



# WidSets Platform for Mobile Web 2.0 Users and Developers

**Tero-Markus Saarimaa**

*Senior Manager, co-inventor of WidSets*

**Marko Lumivuori**

*Developer Community Specialist*



# Agenda 8.11.

## 30 min / T-M Saarimaa

- \* WidSets for Mobile2.0 - why & what?
- \* WidSets Service overview
- \* WidSets for developers - challenge for You !

## 60 min / Marko Lumivuori

- \* Development site ([dev.widsets.com](http://dev.widsets.com)) and SDK intro
- \* Getting started / How to develop for WidSets
- \* Live demo of SDK

# Background. Internet usage has evolved ...



Internet has evolved from a static information source...



WIKIPEDIA  
The Free Encyclopedia

Blogs  
flickr<sup>BETA</sup>

You Tube

skype™



...into Web 2.0 – enabling people to participate and share

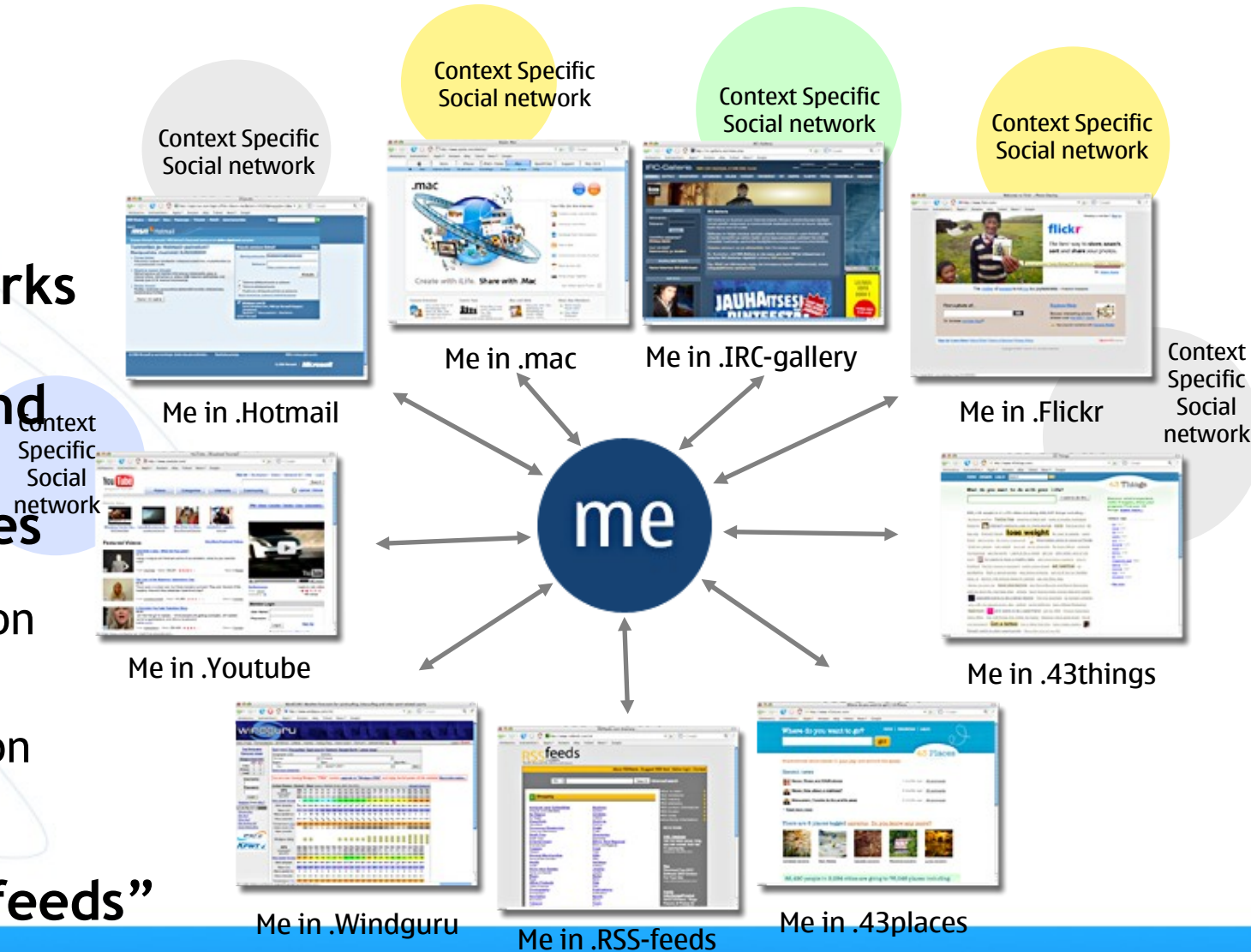
WiFi<sup>TM</sup>  
ZONE



...Access to internet is changing

# ... and become very complex for mobile browsing UI

- \* Many identities
- \* Many contexts
- \* Many social networks
- \* Many interfaces and information sources
  - \* Too much information
  - \* Difficult to find relevant information
- \* Huge amount of “feeds”





# WidSets service

## Overview



# WidSets solution overview



WidSets  
service

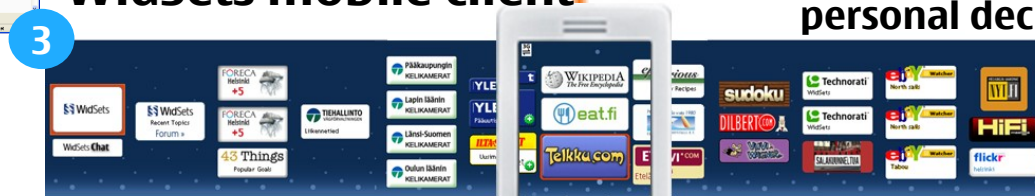
1



2

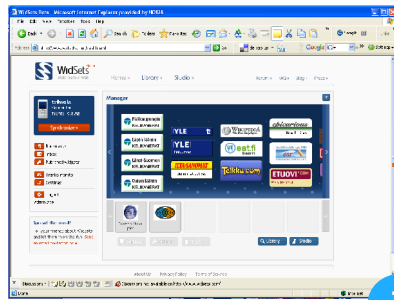
WidSets mobile client

4



- WidSets servers poll Internet for new info.
- WidSets servers push notifications and new content to mobile. Pictures are scaled down to suitable screen size resolution to save bandwidth and mobile's memory.
- Every user has a totally user definable personal deck on mobile and in web.

WidSets website



3

Personal web account to manage, explore and publish content

- WidSets client gives user one click direct access to preferred content and web service.

# WidSets service, mobile + PC

Use on your Mobile



Manage on your PC



# WidSets, widgets, WidSets widgets ?



1. WidSets' mobile client is a Java application to host widgets
2. WidSets' widgets are mobile mini-applications written in Java like scripting language to perform a specific task.

They can be dynamically downloaded to and removed from mobile by user without need to download the whole WidSets mobile client.

4. The size of your personal “virtual dashboard” is limited only by the mobile’s memory limit



# WidSets is about being device agnostic

Besides **NOKIA** Connecting People mobiles users run WidSets on the following phones (examples, December 2006)



**MOTOROLA**

- \* RAZR, ROKR, SLVR
- \* V3, V3i
- \* V235, V300, V360, V500, V600
- \* E398, MPX220



Sony Ericsson

- \* K300i, K750i, M600i
- \* P900, P910i
- \* W300i, W800i, W810i
- \* Z1010



- \* SGH-D500, SGH-D600
- \* SGH-E620
- \* SGH-X480C, SGH-X495
- \* SGH-ZV50

**WidSets runs currently on 300+ phone models from more than 15 manufacturers**

# WidSets website supports the mobile experience

## Drag and drop management of personal mobile portal

## Personal Traffic monitoring per widget

## Personal mobile bookmarking

## And community caring for itself

The screenshot shows the 'Widget Library' page. At the top, there's a search bar and navigation links for Home, Library, Studio, Forum, FAQ, Blog, and Press. Below the search bar, there are 'Slick Picks' featuring various widgets like 'Tech Crunch', 'WidSets Blog', 'flickr', and 'Dribbble'. A sidebar on the left contains navigation options: Bookmarks (7 new), Inbox (1 new), Published widgets, Traffic monitor, Settings, Login, and Admin site. There's also a 'Spread the word!' section with a link to send an email invitation.

The screenshot shows the 'Manager' interface. At the top, it says 'Changes saved 7'. Below this, there's a grid of widgets including 'ETUOVI.COM', 'Nytölkö', 'EST', 'eat.fi baarit', 'SURF+FI', 'North call: r type', 'Sudoku', 'Tabou', 'DILBERT.COM', and 'Bic'. At the bottom, there are more widgets like 'POPSCI.COM', 'epicurious', and 'Yourlogo'. A sidebar on the left shows 'tsihvola Connected Traffic: 10.4 MB' and a 'Synchronize' button. Below that, there are links for Bookmarks, Inbox, Published widgets, Traffic monitor, Settings, Logout, and Admin site. A 'Spread the word!' section is also present.

The screenshot shows the 'Personal Traffic monitoring per widget' interface. It features a 'Traffic monitor' section with a 'Total Traffic' gauge showing 1140 MB. Below this is a 'Traffic Remainder' section with a gauge showing 1500 KB. To the right, there's a 'Widget Set Traffic' section with a list of widgets and their traffic usage, each with a 'Select' button. A sidebar on the left shows 'tsihvola Connected Traffic: 10.4 MB' and navigation options. A 'Spread the word!' section is also present.

## Widget publishing

The screenshot shows the 'Studio' interface. At the top, there's a 'Widget Templates' section with a 'Next' button. Below this is a 'Skin editor' section with a 'Next' button. To the right, there's a 'Promote Your Widgets' section with a 'Next' button. A sidebar on the left shows 'tsihvola Connected Traffic: 10.4 MB' and navigation options. A 'Spread the word!' section is also present.

The screenshot shows the 'Personal mobile bookmarking' interface. It features a list of bookmarks with columns for 'Bookmark', 'Widget', and 'Date'. The list includes various bookmarks like 'WidSets', 'WidSets Blog', 'WidSets Forum', etc. A sidebar on the left shows 'tsihvola Connected Traffic: 10.4 MB' and navigation options. A 'Spread the word!' section is also present.

# Bookmarking feature for full PC access

WidSets - Microsoft Internet Explorer

Address: [http://www.widsets.com/me\\_bookmarks.html](http://www.widsets.com/me_bookmarks.html)

WidSets

Phone not connected  
Update mobile  
Traffic: 8.8 MB  
Log out

Blog | FAQ | Partners | Press | Developers

My profile | Library | Forum | Explore

tmzmaa  
Click here to enter a slogan  
Member since 2006/12/02  
Page views: 7

Welcome  
Dashboard manager  
Create a new widget  
Edit widgets  
Publish to Library  
My published widgets  
Promote widget on your site  
Inbox  
**Bookmarks** (13)  
Settings  
Traffic monitor  
How others see me

## Bookmarks

Subscribe to your bookmarks RSS feed

Bookmarks	Widget	Date	Mark all
Google hyökkää Nokian Ovea vastaan <a href="#">Open »</a>	Taloussanomats rss	2007/11/05	<input type="checkbox"/>
It's official: Google announces open-source mobile phone OS, Android <a href="#">Open »</a>	Reddit	2007/11/05	<input type="checkbox"/>
Nokian musiikkikauppa aukei Britanniassa <a href="#">Open »</a>	Taloussanomats rss	2007/11/01	<input type="checkbox"/>
Open Social: a new universe of social applications all over the web <a href="#">Open »</a>	Tuomo's Jaiku	2007/11/01	<input type="checkbox"/>
Nokian haluama Navteq nousussa <a href="#">Open »</a>	Taloussanomats rss	2007/11/01	<input type="checkbox"/>
Apple's Leopard is Cool, but Time Machine Makes it Astonishing <a href="#">Open »</a>	digg	2007/10/31	<input type="checkbox"/>
Widget development contest open now in dev.widsets.com <a href="#">Open »</a>	WidSets Blog	2007/10/31	<input type="checkbox"/>
Details: Revealed: Google OpenSocial To Launch Thursday <a href="#">Open »</a>	Tuomo's Jaiku	2007/10/31	<input type="checkbox"/>
Open Social: a new universe of social applications all over the web <a href="#">Open »</a>	Tuomo's Jaiku	2007/10/31	<input type="checkbox"/>
Amazing Drum Tenor Solo <a href="#">Open »</a>	digg	2007/10/30	<input type="checkbox"/>
Leopard <a href="#">Open »</a>	Valve Blog	2007/10/30	<input type="checkbox"/>
Yksitoista esikoista tavoittelee HS:n kirjallisuuspalkintoa <a href="#">Open »</a>	kulttuuriuutiset	2007/10/27	<input type="checkbox"/>
Reaaliaikainen valokuvanäytely suomalaisuudesta <a href="#">Open »</a>	kulttuuriuutiset	2007/10/27	<input type="checkbox"/>
The Simpsons + Pulp Fiction [pic] <a href="#">Open »</a>	Reddit	2007/10/26	<input type="checkbox"/>

Delete

It's official: Google announces open-source mobile phone OS, Android - Microsoft Internet Explorer

Address: <http://arstechnica.com/news/ars/post/20071105-its-official-google-announces-open-source-mobile-phone->

ars technica  
the art of technology

Main | Business IT | Apple | Gaming

servercentral  
Colocation and network services provided by ServerCentral

Gulf Hurricane  
Help Support Health Clinics Provide  
www.DirectRe

Public Service Ads by Google

Home | News | Articles | Guides | Journals | Search

## From the News Desk

### It's official: Google announces open-source mobile phone OS, Android

By [Jacqui Chen](#) | Published: November 05, 2007 - 10:46AM CT

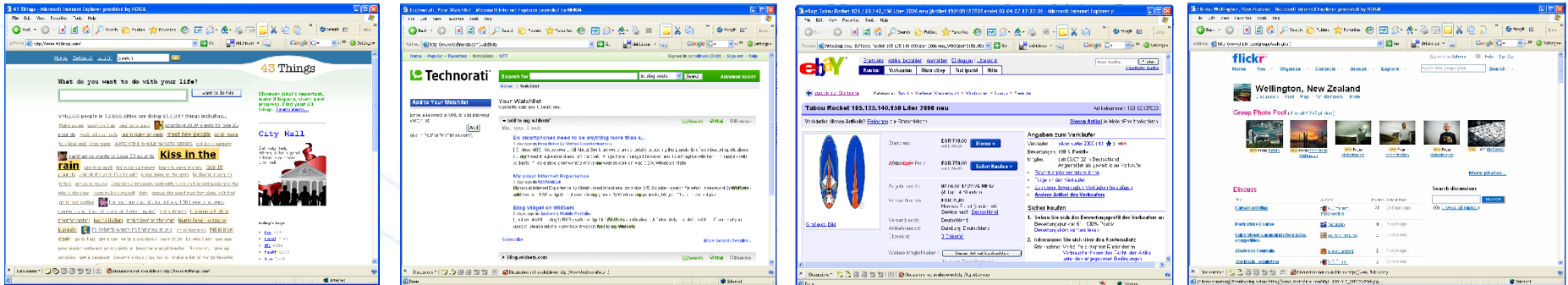
The Google Phone has arrived, sort of, but not in the long-rumored embodiment that many had expected. Google announced this morning that it has developed a new mobile OS called "Android"—a result of its acquisition of a mobile software company of the same name in 2005—that will allow the company to get Google's mobile apps into as many hands as possible starting in mid-2008. Android is Linux-based and open source, and aspects of the platform will be made available to handset manufacturers for free under the Apache license.

Google's handset partners upon launch will include Motorola, HTC, Samsung, and LG, confirming many of the [recent rumors](#) that Google would not be developing the hardware on its own. Google has a number of carrier partners worldwide as well, such as T-Mobile and Sprint in the US, T-Mobile/Deutsche Telekom in Europe, and China Mobile in China, to name a few. The whole thing comes as part of the [Open Handset Alliance](#)—also announced by Google today.

Google has chosen to launch Android in this way is because it wanted to put its focus on the platform for development of its mobile applications. Although Java is widely available on

Related Stories  
[Getting to the bottom of the gPhone rumors](#)

# WidSets is about newest personally relevant info pushed to you...



New comment, new blog entry, new item on auction, new picture, new anything in Web

3G 43 Things

Thank you  
Hmmm  
Untitled  
Untitled  
67675  
Untitled  
save yourself  
That's terrible...  
wow  
Untitled  
you can do it  
Lack of focus

Options

Send Back

3G 12.2007 08:25

Thank you  
by Dazee

Dazee commented on an entry by Dazee titled "snow\_on\_trees" regarding take my camera with me wherever I go so I can get those photos that keep getting away:

I had to hold the steering wheel with one hand and shoot the picture with the other. Probably not such a good idea.

Send Back

3G Technorati

Google mira al mundo de las...  
Pripomočki tudi za mobilne...  
Zen-zui mobile phone user i...

Opera Mini on I  
New Platform -  
Microsoft's new  
WidSets versus Z

Options

3G Technorati

28.03.2007 22:42

WidSets versus Zenzui:  
Mobile Widget  
Deathmatch?

... [WidSets Logo]The launch of Zenzui raises a key question: how does Zenzui differ from WidSets? Both offer widgets for mobile devices. There's much more info on WidSets.com about how it works, and WidSets has first-

Send Back

3G eBay

Tabou XRocket 61 Ltd.

Tabou Manta Slalom 75 Surfbo..

Tabou - SURF LEAS

Tabou COMBO lat

Tabou Wellenreit

SOUNDS AUDIOPHIL

Tabou Rocket 61 Ne

Tabou Wellenreiter

Tabou Wellenreiter

Tolle Sport Kombi v

Tabou Rocket 105

Options

3G 29.03.2007 08:58

Tabou XRocket 61 Ltd.

EUR 470,00

Angebotsende: Samstag Apr-07-2007 22:15:00 CEST

Sofort kaufen für nur: EUR 470,00

Sofort kaufen | Zur Liste beobachteter Artikel hinzufügen

Send Back

3G flickr helsinki

Options

3G flickr helsinki

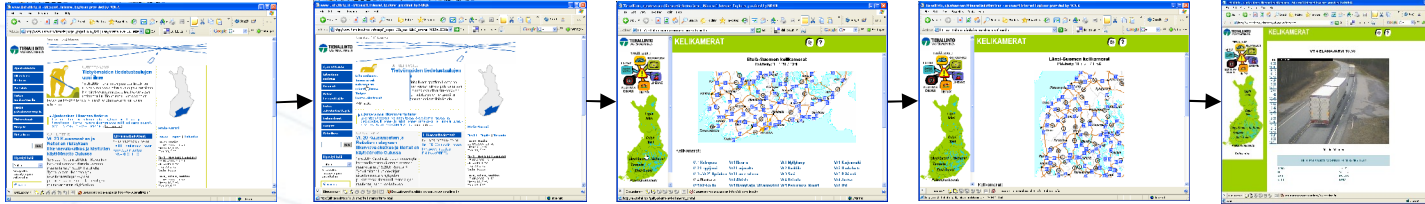
The Tsar and the church  
Tiago Cabral

Bookmark Back

...and WidSets finds it out for you and lets you know

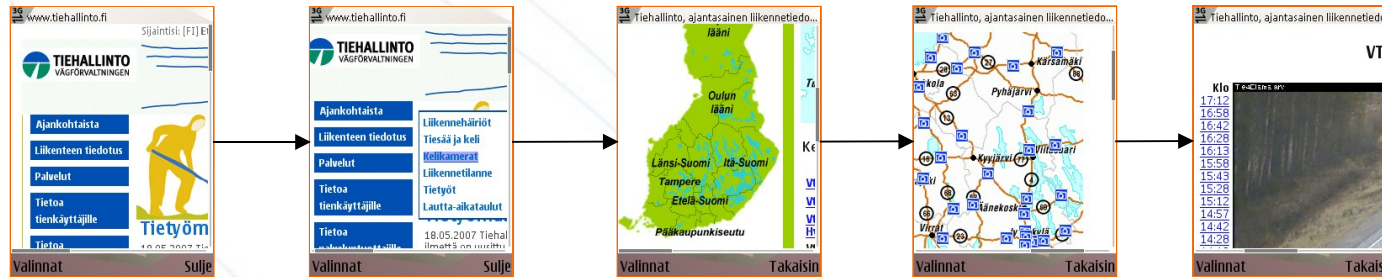
# WidSets is about optimised one click access to web

## Looking for a traffic cam in PC ?



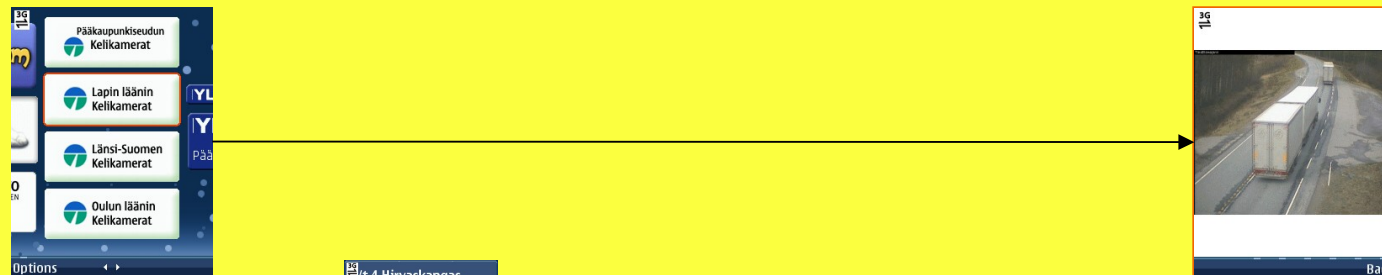
data traffic amount does not really matter and screen size is not an issue

## With (good) mobile web browser



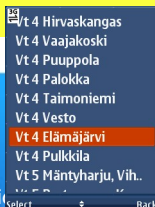
close to one megabyte data is transferred over the air, and lots of screen scrolling is needed

## With widgets



just what is needed for particular screen size is transferred – no scrolling, it just fits

## Direct selection of cams from options



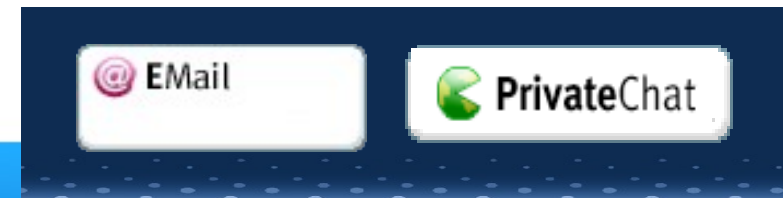
# WidSets is about variety of content



- \* News
- \* Blogs
- \* Search
- \* Pictures
- \* Games
- \* Information
- \* Social networks
- \* Fun/Entertainment
- \* TV Guide, Teletext



... and communication



# WidSets is about UGC

- \* Currently 2800+ widgets in library, ~90% created by users
- \* User created applications for the mobile
  - \* Simple tools and templates for users
    - \* Free to use
    - \* Publish to Library & share
  - \* Create a widget for your business, own blog
- \* SDK for developers
- \* Platform features for sharing and community rating, commenting



# Mobile Widgets - Opportunity for Developers

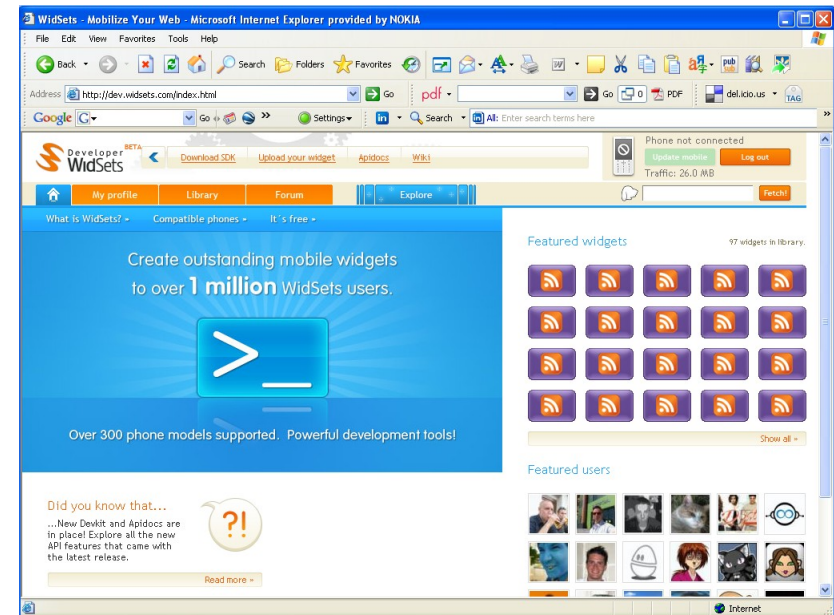




# WidSets for developers & other services

- \* Free widget development environment and documentations for WidSets service in <http://dev.widsets.com>

- \* WidSets offers unique global widget distribution platform
  - \* Developed widgets have a global reach to whole Java MIDP2 enabled mobile device base
  - \* Developed widgets can be centrally redistributed to all users in case of fixes, change of functionality etc.



- \* WidSets offers tools to direct market widgets to specific target groups

# Outlook: Mobile Widgets

- \* “In the U.S. marketing spending on mobile widgets will reach \$500 million by 2010, up from about \$2 million today, figures Scott Ellison, an analyst with consultancy IDC. By then, mobile widgets will account for as much as 15% of total mobile marketing spending”
- \* “The real promise of mobile marketing is the ability to create a very personal experience between a customer and a brand,” Ellison says. And since widgets are selected based on an individual consumer's interests and preferences, they're "about as personal as you can get”

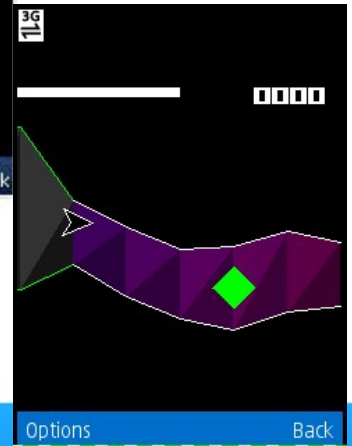
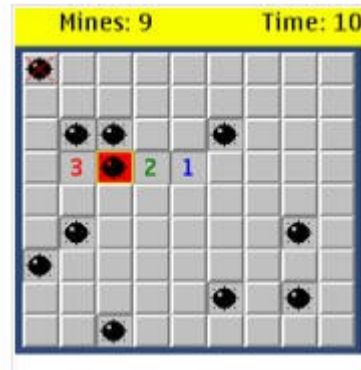
*Extract from Business Week, April 27th, 2007*

# WidSets' potential for developers

- \* Mainstream business model is been build around traditional banner advertising within widgets
- \* One WidSets' revenue model is to offer ways to market widgets for desired user base within that free direct distribution channel
- \* WidSets offers an easy platform to test ideas, applications and real target markets
  - \* distribution, upgrading, reporting and collecting user feedback are all build into the system
  - \* widget distribution channel is free for all developed widgets and to the total WidSets user base in device independent way
- \* Significant user base in Russia
- \* Example: Russian forum widget - 17.000+ users locally

# Developer Challenge

- \* New promising technology & ecosystem
- \* Visibility, credibility as a mobile developer
- \* Competitions, awards :)
- \* See [blog.widsets.com](http://blog.widsets.com)
- \* Thank you!





# Developer WidSets

<http://dev.widsets.com>

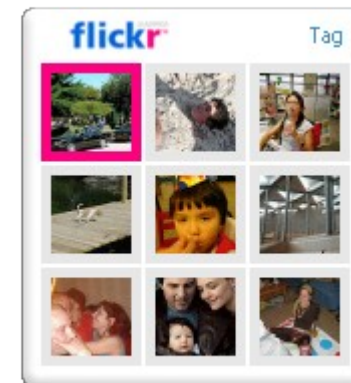
Marko Lumivuori

[marko.lumivuori@nokia.com](mailto:marko.lumivuori@nokia.com)



# Introduction to Widsets development

- \* **WidSets:** a service system that consists of a client side application (WidSets mobile client) and a server (WidSets server) providing services to widgets.
- \* **Widgets** for WidSets are mobile applications, which run on the WidSets Virtual Machine residing on the mobile device.
- \* **Types of widgets:** *standalone* applications (e.g. a calculator, a game etc.) or *server-based* client applications: RSS feeds, image viewers (Flickr etc.) depending on how they are implemented.
- \* **WidSets client:** the WidSets application that hosts widgets when they are installed and run in a mobile device



# Introduction to WidSets development contd.

- \* **WidSets Virtual Machine** is an installable client application for any mobile device, which supports Java MIDP 2.0.
- \* **WidSets server:** a server, where widgets are to be uploaded and compiled before they are downloaded into a mobile device. Also widget server side services are done on WidSets server. Server side services must be defined in your widget.xml file before using. Current services are the http service for fetching http content from the internet to widget, image service for scaling images and syndication/webfeed service for accessing RSS and Atom feeds.
- \* **WidSets manager:** a web application, that provides WidSets users with utilities functions such as synchronizing widgets, managing user's account information, monitoring traffics etc.
- \* **WidSets library:** a server-side library where published widgets reside.
- \* **WidSets Scripting Language (aka Helium):** a scripting language used in composing a widget's functions.

# Supported widget types

- \* **Standalone widget**
- \* In example games, like Sudoku, Fruitsets etc. Can work without network connection.
- \* **Syndication widget** (news readers like BBC, CNN, etc.)
- \* The syndication service gives a common interface to **RSS** and **ATOM** type online feed content.
- \* The syndication service supports polling. This means that if a syndication service is marked as *pollable* the WidSets server can poll the external feed on behalf of the client and notify the client when new content is available.
- \* A client will use the syndication service actions `getItems` and `getPage` to retrieve the actual content.
- \* The feed content is returned to the client as a structured and tagged list format. The feed item description part is represented as a list of text and image url elements. The image url elements represent the images embedded in the feed item description text or in separate enclosures. The order of the text and image url elements represent their order in the original feed.



# Supported widget types contd.

- \* **Image Service widget**

- \* The old picviewer service has been removed and its functionality has been separated into *Image service and Syndication service*. Now the sole purpose of Image service is to fetch images on demand. Syndication service takes care of delivering feed content such as Flickr image feeds.
- \* The image service delivers all images to the client scripts (widgets) in a format that is supported in the mobile terminal.
- \* In the usual case images are converted to 8 bit indexed color PNG format with the colors reduced. You can also request 24 bit PNG or JPEG images. However, note that only 8 bit PNG's are guaranteed to work on all MIDP 2.0 mobile Java implementations.

- \* **HTTP-service widget**

- \* With the HTTP service the script can execute HTTP GET and HTTP POST methods. If the MIME type of the response is human readable text ("text/..."), the response will be a *string*, otherwise it will be *binary*. The output of the http service can be processed with a filter.

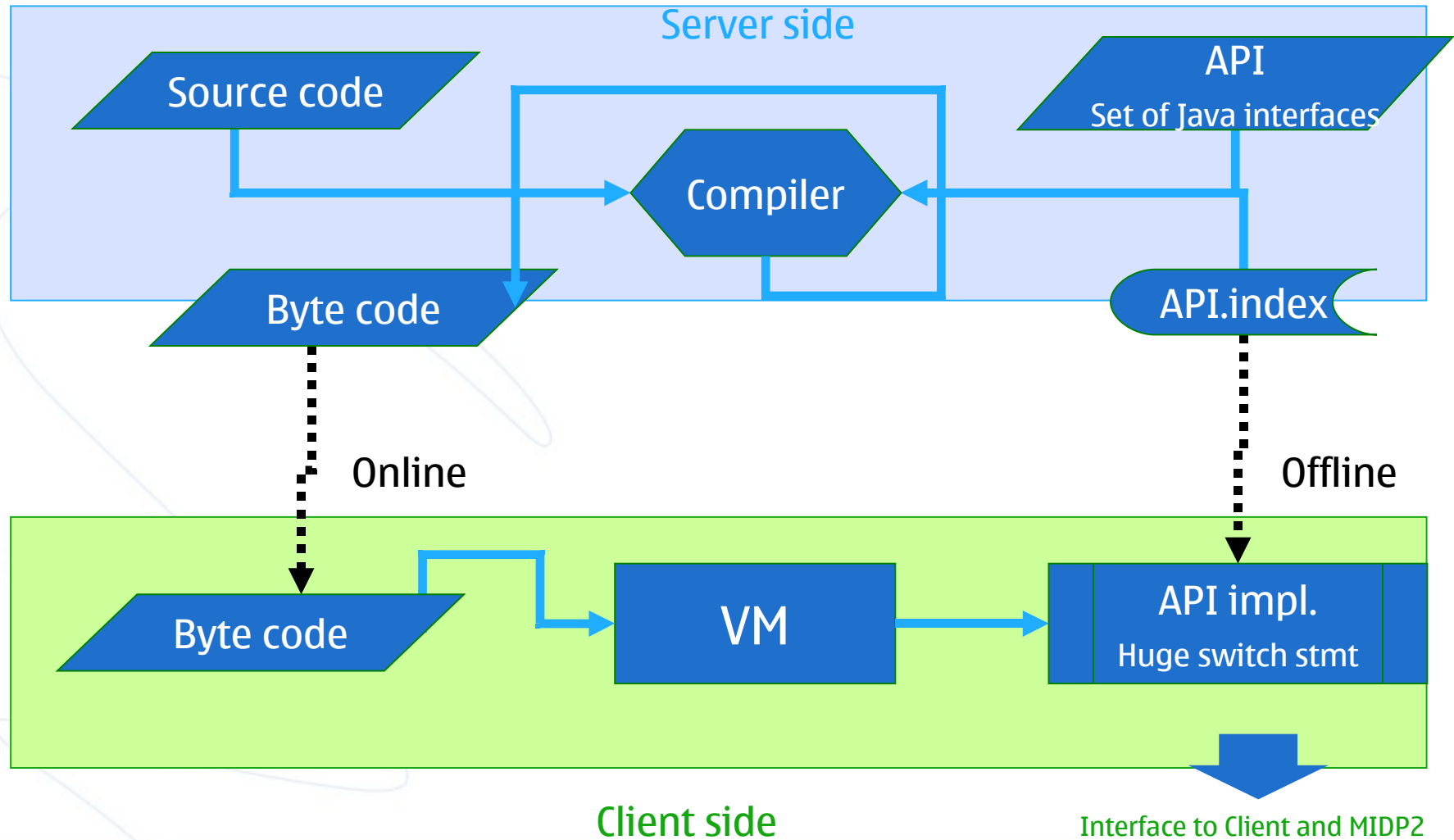
# WidSets Scripting Language summary

- \* It's strongly typed widget "scripting" language
- \* It's the glue that binds the widget building blocks together together
- \* It's needed due to the lack of **ClassLoaders** on MIDP2.0 platform. Without **ClassLoader** it is impossible to import new `.class` files (=functionality) into already installed application.
- \* It's needed in order to minimize the use of resources
- \* It's needed in order to enhance the widget programming experience
- \* It has a look and feel of Java programming language and should be immediately familiar to any Java developer

# Design principles

- \* Low memory footprint
- \* Low codesize footprint
- \* High performance...
- \* Minimum or no use of *wrapper* objects
- \* Familiar syntax
- \* Easily extendable/replaceable API
- \* Applicable for other domains as well, for instance, server side

# WidSets Architecture



# Syntax - example

```
class MyWidget
{

    const int CMD_BACK = 1;

    MenuItem m_back = new MenuItem(CMD_BACK, "Back");
    long m_started;

    void startWidget()
    {
        m_started = currentTimeMillis();
    }

    MenuItem getSoftKey(Shell s, Component c, int
        key)
    {
        if (key == SOFTKEY_OK) {
            return m_back;
        } else {
            return null;
        }
    }

    void actionPerformed(Shell s, Component c, int
        action)
    {
        if (action == CMD_BACK) {
            popShell(s);
        }
    }
}
```

```
Shell openWidget()
{
    Label hello = new Label(getStyle("helloText"),
        "Hello, world!");
    hello.setFlags(VISIBLE|LINEFEED);
    hello.setPreferredWidth(-100);

    int seconds = (currentTimeMillis() - m_started) /
        1000;
    Text time = new Text(getStyle("timeText"),
        "I've been running for "+seconds+" seconds.");
    time.setFlags(VISIBLE|LINEFEED);
    time.setPreferredWidth(-100);

    Flow container = new Flow(getStyle("hello"));
    container.setPreferredSize(-100, -100);
    container.add(hello).add(time);

    return new Shell(container);
}
}
```

# Language Syntax

## \* Comments

```
/* reel hack */  
// horrible kludge
```

## \* Literals

```
"quick brown fox jumps over lazy dog"  
'\n' /* character literal, produces int */  
123456  
0777777 /* octal */  
0xcafebabe
```

## \* Statements

```
if else while do for switch  
case default return break continue  
foreach [Java5]
```

## \* Operators

```
+ - * / % & | ^ << >>  
+= -= *= /= %= &= |= ^= <<= >>=  
< <= > >= == !=  
! && ||  
++ -- ~  
instanceof new
```

# Syntax contd.

- \* No access qualifiers

`public protected private`  
`final static synchronized`  
`abstract transient volatile`  
`native strictfp`

- \* No user defined classes

- \* The whole script is sort of a class
- \* "Member" variables live for the duration of client run
  - \* Initializers are executed just before invoking first widget callback

- \* No `import`

- \* All API declarations are directly accessible

# Syntax - contd.

- \* No exception handling
  - \* When needed error handling can be done with return codes
  - \* Any exception will terminate run and generate error
    - \* It is, after all, *an exceptional situation*
- \* No floating points
  - \* Not available in MIDP2
- \* No language level support for Java-style array[] syntax
  - \* Operator overloading allows close enough emulation

```
ByteArray init_rc4(ByteArray key)
{
    ByteArray perm = new ByteArray(256);
    for(int i=0; i<256; i++) {
        perm[i] = i;
    }
    int j = 0;
    for(int i=0; i<256; i++) {
        j = (j + perm[i] + key[i % key.length()]) & 0xff;
        perm[i], perm[j] = perm[j], perm[i];
    }
    return perm;
}
```



# Syntax - what's different

- \* Flow control statements `break` and `continue` does not support labels
  - \* Needed quite seldom - necessary?

- \* Limited set of primitive types

```
boolean int long
```

- \* All variables must be introduced with an assignment
  - \* Like Java, variables exists on the scope they are declared

```
void test()  
{  
    int j; /* error, missing assigment */  
    {  
        int c = 0;  
    }  
    {  
        int d = c; /* error, 'c' is not visible here */  
    }  
}
```

# Syntax - what's new

- \* Shell style comments

```
# crazy think
```

- \* Binary literals

```
int i = 0b101010101;
```

- \* XOR operator: ^^

- \* Oh well, it's actually equal to !=

- \* But looks l33t

- \* Constants - with primitive and String types

```
const int MAX = 100;
```

```
const String PASSWORD = "n00Bh2xöR";
```

# Syntax - keywords

```
boolean break  
case char class const continue  
default do  
else  
false final for foreach  
if instanceof int  
long  
new null  
return  
struct switch  
true  
void  
while
```

# Supported types

- \* Primitives

  - `boolean` /\* true/false \*/

  - `char` /\* character literal, stored as 32 bit int \*/

  - `int` /\* signed 32 bit \*/

  - `long` /\* signed 64 bit \*/

- \* Boolean is internally handled as int

- \* Any classes declared on API

  - \* Handling of non-primitive types is similar to Java

  - \* `==` and `!=` operators compare the *identity* of objects

  - \* Method `boolean Object.equals(Object)` compares the actual object contents

# Casting

- \* Implicit casting occurs if possible
- \* Any primitive converts to any other primitive
  - \* Even if it would be narrowing (e.g. `long` → `int`)

- \* Explicit casting is different

```
Object f = ...;
```

```
Gizmo g = Gizmo(f); /* whoa, looks like an function call */
```

```
boolean b = ...;
```

```
int zeroOrOne = int(b);
```

- \* Benefit of this syntax is reduced noise and chaining:

```
Object f = ...;
```

```
int length = String(f).length();
```

```
/* ugly java style: ((String)f).length(); */
```

# Casting

- \* On assignments super type is promoted to sub type automatically

```
Object a = ...;
```

```
String b = a; /* yep, that's right ! */
```

- \* Contrary to Java the casting is purely compile time feature

- \* It is not reflected on byte code for performance reasons!

- \* All casts always succeed but the problems may raise later:

```
List a = new List();
```

```
Object b = a;
```

```
String c = String(b);
```

```
int i = c.length(); /* Croak! ClassCastException */
```

- \* NullPointerException has similar semantics as well

# Casting

- \* On the previous example, this is what happens in code...

```
/* byte code */
aload $1 /* b */
astore $2 /* c */
pop
aload $2 /* c */
invoke_native1 #40 /* int String.length() */

/* jump to Java */
public int invoke(int sp, int[] istack, Object[] astack, long[] lstack, int
    method) {
    switch(method) {
    case 40:
        /* int String.length() */
        istack[sp] = ((String)astack[sp]).length(); /* Croak! ClassCastException
        */
        return 0;
    }
}
```

# Method resolution

- \* When selecting target method for invocation all arguments are checked
- \* There are four match categories:
  - \* **Exact match**
    - \* *source* and *target* are equal
  - \* **Close match**
    - \* *source* and *target* are primitives
    - \* *target* is super type of *source*
  - \* **Loose match**
    - \* *source* is null and *target* is class
    - \* *source* is any and *target* is a String
    - \* *source* is string and *target* is primitive
    - \* there exists cast operator from *source* to *target*
  - \* **No match**
- \* Result of argument list is equal to the "worst" match
- \* If there are multiple matches on same category ambiguity is presented as to user
- \* Luckily, the resolution is quite natural and one should not worry too much about it
  - \* Only some rare situations require explicit casts



# Tuples

- \* Functions may return multiple values - *tuples*
- \* Returned tuples may be embedded to function call arguments and return values.
- \* Opportunity to use stack instead of heap for temporary containers
- \* Tuples can be used on both sides of assignment, for instance, to swap values

```
String, int getNameAndAge() {
    return "John", 27;
}

int, int sort(int a, int b) {
    if (a<b) {
        return a, b;
    } else {
        return b, a;
    }
}

void test() {
    int age, String name = getNameAndAge();
    printf("name=%s, name=%i", getNameAndAge());
    int x, int y = sort(random(100), random(100));
    printf("x=%i, y=%i", x, y);
    x, y = y, x;
}
```

# Structs

- \* Behave like the Java-classes with public fields and no methods.
- \* `instanceof` does not work due to lack of RTTI

```
struct Celestial
{
    String name;
    int diameter;
    List satellites;
}

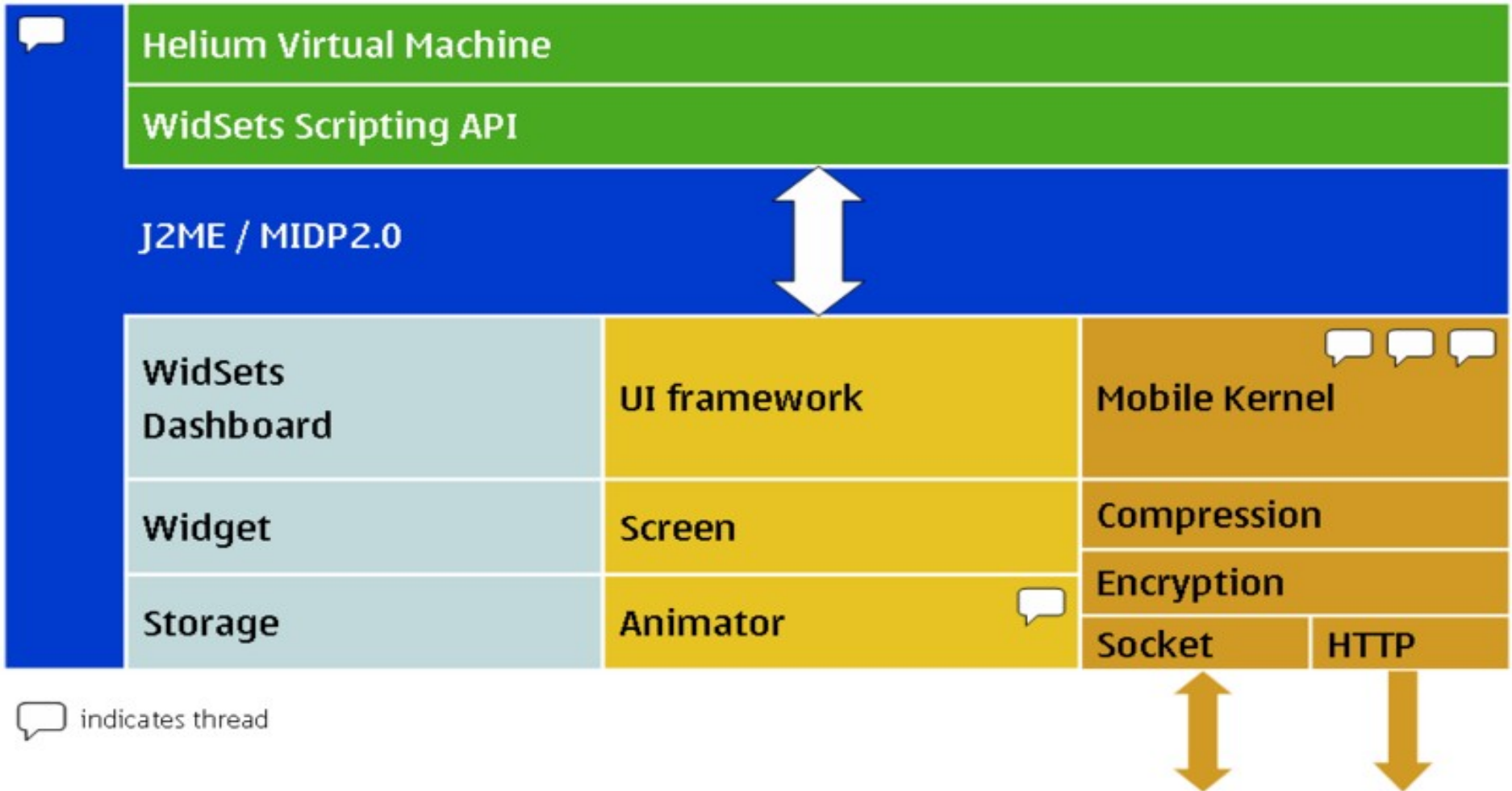
Celestial newBody(String name, int diameter)
{
    return newBody(name, diameter, new List());
}

Celestial newBody(String name, int diameter, List satellites)
{
    Celestial c = new Celestial()
    c.name = name;
    c.diameter = diameter;
    c.satellites = satellites;
    return c;
}

void test()
{
    Celestial solarSystem = newBody("Sun", 696000,
        new List().add(newBody("Mercury", 2439))
            .add(newBody("Venus", 6051))
            .add(newBody("Earth", 6371)));
}
```

# WidSets Client

- Compiler:
- Multipass optimizing compiler
  - All entity declaration are resolved



# Operator overloading

- \* API supports operator overloading
- \* Aim is to make the API natural and useable

```
new Type (...)  
expr[expr]  
expr[expr] = expr  
expr.symbol  
expr.symbol = expr  
Type(expr)           /* cast */  
foreach               /* statement */  
[]                    /* "array" creation */  
expr=>expr            /* key-to-value binding */  
[expr, expr, ...]    /* append to array */
```

# Operator overloading: new

## WidSets Scripting Language

```
Text text = new Text(getStyle("helloStyle"), "Hello  
world");
```

## API:

```
public interface widsets.api.Text {  
    public Object operator_new(Style style, String text);  
}
```

## Implementation:

```
case 0X...:  
    astack[sp-1] = new  
cheetah.ui.Text((cheetah.ui.Style)astack[sp-1],  
                (String)astack[sp]);  
    return 1; /* sp adjustment: sp -= 1 */
```

# Operator overloading: get

## WidSets Scripting Language

```
Map map = ...;
String value1 = map.name;
String value2 = map["name"];

List list = ...;
String value3 = list[3];
```

## Java:

```
public interface widsets.api.Map {
    public String operator_get(String key);
}

public interface widsets.api.List {
    public Object operator_get(int index);
}
```

## Implementation:

```
case 0X...:
    astack[sp-1] = ((Hashtable)astack[sp-1]).get(astack[sp]);
    return 1;
case 0X...:
    astack[sp-1] = ((Vector)astack[sp-1]).elementAt(istack[sp]);
    return 1;
```

# Operator overloading: set

## WidSets Scripting Language

```
Map map = ...;
map.name = "hello";
map["name"] = "hello";
```

```
List list = ...;
list[3] = "fourth";
```

## Java:

```
public interface widsets.api.Map {
    public Object operator_set(String key, Object value);
}
public interface widsets.api.List {
    public Object operator_get(int index, Object value);
}
```

## Implementation:

```
case 0X...:
    ((Hashtable)astack[sp-2]).put(astack[sp-1], astack[sp]);
    astack[sp-2] = astack[sp]; /*return value*/
    return 2;
case 0X...:
    ((Vector)astack[sp-2]).setElementAt(istack[sp-1], astack[sp]);
    astack[sp-2] = astack[sp]; /*return value*/
    return 2;
```

# Operator overloading: cast

## WidSets Scripting Language

```
List list = List("abc");  
/* list[0].equals("a") */  
/* list[1].equals("b") */  
/* list[2].equals("c") */
```

## Java:

```
public interface widsets.api.API {  
    public List operator_cast_List(String source);  
}
```

## Implementation:

```
case 0X...:  
    final String s = (String)astack[sp];  
    final int    n = s.length();  
    final Vector v = new Vector(n);  
    for(int i=0; i<n; i++) {  
        v.addElement(s.substring(i,i+1));  
    }  
    astack[sp] = v;  
    return 0;
```

Cast overloading works also nicely from/to primitive types.



# Operator overloading: foreach

## WidSets

```
List list = ...;
int i = 0;
foreach(Object value: list)
    printf("%i: %s", i++, value);
```

## Java:

```
public interface widsets.api.API {
    public void operator_start(List list, int tmpVar);
    public boolean__Object operator_next(List list /*dummy*/, int tmpVar);
    class boolean__Object {} /* tuple! */
}
```

## Implementation:

```
case 0X..: /* start */
    final int tmp = istack[sp];
    istack[tmp] = 0;
    astack[tmp] = stack[sp-1];
    return 2;
case 0X..: /* next */
    final int tmp = istack[sp];
    final Vector list = (Vector)astack[tmp];
    final int index = istack[tmp];
    if (index < list.size()) {
        zset(sp-1, true);
        astack[sp] = list.elementAt(index);
        istack[tmp] = index+1;
        return 0;
    } else {
        zset(sp-1, false);
        astack[sp] = null;
        return 0;
    }
}
```

# Operator overloading: array & binding

## Helium:

```
Value value = [];  
Value bind = "key"=>"value";
```

## Java:

```
public interface widsets.api.API {  
    public Value operator_array();  
    public Value operator_bind(Value key, Value value);  
}
```

## Implementation:

```
import hecate.kernel.Value;  
  
case 0X...: /* array */  
    astack[sp] = Value.List();  
    return -1;  
  
case 0X...: /* binding */  
    astack[sp-1] = Value.Binding((Value)astack[sp-1],  
                                (Value)astack[sp]);  
    return 1;
```

# Operator overloading: array append

## Helium:

```
Value value = ["one", "two", "three"];
```

## Java:

```
public interface widsets.api.Value {  
    public void operator_append(Value element);  
}
```

## Implementation:

```
import hecate.kernel.Value;  
  
case 0X...:  
    ((Value) astack[sp-1]).add((Value) astack[sp]);  
    return 2;
```

# Compiler passes

1. Tokenize source code
2. Parse syntax and create node tree
3. Resolve symbols and check other grammatic aspects
4. Resolve calls and perform type checking
5. Fold constant expressions
6. Analyze jumps and eliminate unreachable code
7. Produce bytecode
8. Verify and assert bytecode stack integrity

# WidSets VM Implementation

- \* Compiled scripts contains:

```
short    header
ubyte    version
ubyte    staticIntCount
ubyte    staticLongCount
ubyte    staticObjectCount
Value[] constantPool /* large int, large long and String constants
    */
Entry[] exportedFunctionEntryPoints /* Entry { int ip, String name,
    int argc } */
byte[]   code           /* up to 64K */
short[] lineNumberTable /* ip=>line, optional */
```

- \* Symbolic references are kept to minimum
  - \* Only widget "interface" entry points are exposed
  - \* API library access is based on integer indices
- \* Instruction set is "reduced complex": 133 instructions
- \* Scripts are considerably shorter than equivalent .class files

For example: RssWidget ~3.5K vs ~9.5K

# Widsets VM implementation

- \* WVM is stack based - easy to implement
- \* It uses *triple array* idiom for stack, static vars and structs

```
int[]    istack
```

```
long[]   lstack
```

```
Object[] astack
```

- \* On stack only one of the three slots is used between [0..stackPointer]
  - \* With two exceptions (stackframe and foreach)
- \* No wrapper objects needed
- \* Primitives handled naturally
- \* Relies on JVM garbage collection

- \* Example

```
case Asm.iadd:
```

```
    sp--;
```

```
    istack[sp] += istack[sp+1];
```

```
    continue;
```

# WidSets VM Instruction set

**stack:** nop pop popn iswap lswap aswap idup ldup adup imove lmove amove  
**constants:** iconst\_m1 iconst\_0 iconst\_1 iconst\_2 iconst lconst\_0 lconst\_1 lconst  
**constants:** aconst\_null astring\_empty astring bipush sipush  
**conversions:** z2i z2l i2z i2l l2z l2i z2s i2s l2s a2s s2z s2i s2l  
**arithmetic:** iadd ladd isub lsub imul lmul idiv ldiv irem lrem ineg lneg concat  
**binary:** ior lor iand land ixor lxor ishl lshl ishr lshr iinv linv  
**incdec:** iinclocal linclocal iincstatic lincstatic iincfield lincfield  
**update:** iupdatelocal lupdatelocal iupdatestatic lupdatestatic iupdatefield lupdatefield  
**locals:** iload lload aload istore lstore astore  
**statics:** igetstatic lgetstatic agetstatic iputstatic lputstatic aputstatic  
**fields:** igetfield lgetfield agetfield iputfield lputfield aputfield  
**comparison:** ieq leq aeq ine lne ane ilt llt ile lle igt lgt ige lge  
**comparison:** isnull isnotnull not xor  
**jump:** goto1 if\_eq1 if\_ne1 if\_true1 if\_false1  
**jump:** goto2 if\_eq2 if\_ne2 if\_true2 if\_false2  
**instances:** newstruct instance\_of  
**calling:** call call\_native1 call\_native2  
**stackframe:** enter leave ileave lleave aleave nleave loadsp

# Widget interface

All callbacks are optional

```
/* Widget lifecycle */
void startWidget();
void stopWidget();
void updateWidget();

/* Menu control */
Menu getMenu(Shell shell, Component focused);
MenuItem getSoftKey(Shell shell, Component focused, int key);

/* User actions */
Shell openWidget();
void closeWidget(); /* not implemented yet */
void actionPerformed(Shell shell, Component source, int action);
boolean keyAction(Component source, int op, int code);

/* Construction of view elements */
Component createElement(String viewName, String elementName,
                        Style style, Object context);

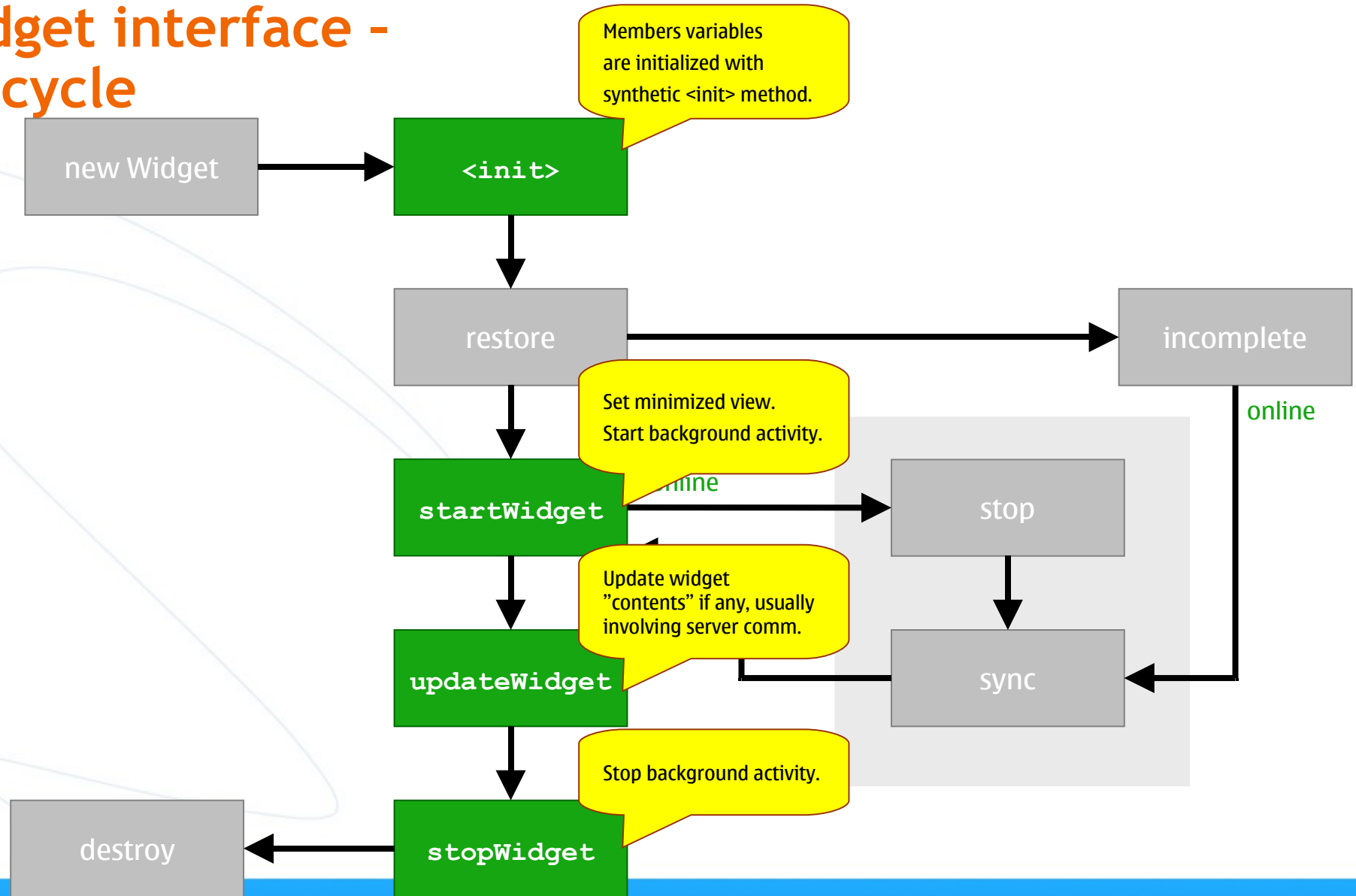
/* Timer events */
void timerEvent(Timer timer);

/* Canvas component */
void paint(Component c, Graphics g, Style style, int width, int height);

/* Server communication */
void onSuccess(Object state, Value returnValue);
void onFailure(Object state, String errorMessage);
```



# Widget interface - lifecycle



# Widget interface - user actions



# Widget interface - menu control



```
const int CMD_BACK = 1; /* must be >0 */
MenuItem m_opts = new MenuItem(OPEN_MENU, "Options");
MenuItem m_back = new MenuItem(CMD_BACK, "Back");

MenuItem getSoftKey(Shell s, Component c, int key) {
    if (key == SOFTKEY_BACK) {
        return m_back;
    } else if (key == SOFTKEY_OK) {
        return m_opts;
    } else {
        return null;
    }
}
```



```
const int CMD_BOOKMARK = 2; /* must be >0 */
const int CMD_MARKREAD = 3; /* must be >0 */

Menu m_menu = new Menu()
    .add(CMD_BOOKMARK, "Bookmark")
    .add(CMD_MARKREAD, "Mark all read");

Menu getMenu(Shell s, Component c) {
    return m_menu.reset()
        .enable(CMD_BOOKMARK, c != null);
}
```

# Widget interface - user actions



```
/* return true if event was consumed */
boolean keyAction(Component src, int op, int key)
{
    if (op == KEY_PRESSED) {
        if (key == KEY_RIGHT) {
            openItem(src);
            return true;
        }
    }
    return false;
}

void openItem(Component c) { ... }
```

```
void actionPerformed(Shell s, Component c, int cmd) {
    switch(cmd) {
        case CMD_BACK: popShell(s); break;
        case CMD_BOOKMARK: bookmark(c); break;
        case CMD_MARKREAD: markAllRead(); break;
    }
}

void bookmark(Component c) {...}
void markAllRead() {...}
```



# Widget interface - views



```
<view id="viewList" mode="default" style="bg">
  <box top="0%" right="100%" bottom="36sp+2px" left="0%">
    <action type="draw" id="sideicon" style="feedIcon" />
  </box>
  <box top="36sp+2px" right="100%" bottom="100%" left="0%">
    <action type="script" id="feedItemList" style="feedItemList" />
  </box>
</view>
```

```
Flow m_flow = null;

Component createElement(String viewName, String elementName,
                        Style style, Object context)
{
  if (elementName.equals("feedItemList")) {
    m_flow = new Flow(getStyle("default"));
    m_flow.setPreferredSize(-100, 0);
    return new Scrollable(style, fillItems());
  } else ...
}

Flow fillItems() { ... }
```

# Widget interface - events

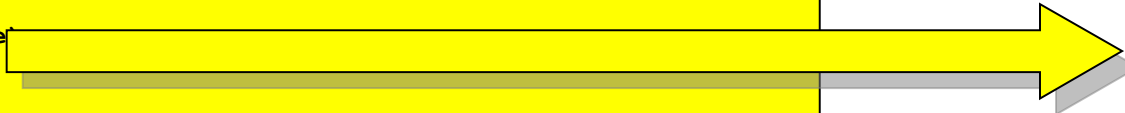
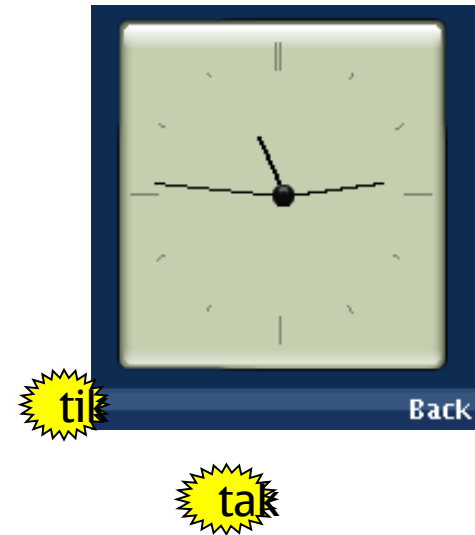
```
Calendar m_time = new Calendar();
Timer m_timer;
Canvas m_analog;

void startWidget() {
    m_timer = schedule(1000, 1000); /* delay, period in ms */
}

void stopWidget()
{
    m_timer.cancel();
    m_timer = null;
}

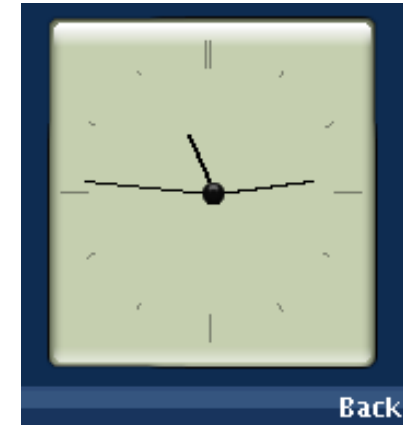
void openWidget() {
    m_analog = new Canvas();
    ...
}

void timerEvent(Timer timer) {
    m_time.setMillis(currentTimeMillis());
    if (m_analog != null) {
        m_analog.repaint(false);
        flushScreen(false);
    }
}
```

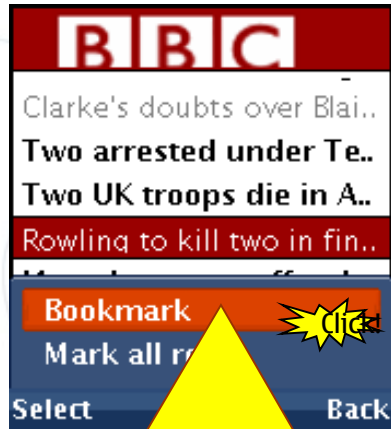


# Widget interface - free paint

```
const int SECOND_HAND = 60;
const int MINUTE_HAND = 45;
const int HOUR_HAND = 30;
void paint(Component c, Graphics g, Style style, int width, int height)
{
    g.setColor(0);
    int h, int m, int s = m_time[ HOUR ], m_time[ MINUTE ], m_time[ SECOND ];
    int cx, int cy = width/2, height/2;
    {
        int angle = 6*s - 90; /* in degrees */
        int dx = cx+(SECOND_HAND*cos(angle))/MATH_SCALE; /* integer trigonometry */
        int dy = cy+(SECOND_HAND*sin(angle))/MATH_SCALE;
        g.drawLine(cx, cy, dx, dy);
    }
    {
        int angle = (6*m)+(s/10) - 90;
        int dx = cx+(MINUTE_HAND*cos(angle))/MATH_SCALE;
        int dy = cy+(MINUTE_HAND*sin(angle))/MATH_SCALE;
        g.drawLine(cx, cy, dx, dy);
        g.drawLine(cx+1, cy, dx+1, dy);
    }
    {
        int angle = (30*h)+(m/2) - 90;
        int dx = cx+(HOUR_HAND*cos(angle))/MATH_SCALE;
        int dy = cy+(HOUR_HAND*sin(angle))/MATH_SCALE;
        g.drawLine(cx, cy, dx, dy);
        g.drawLine(cx+1, cy, dx+1, dy);
    }
    g.drawImage(style.image(0), cx, cy, HCENTER|VCENTER);
}
```



# Widget interface - server communication



```
Object STATE_BOOKMARK = new Object();

void actionPerformed(Shell s, Component c, int cmd) {
    if (cmd == CMD_BOOKMARK) {
        Value data = Value(c.getData()); /* data attachment */
        if (data != null) {
            call(STATE_BOOKMARK, "feed1",
                ["bookmark", ["default", data[3]]]);
        }
    } else ...
}
```

To server and back...

```
void onSuccess(Object state, Value arg)
{
    if (state == STATE_BOOKMARK) {
        setBubble(null, "Bookmark added");
    } else ...
}
```



# WidSets widget structure

# WidSets widget structure

Technology	Purpose	File extension	Example
XML (eXtensible Markup Language)	Structure/Design	.xml	widget.xml
WidSets Scripting Language (Helium)	Logic	.he	helloworld.he
PNG (Portable Network Graphics)	Icon/image	.png	web_icon.png

- \* A typical WidSet's widget consists of the following files
  - \* <Widget\_name>.he
  - \* Widget.xml
  - \* Web\_icon.png
  - \* Web\_minimized.png
  - \* Web\_maximized.png

# WidSets widget structure contd.

C:\devkit\examples\HelloWorld\  
widget.xml

helloworld.he

web\_icon.png

web\_minimized.png

web\_maximized.png

This xml file contains widget's specific details, such as the meta data of the widget, services used by the widget, the stylesheet of the widget, resources and layout of the widgets' views.

```
1 <?xml version="1.0" encoding="utf-8"?>
2
3 <widget spec_version="2.0">
4   <info>
5     <name>hello_world</name>
6     <version>0.1</version>
7     <author>render</author>
8     <clientversion>0.98</clientversion>
9     <shortdescription>Very simple widget</shortdescription>
10    <longdescription>Simplest possible widget saying hello
11    <tags>test example hello world</tags>
12  </info>
13
14  <parameters>
15    <parameter type="string" name="widgetname" description:
16  </parameters>
17
18  <resources>
19    <code src="hello_world.he"/>
20
21  <stylesheet>
```

# WidSets widget structure contd.

C:\devkit\examples\Helloworld\

widget.xml

helloworld.he

web\_icon.png

web\_minimized.png

web\_maximized.png

This file contains the WidSets Scripting Language (aka Helium) scripts for extending widget's functionalities.

```
1 class
2 {
3   const String EU_TEXT = "European Union";
4   const String FINNISH_TEXT = "Finland";
5   const String UN_TEXT = "United Nations";
6
7   const int CMD_BACK = 1;
8   const int CMD_MANUAL = 10;
9   const int CMD_TIMER = 11;
10  const int CMD_ABOUT = 12;
11
12  MenuItem OPTIONS = new MenuItem(OPEN_MENU, "Options");
13  MenuItem BACK = new MenuItem(CMD_BACK, "Back");
14
15  Menu MENU = new Menu()
16    .add(CMD_MANUAL, "Manual")
17    .add(CMD_TIMER, "Timer")
18    .add(CMD_ABOUT, "About");
19
20  Menu MENU_ABOUT = new Menu()
21    .add(CMD_BACK, "Back");
--
```

# WidSets widget structure contd.

C:\devkit\examples\Helloworld\

widget.xml

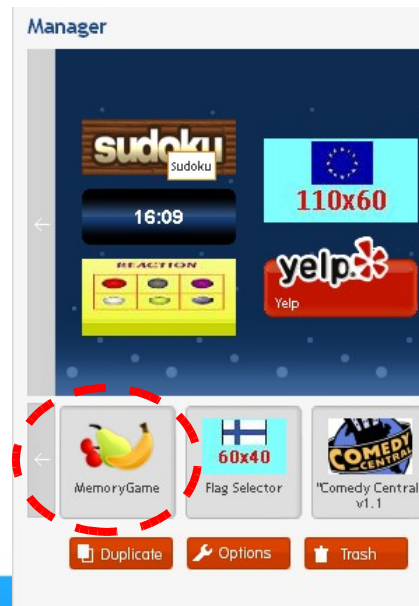
helloworld.he

web\_icon.png

web\_minimized.png

web\_maximized.png

This image is the icon used in the widget library or in the widget shelf of the WidSets manager. The pixel size of this image has to be exactly **60\*40px**. This image is also visible in a mobile device under the system widget's library.



# WidSets widget structure contd.

C:\devkit\examples\Helloworld\

widget.xml

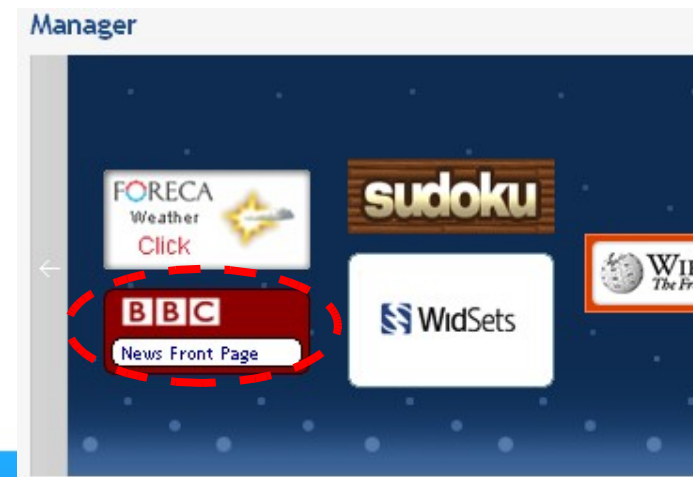
helloworld.he

web\_icon.png

web\_minimized.png

web\_maximized.png

This image is used for displaying the widget in the WidSets manager. It presents the minimized view of the widget on the Website. The width of the image must be exactly **110 pixels** and the height is flexible but should not exceed the maximum height of the widget.



# WidSets widget structure contd.

C:\devkit\examples\Helloworld\

widget.xml

helloworld.he

web\_icon.png

web\_minimized.png

web\_maximized.png

This image is the preview image for the widget in the widget library on the WidSets Web Site. It is used as an image to illustrate how the widget will look like in the open view. The pixel size of this image has to be exactly **176\*208px**.

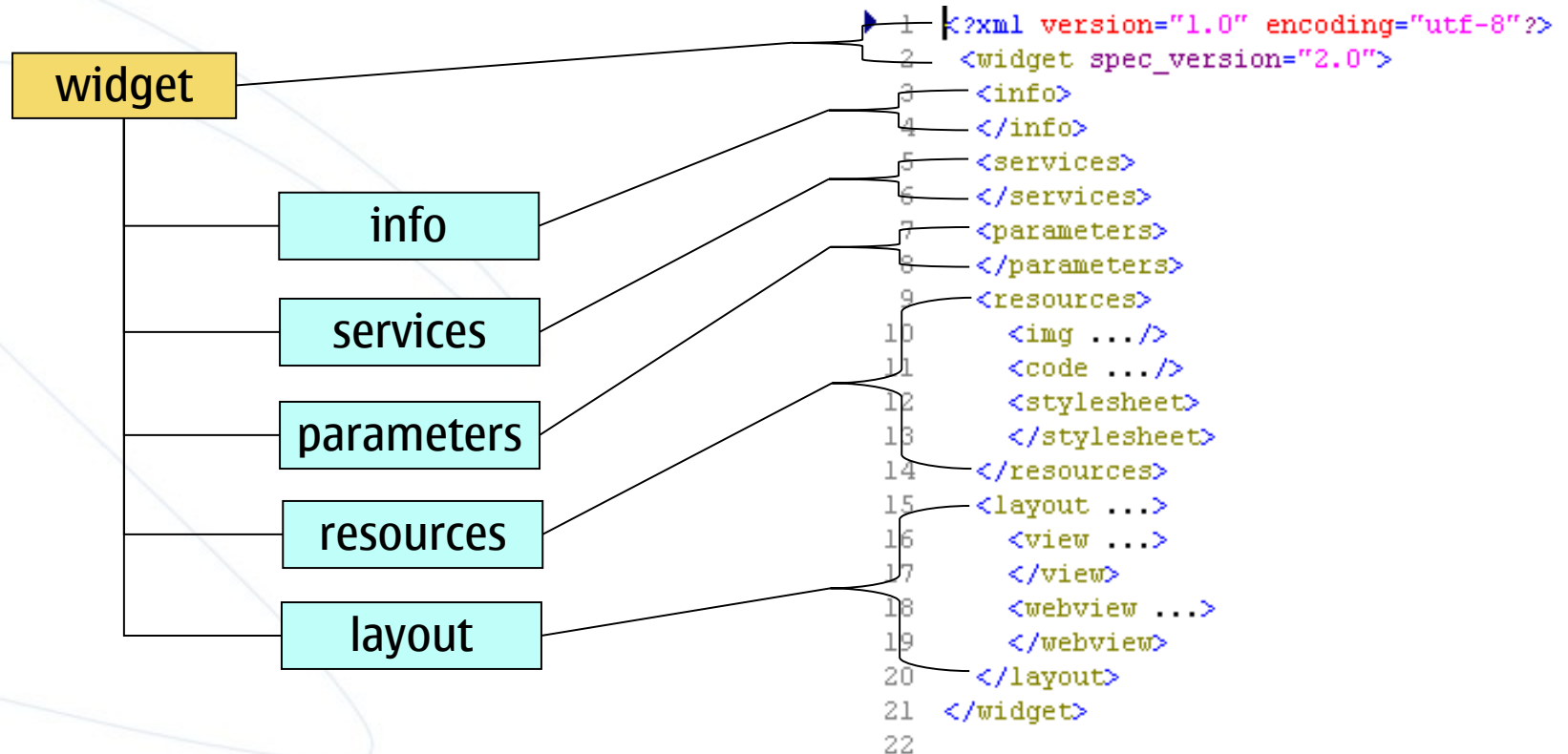


# WidSets Specification



# Widget Specification Version 2.1

## \* Widget.xml element structure



# WidSets Scripting Language

# Scripting Codes

```
<widget>
```

```
  <sources>
```

```
    <code src=Helloworld.he"/>
```

```
  </sources>
```

```
</widget>
```

# WidSets Scripting Language Syntax

## \* Comments

```
/* block/line comments */  
// line comments
```

## \* Literals

```
; // line ends  
"quick brown fox jumps over lazy dog" /* string */  
'\n' /* character, produces int */  
123456 /* integer */  
0777777 /* octal */  
0xcafebabe /* hexa */  
0b101010101 /* binary */
```

## \* Statements

if else while do for switch case default return break continue foreach

## \* Operators

+ - \* / % & | ^ < > ! ~ << >> += -= \*= /= %= &=  
|= ^= <<= >>= <= >= == != && || ++ -- instanceof new ? :

## \* Primitives

```
boolean /* true/false */  
int /* signed 32 bit */  
long /* signed 64 bit */
```

## \* Keywords

class const false null return struct true void

# Widget Class

- \* WidSets scripting code is enclosed within the widget's class
- \* A widget can have only one class

```
class <MyClass>
{
    // some global variables
    Some own functions ()
    {
        // some local variables
    }
    Some system functions ()
    {
        ...
    }
}
```

# Interfaces

```
/* Widget lifecycle */
void startWidget();
void stopWidget();
void updateWidget();

/* Menu control */
Menu getMenu(Shell shell, Component focused);
MenuItem getSoftKey(Shell shell, Component focused, int key);

/* User actions */
Shell openWidget();
void closeWidget();
void actionPerformed(Shell shell, Component source, int action);
boolean keyAction(Component source, int op, int code);

/* Construction of view elements */
Component createElement(String viewName, String elementName,
                        Style style, Object context);

/* Timer events */
void timerEvent(Timer timer);

/* Canvas component */
void paint(Component c, Graphics g, Style style, int width, int height);

/* Server communication */
void onSuccess(Object state, Value returnValue);
void onFailure(Object state, String errorMessage);
```

# Widget's Lifecycle

- \* `void startWidget();`
- \* `void stopWidget();`
- \* When a user starts to run WidSets, WidSets client will call the `startWidget()` function of all installed widgets on a mobile device. In most cases, you should implement this function to create the widget's minimized view to be displayed on the WidSets dashboard.
- \* The `stopWidget()` function is called when a widget is reloaded or removed from the dashboard or when WidSets client is terminated. You should implement this function to release system resources (if any) before it is terminated.

# Widget's Lifecycle

- \* Shell `openWidget()`;
- \* `void closeWidget()`;
- \* When a user selects a widget from dashboard, the system will call the `openWidget()` function. Implement this function e.g. to switch your widget to its maximized mode (most widgets do).
- \* `closeWidget()` in the other hand is called when a widget exits from maximized mode by popping (`popShell()` or `slideOut()`) the last widget-created shell or when widget is stopped for any reason. Here you might usually want to release system resources cancel yours timers or atleast set them to tick less often.



# Widget's Menu

- \* MenuItem getSoftKey(Shell shell, Component focused, int key);
- \* Menu getMenu(Shell shell, Component focused);
- \* When a user clicks on a softkey of a mobile device (a softkey is associated to the widget's menu), WidSets will call the getSoftkey() to inform the widget which one of the two softkeys was just clicked.
- \* Implement this function to detect the softkey and tell WidSets what to do with the event.
- \* The getMenu() function is called when a softkey is designed to associate with an open menu.

# Callback From User Actions

- \* void actionPerformed(Shell shell, Component source, int action);
- \* boolean keyAction(Component source, int op, int code);
- \* The **actionPerformed()** function is called as a result of one of the following actions has occurred:
  - \* Menu action
  - \* Component "fire" action
  - \* Implicit action (such as FOCUS\_CHANGED)
- \* The **keyAction()** function is called when the user clicks on an alpha-numeric key (including the navigation key).
- \* Implement these functions to detect user's actions and process them accordingly.

*Note: Remember that actionPerformed() and keyAction() are system callback functions, so you must never call any callback functions directly.*

# Widget's View Creation

- \* Flow `createView(String name, Object context)`
- \* Component `createElement(String viewId, String elementId, Style style, Object context);`
- \* The `createView()` function is used for creating a view for the widget.
- \* A view can be a minimized view or a maximized view depending on the widget's running status.
- \* Widget's UI components can be constructed within the `createElement()` function. This is a callback function and it gets called as a result of when the `createView()` function is called.

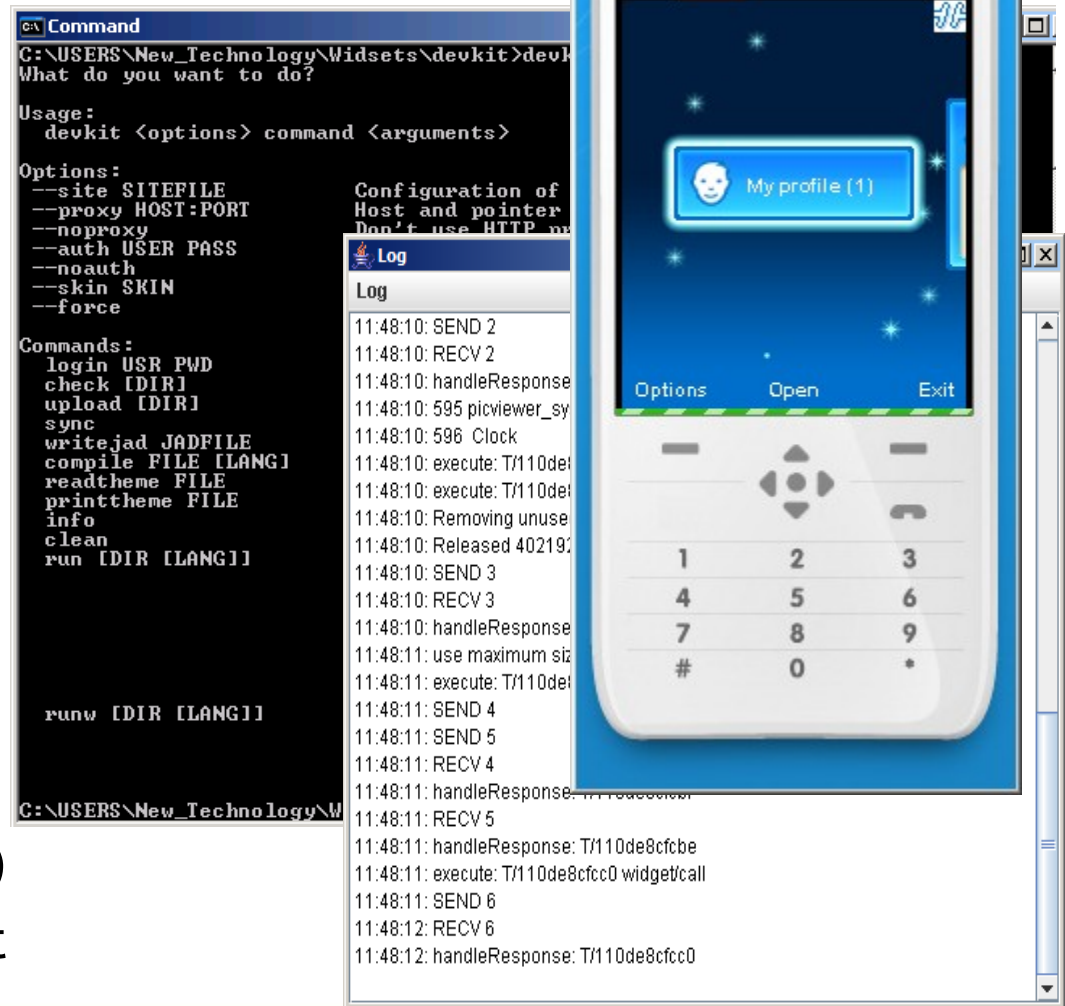
# Callback From Timer Events

- \* `void timerEvent(Timer timer);`
- \* You can set up timers that notify your widget after specified time passed by, or continuously.
- \* Timers are created using `schedule(...)` functions. These timers will then call `timerEvent(Timer timer)` function of your script, or separate `TimerCallback` function if you defined one.
- \* To release timer resource, you need to call the `cancel()` function of the `Timers` object you have created. This should be performed as soon as you don't need the timer anymore or when a widget is terminated via the `stopWidget()` function.

# WidSets Development tools

# WidSets Development Kit (SDK, Devkit)

- \* Available as download on [dev.widsets.com](http://dev.widsets.com) website
- \* Provides basic tools for widget development:
  - \* Emulator
  - \* Networking configuration
  - \* APIs references
  - \* Examples
  - \* Built-in Compiler
  - \* Widget uploading
  - \* Error messages window (log)
  - \* Synchronizing WidSets client



# Environment Configuration

1. Your current network connection goes through a proxy?

If “No”:

```
c:\widsets\devkit\devkit --noproxy [enter]
```

Note: the command above also unsets any previously set proxy.

If “Yes”, set the proxy parameter:

```
C:\widsets\devkit\devkit --proxy yourproxyaddress:port login username pwd [enter]
```

2. Setting for own server

For --site --auth --force

3. Which skin of the emulator do you want to use?

Currently there are 3 different skins: cool, basic and E61. Should you want to set the cool skin?

```
c:\widsets\devkit\devkit --skin cool [enter]
```

Note: All the configuration steps above are done only once if you don't change your environment. Thus, not every time you login or run the emulator.

4. You are now ready to login

```
c:\widsets\devkit\devkit login yourusername yourpassword [enter]
```

5. Ready to see the emulator?

```
c:\widsets\devkit\devkit run [enter]
```

# Developer Site

- \* Developers widget library
- \* Development discussion forums
- \* Get Support, give support
- \* Publish your own widgets
- \* See what others have made
- \* Test, try, vote, comment, participate =)
- \* ...contribute!

THANK YOU! Questions?