An Introduction To Mobile Technologies and Services

by Michael Sharon, Co-founder / CTO, Socialight
1. What does “mobile” mean?
   • Components
   • Typical device features

2. The state of the industry
   • Operators, Devices, Openness, Ease of development

3. Mobile development options
   • Types of devices
   • OSes, languages, platforms
   • Applications
1. What does “mobile” mean?
Mobile

From the Latin *mobilis* - “to move”

“able to move freely or easily”

“able or willing to move freely or easily between occupations, places of residence and social classes”

Device, state of being, industry
Mobile device

Mobile, wireless or cellular phone - a portable, handheld communications device connected to a wireless network that allows users to make voice calls, send text messages and run applications.

AKA keitai, personal handy phone

WARNING: Jargon & Acronym laden
Multimedia Computer

Reinvented Phone

NOKIA
Connecting People
Many devices. Many manufacturers. Many formats.
<table>
<thead>
<tr>
<th>Feature phones</th>
<th>Smart phones</th>
<th>PDAs/handheld</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>$</td>
<td>$$</td>
</tr>
<tr>
<td>OS</td>
<td>Proprietary, Series40</td>
<td>S60, Windows Mobile, Linux</td>
</tr>
<tr>
<td>Applications</td>
<td>Java or BREW</td>
<td>Any</td>
</tr>
</tbody>
</table>
Mobile development ecosystem

- Air interface
- Data bearer
- Certification
- Deployment
- Platform
- Packaging
- Language
- Publishing
- Mobile operator
- Mobile UI
- Mobile OS
why mobile?

★ one handed use
★ limited (input, processing, battery life)
★ rich (sensors, usage)
★ small!
★ truly ubiquitous
Mobile phone capabilities

Bluetooth, WAP, WiFi

GPS, TDMA, PTT, GPRS, EDGE

CDMA, UMTS, W-CDMA

ringtones, colour, RFID

monochrome, images, NFC

text, graphics, WiMax

voice, images, WiMax

speaker, cameras, WiMax

microphone, cameras

Mobile evolution (briefly)
G - 1/2/3/4 G

G refers to the different generations of mobile devices.

First generation (1G) cellphones were analog devices. Second generation (2G) devices were digital, and third generation (3G) allows for voice, data and advanced services.
Early mobile phones
• Expensive
• In cars/trucks/briefcases
• Voice only
• First generation cellular networks
• Radio signals = analog
• Technologies - AMPS / DataTac
• First Blackberry (850)
• Voice + Limited data
• Second generation cellular networks
• Digital. Voice + SMS + Circuit switched data
• GSM, iDEN, CDMA, TDMA

2G
1990’s-now

• Marketing term
• GPRS, HSCSD, WiDEN
• Also EDGE, CDMA2000 1x-RTT

2.5G
1990’s-now
GSM

Global System for Mobile Communications

GSM is the most popular standard for mobile phones worldwide used by 2.2 billion people on over 210 networks.*

US Operators = T-Mobile, Cingular

* according to this http://en.wikipedia.org/wiki/GSM
GPRS

General Packet Radio Services

A mobile data service for use on GSM networks.

Part of the 2.5G standards family
Integrated Digital Enhanced Network

A second generation (2G) mobile telecommunications standard developed entirely by Motorola.

US Operators = Sprint-Nextel / Boost
CDMA

Code Division Multiple Access

A second generation (2G) standard for mobile phones.

US Operators = Sprint, Verizon
• Third generation cellular networks
• Broadband data + voice, streaming video!
• W-CDMA (UMTS, FOMA), 1xEV-DO

• “high-speed broadband for data- and visual- centric information”
• Transmits data at 100mbps while moving and 1Gbs while standing still
some refreshing statistics

- 3.2m Blackberries
- 50m PDAs
- 70m iPods
- 190m Gameboys
- 820m PCs
- 1.5bn TV sets
- 2bn+ Mobile phones*

2. The State of the Industry
<table>
<thead>
<tr>
<th>Service</th>
<th>Cingular</th>
<th>Verizon</th>
<th>Sprint</th>
<th>T-Mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscribers</td>
<td>61m</td>
<td>59.1m</td>
<td>53.1m</td>
<td>25m</td>
</tr>
<tr>
<td>Technology</td>
<td>GSM</td>
<td>CDMA</td>
<td>CDMA/iDEN</td>
<td>GSM</td>
</tr>
<tr>
<td>Platform</td>
<td>J2ME</td>
<td>BREW</td>
<td>J2ME</td>
<td>J2ME</td>
</tr>
<tr>
<td>Openness</td>
<td>Open</td>
<td>Semi-Walled</td>
<td>Open</td>
<td>Semi-Walled</td>
</tr>
<tr>
<td>Network</td>
<td>2.5G/3G</td>
<td>2.5G/3G</td>
<td>2.5G/3G</td>
<td>2.5G/3G</td>
</tr>
<tr>
<td>Location</td>
<td>TDOA (no access)</td>
<td>A-GPS</td>
<td>A-GPS</td>
<td>TDOA (no access)</td>
</tr>
</tbody>
</table>
Sprint (Nextel + Boost), T-Mobile & Cingular* support J2ME

* 3 out of the 4 largest carriers (but who’s counting anyway?)
3. Mobile Development Options
Mobile Development in 2007 is kinda like the web in 1997
Anybody remember <blink>? <marquee>?
This is worse
1997

Netscape vs Microsoft

Proprietary features vs standards
<blink> vs <marquee>

Free environment

Free development tools

Clear development / deployment process

2007

Symbian vs Flash Lite vs Java ME vs Python vs BREW vs .NET vs WAP vs Palm

Platform features / standards
OEM APIs (Java)

$$ environment (contracts)

Mostly free development tools (except for BREW)

Convoluted development & painful deployment process
Java ME / J2ME

Java ME (formerly known as Java 2 Platform, Micro Edition or J2ME), is a collection of Java APIs for developing software on resource constrained devices such as PDAs, cell phones and other consumer appliances.
Flash Lite

Flash Lite is a development platform created by Macromedia, based on their hugely successful Flash web application platform.

v1.1 - most widely deployed, limited

v2.x - improved experience, language
Symbian

Operating system based on original PDAs from Psion. Largest installed base. Multiple versions customized for different manufacturers. Language = C++

UIQ - SonyEricsson
Series 60 - Nokia
MOAP - NTT Docomo FOMA
Python for Series 60

Open source scripting language ported by Nokia

Only on Series 60 smartphones

Python wrappers around low-level APIs, easy access to native OS features
BREW

Binary Runtime Environment Wireless

Proprietary mobile device platform developed by Qualcomm. Development language is C with C++ interfaces. Certification and development process is expensive.
**Wireless Application Protocol**

Originally used to describe lightweight protocol which used **Wireless Markup Language** (WML).

Currently used to refer to Mobile Web, which uses XHTML MP/Basic + CSS.
<table>
<thead>
<tr>
<th>Platform</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java ME</td>
<td>Second best reach, best overall development</td>
</tr>
<tr>
<td>Flash Lite</td>
<td>Good for graphics-heavy applications in supported markets</td>
</tr>
<tr>
<td>Symbian</td>
<td>Strong support from Nokia, best access to hardware</td>
</tr>
<tr>
<td>.NET</td>
<td>PocketPC + Windows Mobile Devices</td>
</tr>
<tr>
<td>BREW</td>
<td>The only option for CDMA networks</td>
</tr>
<tr>
<td>Python</td>
<td>Great for quick prototypes, still immature</td>
</tr>
<tr>
<td>WAP</td>
<td>Largest overall reach, lightweight functionality</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Platform</th>
<th>Language</th>
<th>X-Platform</th>
<th>Learning Curve</th>
<th>Emulator</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java ME</td>
<td>Java</td>
<td>Average</td>
<td>Average</td>
<td>Free</td>
<td>~1.5bn</td>
</tr>
<tr>
<td>Flash Lite</td>
<td>AS</td>
<td>Excellent</td>
<td>Average</td>
<td>With IDE</td>
<td>77-115m</td>
</tr>
<tr>
<td>Symbian</td>
<td>C++</td>
<td>Average</td>
<td>STEEP!</td>
<td>Free</td>
<td>120m</td>
</tr>
<tr>
<td>.NET</td>
<td>C#, C++, VB.NET</td>
<td>WM</td>
<td>STEEP!</td>
<td>IDE</td>
<td>4.5m</td>
</tr>
<tr>
<td>BREW</td>
<td>C++</td>
<td>CDMA only</td>
<td>STEEP!</td>
<td>Simulator</td>
<td>????</td>
</tr>
<tr>
<td>Python</td>
<td>Python</td>
<td>FREE</td>
<td>Gentle</td>
<td>Add-on</td>
<td>Nokia-only</td>
</tr>
<tr>
<td>WAP / Mobile Web</td>
<td>XHTML, WML</td>
<td>FREE</td>
<td>Gentle</td>
<td>Free</td>
<td>2bn+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Platform</th>
<th>GUI</th>
<th>Functionality</th>
<th>Phone Data Access</th>
<th>Developer Community</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java ME</td>
<td>2D/3D, Many widgets, Visual Form Builder</td>
<td>Varies by handset, no CellID, high res pics</td>
<td>Varies by handset, Optional APIs</td>
<td>Extensive</td>
</tr>
<tr>
<td>Flash Lite</td>
<td>2D/3D, Many widgets, Visual IDE</td>
<td>Partial through API</td>
<td>None</td>
<td>Extensive</td>
</tr>
<tr>
<td>Symbian</td>
<td>2D/3D, Many widgets, Visual Form Builder</td>
<td>No restriction</td>
<td>Simulator</td>
<td>Extensive</td>
</tr>
<tr>
<td>.NET</td>
<td>2D/3D, Many widgets, Visual Form Builder</td>
<td>Limited audio</td>
<td>Full</td>
<td>MSDN</td>
</tr>
<tr>
<td>BREW</td>
<td>2D/3D, Many widgets, uiOne</td>
<td>Operator dependent</td>
<td>Full</td>
<td>Limited</td>
</tr>
<tr>
<td>Python</td>
<td>2D Graphics, some widgets</td>
<td>Partial through API</td>
<td>Partial</td>
<td>Small, but growing</td>
</tr>
<tr>
<td>WAP / Mobile Web</td>
<td>Basic forms. Inconsistencies</td>
<td>Limited to browser</td>
<td>None</td>
<td>Extensive</td>
</tr>
</tbody>
</table>

Java ME (J2ME)
Java Sources

• Java Community Process - http://jcp.org
  • JSR specification requests
  • reference implementations

• Sun - http://java.sun.com
  • SDK, tools, community

• Manufacturer
  • SDKs, community, device emulators
Java VM

- Other Apps
- Virtual Machine (KVM)
- Native Apps
- Operating System
- Hardware
A typical Java ME stack

1. Configurations
   - specifies minimum Java technology that we can expect for certain devices
   - Includes language, virtual machine features, core libraries

2. Profiles
   - layer defining APIs and specifications for a particular device or market - MIDP, FP
   - MIDlets

3. Optional Packages
   - includes additional functionality only supported by certain devices - e.g. Bluetooth API, Location API
1. Configurations: CLDC

Connected Limited Device Configuration

- specifies environment for mobile phone, pagers
- 160-512k of memory for Java
- limited power / batteries
- intermittent, low-bandwidth connectivity

CLDC 1.0
- May 2000, JSR 30
- java.lang

CLDC 1.1
- Dec 2002, JSR 139
- adds floating point support
- bug fixes
2. Profiles: MIDP

Mobile Information Device Profile

**MIDP 1.0**
- December 2000, JSR 37
- java.microedition.midlet
- java.microedition.rms
- java.microedition.lcdui
- java.microedition.io.HttpConnection

**MIDP 2.0**
- Nov 2002, JSR 118
- java.microedition.media
- java.microedition.lcdui.game

**MIDP 3.0**
- Q3 2006? No! Sometime 2007...
3. Optional Packages

**Bluetooth API** (JSR 82)
- communication with Bluetooth devices

**Wireless Messaging API** (JSR 120, JSR 205)
- SMS, MMS, multi-part messages

**Mobile Media API** (JSR 135)
- audio, video and multimedia

**Location API** (JSR 179)
- interface to location services
MIDP 3.0
AKA “The Future”

- Background MIDlets (remember TSRs?)
- Drawing to secondary displays
- Improved large screen support
- Auto-start MIDlets
- And much more... to forget about for the moment
MIDlets

MIDlets are like Java applets for mobile devices.

Has a lifecycle with four stages, created, started, paused, destroyed.
Applications
Mogi, item hunt
A new collection game

A game where players move outside, pick-up virtual items through their mobile phone interface and then trade with other players to complete collections. The goal is to get the maximum points completing collections.

It is based on players' location. From the Web interface, players see in real time, on a 3D map, the positions of connected players as well as collection items. From both interfaces, players trade the items picked up with the mobile.

Mogi is a community game, featuring a complete IM system. A web player might help a mobile player by clicking on its character on the map and sending "Lucky you! North, close to you, lies a rare item. Get it, get it! :)" which will pop on the screen of the mobile player.

Let's play!

An Ezplus game

Supported types: GPS, J2ME AU Phones (KDDI, JAPAN)
Supported terminals: A5401, A5402, A5305, A5303, A5302, A5301, A3015, A3014,

http://www.mogimogi.com/
Wayfinder.com » Index

Logga in på MyWayfinder.com

avläsarnamn: lösennord: 

Wayfinder navigator™

» Navigation i mobilen
» Röstinstruktioner
» Kartvisning
» Enkel att använda

Köp via telefon: 0200-89 91 51

Klicka här för mer information

Prova gratis

Kompatibla telefoner

Karttäckning

Butiksadresser

http://www.wayfinder.com/
Synchronize your phone with Google Calendar

Carry your Google Calendar in your pocket! GCalSync is an open source application that lets you do a two-way synchronization between Google Calendar and your phone's built-in calendar. Download events to your phone, or add an event on your phone and upload it to Google Calendar.

To install, visit http://wap.gcalsync.com with your phone's browser.

You can also download the files here if you prefer to install via USB, Bluetooth or infrared (see your phone's manual for how to do this).

gcalsync.jad gcalsync.jar signed version - for most phones

gcalsync.jad gcalsync.jar unsigned version - try this if your phone will not let you install the signed version
gcalsync.jad gcalsync.cod for Blackberry

NEW: Version 1.1.1 released, a bug fix to the first version after I open sourced GCalsync. Source code is available on SourceForge. New in version 1.1:

• Time zone options: you can specify an offset if you have problems with events being a couple of hours off.
• Download/upload options: choose if you want to just download events from GCal, just upload events from the phone, or both.

Known problems:

• Changes and deletions on the phone are not uploaded. I am getting error messages from GCal when I do this so it is disabled for now.

On some phones (Nokia Series 60) events get duplicated in GCal every time you sync. Use the new
http://www.mobup.org

Installing Mobup

Flickr from your mobile device built using the Flickr APIs. Once installed it gives you the possibility to shoot your photo and add title, tags and description, manage sets and groups from the same applications with optimal user experience and to post the shoted photo on your blog.

Mobup is being developed as by the CTIDND team and a bunch of other programmers;

Mobup is actually in public beta: thanks to the group of volunteers that daily apply for testing (if you want to be included just send us an email specifying your Java phone model).

Mobup source code is now available for free under GNU General Public License and is hosted on Sourceforge, the world's largest Open Source software development web site. (Visit developer section for more infos)

Latest Mobupped shots

Opzioni
Python for Series 60
What is Python?

- Created 1990 by Guido van Rossum
- Interpreted, object oriented programming language
- Very powerful language + terse syntax.
- Modules, classes, exceptions, dynamic typing
<table>
<thead>
<tr>
<th>Java</th>
<th>Python</th>
</tr>
</thead>
<tbody>
<tr>
<td>statically typed</td>
<td>dynamically (&quot;duck&quot;) typed</td>
</tr>
<tr>
<td><code>String blah = &quot;&quot;;</code></td>
<td><code>blah = &quot;string&quot;</code></td>
</tr>
<tr>
<td>verbose</td>
<td>concise</td>
</tr>
<tr>
<td><code>public class HelloWorld</code></td>
<td>print &quot;Hello World&quot;</td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>public static void main</td>
<td>print &quot;Hello World&quot;</td>
</tr>
<tr>
<td>(String[] args) {</td>
<td></td>
</tr>
<tr>
<td>System.out.println(&quot;Hello, world!&quot;);</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>Java ME</td>
<td>Python S60</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>freshly open source</td>
<td>open source</td>
</tr>
<tr>
<td>broad manufacturer support</td>
<td>Symbian Series60</td>
</tr>
<tr>
<td>complex, multiple APIs (High Level, Low Level), confusing exception model, runs in sandbox</td>
<td>extremely terse. no checked exceptions. uses Python standard library. simpler APIs, C++ wrapper</td>
</tr>
</tbody>
</table>
Capabilities of PyS60

- GUI: Menu, Forms, Listboxes, Input fields, Dialogs, Notes
- Graphics: - color, font and style attributes, - direct-screen drawing, - displaying images and icons
- Key-down and key-up events
- Sockets: **TCP/IP, Bluetooth** (RFCOMM, OBEX)
- Messaging (**SMS**) + accessing the Inbox
- Networking (**HTTP, FTP, ...**)
- Access to file system, file reading, XML, RSS
- Access to camera, telephone
- Access to calendar, contacts, sysinfo
- Location (cell-id)
- Content handler (download + open videos..)
- Python extensions can be written in C++
- Package scripts into standalone applications - (using SIS files)
WAP
The birth of WAP

- The end of the 1990’s:
- Data service bearers available: CSD (circuit switched data/dialup)/CDPD
- Date connection speeds: CSD=9.6kbs/CDPD=14.4kbs
- Light weight protocol needed to transfer data.
• First generation cellular networks
• Radio signals = analog
• Technologies - AMPS / DataTaq
• First Blackberry (850)
• Voice + Limited data
Enter WAP

• Enter WAP, a light weight protocol stage left.
• Good for data speed at that time
• WAP = Wireless Application Protocol
• Like HTTP with extra bits stripped out
• WAP Gateway (GW) handles translation
• Limited markup language resulted in
  • HDML - Handheld Device Markup Language
  • WML (established by the WAP Forum)
• Second generation cellular networks
• Digital. Voice + SMS + Circuit switched data
• GSM, iDEN, CDMA, TDMA

2G
1990’s-now

• Marketing term
• GPRS, HSCSD, WiDEN
• Also EDGE, CDMA2000 1x-RTT

2.5G
1990’s-now
WAP 2.0 (circa 2002)

- Data service bearers available: GPRS (54kbs)
- Development of 3G networks leads to enhancement of languages
- WAP 2.0 and XHTML-MP released by the WAP forum.
- Smarter phones + faster data (3G).
- WAP GW resembles typical Proxy Server
- WAP GW is largely for legacy device support (WAP 1.1 devices)
• Third generation cellular networks
• Broadband data + voice, streaming video!
• W-CDMA (UMTS, FOMA), 1xEV-DO

4G
the future!

• “high-speed broadband for data- and visual- centric information”
• Transmits data at 100mbps while moving and 1Gbs while standing still
## WML vs XHTML

<table>
<thead>
<tr>
<th>Feature</th>
<th>WML 1.x</th>
<th>XHTML-MP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standards Body</td>
<td>WAP Forum (defunct)</td>
<td>W3C + OMA</td>
</tr>
<tr>
<td>Content displaying</td>
<td>Content + layout in same document. Tailored separately for different devices.</td>
<td>Content + layout separate. Can be rendered separately.</td>
</tr>
<tr>
<td>Content Encoding</td>
<td>Binary</td>
<td>No encoding required</td>
</tr>
<tr>
<td>Document Layout control</td>
<td>Basic</td>
<td>Advanced layout with CSS</td>
</tr>
<tr>
<td>Colour control Support</td>
<td>Only colour images, no colour control for fonts, backgrounds, borders etc.</td>
<td>Full support with CSS, fonts, backgrounds, borders</td>
</tr>
<tr>
<td>Data bearer</td>
<td>WAP</td>
<td>Wireless profile - TCP/IP</td>
</tr>
<tr>
<td>Java</td>
<td>WAP</td>
<td></td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Complex syntax, powerful language</td>
<td>Simple syntax, not so powerful</td>
<td></td>
</tr>
<tr>
<td>Download apps</td>
<td>Use built in browser (no download necessary)</td>
<td></td>
</tr>
</tbody>
</table>

```java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

```html
<p>Hello, WAP</p>
```
Mobile application development can be challenging.
Start small, keep it simple, add constraints
Choose your platform wisely
Thanks!
Questions? Comments? Suggestions?

Michael Sharon

646 591 3681

michael@socialight.com