1

Introduction to Java

Objectives

After completing this lesson, you should be able to do the following:

- Identify the key elements of Java
- Describe the role of the Universal Virtual Machine (UVM)
- Describe how Java is used to build applications
- Identify the key components of the Java SE Java Development Kit (known as JDK or SDK)
- List Java deployment options

What Is Java?

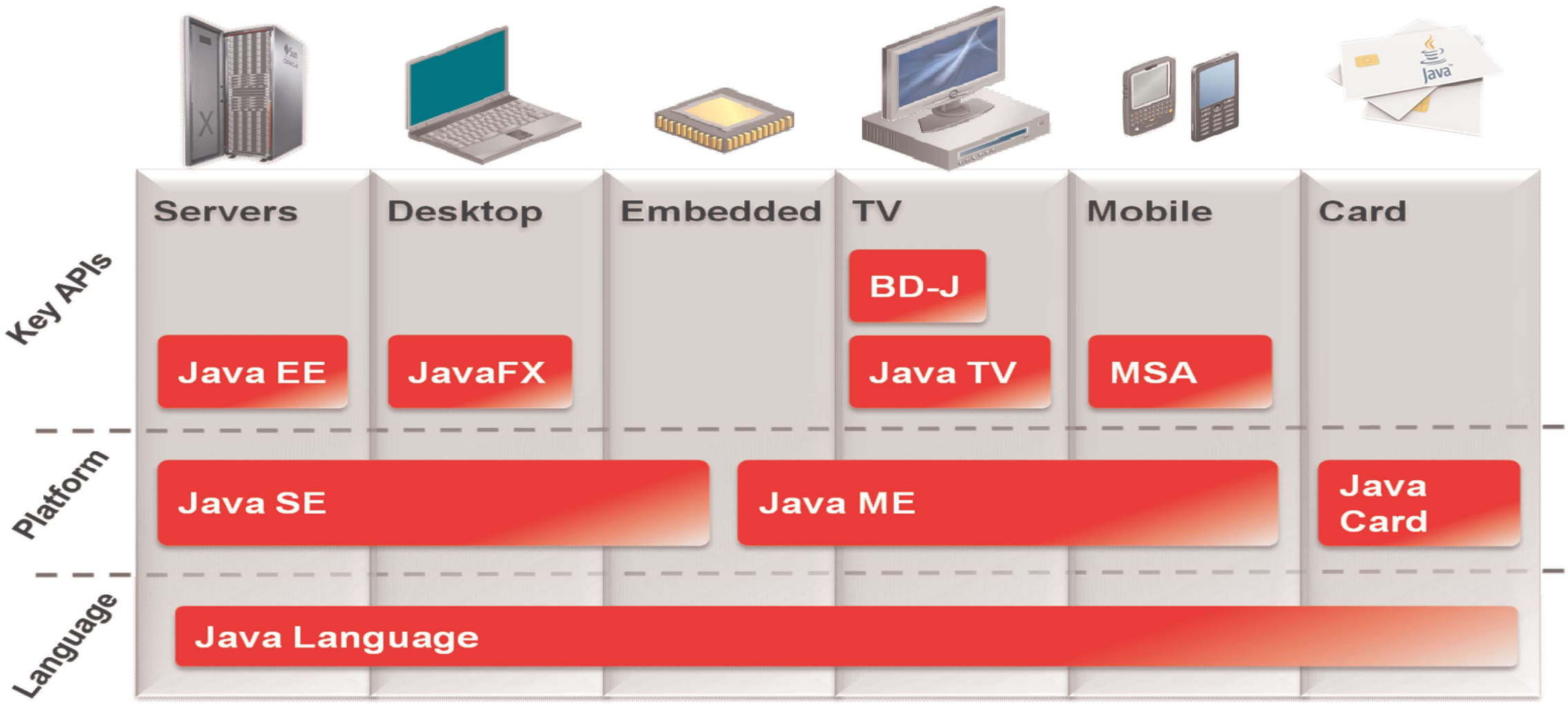
Java:

- Is a platform and an object-oriented language
- Was originally designed by Sun Microsystems for consumer electronics
- Contains a class library
- Uses a virtual machine for program execution

Key Benefits of Java

- Object oriented
- Interpreted and platform independent
- Dynamic and distributed
- Multithreaded
- Robust and secure

Java Technology Product Groups

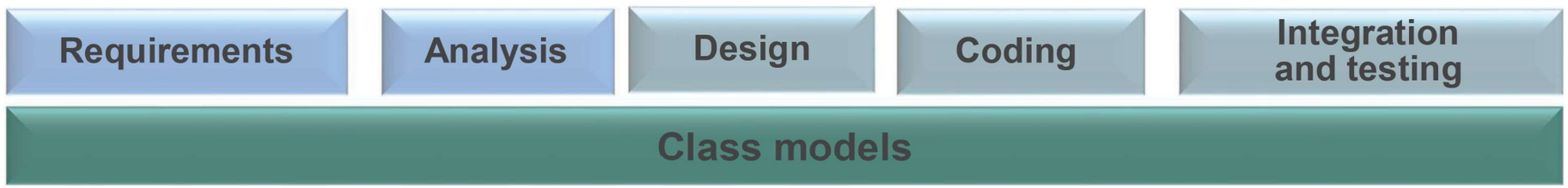


Java SE Platform Versions

Year	Developer Version (JDK)	Platform
1996	1.0	1
1997	1.1	1
1998	1.2	2
2000	1.3	2
2002	1.4	2
2004	1.5	5
2006	1.6	6
2011	1.7	7
2014	1.8	8
2018	11	11

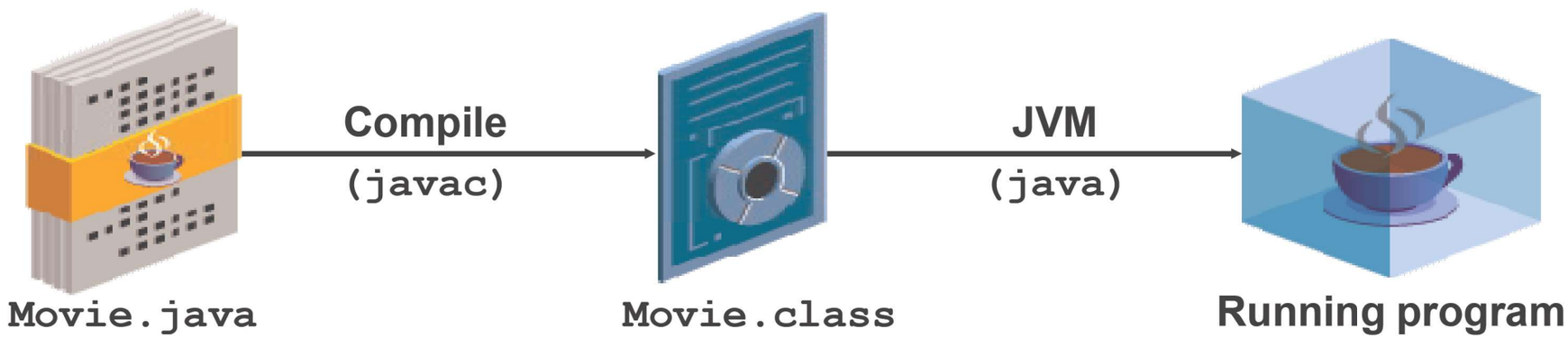
Object-Oriented Approach

- Objects and classes:
  - An object is a run-time representation of a “thing.”
  - A class is a “static definition of things.”
- Class models elaborate:
  - Existing classes and objects
  - Behavior, purpose, and structure
  - Relationships between classes
  - Relationships between run-time objects
- Same models exist throughout the project.

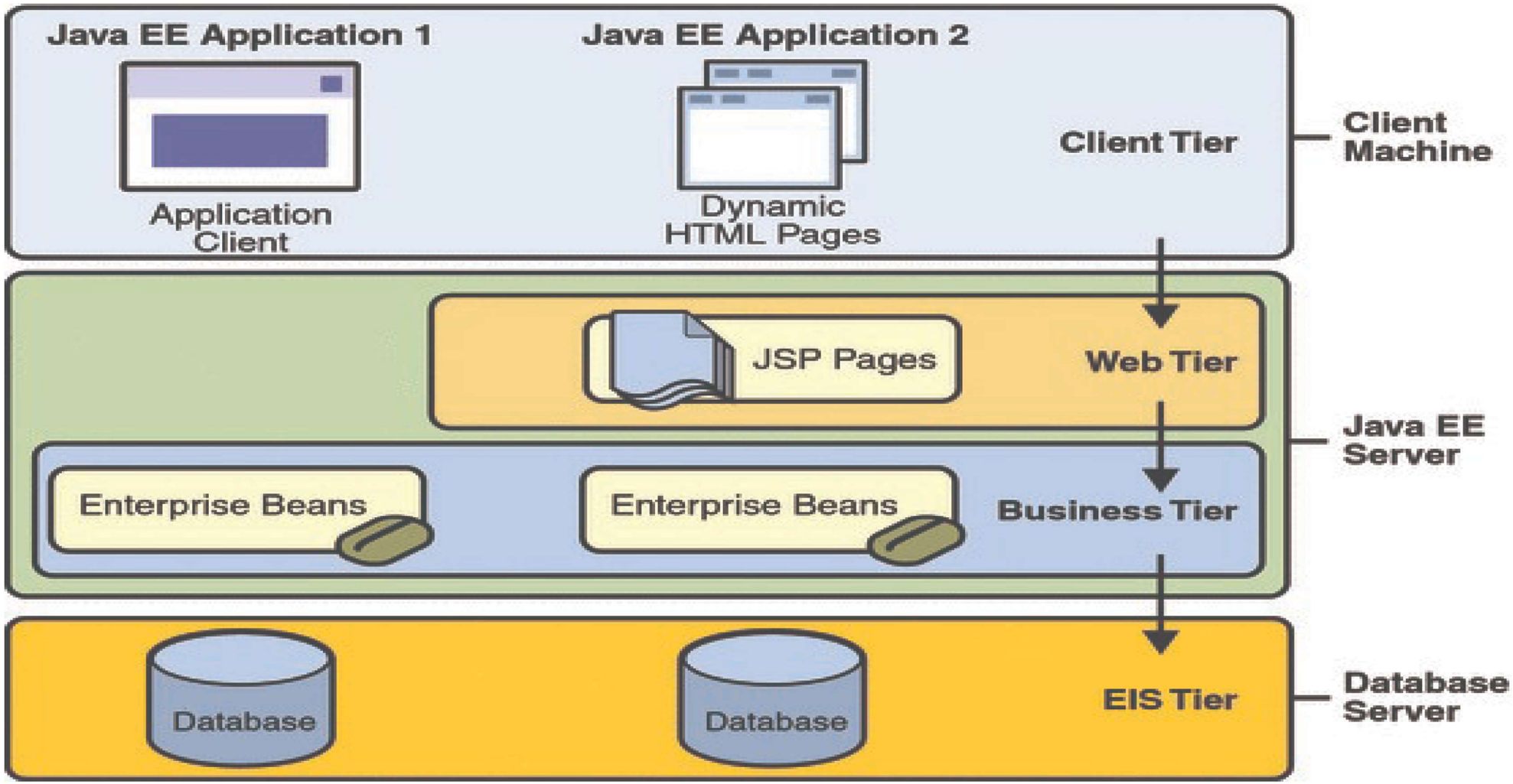


Platform Independence

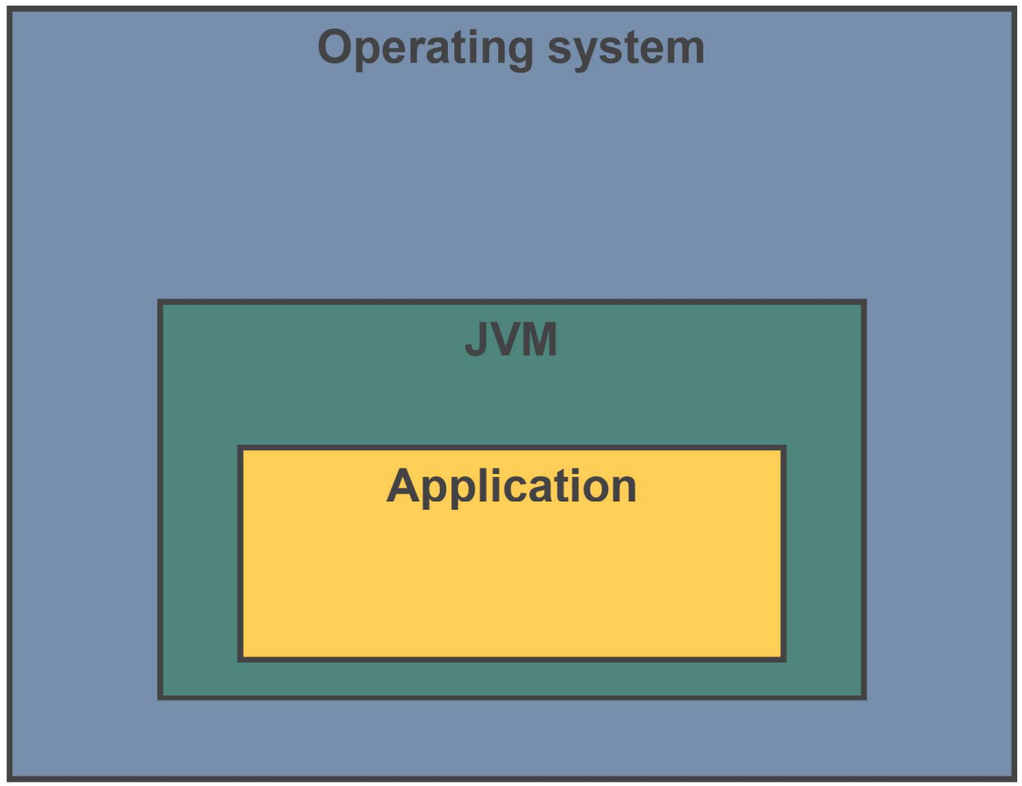
- Java source code is stored as text in a `.java` file.
- The `.java` file is compiled into `.class` files.
- A `.class` file contains Java bytecodes (instructions).
- The bytecodes are interpreted at run time.
  - The Java `.class` file is the executable code.

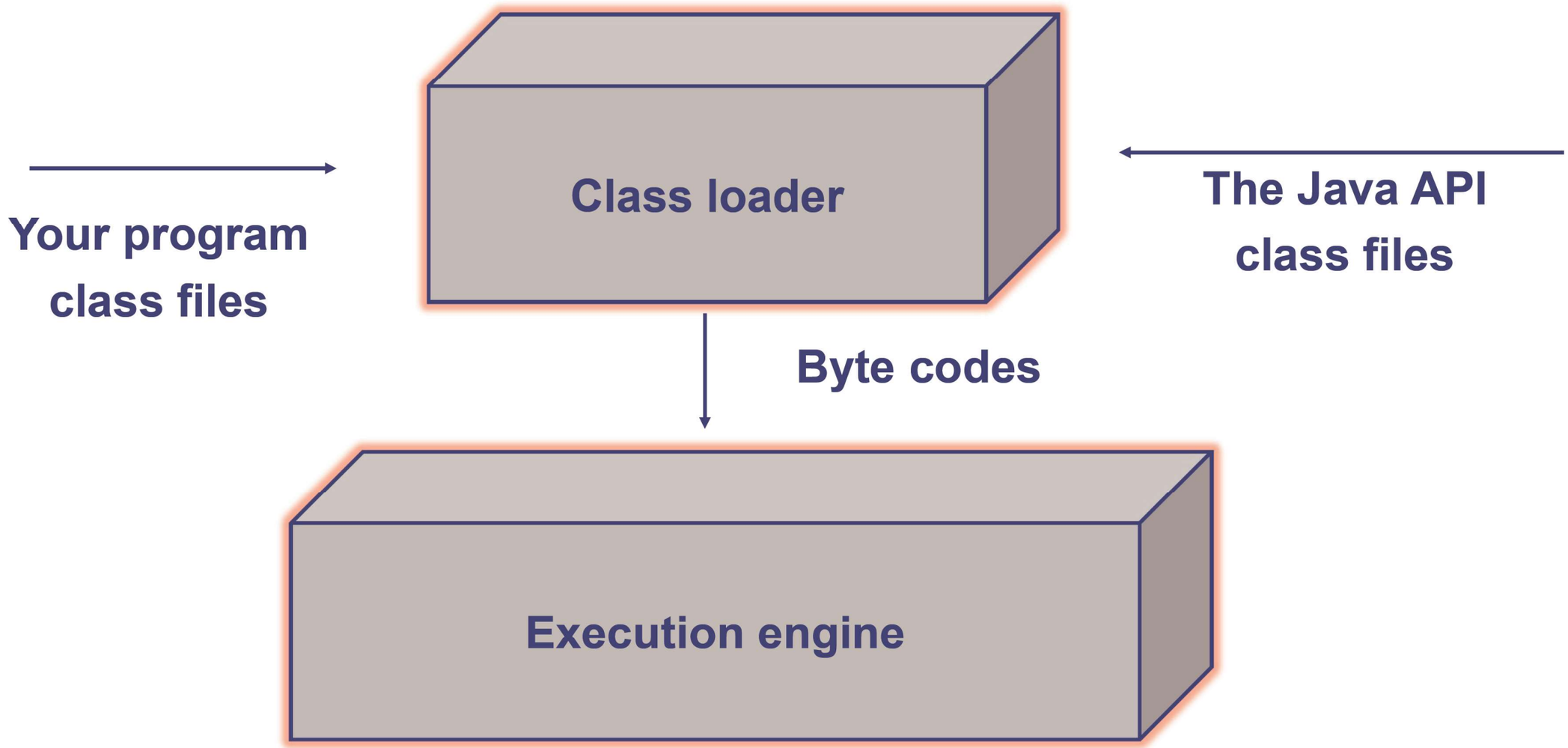


Using Java with Enterprise Internet Computing

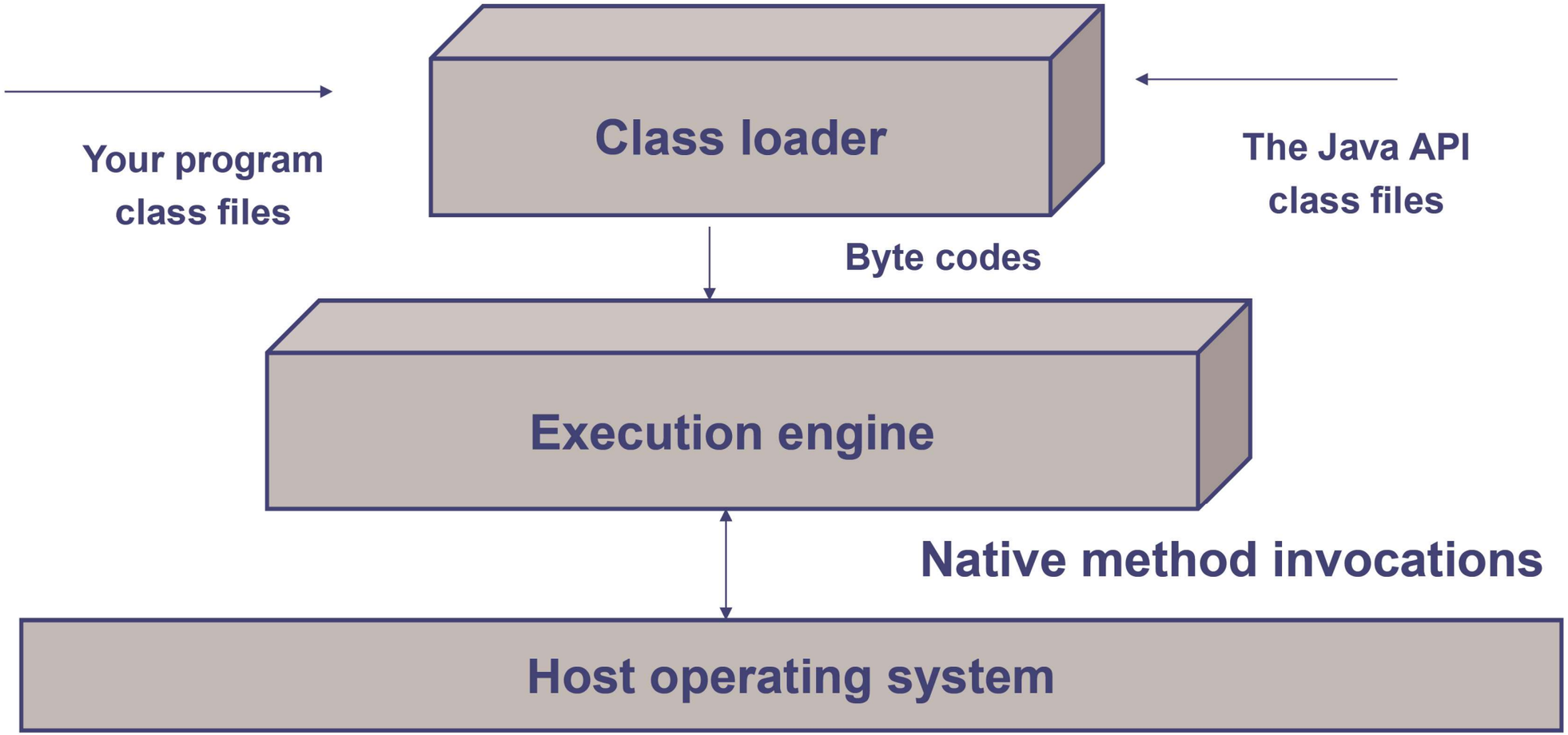


Using the Java Virtual Machine





Basic block diagram of the Java virtual machine



A Java virtual machine implemented in software on top of a host operating system

How Does the JVM Work?

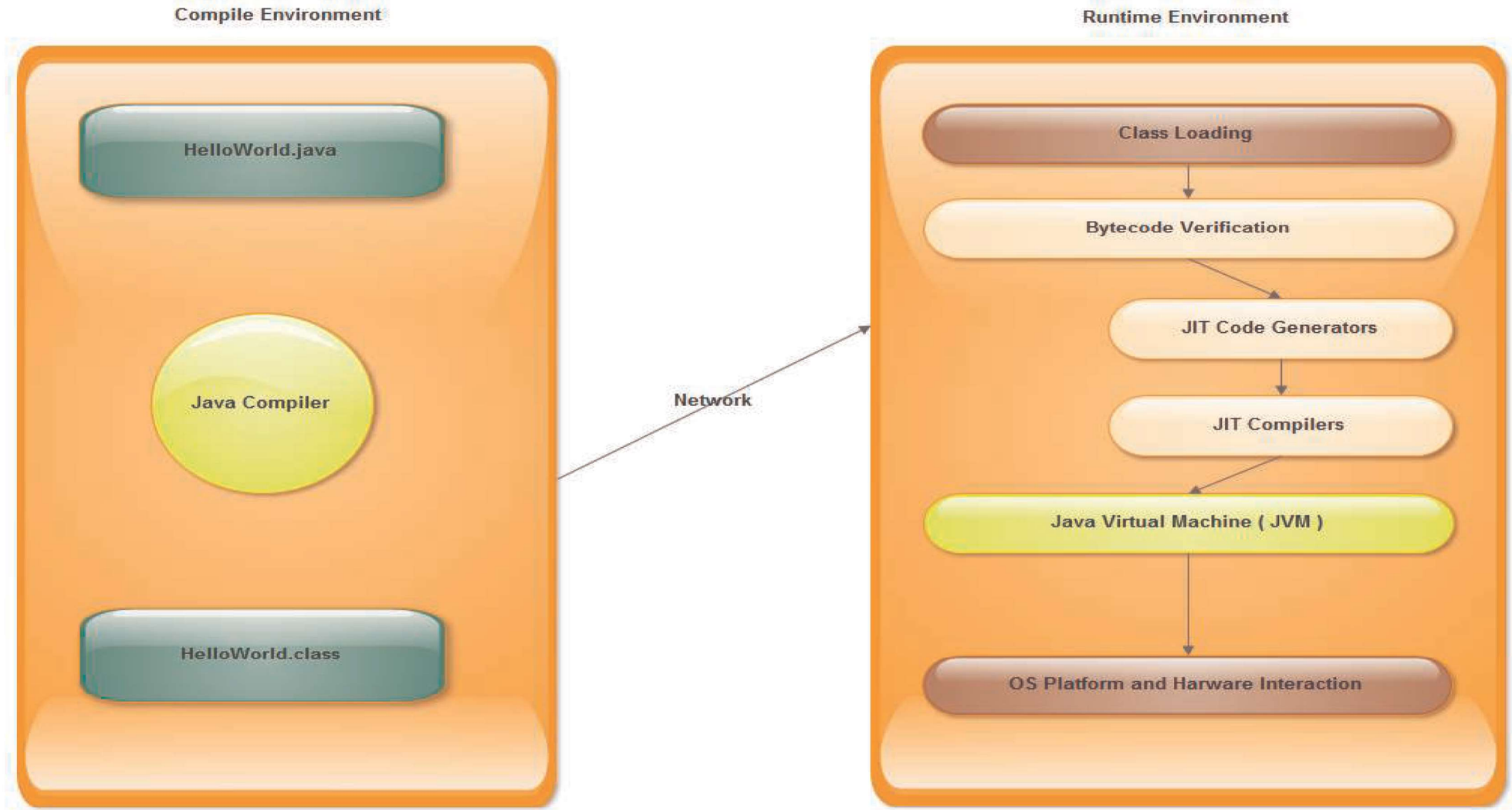
- Class loader loads all required classes.
  - JVM uses a `CLASSPATH` setting to locate class files.
- JVM Verifier checks for illegal bytecodes.
- JVM Verifier executes bytecodes.
  - JVM may invoke a just-in-time (JIT) compiler.
- Memory Manager releases memory used by the dereferenced object back to the OS.
  - JVM handles garbage collection.

Benefits of Just-In-Time (JIT) Compilers

JIT compilers:

- Improve performance
- Are useful if the same bytecodes are executed repeatedly
- Translate bytecodes to native instructions
- Optimize repetitive code, such as loops
- Use Java HotSpot VM for better performance and reliability

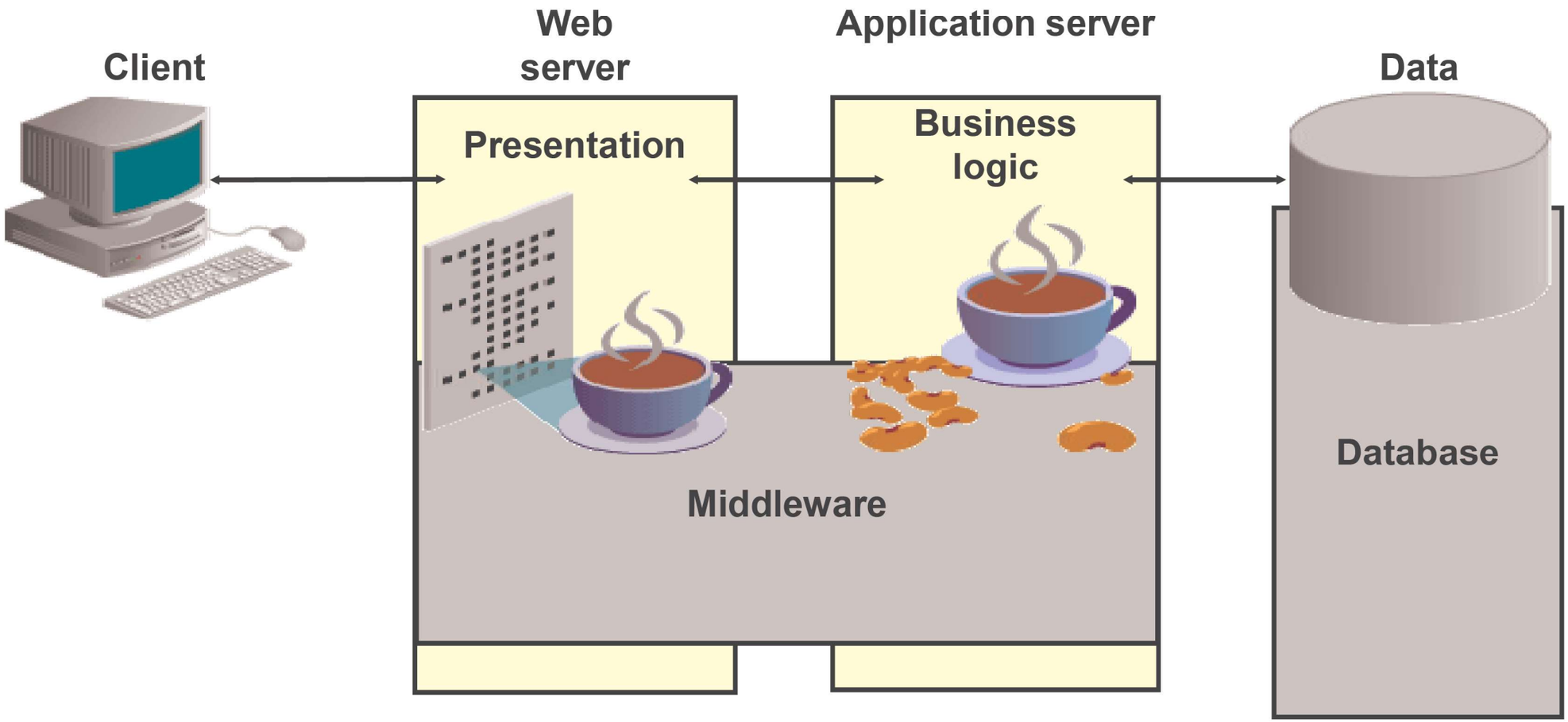
Implementing Security in the Java



Deployment of Java Applications

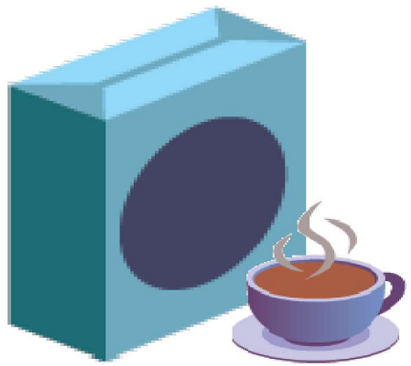
- Client-side deployment:
  - JVM runs stand-alone applications from the command line.
  - Classes are loaded from a local disk, thereby eliminating the need to load classes over a network.
- Server-side deployment:
  - Serves multiple clients from a single source
  - Is compatible with a multitier model for Internet computing

Using Java with Oracle 11*g*



Java Software Development Kit

- Sun Java SE (known as JDK and Java SDK) provides:
  - Compiler (`javac`)
  - Core class library
    - `rt.jar`
  - Debugger (`jdb`)
  - Bytecode interpreter: JVM (`java`)
  - Documentation generator (`javadoc`)
  - Java Archive utility (`jar`)
  - Others



Java SE

Using the Appropriate Development Kit

Java comes in three sizes:

- Java ME (Micro Edition): Version specifically targeted to the consumer space
- Java SE (Standard Edition): Complete ground-up development environment for the Internet
- Java EE (Enterprise Edition): Everything in Java SE plus an application server and prototyping tools

- Previous releases of Java used the following naming convention: J2ME, J2SE, and J2EE.

Integrated Development Environment

Development

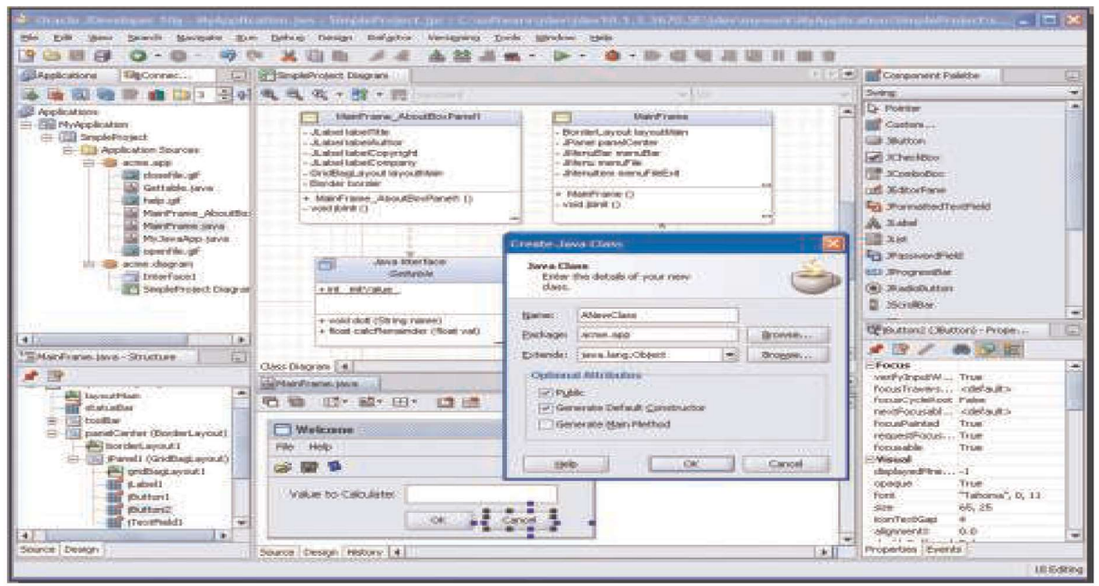
UML

ADF

JSF

EJB

XML



Debug

Exchange

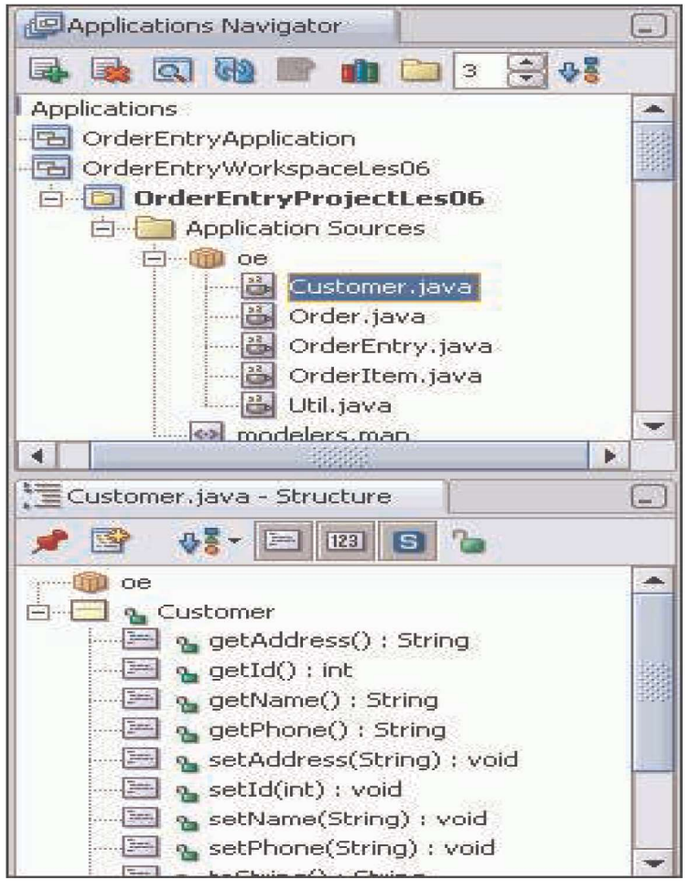
Database

HTML

Source control

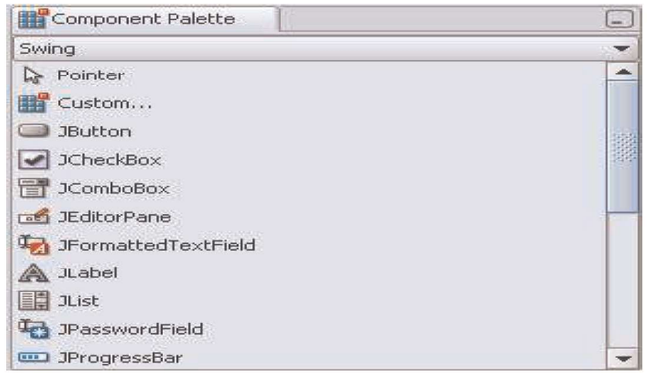
Deployment

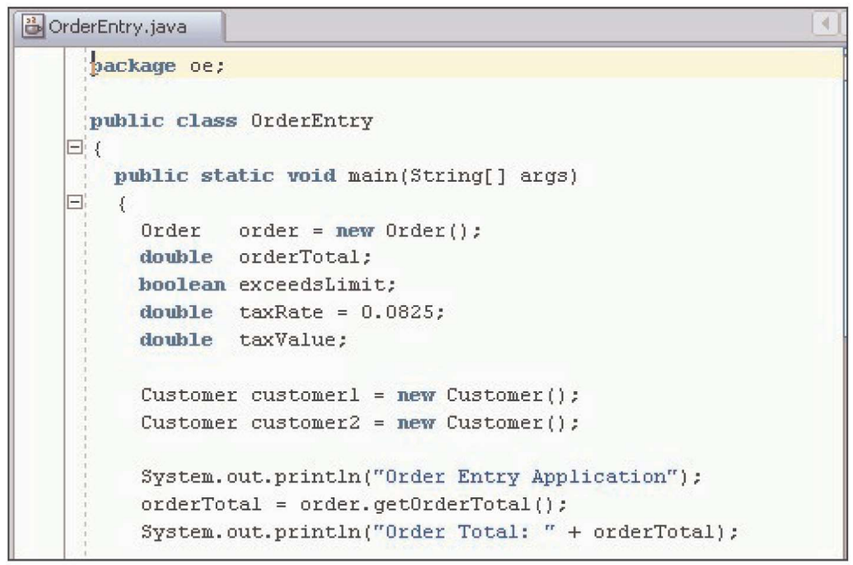
Exploring the JDeveloper Environment



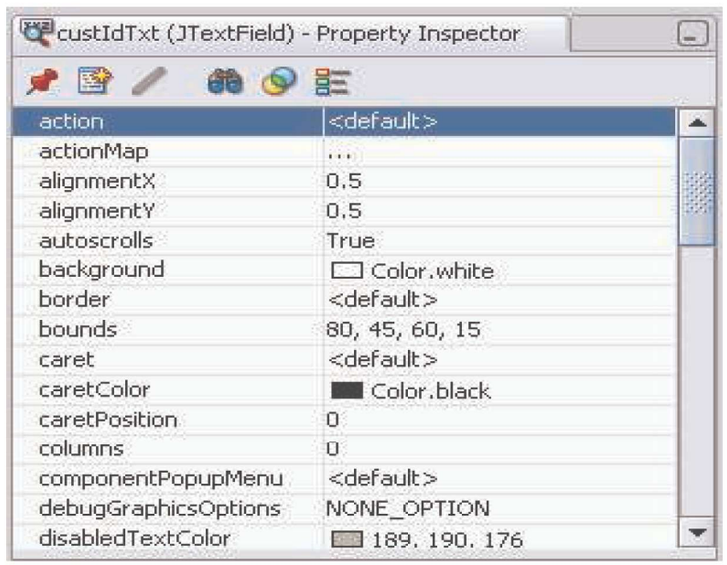
Applications Navigator and Structure window

Component Palette





Code Editor



Property Inspector

Summary

In this lesson, you should have learned how to:

- Java code is compiled into platform-independent bytecodes.
- Bytecodes are interpreted by the JVM.
- Java applications can be stand-alone or implemented across an Internet-computing model.

Practice 1: Overview

This practice covers the following topics:

- There is no practice for this session