Introduction

Agenda

- Introduction of Mobile Technology
- Mobile Phone History
- Software Technology for Mobile Device
- J2ME Overview
  - CDC
  - CLDC
  - MIDP
- J2ME Resources

Mobile Technology Overview

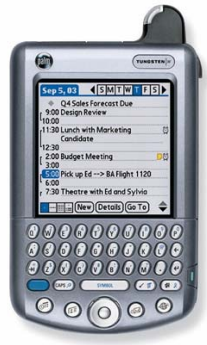
- The Goals of Mobile Technology
  - Connecting people
  - Information sharing
  - Internet access
  - Entertainment

  key phrase:

  “communication any time, anywhere”

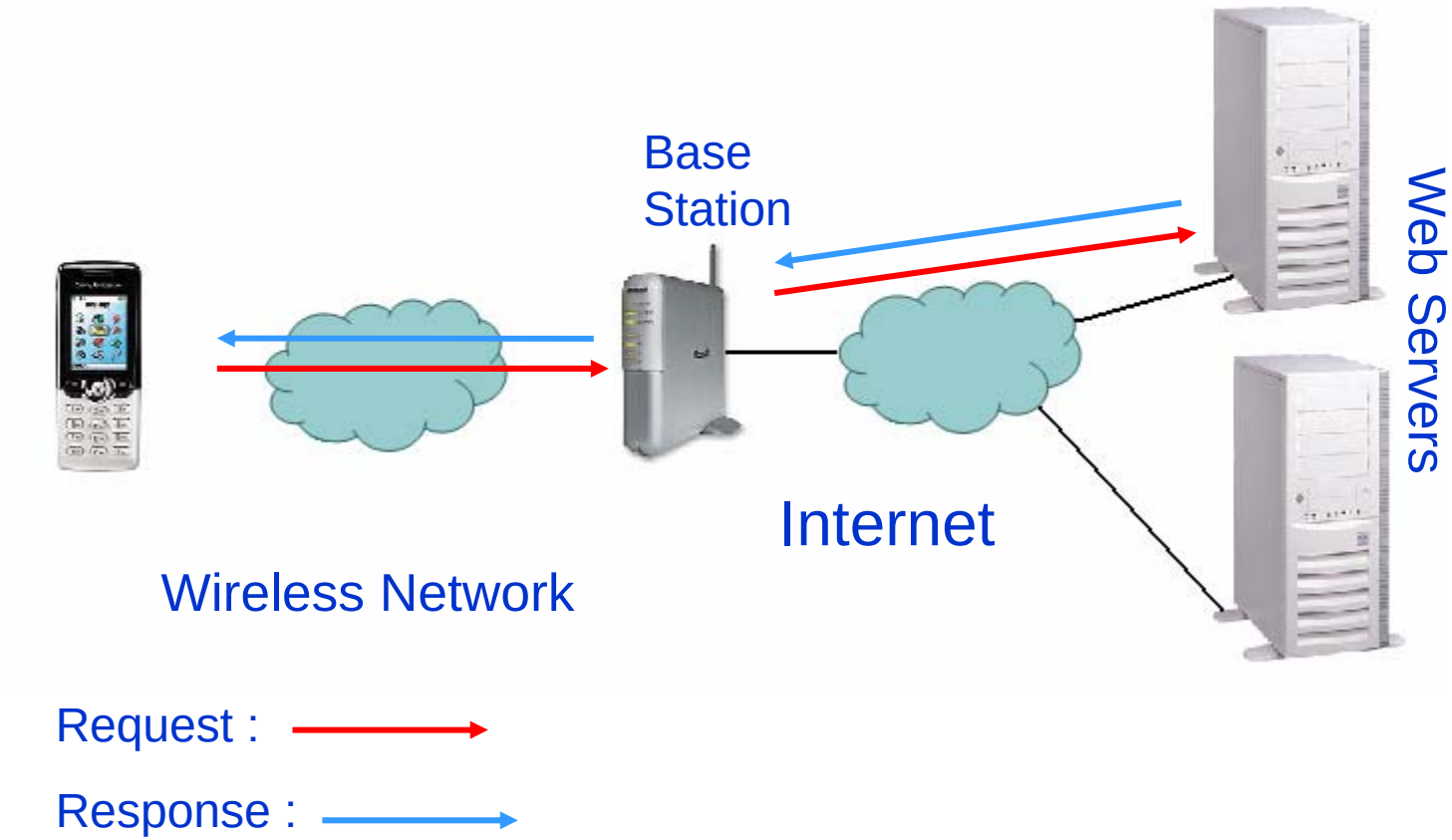
Mobile Technology

- Includes
  - Notebook
  - Palmtops
  - PDAs
  - Mobile Phones
  - Tablet PCs
  - And more ...



Continued. . .

- The Internet Scenario for retrieving information in a wireless network



Continued. . .

- Mobile Networking
  - GSM, GPRS, EDGE
  - 1G, 2G, 2.5G, 2.75G, 3G and 4G
  - IEEE802.11
  - InfraRed and Bluetooth

Current Technology

- GSM
  - Short for Global System for Mobile Communications, one of the leading digital cellular systems. Data speed for GSM is 9.6 kilobits.
- EDGE
  - Enhanced Data GSM Environment. EDGE is a faster version of GSM wireless service. EDGE enables data to be delivered at rates up to 384 Kbps on a broadband.

Current Technology

- GPRS
  - Short for General Packet Radio Service, a standard for wireless communications which runs at speeds up to 115 kilobits per second.

Evolution

- First Generation (1G)
  - Analog system designed for voice only communication. 1G systems are almost extinct now,
- Second Generation (2G)
  - Use GSM and IS-95 CDMA technologies
  - CDMA
    - Allows users to communicate with different codes
  - Still designed for voice communication

Evolution

- 2.5 and 2.75 Generation
  - GPRS and CDMA2000 (Phase 1) are belonged to 2.5 G
  - EDGE is belonged to 2.75G
  - As higher data rate is provided, allows some data transmission

Evolution

- Third Generation (3G)
  - Two 3G, UMTS and CDMA-2000, are used. UMTS is broadly deployed in Europe and CDMA-2000 is being deployed in North American and parts in Asia
  - Higher data transmission rate (up to 2Mbps) which allows video conferencing

Evolution

- Forth Generation (4G)
  - Combined the technologies of Wireless local area network (will be introduced soon) and 3G

Evolution

- IEEE802.11
  - Wireless Local Area Network
  - 802.11a, 802.11b and 802.11g
  - Allows 54Mbps data transmission speed (even 108Mbps is provided in the market)
  - Used for PDA, Notebook or even desktop

Facilities

- InfraRed
  - Short range data transmission technology
  - Normally used as remote controller
  - Data rate is around tens kbps
  - Line of sight transmission, hence the space between two InfraRed Devices should be clear

Facilities

- Bluetooth
  - Commonly used for hand-free earphone
  - Short range data transmission – around 10 meters
  - Data rate is up to 1Mbps
  - Do not have light of sight requirement

Mobile Phone History


Michael Faraday
1791 to 1867

- In1843
  - Michael Faraday a talented chemist begins researching the possibility that space can conduct electricity. His research starts the wheels turning for many other 19th century scientists. At the time, many of them were referred to as “crackpots”.
- In 1865
  - A Virginia Dentist/Scientist, Dr. Mahlon Loomis, develops a method of communicating through the earth’s atmosphere by using an electrical conductor. He does this by flying two kites, that are rigged with copper screens and wires, which are connected to the ground on two separate mountains about 18 miles apart. He later received a grant from the U.S. Congress for $50,000. (A fairly large chunk of change for 1865)

Mobile Phone History

- In 1866
  - The first trans-Atlantic telegraph is built (not much to do with cell phones, but a major advancement in communication nonetheless)
- In 1921
  - The Police Department in Detroit, Mich. begins installing mobile radios, operating around 2 MHz, in their squad cars. They encounter many problems such as overcrowding on the channels and terrible interference.

Mobile Phone History

- In 1934
  - The U.S. Congress creates the Federal Communications Commission. They decide who gets to use certain radio frequencies. Most channels are reserved for emergency use and for the government. Radio is still a baby.
- In 1940's
  - The mobile radios are able to operate at 30 to 40 MHz and become much more common between police departments, and the wealthy. Several private companies and organizations begin using these same radios for personal gain.

Mobile Phone History

- In1945
  - The first mobile-radio-telephone service is established in St. Louis, Miss. The system is comprised of six channels that add up to 150 MHz. The project is approved by the FCC, but due to massive interference, the equipment barely works.
- In 1947
  - AT&T comes out with the first radio-car-phones that can be used only on the highway between New York and Boston; they are known as push-to-talk phones. The system operates at frequencies of about 35 to 44 MHz, but once again there is a massive amount of interference in the system. AT&T declares the project a failure.

Mobile Phone History

- In 1949
  - The FCC authorizes the widespread use of many separate radio channels to other carriers. They are know as Radio Common Carriers (RCC) and are the first link between mobile phones and the telephone, rather than just radio to radio. The RCCs are the first step toward the cellular phone industry, which were designed more for profit than for the general public.
- In 1956
  - The first real car phones, not car radios, come into play across the United States. Although, the system is still using push-to-talk phones, it is an improved version that actually works. However, the units are big and bulky, and require a personal radio operator to switch the calls. A similar system appeared in Sweden a few years earlier.

Mobile Phone History

- In 1964
  - A new operating system is developed that operates on a single channel at 150 MHz. In essence, this removes the need for push-to-talk operators. Now customers can dial phone numbers directly from their cars. RCCs are finally taken seriously by the FCC as legitimate competitors to the land-line phone companies.
- In 1969
  - The self-dialing capability is now upgraded to 450 MHz and becomes standard in the United States. This new service is known as (IMTS) Improved Mobile Telephone Service.

Mobile Phone History

- In 1970
  - Cell Phone lobbyists finally win with the FCC and get a window of 75 MHz in the 800 MHz region, which is allocated specifically for cell phones. The FCC realizes the potential of the industry and can't ignore it any longer.
- In 1971
  - AT&T is the first company to propose a modern-day mobile-phone system to the FCC. It involves dividing cities into "cells". It is the first company to experiment on this.

Mobile Phone History

- In 1973
  - Dr. Martin Cooper invents the first personal handset while working for Motorola. He takes his new invention, the Motorola DynaTac., to New York City and shows it to the public. His is credited with being the first person to make a call on a portable mobile-phone.
- In 1974
  - The FCC actually starts to encourage cell phone companies to push forward the “cellular idea”. But unfortunately a law suit arises with Western Electric, who is the closest company to succeed at the time, and it rules that they are not allowed to manufacture terminal and network phone systems under the same roof. This is an effort to prevent a monopoly. But it also prevented progress.

Mobile Phone History

- In 1975
  - AT&T adapts its own cellular plan for the city of Chicago, but the FCC is still uneasy about putting the plan into action.

- In 1977
  - Finally cell phone testing is permitted by the FCC in Chicago. The Bell Telephone Company gets the license; they are in a partnership with AT&T which was then, a collaborative effort to battle the stubborn FCC.

Mobile Phone History

- In 1981
  - The FCC makes firm rules about the growing cell phone industry in dealing with manufactures. It finally ruled that Western Electric can manufacture products for both cellular and terminal use. (Basically they admitted that they have put the phone companies about 7 years behind)
- In 1988
  - One of the most important years in cell phone evolution. The Cellular Technology Industry Association is created and helps to make the industry into an empire. One of its biggest contributions is when it helped create TDMA phone technology, the most evolved cell phone then. It became available to the public in 1991.

Mobile Phone History

- 1991-2000
  - Marked the appearances of giant cell phone companies like NOKIA and Ericsson. SDMA, GPS, GSM technologies sprouted. This was considered the “golden decade of cellular phone technology”
- In 2001
  - BellSouth announces that it is leaving the pay phone business because there is too much competition from cell phones.

Mobile Phone Future

- 2007 and beyond . . .

  A UE CCSS Graduate developed the first all terrain cell phone. The device can be used undersea with depths of 500m, and can still sense signals through 300m of earth (caves) ☺

Software Technology for Mobile Device

- There are many platforms around mobile applications development
  - Windows Mobile
  - Symbian
  - Embedded Visual C++ and .Net
  - J2ME

Software Technology for Mobile Device

- Windows Mobile
  - Windows Mobile is Microsoft's software platform for Pocket PCs and Smartphones. Windows Mobile extends the familiarity of the Windows desktop to personal devices.
  - Windows Powered mobile devices include Pocket PC, Pocket PC Phone Edition, Smartphone, and Handheld PC

Software Technology for Mobile Device

- Symbian
  - Symbian is like an Operation System on mobile phone. It provides the required low level functions for application development

symbian

Symbian OS - the mobile operating system

Software Technology for Mobile Device

- Embedded Visual C++ 4.0 and .Net
  - Programming language and related libraries that allows programmer to develop applications on Windows Mobile platform

Software Technology for Mobile Device

- J2ME (Java 2, MicroEdition)
  - Normally used for less memory and low processing power devices
  - A collection of packages and classes for application development on mobile devices
- We will use J2ME to develop applications in this course

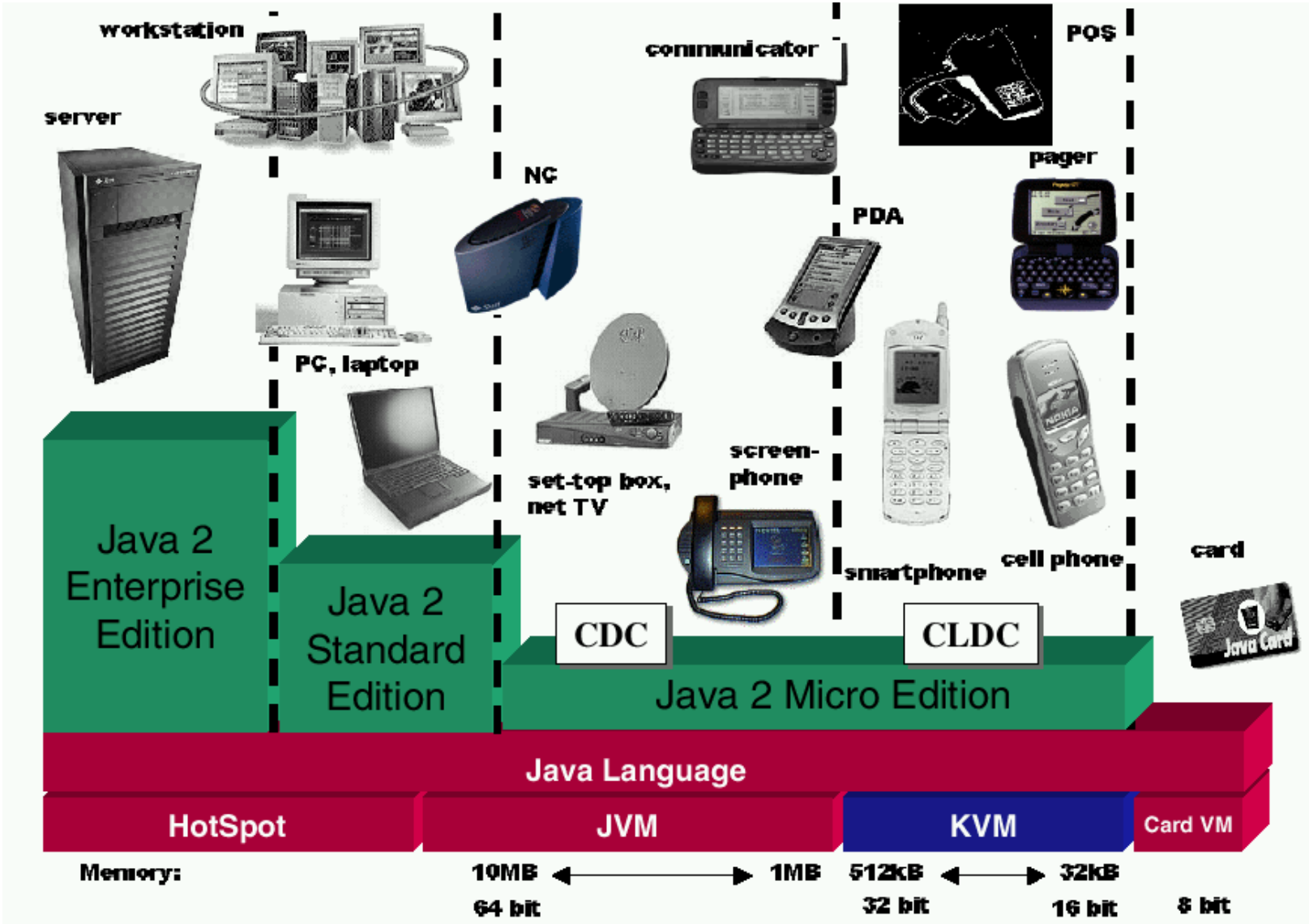
J2ME Overview

- Java
  - A programming language developed by Sun Microsystems
  - Required a Virtual machine to interpret the source codes and generate bytecodes
  - Syntax is similar to C++
  - Platform independent feature

J2ME Overview

- Java includes three different editions
  - J2SE (Java 2 Standard Edition)
  - J2EE (Java 2 Enterprise Edition)
  - J2ME (Java 2 Micro Edition)
- The above three editions target different devices or systems

J2ME Overview



J2ME Overview

- J2SE
  - Provides a complete environment for applications development on desktops and servers
  - The foundation of J2EE
  - J2SE 1.5 (Tiger) is available now!

J2ME Overview

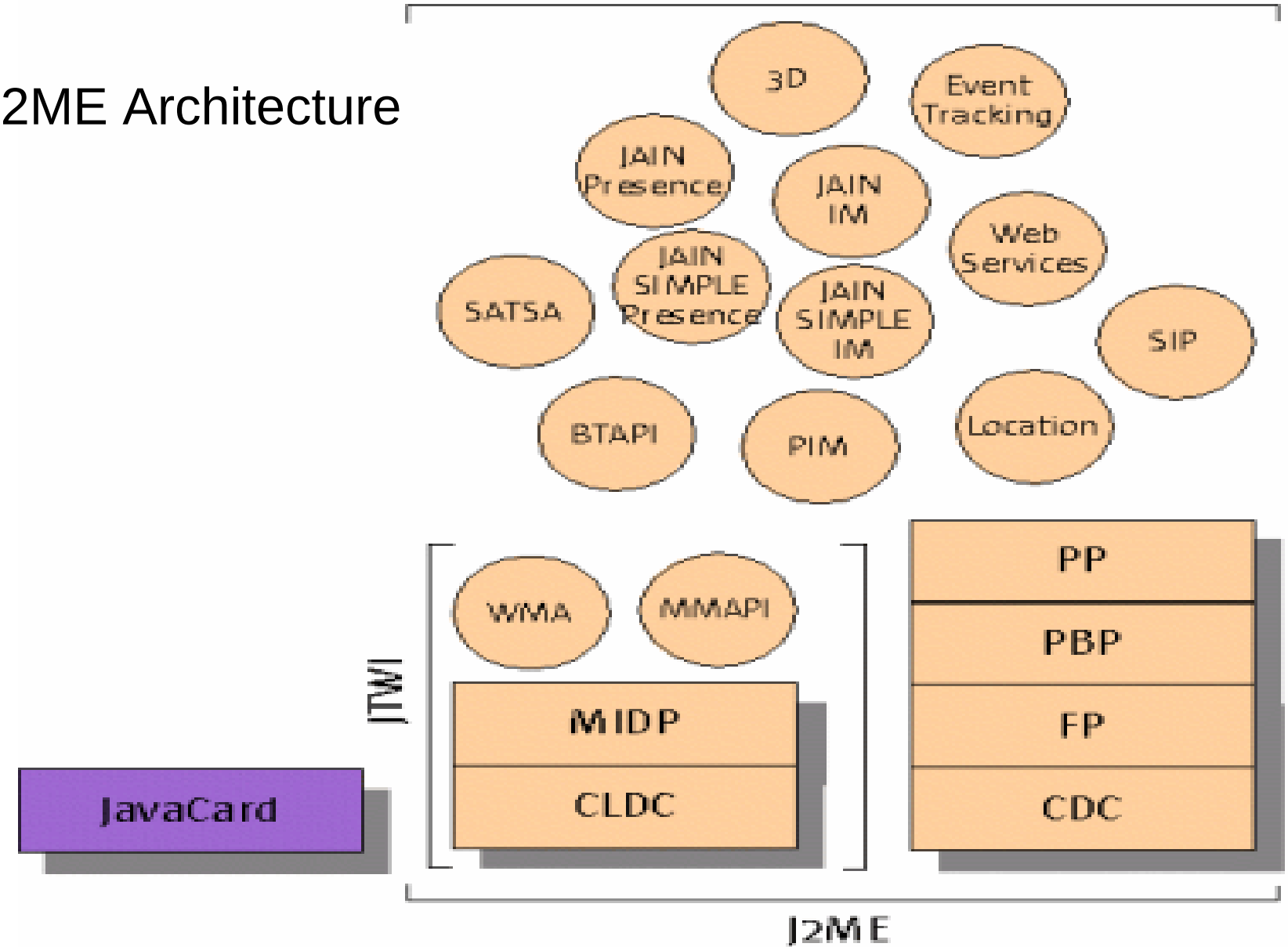
- J2EE
  - Targeted for business use
  - Large scale systems which may contain hundreds of servers and millions of users
  - Web based services
  - High performance machines

J2ME Overview

- J2ME
  - The Micro Edition of the Java 2 Platform provides an application environment that specifically addresses the needs of commodities in the vast and rapidly growing consumer and embedded space, including mobile phones, pagers, PDAs, set-top boxes, and vehicle telematics systems

J2ME Overview

J2ME Architecture



J2ME Overview

- J2ME architecture is divided in to four different level
  - KVM (Kilobyte Virtual Machine)
  - Configurations
  - Profiles
  - Optional packages

J2ME Overview

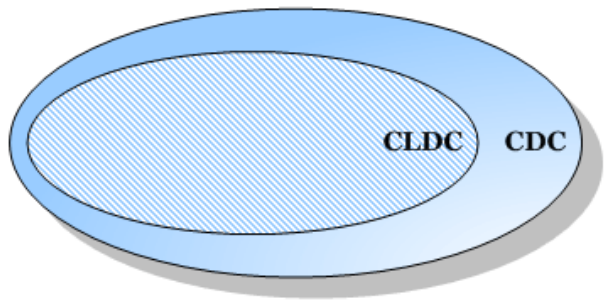
- KVM (Kilobyte Virtual Machine)
  - As the name implies, it is used for small program
  - A subset of JVM
  - A pool for running java code on the device

J2ME Overview

- Configurations
  - defines a basic, lowest-common-denominator J2ME runtime environment
  - includes the virtual machine and a set of core classes derived primarily from J2SE

J2ME Overview

- There are two different configurations
  - Connected Device Configuration (CDC)
  - Connected Limited Device Configuration (CLDC)



Relation between CLDC and CDC

J2ME Overview - CDC

- Targeted for devices that have
  - 2 MB or more total available memory
  - Memory dedicated to J2ME environment
  - More than 2MB ROM/Flash
  - More than 512 KB RAM
  - Network connectivity
- Full Java 2 Virtual Machine specification

J2ME Overview - CDC

- CDC uses
  - Wireless communicators
  - High-end PDAs
  - TV set-top boxes
  - Gateways
  - Automotive entertainment and navigation systems
  - Telecomm/Networking Equipment
  - Industrial Controllers

J2ME Overview - CDC

- Full-featured Java 2 platform runs on small devices
  - Sharp Zaurus can run J2ME CDC

J2ME Overview - CDC

- Three profiles build on CDC
  - Foundation Profiles (FP)
    - Provides core Java functionalities, such as input/output streams, security, networking
  - Personal Profiles (PP)
    - Support Java Graphical User Interface
  - Personal Basic Profiles (PBP)
    - A smaller set of Personal Profiles
- We will stop at CDC here and focus on CLDC

J2ME Overview - CLDC

- Targeted at devices with:
  - 160KB to 512KB total memory available for Java technology
  - Limited power (battery), connectivity (often intermittent), UI (small screen)

J2ME Overview - CLDC

- CLDC scope

- Core java.* libraries
- Additional I/O and
- networking libs
- Security features
- Internationalization

J2ME Overview - CLDC

- Classes extended from Java 2 Platform, Standard Edition (J2SE) are in packages:
  - java.lang.*
    - Contains the basic Mathematics classes and data types
  - java.util.*
    - Contains some commonly used functionalities like Random and Vector (store a set of values)
  - java.io.*
    - Contains the data types for Input/output data
- New classes introduced by CLDC are in package:
  - javax.microedition.*

J2ME Overview - CLDC

- But. . . CLDC is not enough for building applications
- CLDC is the base for building the above profiles
- It separates the machine dependent issues from the applications

J2ME Overview - CLDC

- There are two version of CLDC
  - CLDC 1.0
  - CLDC 1.1
- Today, most mobile phones only support 1.0 version
- Main differences between 1.0 and 1.1 are
  - Floating point is added in 1.1
  - More Date related classes is added in 1.1
  - The minimum memory budget has been raised from 160 to 192 kilobytes
  - And more …

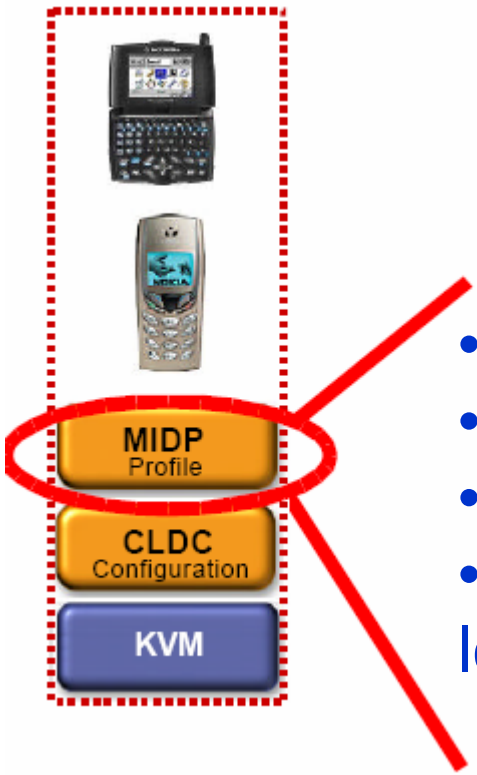
    [Reference: http://developers.sun.com/techtopics/mobility/getstart/articles/survey/]

J2ME Overview - MIDP

- Targets mobile two-way devices implementing J2ME CLDC
- Profile addresses
  - Display toolkit, User input methods
  - Persistent data storage
  - HTTP 1.1-based networking using CLDC Generic Connection framework

J2ME Overview - MIDP

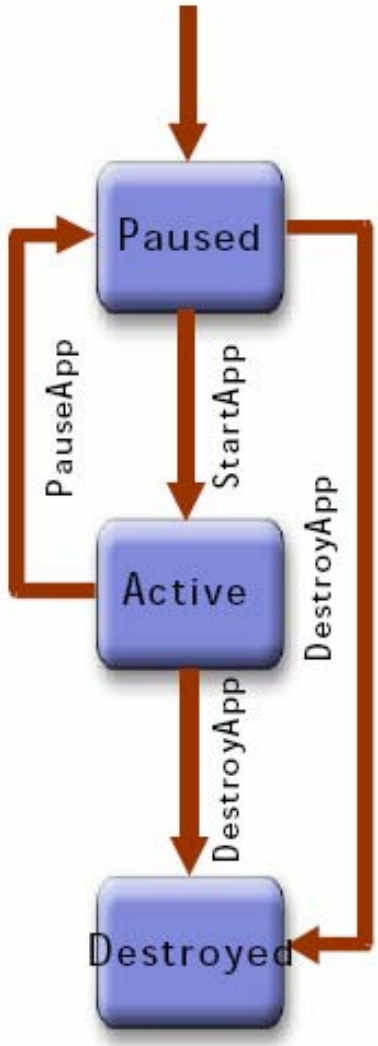
- Scope



- Application model
- Persistent storage (RMS APIs)
- Networking (HTTP, etc.)
- User interface (High and low level APIs)

J2ME Overview - MIDP

- MIDP Application Lifecycle
  - MIDP applications, or “MIDlets”, move from state to state in their lifecycle according to a state diagram
  - Paused – initialized and waiting
  - Active – has resources and is executing
  - Destroyed – has released all resources, destroyed threads, and ended all activity



J2ME Overview - MIDP

- MIDlet Packaging
  - MIDlets are packaged in a JAR (like a zip file) file including Class files of the MIDlet(s)
  - Resource files
  - Manifest with application properties
- Application Descriptors (JAD files) accompany MIDlet JARs and provide deployment information (name, version, size, etc.)

J2ME Overview - MIDP

- MIDlet Development Steps
  - Write your Java application
  - Compile it
  - Preverify it
  - Package it into a JAR file
  - Create the application descriptor
  - Deploy and run your application in the
- J2ME Wireless Toolkit or your device of choice

J2ME Overview - MIDP

- There are two version of MIDP: 1.0 and 2.0, we will focus on 2.0 because
  - Supported by most mobile phone
  - Enhanced UI
  - Game and Sound APIs
  - New Security Model through Signed MIDlets
  - Enhanced Networking, including HTTPS

J2ME Overview - MIDP

- There are many functionality supported by MIDP
  - Graphical user interface
  - Media
  - Networking
  - Security

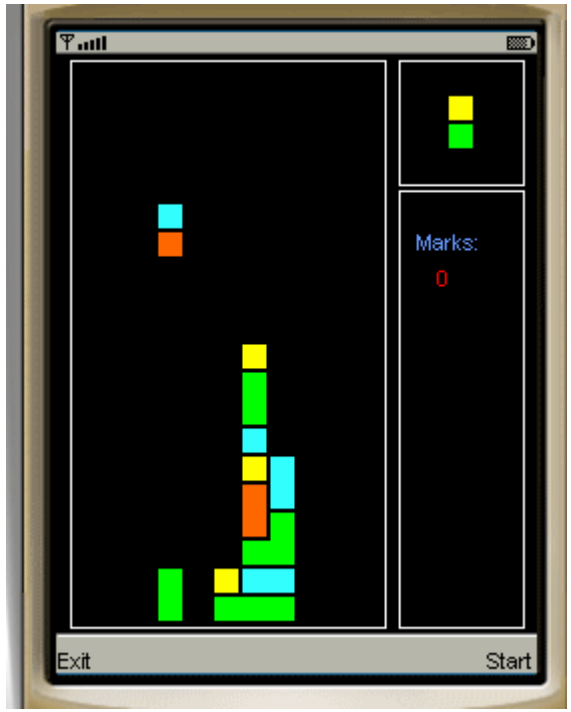
J2ME Overview - MIDP

- Graphical user interface includes
  - Included in javax.microedition.lcdui.*
  - “Form” which conatins Basic items :
    - ChoiceGroup -A ChoiceGroup is a group of selectable elements intended to be placed within a Form
    - ImageItem –Hold an image for display
    - StringItem – display text
    - And some others like CustomItem, Gauge, DateField …
    - Form is used for simple screen output and text input

A simple calendar application developed using Form

J2ME Overview - MIDP

- Canvas which contains Graphics
  - Draw images and strings
  - Draw rectangles, lines and arcs
  - Set the color used
  - Canvas also allows you to get “key” input from user
- Form is used for some simple applications which only have text input
- Canvas is used for more interactive applications like games



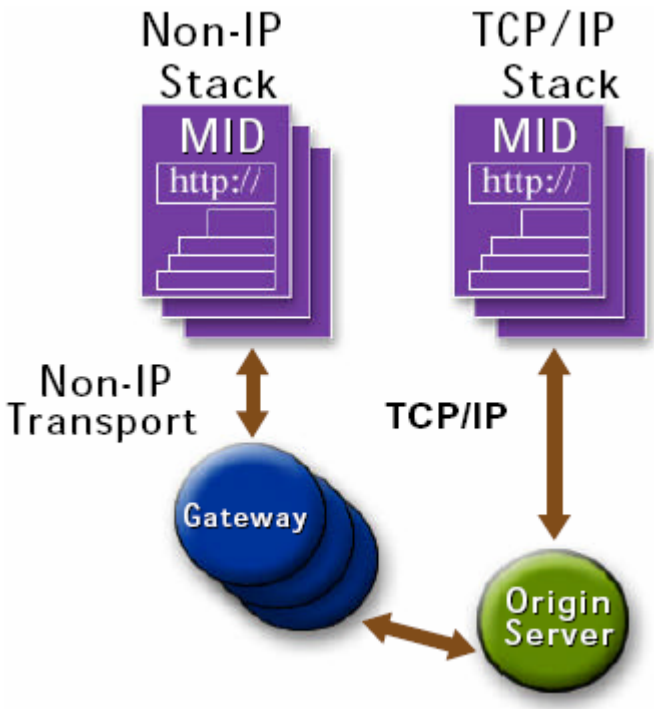
A puzzle game developed using Canvas

J2ME Overview - MIDP

- Media
  - Included in javax.microedition.media.*
  - Allows playing of audio and video
  - Allows control the playing of the media like the volume and tone of audio

J2ME Overview - MIDP

- Networking
  - Included in javax.microedition.io.*
  - MIDP devices must implement client portion of HTTP 1.1 protocol
- May or may not use IP-based transport
  - Non-IP Examples: HTTP layered on top of PDC-P for i-mode on WSP for WAP
- IP is the protocol that we used in Internet



J2ME Overview - MIDP

- Security
  - Included in javax.microedition.pki.*
  - Handling security certificate
  - Allows you to get the public and perform encryption or verification

J2ME Resources

J2ME devices

J2ME Resources

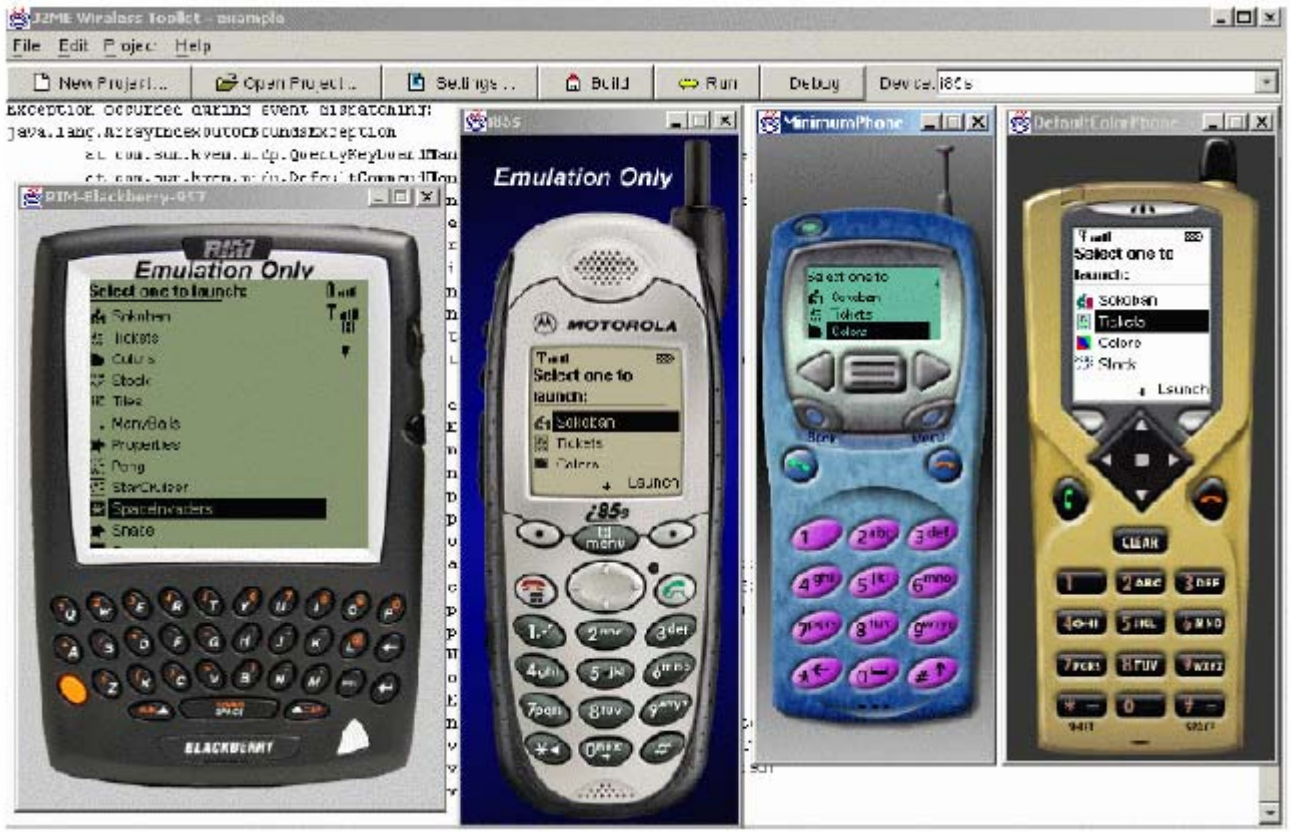
Key J2ME Deployments

J2ME Resources

- J2ME Development Tools
  - Sun J2ME Wireless Toolkit (SolarisTM platform, Linux, and Win32), standalone or plugged into Sun ONE Studio Mobile Edition IDE
  - MIDP reference (Palm OS and Win32), CLDC SDK (Solaris platform, Linux, and Win32)
  - Numerous third party tools (Borland/Nokia, Metrowerks/Motorola, Siemens, etc.)
  - Any IDE or Java tool, including J2SE SDK
  - Tools from the *J2ME Archive* (XML parsers and SOAP tools, J2ME-based microbrowsers, kAWT, etc.)

J2ME Resources

- Sun J2ME Wireless Toolkit



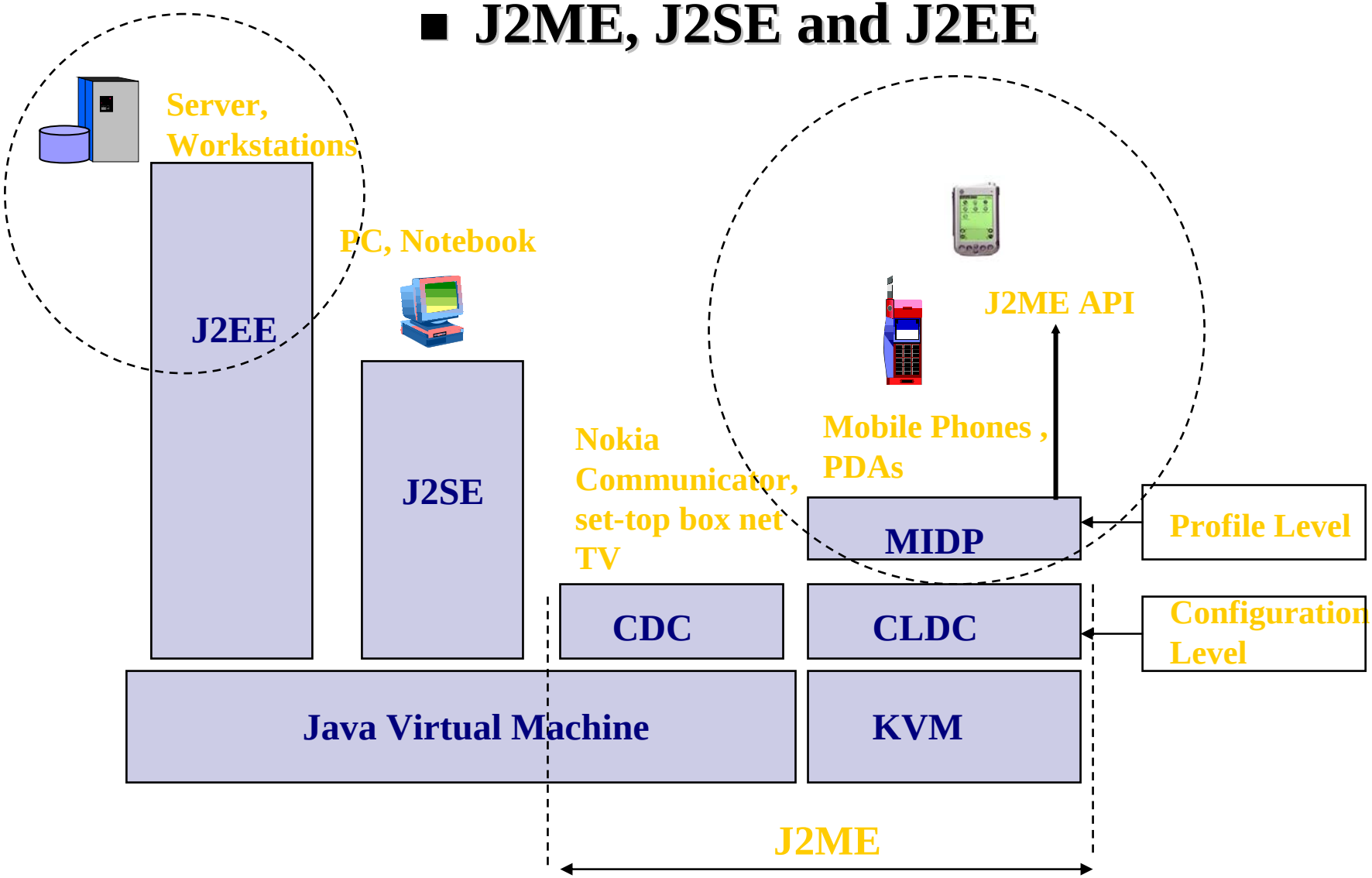
J2ME Resources

- Development Tools for the J2ME Platform
  - Devices are available NOW
  - Everything needed to start writing great Java applications for mobile devices is available TODAY
  - Many of the best tools are FREE

WHY J2ME?

- Different mobile devices utilize different platforms. An open platform is needed;
- Many devices have no option to download and install software dynamically
- Take advantages of Java language; But J2SE not realistic for mobile devices;
- Over 20 million J2ME mobile phones produced in 2001. It is estimated that over 450 million J2ME mobile phones will be sold in 2007.
- Java will be the dominant mobile terminals platform in the wireless sector.

■ J2ME, J2SE and J2EE



J2ME Again ☺

- To allow development of Java applications for devices that do not have the same processing power and memory found on a desktop platform. The devices may include cellular phones, PDAs, pagers, or others.
- J2ME is divided into two broad categories, known as Configurations, Connected Devices Configuration (CDC) and Connected Limited Device Configuration (CLDC).
- The Virtual Machine (KVM), which takes into consideration the limited resources available on devices that fit this configuration, was developed for the CDLC.
- On top of Configurations are device Profiles, such as, Mobile Information Information Profile (MIDP) and the associated libraries, which are our main focus for this subject.

K Virtual Machine (KVM)

- KVM was designed to be as small and efficient as possible, in order to support small resource constrained devices;
- The runtime footprint is only about 60 Kilobytes;
- The virtual machine will run in only a few tens of Kilo-bytes of memory.
- Sun’s reference implementation of the KVM is written in C to facilitate portability.
- The C codes can be downloaded.

Configurations

- A *Configuration* defines a Java platform for a broad range of devices. In fact, it defines the Java language features and the core Java libraries of the JVM for particular Configurations;
- The dividing line between the CDC and the CLDC depends on the memory, display, network connectivity and processing power available on a device;
- The current division is shown in the next slide. But it will not be always be the case.

Two Configurations

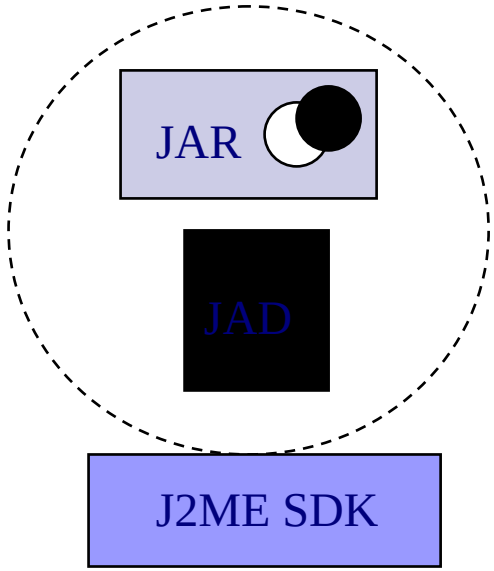
- **CDC**
- 512 KB (min.) memory for running Java
- 256 KB (min.) for runtime memory allocation
- Network connectivity, possibly persistent and high bandwidth

- **CLDC**
- 128 KB (min.) memory for running Java
- 32 KB (min.) for runtime memory allocation
- Restricted user interface
- Low power, typically battery powered
- Network connectivity, typically wireless, with low bandwidth and intermittent access

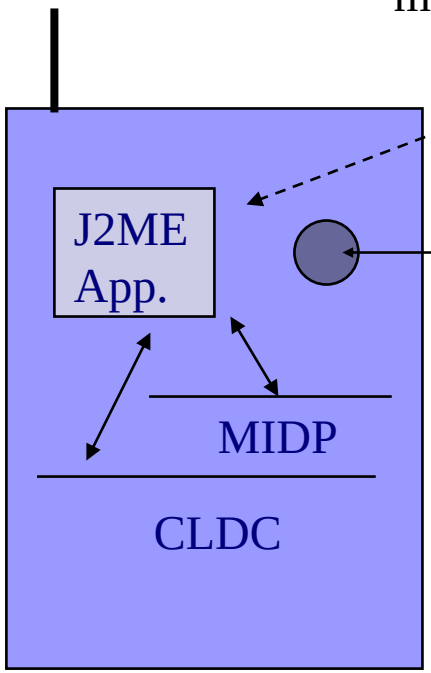
Profiles

- It’s all well and good that devices will fall within one Configuration or other. However, what seems limited to one device in a Configuration may be an abundance to another. Recall the analogy of cellular phone screen size versus that of a PDA;
- To address this broad range of capacities, and to provide for more flexibility as technology changes, Sun introduced the concept of a Profile to the J2Me platform.
- A ***Profile*** is an extension, if you will, to a Configuration. It provides the libraries for a developer to write applications for a particular type of device.
- For example, the MIDP defines APIs for interface components, input and even handling, persistent storage, networking and timers, taking consideration the screen and memory limitation of mobile devices.

J2ME App. – a set of MIDlets

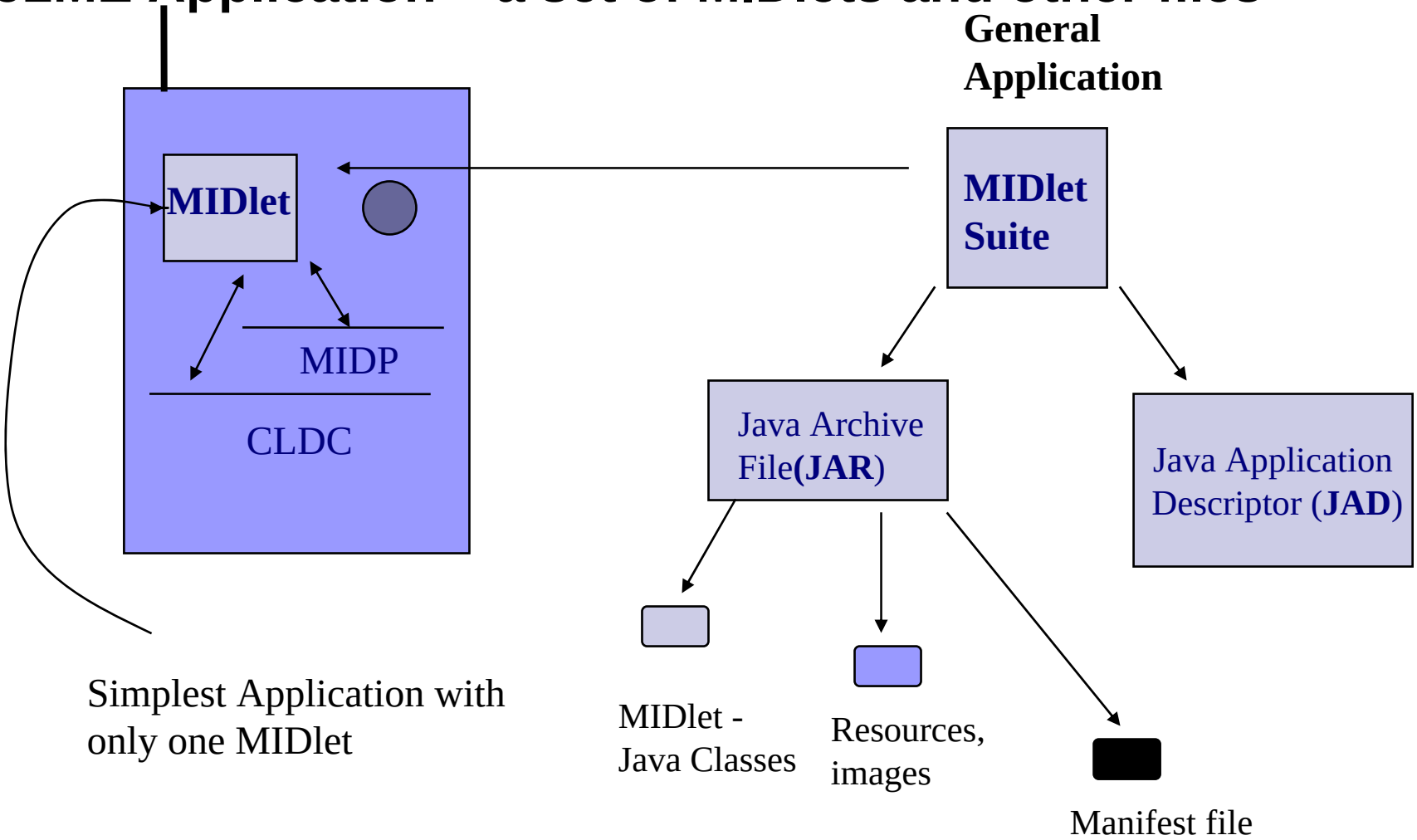


Downloading MIDlets either "Over Air" or cable, depends on manufacturers



Application Manager (software):
Install,
run and remove MIDlet;
It is device dependent.

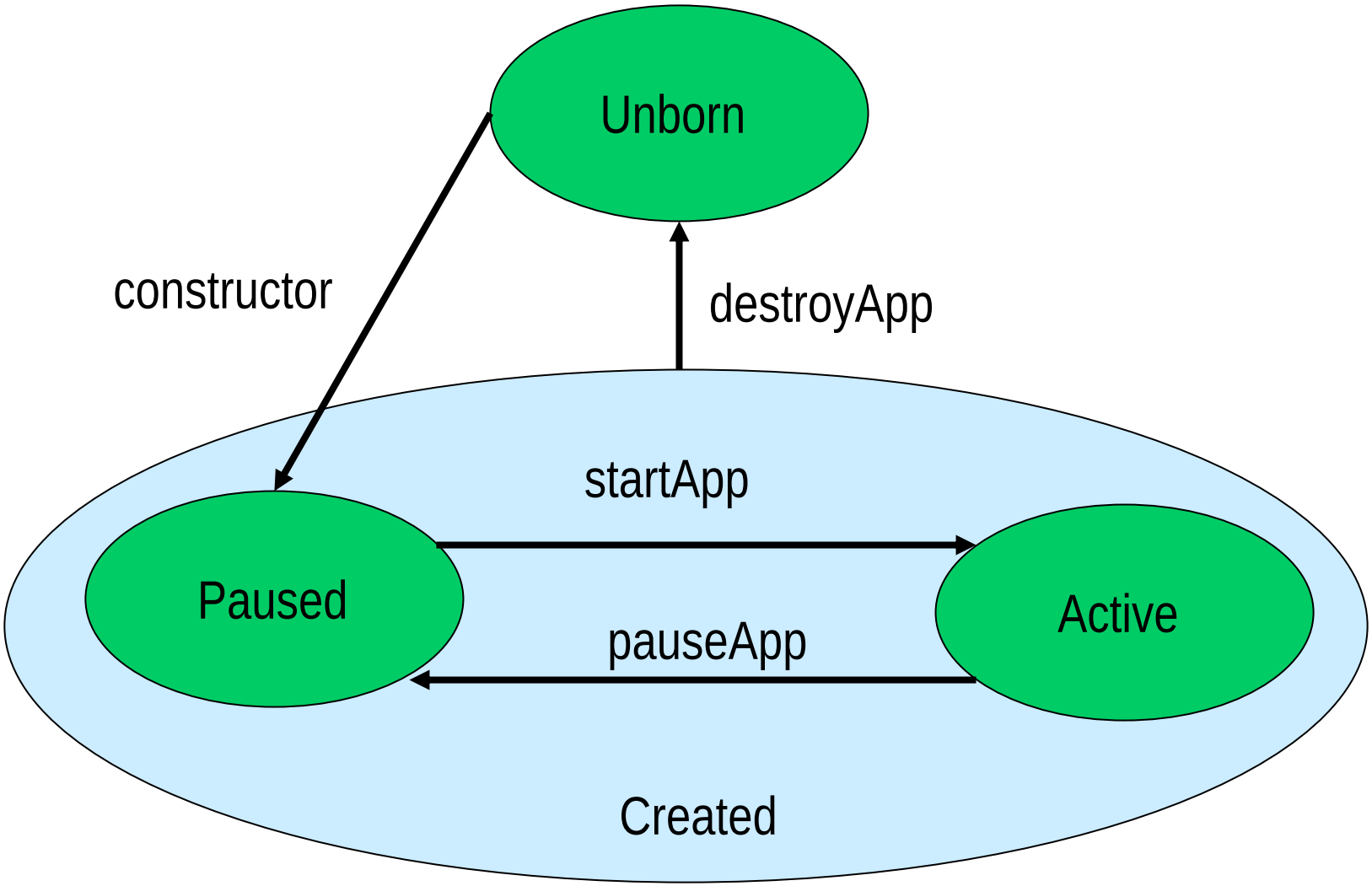
J2ME Application – a set of MIDlets and other files



MIDlet application model

- Defines the concept of an application in MIDP environment
- `javax.microedition.midlet`
- Every application must be a subclass of class MIDlet
  - `constructor`
  - `startApp()`
  - `pauseApp()`
  - `destroyApp()`
- In many cases, one also implements interface `CommandListener`

MIDlet states



Sample code: HelloMIDlet

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import java.util.*;

public class HelloWorld extends MIDlet implements CommandListener {
  private Command exitCommand;
  private TextBox tb;
  public HelloWorld() {
    exitCommand = new Command("exit", Command.EXIT, 1);
    tb = new TextBox("HelloWorld", "Hello world!", 15, 0);
    tb.addCommand(exitCommand);
    tb.setCommandListener(this);
  }
  protected void startApp() {Display.getDisplay(this).setCurrent(tb);}
  public void commandAction(Command c, Displayable d) {
    if (c== exitCommand) { destroyApp(false); notifyDestroyed();}
  }
  protected void destroyApp(boolean u) {}
  protected void pauseApp() {}
}
```

MIDlet suites

- Applications are represented publicly as JAR files
  - Class files
  - Resource files
  - Manifest that describes the JAR contents
- Multiple MIDlets can reside in the same JAR file
  - MIDlet suite
  - Applications can share data only with those applications that are in the same MIDlet suite
  - Therefore MIDlet design is an important architectural decision when aiming at larger systems
- Each MIDlet suite has a short textual descriptor file
  - A device user does not have to download the entire application before knowing if it will actually run in the device
  - JAD file with the contents that are (almost) identical with JAR manifest

JAD/JAR manifest contents

- MIDlet-Name --- MIDlet suite name
- MIDlet-Version
- MIDlet-vendor
- MIDlet-Icon
- MIDlet-Info-URL
- MIDlet-<n> --- name, icon and class per midlet
- MIDlet-Jar-URL
- MIDlet-Jar-Size
- MIDlet-Data-Size
- MicroEdition-profile --- J2ME profile
- MicroEdition-Configuration --- J2ME configuration

Example: JAD/JAR manifest of a simple application

```
MIDlet-1: HelloWorld, /HelloWorld.png,
  HelloWorld
MIDlet-Jar-Size: 2179
MIDlet-Jar-URL: HelloWorld.jar
MIDlet-Name: HelloWorld
MIDlet-Vendor:
MIDlet-Version: 1.0
MicroEdition-Configuration: CLDC-1.0
MicroEdition-Profile: MIDP-1.0
```

MIDlet suite execution

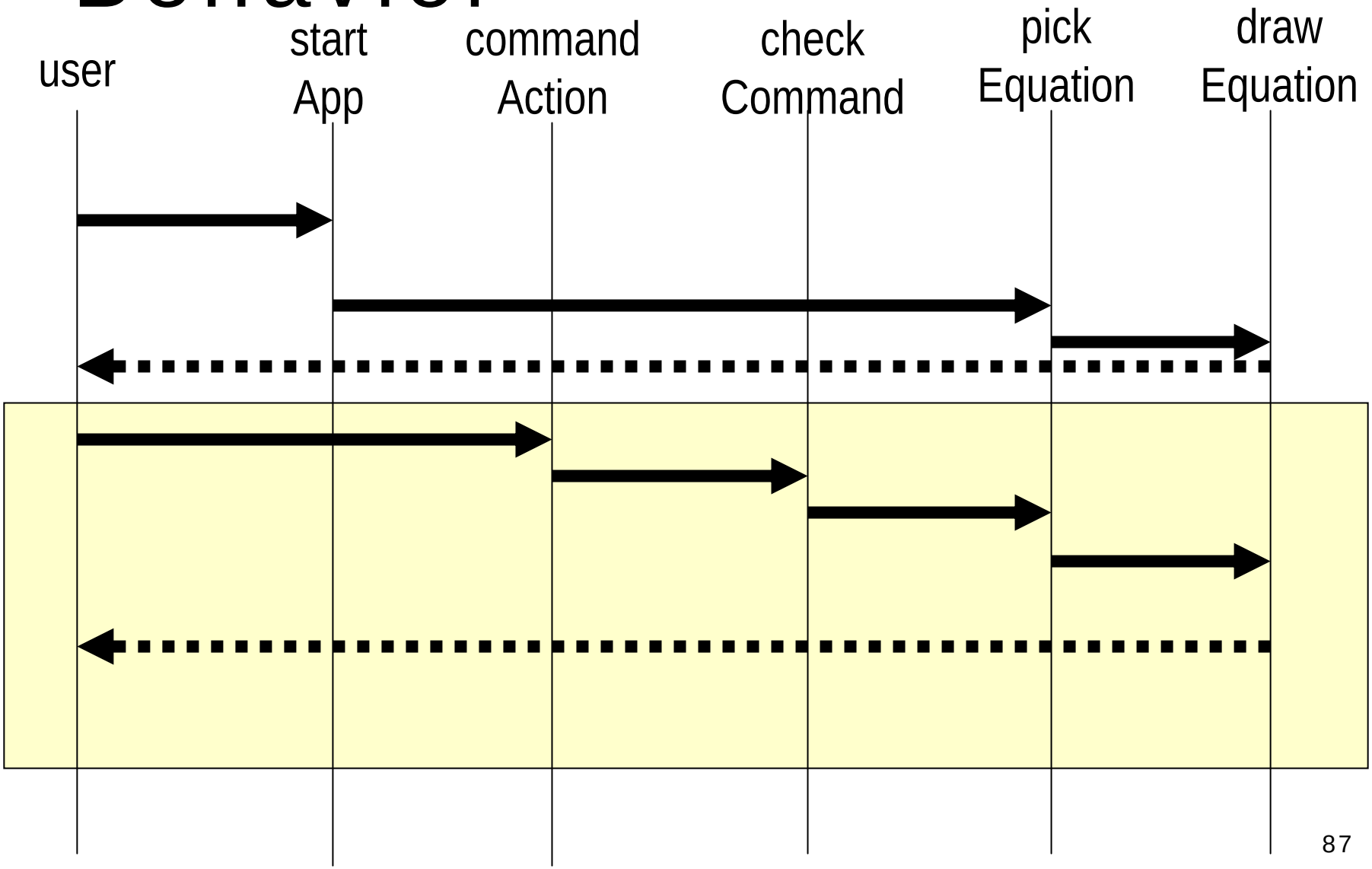
- Classes and native code that implement CLDC and MIDP
  - Only shared name space for all MIDlet suites
- Classes within MIDlet suite's JAR file
- All non-class files in the MIDlet's suite's JAR file
  - Icons, images, JAR manifest
  - Accessible with java.lang.Class.getResourceAsStream
- Contents of Application Descriptor File
  - Accessible with java.lang.Class.getResourceAsStream
- Separate name space for RMS record stores

Application: MathHangman

- Simple MIDP Application
- Takes around 1 hour to build
- Puzzle type
- Next slide shows the process flow/behaviour
- Uses basic MIDP components and J2ME packages

  i.e. javax.*

Behavior


user
start App
command Action
check Command
pick Equation
draw Equation

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class mathHangman extends MIDlet implements
   CommandListener {
  private Command exitCommand, rightCommand, wrongCommand;

  private TextBox tb;  /* Quiz canvas */
  private int nth;     /* Counter for equation and answer
   pair. */
  String Questions[] = /* Questions */
       { "1 + 3 = 4",
         "3 + 12 = 14",
         "12 + 7 = 19",
         "4 + 9 = 12",
         "1 + 2 = 3",
         "2 + 12 = 15",
         "3 + 12 = 15",
         "12 + 2 = 13",
         "10 + 5 = 15",
         "12 + 5 = 16" };
  int Qlen = Questions.length;
```

```
public mathHangman() {
    exitCommand = new Command("exit", Command.EXIT, 1);
    wrongCommand = new Command("WRONG", Command.CANCEL, 2);
    rightCommand = new Command("RIGHT", Command.OK, 3);
}
protected void startApp() { pickEquation(); }

protected void destroyApp(boolean u) {}

protected void pauseApp() {}

 public void commandAction(Command c, Displayable d) {
    if (c == exitCommand) {
      destroyApp(false);
      notifyDestroyed();
    } else ( checkCommand(c == rightCommand);
    } // else
  }
```

```
private void pickEquation(){
  displayEquation(Questions[nth]);}


private void displayEquation(String equation)
  {
  tb = new TextBox(”mathHangman", equation,
  15, 0);
  tb.addCommand(wrongCommand);
  tb.addCommand(rightCommand);
  tb.setCommandListener(this);
  Display.getDisplay(this).setCurrent(tb);
}
```

```
  private void checkCommand(boolean b) {
     if(((nth%2) == i) = b) {
       tb = new TextBox("mathHangman",
                        "You failed the challenge", 30, 0);
       tb.addCommand(exitCommand);
       tb.setCommandListener(this);
       Display.getDisplay(this).setCurrent(tb);
     } else {
       nth = nth + 1;
       if (nth == Qlen) {
          tb = new TextBox("mathHangman",
                           "You mastered the challenge", 30, 0);
          tb.addCommand(exitCommand);
          tb.setCommandListener(this);
          Display.getDisplay(this).setCurrent(tb);
        } else {
          pickEquation();
        }
     }
  }
} // class mathHangman
```

J2ME WTK Installation