Java for Small Devices: The Squawk Java Virtual Machine

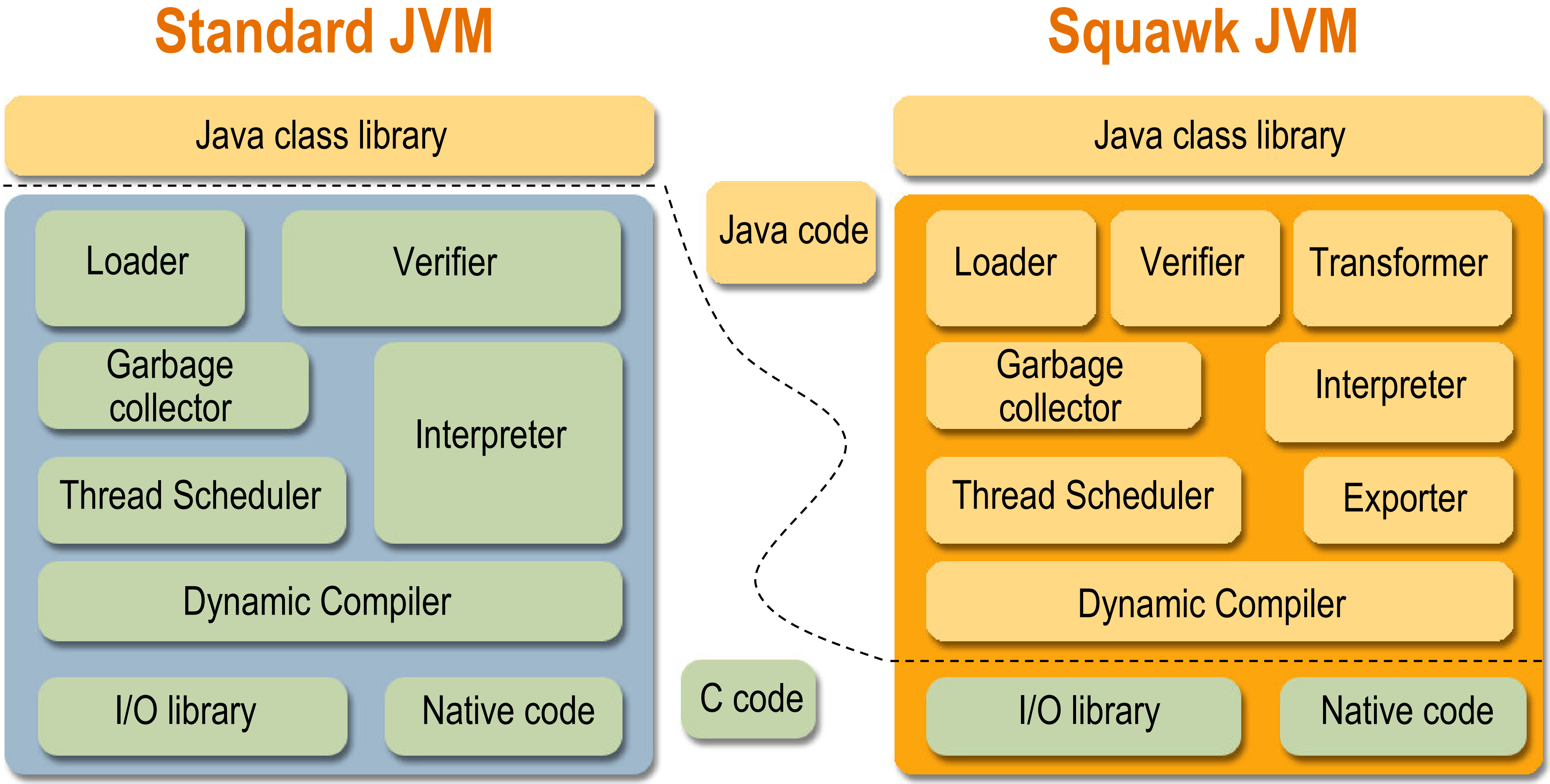
Cristina Cifuentes

Sun Labs

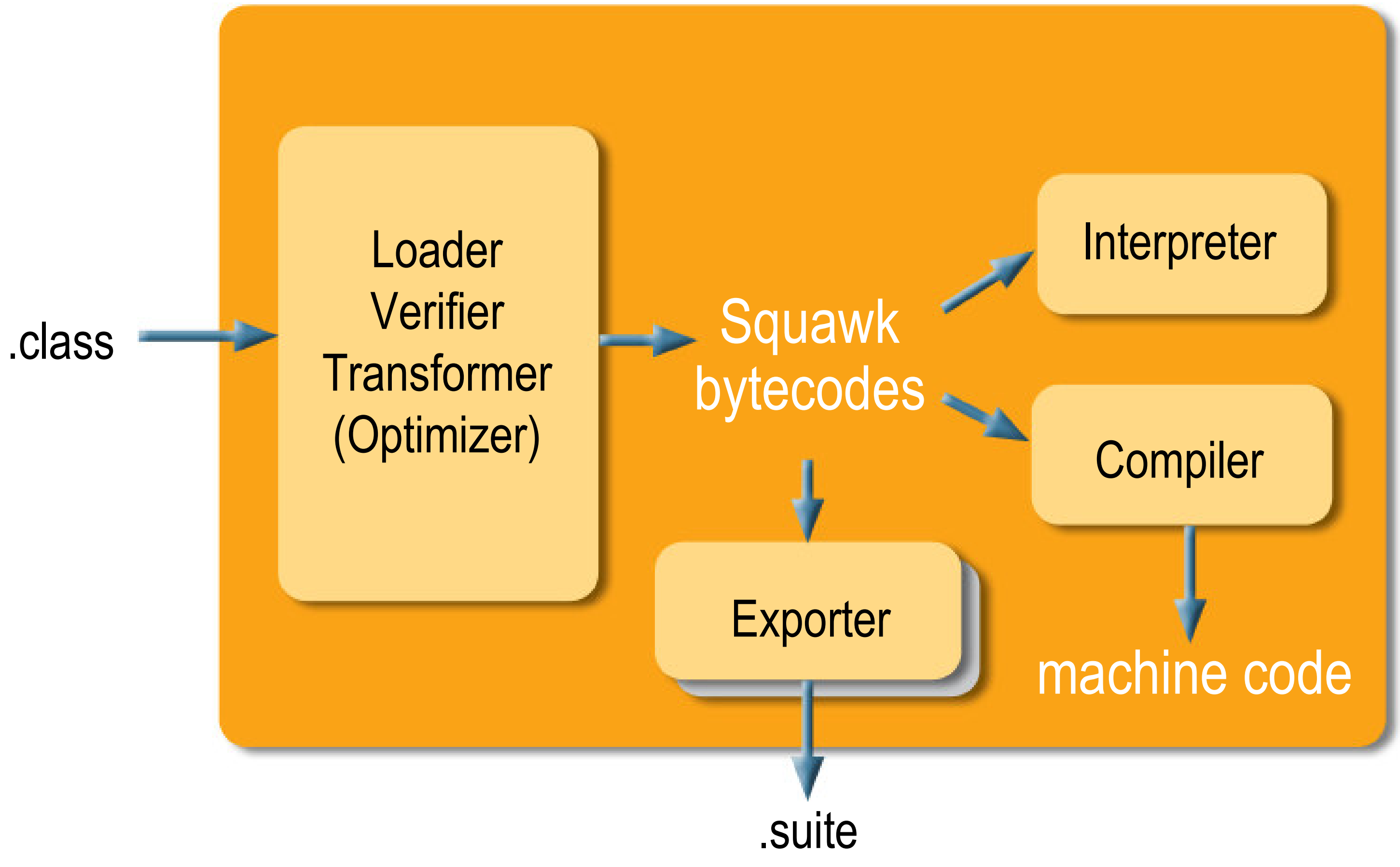
The Squawk JVM:
J2ME + OS Functionality

- Written in Java
- Programmable, i.e., extensible, flexible
- Runs without an OS on ARM
- Runs also on Solaris, Linux, MacOS, Windows
- Optimized for small devices
- AppServer model (isolates)
- Connected: network/wireless/sensors/actuators
- Ports easily

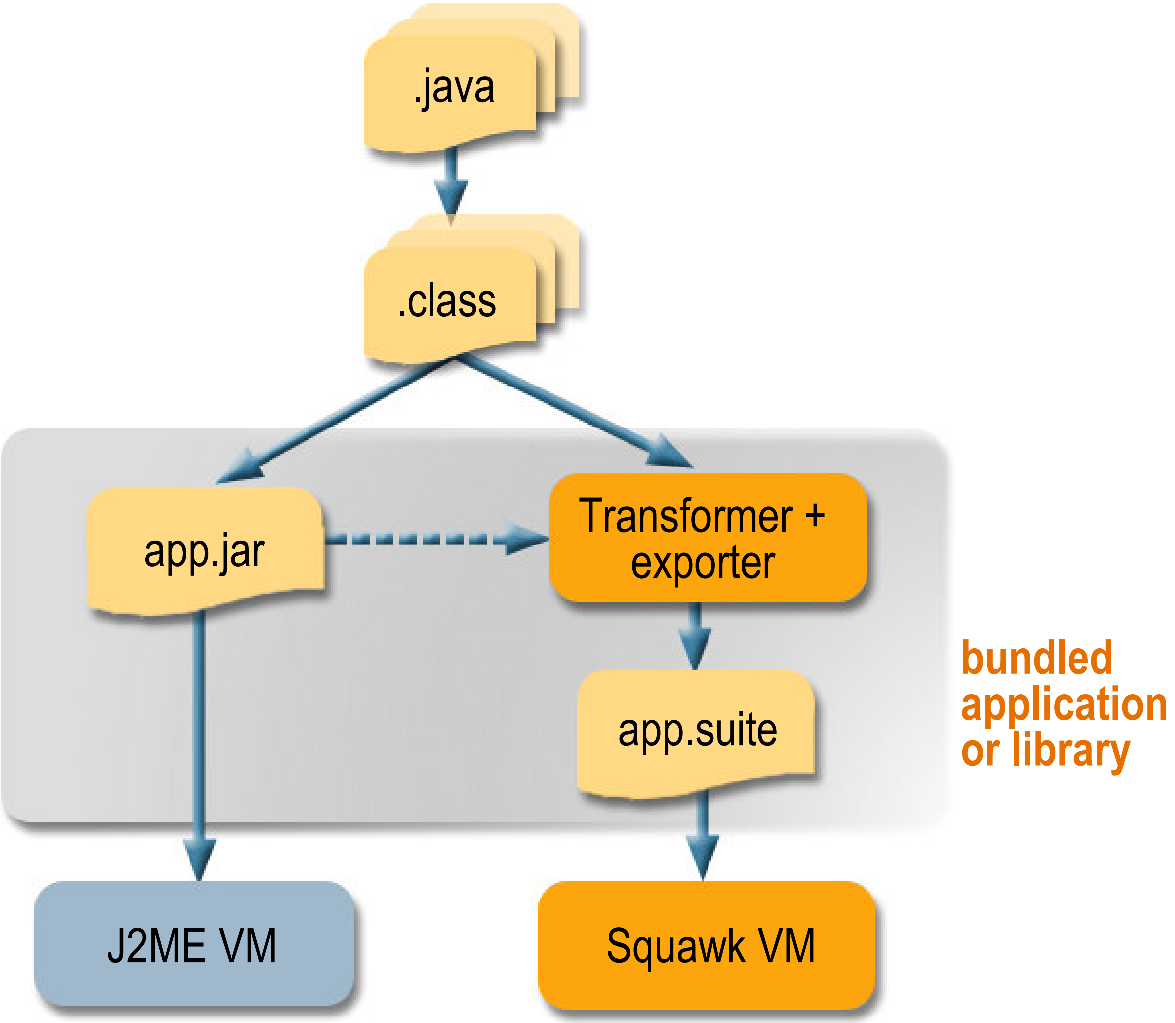
Standard JVM vs Squawk JVM



The Squawk Architecture



The Execute-in-Place File Format: Suites



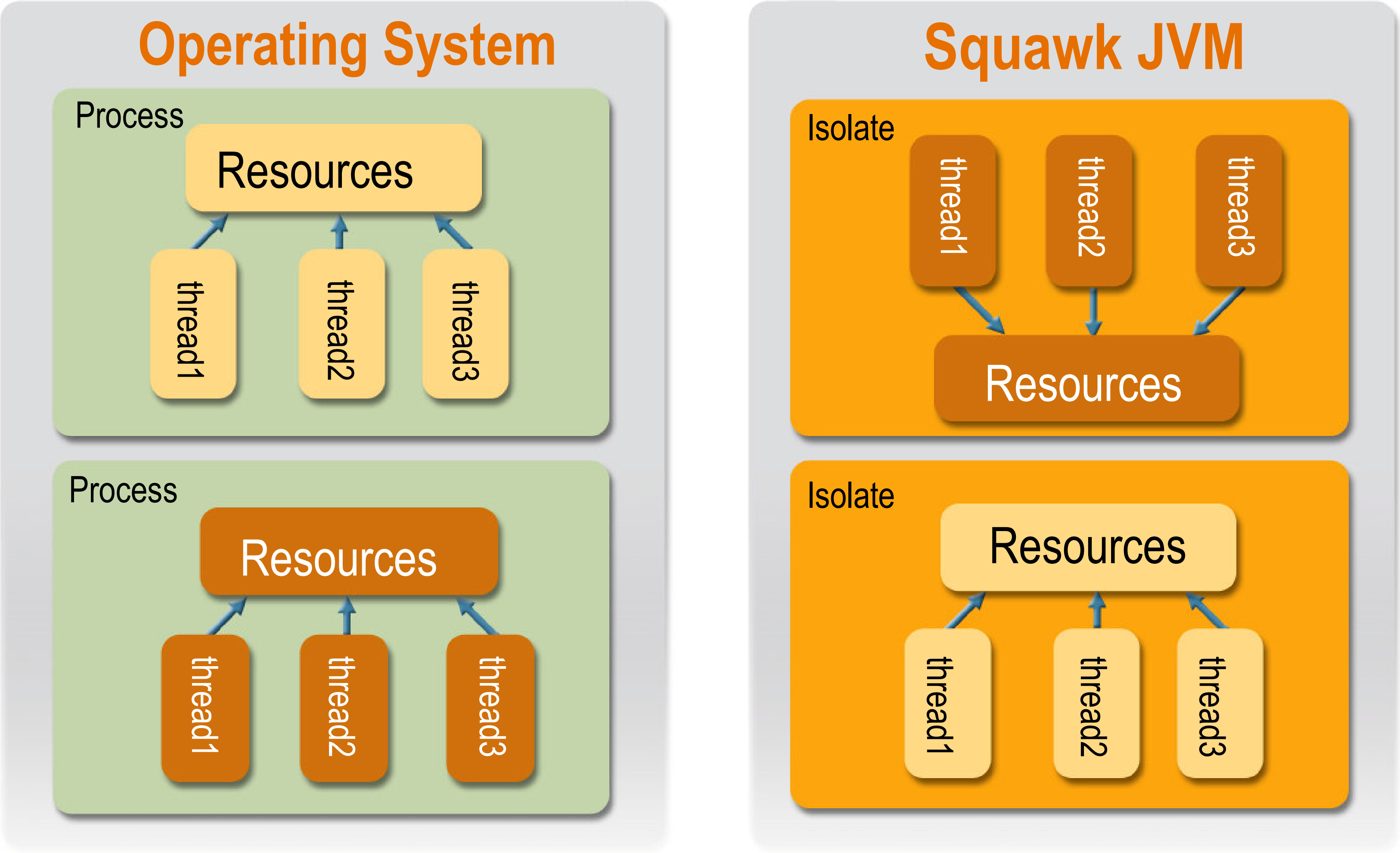
Uncompressed JAR vs Suite File Size Comparison

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**

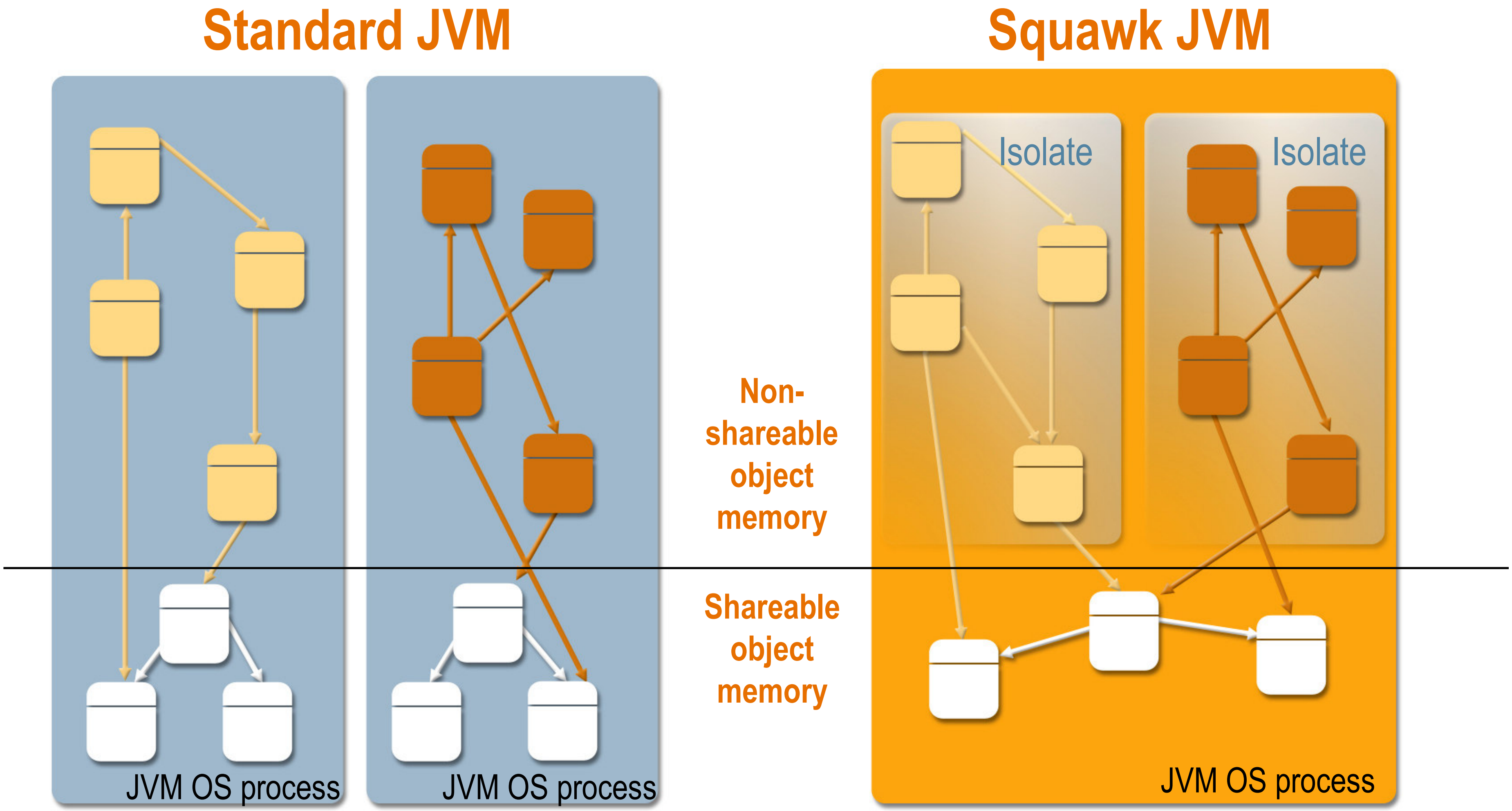
Squawk Bytecodes vs. Java Bytecodes

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

JVM Isolates and OS Processes Analogy



AppServer Model Allows Multiple Applications Within One VM (JSR121)



Standard JVM

Squawk JVM

Isolate

Isolate

Non-shareable object memory

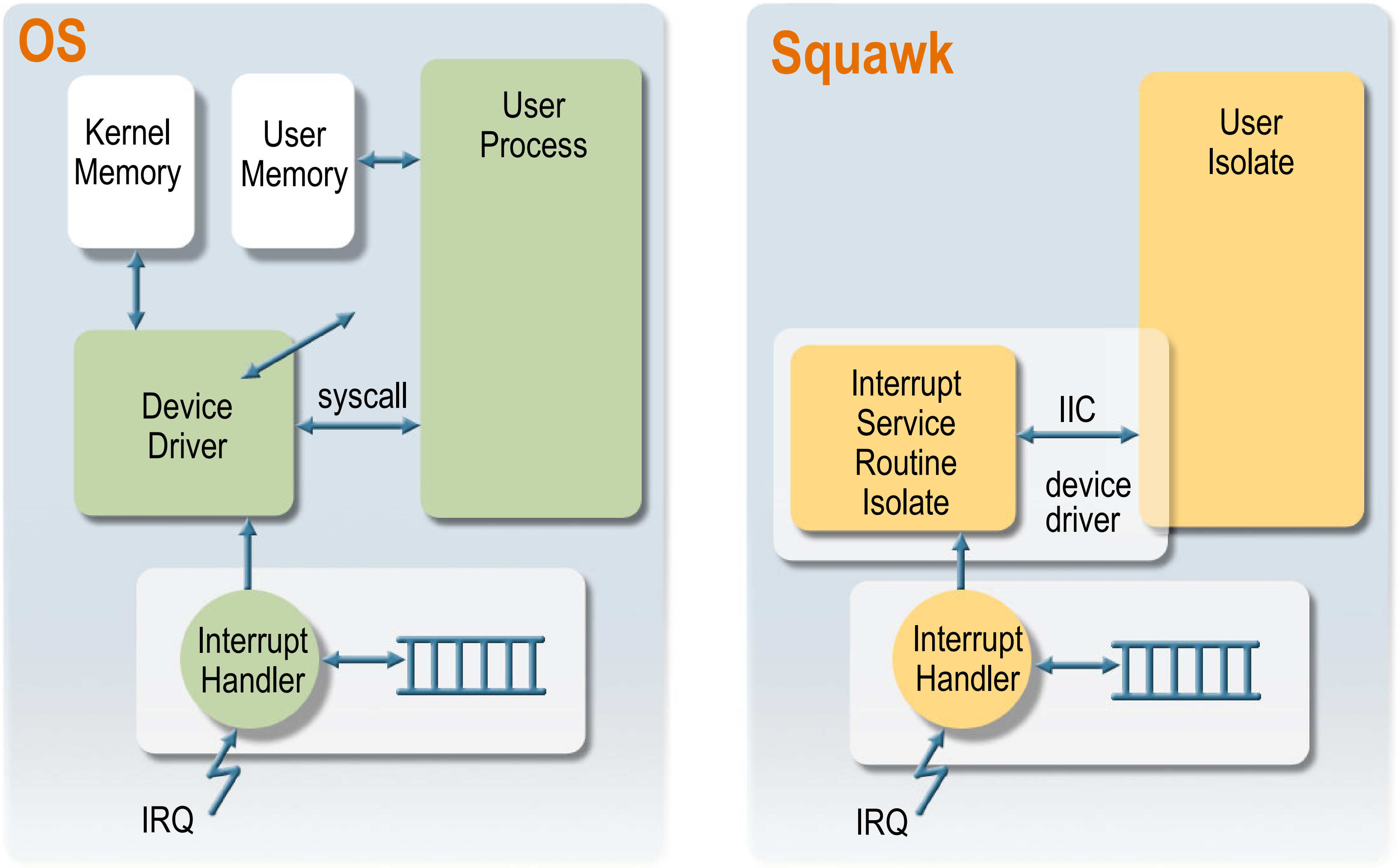
Shareable object memory

JVM OS process

JVM OS process

JVM OS process

Low-Level Hardware Support from Java (“Device Drivers in Java”)



Current Squawk Deployment: Sun Labs Sun Spot Platform

- Hardware
  - 32-bit ARM core
  - Chipcon CC2420 based wireless platform
  - SPI based peripherals
  - Simple sensor board
  - "Bot"-type device with variety of interfaces

- Software
  - Squawk Java VM
  - Desktop build and deploy scripts
  - Libraries for
    - Driving hardware: radio, sensor boards, ...
    - Basic 802.15.4 network functionality
  - SpotWorld: graphical desktop interface

Squawk Software Libraries

- Standard J2ME Java libraries
  - CLDC 1.0
- Hardware libraries
  - SPI, AIC, TC, PIO drivers all in Java
  - Sensor board hardware driven by Java (no C)
    - ADCs, GPIO, IRDA, etc.
- Radio libraries
  - Drive Chipcon CC2420 hardware from Java (no C)

Squawk Software Libraries (2)

- Network libraries
  - 802.15.4 MAC layer in Java (no C)
  - Simple GCF implementations of connections
- Desktop libraries
  - Create connections from standard J2SE VMs to wireless devices
  - Utilize Spot in testboard as a gateway

Example: Application

```
//Open a stream over the radio

StreamConnection conn = (StreamConnection) Connector.open
                                              ("radio://"+otherSpotAddress+":100");

DataOutputStream output = conn.openDataOutputStream();


//Read pin 4 of the ADC on the Sensor board (ADT7411 is the type of ADC)

RangeInput input = new ADT7411RangeInput(Sensorboard.getADC(),4);


//Loop and send the data

while(true) {

    try {

        output.writeInt(input.getValue());

        output.flush();

        Thread.yield();

    } catch (Exception e) {

        System.err.println("SENDER problem "+e);

    }
```

Example: Sensor

```
public synchronized static Accelerometer3D getAccelerometer() throws IOException {
        if (accelerometer == null) {
            //get the ADC inputs
            RangeInput xInput = new ADT7411RangeInput(getADC(),4);
            RangeInput yInput = new ADT7411RangeInput(getADC(),5);
            RangeInput zInput = new ADT7411RangeInput(getADC(),6);

            //get the contol pins
            SingleBitOutput selfTest = new MAX6966SingleBitOutput(getIOPort1(),7);
            SingleBitOutput powerDown = new MAX6966SingleBitOutput(getIOPort1(),8);
            SingleBitOutput fullScale = new MAX6966SingleBitOutput(getIOPort1(),9);
            accelerometer = new LIS3L02AQAccelerometer
                                (xInput,yInput,zInput,selfTest,powerDown,fullScale);
        }
        return accelerometer;
    }
```

Experimental Results (April 15, 2005)

Benchmark	.class	.suite
Richards (Gibbons)	11,770	4,584
Richards (Deutsch)	19,655	6,788
DeltaBlue	27,520	9,724
Game of Life	7,390	3,396

Sampling (samples/sec)	
ARM PIO lines	11,760
Sensor board input lines	300-800

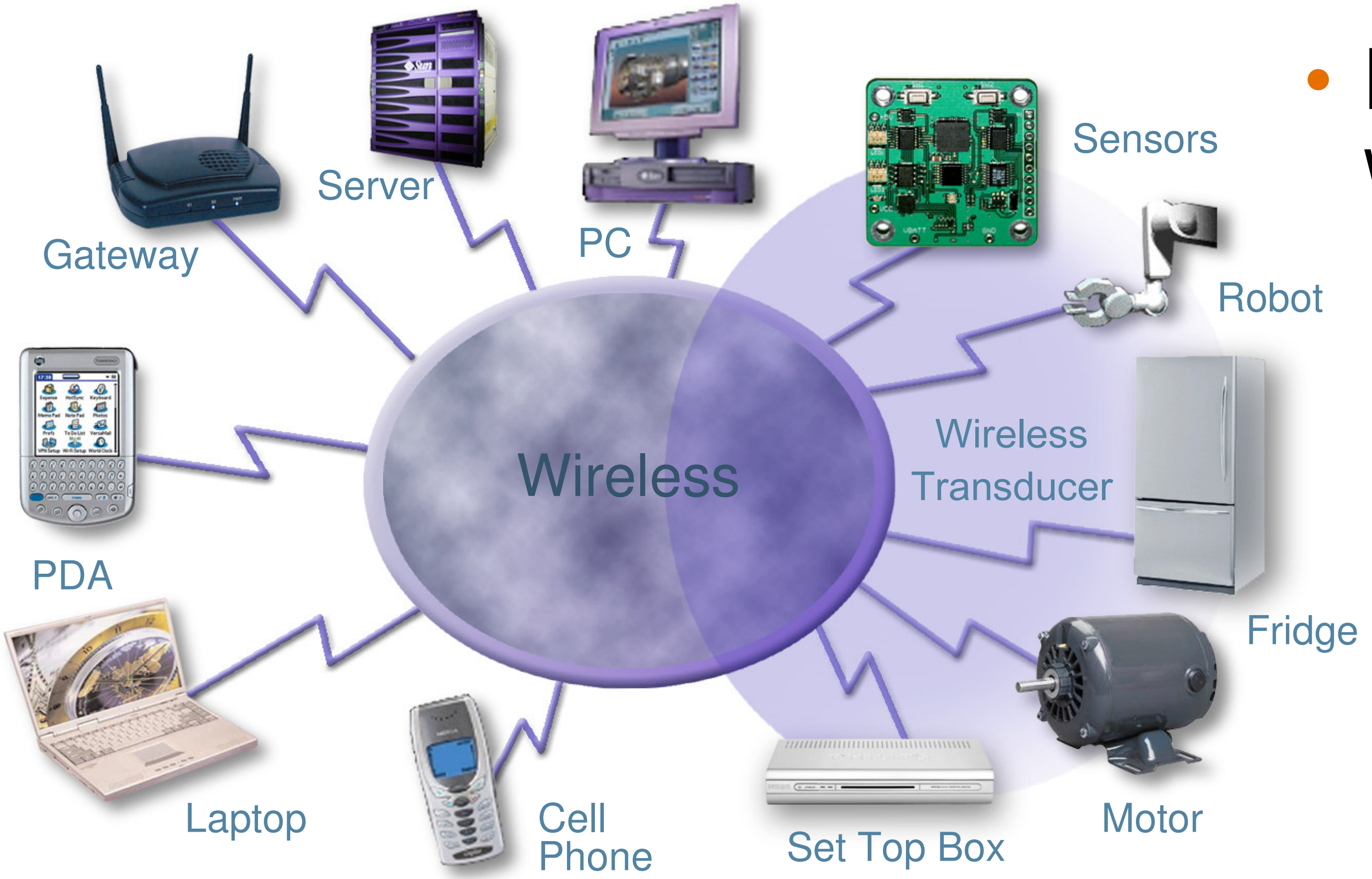
Radio range:	90 mts

Benchmark	LOC	ms on ARM7 EB40 board
Richards (Gibbons)	410	5,277
Richards (Deutsch)	456	8,382
DeltaBlue	984	4,766
Game of Life	354	4,032

Squawk on SunSpot Facts

- Java VM written in Java
- Interpreter based (at present)
- Memory sizes:
  - 80K RAM for VM
  - Libraries 380K flash
- Suites
  - 38% smaller than jar'd class files
- Performance: comparable to KVM
- Device drivers written in Java (no C)
- 802.15.4 MAC layer in Java (no C)

Future:
A Platform for Cooperating Devices



- Network/ wireless
  - > Cell phones
  - > Set top boxes
  - > Home appliances
  - > Robots

Summary

- Developers
  - Fully capable J2ME that runs on “bare” metal
  - Portable: no native code nor OS needed
  - Standard Java development and debugging tools
- Business
  - Powerful and flexible technology for connecting mobile devices and big computers using Java
  - Reduces programming and testing complexity across multiple OS/CPU configurations

Cristina Cifuentes

cristina.cifuentes@sun.com