



Java-Linux-Mobile Plattform

Robert Schuster  
Sebastian Mancke

# What is Jalimo

- Project to bring free Java to mobile and embedded platforms.
- Composition and maintainment of a complete solution stack for the target platforms.
- Support for the development lifecycle to target mobile devices.
- Currently supported: maemo, OpenMoko

## What is Jalimo not:

- No additional JVM!

## Why

- Small devices have special Java requirements
- Integration in mobile windowing systems
- Most free linux mobile distributions are not aware of Java
- Java developer need special support
- Java developer need a maintained platform to rely on – especially commercial ones

... and it's fun

... and because Tarent want's to have the same platform we are used to, even when making mobile bussines

# Components

- Mostly J2SE 1.5 focused
- Different alternatives for different requirements
- The goal to add more alternatives
- In focus: cacao, classpath, swt
- Also supporting: jamvm, swing, java-gnome, midpath
- Additional libraries: java-dbus, scio, (more coming)
- Toolchain support: OpenEmbedded, maven-pkg-plugin

# What we are missing most

- A VM which is as fast as cacao with the startup of jamvm
- Debugging support (JVMTI)

## What's next

- More library code and integration
  - Platform integration, Messaging, SQLite driver, ...
  - Helping OpenMoko to integrate and understand Jalimo
  - Attract more developers

## What would be great:

- Port Android APIs to other platforms
- Support for iPhone
- Get Nokia **officially** support free Java on maemo
- Get additional VMs for: WindowsMobile, Symbian S60
- Get Jazelle hacked (because they will not free it)

```
sed -i „e|Sebastian|Robert|“ speaker
```

2 more words to „why“



Free as in

# FREEDOM!

## How is it done?

- OpenEmbedded infrastructure
  - Self-hosting toolchain
- OpenEmbedded build recipes
  - „swt“, „dbus-java“, ...
- Maven2 Packaging plugin (maven-pkg-plugin)
  - Package your Maven2 program for specific distro

# A Java Bitbake recipe:

```
SRC_URI = „http://dbus-java.sf.net/${PN}-${PV} \
        file://makefile.diff;patch=1“
```

```
inherit java-library
```

```
do_compile() {
    oe_runmake \
        compile \
        JAVA_FLAGS="-cp ${STAGING_DATADIR_JAVA}/
        matthew.jar“
}
```

```
PACKAGES=„libdbus-java“
```

```
FILES_${PN} = „${datadir_java}/dbus.jar“
```

## Why this way?

- Every OS has its 'natural' way of installing software
- Following this means taking advantage of the functionality provided through this
- OE-build very flexible: 'foreign' distros, OE-made distro and custom addition (e.g. installs to /opt)

## Why this way?

- Those who want binary Java packages [in their distro] can build them (without knowledge of the details)
- Those who want a new supported package can just add the build recipe

# Resources

[jalimo.org](http://jalimo.org)

- Documentation of how to install binary packages for Maemo & OpenMoko
- Simple development & packaging tutorial
- Future: Hints for writing OE Java build recipes

[mvn-pkg-plugin.evolvis.org](http://mvn-pkg-plugin.evolvis.org):

- Project site and documentation (examples!) for packaging plugin

# Thank you!

CC-BY-SA 3.0 or GNU FDL 1.2 or, at your option, any later version.