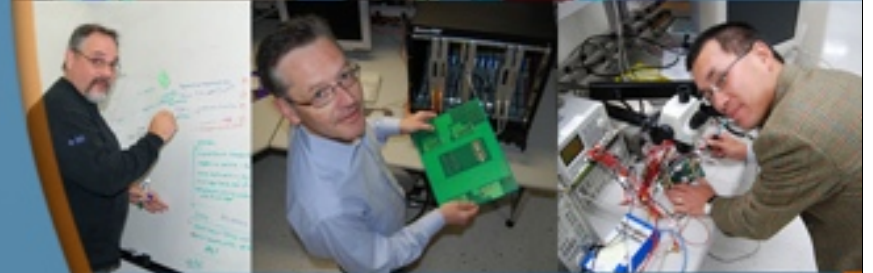




Project Squawk Java™ for Small Devices

Eric Arseneau, PI Squawk
Sun Microsystems Laboratories

<http://research.sun.com/projects/squawk/>



2008
Sun Labs
Open House



Java™ Growth ?

- Java™ on billions of devices today
 - > Servers, Desktops, Phones, ...
 - > Great !!!
- Implied limits: 1 per 1000s users, 1 per person, ...
- Embedded potential MUCH larger
 - > Multiple embedded CPUs/MCUs per thing
- Smaller devices/Micro-controllers
- Limit is > 1 per unit: No limit of one per person !!!
- Consumer and Industrial devices

Why Java™ ?

- Embedded development is hard
- Leverage Java™ expertise and code-base
- Use standard tools
- Ease of porting and emulation
- Java™ technology end to end

Squawk

- Java™ Operating Environment for small devices
- Bootloader/Micro Kernel
- Java™ ME CLDC 1.1 IMP 1.0
- Java™ on the “bare metal”
- Virtual Machine
- Device Drivers/Interrupt Handlers
- Sun Microsystems Laboratories project



Design Goals

- Optimize for small devices
- Java™ technology enabler
- Java™ in Java™
- Java™ on Java™
- Research and production

Small ?

- Resource constrained, not size
- Memory
- FLASH/ROM/RAM
- Power/Energy
- Processing
- Hardware components

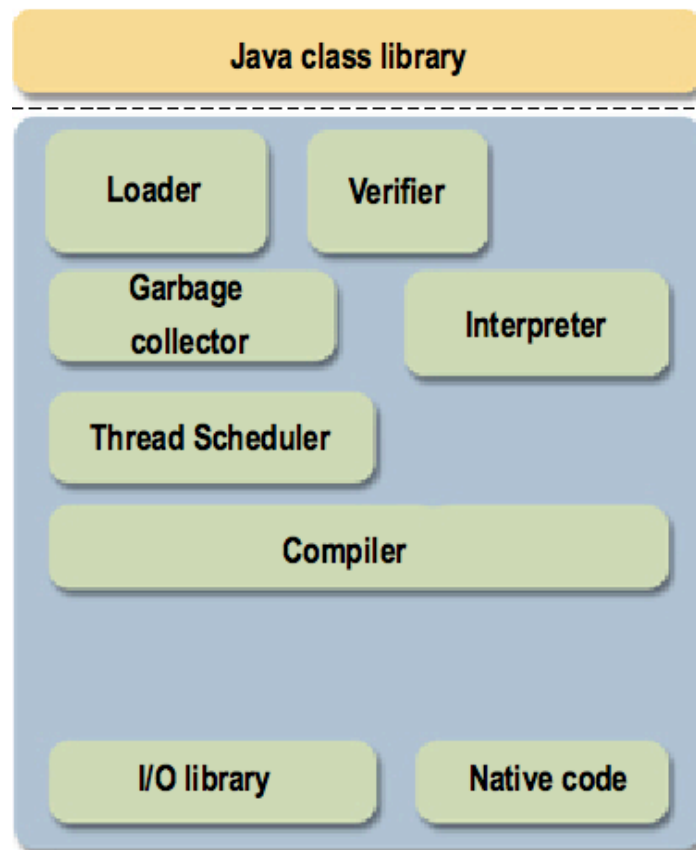
Is there a market ?

“If you could fit that platform into low end MCUs that are used in the toy industry, everyone would adopt it for convenience, for its speed to market and its upgradeability.”

Davin Suffer
CTO
WOWWEE Corporation

Standard VM vs Squawk

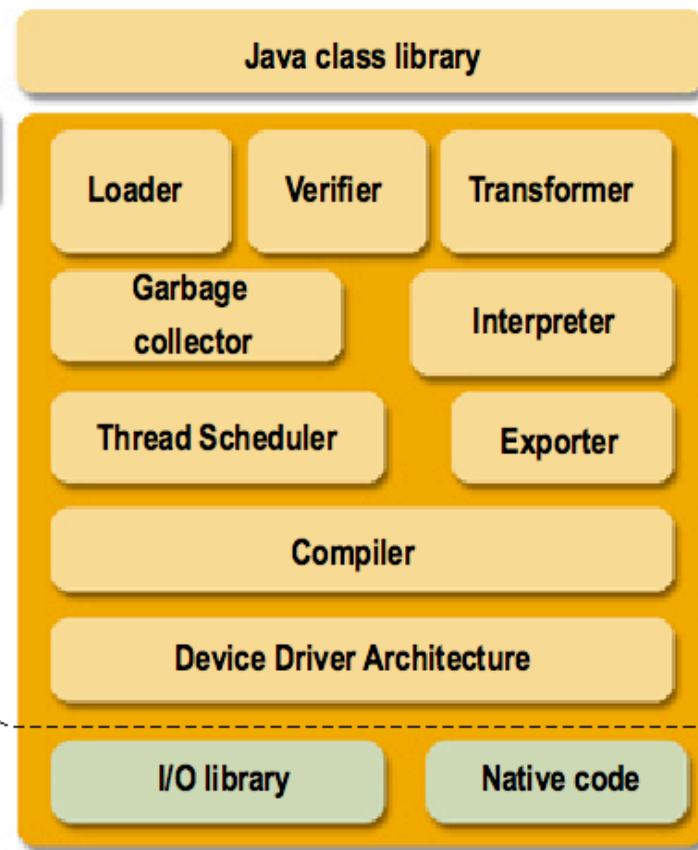
Standard JVM



Java code

C code

Squawk JVM

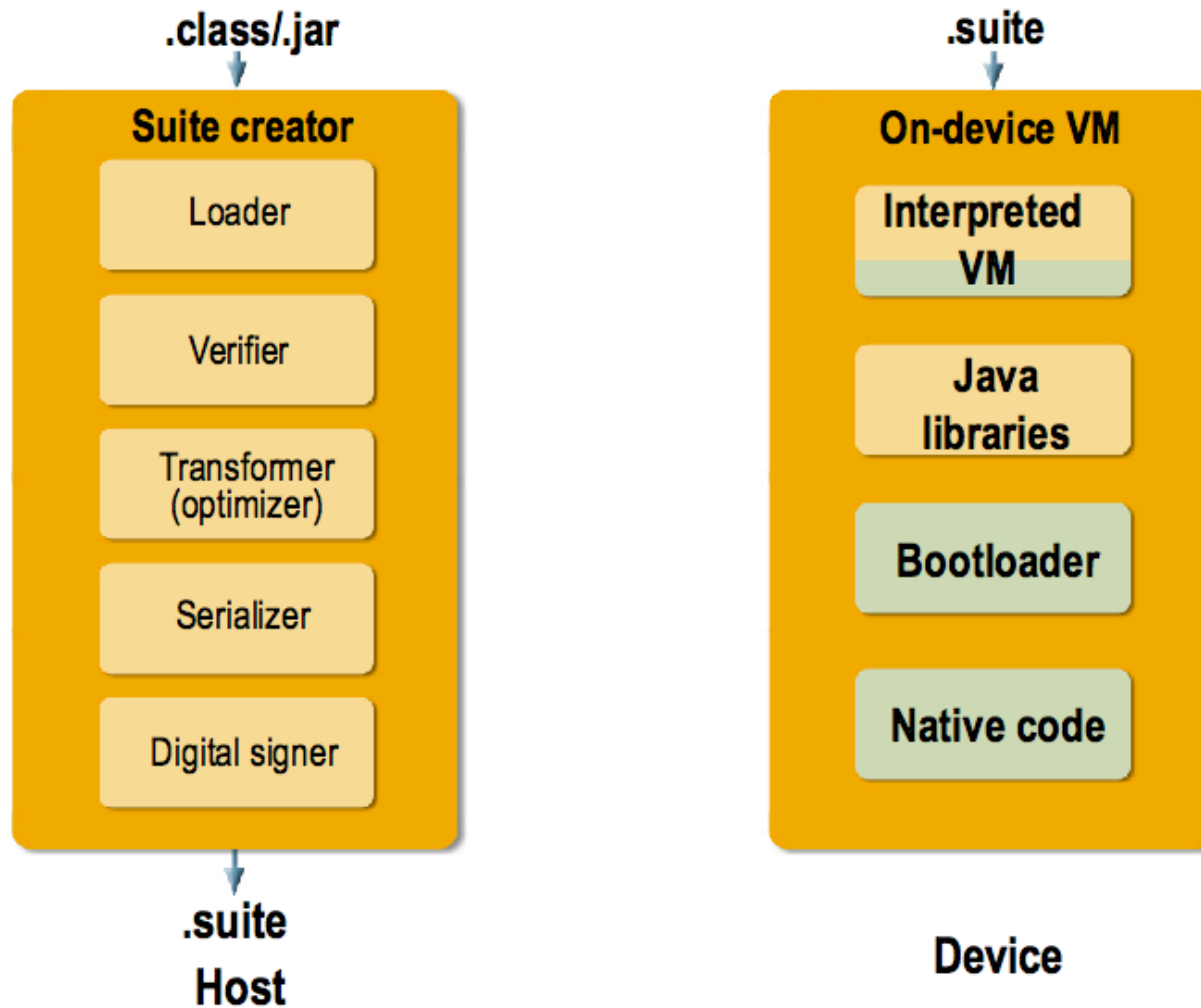


Device Driver Architecture

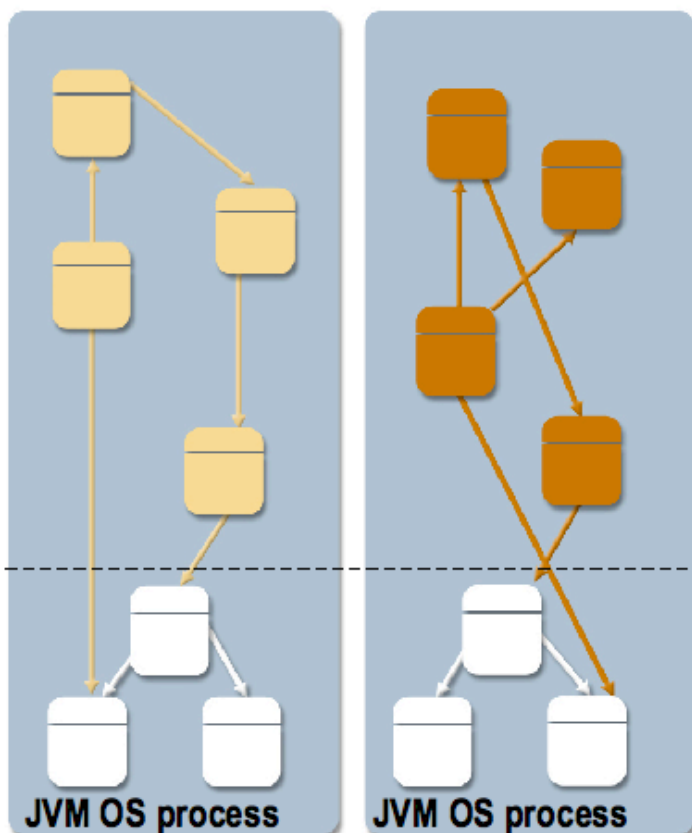
Interpreter/AOT

- Classic interpreter today
- GC written in Java -> C
- JAR -> Suite
- Intention
 - > No JIT
 - > Complete VM in Java -> C
 - > Application & VM -> C
 - > C is a step towards native language of target

Split VM

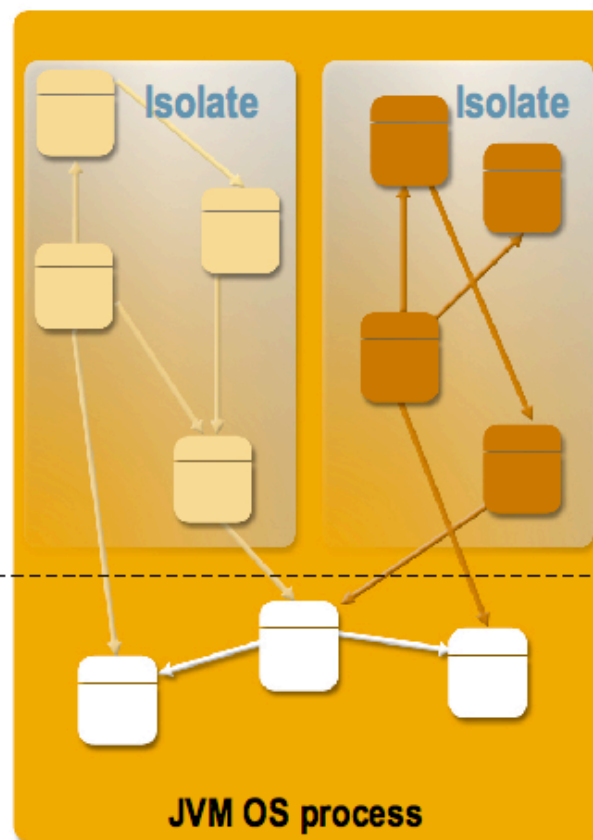


Isolates

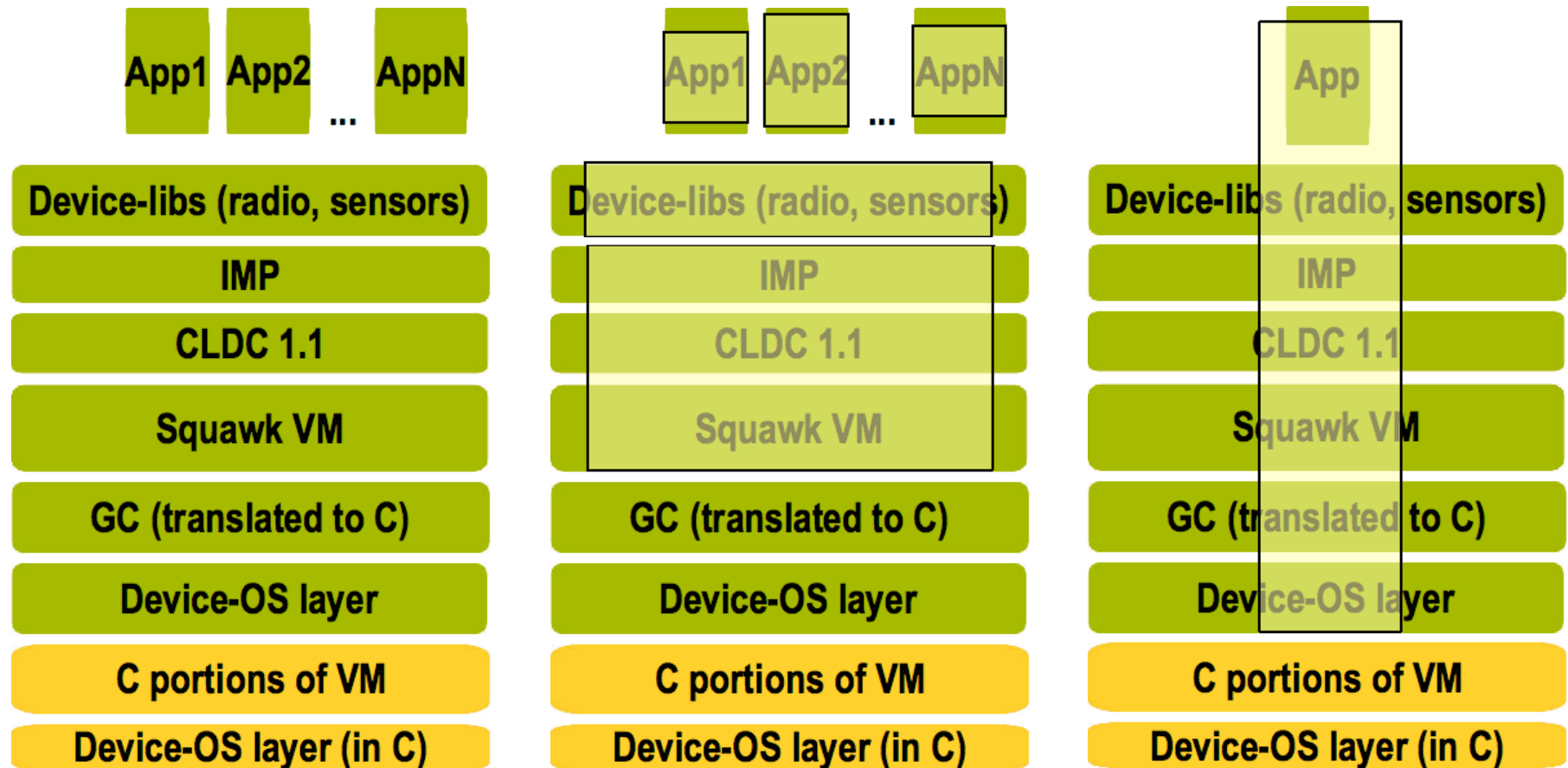


Non-shareable object memory

Shareable object memory

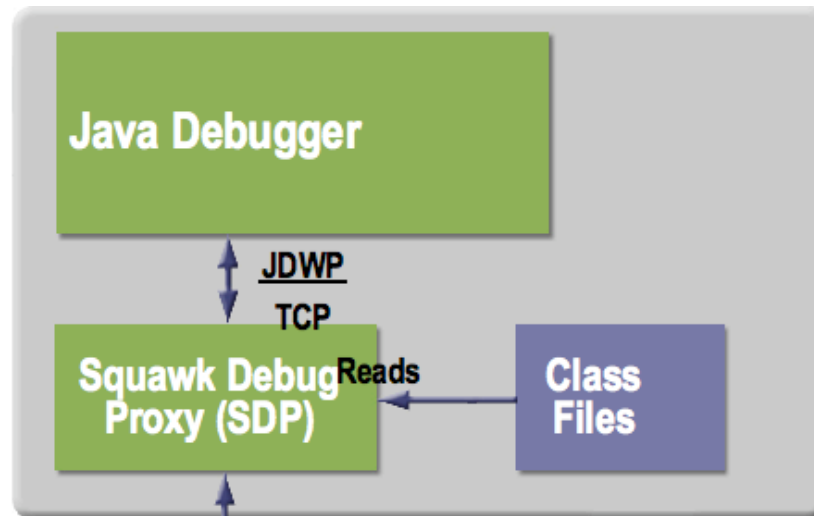


Trimming

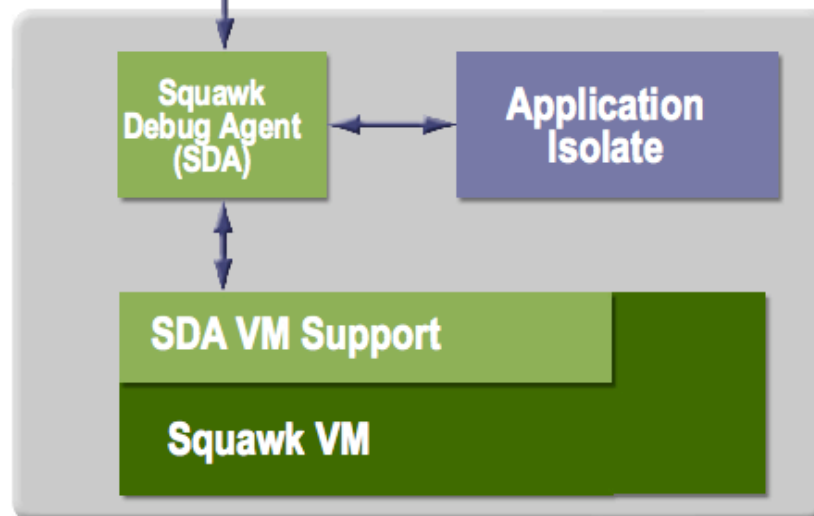


Debugging

Developer Workstation



Sun SPOT



SDWP/Wireless

JDWP
TCP

Reads

Squawk
Debug Agent
(SDA)

Application
Isolate

SDA VM Support

Squawk VM

Real-Time

- We are soft real-time, need to be real-time
- Latency is controlled by GC
- We are making significant progress, but this is HARD for small
- Defining a new platform for RTSJ to target

Open Source

- Released Jan 22, 2008
 - > <https://squawk.dev.java.net>
- Leverages Open Sourced phoneME Feature
- Even with different VM implementation
- Official source is the best way to be compliant
- Functional and tested code base for core API
 - > 75% of core API is phoneME source
 - > Can achieve much more
- Community is an excellent source of help
- Open source licensing simplified

What's there ?

- Complete source code
- Build system for all desktop platforms
- Web site with limited information now
- On-going development to be live
- Bug tracking live
- Forums

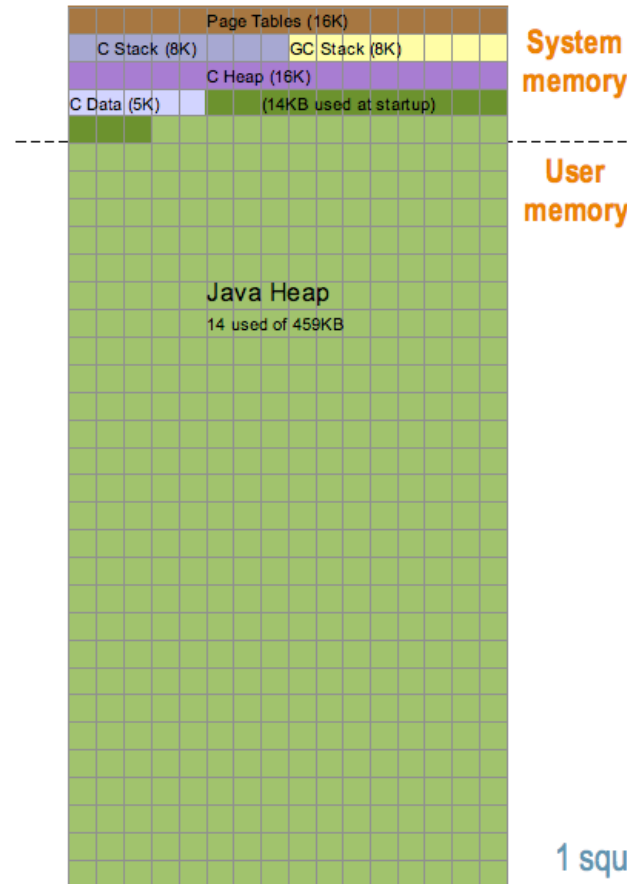
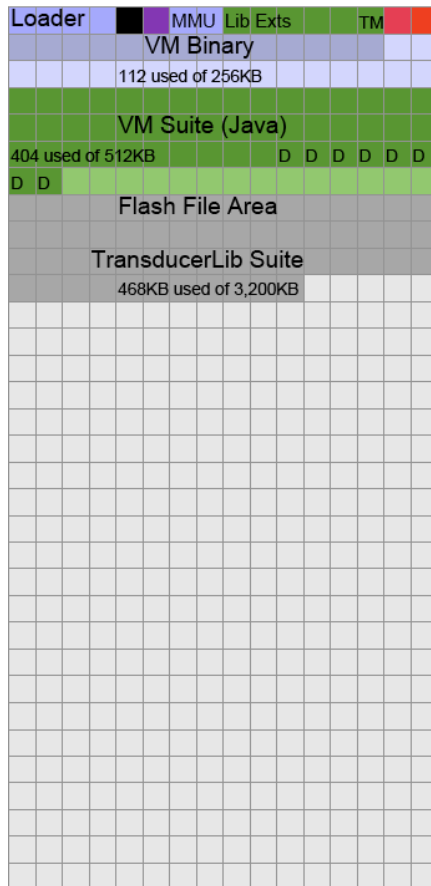
Platforms

- Certified
 - > Solaris SPARC/x86/kernel
 - > Linux
 - > Mac OS X PPC/x86
 - > Windows
 - > Sun SPOT
- Open source ports underway
 - > LEGO Mindstorm NXT
 - > 8051 MCU
 - > Custom FPGA
- Stay tuned for more coming soon



Sun SPOT Port Numbers

- Benchmarks - http://practicalembdedjava.com/benchmark/11a/benchmark_11b.html



Is there a need ?

“Sentilla is not in the VM business, if Squawk could fit on our platform and provide the same power characteristics and capabilities, Sentilla could better focus our efforts on the higher level APIs, services, and customer needs that drive industry adoption.”

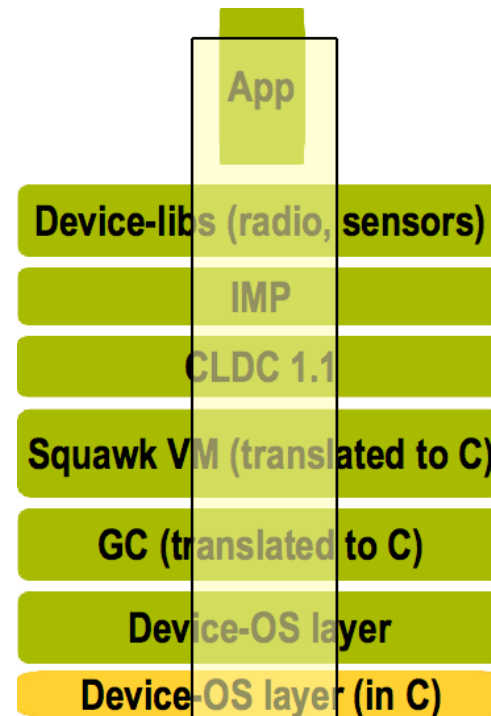
Joe Polastre

CTO

Sentilla

Future

- Complete Java™ in Java™
- Complete generation of VM
- Introduce ahead-of-time compilation
- Subset of RTSJ
- Extend trimming



Q&A

- Thank you
- eric.arseneau@sun.com
- <http://squawk.dev.java.net>