

1.3 MOBILE DEVICE PROFILES

What is Java ME?

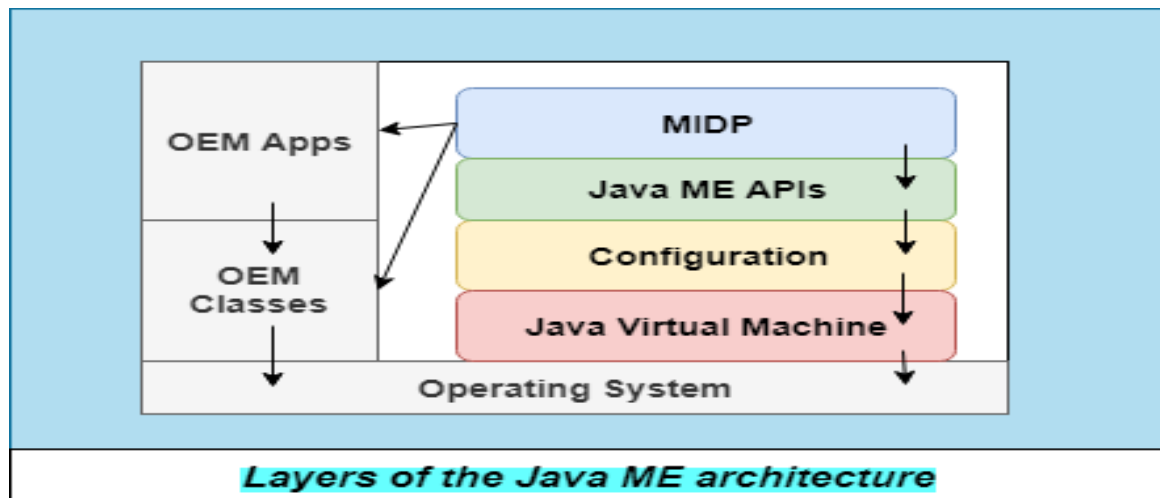
- **The Java ME stands for Java Micro Edition.**
- It is a development and deployment platform of portable code for embedded and mobile devices (sensors, gateways, mobile phones, printers, TV set-top boxes).
- The Java ME has a robust user interface, great security, built-in network protocols, and support for applications that can be downloaded dynamically.
- **Applications which are developed on Java ME are portable** and can run across various devices and can also leverage the native capabilities of the device.

Java ME SDK

- Java ME Software Development Kit (SDK) provides the standalone runtime environment and various utilities required for development Java ME applications.
- It combines the Connected Limited Device Configuration (CLDC) and the Connected Device Configuration (CDC) into one single environment.

JAVA ME ARCHITECTURE

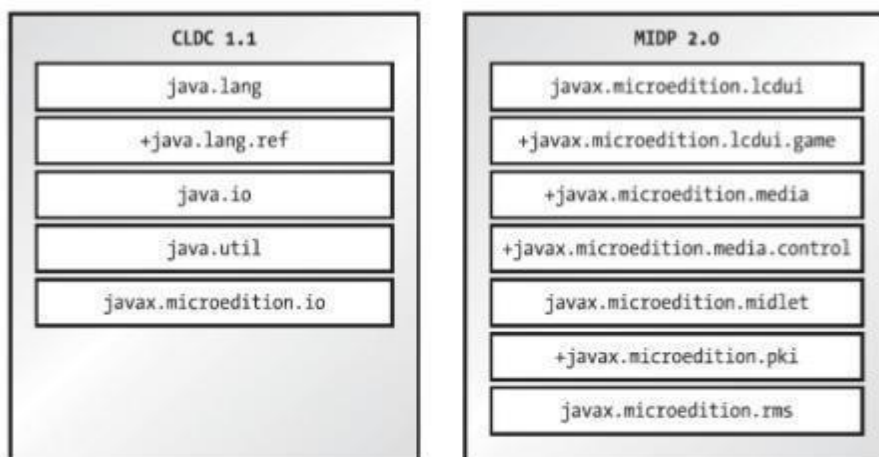
- Java ME does not simply replace the operating system, rather it stacks up layers on the native operating system and makes an environment for the application to run.
 - These layers are collectively named as **Connected Limited Device Configuration (CLDC)**.
1. The first layer is the **configuration layer that includes the Java Virtual Machine**. This layer interacts directly with the native operating system and builds the connection between the profile and the JVM.
 2. The **second layer is the profile which contains the minimum set of APIs for the small computing device**. The profile contains a set of classes which are made to implement the features of a related group of small computing devices.
 3. The **third layer is the Mobile Information Device Profile (MIDP)**. The **MIDP layer consists of APIs which are for user network connections, persistence storage, and the user interface**. It also has access to Connected Language Device Configuration (CLDC) and Mobile Information Device Profile (MIDP) libraries.
 - **A small computing device has two components supplied by the Original Equipment Manufacturer (OEM). They are namely OEM apps and OEM classes.**
 - The MIDP communicates with the OEM classes to gain access to features like sending and receiving messages and accessing device-specific persistent data.
 - **OEM applications are small programs such as address book etc.**



MIDP

- **Mobile Information Device Profile (MIDP) is a specification for the use of Java technology for mobile devices.**
- MIDP sits on top of the **Connected Limited Device Configuration (CLDC)**.
- Because MIDP is primarily used with CLDC, which is designed for **highly constrained devices with limited CPUs, screen size, RAM, battery power and user interface, midlets are ideal for low-end cell phones.**
- **MIDP is part of the Java™ 2 Platform, Micro Edition (J2ME™).**
- An application that runs in the **MIDP** environment is called a **MIDlet**.
- **The first MIDP devices were launched in April 2001.**

Anatomy of MIDP Apps



GENERAL API

- The core [application programming interfaces](#) are defined by the underlying [Connected Limited Device Configuration](#) system.
- **javax.microedition.io** Contains the **Java ME-specific classes used for I/O operations.**
- **javax.microedition.lcdui** Contains the **Java ME-specific classes used for the GUI.**
- LCDUI has a simple screen based approach where a single Displayable is always active at a time in the application user interface. **LCDUI API provides a small set of displayable common in mobile device user interfaces: List, Alert, Textbox, Form and Canvas.**
- LCDUI also has a **quite unique approach of abstract operations, called Commands.** The placement of commands added to a displayable is completely up to the device implementation of this toolkit. The application programmer uses API specified command types to indicate the usage or purpose of the command in an application user interface. Common types are BACK, EXIT, ITEM, SCREEN.
- LCDUI **Limited Capability Device User Interface**
- **javax.microedition.rms** The **Record Management System provides a form of persistent storage for Java ME; a database for the mobile device.**
- **javax.microedition.midlet** Contains the **base classes for Java ME applications, and allows applications to be notified of changes to their state.**

OPTIONAL JSR

javax.microedition.messaging Wireless messaging API (optional), for sending SMS and MMS messages.

javax.microedition.pim Personal information management API (optional), access the device's Address Book, to-do List, Calendar.

javax.microedition.io.file The File Connection Optional Package (FCOP) is one of two optional packages defined by JSR 75 through the Java Community Process. The FileConnection API specified in JSR 75 gives access to the local file systems on devices like PDA. In order to overcome security issues MIDlet needs to include requested file permission in its JAD file under MIDlet-Permission property.

Development tools There are several different ways to create MIDP applications: code can be written in a plain [text editor](#), or one can use a more advanced [IDE](#) such as [NetBeans](#), [IntelliJ](#) , or [Eclipse](#)

HISTORY

- MIDP was developed under the Java Community Process.
- MIDP 1.0 (JSR 37) - Approved on September 19, 2000
- MIDP 2.0 (JSR 118) - Approved on November 20, 2002
- MIDP 3.0 (JSR 271) - Approved on December 9, 2009
 - A. Shared libraries for midlets
 - B. Improved cross-device interoperability
 - C. Improved UI expressability and extensibility
 - D. Support of devices with larger displays
 - E. High-performance games
 - F. Auto-launched midlets
 - G. Inter-midlet communications

MIDP is now succeeded by MEEP as of Java ME 8

MID Profile Architecture

