

S60 Widgets on Mobile

Wai Seto, Nokia
Widget Summit 2007

NOKIA
Connecting People

Authors

- Petro Soininen
 - Forum Nokia Technical Services and Consultancy
- Jure Sustersic
 - Forum Nokia Business Development & Channel
- Wai Seto
 - Forum Nokia Technical Services and Consultancy

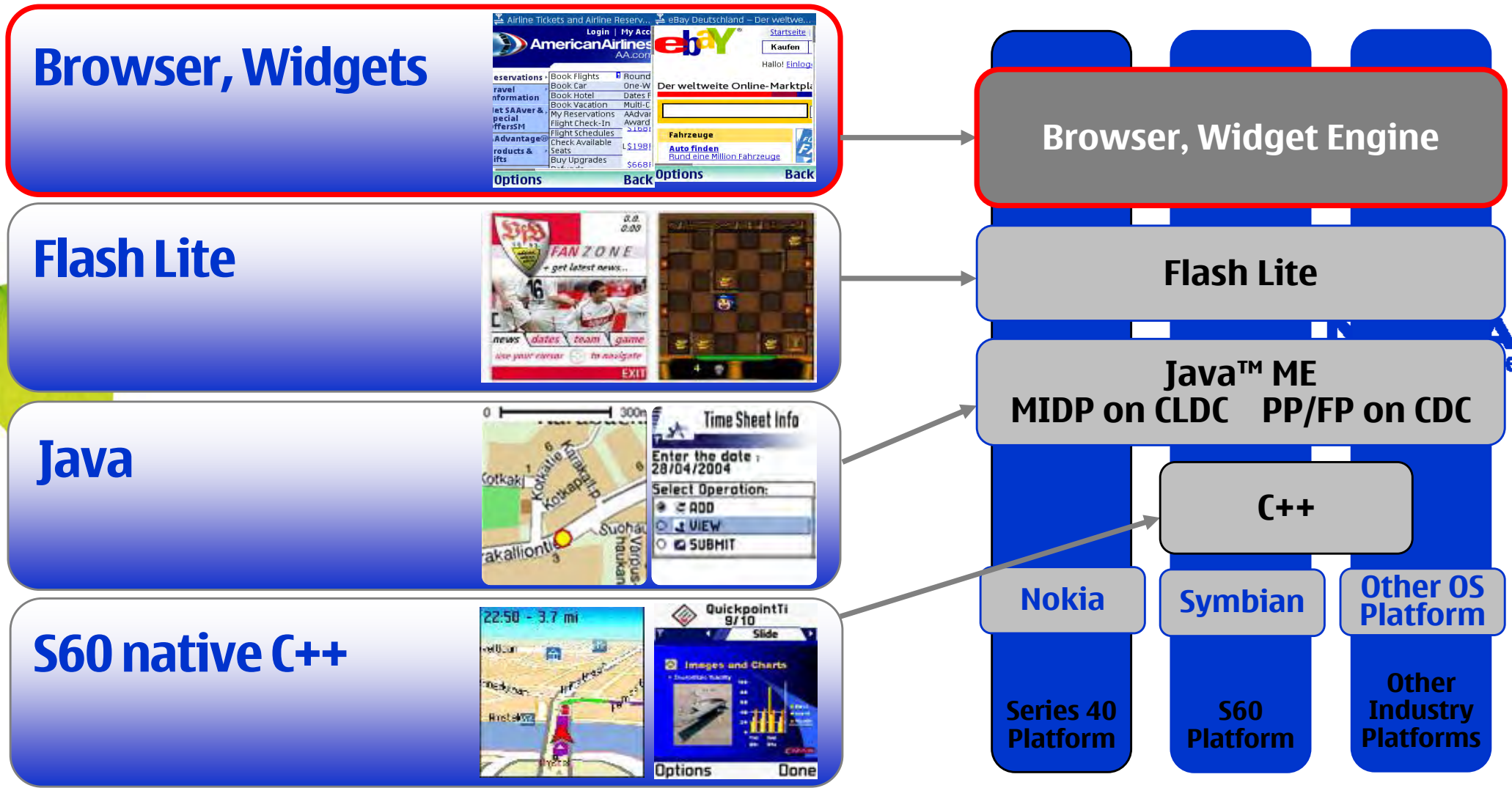


Objectives

- Share Nokia's "Widgets on Mobile" perspectives and goals
- What you need to get started on S60 Widget development
- Overview on Widget framework & APIs



Various technologies to develop on Nokia S60 platform



Web Browser for S60

Flexible for plug-ins & extensions

APIs supporting application interworking

S60 plug-ins, incl. Flash Lite, SVG, and audio.

Netscape plug-in API (NPL)

S60 Browser UI

Widget Runtime

Browser Control

Nokia UI features

S60 WebKit

WebCore KHTML

JavaScript Core KJS

OS adaptations

Memory manager

Symbian OS

Symbian HTTP framework



Legend

open sourced by Nokia

Already open source

Closed source (S60, Nokia, Symbian)

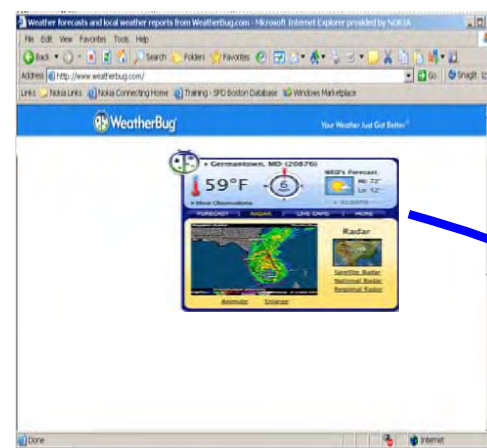
Design not limited to particular OS

Widgets

- Widget is a **standalone Web application**
 - Runs, feels & acts like a native application
 - **Local on the handset** & distributed like any other application
 - Not necessarily dependent on network access
 - Updates via server logic modifications or by redistributing the content to local device
- Widget at its core is a **Web page designed for a specific purpose**
 - Widgets use standard web technologies **HTML, CSS, JavaScript, AJAX...**
 - **Rapid development** - days - not months or years
 - Development to Deployment in “clicks”



Web page without a browser UI



NOKIA
Connecting People



Widgets on Mobile

- S60 as the 1st mobile platform announcing widget support
- Empowered by the mobile widget engine



- Compliant with standard-based Internet technologies



Widgets on mobile - Current Goals

- **Widgets leverage the Web**
 - use internet services and **Web APIs** to access information
 - use XMLHttpRequest and **AJAX** techniques
 - **low effort** to port widgets from other platforms
- **Widgets integrated into S60 user experience**
 - display the icon in the Application Menu
 - can be assigned to Idle screen soft keys and shortcuts
 - appear in the Active Applications list
 - managed the same way as existing S60 applications
 - Familiar **(Un)Installer user experience**
 - Visible in Application manager
 - Enable access to **Widget specific APIs** (widget, system info, menu)



Widgets on mobile - Future Goals

- Widgets leverage the smart phone platform
 - **combine** information from **Web with platform services**
 - improves user experience
 - **Location, PIM & media...** information
 - integrate widgets into S60 Standby Screen
 - additional UI enhancements and features
- Widgets evolve as a development platform
 - Integrated safely/securely with S60 platform services
 - access platform services via S60 APIs
 - provide **security and signing model** for widgets



NOKIA
Connecting People



Widgets on mobile – How can you help?

- **MOSH**
 - Publish and share your Widgets
 - <http://mosh.nokia.com>



- **Forum Nokia Wiki Page**
 - Contribute what you know and have found
 - <http://wiki.forum.nokia.com>
- **Discussion Board**
 - Interact with other Nokia and 3rd party developers
 - <http://discussion.forum.nokia.com>
- **Forum Nokia**
 - Official Nokia documents, SDKs
 - <http://www.forum.nokia.com>

NOKIA
Connecting People



Web Widget Framework



NOKIA
Connecting People

Widget component structure

- A widget is constructed by a bundle of files.
 - **info.plist** (mandatory)
 - icon.png
 - **[name].html** (mandatory)
 - [name].css
 - [name].js
- A widget project is a file-system directory, in which widget's component files are stored.
- Widget's mandatory files and the icon.png (if included) **MUST** be located at the root directory of a widget project.



Info.plist – Property of a Widget

- A manifest file in XML format, containing the property and configuration information of a widget.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Nokia/DTD PLIST 1.0//EN"
    "http://www.nokia.com/NOKIA_COM_1/DTDs/plist-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>DisplayNa me</key>
    <string> WidgetNa me</string>
    <key>Identifier</key>
    <string>com.company.widget.project</string>
    <key>MainHTML</key>
    <string>Main.ht ml</string>
  </dict>
</plist>
```



Supported Properties



Name	Type	Use	Description
DisplayName	String	Required	Specify the actual name of the widget listed on the Installed application bar
Identifier	String	Required	Specify unique string identifier for the widget in reverse domain format
MainHTML	String	Required	Specify the name of the main HTML page that points to the widget
AllowNetworkAccess	Boolean	Optional	Specify access to the network based resources from the widget
ShortVersionString	String	Optional	Specify release version of the widget bundle
Version	Number	Optional	Specify build version of the widget bundle

Icon.png

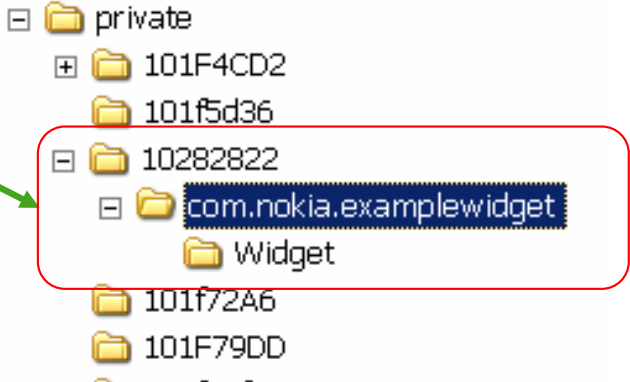
- PNG image to be used as application icon
 - Recommended size 88x88 px
 - Default S60 application icon used if icon.png is missing



...just to clarify



```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE plist PUBLIC "-//Nokia//DTD PLIST 1.0//EN" "">  
<plist version="1.0">  
<dict>  
  <key>DisplayName</key>  
  <string>Widget</string>  
  <key>Identifier</key>  
  <string>com.nokia.examplewidget</string>  
  <key>MainHTML</key>  
  <string>widget.html</string>  
</dict>  
</plist>
```



Web Widget Development



NOKIA
Connecting People

What You Need?

- For **implementing** widget code:
 - Text editor or a web IDE
- For **debugging** a running widget
 - Firefox
 - firebug
 - Greasemonkey
 - " XMLHttpRequest Bypass Security" script for free XMLHttpRequest access from local file.
- For **testing** a widget:
 - S60 3rd Edition FP2 SDK emulator
 - Beta available from [Forum Nokia web site](#)
 - Up-coming compatible S60 devices

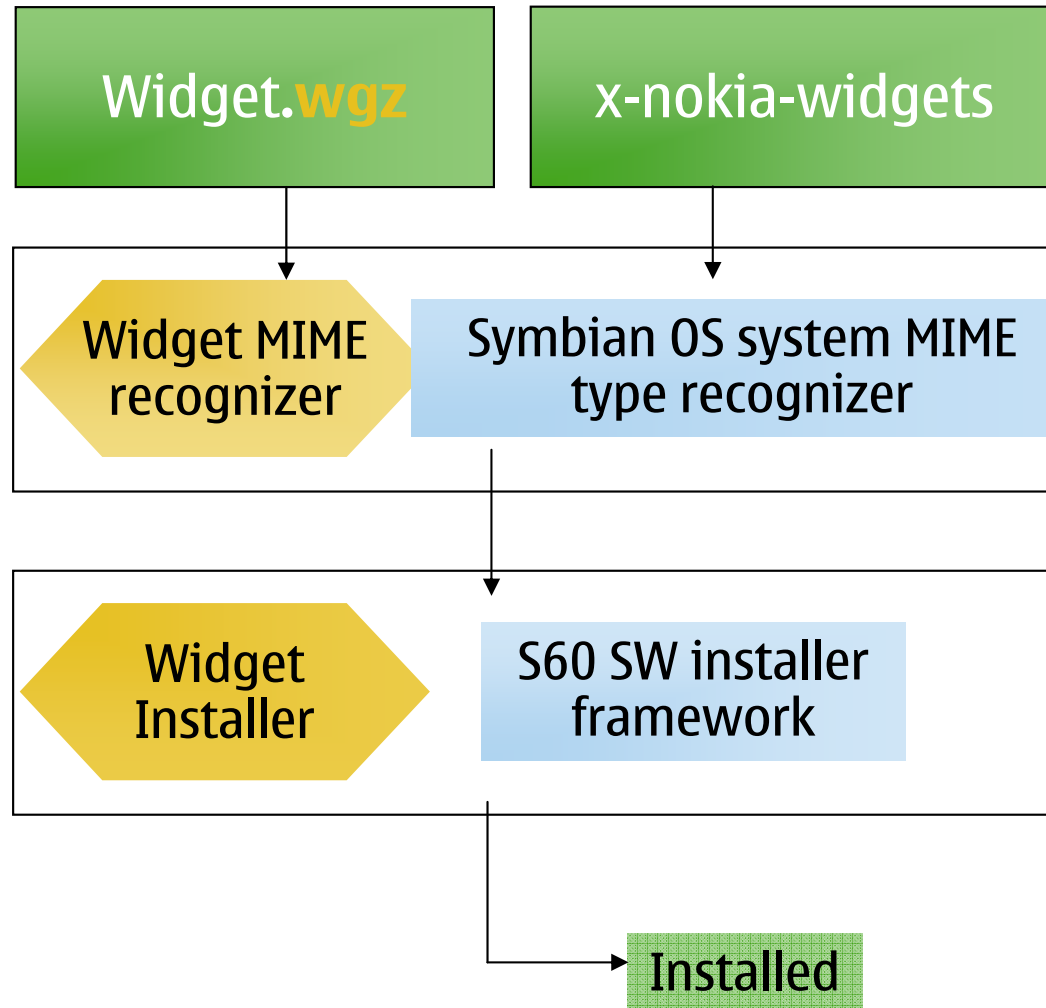


Widget Installation Package

- Widget installation file format
 - Compressed with any ZIP application
 - ZIP file contains html, CSS, js, icons, graphics and the widget info file
- Widget installation file extension
 - WidgetName.wgz
- Widget installation MIME type
 - x-nokia-widgets



Widget Installation Process



Widget APIs Introduction



NOKIA
Connecting People

Widget Object

- Widget object is a built-in module of the widget engine

`widget` or `window.widget`

- Widget object provides basic **utility functions** to manipulate **widget's properties**



```
var h = window.screen.height;
var w = window.screen.width;
if (h < w)
    widget.setDisplayLandscape();
```

```
widget.onhide = pause;
function pause()
{
    clearTimeout(timer);
}
```

• Widget methods

- `openURL(String:url)`
- `setPreferenceForKey(String:preference, String:key)`
- `preferenceForKey(String:key)`
- `prepareForTransition(String:transitionState)`
- `performTransition(void)`
- `setNavigationEnabled(Boolean:flag)`
- `openApplication(Uid, param)`
- `setDisplayLandscape(void)`
- `setDisplayPortrait(void)`

• Widget properties

- `identifier` [readonly, String]
- `onshow` [assigned callback function]
- `onhide` [assigned callback function]
- `isRotationSupported` [readonly, Boolean]

NOKIA
Connecting People

Menu Object

- Menu object is an extension from the widget object
menu OR window.menu
- Menu object provides interfaces to **manipulate the options menu and softkeys** (right and middle keys) of a widget




```
var opMenu = window.menu;
// ! menu is reserved

{
  ...
  opMenu.setRightSoftKeyLabel(
    'Back', toMainView);
}

function toMainView()
{
  ...
  opMenu.setRightSoftkeyLabel("", null);
}
```

• Menu methods

- **append**(MenuItem menuItem)
- **remove**(MenuItem menuItem)
- **replace** (MenuItem oldMenuItem, MenuItem newMenuItem)
- **getMenuItemById**(Integer:id)
- **getMenuItemByName**(String:label)
- **setRightSoftKeyLabel**(String:label,  function:callbackfunction)
- **showSoftkeys**(void)
- **hideSoftkeys**(void)
- **clear**(void)

• Menu properties

- **onShow** [assigned callback function]

MenuItem Object

- MenuItem provides interfaces to create **menu items and submenus (cascading) for the options menu**



- Menu Item methods
 - **new MenuItem**(String:label, Integer:id)
 - **append**(MenuItem:childMenuItem)
 - **remove**(MenuItem:childMenuItem)
 - **replace**(MenuItem:oldMenuItem, MenuItem:newMenuItem)
 - **setDimmed**(Boolean:flag)
- Menu Item properties
 - **onSelect** [assigned callback function]

NOKIA
Connecting People

```
function createMenu
{
    var settings = new MenuItem(
        'Settings', 10);
    settings.onSelect = showSettings;
    window.menu.append(settings);
}

function showSettings(id)
{
    document.getElementById(
        'setting').style.display = 'block';
}
```

System Info Service API

- System info service API is a **scriptable Netscape plug-in module**
 - **Subset of system properties** accessible
 - Memory, network and battery status, etc.
- System info service plug-in module is loaded via an **HTML <embed>** element

```
<embed type="application/x-system-info-widget" hidden="yes" ></embed>
```
- System info service plug-in module is **not loaded in Browser context**



System Info Service API

- Power properties
 - chargelevel [readonly, int]
 - chargerconnected [readonly, int]
 - onchargelevel [writeonly, function]
 - onchargerconnected [writeonly, function]
- Network properties
 - signalbars [readonly, int]
 - signalstrength [readonly, int]
 - networkname [readonly, string]
 - registrationstatus [readonly, int]
 - onsignalstrength [assigned callback function]
 - onregistrationstatus [assigned callback function]



System Info Service API

```
var sysInfo = null;

function init()
{
  // get the handle to the system info object
  sysInfo = document.embeds[0];
  var battLevel = sysInfo.chargelevel;
  sysInfo.onchargelevel = "batteryStatusCallback()";
}

function batteryStatusCallback() {
  var battLevel = sysInfo.chargelevel;
}
```



System Info Service API cont.

- Light methods
 - lighton(Int:lighttarget, Int:duration, Int:intensity, Int:fadein)
 - lightblink(Int:lighttarget, Int:duration, Int:onduration, Int:offduration, Int:intensity)
 - lightoff(Int:lighttarget, Int:duration, Int:fadeout)
- Light properties
 - lightminintensity [readonly, Int]
 - lightmaxintensity [readonly, Int]
 - lightdefaultintensity [readonly, Int]
 - lightinfiniteduration [readonly, Int]
 - lightmaxduration [readonly, Int]
 - lightdefaultcycletime [readonly, Int]
 - lighttargetprimarydisplayandkeyboard [readonly, Int]
 - lighttargetsystem [readonly, Int]



System Info Service API cont.

- Vibra methods
 - startvibra(Integer:duration, Integer:intensity)
 - stopvibra(void)
- Vibra properties
 - vibraminintensity [readonly, Integer]
 - vibramaxintensity [readonly, Integer]
 - vibramaxintensity [readonly, Integer]
 - vibrasettings [readonly, Integer]
- Beep methods
 - beep(Integer:frequency, Integer:duration)
- Memory properties
 - totalram [readonly, Integer]
 - freeram [readonly, Integer]



System Info Service API cont.

- File system methods
 - drivesize(String:drive)
 - drivefree(String:drive)
- File system properties
 - drivelist [readonly, String]
- Language properties
 - language [readonly, String]





Thank you

NOKIA
Connecting People

Forum **NOKIA**