The Squawk Java Virtual Machine

Doug Simon, Cristina Cifuentes

Interns: Andrew Crouch, David Liu

Sun Labs

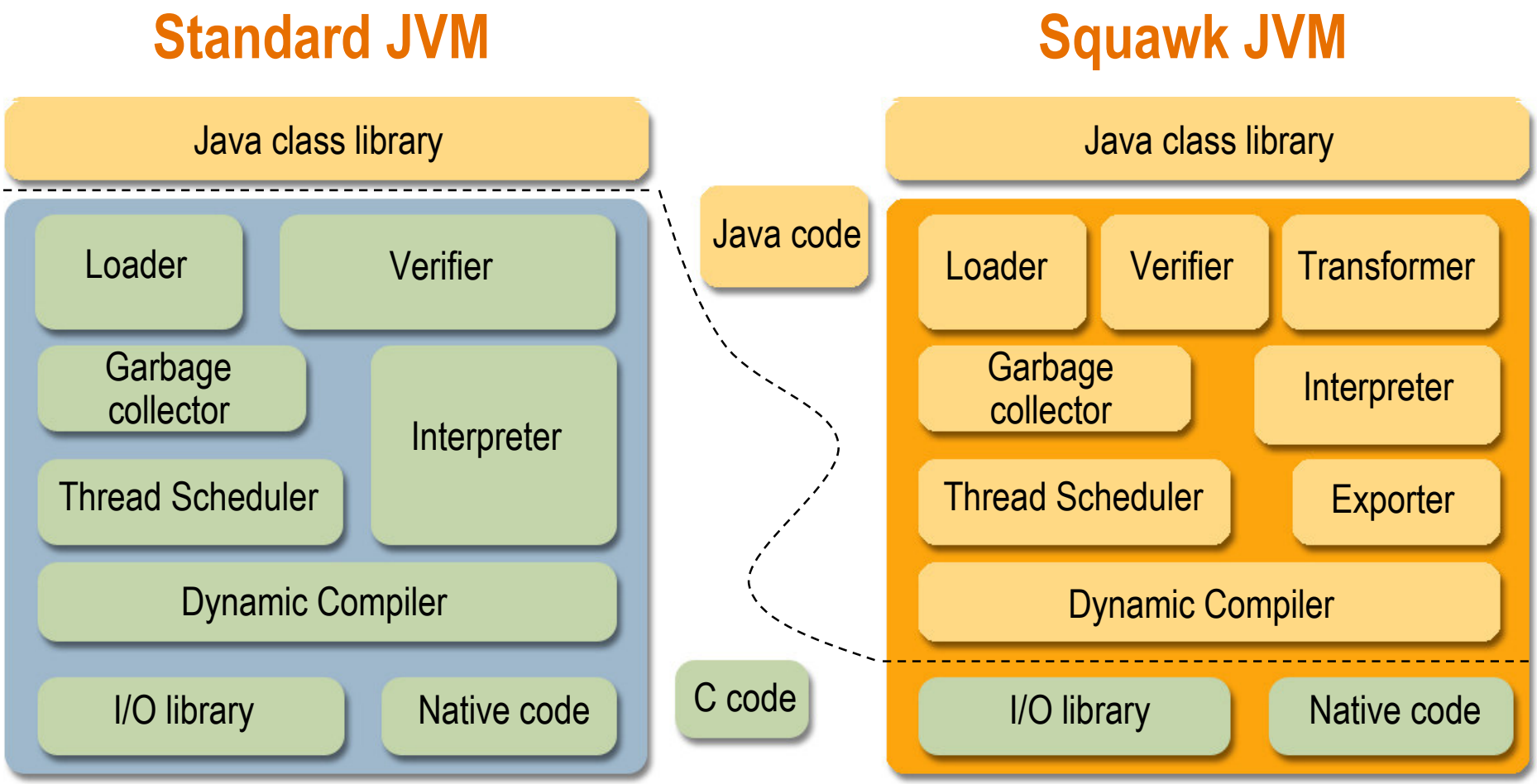
The Squawk JVM

Squawk is a small, execute-in-place, CLDC 1.0 Java VM written in Java, designed for simplicity and ease of portability.

The JVM implements an isolate mechanism and can run on the bare metal, i.e., without need for an underlying OS.

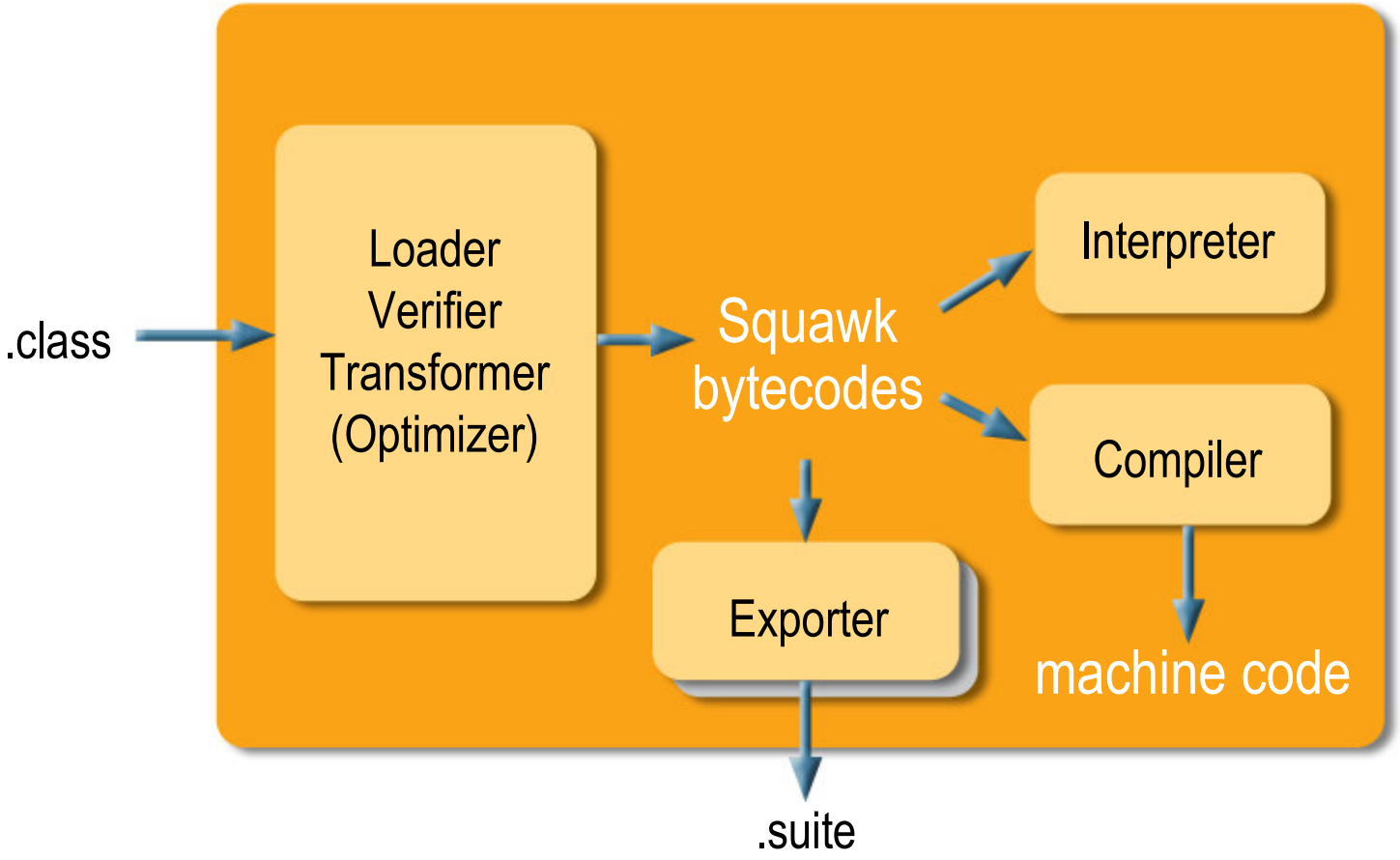
Squawk runs on ARM7, PPC/MacOSX, (x86,PPC)/Linux, SPARC/Solaris, and x86/Windows.

Standard JVM vs Squawk JVM

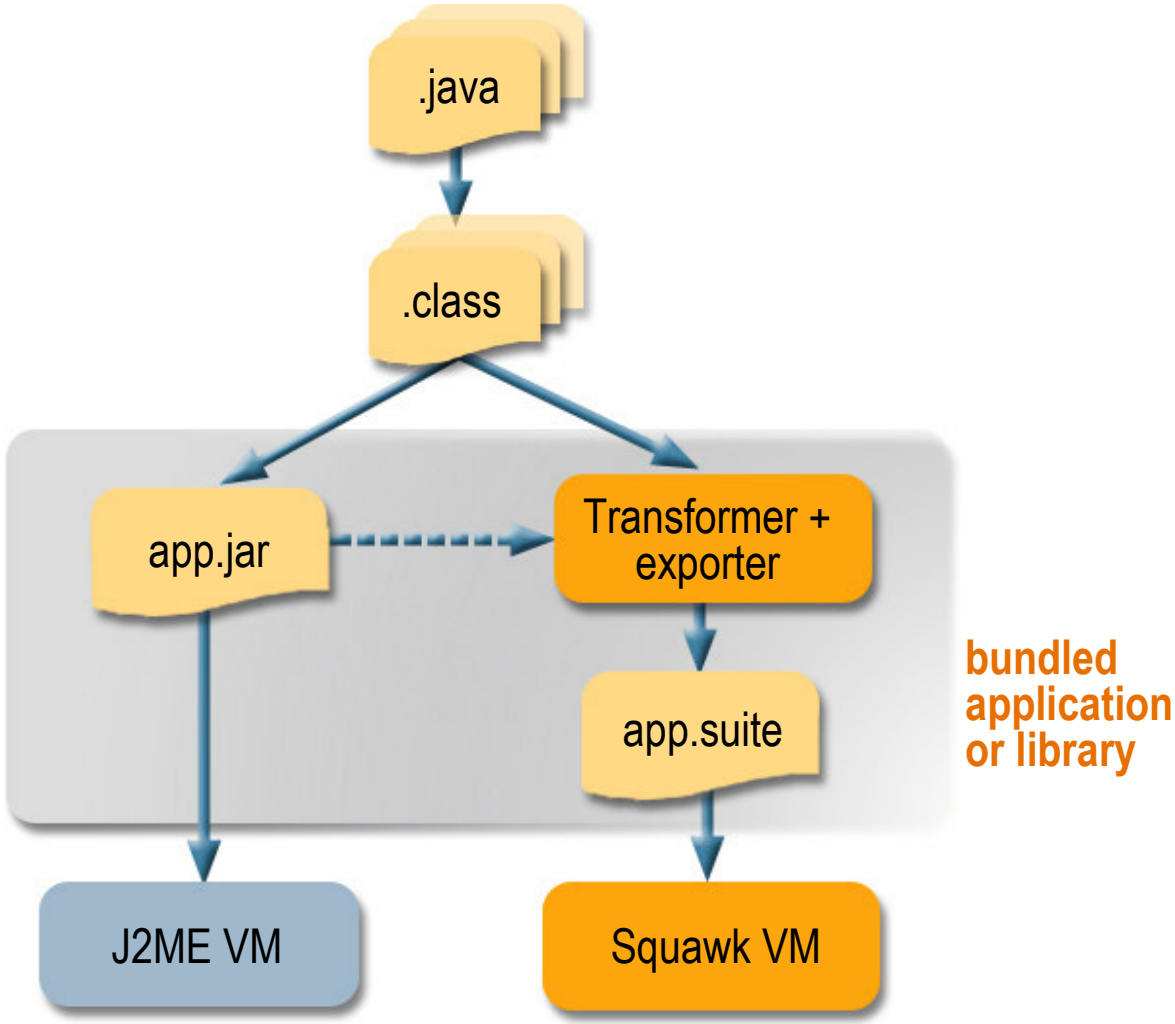


Note: There is a lighter blue behind this one, depending on How it prints

The Squawk Architecture



The Execute-in-Place File Format: Suites



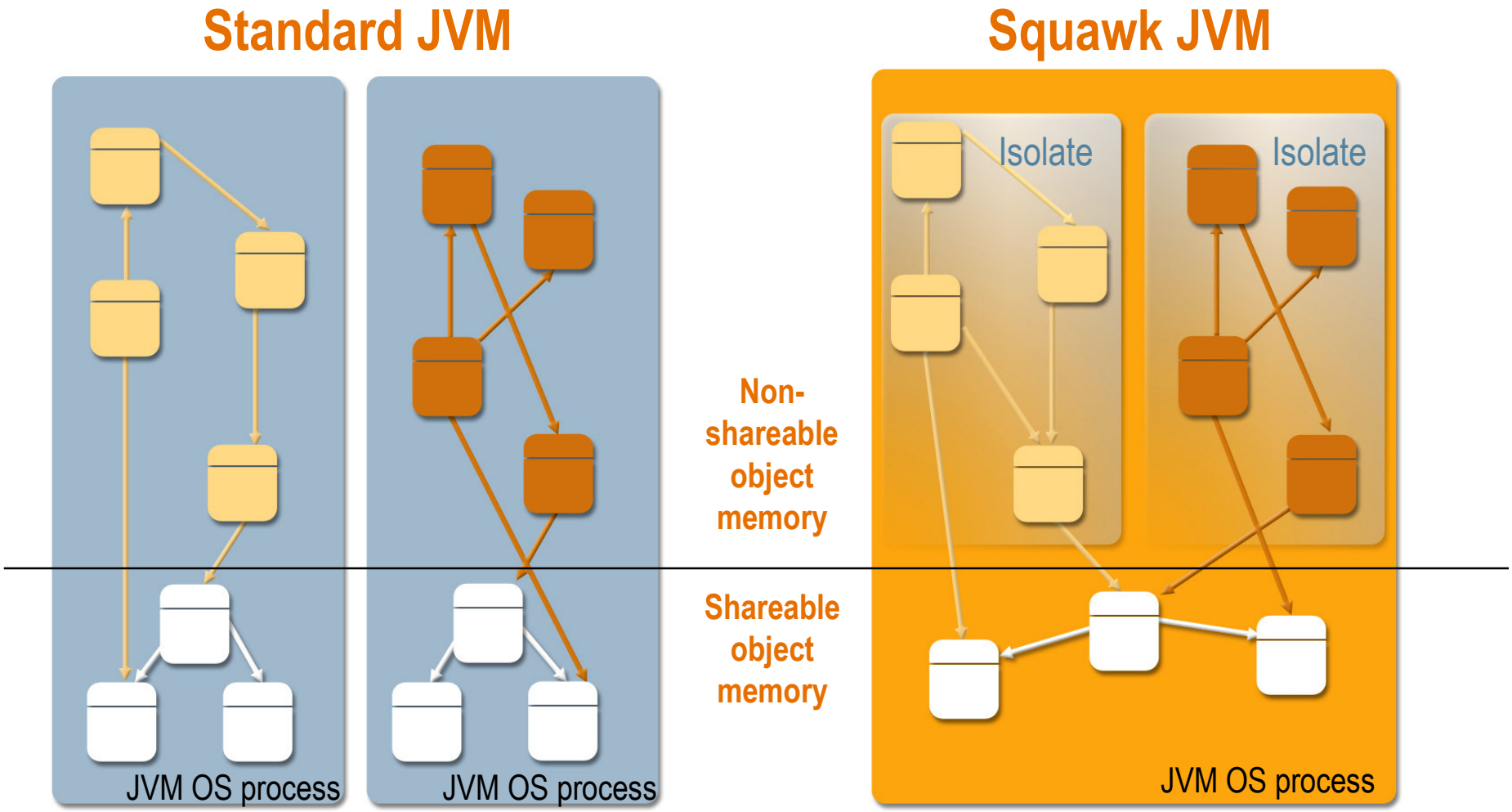
Uncompressed JAR vs Suite File Size Comparison

Application	JAR	Suite	Suite/JAR
CLDC	458,291	149,542	0.33
cubes	38,904	16,687	0.42
hanoi	1,805	835	0.46
delta blue	30,623	8,144	0.27
mpeg	100,917	54,888	0.54
manyballs	12,017	6,100	0.51
pong	17,993	7,567	0.42
spaceinvaders	50,854	25,953	0.51
tilepuzzle	18.516	7,438	0.40
wormgame	23,985	9,131	0.38
Total	753,905	286,285	0.38

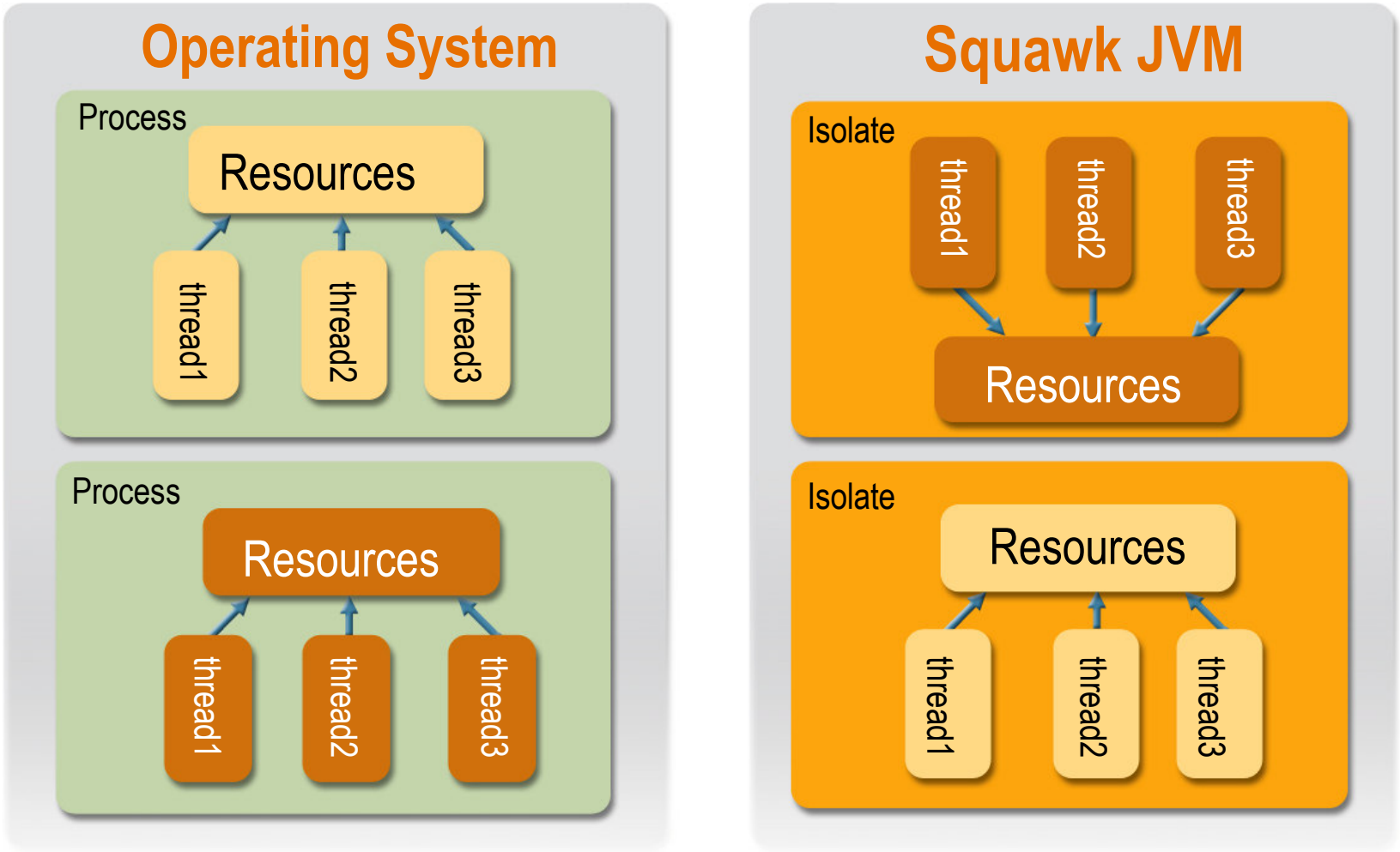
Squawk Bytecodes vs. Java Bytecodes

Squawk Bytecode Property	Benefit
Commonly used bytecodes are 2 bytes instead of 3 bytes	↑ More compact
References to fields and methods resolve into physical offsets	↑ More efficient for interpretation
Local variables are typed	↑ More efficient for compilation
One OOP map per method, nothing on the operand stack at GC points	↑ Simplifies garbage collection
	Eliminates need for static interpretation to decipher activation frames

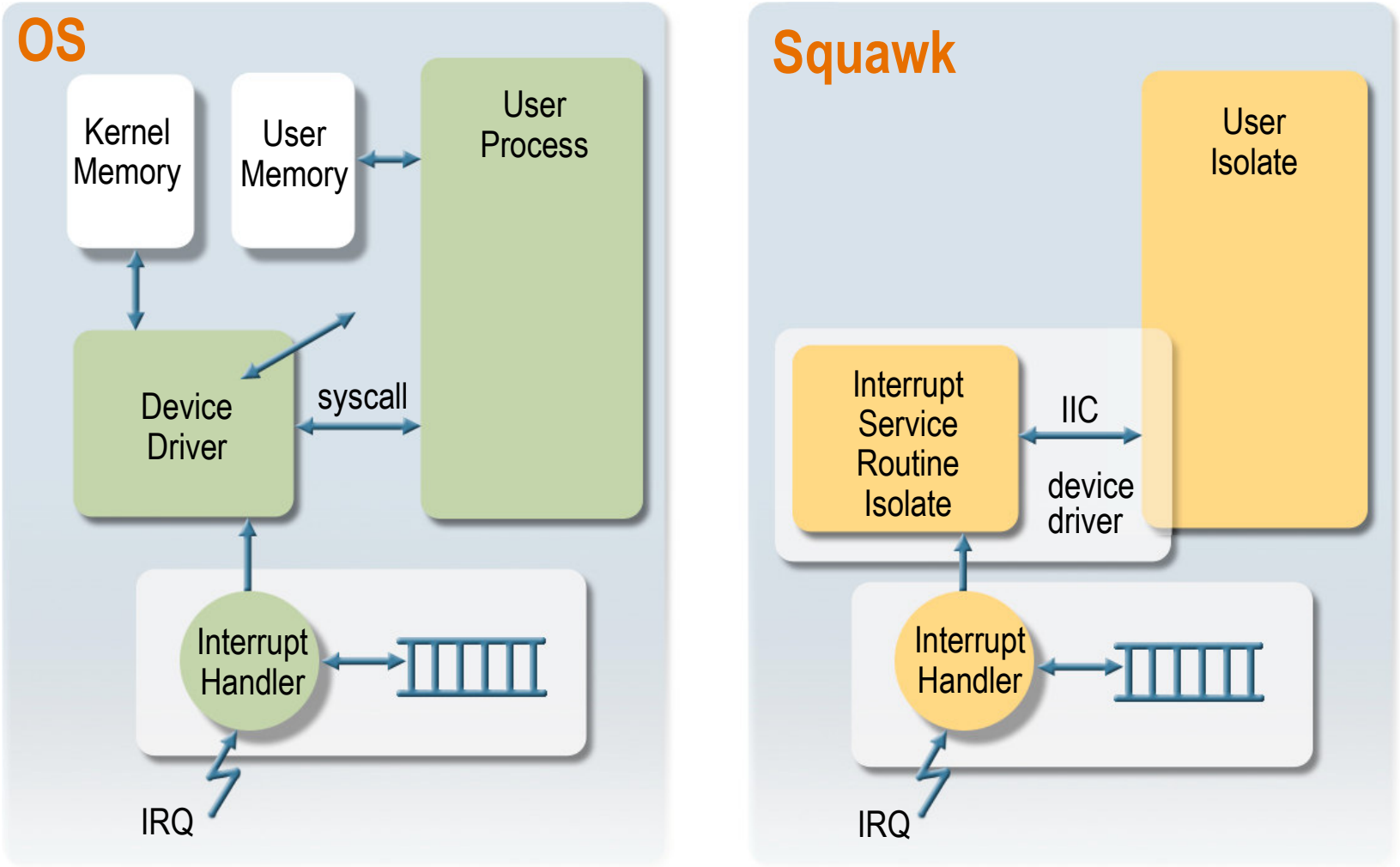
The Isolate Mechanism: Multiple Applications in the One VM



JVM Isolates and OS Processes Analogy



The Interrupt Architecture: Device Drivers in Java



Doug Simon, Cristina Cifuentes

doug.simon@sun.com, cristina.cifuentes@sun.com