

Introduction to Mobile Systems

EE1072

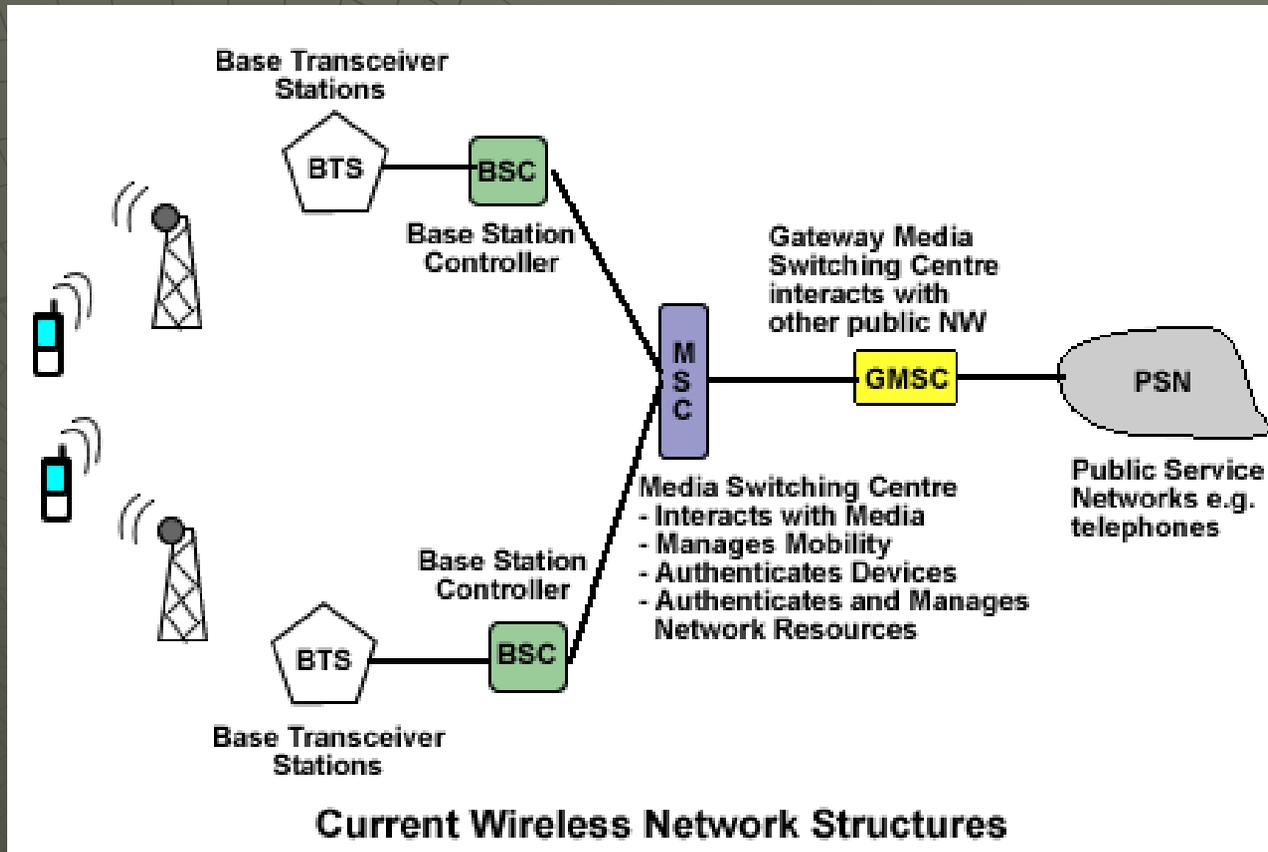
By: Alireza Mousavi

(2)

Topics

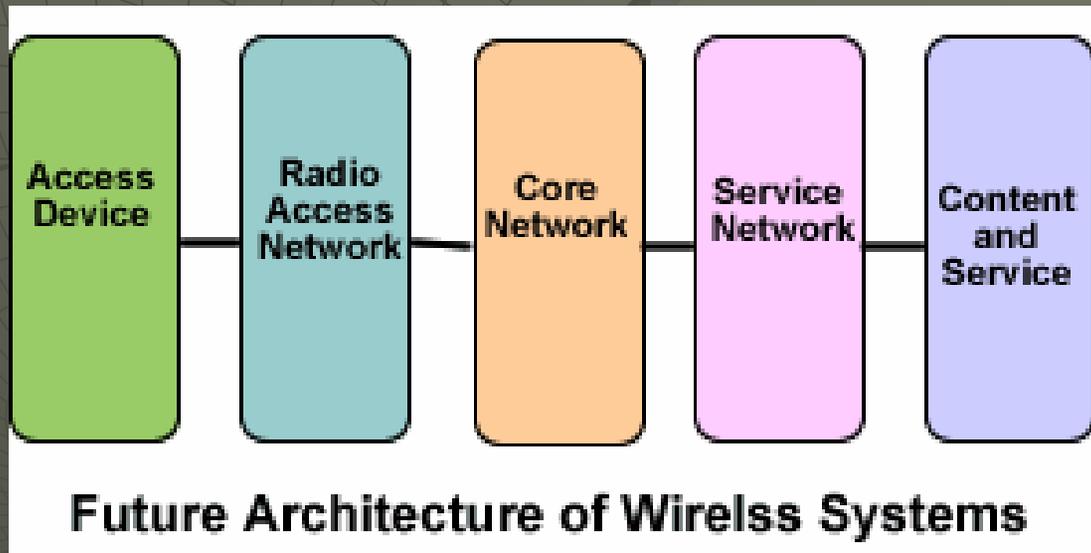
- ◆ How Mobile Systems Work (Present & Future)
- ◆ Introduction to enabling technologies (specifically Java Technology)

Wireless Systems Architecture (Current)



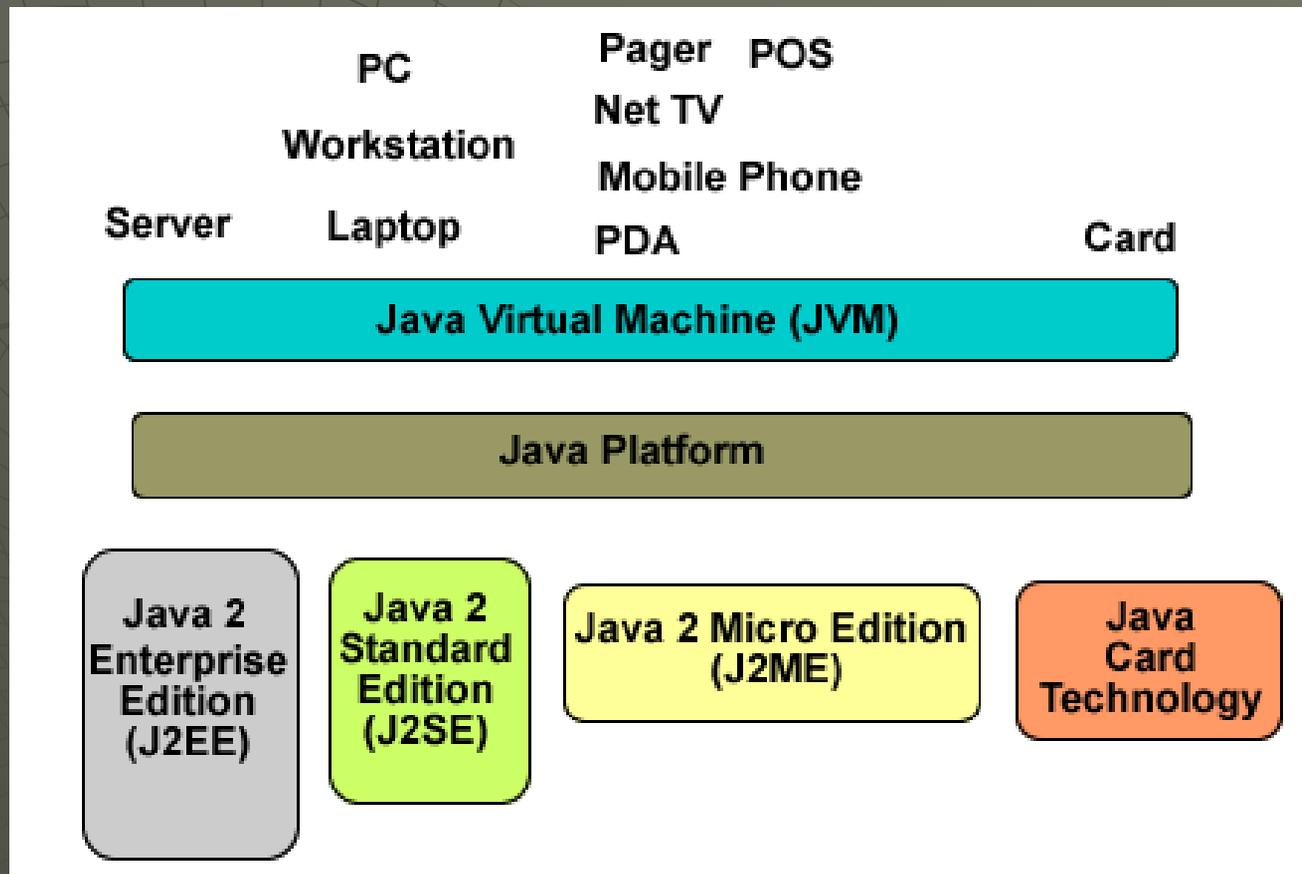
Source: Sun Micro Systems Educational Services

Wireless Systems Architecture (Possible Future Scenario)

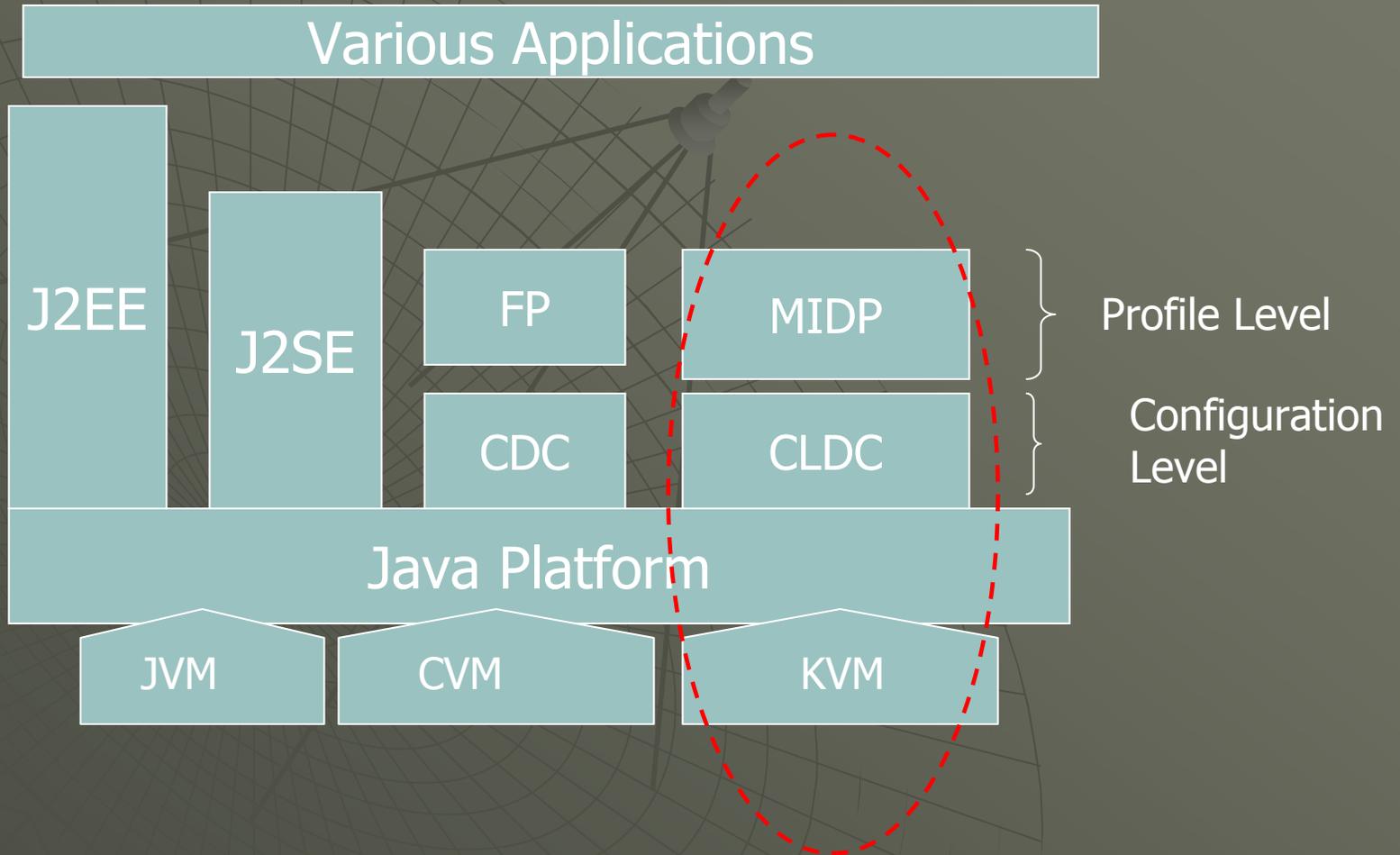


Source: Sun Micro Systems Educational Services

Java™ Platform



JME Schematic



JME Platform

- ◆ Extends java programming language to consumer and embedded devices - new virtual machine (CVM)
- ◆ Categorises devices based on memory capacity, power and display
- ◆ Guarantees smooth integration of device applications into existing J2SE and J2EE platform application

JME Components

◆ *Configurations (CLDC):*

- ◆ defines a minimum Java platform for a family of devices
- ◆ Members of this family all have similar requirements for memory and processing power

◆ *Profiles (MIDP):*

- ◆ specifies application-level interface
- ◆ consists of a set of Java class libraries that provides this application level interface.
- ◆ can theoretically specify all kinds of functionality and services.

Connected Limited Device Configuration (CLDC)

- ◆ Supports PDAs, mobile devices, pagers etc.
- ◆ Defines a standard java Platform for consumer devices
- ◆ Focuses on generality and portability
- ◆ Target devices have 160-512 KB memory available
- ◆ 16-bit or 32-bit processor
- ◆ Power consumption constraints (mainly battery powered)
- ◆ Intermittent network connectivity

Kilo-Byte Virtual Machine (KVM)

- ◆ CLDC defines a VM that is highly portable and designed for resource constrained devices
- ◆ The VM that comes with CLDC reference implementation is called KVM. It is not a full featured JVM

Mobile Information Device Profile (MIDP)

- ◆ Most popular and well-known
- ◆ Sits on top of **CLDC** configuration
- ◆ Defines a set of user interface (UI), networking, local data storage and APIs for current mobile devices
- ◆ A MIDP application is called a **MIDlet**
 - A MIDlet is a java application that uses MIDP and CLDC to be run on a mobile device
 - A MIDlet suite is one or more MIDlets packaged together using a Java Archive (*JAR*) file
- ◆ MIDP commonly exists on mobile phones, PDAs and pagers

JME Platform Layers

Application e.g. MIDlets

MIDP

CLDC

Operating System

Device Hardware

Running a JME application on a device

- ◆ Creating a Run-Time environment (CLDC & MIDP)
- ◆ Application Management Software (AMS)
- ◆ Java Archive File (JAR)
- ◆ Java Descriptor File (JAD)

Application Management Software (AMS)

The **Application Manager** Software is installed on the device and is:

1. Device Dependent (manufacturer specification)
2. Responsible for installing, running and removing MIDlets

Java Archive Files (JAR)

JAR files contain:

1. Java classes

2. Resources (images and application data files)

3. Manifest file (manifest.mf): The system attributes are defined in Manifest files such as:

- ◆ MIDlet Name, MIDlet Version, MIDlet Vendor, MIDlet <n>, Micro-Edition Profile, Micro-Edition Configuration, etc.
- ◆ Compulsory / Optional

Java Descriptor File (JAD)

JAD files may be available as part of the MIDlet suite to provide information about the MIDlets within a *JAR* file.

The rationale behind using *JAD* file:

1. Provides information to the application manager about the contents of *JAR*
2. Provides means for parameters to be passed to a MIDlet without having to make changes to the *JAR*

The steps for developing and testing MIDlets (JME Application)

1. Write the program in Java for the MIDlet
2. Compile the MIDlet source code
3. Pre-verify the MIDlet compiled code – ensure that the code executes safely – it verifies codes before downloading on device
4. Provide attributes that are used during the application installation and execution (create the “manifest.mf” file)
5. Package the MIDlet by creating the *JAR* file containing the application resources and the *JAD* file containing application information. It helps to keep applications organised
6. Execute the application using the emulator

Example

