

Java Micro Edition Platform & Android

Seminar on Small and Mobile Devices
May 13th, 2008

Dejan Juric (djuric@student.ethz.ch)

Outline

- Motivation & Goal
- Java Micro Edition Platform
- Android
- Java ME vs Android
- Conclusion & Discussion

Motivation

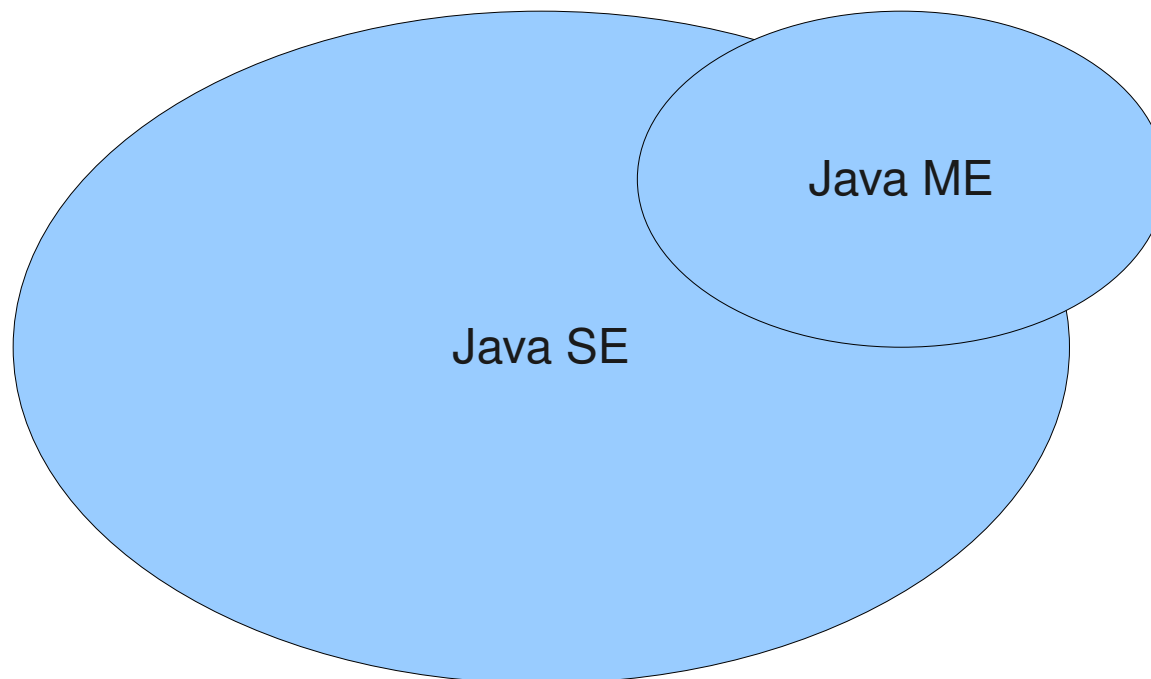
- Number of mobile phone subscribers by November 2007: 3.3 billion. (personal computers: 1 billion)
- Problems with mobile devices:
 - Heterogeneous: different form factors, features, functions
 - Resource-constrained

Goal

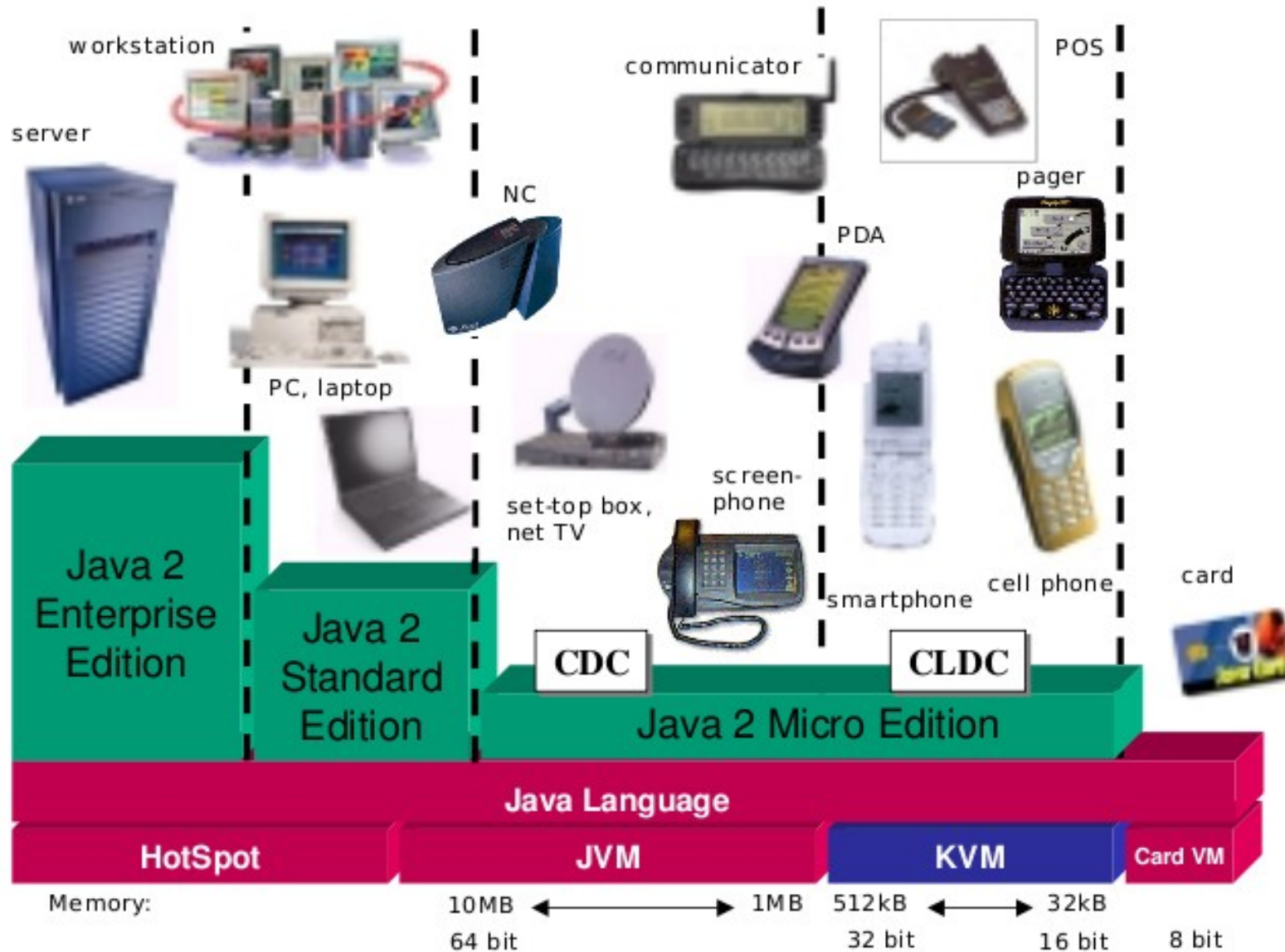
- Allow customization of devices
 - Download new services and applications from the Internet
- Development of applications and services
 - Device-independent
 - Cost efficient

Java Micro Edition Platform

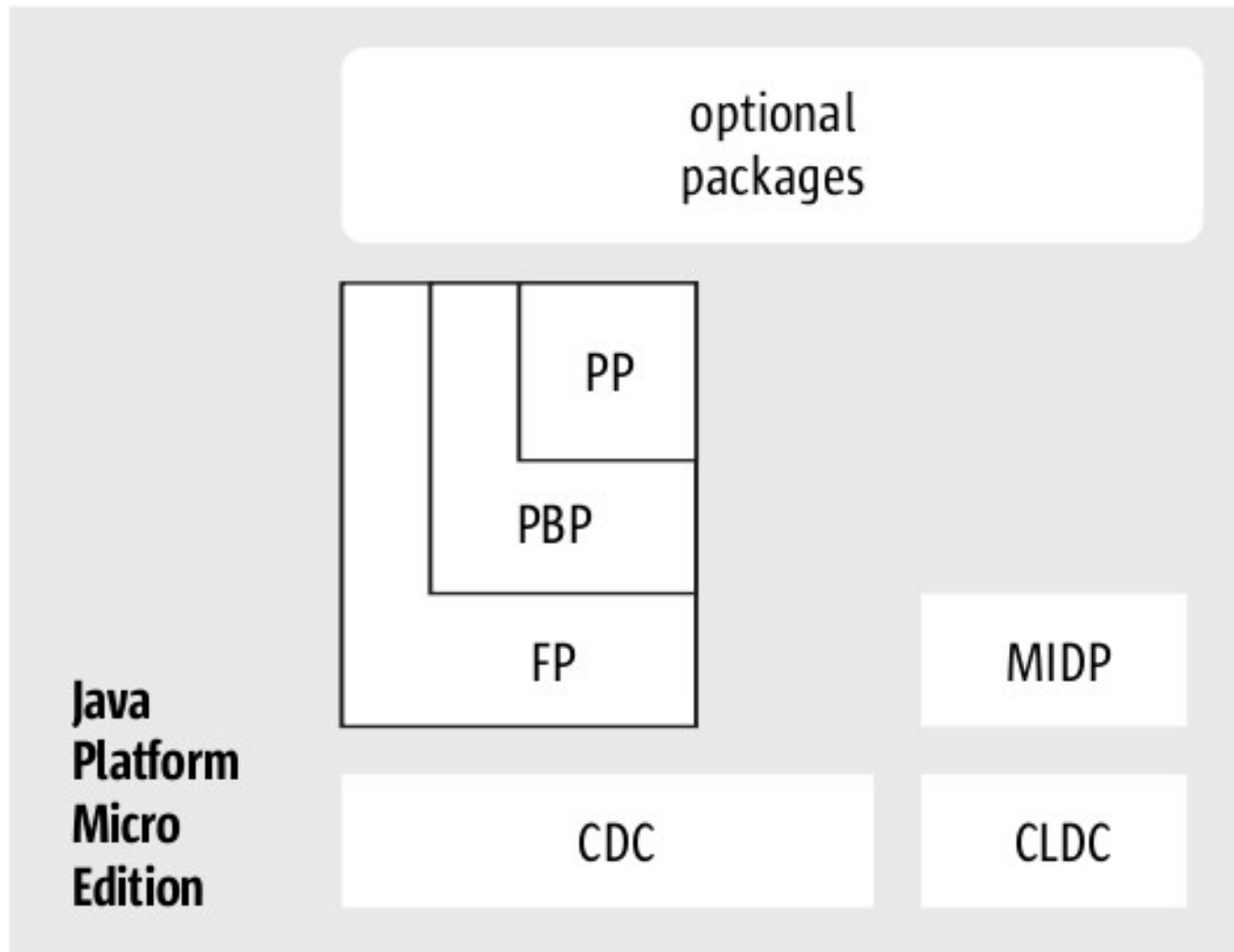
- Available on most current mobile phones
- Subset of the Java Platform
 - for resource-constrained devices



Java Platforms



Java Micro Edition - Overview

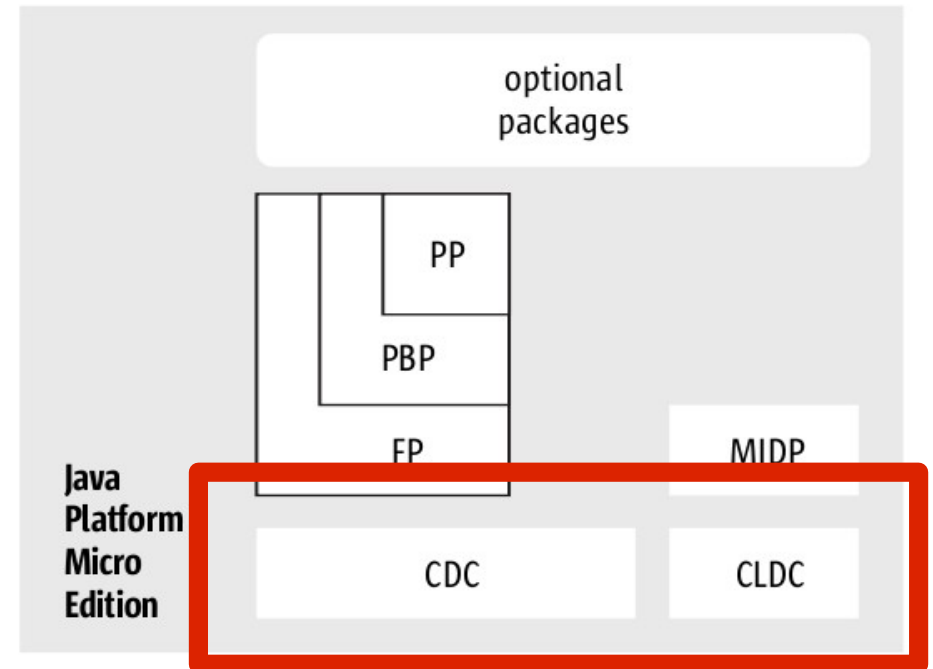


Profiles

Configurations

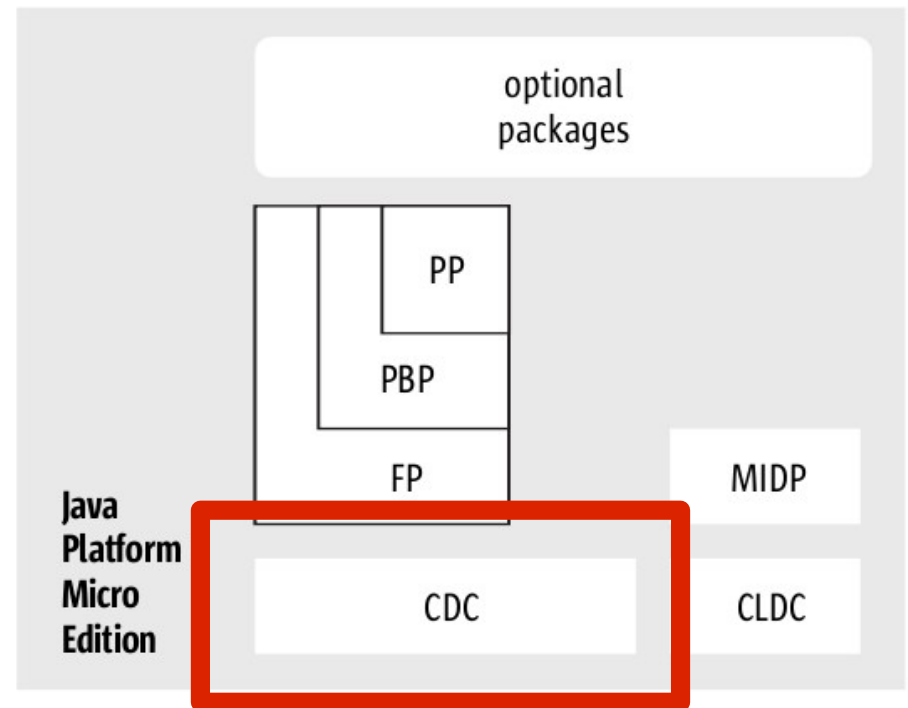
Java ME - Configurations

- Configuration:
complete Java
runtime environment
 - Defines the minimum
set of Java virtual
machine features &
class libraries



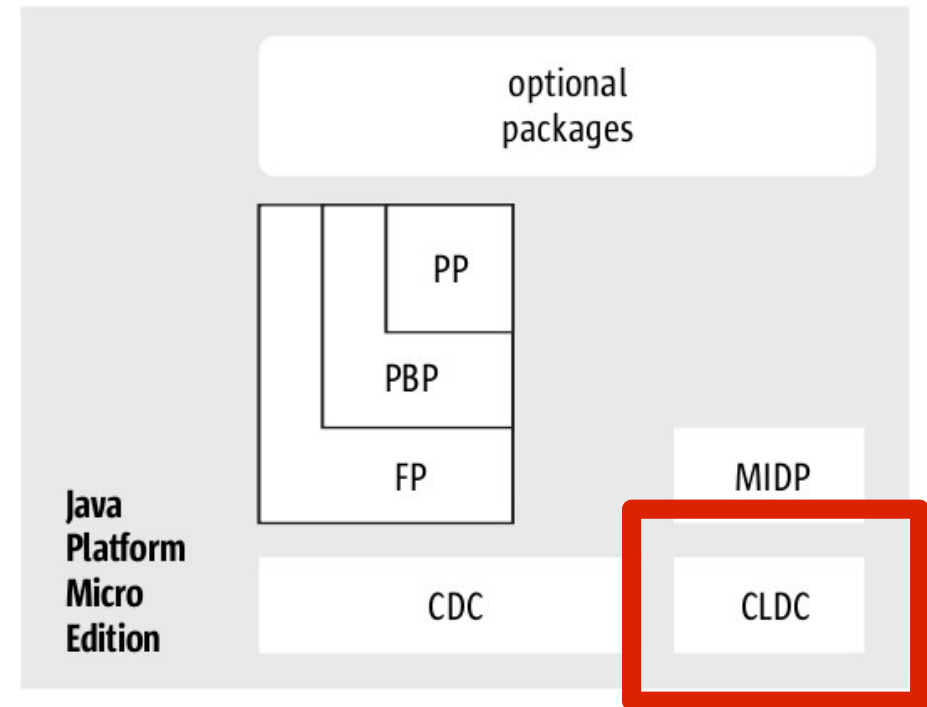
Java ME – Configurations - CDC

- Connected Device Configuration
 - 32bit CPU, 2MB of RAM, 2 MB of ROM
 - Modified Java SE core class libraries
 - CDC 1.1 matches JDK 1.4 APIs



Java ME – Configurations - CLDC

- Connected Limited Device Configuration
 - At least 160 kB of non-volatile memory
 - At least 32 kB of volatile memory
 - Minimal host operating system or kernel available



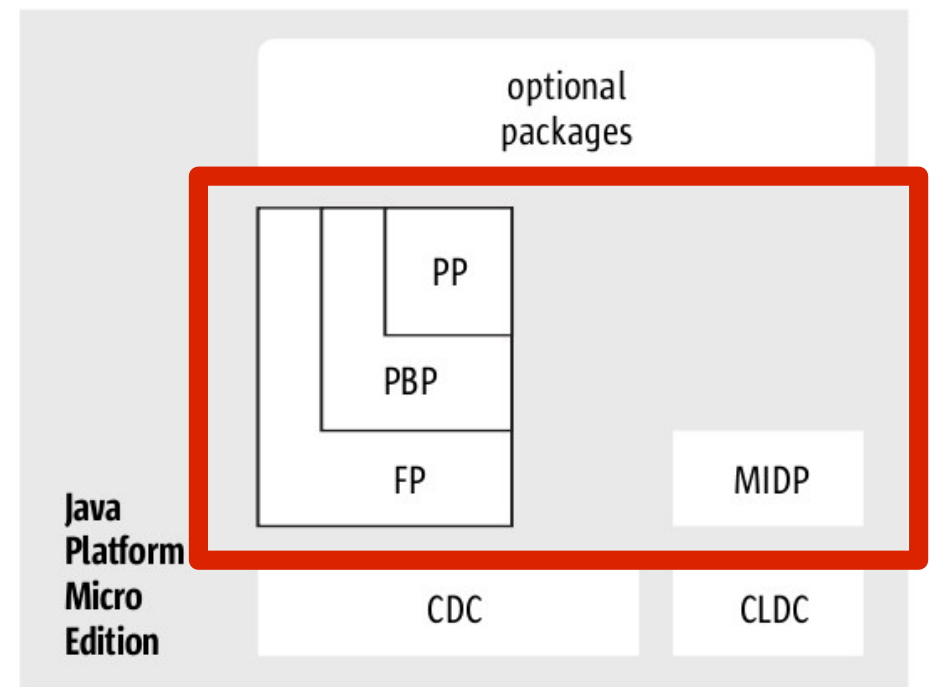
CLDC 1.1
java.lang +java.lang.ref java.io java.util javax.microedition.io

Java ME – CLDC

- Eliminated features
 - User-defined class loaders
 - Thread groups and daemon threads
 - Finalization of class instances
 - Asynchronous exceptions
 - Floating point types (since CLDC 1.1)
- Class file verification
 - Off-device preverification and runtime verification

Java ME - Profiles

- Define a minimum set of APIs available on a “family” of devices
- Applications are written for a particular profile
 - Portable across devices that support that profile



Java ME - MIDP

- Mobile Information Device Profile
 - Screen of at least 96x54 pixels
 - Keypad, keyboard or touchscreen input
 - 256 KB of nonvolatile memory
 - 128 KB of RAM
 - 8 KB of nonvolatile writable memory for persistent data
 - Two-way network connection

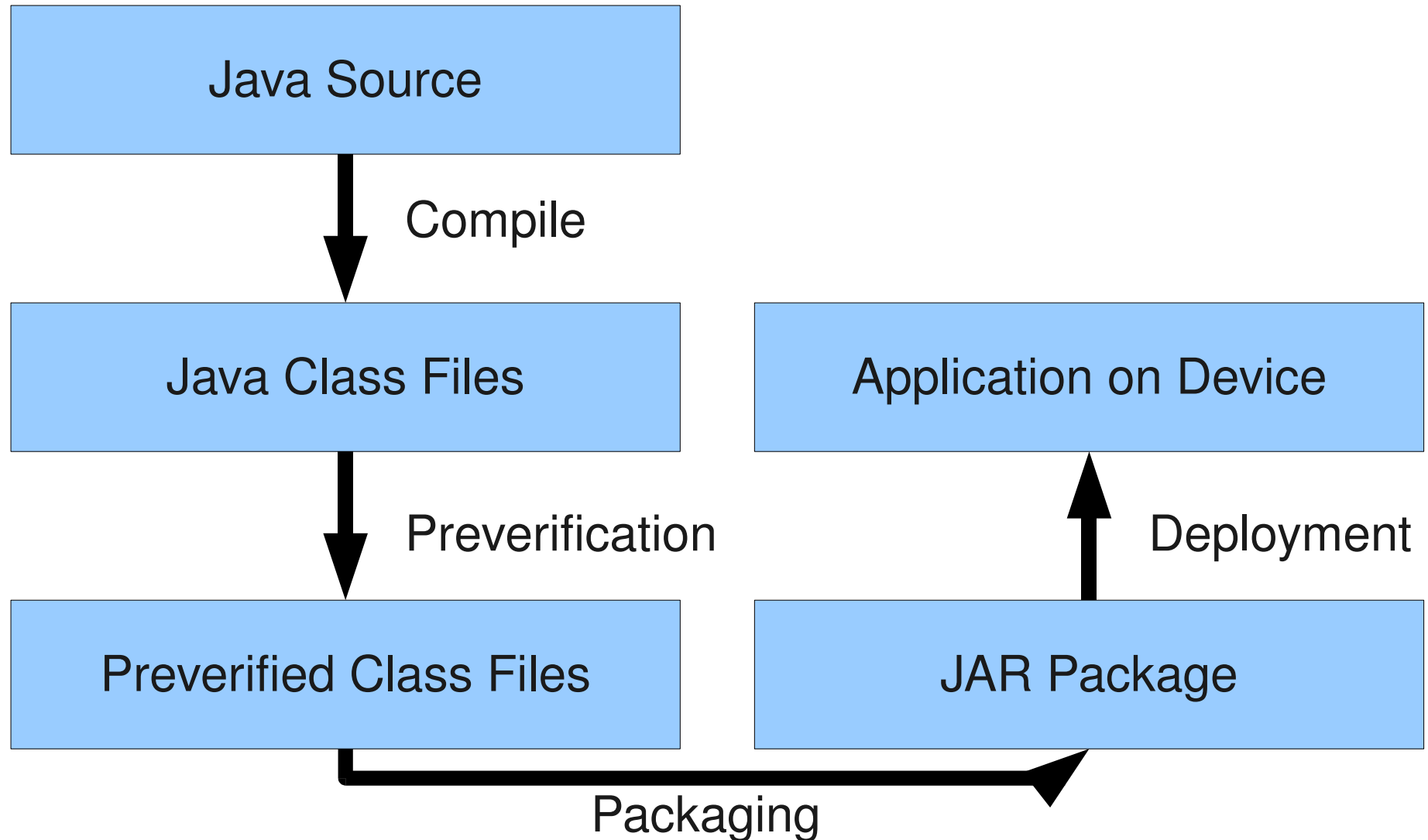
Java ME - MIDP

- Mobile Information Device Profile – Building Blocks

CLDC 1.1
java.lang
+java.lang.ref
java.io
java.util
javax.microedition.io

MIDP 2.0
javax.microedition.lcdui
+javax.microedition.lcdui.game
+javax.microedition.media
+javax.microedition.media.control
javax.microedition.midlet
+javax.microedition.pki
javax.microedition.rms

Java ME – Development Process



Java ME – Development Tools

- Wireless toolkits:
 - Sun Wireless Toolkits
 - Motorola SDK for J2ME™
 - Sony Ericsson J2ME SDK
- Development Environments
 - EclipseME
 - Netbeans Mobility

Java ME – MIDlet Demo

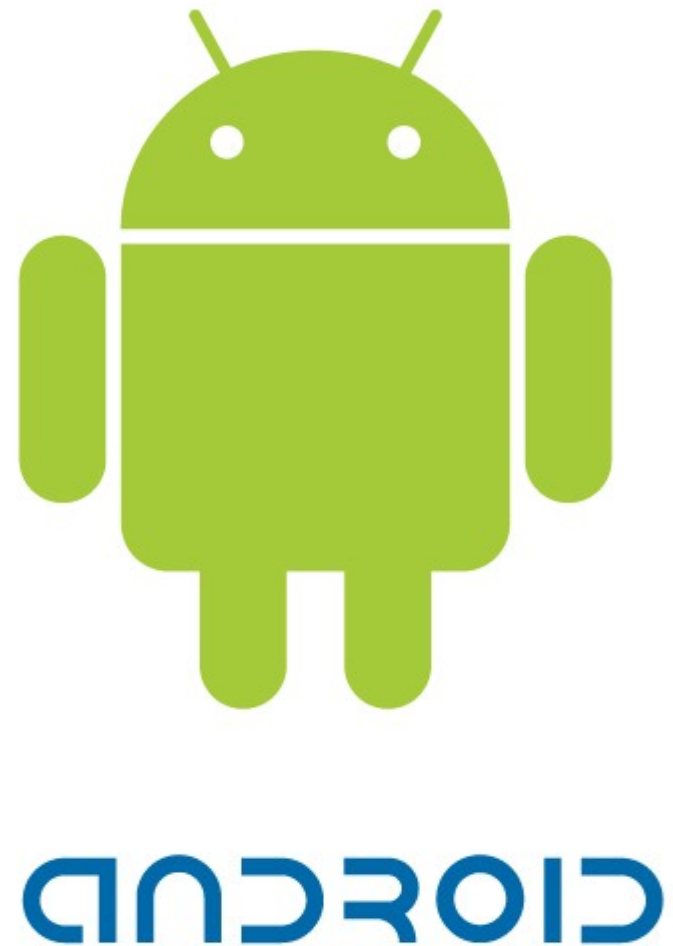
```
// Exceptions omitted for clarity
public class TestMIDlet extends MIDlet implements CommandListener {
    private Form mMainForm;
    public TestMIDlet() {
        mMainForm = new Form("TestMIDlet");
        mMainForm.append(new StringItem(null, "Hello, MIDP!"));
        mMainForm.addCommand(new Command("Exit", Command.EXIT, 0));
        mMainForm.setCommandListener(this);
    }
    protected void destroyApp(boolean arg0) {
    }
    protected void pauseApp() {
    }
    protected void startApp() {
        Display.getDisplay(this).setCurrent(mMainForm);
    }
    public void commandAction(Command arg0, Displayable arg1) {
        notifyDestroyed();
    }
}
```

Java Micro Edition

- Questions?

Android

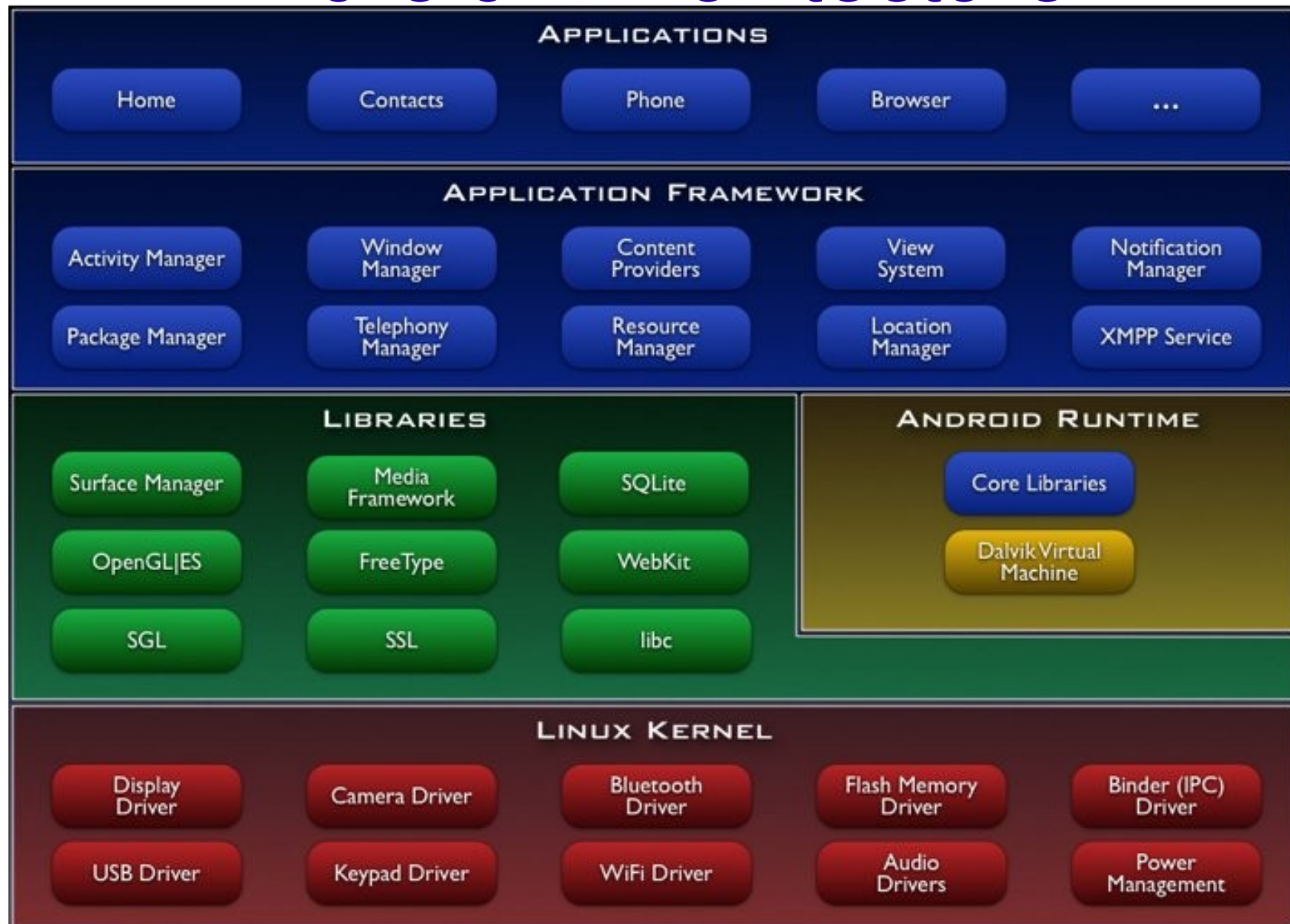
- Software platform and operating system for mobile devices based on the Linux operating system
- Developed by Google and the Open Handset Alliance



Android - Open Handset Alliance

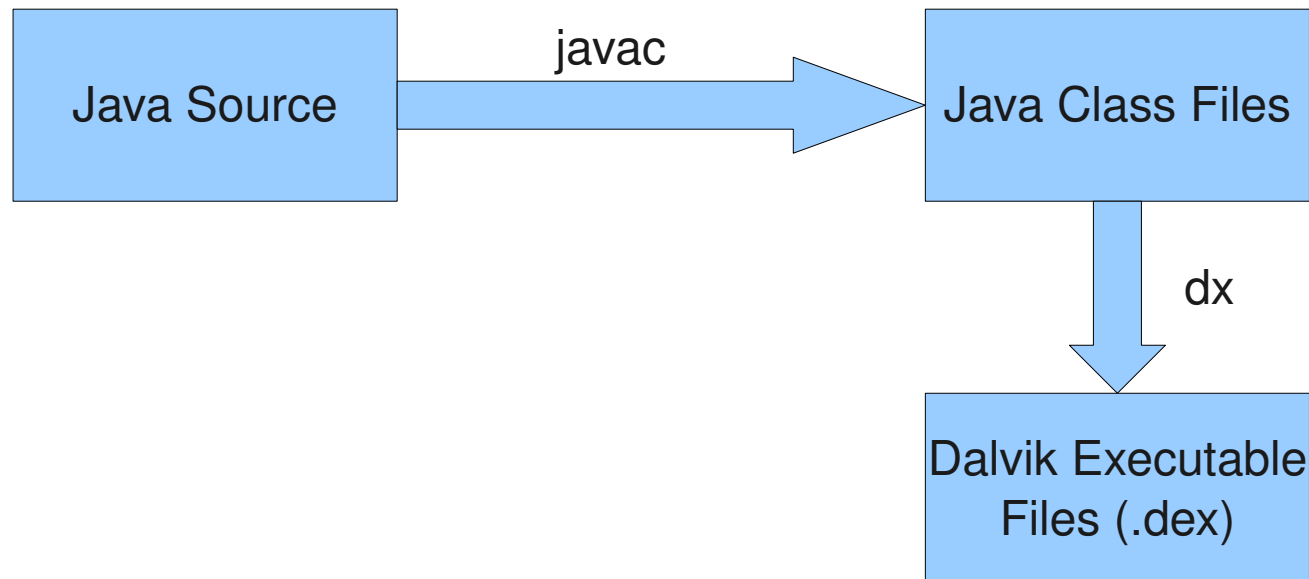
- Vision: “... accelerate innovation in mobile and offer consumers a richer, less expensive, and better mobile experience.”
- Members (more than 30):
 - Google, HTC, NTT DoCoMo, China Mobile, T-Mobile, Motorola, Samsung, Intel ...
 - Swiss members: Esmertec, Noser

Android - Architecture



Android - Runtime

- Dalvik: It's not a Java virtual machine
- Every Android application runs in its own process within its own instance of the Dalvik VM.



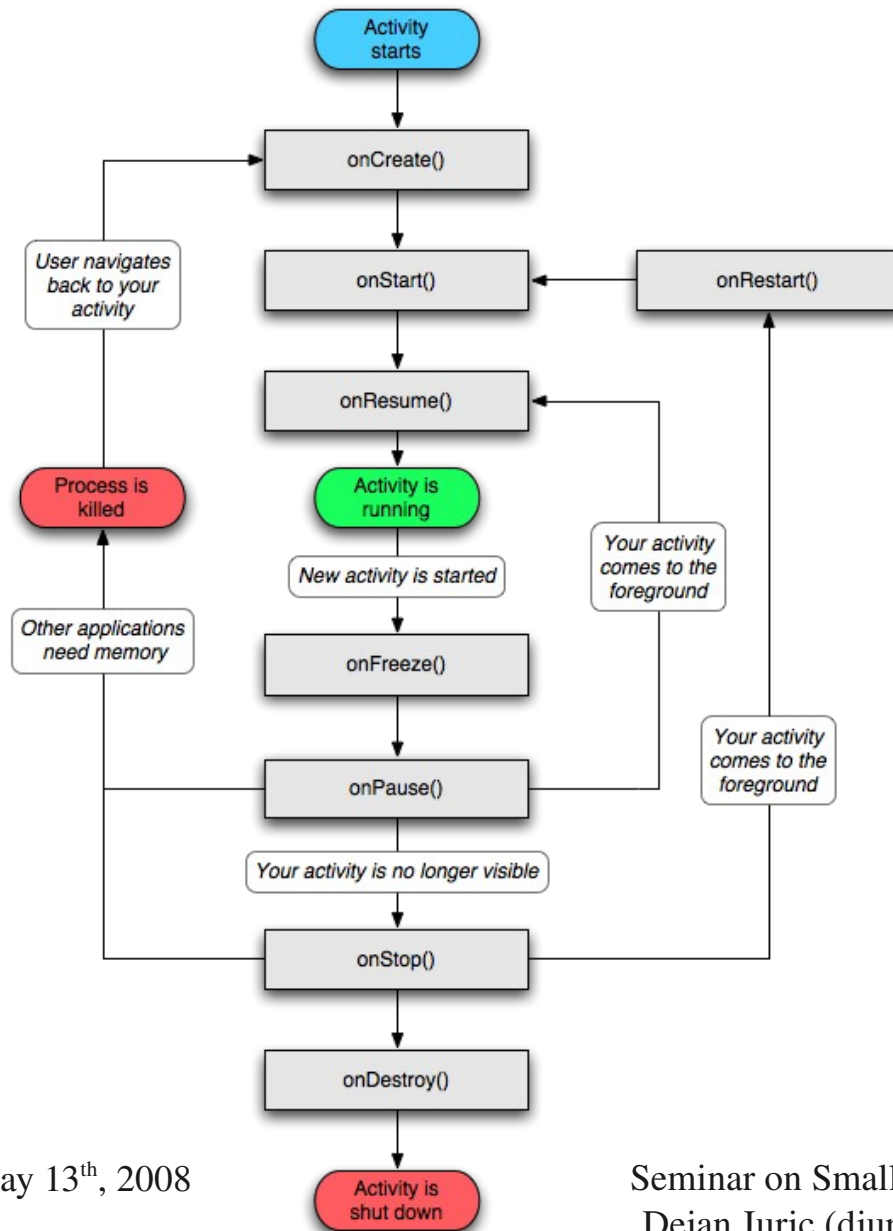
Android – Application Framework

- Views: User Interface
- Content Providers: Access & share data
- Resource Manager: non-code resources (graphics, external strings ...)
- Notification Manager: custom alerts in status bar
- Activity Manager: Manages life cycle of applications

Android – Activity & Intent

- Activity: single, focused thing that the user can do
- Intent: abstract description of an operation to be performed

Android – Life cycle - Activity



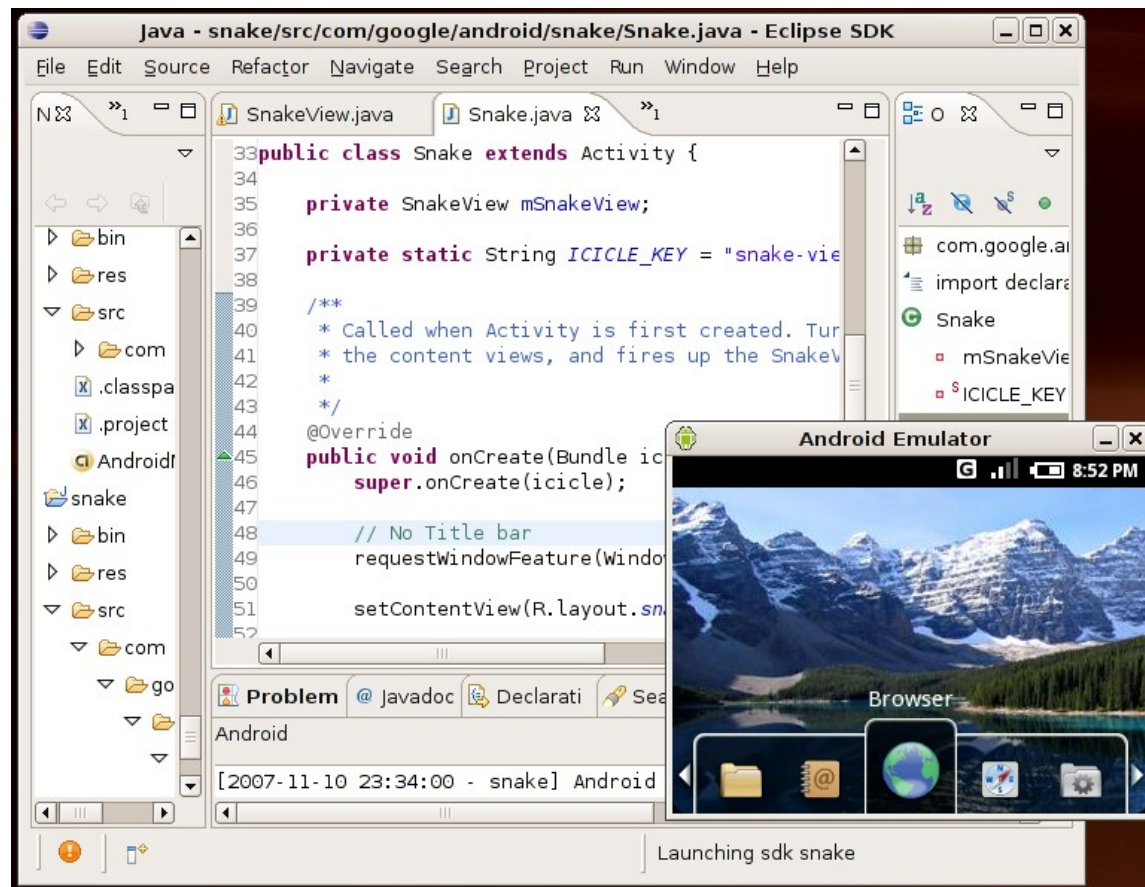
- Activity can be killed!
- We have to store the state

Android - Development

- Software Development Kit:
 - Eclipse Plugin
 - Command line tools
 - Dalvik Debug Monitor Service (ddms)
 - Android Debug Bridge (adb)
 - Android Asset Packaging Tool (aapt)
 - Android Interface Description Language (aidl)
 - Emulator

Android - Demo

- Demonstration with SDK and emulator



JavaME vs Android – Developer's Perspective

- Java ME
 - + Security model
 - + Works on many devices
 - On mobile phones: Integration into system not seamless
 - Too many APIs
- Android
 - + Security model
 - + Optimized for smartphones
 - + Seamless integration into system
 - API: more complicated than Java ME

Java ME vs. Android - General

- Java ME
 - + Lots of devices already available
 - Fully controlled by Sun (Licensing!)
 - Formal specification process (slow)
- Android
 - + Opensource
 - + Defines unique software stack for mobile devices
 - No devices available
 - No standards are used (e.g. VM, IDL ...)

Conclusions

- Rapid growth in the mobile market expected
- Wireless devices getting more powerful
 - More features
 - Better connectivity
- Both technologies could play a big role in the future

Discussion

- Questions?