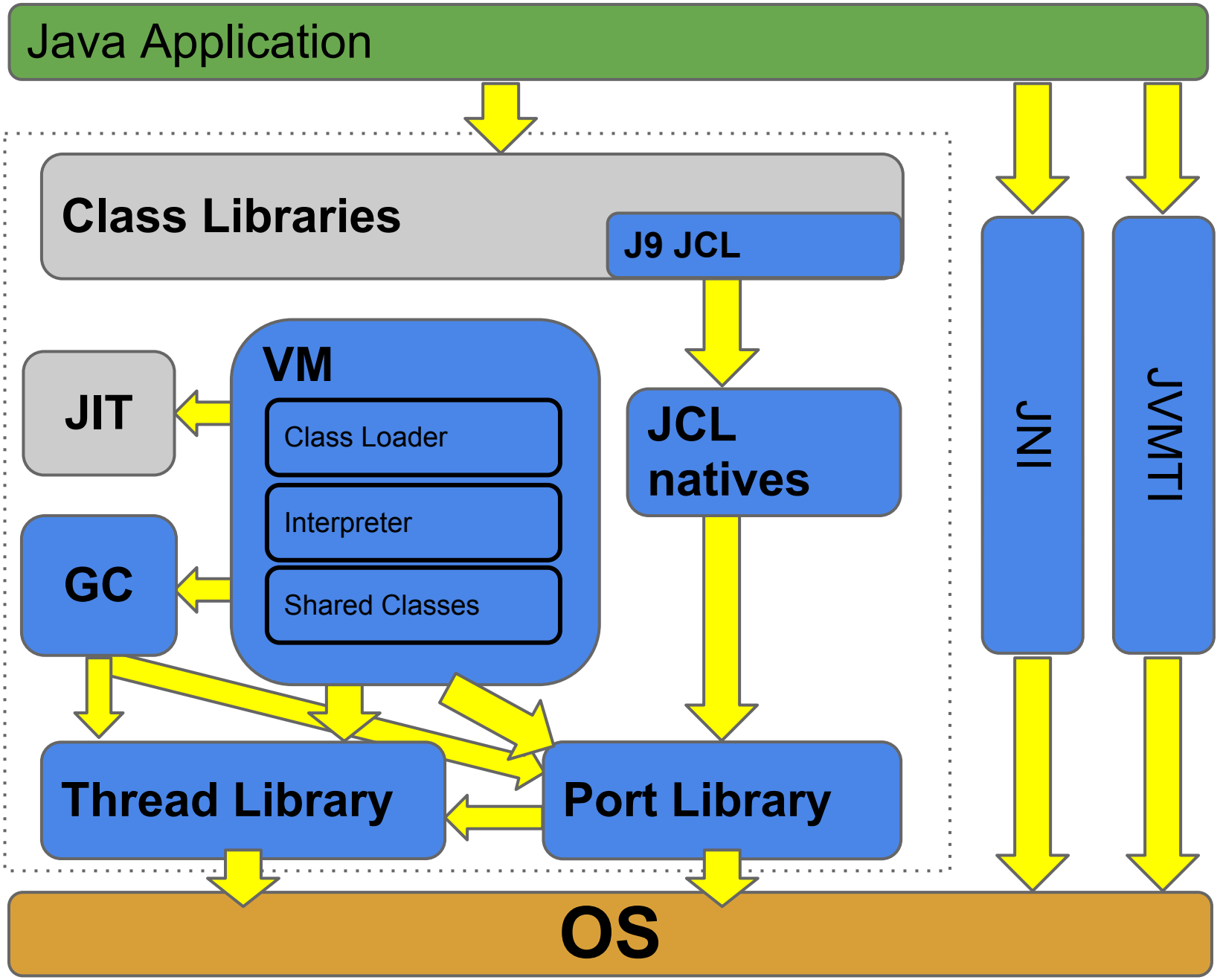
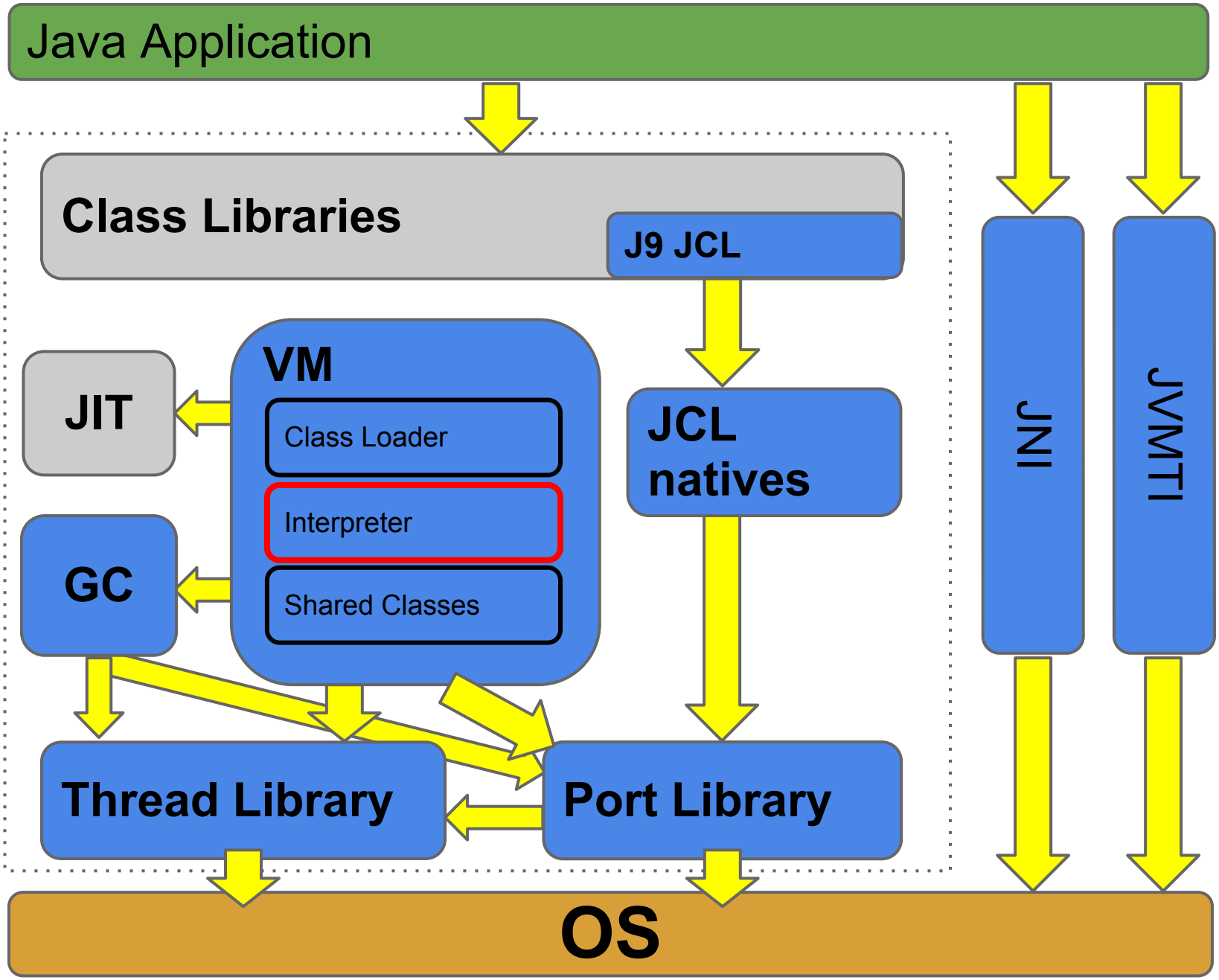
J9 Component Overview


Java Application
Class Libraries
J9 JCL
VM
Class Loader
Interpreter
Shared Classes
JIT
GC
JCL
natives
JNI
JVMTI
Thread Library
Port Library
OS


Java Application
Class Libraries
J9 JCL
VM
Class Loader
Interpreter
Shared Classes
JIT
GC
JCL natives
JNI
JVMTI
Thread Library
Port Library
OS

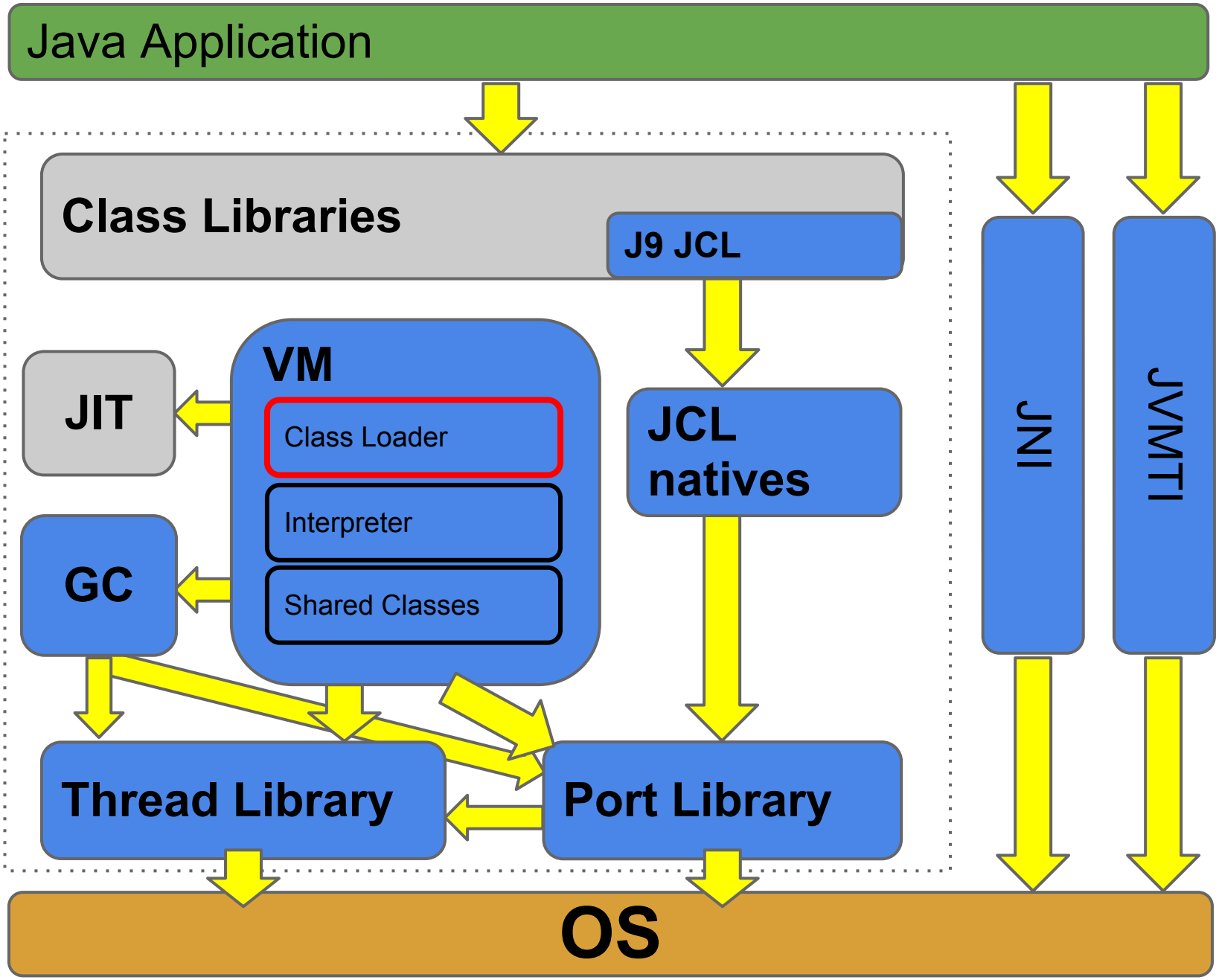
Interpreter

VM_Common/vm/BytecodeInterpreter.cpp

- interprets java bytecode loaded from .class files

```
Compiled from "helloWorld.java"
class helloWorld {
  helloWorld();
    Code:
       0: aload_0
       1: invokespecial #1                  // Method java/lang/Object."<init>":()V
       4: return

  public static void main(java.lang.String[]);
    Code:
       0: getstatic     #2                  // Field java/lang/System.out:Ljava/io/PrintStream;
       3: ldc           #3                  // String hello world
       5: invokevirtual #4                  // Method java/io/PrintStream.println:(Ljava/lang/String;)V
       8: return
}
```


Java Application
Class Libraries
J9 JCL
VM
Class Loader
Interpreter
Shared Classes
JIT
GC
JCL natives
JNI
JVMTI
Thread Library
Port Library
OS

Class Loader

VM_Common/bcutil/cfreader.c

- transforms a bag of bytes (binary .class file) from various sources (filesystem, network, constructed at runtime) to J9ROMClass
- When loading a class, the JVM internally stores the class in two parts:
  The immutable (read only) portion of the class. (ROMClass)
  The mutable (writeable) portion of the class. (RAMClass)

Verifier

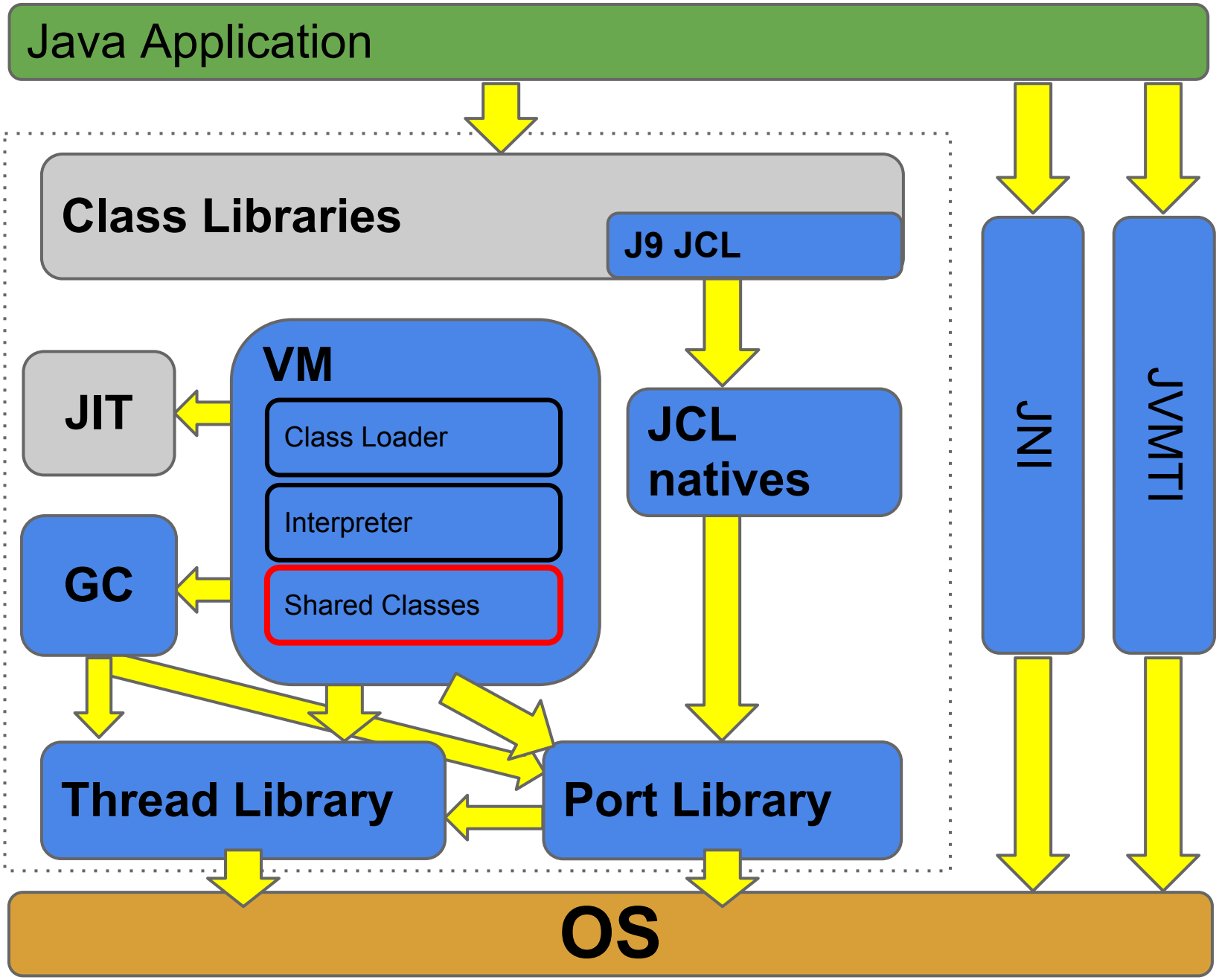
VM_Common/bcverify/{bcverify.c, staticverify.c, rtverify.c}

- confirm that bytecode from .class files cannot perform illegal operations or exploit the virtual machine
- bootclasspath is not verified
- examples
  - .class structure (extra bytes, early termination, etc)
  - 0xCAFEBABE
  - branches always to valid locations
  - references are type-safe

zlib

VM_zlib

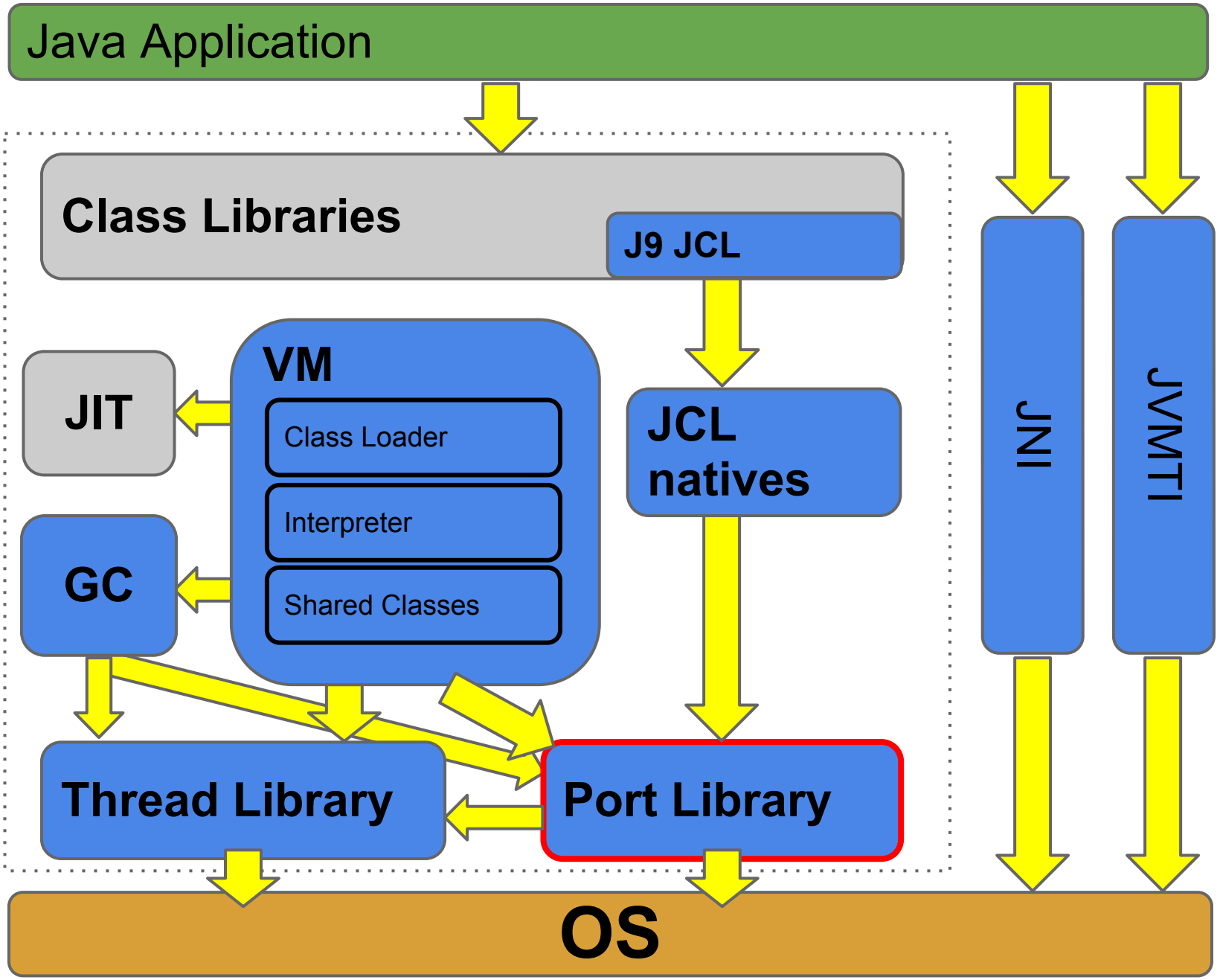
- very popular open source implementation of the DEFLATE compression algorithm
- j9 absorbs zlib version 1.2.3 into the VM_zlib project
- used to inflate jar files (collection of .class files)


Java Application
Class Libraries
J9 JCL
VM
Class Loader
Interpreter
Shared Classes
JIT
GC
JCL
natives
JNI
JVMTI
Thread Library
Port Library
OS

Shared Classes

VM_Shared-Classes

- share immutable parts of loaded classes between JVMs
- classes stored in a memory mapped file or an area of shared memory
- faster to load classes from a populated cache than loading from disk
- amount of physical memory used can be significantly less when using more than one JVM instance
- enabled with -Xshareclasses


Java Application
Class Libraries
J9 JCL
VM
Class Loader
Interpreter
Shared Classes
JIT
GC
JCL
natives
JNI
JVMTI
Thread Library
Port Library
OS

Portability Library

VM_Port-Library

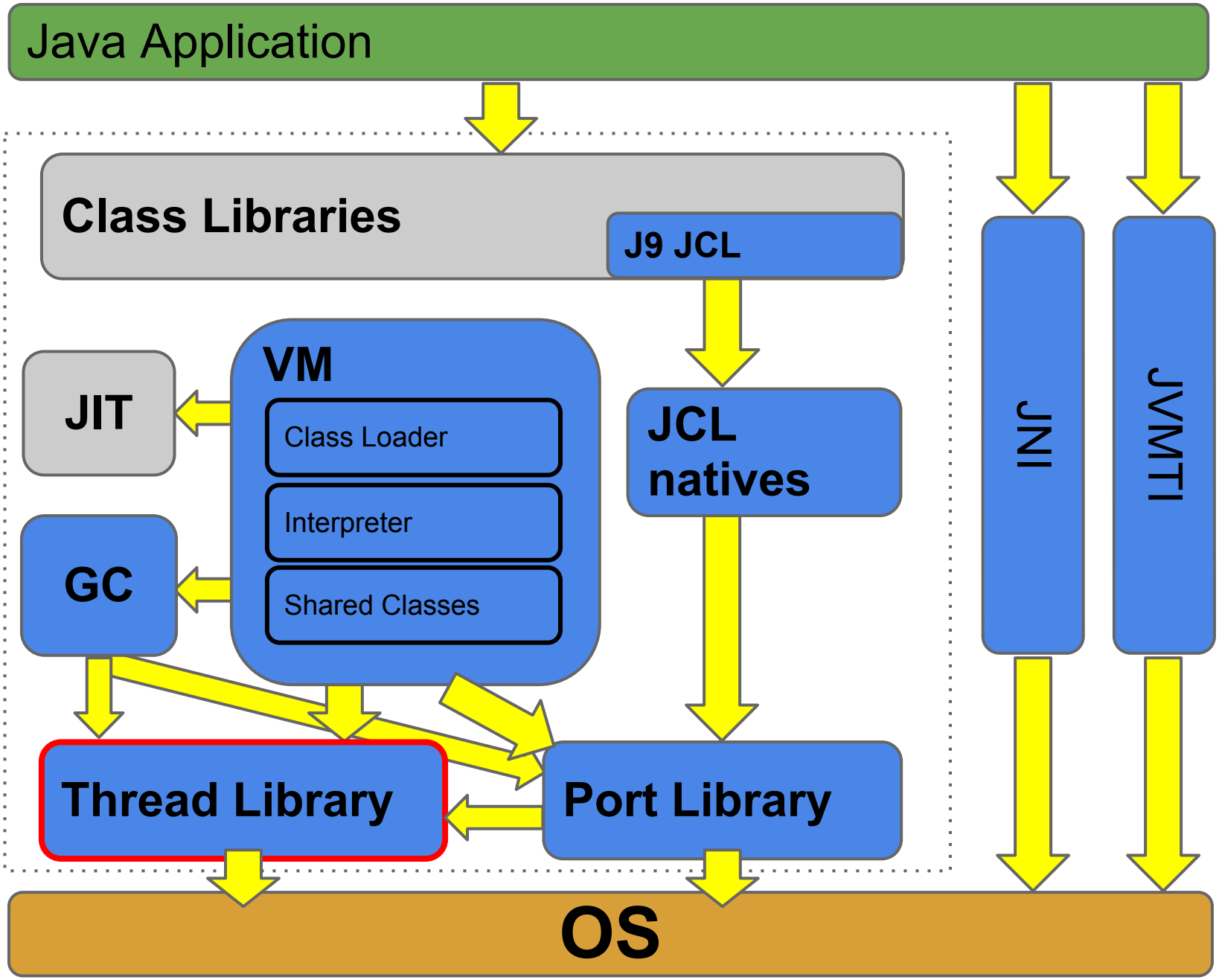
- memory allocation, file io, sockets, etc
- consistent interface to platform-specific implementations

examples

j9mem_allocate_memory (struct J9PortLibrary *portLibrary, UDATA byteAmount, const char *callSite, U_32 category)

j9file_open (struct J9PortLibrary *portLibrary, const char *path, I_32 flags, I_32 mode)

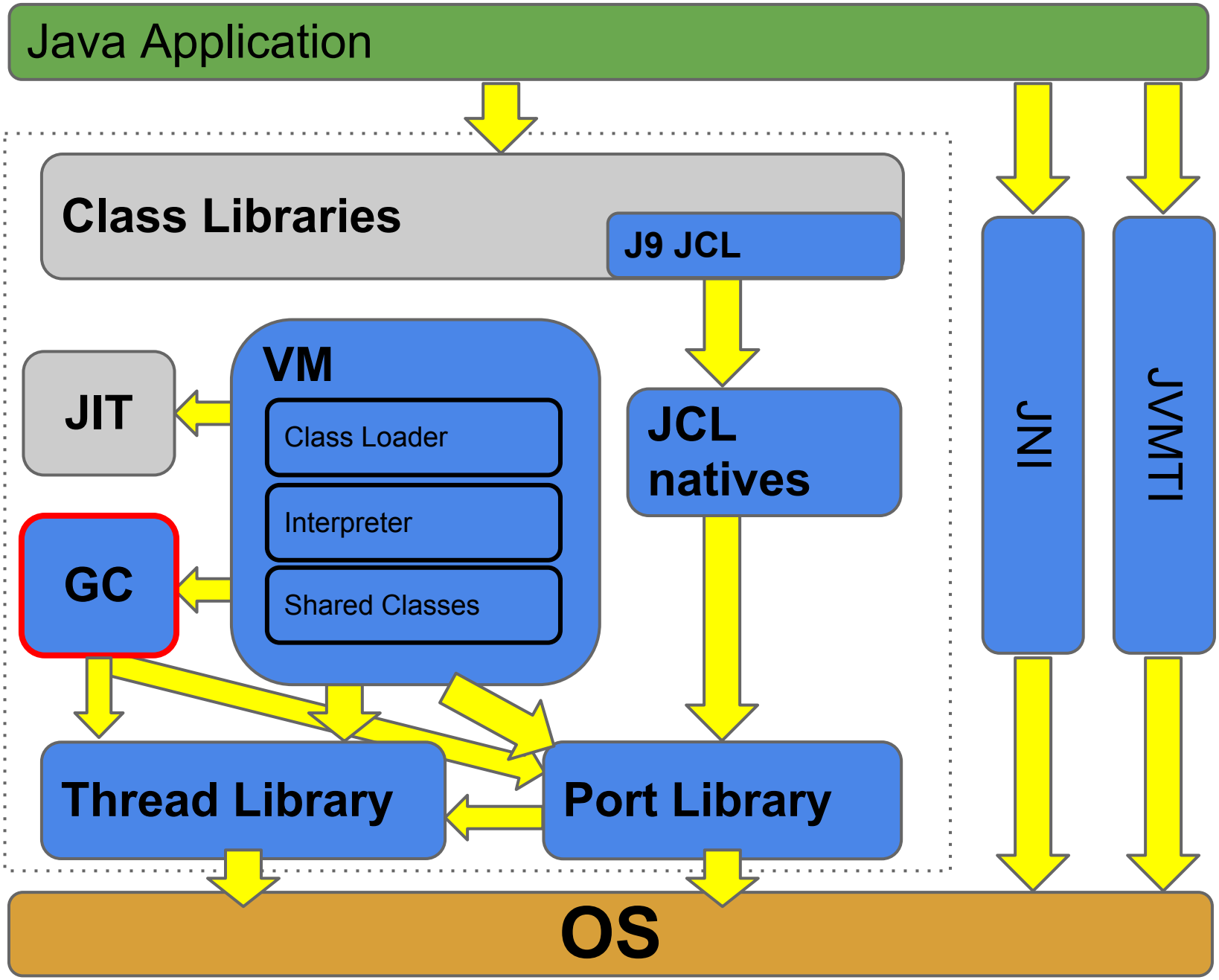
j9tty_err_printf (struct J9PortLibrary *portLibrary, const char *format, ...)


Java Application
Class Libraries
J9 JCL
VM
Class Loader
Interpreter
Shared Classes
JIT
GC
JCL natives
JNI
JVMTI
Thread Library
Port Library
OS

Thread Library

VM_Thread-Library

- a standalone generic threading library (doesn't even require the port library)
- thin abstraction of OS's thread management and thread synchronization
- everything to do with threading in J9 is based on thread library functionality


Java Application
Class Libraries
J9 JCL
VM
Class Loader
Interpreter
Shared Classes
JIT
GC
JCL
natives
JNI
JVMTI
Thread Library
Port Library
OS

Garbage Collector

Modron

- automatic memory management and object allocation
- reclaims objects that are no longer referenced (aka garbage)
- gives the illusion of infinite memory
- different gc policies for different workloads
  - -Xgcpolicy:gencon (generational concurrent)
  - -Xgcpolicy:optthruput
  - -Xgcpolicy:balanced

National Language Support

VM_NLS

- error message i18n

example

j9nls_printf(PORTLIB, J9NLS_ERROR, J9NLS_VM_UNRECOGNISED_CMD_LINE_OPT, optString);

java -lasjff
Command-line option unrecognised: -lasjff
Nicht erkannte Befehlszeilenoption: -lasjff
Nierozpoznana opcja wiersza komend: -lasjff
Opzione riga comandi non riconosciuta: -lasjff

RAS: Reliability, availability and serviceability

Trace Engine (-Xtrace)

- optimized way of capturing vm and class library runtime diagnostics
- thousands of tracepoints throughout vm and class libraries
- in case of unhandled exception, trace buffers are dumped to a binary trace file (Snap*.trc)

RAS: Reliability, availability and serviceability

example

VM_Common/vm/jvminit.c

Trc_VM_VMInitStages_Event1(vm->mainThread);

VM_Common/vm/j9vm.tdf

TraceEvent=Trc_VM_VMInitStages_Event1 Overhead=1 Level=1 Template="
Trace engine initialized for module j9vm"

```
java -Xtrace:print=j9vm helloWorld

03:39:02.633*0xf6c41e00          j9vm.0       - Trace engine initialized for module j9vm
03:39:02.633 0xf6c41e00          j9vm.185     - J9JavaVM located at F6C0A1A0, internalVMFunctions at F6BFB8C0,
portLibrary at F6D6F0E0, j9ras at 00010000
03:39:02.633 0xf6c41e00          j9vm.445     - Thread yield algorithm information: sched_compat_yield= , yieldAlgorithm=0,
yieldUsleepMultiplier=1.
03:39:02.633 0xf6c41e00          j9vm.427     - String interning is enabled.
03:39:02.633 0xf6c41e00          j9vm.463     - VM classloader locking enabled
03:39:02.634 0xf6c41e00          j9vm.47      - hook registration (register/unregister=1 event=72 function=F58F58ED
userData=F6CEEEE8)
03:39:02.634 0xf6c41e00          j9vm.47      - hook registration (register/unregister=1 event=19 function=F58F57FB
userData=F6CEEEE8)
03:39:02.634 0xf6c41e00          j9vm.47      - hook registration (register/unregister=1 event=41 function=F58F55A3
```

RAS: Reliability, availability and serviceability

Trace Engine (-Xtrace)

- can be triggered by method entry, group (j9vm, j9prt, etc), or tracepoint number
- triggers can perform actions (coredump, javadump, etc)

RAS: Reliability, availability and serviceability

Dump Engine (-Xdump)

Add/remove dump agents for various JVM events.

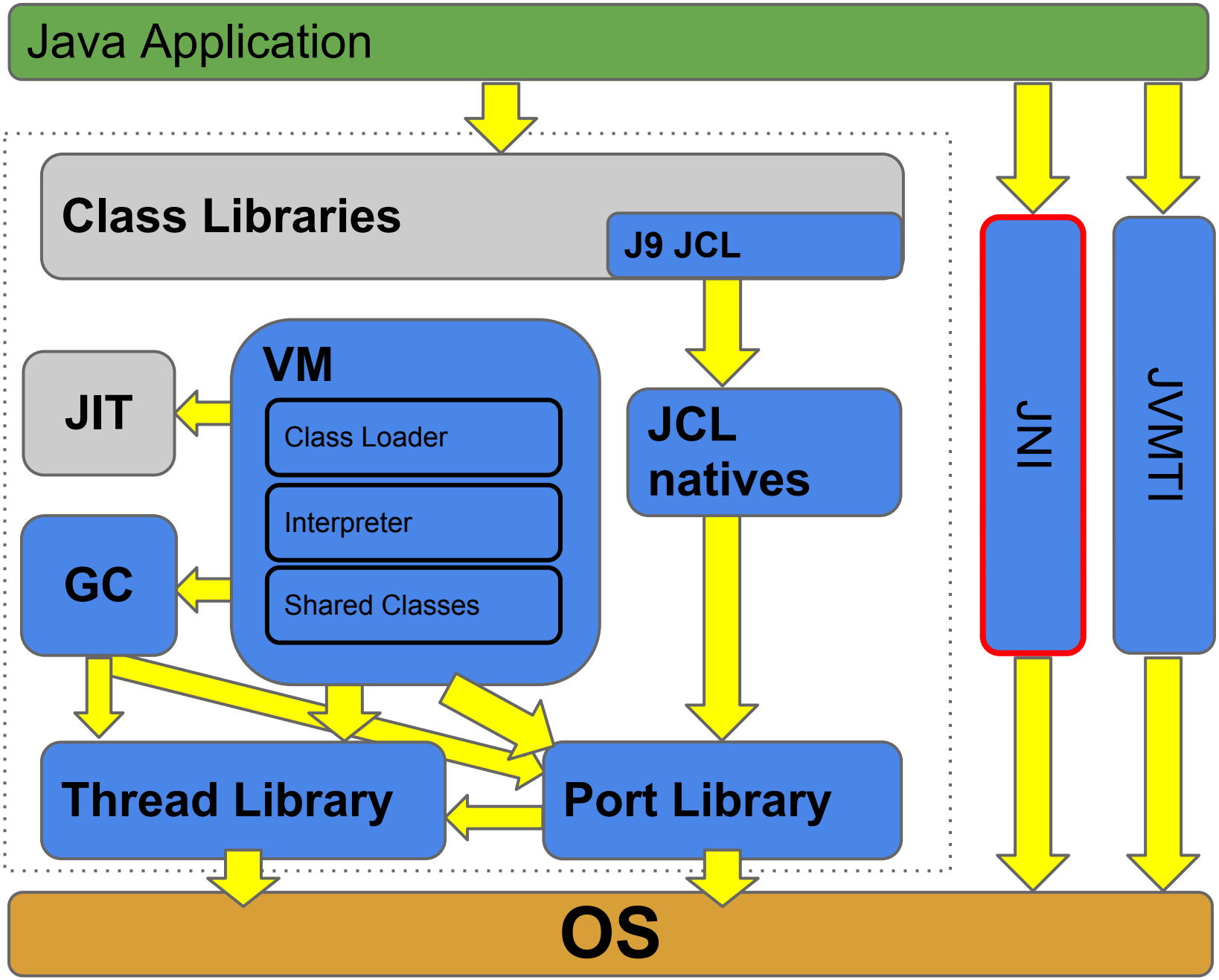
Dump Agents

java, heap, snap, etc

Dump Events

vmstart, vmstop, load, unload, fullgc, any

```
java -Xdump:heap+java:events=vmstart+vmstop
```

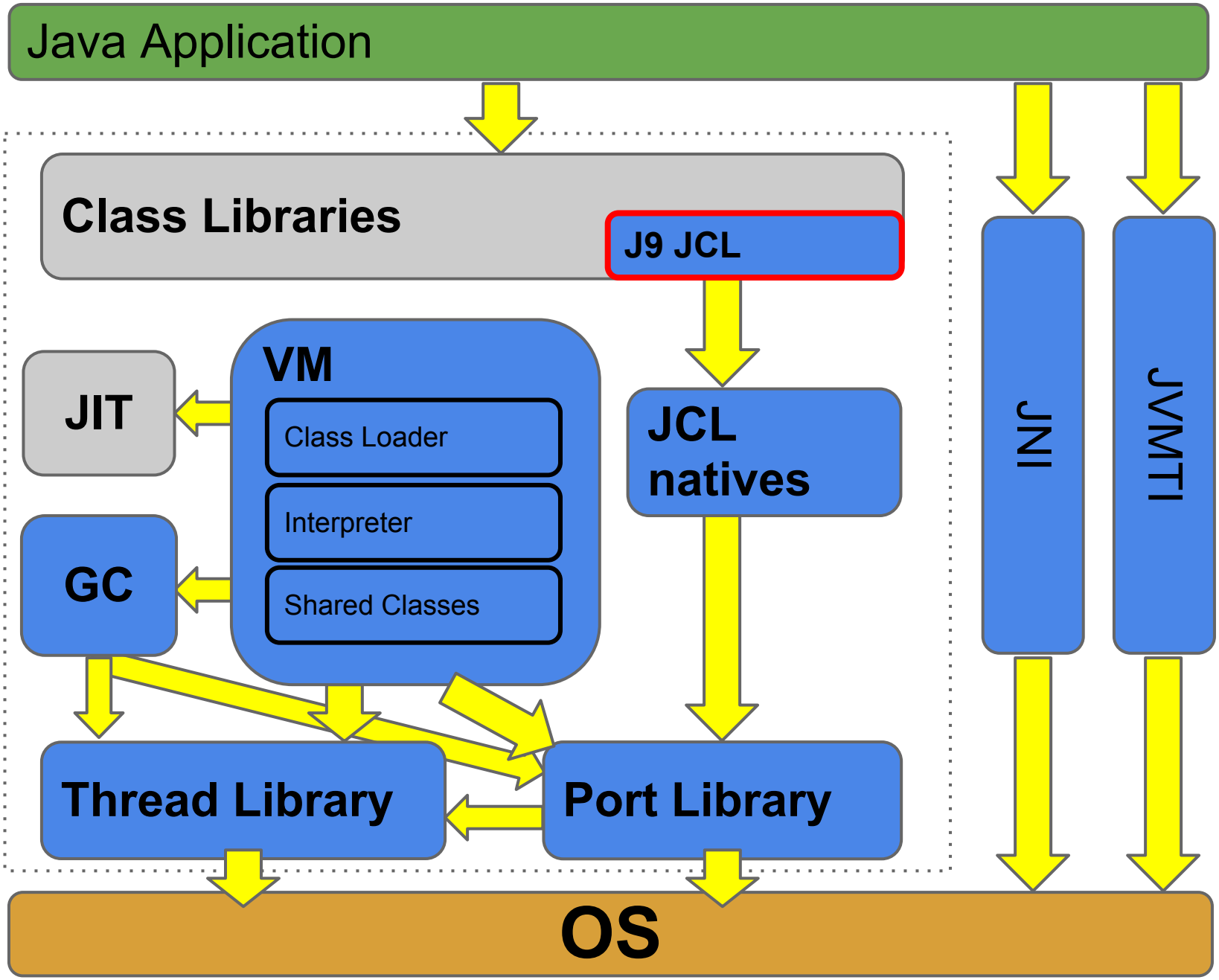

Java Application
Class Libraries
J9 JCL
VM
Class Loader
Interpreter
Shared Classes
JIT
GC
JCL
natives
Thread Library
Port Library
JNI
JVMTI
OS

JNI - Java Native Interface

- allows interaction between native code (C, C++, etc) and Java
- establishes a well-defined and platform-independent interface between native and java code

native methods

- declared in java code
- native method implemented as external entry point in a library
- when the JVM is in the native method JNI provides a way to call back to the JVM


Java Application
Class Libraries
J9 JCL
VM
Class Loader
Interpreter
Shared Classes
JIT
GC
JCL natives
JNI
JVMTI
Thread Library
Port Library
OS

J9 JCL - kernel class libraries

J9 Owns (jre/lib/vm.jar)

- internal java.lang.Annotation support (but not the API classes themselves)
- shared classes
- the bootstrap classloader
- java.lang.ref
- classes in java.lang:
  - Class
  - ClassLoader
  - Compiler
  - Object
  - String, StringBuffer, StringBuilder
  - System
  - Thread, ThreadGroup
  - Throwable, StackTraceElement
- java.lang.reflect.Arrays
- java.math.BigDecimal
- java.security.AccessControllContext, AccessController
- sun.reflect.DelegatingClassLoader

Oracle provides the rest (rt.jar)

DDR - Direct Dump Reader

- a diagnostic component that reads J9 state from core files or running processes
- provides an interface for diagnostic tooling

DDR Components

- C/C++ code that produces the **structure blob** - a binary file that describes the shape of J9 structures (name, type and offset of every field + constants)
- a small amount of VM runtime code that loads the structure blob into memory (so that it will be written into any core files produced by the process)
- a large amount of Java code that reads the structure blob from a core file (or potentially a running process), builds a model of the J9 VM being run, and can extract its state

DDR works by embedding the structure blob in the process image, such that the DDR Java library can read a description of the structures used in the process and walk them with algorithms written in Java.

Builder

- Portable assembly language written in Smalltalk
- old interpreter, interpreter to jit transitions
  - i2j transition (See J9VMJitSendTargets)
  - j2i transition (See J9VMJITCallToInerpreterTargets & J9VMJITCallToInterpreterRetunBytecodes)
- found in 8096_*
- originally entire vm was written in builder

What's "8096"?

old buffer bug that took some time to find

someone thought **8*1024 = 8096**

In honour of 8096 bug, wanted to call it 8K

"8K" not a valid name for a Smalltalk class

"K8" is though

J9 Name

Next version of K8 to be smaller and better

K9 doesn't work that well.... arf!

J is less (smaller) than K, 9 is better than 8

java -Xint helloWorld