MOBILE GAMES

Mobile Game Architecture and Design
J2ME Platform and Tools

NUS
National University of Singapore

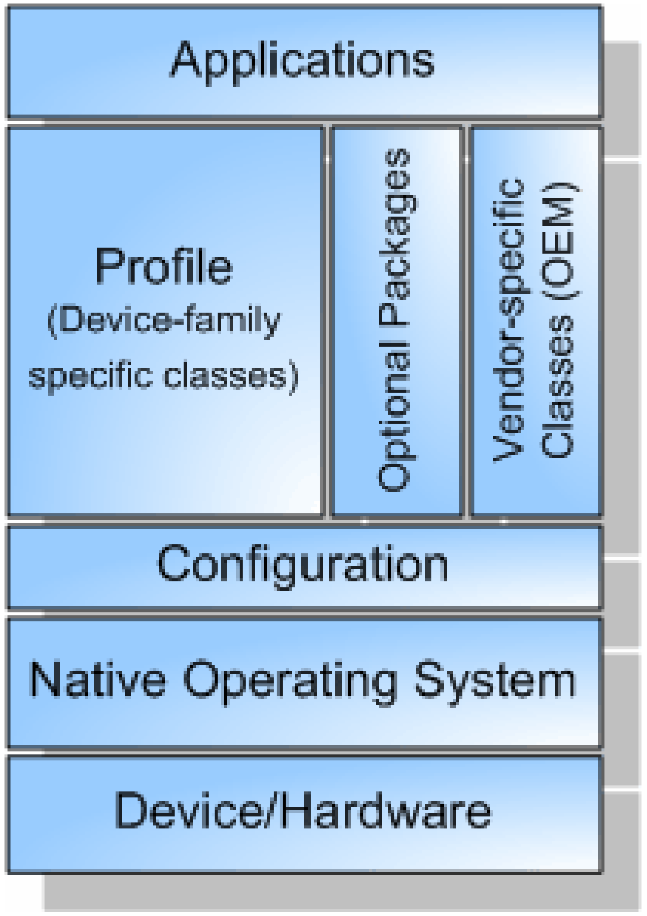
In this session…

- ✶ J2ME platform
- ✶ J2ME architecture
- ✶ J2ME development tools
- ✶ OTA Provisioning

J2ME Platform – overview

- ✶ Java 2 Standard Edition
  - – Standard Client/Server applications including web based applications.
- ✶ Java 2 Enterprise Edition
  - – Multi-tiered and potentially distributed applications.
  - – Collection of vendor independent APIs for server-side programming.
- ✶ Java 2 Micro Edition
  - – Client/Server applications for mobile devices with limited power, network connectivity and GUI capabilities.
  - – Goals:
    - – Focuses on the personal mobile devices with limited resources and differences in capabilities, features and processing abilities.
    - – To Provide facility to connect devices to various types of networks.
    - – To provide facility to deliver applications and data over a network connection.

J2ME Platform Organization

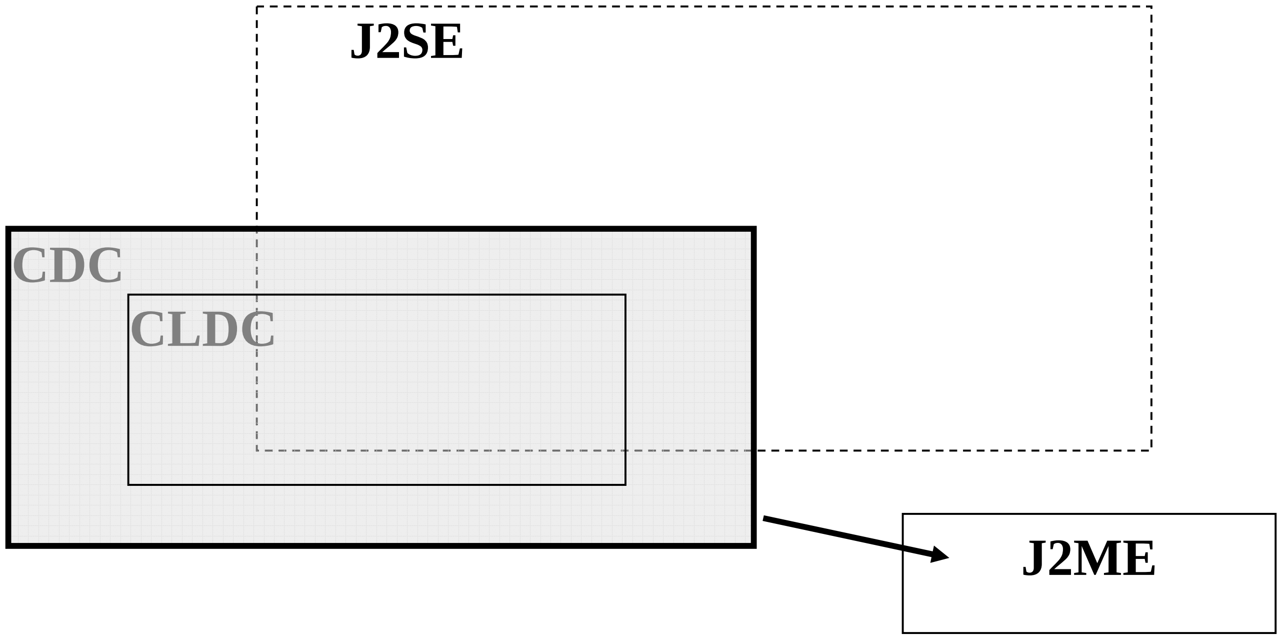


SOURCE: WWW.SUN.COM

J2ME Platform – Conceptual Layers

Higher end PDAs, Set-top Boxes	Mobile Phones Pagers, Industry devices	Smartcard	Example Devices
Foundation, Personal, Per. Basis	MIDP, IMP	GSM, Open Platform	Profile Layer
CDC	CLDC	Java Card API, Security, RMI	Configuration Layer
Full J2SE VM	KVM	Java Card VM	Virtual Machine Layer

J2ME Configurations



CDC (Connected Device Configuration)

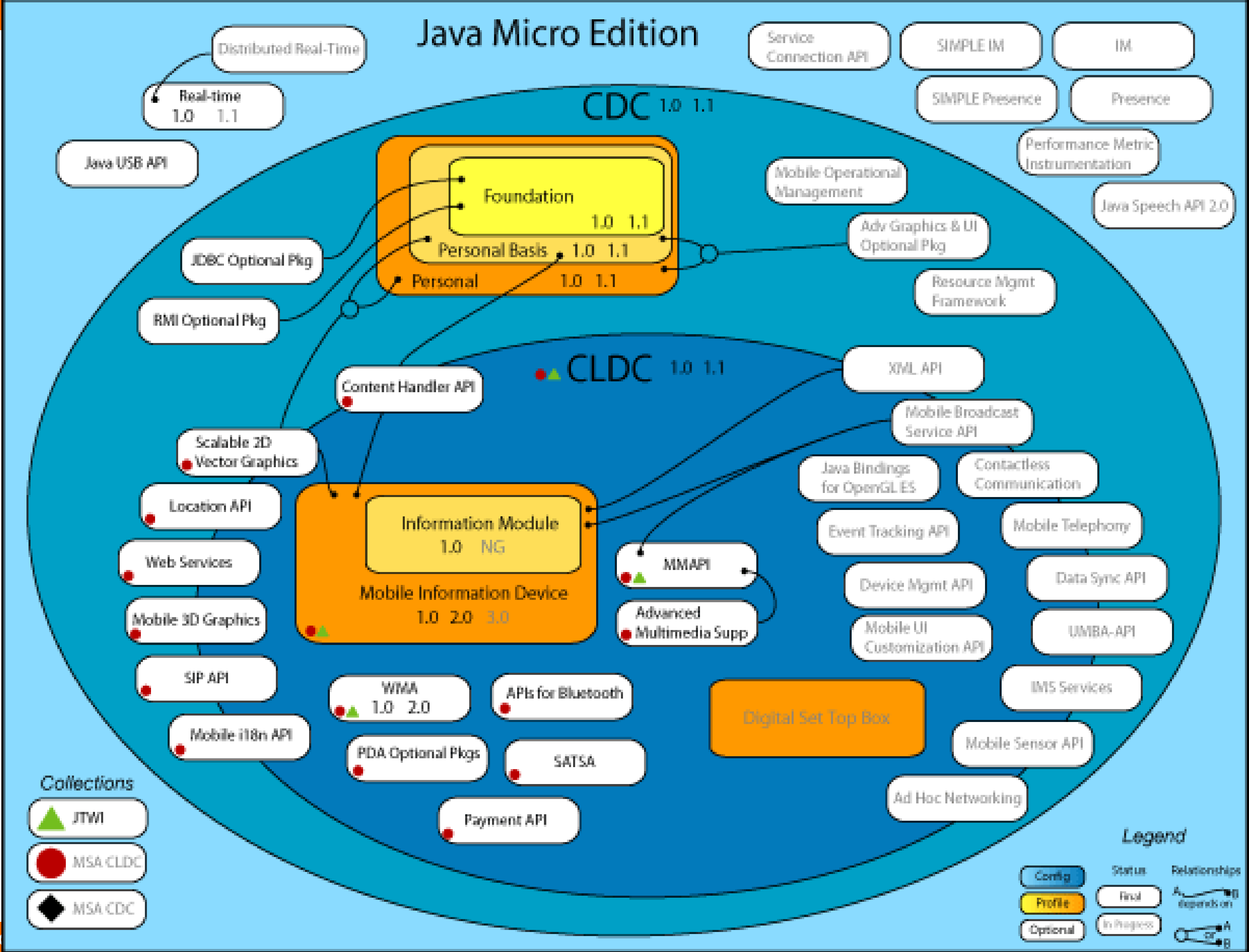
- ✶ Shared, Fixed, Connected Information Devices
- ✶ Robust UI Functions
- ✶ 2-16 MB Memory Range (RAM and ROM)
- ✶ Greater than 32-bit CPU
- ✶ Persistent, High Bandwidth Network Connections
- ✶ Examples: TV Set Top Boxes, Internet TV's, Internet Enabled Screen Phones, High End Communicators, Auto Entertainment/Navigation Systems

CLDC (Connected Limited Device Configuration)

- ✶ Simple UI
- ✶ 128KB-1MB Memory Range. The virtual machine and the libraries take 128KB of memory.
- ✶ 16-bit, 32-bit CPU
- ✶ Low Bandwidth, Intermittent Networks
- ✶ Generally don’t use TCP/IP
- ✶ Examples: Low End Cell Phones, Two-Way Pagers, and Palm OS Handholds

NUS National University of Singapore

J2ME TODAY



SOURCE: WWW.SUN.COM

Development Tools - Overview

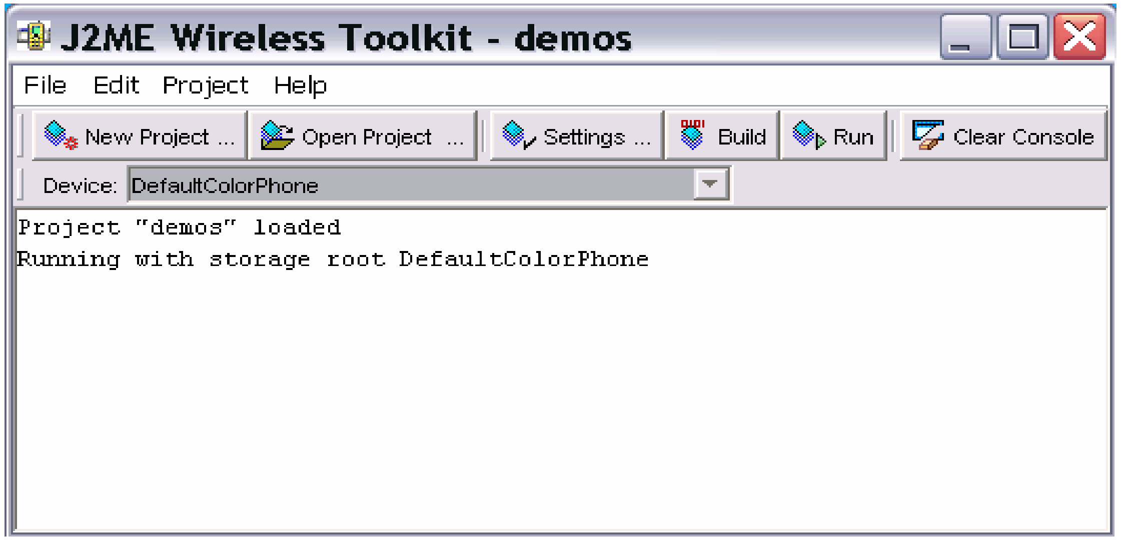
- J2ME Wireless Toolkit (J2ME WTK 2.5) (http://java.sun.com)
  - J2ME API library, Emulator, Compiler
  - No editor (can use free editors like, JCreate LE, **Text Pad 4.7.3**)
  - Vendor Specific kits based on J2ME WTK:
    - » Sony Ericsson SDK 2.2.4 for the Java(TM) ME Platform (http://developer.sonyericsson.com)
    - » The Java SDK for S60 3rd Edition platform (http://forum.nokia.com)
- IDE
  - Commericial: JBuilder (borland.com), JDeveloper (oracle.com)
  - Open Source: eclipse (http://www.eclipse.org), **Netbeans 5.0** (www.netbeans.org)
  - Plug-ins for j2me
    - » Jbuilder Mobile Set
    - » **Netbeans Mobility Pack 5.0**

Development Tools - Overview

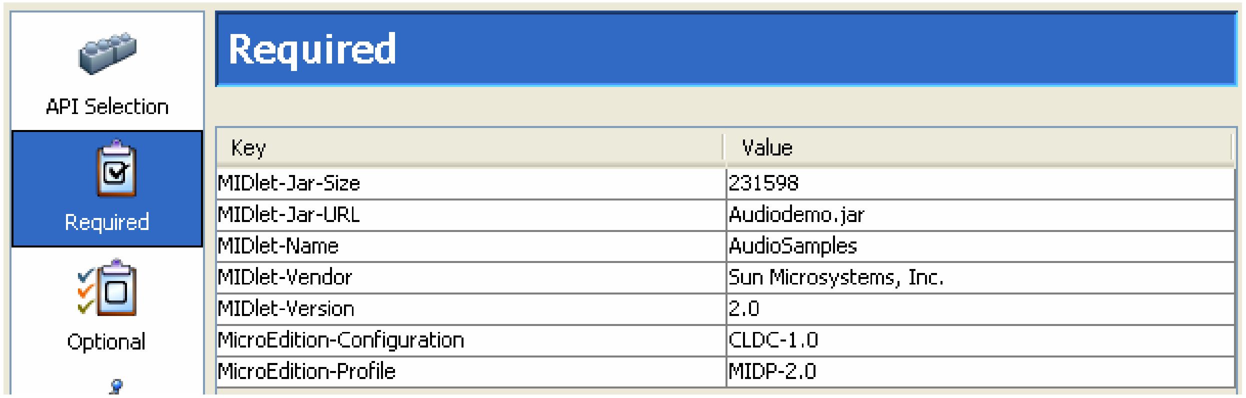
- 3D Modeling
  - Commercial: 3D Studio Max, Maya, Softimage, Litewave 3D
  - Open Source: **Blender 2.42** (http://www.blender.org), ogre3d (http://www.ogre3d.org/), Irrlicht Engine (http://irrlicht.sourceforge.net/), NeoEngine (http://www.neoengine.org/), **Panda3D 1.2.3** (www.**panda3d**.org/)
    - Plug-in for J2ME
      - » M3g-export.py – (www.nelson-games.de/bl2m3g/)
        - Needs python 2.4.3 (www.python.org)

- Game Server
  - Client Server Communication APIs (such as, **J2SE/J2EE based Network API**, .Net or WinSock API, SNAP)
  - Simple Database System

J2ME WTK & Project Settings



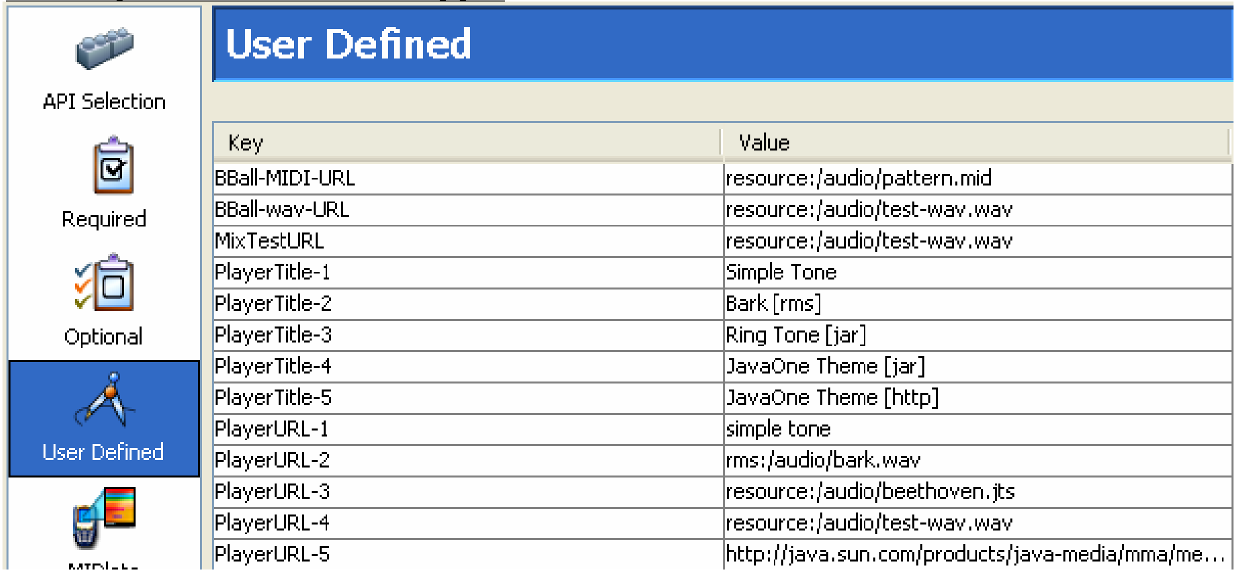
```
Project "demos" loaded
Running with storage root DefaultColorPhone
```



Required

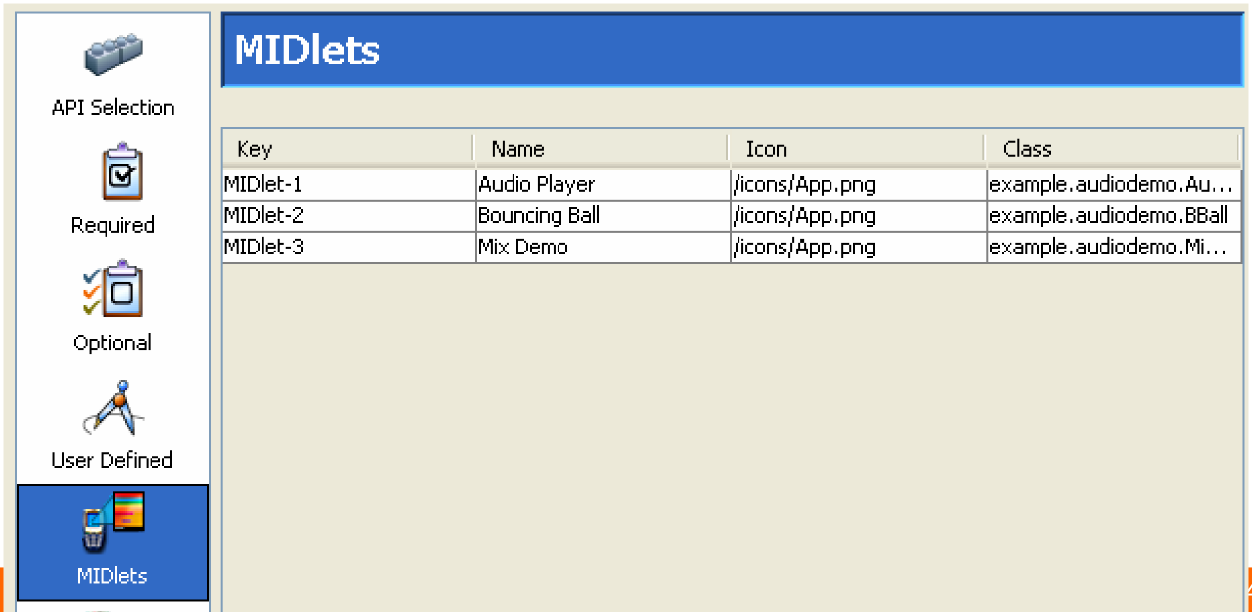
Key	Value
MIDlet-Jar-Size	231598
MIDlet-Jar-URL	Audiodemo.jar
MIDlet-Name	AudioSamples
MIDlet-Vendor	Sun Microsystems, Inc.
MIDlet-Version	2.0
MicroEdition-Configuration	CLDC-1.0
MicroEdition-Profile	MIDP-2.0

Project Settings



User Defined

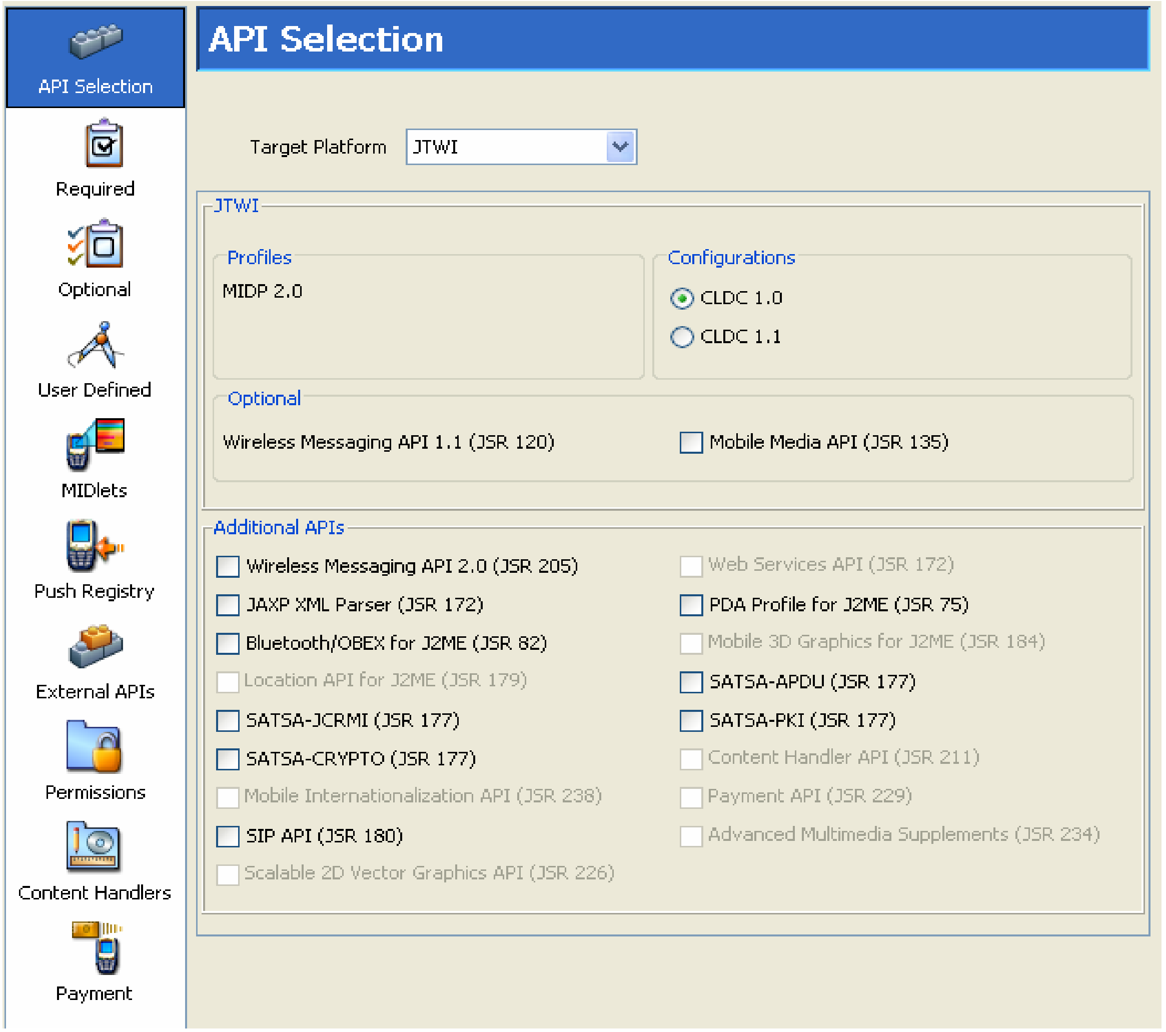
Key	Value
BBall-MIDI-URL	resource:/audio/pattern.mid
BBall-wav-URL	resource:/audio/test-wav.wav
MixTestURL	resource:/audio/test-wav.wav
PlayerTitle-1	Simple Tone
PlayerTitle-2	Bark [rms]
PlayerTitle-3	Ring Tone [jar]
PlayerTitle-4	JavaOne Theme [jar]
PlayerTitle-5	JavaOne Theme [http]
PlayerURL-1	simple tone
PlayerURL-2	rms:/audio/bark.wav
PlayerURL-3	resource:/audio/beethoven.jts
PlayerURL-4	resource:/audio/test-wav.wav
PlayerURL-5	http://java.sun.com/products/java-media/mma/me...



MIDlets

Key	Name	Icon	Class
MIDlet-1	Audio Player	/icons/App.png	example.audiodemo.Au...
MIDlet-2	Bouncing Ball	/icons/App.png	example.audiodemo.BBall
MIDlet-3	Mix Demo	/icons/App.png	example.audiodemo.Mi...

Project Settings



API Selection

Target Platform: JTWI

JTWI

Profiles: MIDP 2.0

Configurations: [✓ CLDC 1.0] CLDC 1.1

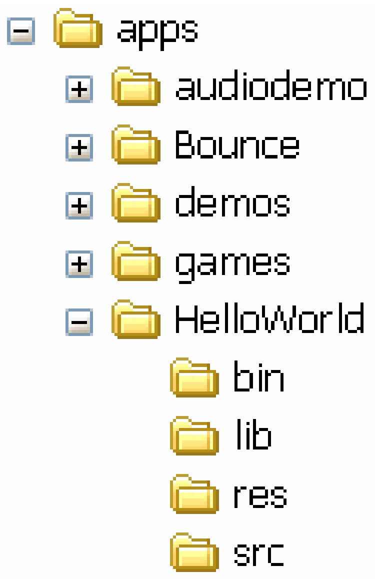
Optional

Wireless Messaging API 1.1 (JSR 120)

☐ Mobile Media API (JSR 135)

Additional APIs

☐ Wireless Messaging API 2.0 (JSR 205)	☐ Web Services API (JSR 172)
☐ JAXP XML Parser (JSR 172)	☐ PDA Profile for J2ME (JSR 75)
☐ Bluetooth/OBEX for J2ME (JSR 82)	☐ Mobile 3D Graphics for J2ME (JSR 184)
☐ Location API for J2ME (JSR 179)	☐ SATSA-APDU (JSR 177)
☐ SATSA-JCRMI (JSR 177)	☐ SATSA-PKI (JSR 177)
☐ SATSA-CRYPTO (JSR 177)	☐ Content Handler API (JSR 211)
☐ Mobile Internationalization API (JSR 238)	☐ Payment API (JSR 229)
☐ SIP API (JSR 180)	☐ Advanced Multimedia Supplements (JSR 234)
☐ Scalable 2D Vector Graphics API (JSR 226)	



```
apps
  audiodemo
  Bounce
  demos
  games
  HelloWorld
    bin
    lib
    res
    src
```

Folder Structure for HelloWorld project

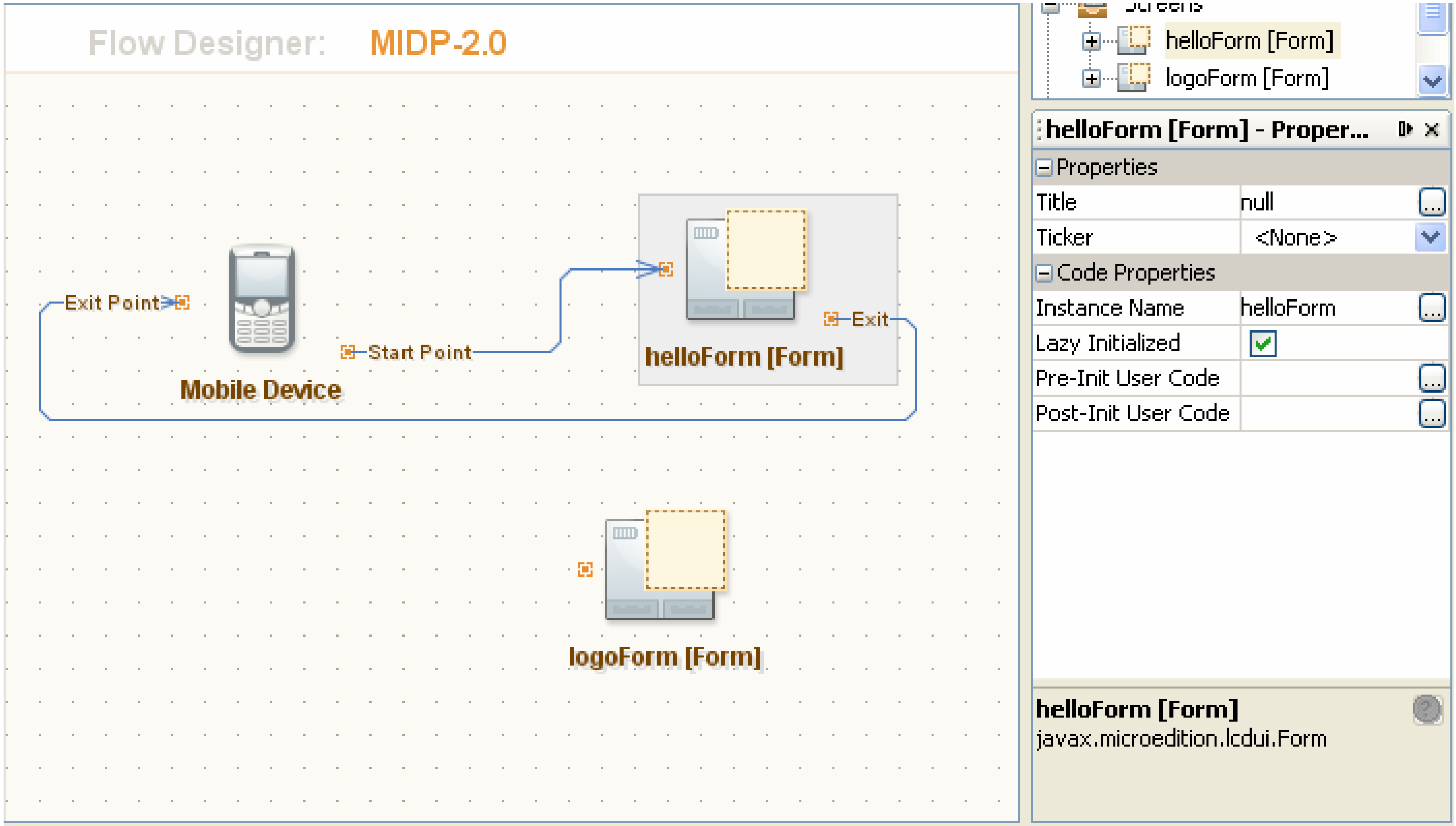
Pre-verification, Packaging & Deployment

- The process of doing class verification before deploying the application into the mobile device is referred to as *Pre-verification*.
- While you ‘package’ (create JAR & JAD files) the application in a desktop for deployment into a mobile device, the class verification takes place and it creates a pre-verification file and packaged together with the application.
- Deployment:
  - BlueTooth, IrDA, Data cable….
  - Operator, aggregator/publisher, developer, OTA Provisioning (discussed later…),

Pre-verification, Packaging & Deployment

1. Create Project
   - ✶ project folder and; bin, src, res and lib sub folders)
   - ✶ Add source code, resources and additional libraries .java,.m3g,.png.,mpg.,wma., mp3, ui library
2. Compile/Build
   - * .class files
3. Package (create JAD & JAR for deployment)
   - Normal package
   - Obfuscated package (uses **proguard**)
     - http://proguard.sourceforge.net
     - Download proguard.zip and extract the JAR file in it into WTK\bin folder

Netbeans IDE features – Designer View



Properties

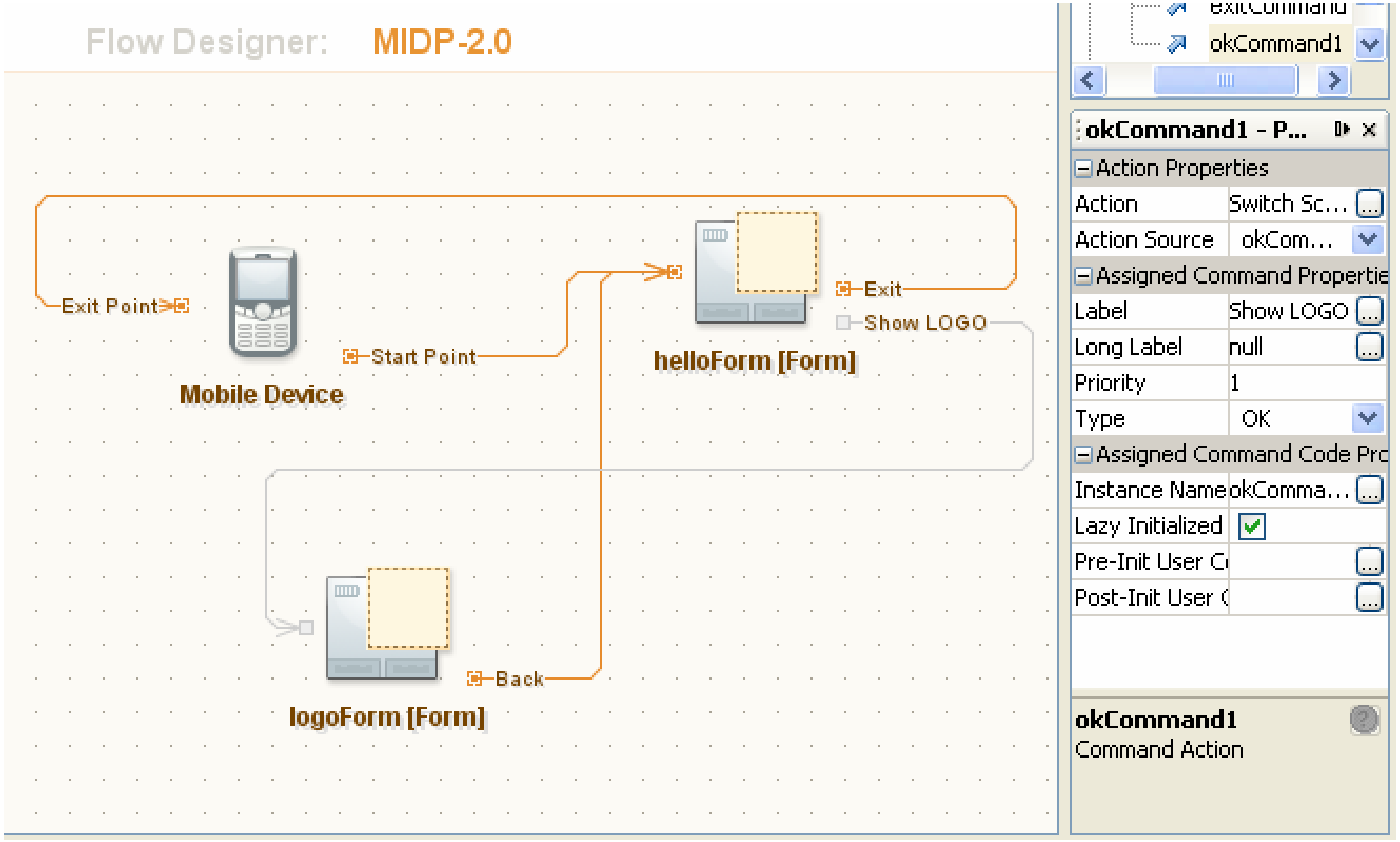
Title	null
Ticker	<None>

Code Properties

Instance Name	helloForm
Lazy Initialized	✓
Pre-Init User Code	
Post-Init User Code	

helloForm [Form]
javax.microedition.lcdui.Form

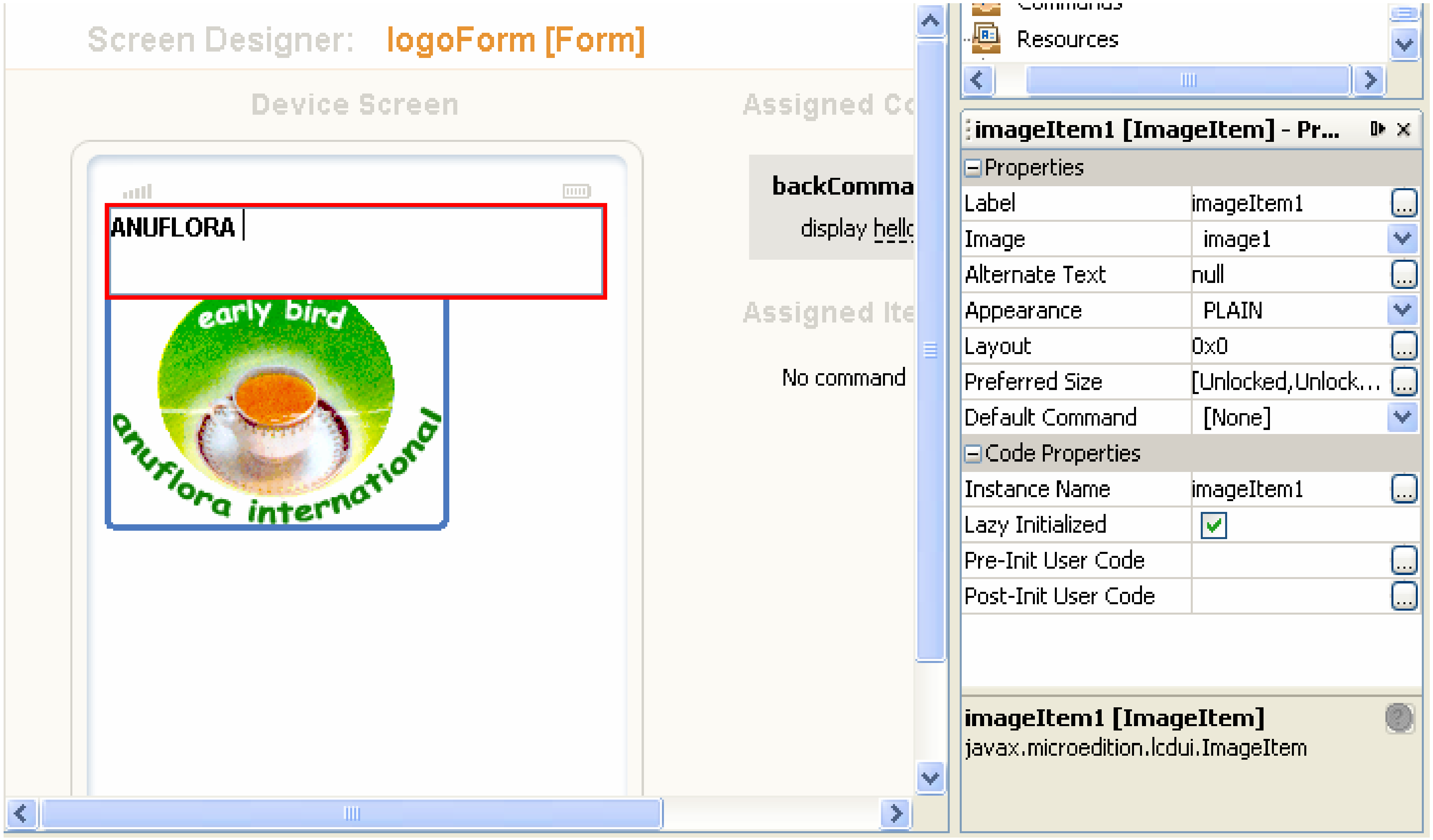
Netbeans IDE features – Adding new Form & Navigation Setting



Action Properties	
Action	Switch Sc...
Action Source	okCom...
Assigned Command Propertie	
Label	Show LOGO
Long Label	null
Priority	1
Type	OK
Assigned Command Code Pro	
Instance Name	okComma...
Lazy Initialized	✓
Pre-Init User C	
Post-Init User (	

okCommand1
Command Action

Netbeans IDE features – User Friendly Form Design



Screen Designer: logoForm [Form]

Device Screen

Assigned Co

ANUFLORA

backComma

display hello

Assigned Ite

No command

imageItem1 [ImageItem] - Pr...

Properties

Label	imageItem1
Image	image1
Alternate Text	null
Appearance	PLAIN
Layout	0x0
Preferred Size	[Unlocked,Unlock...
Default Command	[None]

Code Properties

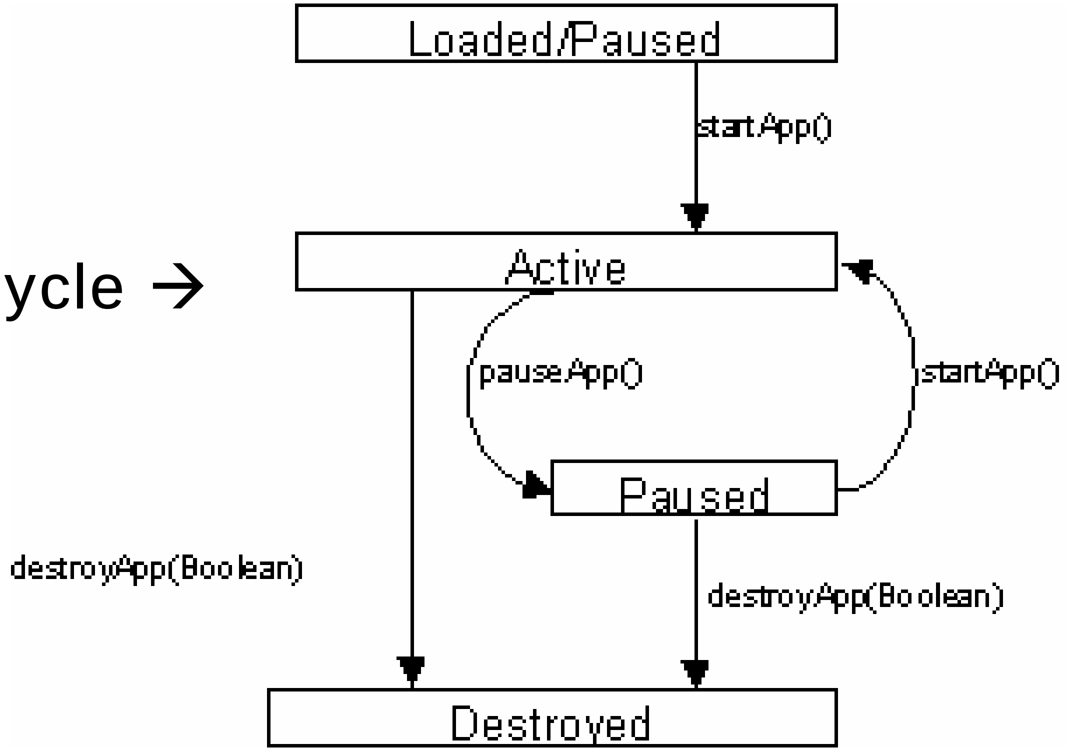
Instance Name	imageItem1
Lazy Initialized	✓
Pre-Init User Code	
Post-Init User Code	

imageItem1 [ImageItem]
javax.microedition.lcdui.ImageItem

MIDP Application Lifecycle (MIDlet Lifecycle)

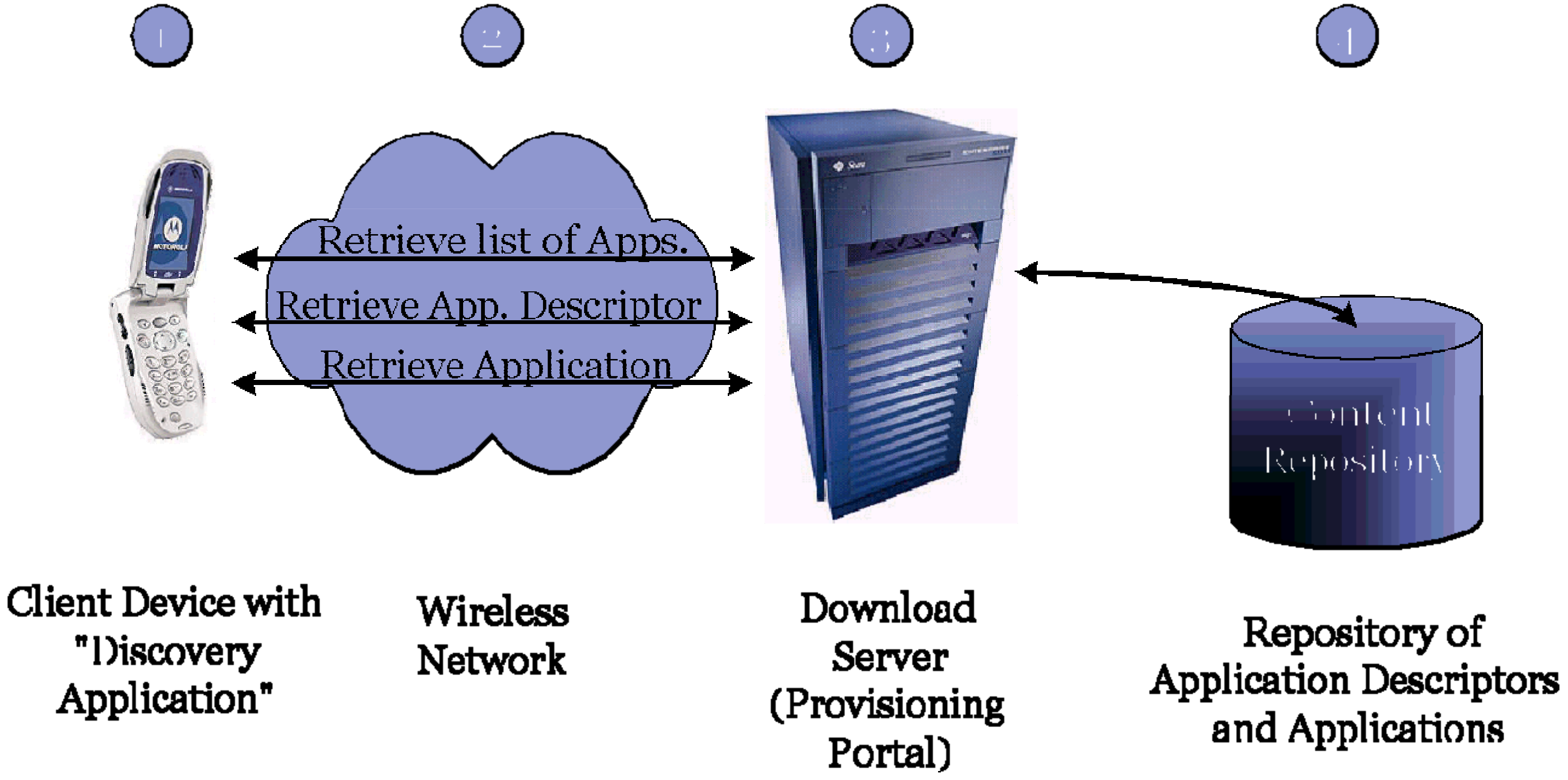
- MIDlet – is a J2ME-MIDP application. Extends MIDlet class defined in javax.microedition.MIDlet
- MIDlet Suite – collection of MIDlets
- MIDlet suits can share information, are packaged & deployed together as a single JAR file.

MIDP Application Lifecycle →



OTA Provisioning

Simplest form:


1
2
3
4
Retrieve list of Apps.
Retrieve App. Descriptor
Retrieve Application
Content Repository
Client Device with "Discovery Application"
Wireless Network
Download Server (Provisioning Portal)
Repository of Application Descriptors and Applications

Both Client and Server should use the same DA protocol.
DA protocol of MIDP OTA is HTTP

source: developers.sun.com

OTA Provisioning

- An OTA provisioning system typically encompasses
  - content publication and management,
  - access control,
  - installation (and upgrading) of applications,
  - and tracking the use of applications (content) for billing purposes.

OTA provisioning

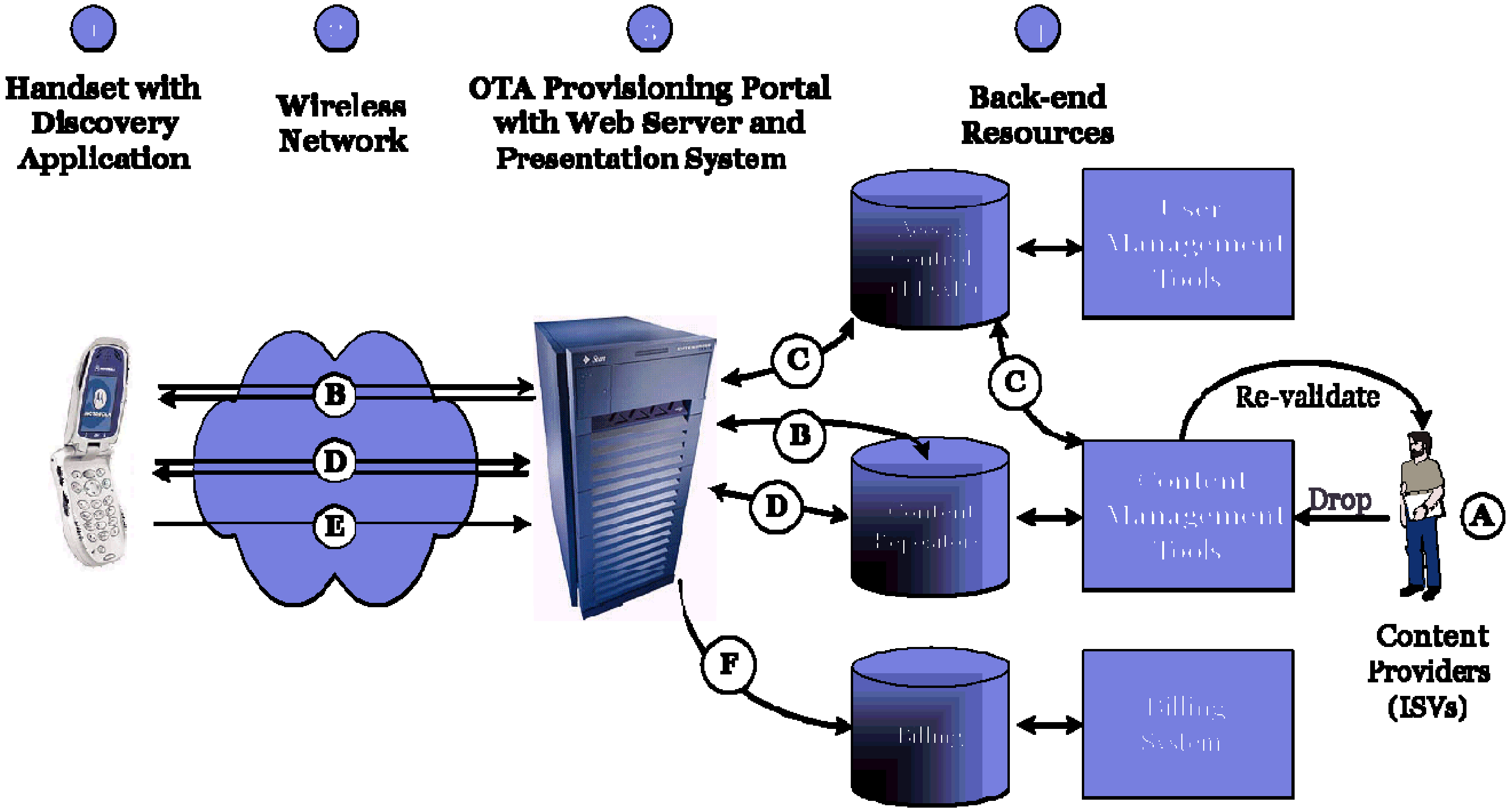


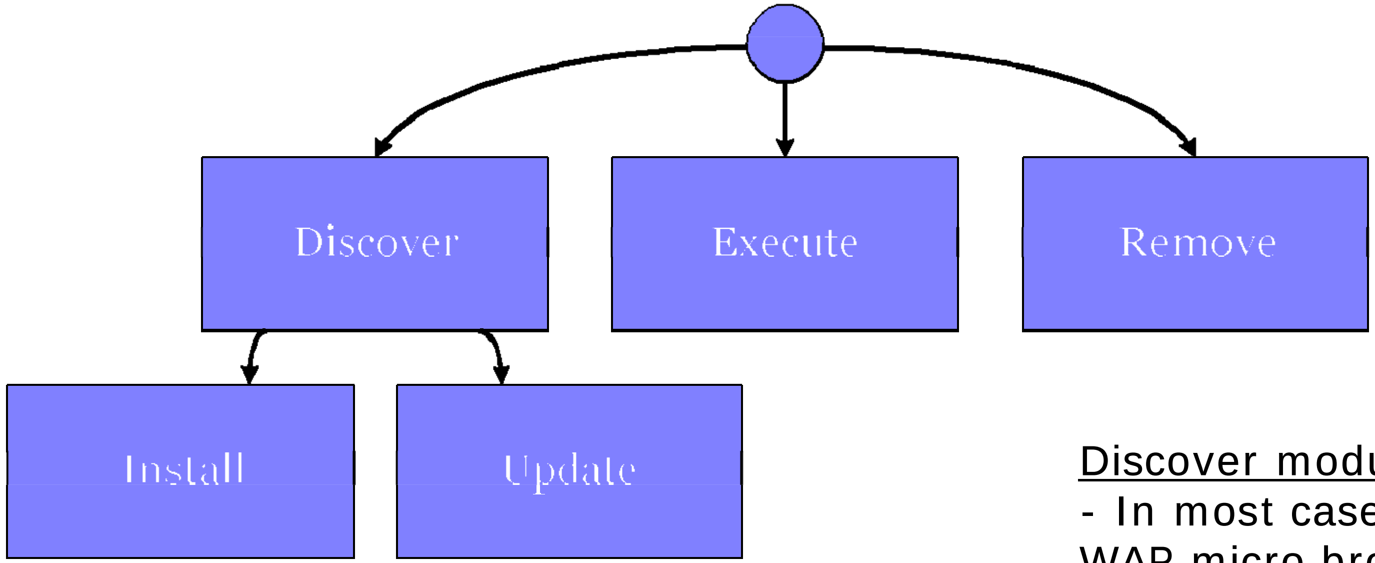
Image source: developers.sun.com

MIDP OTA Specification

- Device Functionality
  - Support for HTTP 1.0
  - Discovery Application (to locate application and to download. Eg. Micro-browser)
  - AMS to manage OTA Application Provisioning life cycle. In MIDP it is called JAM-Java Application Manager
- OTA Application Provisioning life cycle (next slide)

MIDP OTA Specification

– OTA Application Provisioning life cycle



Discover module:-
- In most cases it uses the HTTP or WAP micro browser. When the browser gets a MIDP application it sends it to the JAM to download and install.

Execute module:-
- Allows user to select MIDlet suite and MIDlet.
- Starts the MIDlet in *Paused* state.
- Calls startApp() in the MIDlet to bring it to *Active* state

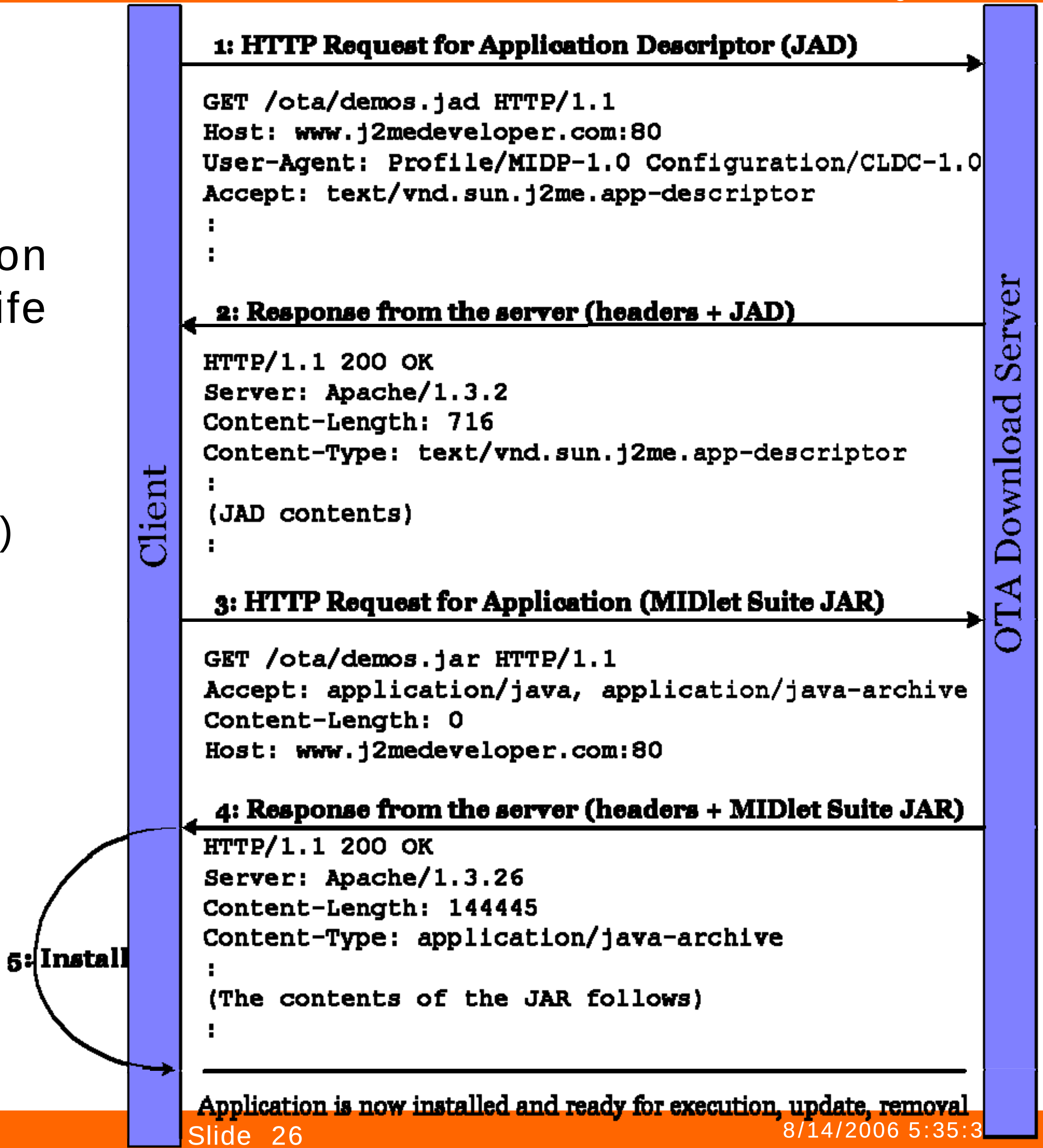
(Refer MIDlet Life cycle in previous slides.)

Image source: developers.sun.com

MIDP OTA Specification

– OTA Application Provisioning life cycle

(Installation and update module)


Client
OTA Download Server
1: HTTP Request for Application Descriptor (JAD)
2: Response from the server (headers + JAD)
3: HTTP Request for Application (MIDlet Suite JAR)
4: Response from the server (headers + MIDlet Suite JAR)
5: Install
Application is now installed and ready for execution, update, removal

```
GET /ota/demos.jad HTTP/1.1
Host: www.j2medeveloper.com:80
User-Agent: Profile/MIDP-1.0 Configuration/CLDC-1.0
Accept: text/vnd.sun.j2me.app-descriptor
:
:
```

```
HTTP/1.1 200 OK
Server: Apache/1.3.2
Content-Length: 716
Content-Type: text/vnd.sun.j2me.app-descriptor
:
(JAD contents)
:
```

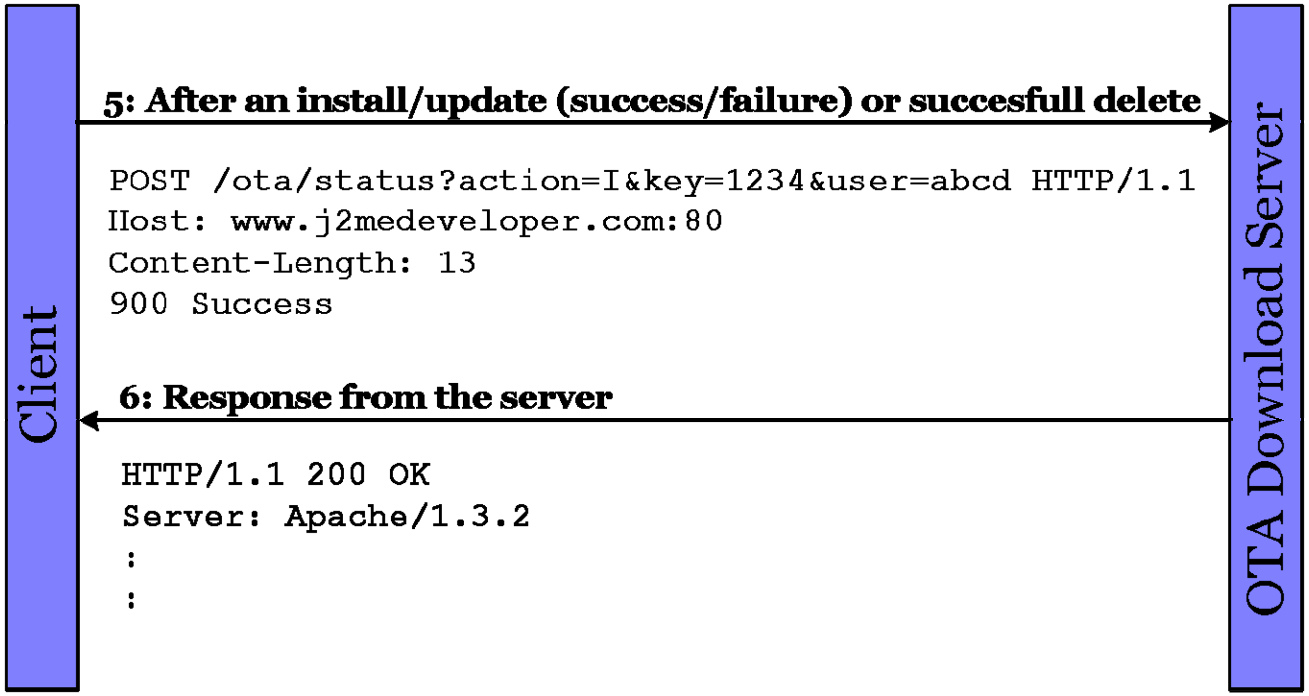
```
GET /ota/demos.jar HTTP/1.1
Accept: application/java, application/java-archive
Content-Length: 0
Host: www.j2medeveloper.com:80
```

```
HTTP/1.1 200 OK
Server: Apache/1.3.26
Content-Length: 144445
Content-Type: application/java-archive
:
(The contents of the JAR follows)
:
```

Image source: developers.sun.com

MIDP OTA Specification

– OTA Application Provisioning life cycle (Removal module)


Client
OTA Download Server
5: After an install/update (success/failure) or succesfull delete
6: Response from the server

```
POST /ota/status?action=I&key=1234&user=abcd HTTP/1.1
Host: www.j2medeveloper.com:80
Content-Length: 13
900 Success
```

```
HTTP/1.1 200 OK
Server: Apache/1.3.2
:
:
```

Removal Module: The application (complete MIDlet suite) and its associates RMS Entries will be removed. RMS – record store management system (local storage).

Image source: developers.sun.com

MIDP OTA Specification

Status Reports:

900 Success
901 Insufficient Memory
902 User Cancelled
903 Loss of Service
904 JAR size mismatch
907 Invalid JAR
909 Application authentication failure
910 Application authorization failure
912 Deletion Notification .....

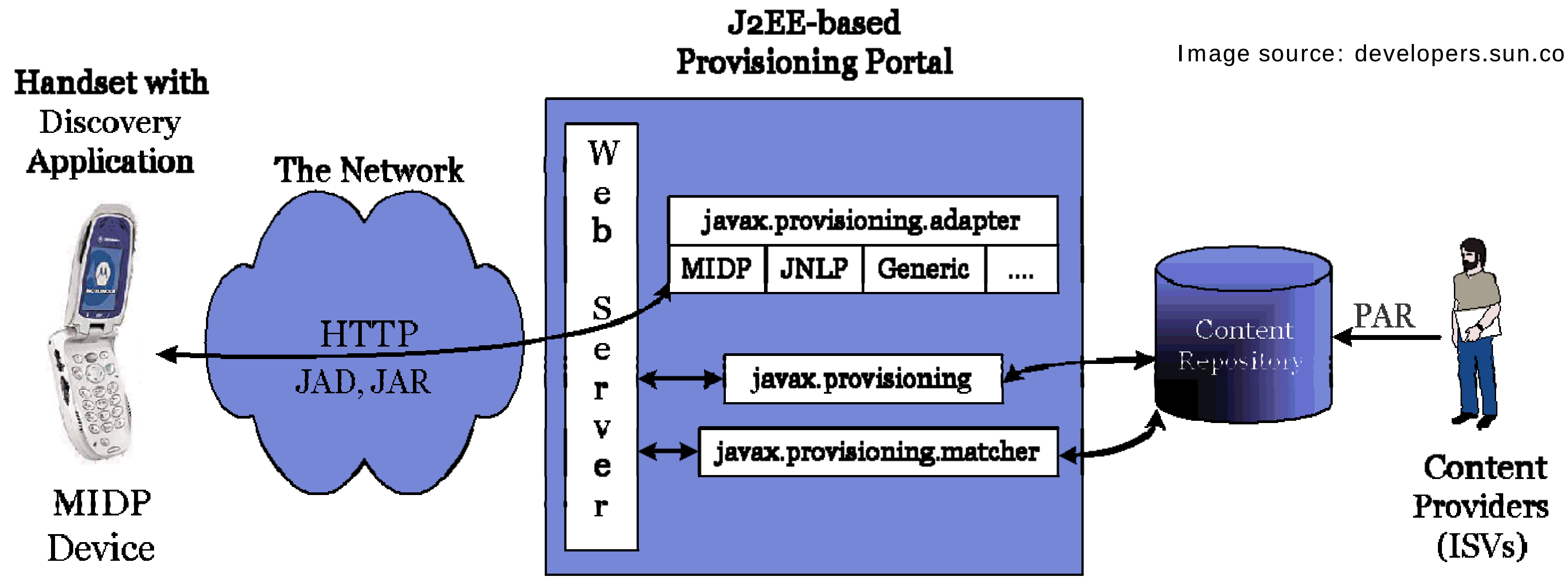
Refer: http://java.sun.com/products/midp/OTAProvisioning-1.0.pdf for full list

Provisioning portals (download Servers) may take advantage of status reports to track the use of an application - for example, for billing purposes or to prioritize their content repository.

MIDP OTA Specification

Provisioning Portal (eg. J2EE based provisioning portal)

Image source: developers.sun.com



MIME types:
JAD → text/vnd.sun.j2me.app-descriptor
JAR → application/java-archive

Core packages:
javax.provisioning
javax.provisioning.adapter
javax.provisioning.matcher

Further reading: http://developers.sun.com/techtopics/mobility/midp/articles/ota/