



J-Spring 2007

Java ME - Device Fragmentation

Arjan Schaaf
luminis®



Even voorstellen

- 🦋 Arjan Schaaf
- 🦋 Senior Software Engineer bij luminis®
- 🦋 Sinds 1997 bezig met Java en sinds 2002 veel ervaring opgedaan met Java ME
- 🦋 Verder gespecialiseerd in OSGi en Swing user interfaces



luminis®

- 🦋 April 2002 opgericht
- 🦋 Vanaf het begin betrokken bij de NL-JUG
- 🦋 Kantoor in Arnhem
- 🦋 Vanaf September ook in Enschede!
- 🦋 22 medewerkers
- 🦋 Iedereen is van harte welkom op onze stand hier op de J-Spring!



Device Fragmentation

- ✦ Diversity in mobile phone platforms and implementations
- ✦ Different form factors
- ✦ Different capabilities
- ✦ Different implementations of standards (including errors!)
- ✦ Different generations of mobile phones



Praktijk situaties

- 🚩 Screen size
- 🚩 Support for bluetooth, SSL, etc
- 🚩 Vendor specific implementation classes
- 🚩 Vendor/device specific errors, for example
`Canvas.setFullScreenMode(boolean)` causes error when not called in `paint()` method. (Sony Ericsson Z1010)



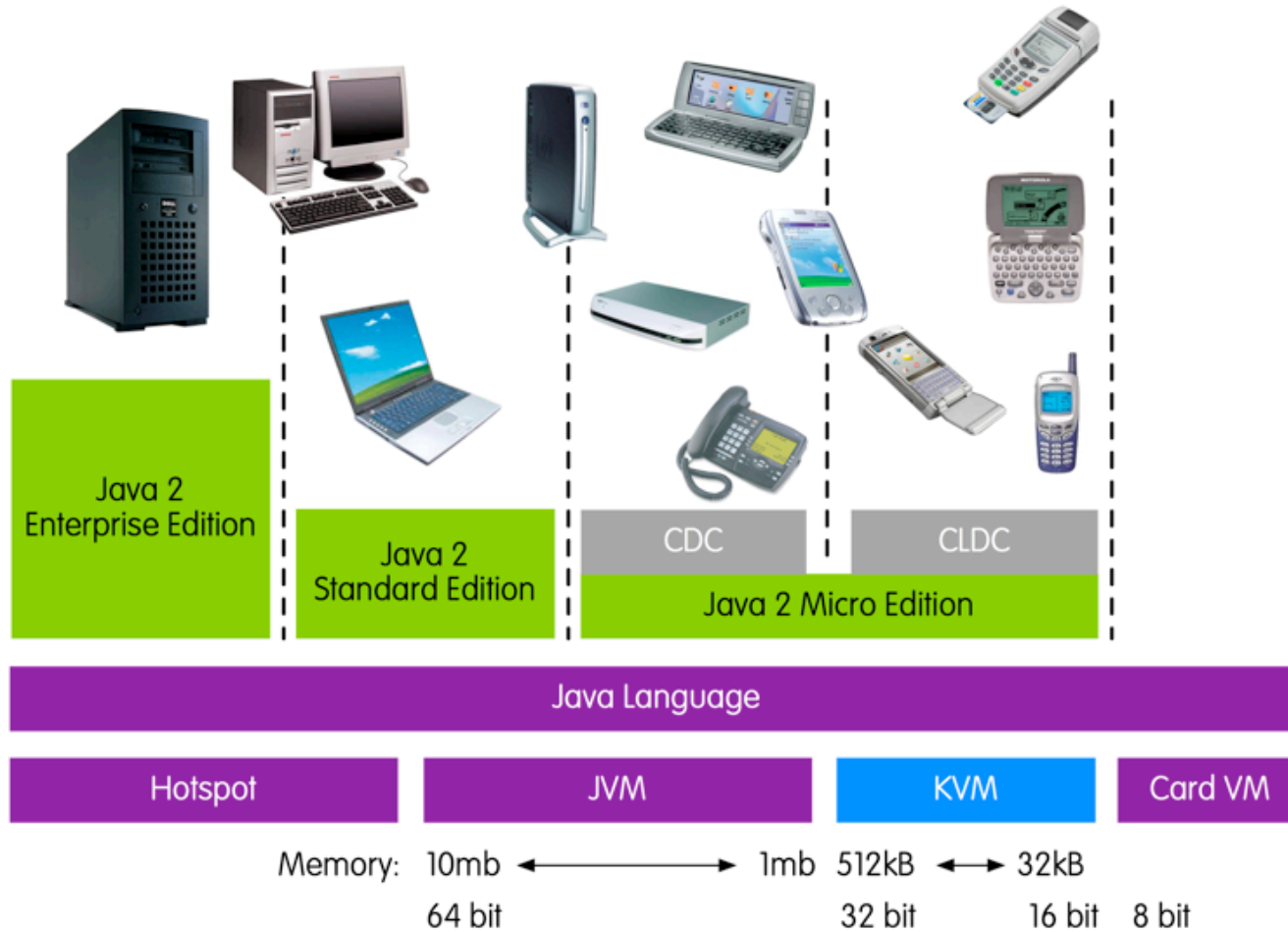
Future developments

- ✦ New standards should make life easier
 - ☛ JTWI (JSR 185)
 - ☛ MIDP 3.0

- ✦ But this does not offer a solution for the current install base!



Java ME Overview





CLDC configuration

- ⚡ CLDC 1.0 & CLDC 1.1
- ⚡ For resource-constrained devices like mobile phones, pages and simpler PDA's
- ⚡ CLDC 1.1 -> Floating point and weak reference support



MIDP Profile

🚀 MIDP 1.0 (JSR 37)

🚀 MIDP 2.0 (JSR 118)

☕ Enhanced UI

☕ Game API

☕ Connectivity (HTTPS, sockets, serial port)

☕ Over-the-air (OTA) provisioning

☕ End-to-end security model



Optional packages

- 🦋 Wireless Messaging API (WMA)
- 🦋 Mobile 3D
- 🦋 Scalable Vector Graphics
- 🦋 Location API
- 🦋 Web services
- 🦋 Bluetooth API



Device fragmentation initiatieven

- 🚀 WURFL
- 🚀 Adobe Device Central CS3
- 🚀 Netbeans Mobility Pack
- 🚀 J2ME Polish
- 🚀



WURFL

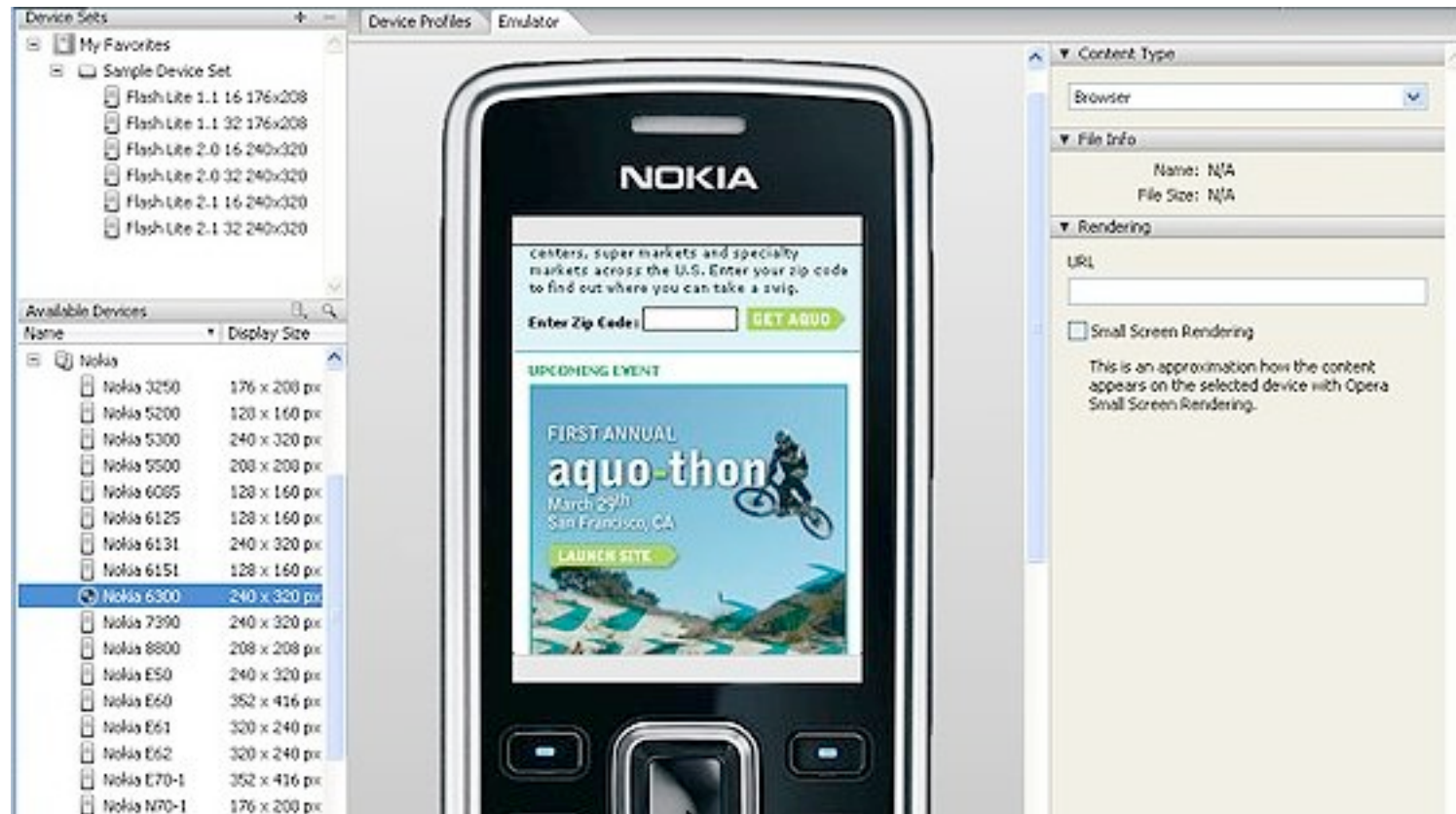
- 🚀 Wireless Universal Resource File
- 🚀 XML based
- 🚀 Description of capabilities and features of mobile phones.
- 🚀 Developed for WAP/WML
- 🚀 <http://wurfl.sourceforge.net>



Adobe Device Central CS3

- ✦ Not for Java ME but Flash lite
- ✦ But still an interesting setup
- ✦ Library of device profiles
- ✦ Integration with other Adobe development tools
- ✦ Loads of test / emulation possibilities
- ✦ Challenges with Flash lite are comparable with Java ME (different version with different capabilities)

Adobe Device Central CS3





Netbeans Mobility Pack

- ✦ Can be used with existing Java ME projects and new projects
- ✦ Based on 'project configurations'
- ✦ Each configuration creates a distribution Jar/Jad



Creating a configuration

DEMO



Custom code per configuration

```
try
{
    Class.forName("com.nokia.mid.ui.DeviceControl");
    com.nokia.mid.ui.DeviceControl.setLights(0,100);
}
catch(Exception e){}
```



Custom code per configuration

DEMO



Custom code per configuration

```
///if nokia  
com.nokia.mid.ui.DeviceControl.setLights(0,100);  
///endif
```



J2ME Polish

🚀 Project exist of multiple tools:

- ☕ Build tools
- ☕ Device database
- ☕ User interface utilities
- ☕ Utility classes
- ☕ Localization
- ☕ And more!



J2ME Polish community model

- 🚀 Dual licensing model
- 🚀 Open source: GPL
 - ☕ Free to use
 - ☕ Source code available
 - ☕ Developed applications have to be made available under GPL
- 🚀 Commercial license
 - ☕ Single license
 - ☕ Enterprise license



J2ME Polish build tool

- ✦ Apache Ant based
- ✦ Includes the following build steps:
 - ☛ Selection of supported devices
 - ☛ Assembling of the resources
 - ☛ Preprocessing
 - ☛ Complilation
 - ☛ Obfuscation
 - ☛ Preverification
 - ☛ Creation of JAD + JAR per device



J2ME Polish device database

- ✦ Like all the other device databases, but...:
 - ☛ Contains know issues
 - ☛ Groups devices



J2ME Polish device database

DEMO



J2ME Polish UI

- ✦ J2ME Polish enables you to customize your screens in CSS text file
- ✦ J2ME Polish provides extra custom components
- ✦ Enables direct input fields



J2ME Polish Java 5 syntax

- 🚀 Java ME is limited to Java 1.2 class format
- 🚀 J2ME Polish can enable:
 - ☕ Generics
 - ☕ Enums
 - ☕ Autoboxing
 - ☕ Foreach loops
- 🚀 Enable it in your build: `<postcompiler name="java5" />`



Example






Conclusions

- ✦ Developing Java ME applications is hard
- ✦ Dealing with the exception on one devices makes it even harder
- ✦ Supporting a larger set of devices is almost impossible
- ✦ A device fragmentation solution makes it possible to deal with this
- ✦ New standards hopefully bring some light in the darkness



Links

-  <http://wurfl.sourceforge.net>
-  <http://www.j2mepolish.org>
-  <http://www.netbeans.org/products/mobility>
-  <http://www.luminis.nl>



Tot ziens!

