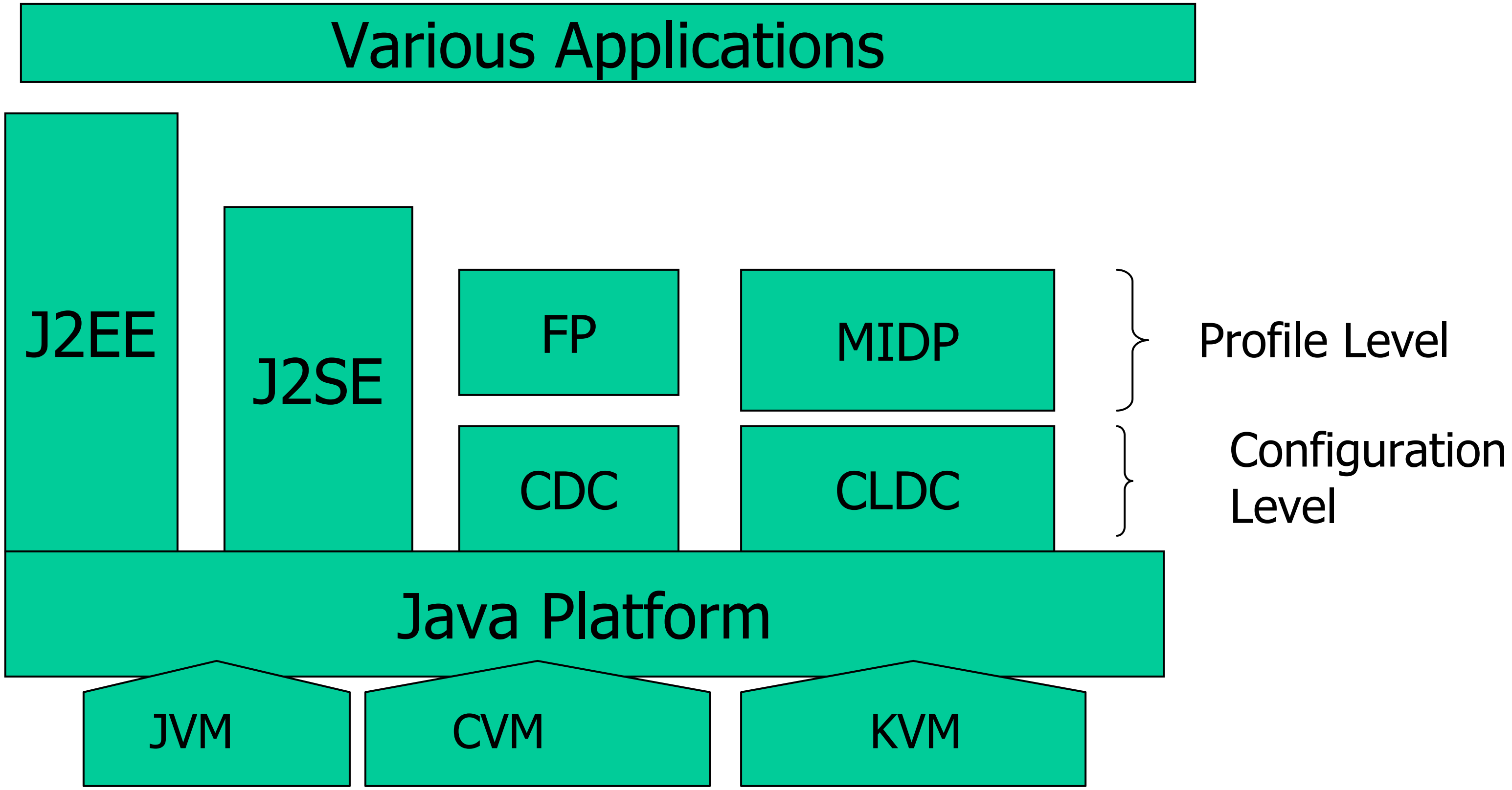
Mobile Information Device Programming (2)

Lecturer: Alireza Mousavi
School of Engineering & Design
www.brunel.ac.uk/~emstaam

JME Platform (Schematic)



JME Platform

What does JME platform offer?

- 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

Building Blocks of JME

The main elements that comprise the JME modular design are:

- *Configurations*

- *Profiles*

“A *configuration* and a *profile* creates JME runtime environment”

JME Configurations

A configuration specifies three basic elements:

- A set of Java programming language features
- A set of Java virtual machine features
- A set of Java libraries and APIs.

JME Two Configurations

The configurations that are currently defined are:

1. Connected Limited Device Configuration(CLDC) for personal, intermittently connected mobile device

2. Connected Device Configuration(CDC) for constantly connected network devices

1. CDC Configuration

- Targets devices with at least 2 MB memory(RAM & ROM) and 32-bit Processor

- Provides a more reliable network connectivity

- Specifies the use of the full Java 2 platform VM, which is called Compact Virtual Machine(CVM)

1. CVM features

- Advanced memory system
- Specified garbage management and collection system
- Full separation of VM from memory system
- CVM offers: portability, fast synchronisation, native thread support, … (see V. Piroumian, 2002)

2. CLDC Configuration

- 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

2. Kilobyte 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

CDC comes with CVM
CLDC comes with KVM

JME Platform Profiles

- Foundation Profile (FP)
- Personal Profile (PP)
- Personal Basis Profile (PBP)
- Mobile Information Device Profile (MIDP)

Foundation Profile (FP)

- Sits on top of ***CDC***
- Provides network capable implementations where UI is not required
- FP commonly exist on embedded devices such manufacturing machines and vehicles

Personal Profile (PP)

- Sits on top of ***Foundation Profile***
- Extends a JME platform implementation by providing a user interface based on J2SE Abstract Window Toolkit (AWT) API
- Its is commonly found on high-end PDAs and game consoles (Source: Sun Micro Systems Educational Services)

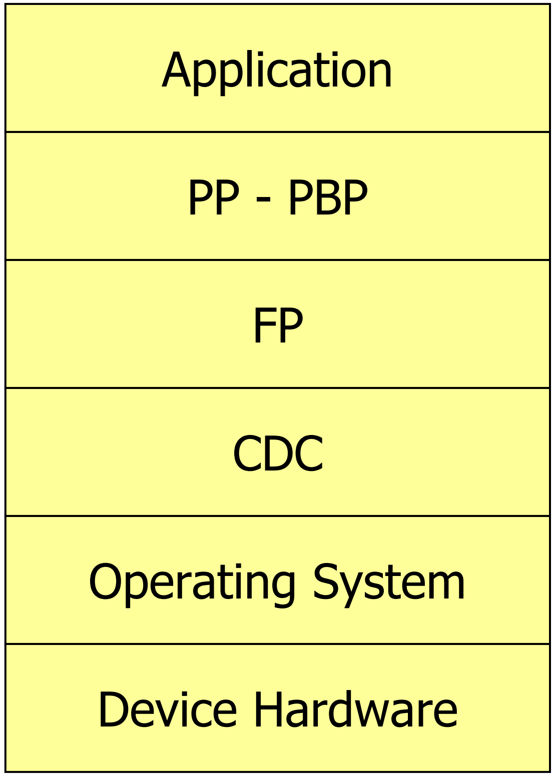
Personal Basis Profile (PBP)

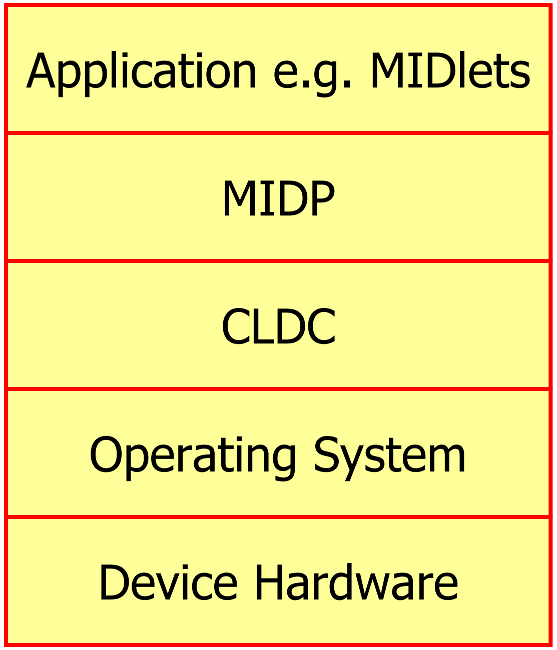
- Works on top of ***FP***
- Provides graphics capabilities
- Found on TV set-top boxes, vehicle systems, etc.

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





Runtime Environment and Application Management

The **Application Manager** is a software on the mobile device that is:

1. Device Dependent (manufacturer specification)

2. Responsible for installing, running and removing MIDlets

Application Manager

When an application Manager runs a MIDlet it provides:

1. Access to CLDC and Java Virtual Machine
2. Access to MIDP class libraries i.e. UI, storage, network support, and user device interaction
3. Access to Java Archive(*JAR*) files
4. Access to Java Descriptor(*JAD*) file

Java Archive(JAR) Files

Jar files contain:

- Java classes
- Resources (images and application data files)
- Manifest file (manifest.mf): The system attributes are defined in Manifest files.

Note: There are six attributes in a manifest file that needs to be properly defined. The application manager will not load the *JAR* without them.

Required Manifest Attributes

Attribute (Required)	Description	Example
MIDlet-Name	Name of the MIDlet suite	MIDlet-Name: MyFirst
MIDlet-Version	MIDlet Version	MIDlet-Version: 2.0
MIDlet-Vendor	Application Developer	MIDlet-Vendor: MIDP LAB
MIDlet-<n>	Reference to a specific MIDlet inside a MIDlet suite. It contains the MIDlet name, icon and class name the application manager will call to load the MIDlet (comma separated)	MIDlet-1: MyFirst, src/images/MyFirst.png, MyFirst
MicroEdition-Profile	The JME profile required to execute the MIDlet	MicroEdition-Profile: MIDP-2.0
MicroEdition Configuration	The JME configuration to execute the MIDlet	MicroEdition-Configuration: CLDC-1.0

Optional Manifest Attributes

Attribute (Optional)	Description	Example
MIDlet-Description	A description of the MIDlet suite	MIDlet-Description: My very first MIDlet
MIDlet-Icon	The name of the PNG file contained by the jar	MIDlet-Icon: MyFirst.png
MIDlet-Info-URL	A URL that contains more info about this MIDlet	MIDlet-Info-URL: http://www.myfirst.ed/index.html
MIDlet-Data-Size	The minimum number of bytes of persistent data that the MIDlet requires	MIDlet-Data-Size: 1500

Java Application Descriptor (*JAD*) file

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*

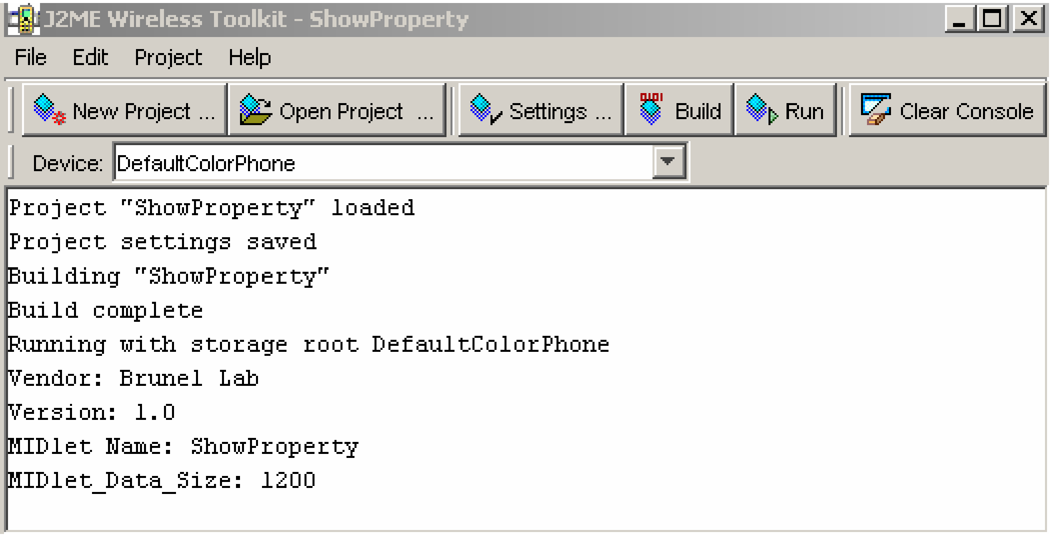
Short Exercise

- Create a manifest.mf file
- Create a *JAD* file
- View attributes declared in manifest and *JAD* file writing a MIDlet.

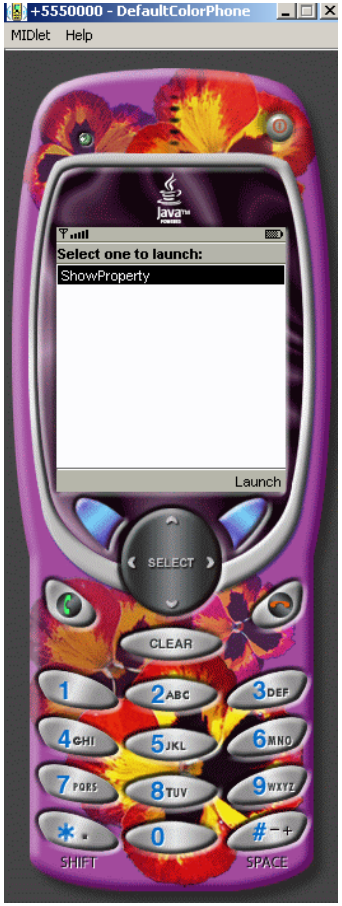
  You need to call the following method:

  *Javax.microedition.midlet.MIDlet.***getAppProperty(**String *Attribute Name)*

Result



```
Project "ShowProperty" loaded
Project settings saved
Building "ShowProperty"
Build complete
Running with storage root DefaultColorPhone
Vendor: Brunel Lab
Version: 1.0
MIDlet Name: ShowProperty
MIDlet_Data_Size: 1200
```



Mobile Services Architecture (MSA)

[Source: Sun Microsystems Inc. (2007), www.sun.com/software

- The MSA platform builds on JME specifications (MIDP, CLDC and Java technology for Wireless Industry)
- Incorporates new technologies such as Bluetooth, vector graphics and Web services
- Its aim is to create a manageable and standardised environment for developers to utilise the latest technologies in mobile and wireless devices

MSA cont.

- The MSA specifications defined in Java Specification Request (JSR) 248 is proposed to handle a larger functionality base for the large number of devices
- It defines the next mobile platform for applications and services

MSA deliverables

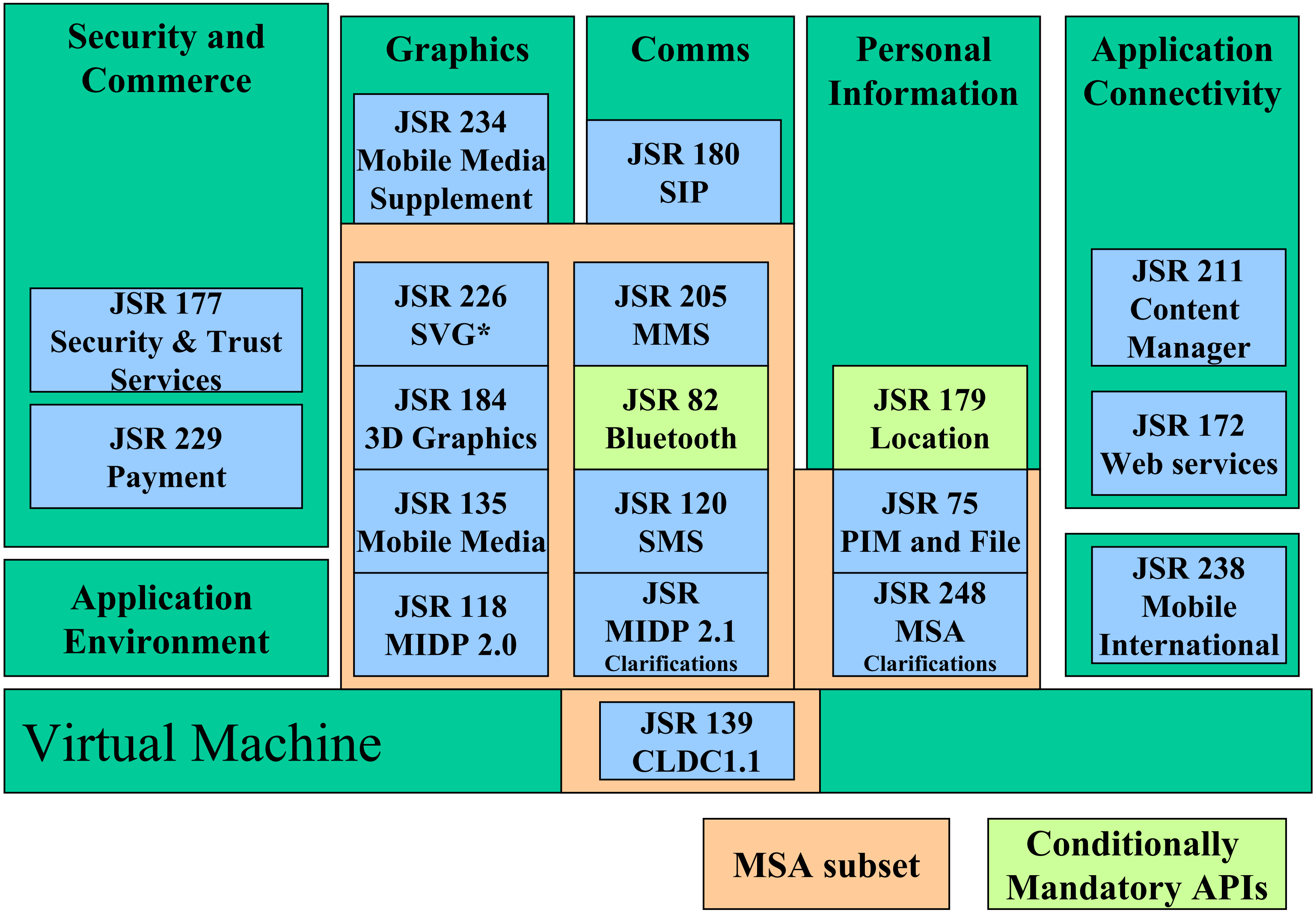
1. A roadmap of handset-related JSRs and their availability in various markets. Different markets historically have different standards.

2. A specification for describing client components and the mobile phone application environment plus a series of JME platform technologies (to enhance end-to-end compatibility)

3. Technology Compatibility Kit (TCK) that verifies compatibility based on specifications defined in MSA for integrated implementation of the technology

MSA – JME platform

[Source: www.sun.com/software]



Security and Commerce

JSR 177 Security & Trust Services

JSR 229 Payment

Application Environment

Graphics

JSR 234 Mobile Media Supplement

JSR 226 SVG*

JSR 184 3D Graphics

JSR 135 Mobile Media

JSR 118 MIDP 2.0

Comms

JSR 180 SIP

JSR 205 MMS

JSR 82 Bluetooth

JSR 120 SMS

JSR MIDP 2.1 Clarifications

Personal Information

JSR 179 Location

JSR 75 PIM and File

JSR 248 MSA Clarifications

Application Connectivity

JSR 211 Content Manager

JSR 172 Web services

JSR 238 Mobile International

Virtual Machine

JSR 139 CLDC1.1

MSA subset

Conditionally Mandatory APIs

* SVG: Scalable 2D Vector Graphics API for JME