Java is a trademark of Sun Microsystems, Inc.

Java

JavaOne℠

From Parking Meters to Netbooks: Java™ Platform, Standard Edition 6 for ARM-Based Embedded Devices

Bob Vandette
Sun Microsystems, Inc.
Java SE Embedded Technical Lead

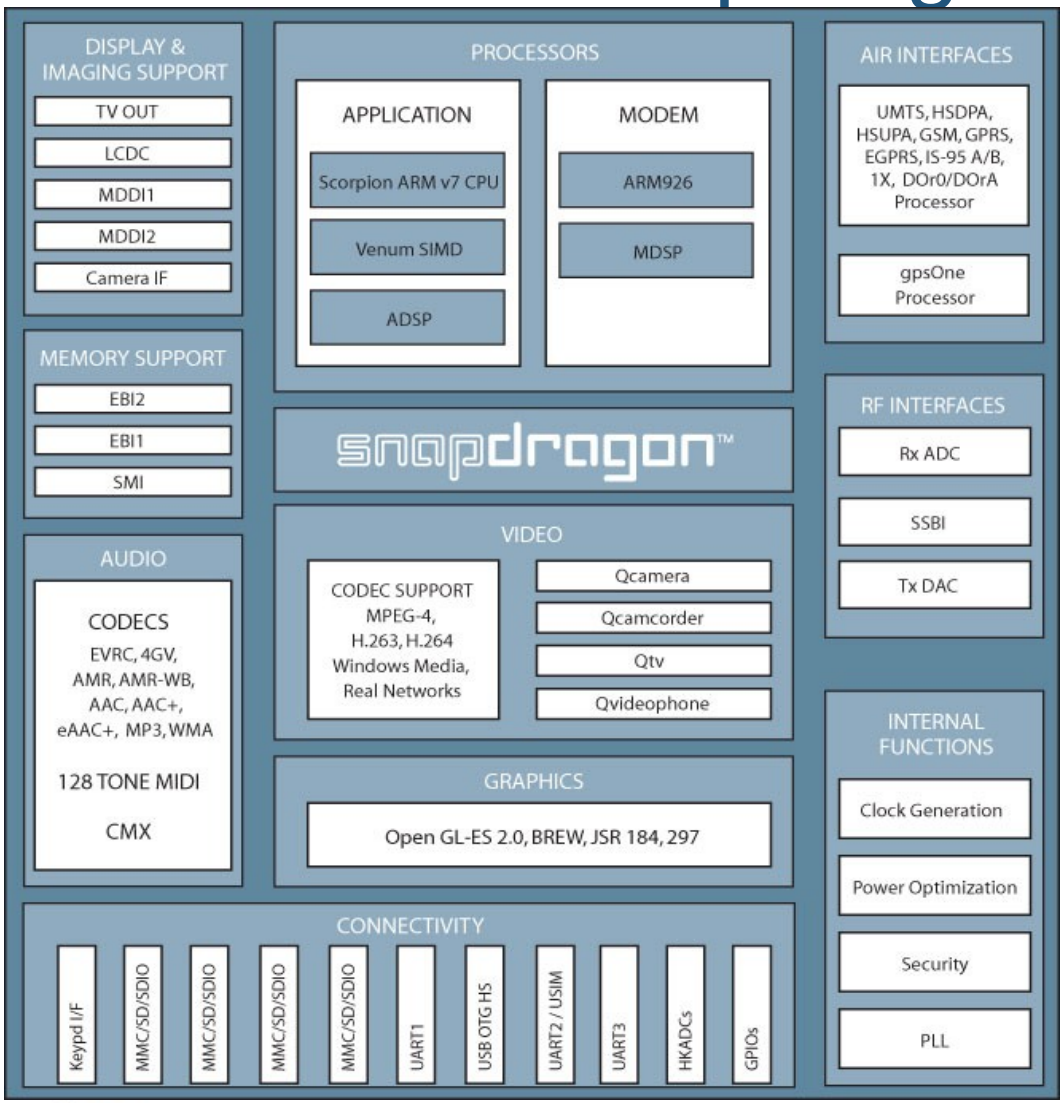
Introduction
Java SE 6.0 for Linux ARM Devices

- ARM Processors
- Example Embedded Devices Using ARM
- Challenges supporting ARM CPU's
- Java SE Embedded Enhancements
- Currently Supported Configurations
- Tips for Developing Java on Java SE Embedded
- Future Java SE Embedded Directions

ARM Processor

- ARM CPU History
  - First design in 1983.
- Why is ARM popular?
  - Low power, most popular mobile embedded CPU
- Multiple Vendors
  - ARM, Qualcomm, TI, Marvell, many more
- What's new in ARM processors
  - Higher Clock Speeds
  - Multi-CPU and Multi-Core designs coming
  - High Level of Integration with video, 3D, wireless, bluetooth, gps

Qualcomm Snapdragon Chipset



- Fully integrated solution including modem, Cortex-A8 ARMv7 applications processor, GPU, multimedia, peripherals
- CDMA and UMTS modems
- First Superscalar 1GHz application processor for mobile devices
- Advanced hardware accelerated codecs, 2D and 3D graphics
- Sub-500mW applications processor

Java on ARM History

J2ME

2001 CLDC – Connection Limited Device Configuration
Used primarily in Cell Phones

2002 CDC – Connected Device Configuration

JAVA SE

2007 Java 2 Standard Edition for Embedded 1.4.2
Linux ARM Headless

2009 Java Standard Edition for Embedded 6.0
Linux ARM Headless and Headful Configurations

Sampling of Embedded Applications
using Java SE Embedded on ARM

Parking Terminal

NetBook

Plug Computer

Video Gaming System

Multi Function Printer

Technical Challenges for Hotspot Support of ARM CPU's

- Several popular ARM architecture versions
  - ARMv5, ARMv6, ARMv7
- Many sub features to choose from
  - Thumb, Thumb2, Jazelle, RCT
    - Not available on all versions
  - Big Endian and Little Endian
- Floating Point Support
  - Optional (Requires Software Floating Point Support)
  - Different implementations (VFPv2, VFPv3)

Java SE for Embedded ARM Processor Support

- Goals
  - 100% compatible with Desktop Java SE 6.0
  - Broadest coverage with least number of binaries
  - Take advantage of specific performance improvement features on each architecture
- Implementation
  - Four different little endian binaries supporting the most common use cases

armv5te	Software Floating Point
armv5te	VFP Hardware Floating Point
armv6	Software Floating Point
armv6	VFP hardware Floating Point

The armv6 binaries auto detect armv7 (Cortex-a8) and take advantage of v7 specific features

Java SE for Embedded ARM Processor Support cont.

- armv5te specific support
  - atomic ops and Java level locking use mutexes after attempting a swap hashed on the address of the lock
  - Use literal pools for loading 32 bit pointers and constants with relocation support. The JIT allocates literal pool in jit'd method memory.
- armv6 support
  - Uses ldrex/strex for atomics and locking rather than slower armv5 support
  - Use rev instruction for performing byte swap
- armv7 (Cortex-A8)
  - All armv6 improvements plus, movw/movt instruction combinations are used for faster literal loading

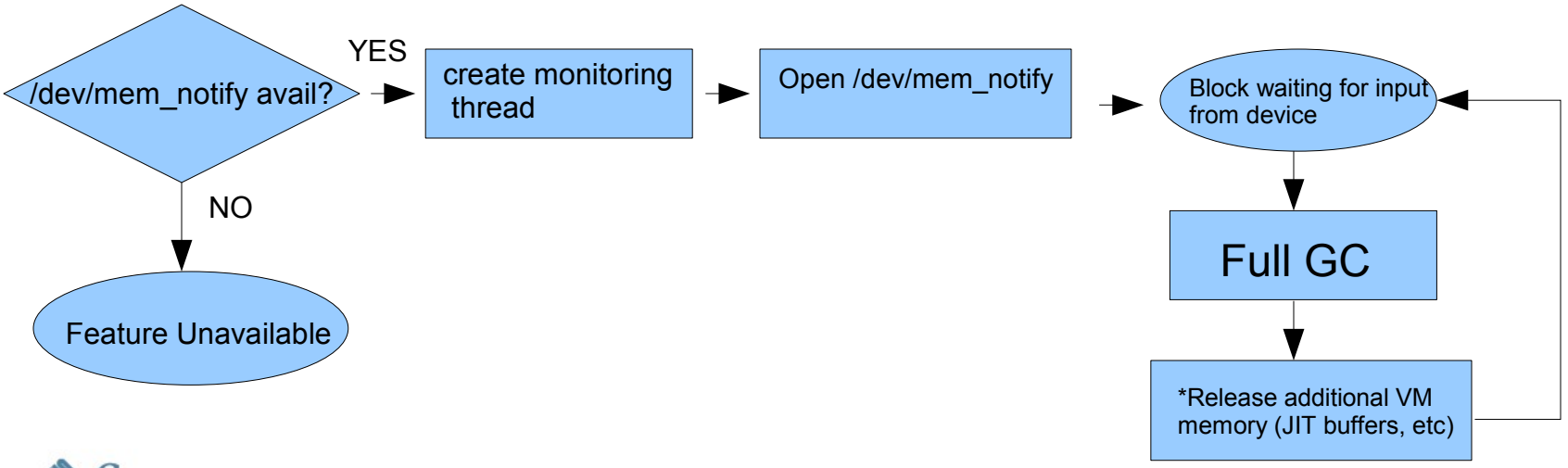
Java SE Embedded Specific Features

- Headful and Headless builds
- Software Floating Point
- Low Memory Detection
- Power Conservation
- Gstreamer Audio Support
- Alternate UI and Window Manager Support
  - **xfwm4***, Matchbox, ICEWM, metacity
- Browser, Plug-in Support for Mobile Device Browsers
  - **Firefox***, Coolfox
- Tested Linux Distributions
  - **Xandros***, Redflag Midinux, Fedora

* Full Certification Planned for First Release

Java SE Embedded Feature
Low Memory Support

- The Linux community has been searching for solutions to handle low memory conditions
  - Killing processes is NOT the answer (SIGABORT, ouch!)
- We chose to support /dev/mem_notify
  - Linux kernel driver monitoring available memory



Java SE Embedded Feature
Power Conservation

- Goal
  - Reduce CPU usage to 0% when Java applications are idle in order to allow CPU clock to be reduced or suspended
- Modifications Performed
  - Hotspot Virtual Machine
    - Eliminated all Polling Threads
  - Java SE AWT/Swing Changes
    - Eliminated Polling for X Events when idle

Java Tips for Netbook devices

- Don't assume overlapping Windows
- Don't assume 640x480 resolution or higher
- Eliminate polling for I/O in your Java and Native code
- Only blink cursor when activity present

ThinkFree Java Application Experience
Optimizing ThinkFree Office Suite for Netbooks

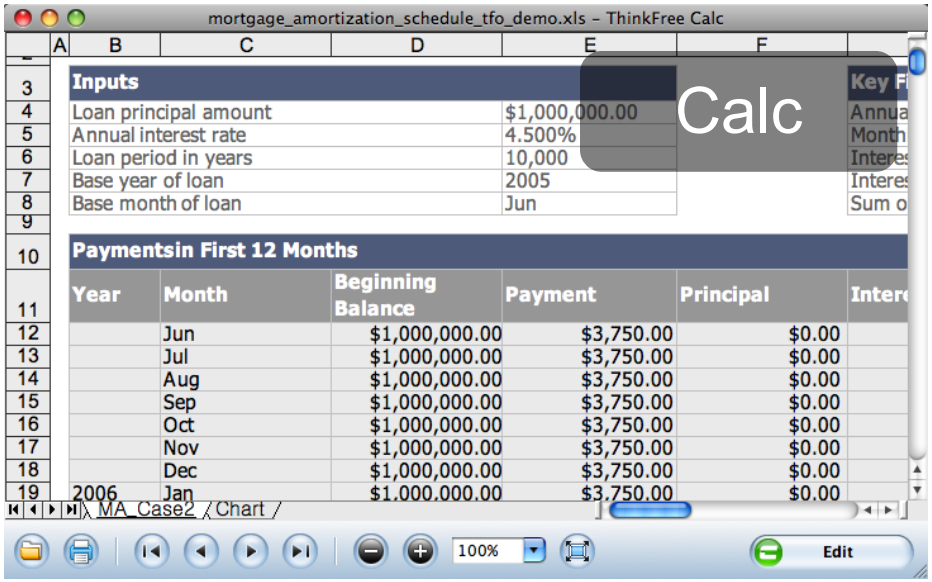
- Netbook is ....
  - Lower in Price
  - Lower Power CPU
  - Less RAM and Disk Space
  - Lower Screen Resolution
  - Consumer Oriented Device
  - (Optionally) Touch Screen

Thanks to Geon Goo (accent@thinkfree.com) for input on ThinkFree optimizations

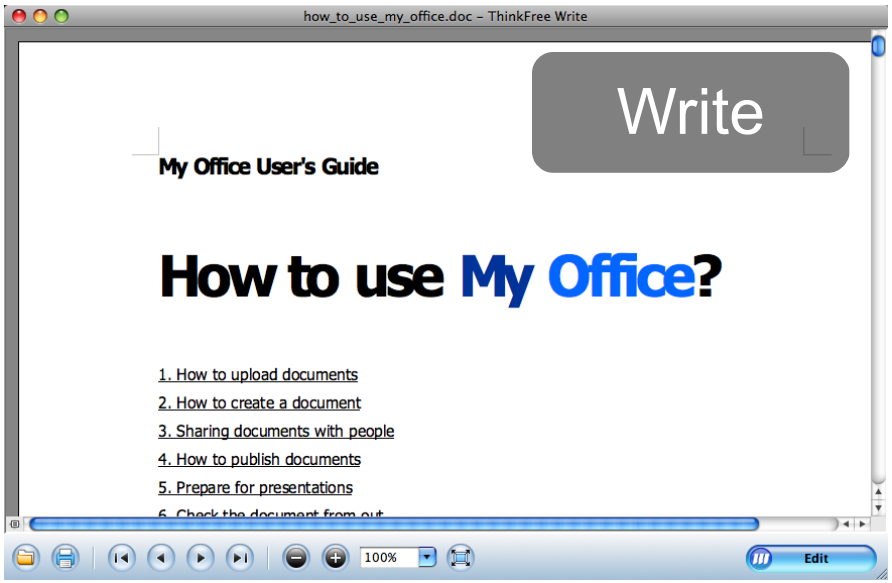
ThinkFree Java Application Experience
ThinkFree Mobile Viewer

- Optimized for Smaller Screen (Simplified UI)
- Faster Startup
  - Split Viewer functionality from Document Editor
    - Document Editor is one click away
  - Less Java Code Loading
  - Class Data Sharing Enabled helped by 25%
  - Used Native Splash Screen Support from SE 6
- Issue GC operations during idle times to reduce interruptions but eventually block to conserve power
- Reduced Cursor Polling by Decaying timer


Touch UI
Open
View and edit your documents
Sync
Synchronize your documents with ThinkFree Online
Write
New Document
Calc
New Spreadsheet
Show
New Presentation


mortgage_amortization_schedule_tfo_demo.xls - ThinkFree Calc
Calc
Inputs
Loan principal amount
$1,000,000.00
Annual interest rate
4.500%
Loan period in years
10,000
Base year of loan
2005
Base month of loan
Jun
Paymentsin First 12 Months
Year
Month
Beginning Balance
Payment
Principal
Jun $1,000,000.00 $3,750.00 $0.00
Jul $1,000,000.00 $3,750.00 $0.00
Aug $1,000,000.00 $3,750.00 $0.00
Sep $1,000,000.00 $3,750.00 $0.00
Oct $1,000,000.00 $3,750.00 $0.00
Nov $1,000,000.00 $3,750.00 $0.00
Dec $1,000,000.00 $3,750.00 $0.00
2006 Jan $1,000,000.00 $3,750.00 $0.00
MA_Case2
Chart
100%
Edit


what_is_workspace.ppt - ThinkFree Show
Show
thinkfree
What is Workspace?
A Collaborative Space
1 / 9
53%
Edit

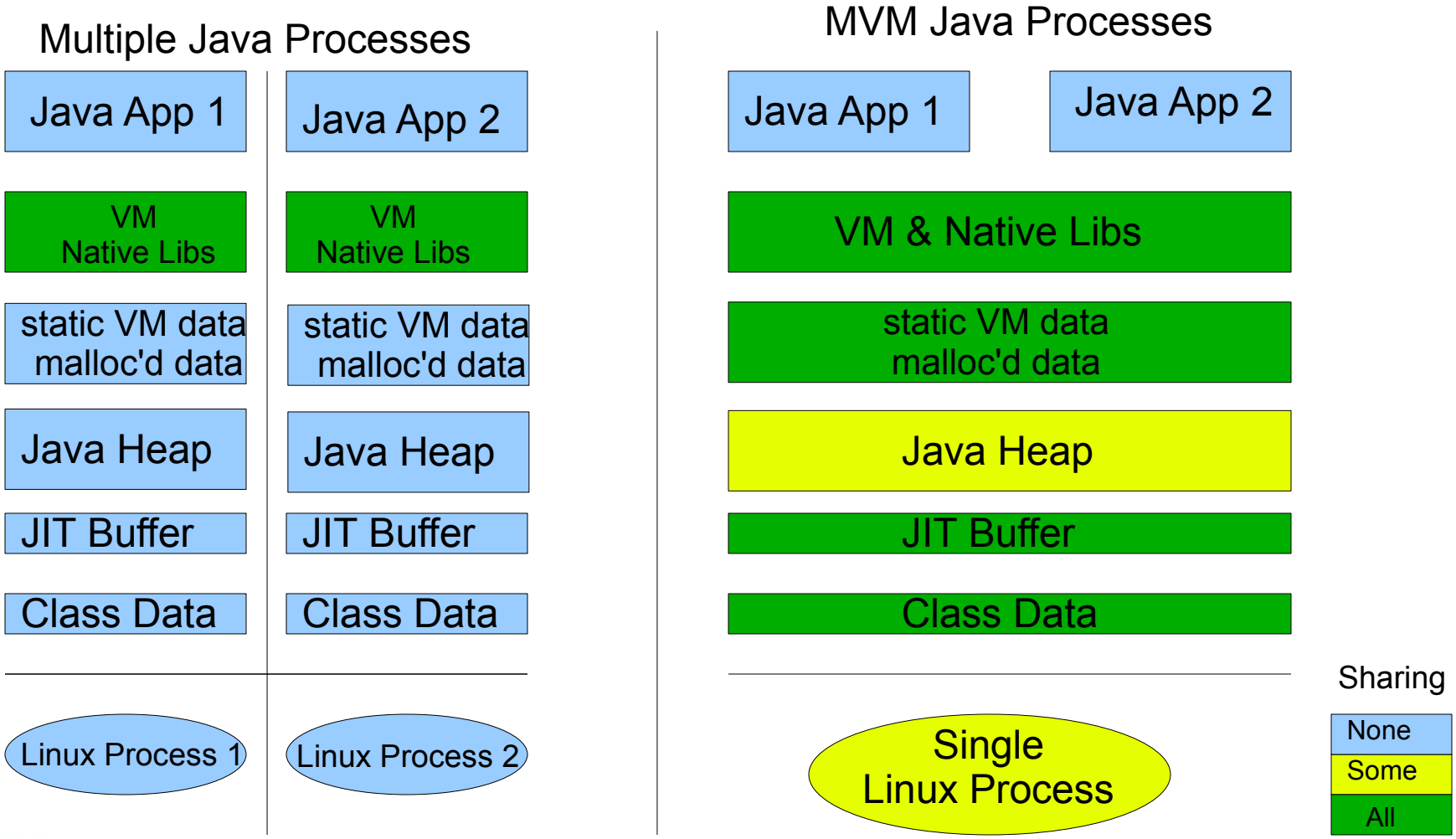

how_to_use_my_office.doc - ThinkFree Write
Write
My Office User's Guide
How to use My Office?
1. How to upload documents
2. How to create a document
3. Sharing documents with people
4. How to publish documents
5. Prepare for presentations
100%
Edit

Java SE Embedded on ARM
Futures

- MVM Support
  - Multi Tasking Virtual Machine
- Multi-Processor ARM Support
- OpenGL ES Support
- Thumb2 Support Evaluation
- Java FX Graphics Acceleration

Multi-Tasking Virtual Machine

memory savings



Demonstrations

- Linux ARM Netbook
  - Full Java SE 6.0 Compatibility on ARM Processor
    - Plug-In Support Launching FX Application
    - Java Office Productivity Application (ThinkFree)
- Multi Tasking Virtual Machine
  - Startup Improvements
  - Memory Savings

Web addresses for more information on Java SE for Embedded

Java SE Embedded home page:
http://java.sun.com/javase/embedded/index.jsp

Java SE Embedded downloads page:
http://java.sun.com/javase/downloads/embedded.jsp

Java SE Embedded system requirements page:
http://java.sun.com/javase/embedded/system_requirements.jsp

Java SE Embedded customers page:
http://java.sun.com/javase/embedded/customers.jsp

Thank You

Bob Vandette
bob.vandette@sun.com

javase-embedded-interest@sun.com

Sun microsystems