

UNITE 2005 - Minneapolis

Understanding Java/JBoss in the MCP Environment

Michael S. Recant
MGS, Inc.

Session: MCP4034
Room: Nicollet D1
1:30 pm – 2:30 pm
Tuesday, October 18, 2005

MGS, Inc.

- Software Engineering, Product Development & Professional Services firm founded in 1986
- We provide products and services to solve business problems:
 - Software Engineering Services
 - Professional Services
 - ❖ Management Consulting Services
 - ❖ Performance/Capacity Services
 - ❖ Application Services
 - Product Development
 - ❖ Performance/Capacity Monitoring
 - ❖ Web Services
 - ❖ Java/J2EE Performance

Overview

- What this presentation IS NOT:
 - This is not a Java language tutorial
- What this presentation IS:
 - Java architectural concepts
 - JBoss architectural concepts
 - MCP related specific
- It represents the results of our research into performance analysis of the MCP Java/JBoss environment

Overview

- Development history
 - “Oak” development started by Sun Microsystems in December 1990
 - ❖ 13 people secret “Green” team
 - ❖ 18 months development
 - An environment for embedded consumer electronic appliances
 - Device/OS independent language
 - 1995 – Renamed Java
 - Netscape Navigator announces support for Java applets
 - Not related to Javascript embedded HTML scripting

Overview

■ Development history

- 1996 - Java Standard Edition 1.0
- 1998 – Java 2 Standard Edition 1.2
 - ❖ Just-in-time compiler
 - ❖ Reflection
- 2000 – J2SE 1.3
 - ❖ Hotspot compiler
 - ❖ Corba
- 2002 – J2SE 1.4
- 2004 – J2SE 5.0

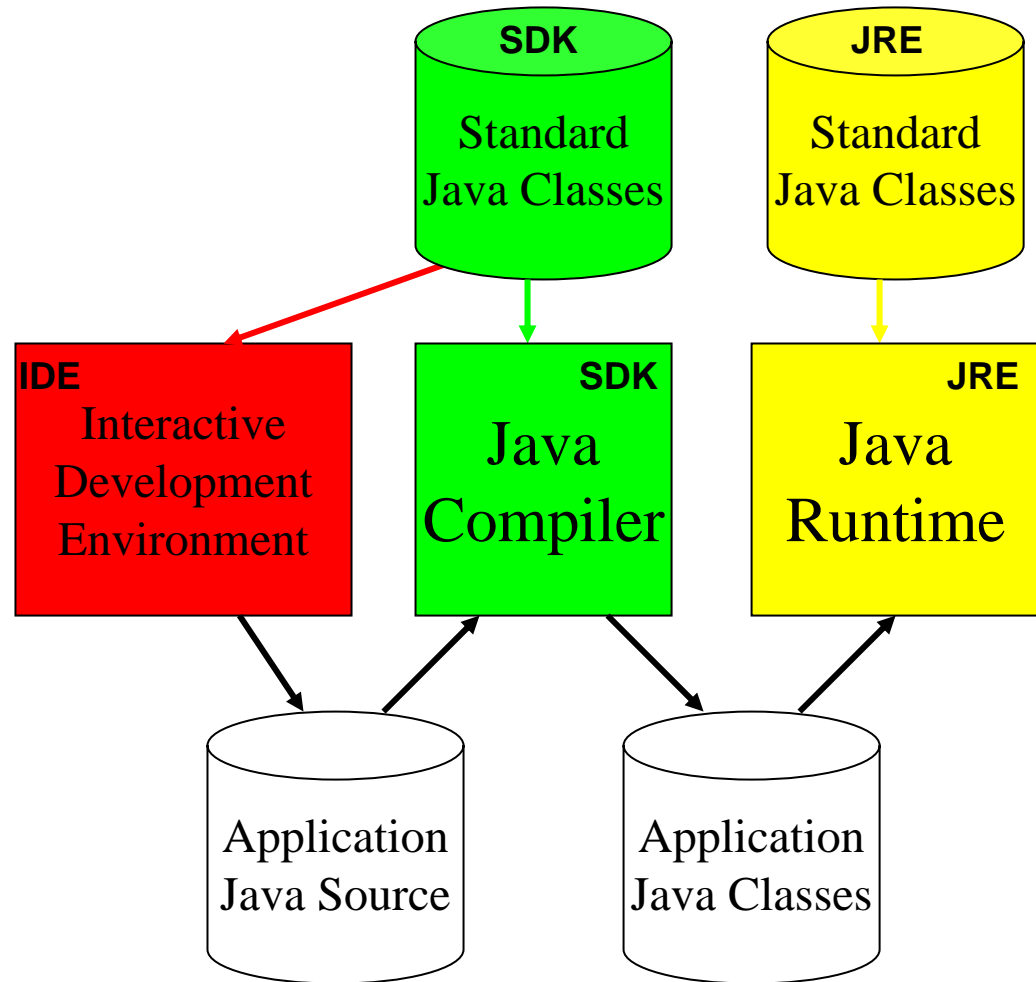
Overview

- Design Goals
 - Simple
 - Object oriented
 - Distributed
 - Interpreted
 - Reliable
 - Secure
 - Architecture neutral
 - Portable
 - Multithread
 - Dynamic
 - Motherhood, apple pie, etc...

Java Architecture

- The Java world is made up of two entities, often with the similar name
 - Software Specifications
 - ❖ Java 2 Standard Edition 1.4
 - ❖ Tomcat 5.5
 - Products
 - ❖ Sun J2SE 1.4.2_09
 - ❖ APACHE Tomcat 5.5.12

Java Architecture



Java Architecture

- IDE
Interactive Development Environment
- JDK
Java Developers Kit
 - Compiler
 - Support Utilities
- JRE
Java Runtime Environment
 - Virtual Machine
 - Support Utilities

Java Architecture

- Produces two basic types of executables
 - Applications
 - ❖ Runs in one instance of the JVM
 - Applets
 - ❖ Launched from within a browser

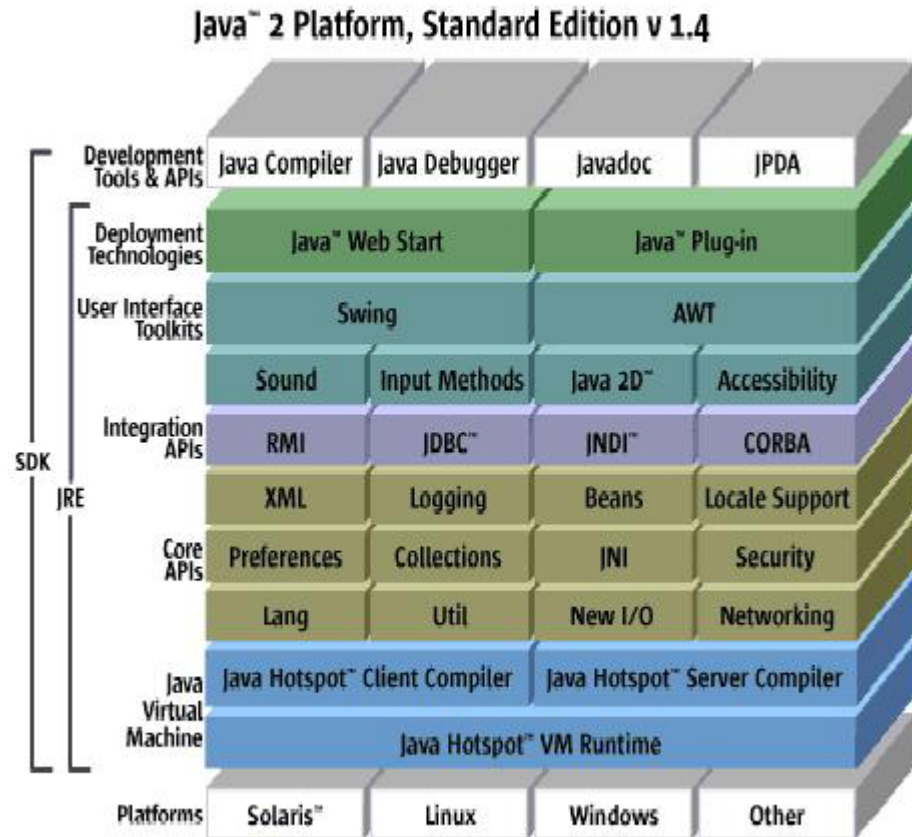
Java IDE

- The standard is just a text editor and javac (command line compiler)
- Many 3rd party products
 - Eclipse
 - JBuilder
 - Netbeans
 - JCreator
- IDE Includes
 - Source editor
 - JDK aware
 - Compile
 - Run
- Many products are open source

Java SDK

- Files and programs necessary for application development
- Compiler and Utilities written in Java
- IDEs are built on top of the standard SDK
- Uses a directory of pre-compiled standard classes
- Creates Java bytecode from Java source

Java Language



<http://java.sun.com/j2se/1.4.2/docs/index.html>

Java Language

- **Terminology**
 - **Class** – a template (definition) used to create an object (data and code)
 - **Method** – a procedure or function call to access the object's functionality
 - **Interface** – A class without the actual code or data
 - **Descendant class** – creating a new class based on an old class but with functional differences.
 - **Instance** – the actual object created from a class
 - **JNI** – Java native interface
 - **JAR File** – Class/support file package

Java Language

- Object oriented
- Java environment provides a base set of “classes” providing a number of capabilities and services
- Unisys extends that with MCP specific “classes”
- Users extend that with application/site specific “classes”

Java Language

- Application – A standalone Java program
- Started by invoking a specific class file that contains a public method called “main”
- For MCP, most standard Java classes are available

Java Language

- Applet – A Java program designed to be run from within a browser
- Applets descendant from Applet class
- Not applicable to the MCP environment

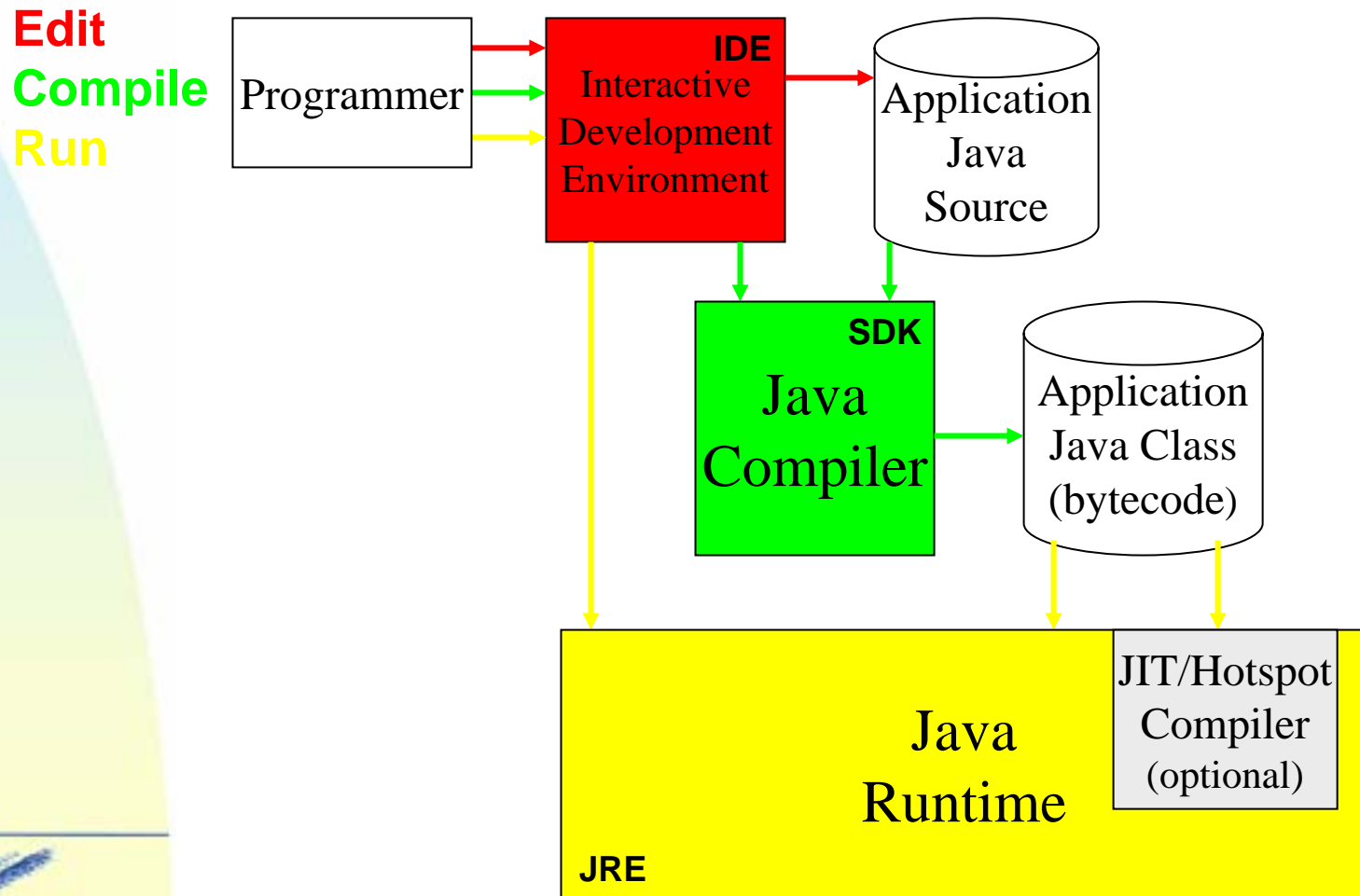
Java Runtime Environment

- Also contains a set of the standard Java class files
- Java virtual machine
- Native platform support code

Java Runtime Environment

- JRE has facilities that compiles Java Compiler bytecode into natively executable code
 - Just-in-time Compiler
 - Hotspot Compiler
- MCP does not provide JIT or Hotspot support
 - JCODE compiler available

Typical Java Development



Java under MCP

- Unisys JVM is a port of a C language Sun J2SE implementation
- Additional MCP specific JNI routines added
- Current Unisys J2SE release is 1.4.2_05
- Current Sun J2SE 1.4 release is 1.4.2_09

Java under MCP

■ List of Supported Core Classes

- java.beans
- java.security.interfaces
- java.beans.beancontext
- java.security.spec
- java.io
- java.sql
- java.lang
- java.text
- java.lang.ref
- java.util
- java.lang.reflect
- java.util.jar
- java.math
- java.util.mime
- java.net
- java.util.zip
- java.rmi
- org.omg.CORBA
- java.rmi.activation
- org.omg.CORBA.ORBPackage
- java.rmi.dgc
- org.omg.CORBA.TypeCodePackage
- java.rmi.registry
- org.omg.CORBA.portable
- java.rmi.server
- org.omg.CosNaming
- java.security
- org.omg.CosNaming.NamingContextPackage
- java.security.acl
- java.security.cert

Java under MCP

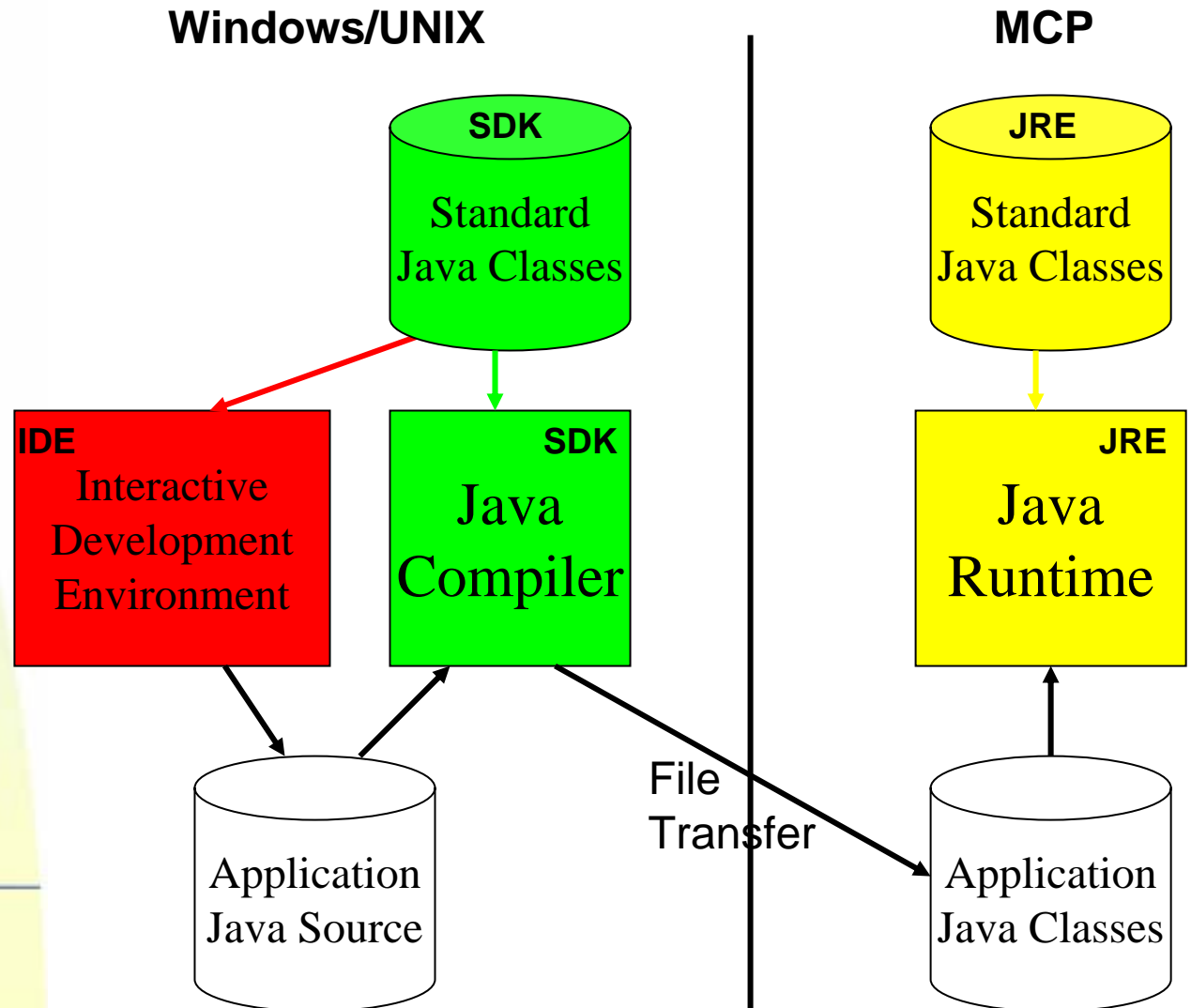
■ List of Core Classes Not Supported

- java.applet
- java.awt.imagecodec
- java.awt.accessibilityjava.awt.color
- java.awt.imagecodec
- java.awt.datatransfer
- java.awt.image.renderable
- java.awt.dnd
- java.awt.print
- java.awt.event
- java.awt.swing
- java.awt.font
- java.awt.swing.table
- java.awt.geom
- java.awt.swing.text.htm
- java.awt.image
- java.awt.swing.text.rtf
- java.awt.imagecodec
- java.awt.swing.tree
- java.awt.im
- java.awt.swing.undo

Java under MCP

- MCP Specific Classes
 - COMS class
 - TIME class
 - MCPFILE class
 - JDMI class (DMSII via Dminterpreter)
 - JHLI class (DMSII via Native interface)
- All based on JNI interface
- MCP.JAR file must be copied to your workstation and be identified to your IDE

Java under MCP



Java under MCP

- How to run a Java Application on your MCP system
 - Create a development environment on a Windows PC
 - Install the JVM on your MCP system
 - Create and compile a simple Java application under Windows
 - Copy the class files to your MCP permanent directory
 - Run OBJECT/JAVA specifying your application name

Java under MCP

- Create a development environment on a Windows PC
 - <http://java.sun.com/j2se/1.4.2/download.html>
 - Download/install:
 - ❖ Java Runtime (JRE)
 - ❖ Java Developers Kit (JDK)
 - ❖ Java Documentation
 - <http://www.jcreator.com/download.htm>
 - Download/Install JCreator LE 3.50
 - ❖ Direct to JDK, JRE and JavaDocs

Java under MCP

- Install the JVM on your MCP system
 - Turn on permanent directories and long file names
 - ❖ SYSOPS PERMDIRECTORIES+
 - ❖ SYSOPS LONGFILENAMES+
 - See Unisys documentation in “Virtual Machine for the Java Platform on ClearPath MCP #3835 4759-001”
 - Establish a permanent directory for your Java application
 - ❖ WFL MKDIR *DIR/MYJAVA ON DISK

Java under MCP

- Create and compile a simple Java application under Windows
 - Run JCreator
 - ❖ FILE | NEW | BLANK WORKSPACE
 - ❖ FILE | NEW | PROJECT | BASIC JAVA APPLICATION
 - ❖ Open the main source file
 - ❖ Comment out creation and method call on object FRAME
 - ❖ ADD “System.out.println(“Hello world”);” in “main”
 - ❖ Click BUILD | COMPILE PROJECT

Java under MCP

- Create and compile a simple Java application under Windows
 - Hello World Application:

```
public class Demo1 {  
    public static void main(String[] args) {  
        System.out.println("Hello world");  
    }  
}
```

Java under MCP

- Copy the class files to your MCP permanent directory
 - Copy file
`<path/workspace name>/CLASSES/<project name>.class`
to `*DIR/MYJAVA ON DISK`
 - It is convenient to make the permanent directory a windows shared folder

Java under MCP

- Run OBJECT/JAVA specifying your application name

```
RUN *OBJECT/JAVA  
("-cp /-/DISK/DIR/MYJAVA <project name> ")
```

```
RUN *OBJECT/JAVA("<project name>");  
CURRENTDIRECTORY="/-/DISK/DIR/MYJAVA"
```

Or set your usercode's USERDATA attribute
"POSIXINITDIR" to "/-/DISK/DIR/MYJAVA"

- <project name> case must match
- Classes must be in a permanent directory, JAR files can be anywhere (-JAR command line option)
- Note use of POSIX file names
/-/<family>/USERCODE/<filename>

Java under MCP

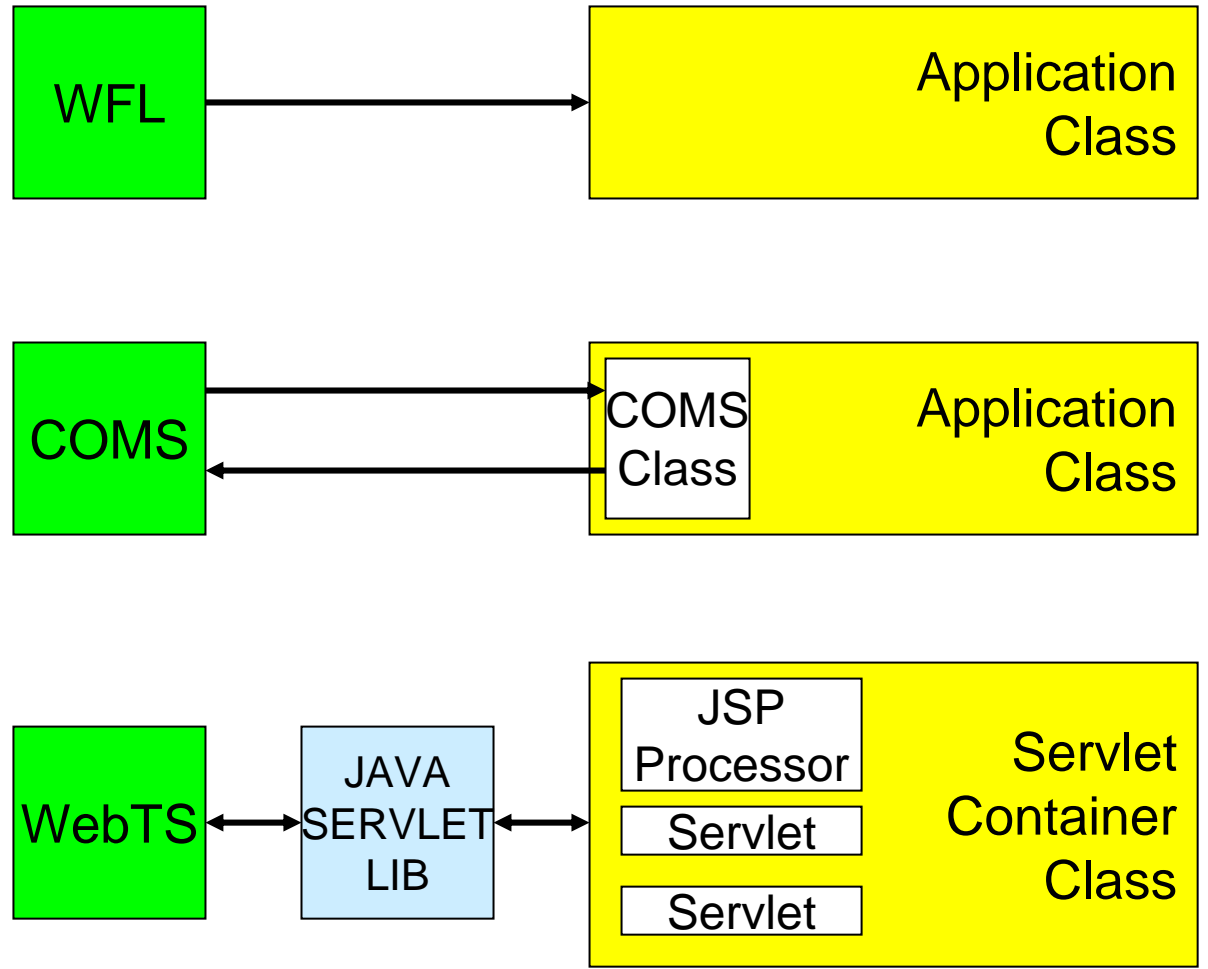
- OBJECT/JAVA is the MCP JVM
 - One OBJECT/JAVA per application run
 - No direct communication between JVMs
 - All resource statistics are logged against OBJECT/JAVA
 - Threads show in MIX but not in Sumlog
 - Efficiency: MCP JCODE Compiler
 - ❖ User compiles a Java class normally
 - ❖ MCP provides JCODE compiler
 - ❖ User binds JCODE compiler output into a specific OBJECT/JAVA code file

Server Side Java

- Two type of JVMs
 - Client side (presentation)
 - Server side (functionality)
- The JVM for MCP systems is a Server side implementation
 - No applets
 - No GUI (awt or swing)
- Non JBoss MCP Java Processing
 - Batch application
 - COMS Transaction Processor
 - Web Servlet

Server Side Java

OBJECT/JAVA



Server Side Java

- Batch Application
 - One application per run of OBJECT/JAVA
 - May utilize a variety of classes
 - Communicates with other applications via
 - ❖ Files
 - ❖ DMSII
 - ❖ TCP/IP socket
 - ❖ RMI (Remote Method Invocation)
 - ❖ Corba
 - ❖ user coded JNI interface

Server Side Java

- COMS Transaction Processor
 - Defined to COMS as a run of OBJECT/JAVA with specific parameters
 - Java application uses MCP COMS classes for normal COMS communication
 - ❖ Direct Window
 - ❖ Remote file

Server Side Java

■ Web Servlet

- MCP JVM comes with a Servlet Container
 - ❖ A special Java application that loads user created Servlet classes
 - ❖ Routes web requests to specific servlets
 - ❖ This is J2EE/JBoss technology
- Web TS can be configured to route specific web transactions to the Servlet Container
- Servlets provide server side processing for Web transactions
- The Servlet API is one of the Java standards
- The Java Server Page (JSP) interpreter runs as a servlet

Server Side Java

- Data Access under MCP JVM
 - Java standard IO objects
 - Unisys MCPFILE classes
 - Unisys DMSII classes (JDMI, JHLI)
 - Java Database Connectivity API (JDBC)
 - ❖ Supports any Type 4 driver (pure Java)
 - ❖ Paradigm
 - Connection
 - Statement
 - Result Set

J2EE/JBoss

- Java 2 Enterprise Edition (J2EE) is a specification for a fully functional application server.
- JBoss Application Server 3.2 is an open-source J2EE compliant implementation with some JBoss specific extensions
- Unisys has chosen JBoss as the J2EE framework to deploy on its MCP, OS2200 and Windows Enterprise servers

J2EE/JBoss

- Generally, JBoss is a pure Java implementation with the following exceptions:
 - WebTS Tomcat/Coyote Connector implementation
 - MCP Userdatafile Authentication
- MCP JBoss files can be run on your desktop PC
 - Copy *DIR/JBOSS to a folder on your PC (eg. C:\JBoss)
 - Run C:\JBoss\bin\run.bat

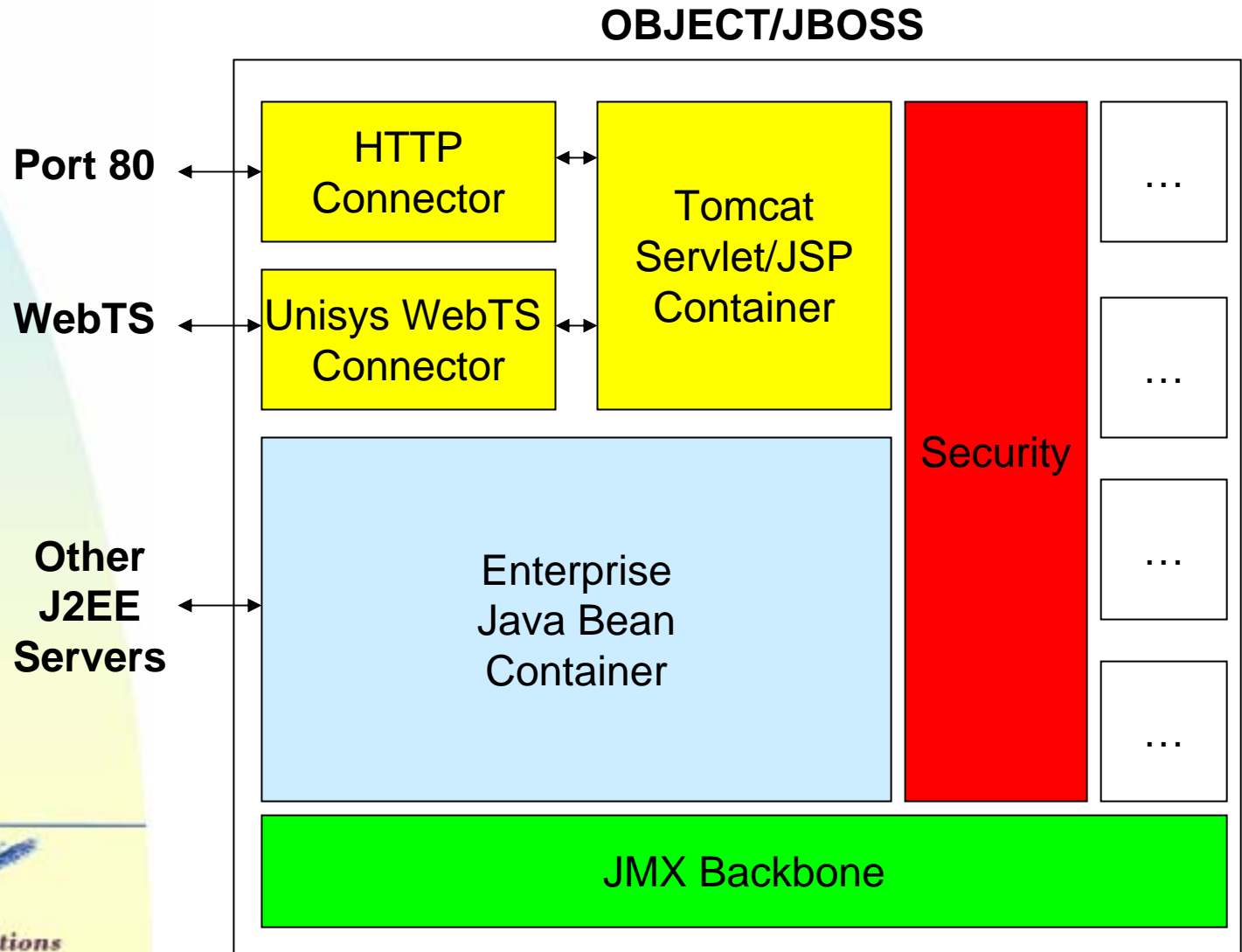
JBoss Architecture

- Java Management eXtensions (JMX)
 - A framework to discover, initiate, manage and monitor JBoss capabilities
 - Any component, either vendor or application, can participate in JMX management through MBeans

JBOSS Architecture

- Main JBoss Components
 - Tomcat connectors
 - Servlet container
 - Enterprise Java Bean container
 - Security
- JBoss Components compiled with JCODE compiler and bound into a special copy of the JVM called OBJECT/JBOSS

JBoss Architecture

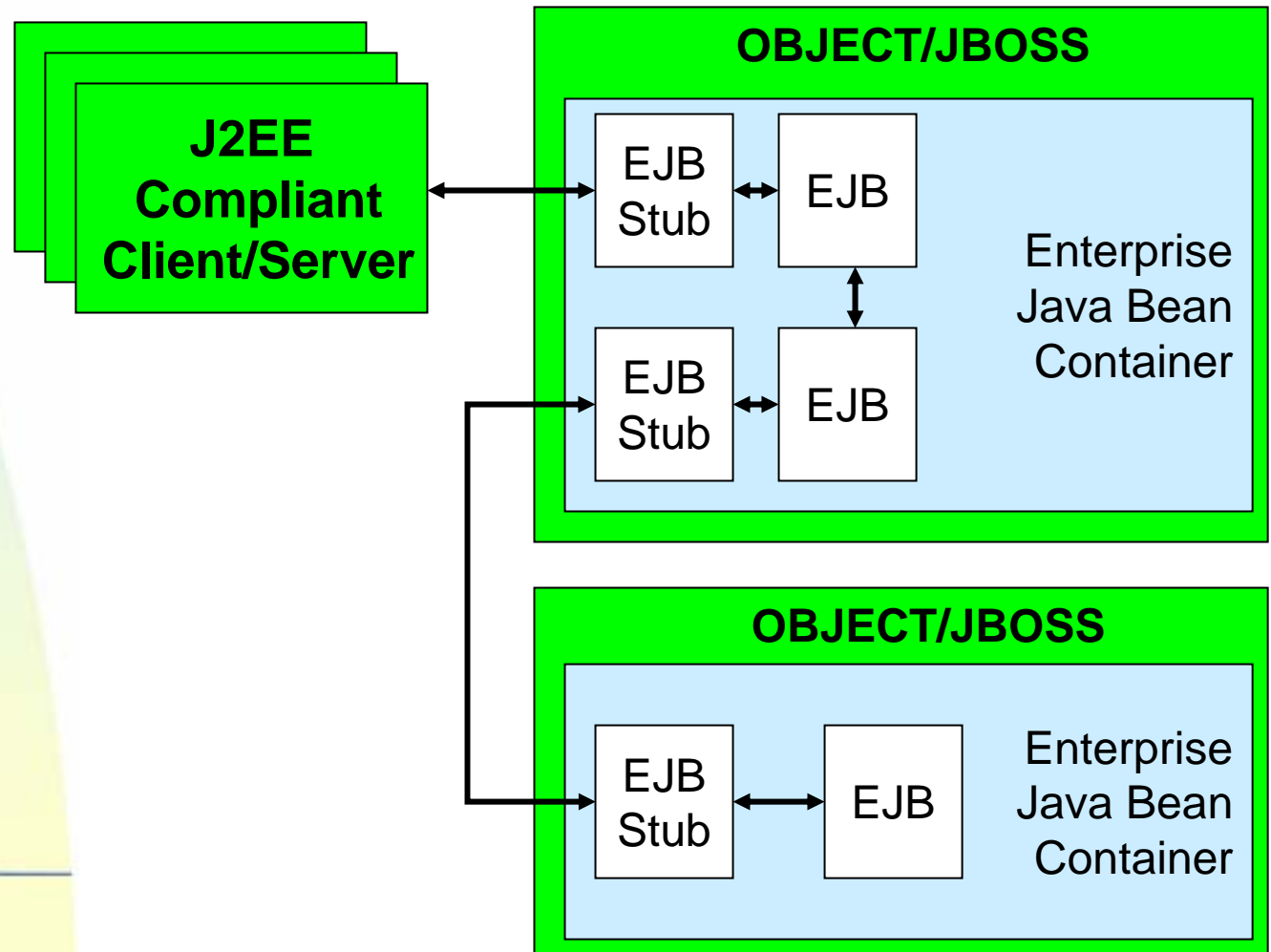


JBoss Architecture

■ JBoss Functionality

- JMX – Java Management eXtensions
- EJB – Enterprise Java Beans
- Web - Web Server, Coyote Connector, Servlet Container, JSP Processor
- JAAS – Java Authentication and Authorization Service
- WS - Web Services
- Hibernate – Object relational/Query service
- JTA – Java transaction manager
- JMS – Java Message Service
- JCA – J2EE Connector architecture
- JNDI – Java Naming and Directory Services
- JBoss Mail Server

JBoss Architecture



References

■ Literature References

- ClearPath Enterprise Servers, JBoss Application Server for ClearPath MCP, Installation Administration and Programming Guide, MCP 10.0 (4729 2172-001)
- ClearPath Enterprise Servers, Virtual Machine for the Java Platform on ClearPath MCP, Installation Instructions, MCP 10.0 (3835 4759-001)
- ClearPath Web Enablement Solutions, Java Servlet API for ClearPath MCP, Programming Guide (4310 4462-005)
- ClearPath Enterprise Servers, Virtual Machine for the Java™ Platform on ClearPath MCP, Programming Guide (4199 9913-006)
- SAMS, Teach Yourself Java 2 in 21 Days, SAMS/Cadenhead & Lemay
- Introduction to Java Programming, Comprehensive Version, Prentice Hall/Liang

References

■ Web References

- Sun Microsystems J2SE documentation
<http://java.sun.com/docs/index.html>
- Sun Microsystems J2EE documentation
<http://java.sun.com/j2ee/1.4/docs/index.html>
- JBoss Application Server Doc Library
<http://www.jboss.com/products/jbossas/docs>
- JBossWiki
<http://www.jboss.com/wiki/Wiki.jsp>

Additional Questions?

**Michael S. Recant,
VP Software Development**

**MGS, Inc.
10901 Trade Road, Suite B
Richmond, VA 23236**

**Voice: (804)379-0230
Fax: (804)379-1299
Email: Mike.Recant@mgsinc.com
Web: www.mgsinc.com**

(presentation is available on web site)

UNITE 2005 - Minneapolis

Understanding Java/JBoss in the MCP Environment