SOCIALIGHT

An Introduction To Mobile Technologies and Services

by Michael Sharon, Co-founder / CTO, Socialight

Overview

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

NOKIA
Connecting People

Reinvented
Phone

Many devices. Many manufacturers.
Many formats.

NOKIA
NOKIA
SAMSUNG
NOKIA
NOKIA

Motorola

RIM

Samsung

Kyocera

Palm

Nokia

BenQ

Fujitsu

Mobile device manufacturers

Sanyo

Sharp

LG

SonyEricsson

Apple

	Feature phones	**Smart phones**	**PDAs/ handheld**
Price	$	$$	$$$
OS	Proprietary, Series40	S60, Windows Mobile, Linux	PalmOS, PocketPC
Applications	Java or BREW	Any	Any

Mobile development ecosystem

Publishing

Certification

Air interface

Mobile operator

Data bearer

Mobile UI

Deployment

Mobile OS

Packaging

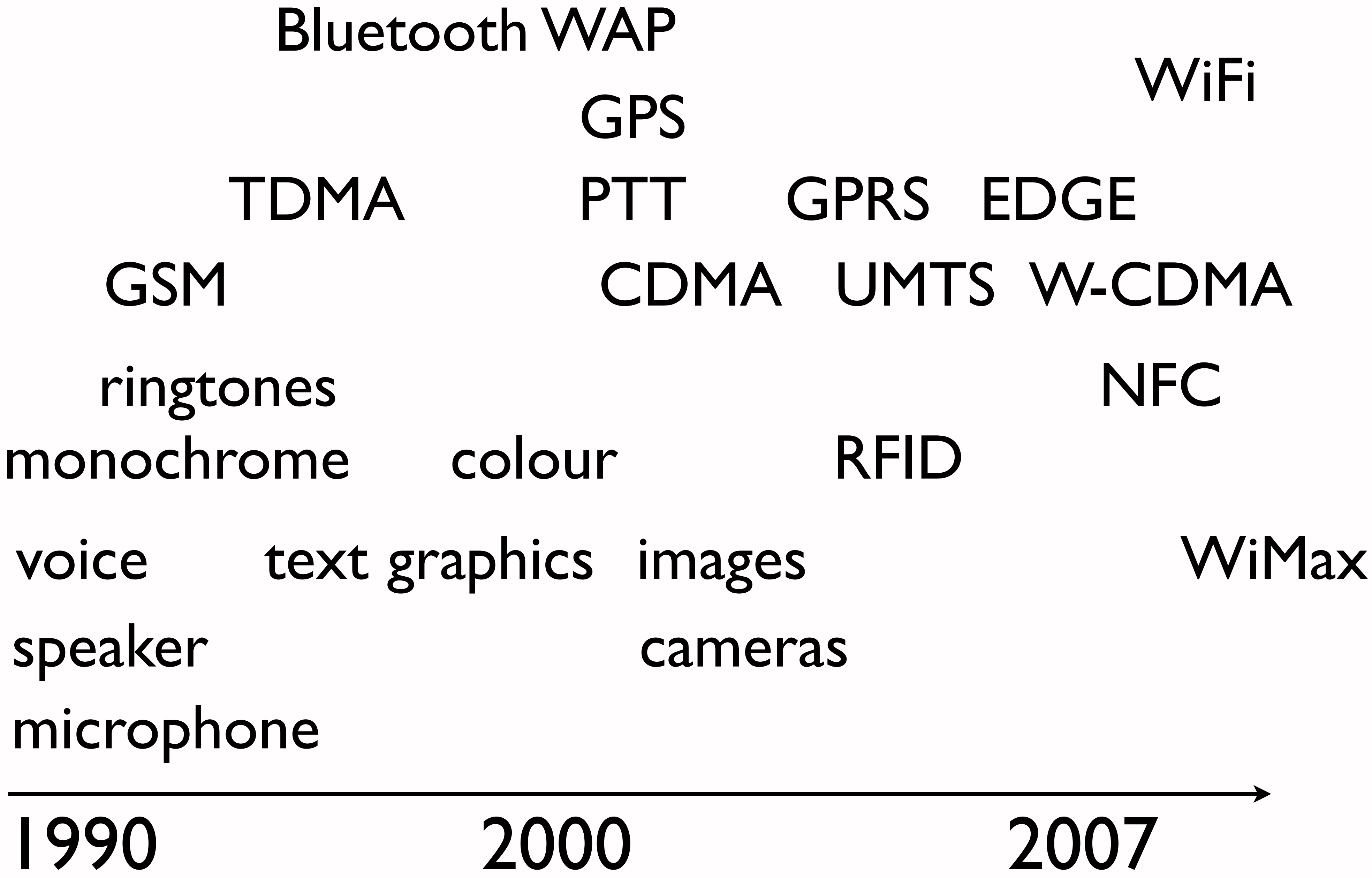
Platform

Language

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

GSM CDMA UMTS W-CDMA

ringtones NFC

monochrome colour RFID

voice text graphics images WiMax

speaker cameras

microphone

1990 2000 2007

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.

0G

1946-1980's

Early mobile phones

- Expensive
- In cars/trucks/briefcases
- Voice only

1G

1980’s-now

- First generation cellular networks
- Radio signals = analog
- Technologies - AMPS / DataTac
- First Blackberry (850)
- Voice + Limited data

2G

1990’s-now

- Second generation cellular networks
- Digital. Voice + SMS + Circuit switched data
- GSM, iDEN, CDMA, TDMA

2.5G

1990’s-now

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

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

iDEN

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

3G

2004-now

- 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

some refreshing statistics

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

Source: Charlie Schick's blog - http://cognections.typepad.com/lifeblog/2006/08/eh_kinda_quiet_.html

2. The State of the Industry

Operators in the US

Service	Cingular	Verizon	Sprint	T-Mobile
Subscribers	61m	59.1m	53.1m	25m
Technology	GSM	CDMA	CDMA/ iDEN	GSM
Platform	J2ME	BREW	J2ME	J2ME
Openness	Open	Semi-Walled	Open	Semi-Walled
Network	2.5G/3G	2.5G/3G	2.5G/3G	2.5G/3G
Location	TDOA (no access)	A-GPS	A-GPS	TDOA (no access)

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	2007
Netscape vs Microsoft	Symbian vs Flash Lite vs Java ME vs Python vs BREW vs .NET vs WAP vs Palm
Proprietary features vs standards <blink> vs <marquee>	Platform features / standards OEM APIs (Java)
Free environment	$$ environment (contracts)
Free development tools	Mostly free development tools (except for BREW)
Clear development / deployment process	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 **R**untime **E**nvironment **W**ireless

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

WAP

Wireless **A**pplication **P**rotocol

Originally used to describe lightweight protocol which used **W**ireless **M**arkup **L**anguage (WML).

Currently used to refer to Mobile Web, which uses XHTML MP/Basic + CSS.

Platform	Overview
Java ME	Second best reach, best overall development
Flash Lite	Good for graphics-heavy applications in supported markets
Symbian	Strong support from Nokia, best access to hardware
.NET	PocketPC + Windows Mobile Devices
BREW	The only option for CDMA networks
Python	Great for quick prototypes, still immature
WAP	Largest overall reach, lightweight functionality

sources: http://www.biskero.org/?p=430, http://alindh.iki.fi/2006/06/27/mobile-platform-statistics/, http://en.wikipedia.org/wiki/Mobile_development

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

sources: http://www.biskero.org/?p=430, http://alindh.iki.fi/2006/06/27/mobile-platform-statistics/, http://en.wikipedia.org/wiki/Mobile_development

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

sources: http://www.biskero.org/?p=430, http://alindh.iki.fi/2006/06/27/mobile-platform-statistics/, http://en.wikipedia.org/wiki/Mobile_development

Java ME (J2ME)

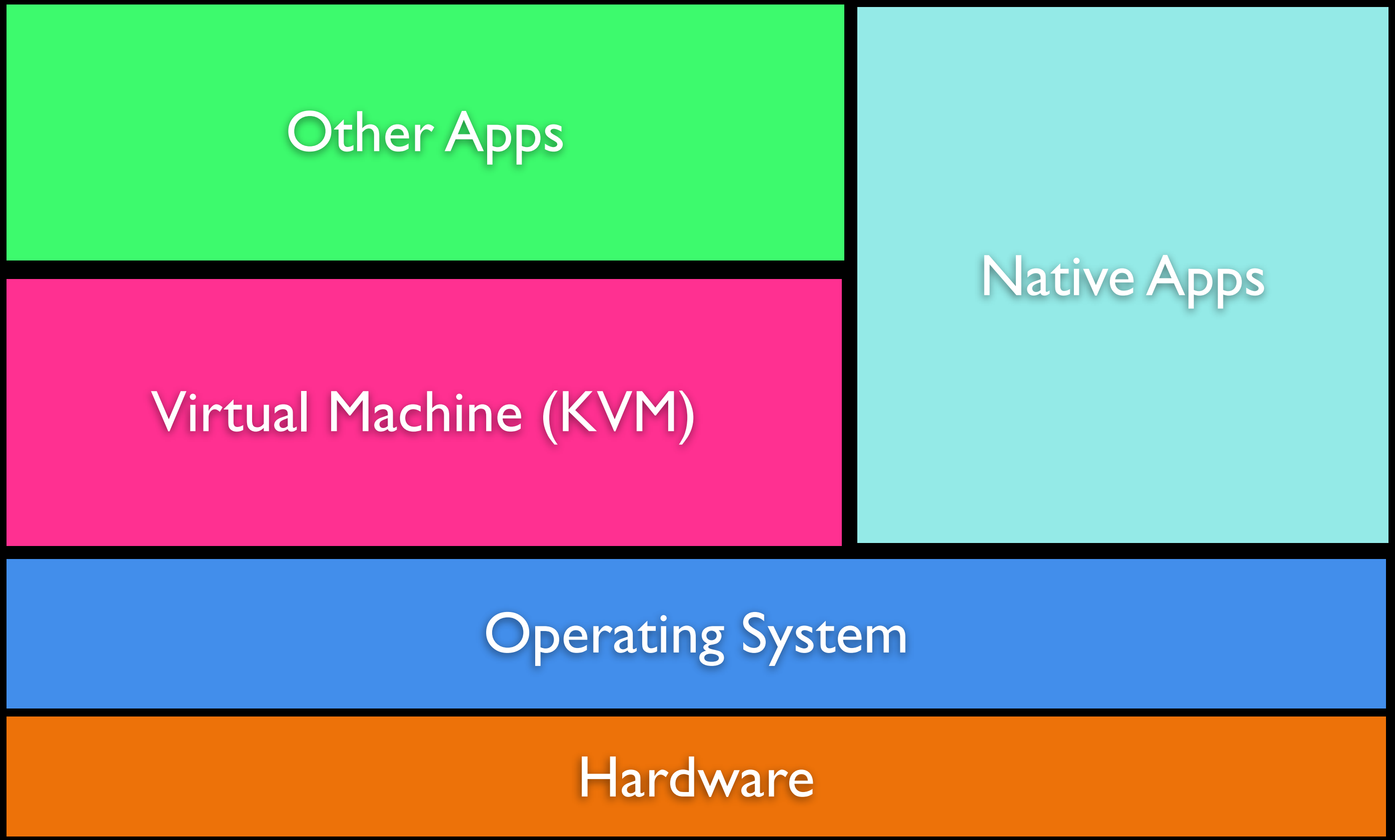
Java Sources

- 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

Native Apps

Virtual Machine (KVM)

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

Games
Pang
The Sims2

Mapping
Google Maps
mGmaps
uLocate

Photos
Mobup
Shozu
Zonetag

Web
Opera Mini
GCalSync

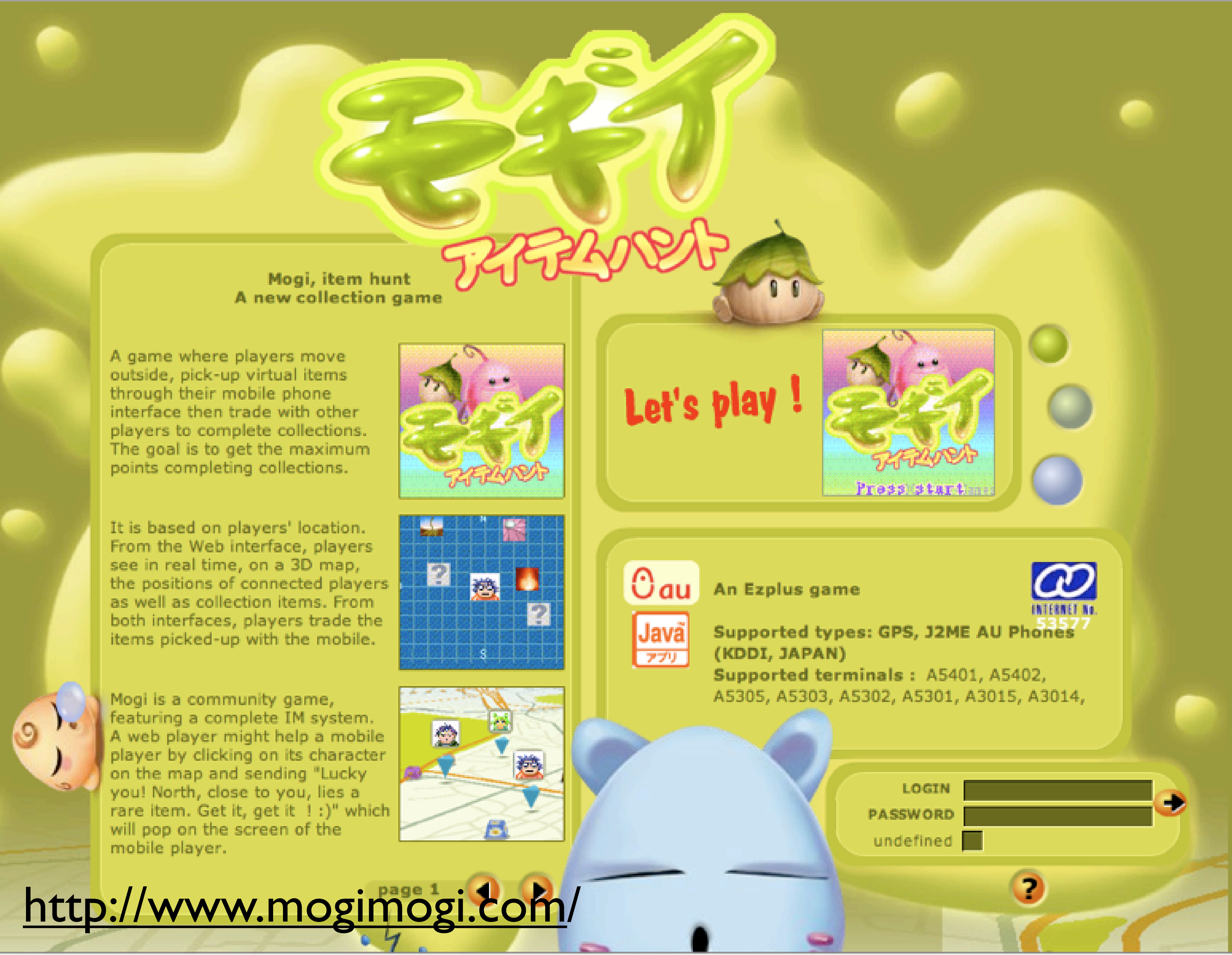
Mapping
Wayfinder

Art
Balldroppings

Social
BEDD
Flirtomatic
Loopt

RSS
Widsets
MobileGlu

Hybrids
MogiMogi
Socialight
Yahoo Go!



Mogi, item hunt
A new collection game

A game where players move outside, pick-up virtual items through their mobile phone interface 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.

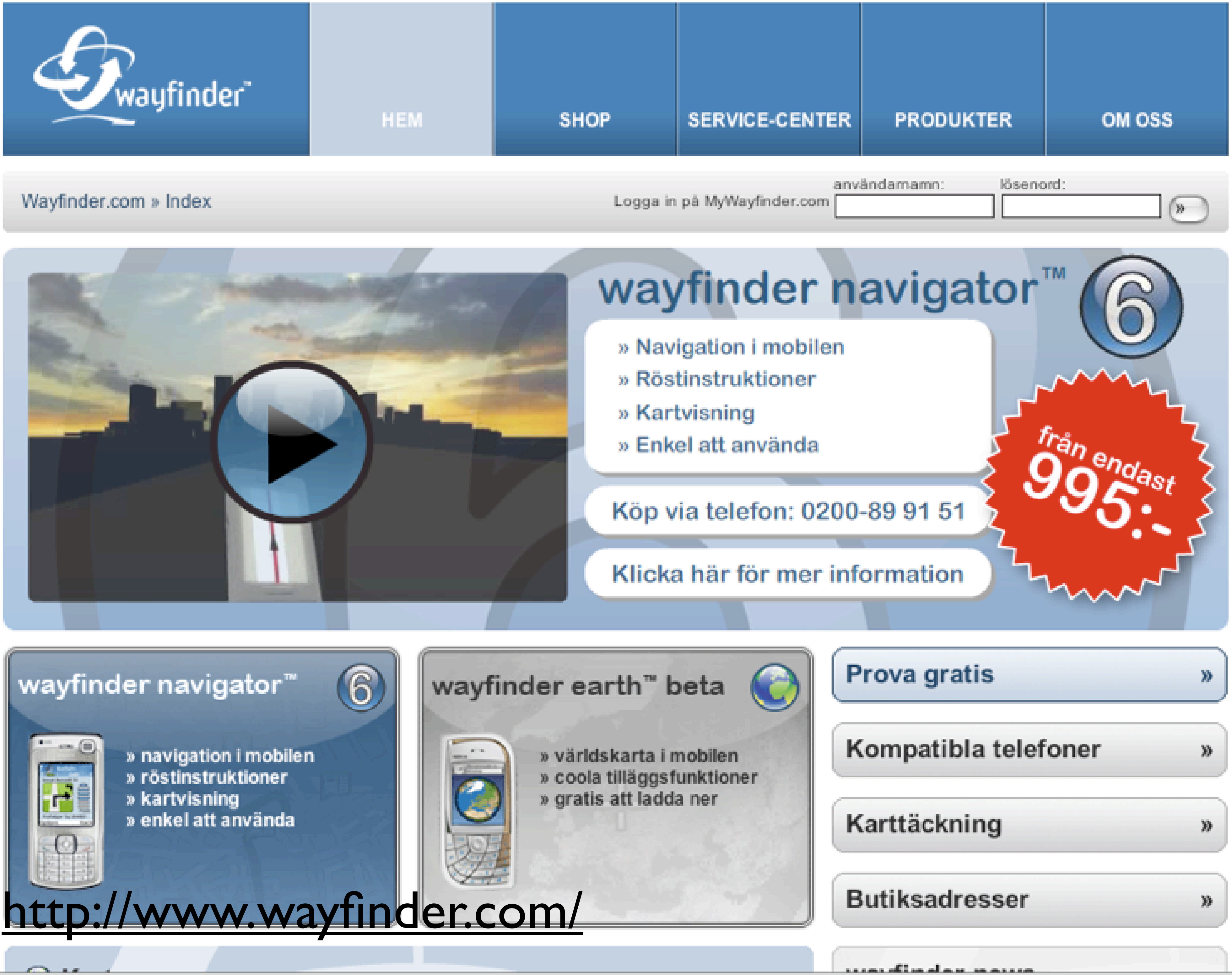
An Ezplus game

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

LOGIN
PASSWORD
undefined

page 1

http://www.mogimogi.com/



HEM | SHOP | SERVICE-CENTER | PRODUKTER | OM OSS

Wayfinder.com » Index

Logga in på MyWayfinder.com

användarnamn:

lösenord:

»

wayfinder navigator™ 6

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

Köp via telefon: 0200-89 91 51

Klicka här för mer information

från endast 995:-

wayfinder navigator™ 6

- » navigation i mobilen
- » röstinstruktioner
- » kartvisning
- » enkel att använda

wayfinder earth™ beta

- » världskarta i mobilen
- » coola tilläggsfunktioner
- » gratis att ladda ner

Prova gratis »

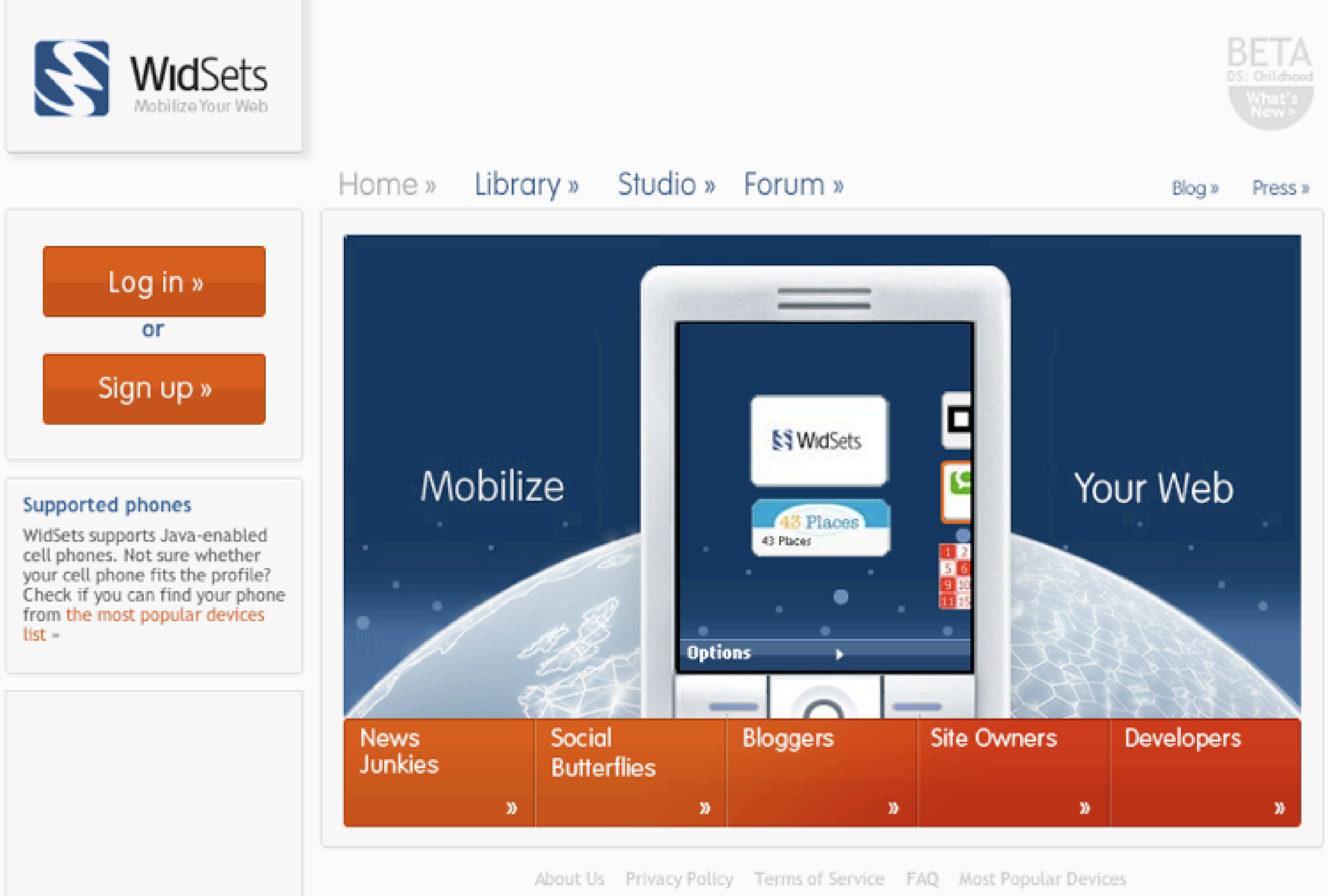
Kompatibla telefoner »

Karttäckning »

Butiksadresser »

wayfinder news

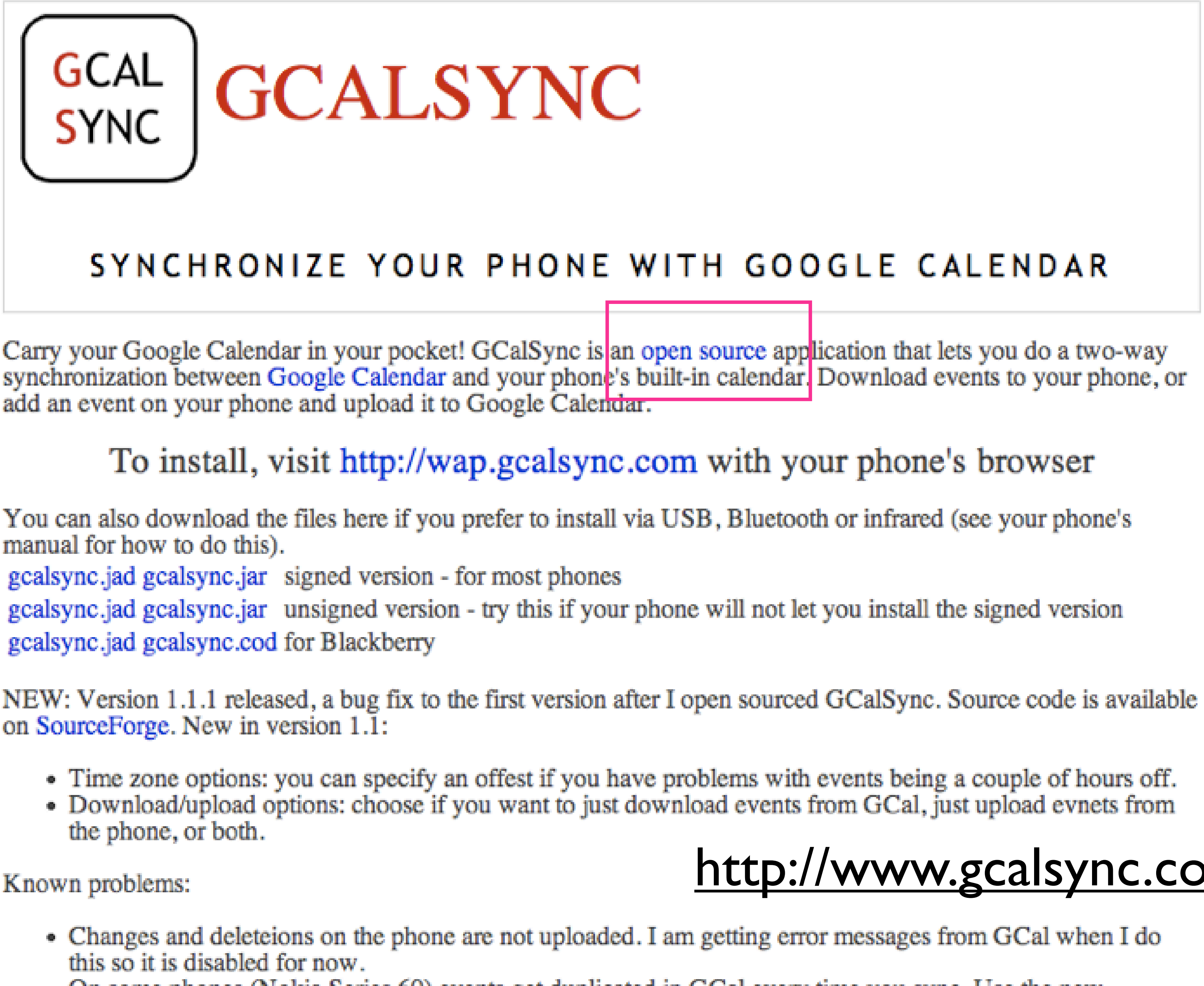
http://www.wayfinder.com/



Supported phones

WidSets supports Java-enabled cell phones. Not sure whether your cell phone fits the profile? Check if you can find your phone from the most popular devices list »

http://www.wayfinder.com/



GCAL SYNC

GCALSYNC

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 offest 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 evnets from the phone, or both.

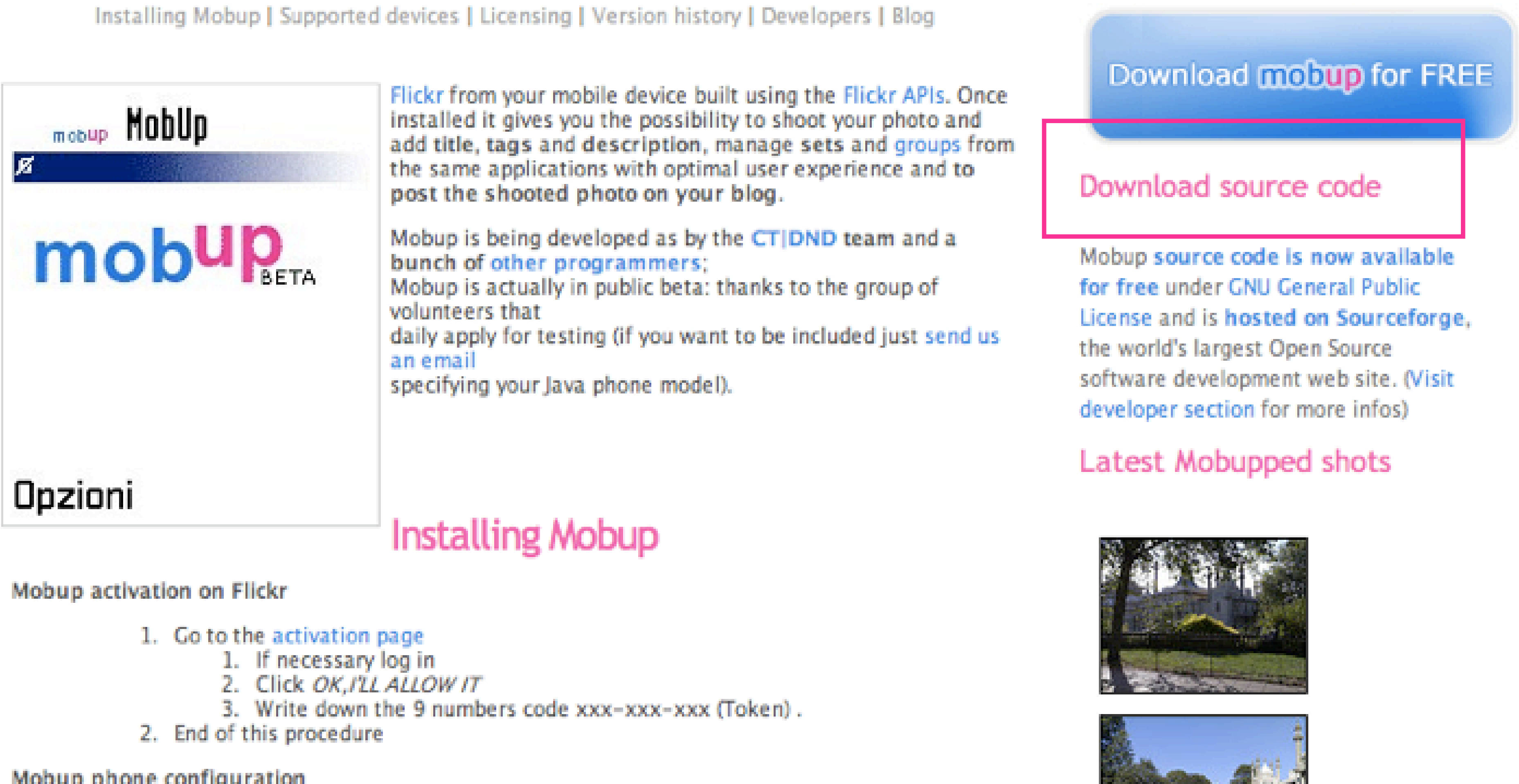
Known problems:

- Changes and deleteions 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.gcalsync.com

mobup beta

http://www.mobup.org



Installing Mobup | Supported devices | Licensing | Version history | Developers | Blog

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 shooted photo on your blog**.

Mobup is being developed as by the CT|DND **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).

Download mobup for FREE

Download source code

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

Installing Mobup

Mobup activation on Flickr

1. Go to the activation page
   1. If necessary log in
   2. Click *OK, I'LL ALLOW IT*
   3. Write down the 9 numbers code xxx-xxx-xxx (Token) .
2. End of this procedure

Mobup phone configuration

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**

Java	Python
statically typed *String blah = “”;*	dynamically (“duck”) typed *blah = “string”* *blah = 1*
verbose	concise

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

Java ME	Python S60
freshly open source	open source
broad manufacturer support	Symbian Series60
complex, multiple APIs (High Level, Low Level), confusing exception model, runs in sandbox	extremely terse. no checked exceptions. uses Python standard library. simpler APIs, C++ wrapper

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 connnection speeds: CSD=9.6kbs/ CDPD=14.4kbs
- Light weight protocol needed to transfer data.

1G

1980's-now

- First generation cellular networks
- Radio signals = analog
- Technologies - AMPS / DataTac
- 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)

2G

1990's-now

- Second generation cellular networks
- Digital. Voice + SMS + Circuit switched data
- GSM, iDEN, CDMA, TDMA

2.5G

1990's-now

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

WAP 2.0 (circa 2002)

- Data service bearers available: GPRS (54kbs)
- Development of 3G networks leads to enhancement of languges
- 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)

3G

2004-now

- 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

	WML 1.x	XHTML-MP
Standards Body	WAP Forum (defunct)	W3C + OMA
Content displaying	Content + layout in same document. Tailored separately for different devices.	Content + layout separate. Can be rendered separately.
Content Encoding	Binary	No encoding required
Document Layout control	Basic	Advanced layout with CSS
Colour control Support	Only colour images, no colour control for fonts, backgrounds, borders etc.	Full support with CSS, fonts, backgrounds, borders
Data bearer	WAP	Wireless profile - TCP/IP

Java	WAP
Complex syntax, powerful language	Simple syntax, not so powerful
Download apps	Use built in browser (no download necessary)

public class HelloWorld
{
public static void main (String[] args)
{
System.out.println("Hello, world!");
}
}
	<p>Hello, WAP</p>

Mobile application development can be challenging.

Start small, keep it simple,
add constraints

Choose your platform wisely

Thanks!

SOCIALIGHT

Questions? Comments? Suggestions?

Michael Sharon

646 591 3681

michael@socialight.com