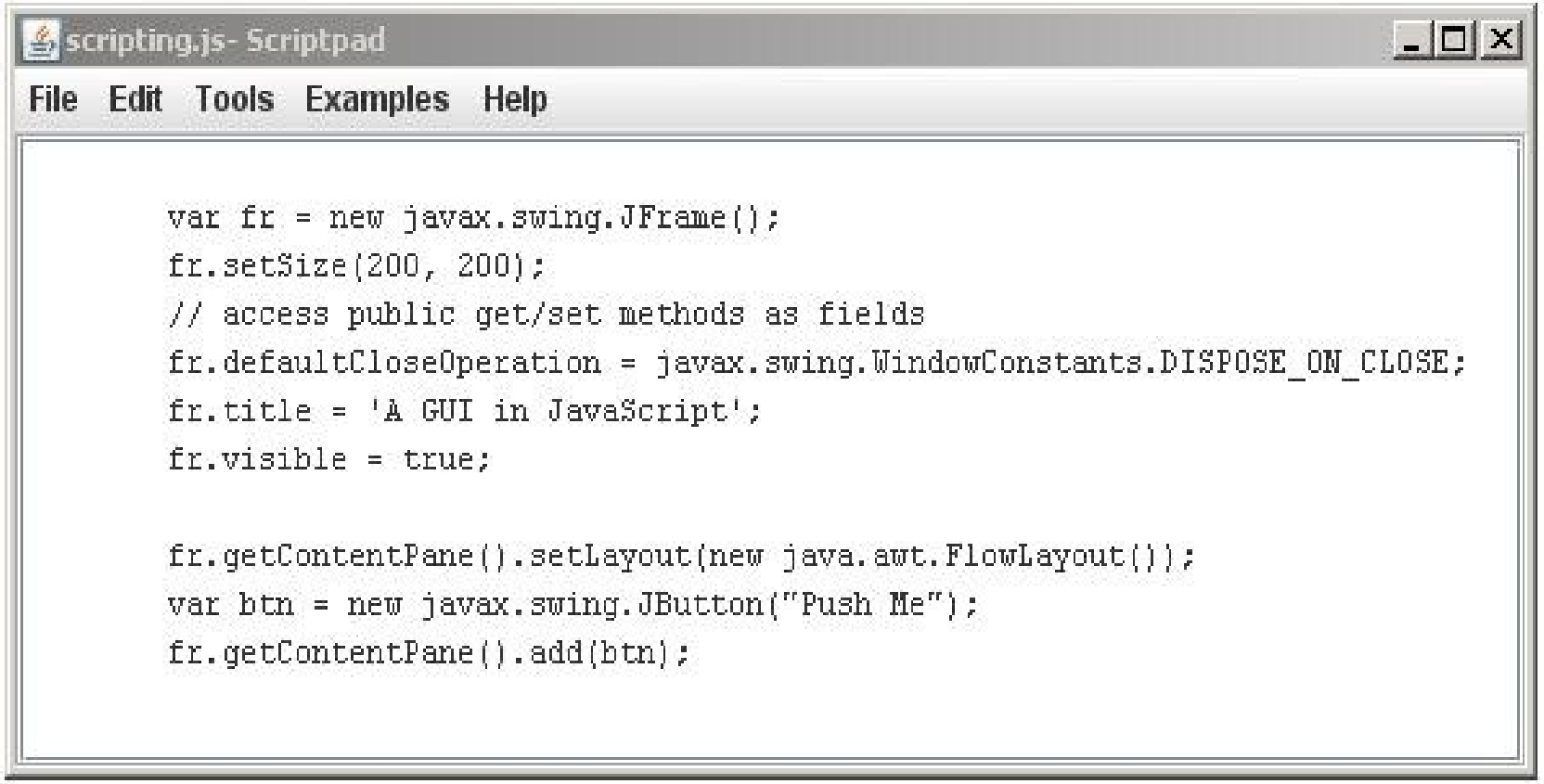
```
    // create a ScriptEngineManager
ScriptEngineManager m = new ScriptEngineManager();

// get an instance of JavaScript script engine
ScriptEngine engine = m.getEngineByName("js");

// evaluate few scripts that are bundled in "resources"
eval(engine, " alert(\"Hello World!\") ");
```



Insert any other JavaScript programming language here

Scripting Sample in JDK Version 6

Java SE 6 platform



Scripting on Java SE 6 Platform

Java SE 6 platform

Multiple Languages in JDK Version 7

Java SE 7 platform

- Turbo-charging scripting engines
  - New bytecode for dynamic method dispatch
  - 'Supporting Dynamically Typed Languages on the Java Platform' (JSR 292)
  - Investigate hotswapping
- Bundling more dynamic language engines
  - 'The Beanshell Scripting Language' (JSR 274)
  - JRuby, Jython, BeanShell, Groovy
  - JavaFX™ technology Script

Evolving the Java Programming Language

Java SE 7 platform

- Reading is more important than writing
- One language same meaning; everywhere
- Simplicity matters

- Seeking a small number of changes for Java SE 7 platform

Evolving the Java Programming Language

Java SE 7 platform—potential changes

- 'superpackages'
- Extensions to the annotation syntax (JSR 308)
- Language support for Java technology properties
- Control abstraction constructs
  - Closures
  - Concise instance creation expressions
  - First-class methods
- Operator overloading
- 'Rough edges'
  - Shorter variable declaration, strings in switch statements, Enum comparisons

Breaking Up Is Hard to Do

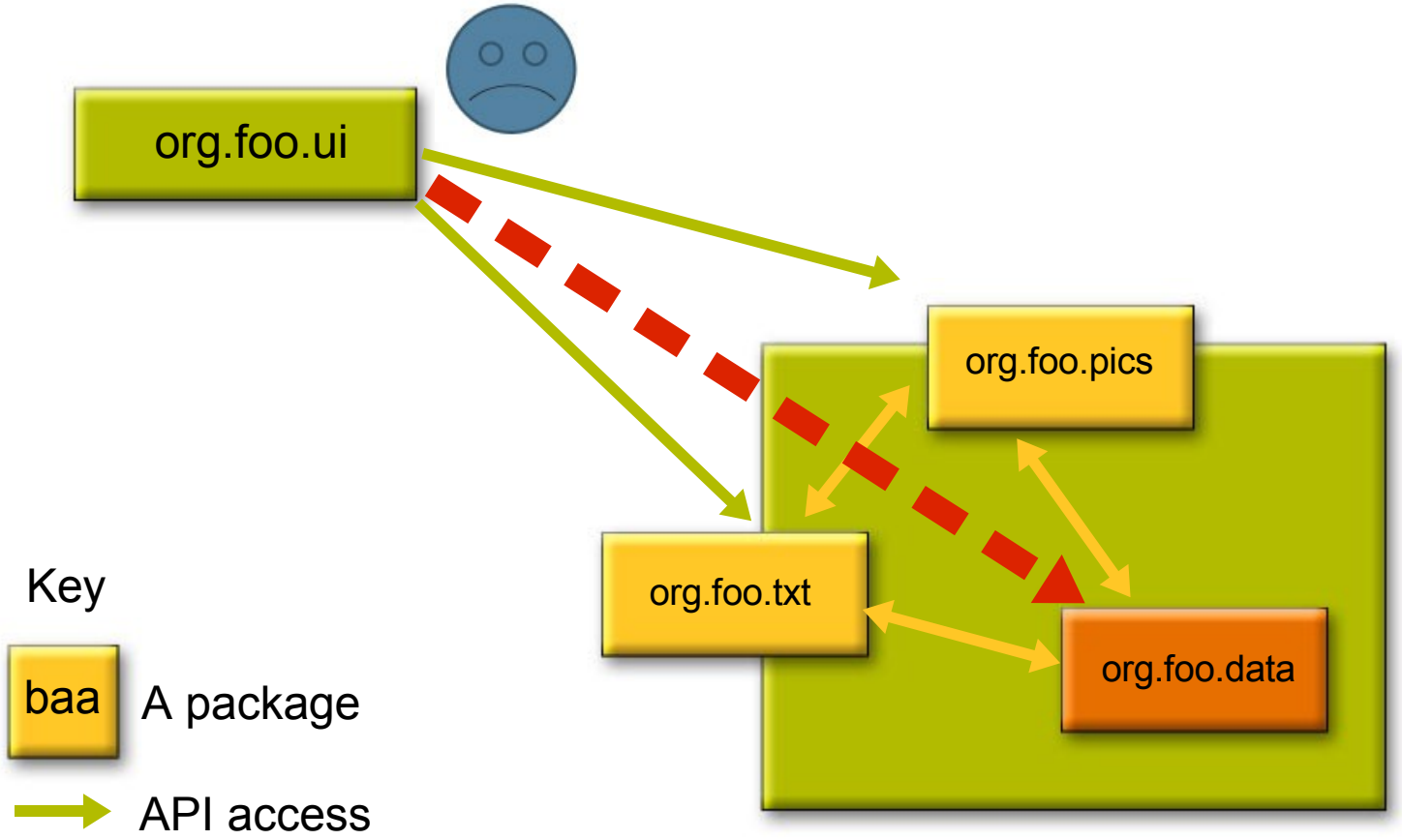
Modularity in Java SE Platform

Java SE 6 platform

- Development time
  - Interfaces and implementation classes
  - Information hiding
    - `public`, `private` and `protected`, class level and below
  - Assertions
- Deployment time
  - Java ARchives (JAR)
  - Resources framework

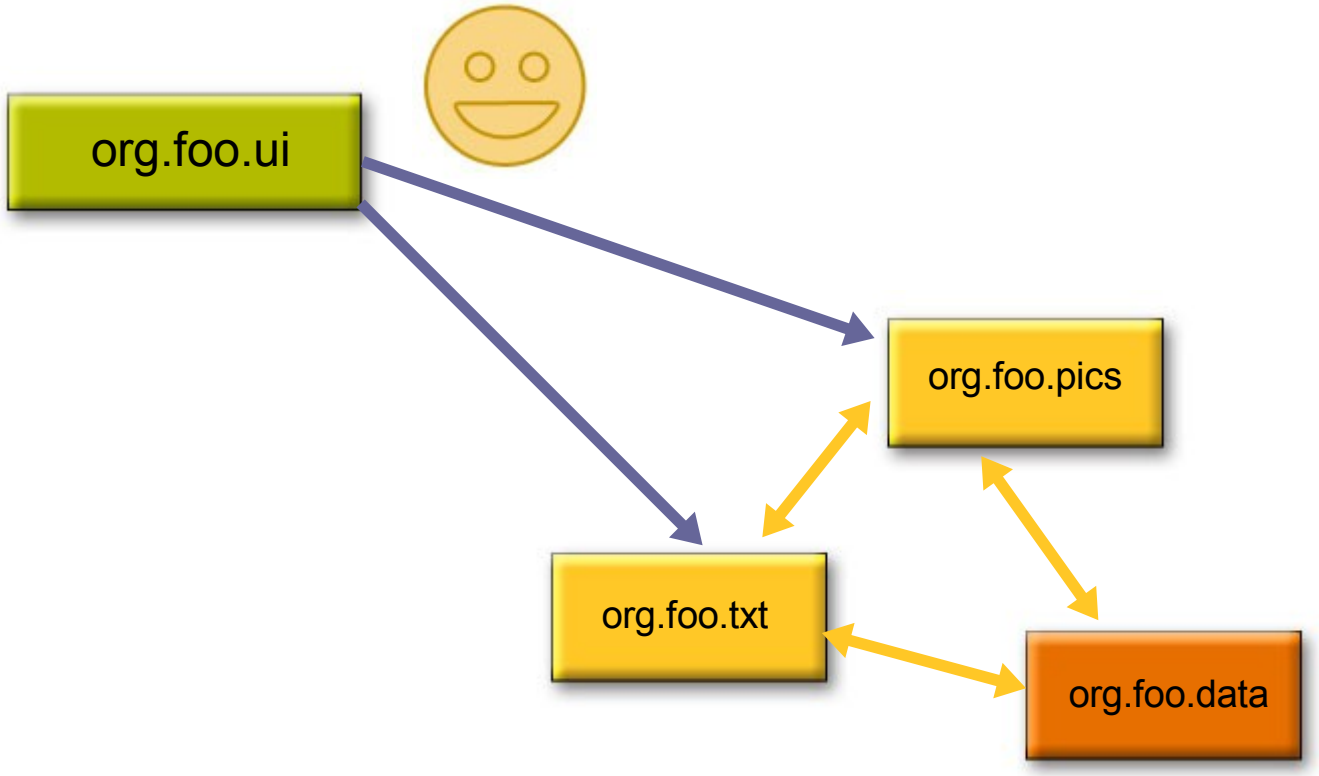
Java Technology Packages Dilemma

Java SE 6 platform



Dilemma Resolved by superpackage

Java SE 7 platform



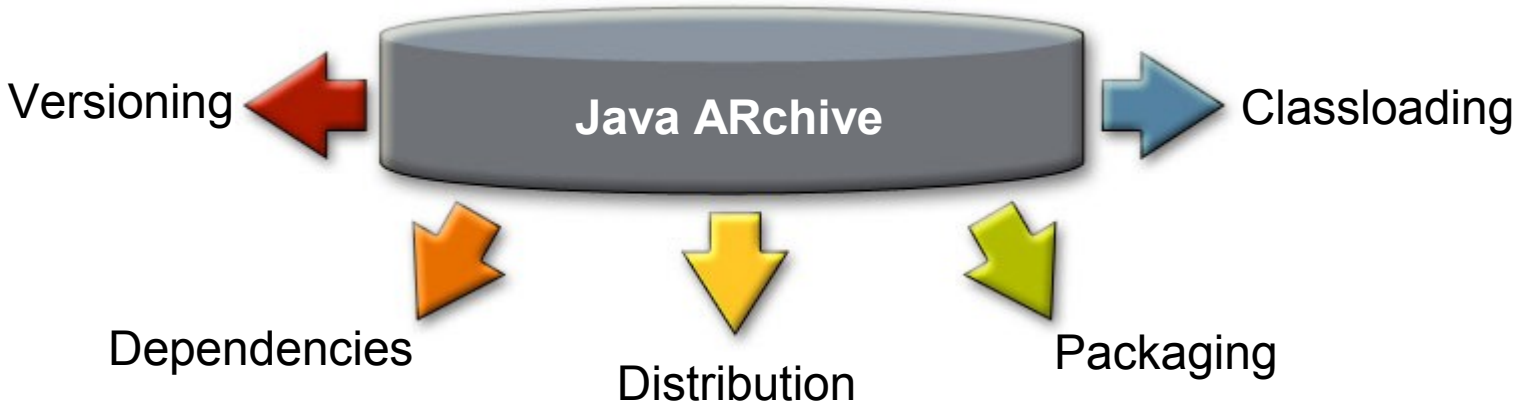
org.foo.document
'superpackage'

Draft Syntax for superpackages

```
super package org.foo.document{
   // super-package exports:
   export org.foo.pics.MyImageInterface;
   export org.foo.txt.*;
   // super-package members:
   org.foo.pics.MyImageInterface;
   org.foo.txt.*
   org.foo.data.*;
}
```

Packaging Applications

Java SE 6 platform

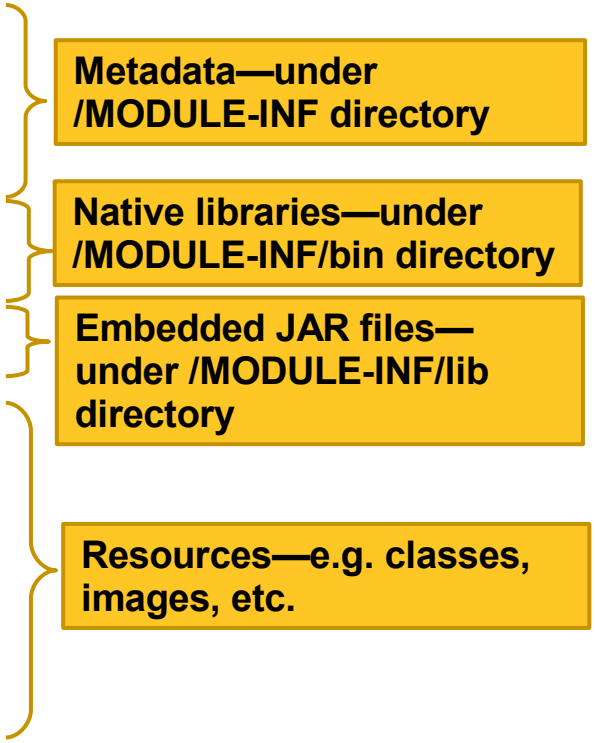


- JAR file format is stretched to its limit
  - Hard to version
  - Difficult to manage
  - Can't express dependencies
- Due for major upgrade in Java SE 7 platform
  - Java Application Modules (JAM)

Anatomy of a JAM File

Java SE 7 platform

org.foo.document-1.2.3.jam:
- /META-INF/ MANIFEST.MF
- /MODULE-INF/METADATA.module
- /MODULE-INF/bin/doc-windows.dll
- /MODULE-INF/bin/doc-linux.so
- /MODULE-INF/lib/helper.jar
- /org/foo/pics/MyImageInterface
- /org/foo/txt/Word.class
- /org/foo/txt/Sentence.class
- /org/foo/data/Letter.class
- /org/foo/data/Number.class
- /icon/graphics.jpg



Summary: Modularity in Java SE Platform

Java SE 7 platform

- Development time: superpackages
  - JSR 294: Improved Modularity Support
- Deployment time: superJARs
  - JSR 277: Java Module System
  - Aiming for interoperability with JSR 291: OSGi
- Both have open mailing lists
  - Project at openjdk.java.net/projects/modules

Swing

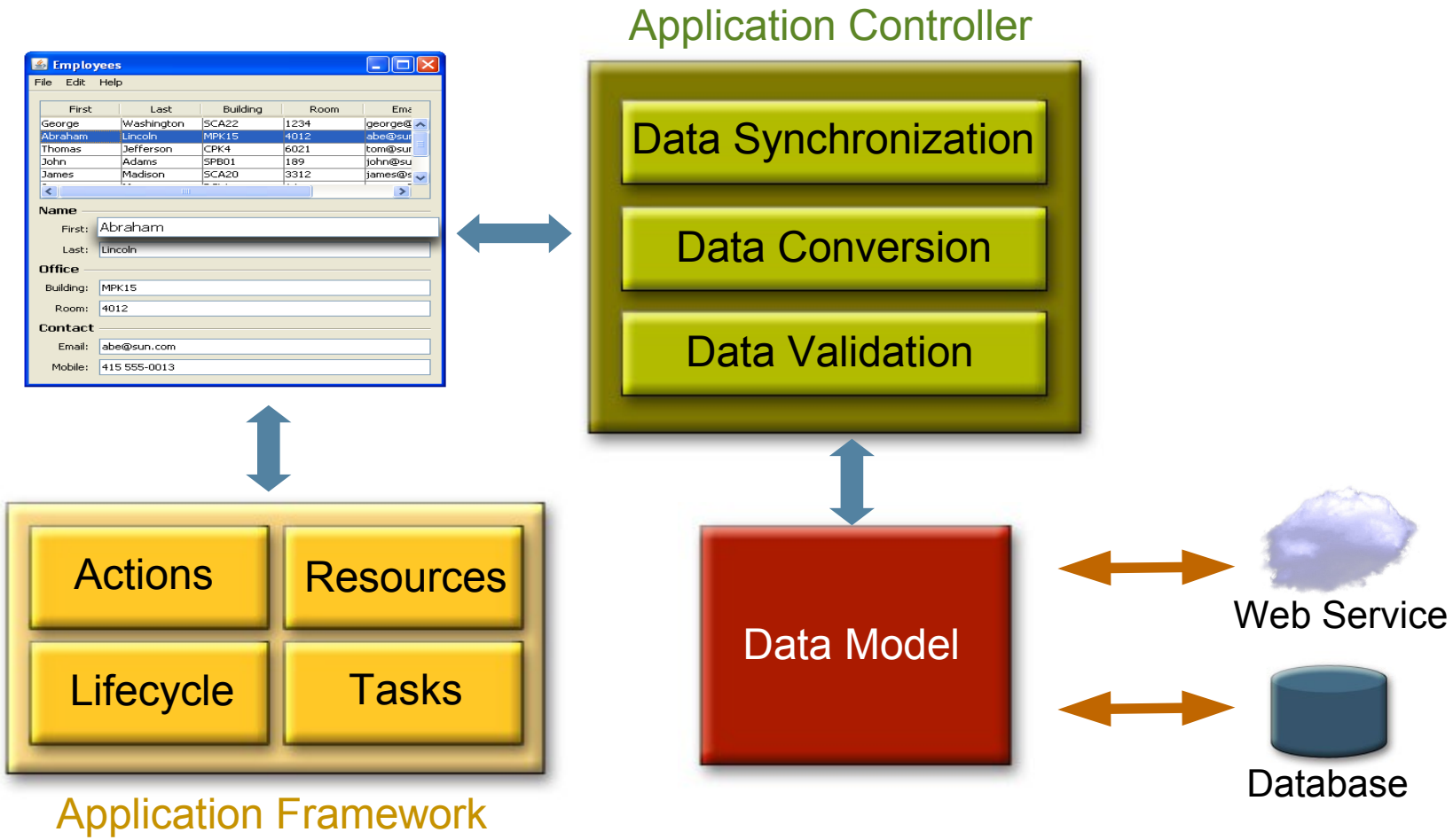
Swing Development

Java SE 6 platform

- Swing is a powerful toolkit
- Some developers are put off
  - “Routine things should be easier”
  - “There’s too much choice”
  - “I end up writing too much code”
  - “It’s difficult to teach”
- Time to make it easier!

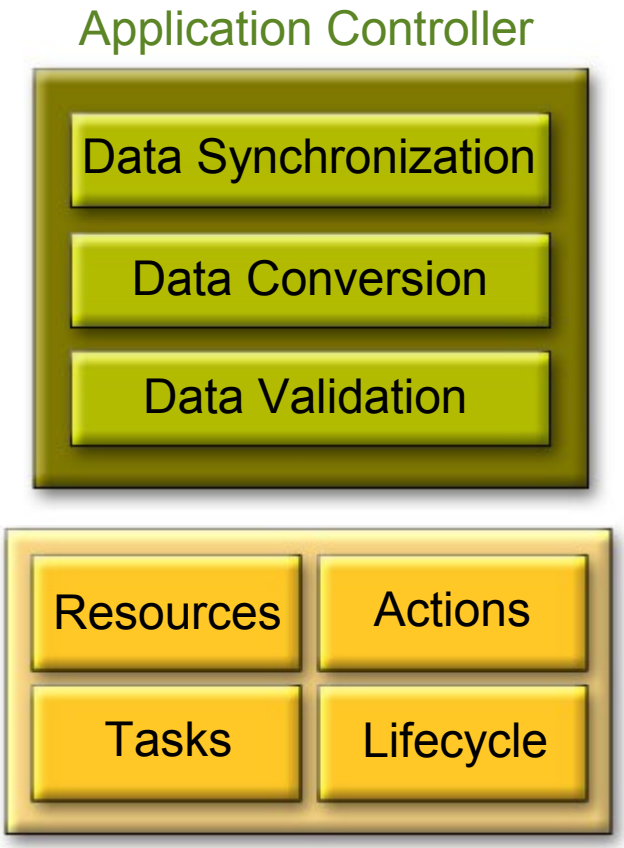
A Swing Application

Java SE 7 platform



Easier Swing Development

Java SE 7 platform



- JSR 295: Beans Binding
  - Formalizing the Swing Controller
  - API for connecting JavaBeans™ specifications
- JSR 303: Bean Validation
  - Metadata model to express validation constraints
- JSR 296: Swing Application Framework
  - Formalizing support functions

Consumer JRE

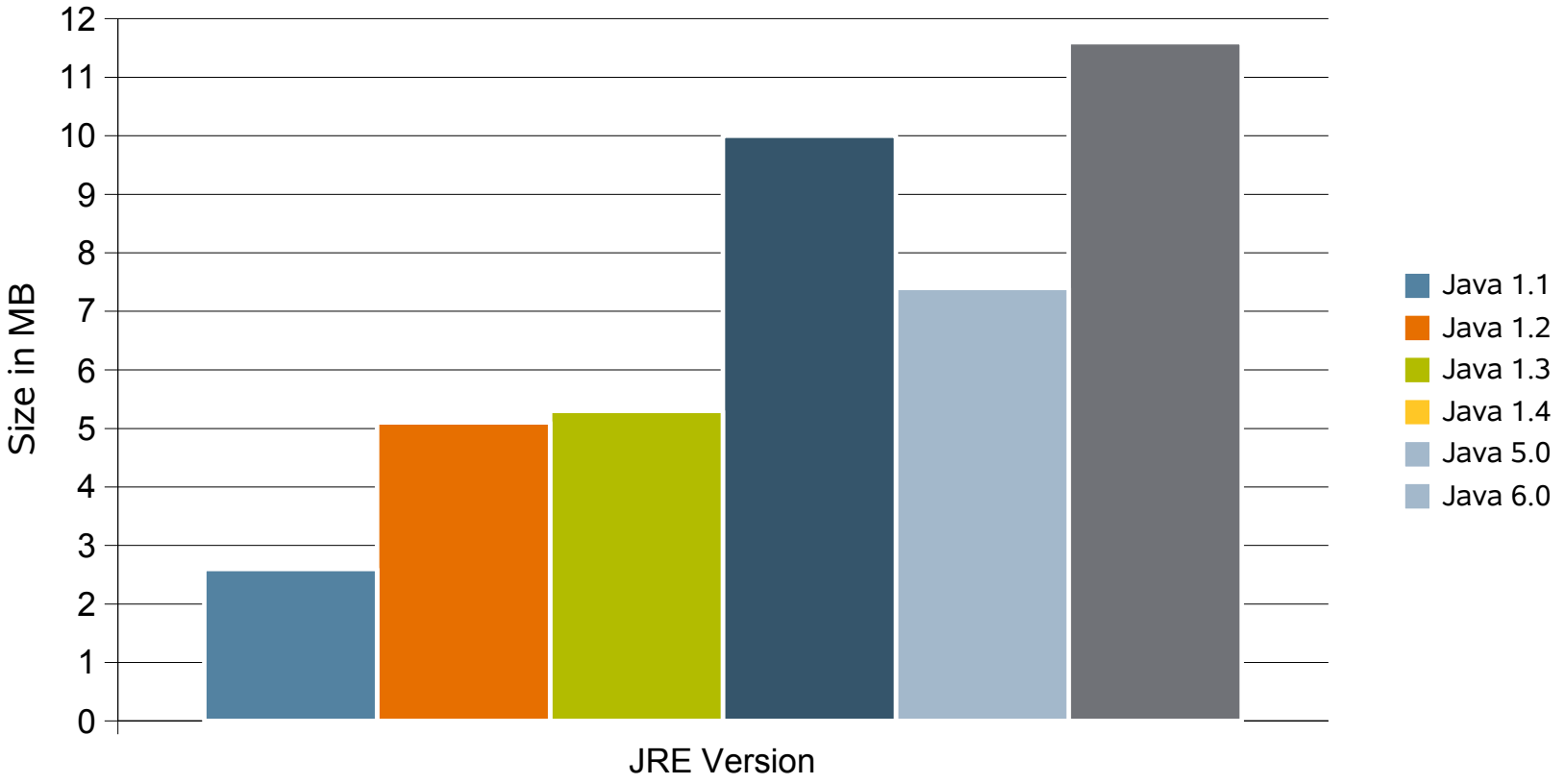
Desktop Deployment

Java SE 6 platform

- Startup time is poor
  - Especially starting from cold
- Installing is slow and complicated
  - The JRE software has become very large
- Irritating for enterprise applications, worse for consumer applications

JRE Software Download Size

Past and present



The Consumer JRE Release

Early 2008

- Quickstarter
  - Pre-load the cache, before launch
  - Not the same as having a running VM
  - Cooperates with the OS
- A radically improved install experience
- A Modular JRE version

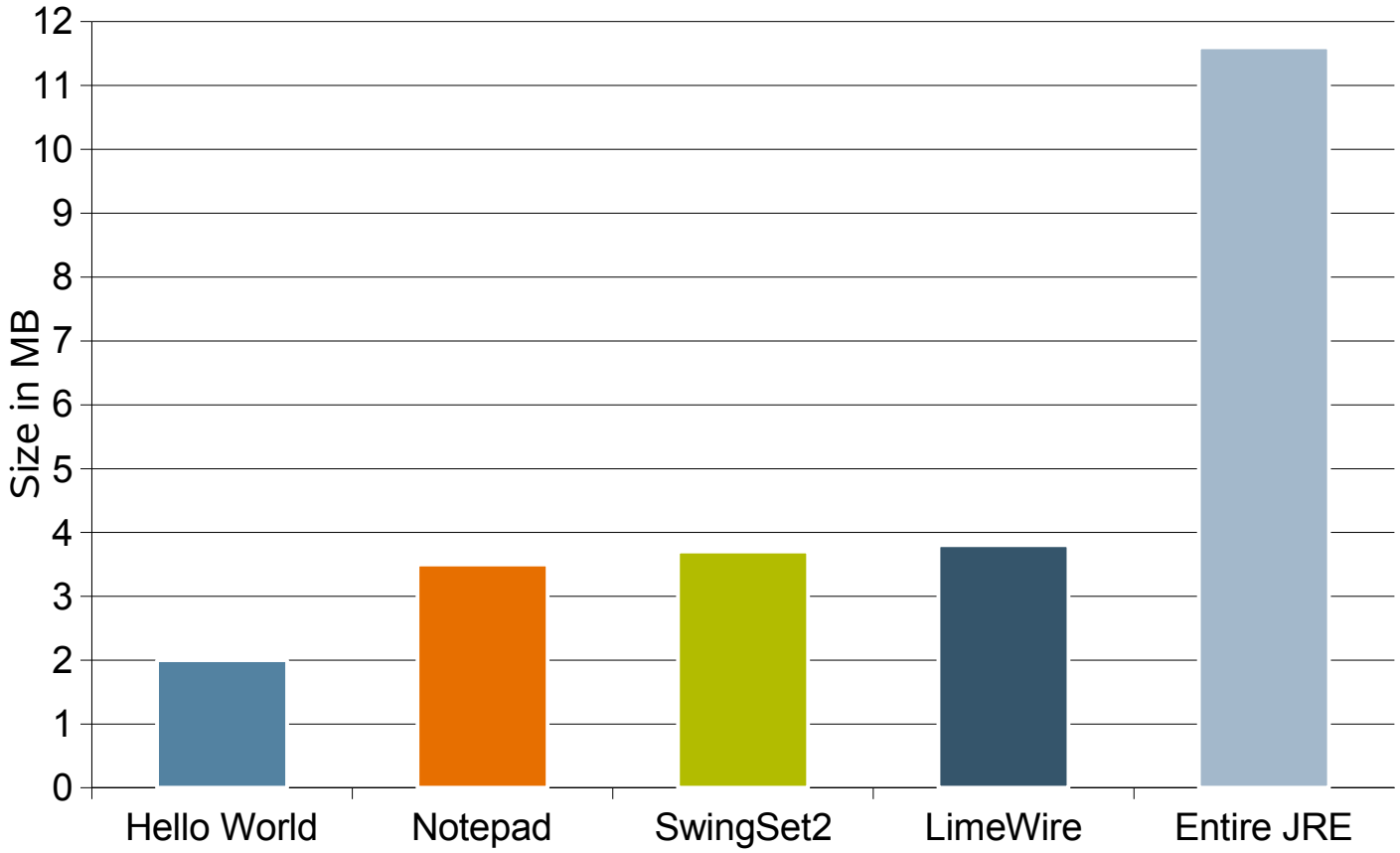
‘Java Technology-Based Kernel’ : Modularizing the JRE Software

Consumer JRE Release

- Just enough to run “Hello World”
- Install the rest in background
  - Referencing a class
  - Class.getResource() or equivalent
  - System.loadLibrary() or equivalent
- Custom JRE versions for applications that can’t pause

Promising Results

Consumer JRE Release



Some Important Updates

Management Updates

Java SE 7 platform

- JSR 255: Java Management Extensions (JMX™) Specification, version 2.0
  - Namespaces, federated JMX technology servers
  - opendmk.dev.java.net
- JSR 262: Web Services Connector for JMX Agents
  - Based on ws management standards
  - Early Draft Available
  - ws-jmx-connector.dev.java.net

More New IO APIs for Java SE Platform

Java SE 7 platform

- JSR 203: In early draft
- New file system API `java.nio.filesystem.Filesystem`
  - Listen for filesystem changes
  - Security attributes (file permissions, ACLs,…)
  - Symbolic links
  - Adds operations: copy, move, walk file tree,…
  - Scalable access to directories

Summary Proposed JSRs for Java SE 7 Platform

Subject change, based on Java Community Process℠ (JCP℠), schedule

- JSR 277: Java Module System
- JSR 294: Improved modularity support in the Java programming language
- JSR 295: Beans binding
- JSR 303: Bean validation
- JSR 292: Supporting Dynamically Typed Languages
- JSR 296: Swing application framework
- JSR 203: More new I/O APIs for the Java Platform (NIO.2)
- JSR 220: Enterprise JavaBeans™ 3.0
- JSR 255: JMX specification, version 2.0
- JSR 262: Web services connector for JMX agents
- JSR 260: Javadoc™ Tag Technology Update
- *JSR(s) TBD Java Language changes*
- JSR 308: Annotations on Java Types
- JSR 310: Date and Time API

Proposed Technologies for Java SE 7 Platform

You can look at most of these TODAY!

Specification	Draft ?	Implementation
JSR 277: Java Module System	Early Draft	openjdk.java.net/projects/modules
JSR 294: Improved Modularity Support in the Java Programming Language	-	openjdk.java.net/projects/modules
JSR 295: Beans Binding	-	beansbinding.dev.java.net
JSR 303: Bean Validation	-	-
JSR 296: Swing Application Framework	-	appframework.dev.java.net/
JSR 203: NIO.2	Early Draft	-
JSR 292: Supporting Dynamically Typed Languages	-	-
JSR 220: Enterprise JavaBeans 3.0	Final Spec	glassfish.dev.java.net
JSR 255: JMX Specification, version 2.0	-	opendmk.dev.java.net
JSR 262: Web Services Connector for JMX	Early Draft	ws-jmx-connector.dev.java.net/
JSR 260: Javadoc Tag Technology Update	-	-
JSR(s) TBD Java Language changes	-	ksl.dev.java.net
JSR 308: Annotations on Java Types	-	pag.csail.mit.edu/jsr308/
JSR 310: Date and Time API	-	jsr-310.dev.java.net/

Summary

Java SE platform present and future

- Java SE 6 platform
  - > **Use it now!**
- OpenJDK
  - > **Join us in building it!**
- Java SE 7 platform
  - > **Join us in defining it!**

- Visit us at planetjdk.org

Java™ SE: Now and Future

Sang Shin, sang.shin@sun.com
Java Technology Architect
www.javapassion.com
Sun Microsystems, Inc.