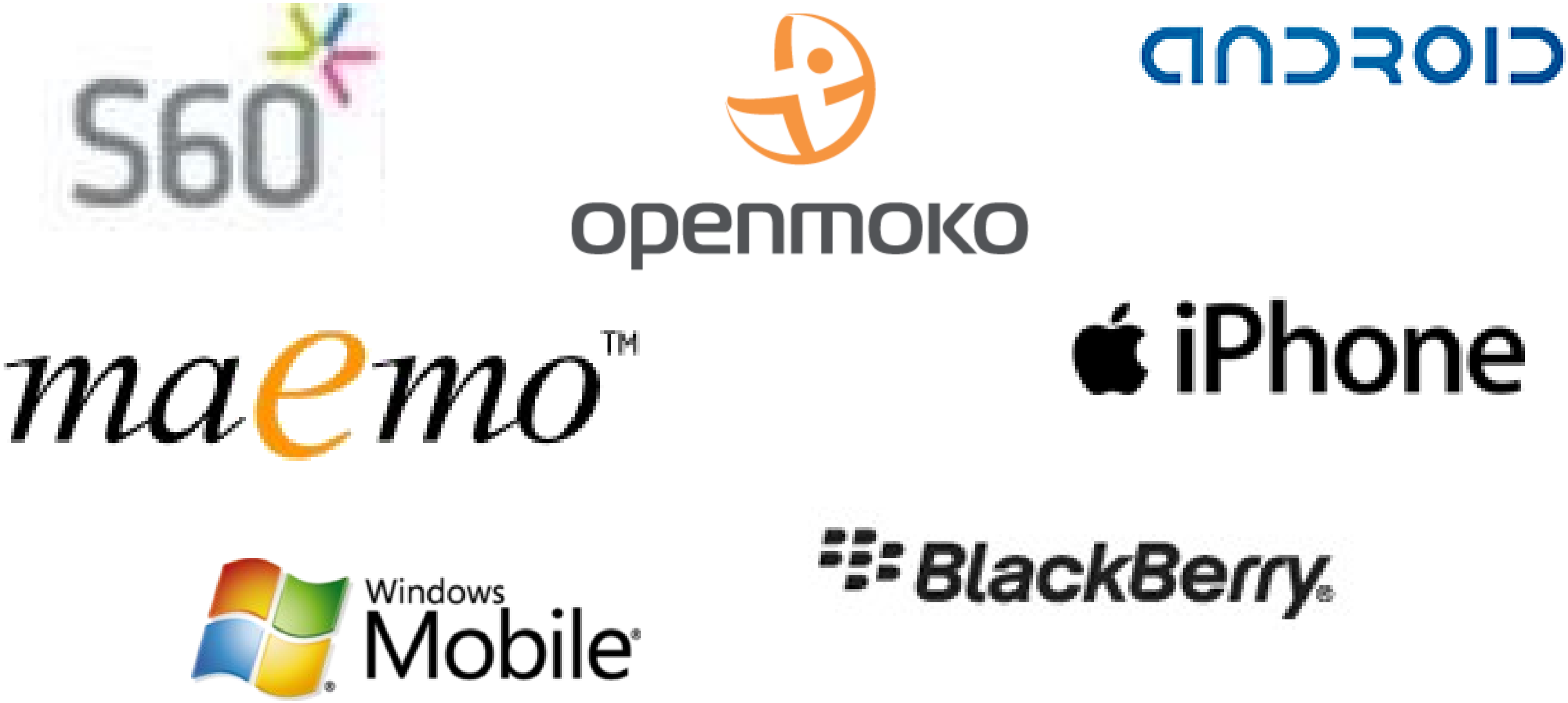
tarent

jalimo

mobile development platforms

Sebastian Mancke

Mobile platforms

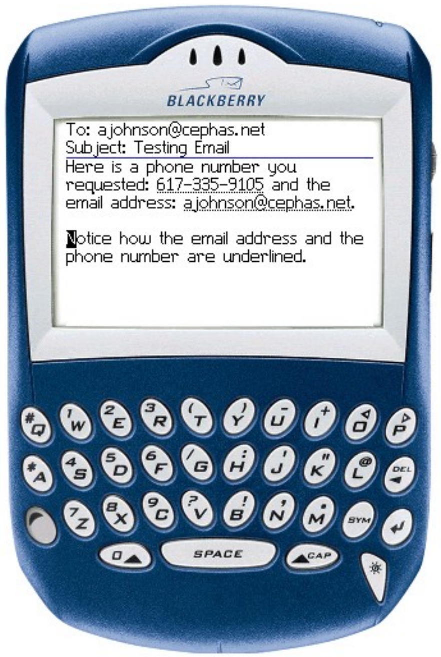


Windows Mobile

- Very Closed
- Main language: Visual C++
- Supported developing:
  - Visual Basic
  - .NET
  - ASP.NET
- Supported devices: many phones
- Company behind: Microsoft

BlackBerry

- Very Closed
- Main language: Java (J2ME) + proprietary extensions
- Supported devices: BlackBerry
- Company behind: RIM



To: ajohnson@cephas.net
Subject: Testing Email
Here is a phone number you requested: 617-335-9105 and the email address: ajohnson@cephas.net.

Notice how the email address and the phone number are underlined.

iPhone

- Very Closed
- Proprietary marketing model
- Based on BSD + iPhone Framework
- Main language: Objective C
- Device: iPhone (1 model)
- Company behind: Apple

- Unofficial gcc based SDK available
- You have to hack the phone

Symbian S60

- Proprietary, but open for development
- Based on Symbian
- Main language: Symbian C++
- Supported frameworks:
  - Open C (Posix porting layer)
  - Java (J2ME)
  - Python
  - Adobe Flash Lite / Web Runtime
- Supported devices: many phones
- Company behind: Nokia

Symbian Foundation

- New initiative of symbian companies
- Conflation of different symbian flavors
  - S60
  - UIQ
  - MOAP(S)
- Goal is to OpenSource Symbian until 2010
- Planed License: Eclipse Public License (EPL)

Android

- Free & Open? Not known jet!
- Based on Linux + Android runtime
- GUI Toolkits: Android
- Language: Java subset only
- Written from scratch
- Supported devices: emulator + HTC (announced)
- Company behind: Google + Open Handset Alliance

Maemo/ITOS

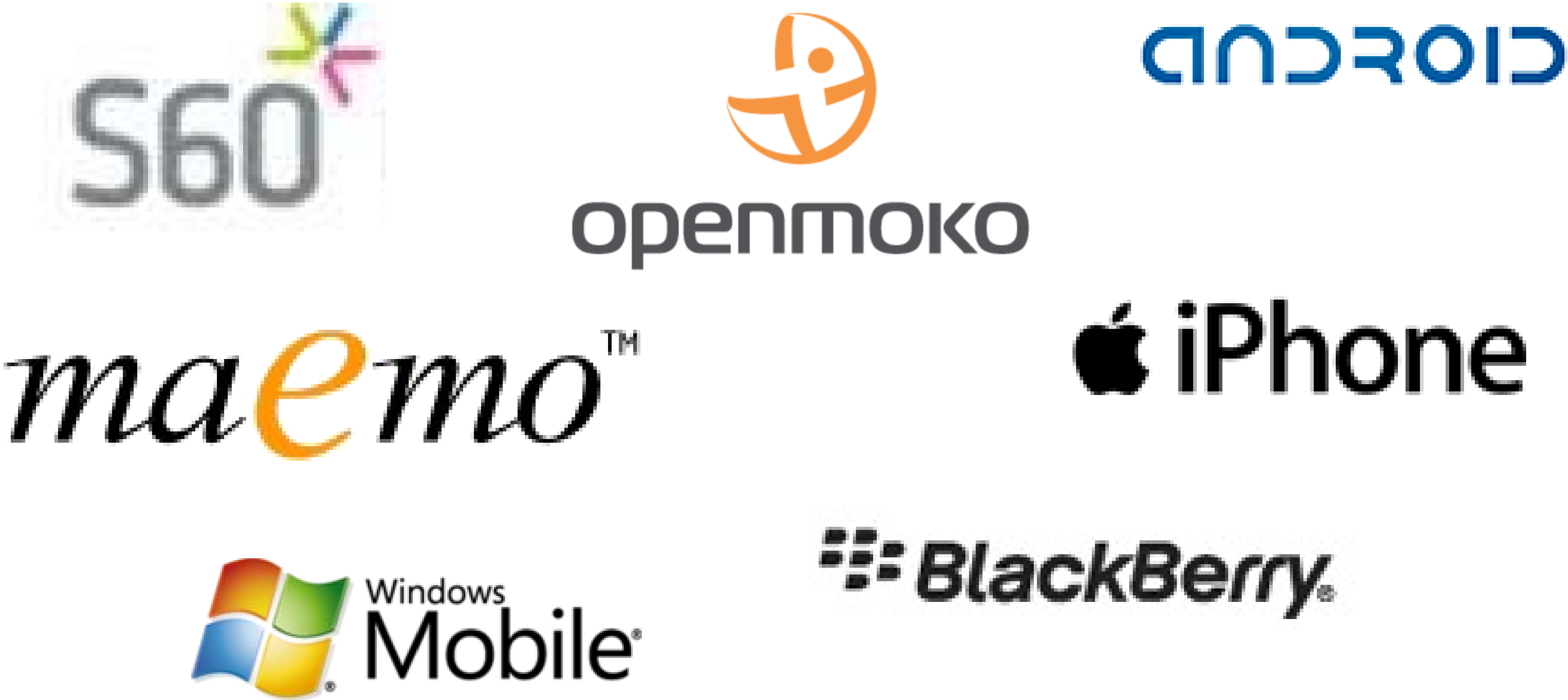
maemo™

- 95 % Free & Open
- Based on Linux, DBus and X11
- GUI Toolkits: GTK/hildon, QT (soon)
- Main languages: C, Python, C++
- Based on Debian (forked)
- .deb based packaging
- Supported device: n810 (1 model)
- Company behind: Nokia

OpenMoko

- 100 % Free & Open
- Based on Linux, DBus and X11
- GUI Toolkits: GTK, QT, EFL
- Main languages: C, Python
- Based on OpenEmbedded
- .ipk based packaging
- Device: Neo freerunner (1 model)
- Company behind: FIC/OpenMoko

Biggest problems in mobile development



Problem 1: Too many restrictions

- Only small control over the system
- Often only limited APIs are available
- Features are locked, signing processes are forced
- The core components are not replaceable

Free platforms change this:

openmoko maemo™ android

Problem 2: Too many platforms

- Large number of different platforms
- Few standards for cross platform development
  - J2ME
  - HTML/Web Applications
- J2ME often relies on proprietary extensions
- Some platforms cover only one device

solution:

- Cross platform development standards
- Or: Focus on widespreaded platforms

Problem 3: Different development approach

- Experienced developers want to reuse their knowledge
- Companies don't want to hire additional staff for mobile development
- Development Environments should be the same in mobile and desktop development
- Applications/frameworks should be reused
- Multi tier applications should use a homogeneous software stack if possible

Solved by:

openmoko maemo™ Windows Mobile

Problem 3: Different development approach

- Experienced developers want to reuse their knowledge
- Companies don't want to hire additional staff for mobile development
- Development Environments should be the same in mobile and desktop development
- Applications/frameworks should be reused
- Multi tier applications should use a homogeneous software stack if possible

Solved by:

For a small group of target developers, only!

openmoko

maemo™

Windows Mobile

What's wrong with J2ME?

Targets much of the problems, but ..

- Has too much restrictions
- Differs to much from usual Java
  - Completely different APIs
  - No code reuse
- Even if it is standardized:
  - Every manufacturer has different implementations
  - Applications have to be device specific
- Not powerful enough for much application types

What's about Android?

Targets much of the problems ...

- has only few restrictions
- uses real Java
- promises wide availability

... but ...

- still differs from usual Java
  - special APIs
  - code reuse only below the GUI
- is not standardized
  - is not designed for integration in other platforms
  - does not integrate other approaches

What's about Maemo & OpenMoko?

They do a lot of things right ...

- eliminate restrictions
- use real desktop toolkits

openmoko

... but ...

maemo™

- each platform has only one device
- no API standardization (between those platforms)
- focus only on a very special developer community

Cross platform solutions

Cross platform development is not a language problem, but a problem of cross platforms the libraries and toolkits.

Upcoming solutions

- QT
- FreeSmartPhone.org standards
- LWUIT
- Jalimo
- OpenEmbedded

What's about QT?

QT has the chance to become a solution!

- Good, powerful Toolkit
- Wide availability
  - Windows, Windows Mobile
  - Linux, OpenMoko
- Good Java bindings: Jambi

Acquired by Nokia

- Soon supported on Symbian and Maemo

FreeSmartPhone.org standards

Origin

- OpenMoko framework initiative
- Most work founded by OpenMoko
- Independent Project

Standardisation of Smartphone APIs

- DBus API design
- Reference implementation of services
- Base for any interaction of applications with the platform

LWUIT

- GUI Library for J2ME
- Similar to Swing
- Small but powerful
- Released as GPL from Sun, just now

Benefits of LWUIT

- Usable on all J2ME phones
- Easy portable backend
- Ports in work
  - Java AWT (CDC)
  - Microbackend (X, SWT, AWT, ...)
- Sun's way to unify mobile and desktop development?

jalimo

- Project to bring free full Java to mobile and embedded platforms
- Support for the development lifecycle to target mobile devices with java
- Integrative project, doing most work in upstream projects
- Current targets: maemo, OpenMoko, BugLabs, BeagleBoard, Irex Iliad

What is jalimo not:

- No additional mobile platform!
- No additional JVM!

Who is behind jalimo?

- Initiated by Tarent
- Active members from: Bug Labs & MIDPath

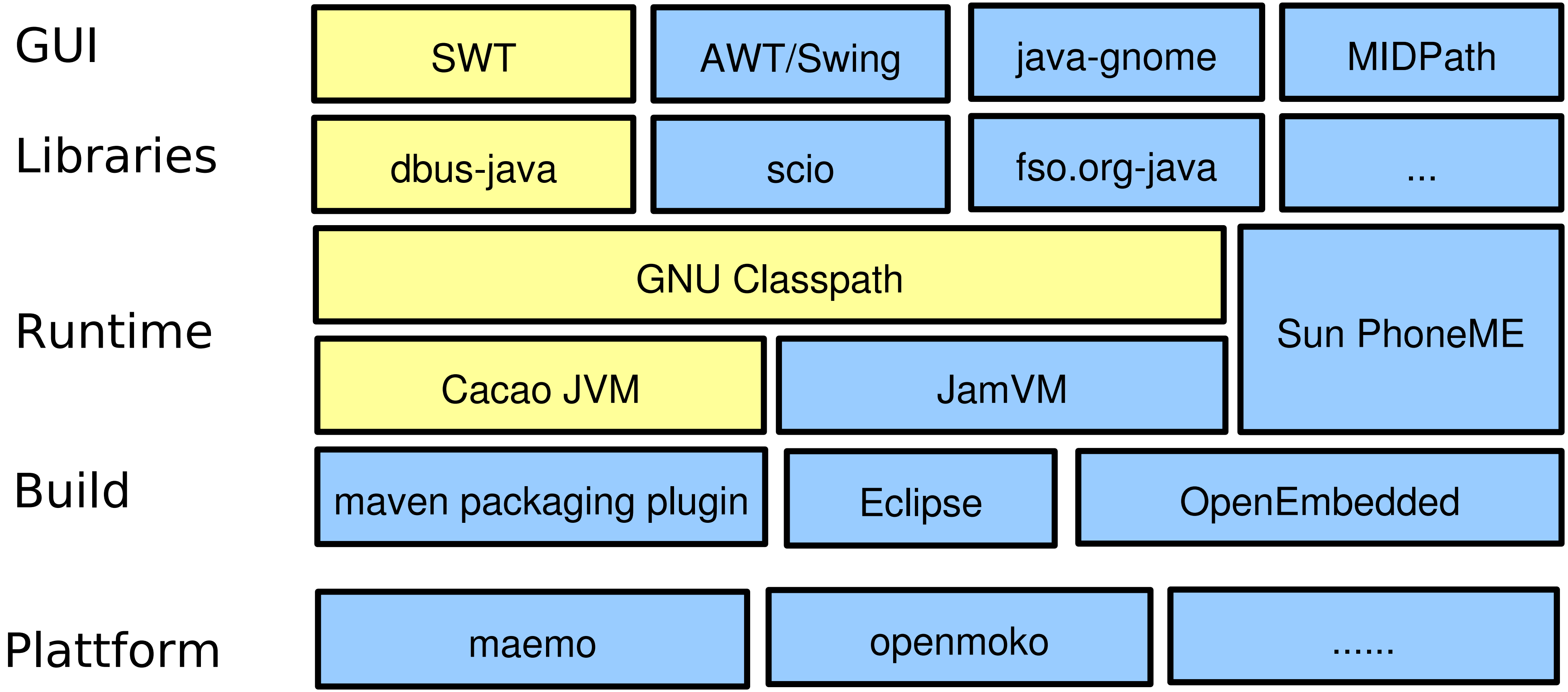
Why is tarent doing jalimo?

- Tarent has employed 60 people (~40 Java developer)
- Most of tarent's projects use java on the server side

- One technology among the applications!
- Use the same staff for mobile and server side development!

Parts of jalimo

- J2SE 1.5 focused
- Different alternatives for different requirements



GUI: SWT | AWT/Swing | java-gnome | MIDPath

Libraries: dbus-java | scio | fso.org-java | ...

Runtime: GNU Classpath | Sun PhoneME

Cacao JVM | JamVM

Build: maven packaging plugin | Eclipse | OpenEmbedded

Plattform: maemo | openmoko | ......

OpenEmbedded

- cross-compile environment
- generation of complete distributions
- support for 1000's packages
- builds the base for
  - OpenMoko
  - Angstroem
  - Jalimo
  - Buglabs
  - Irex Iliad
  - ...
- Nice solution for multi platform development

Different level of openness

Closed

- usual mobile, iPhone

Open for application development

- Symbian, Android, J2ME phones, Windows Mobile, BlackBerry

Open and customizable

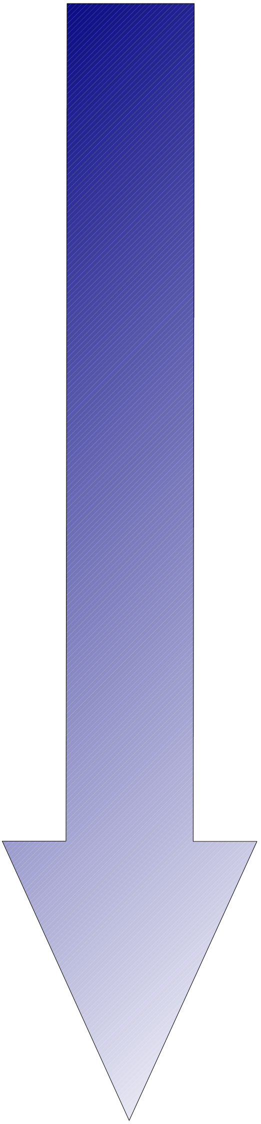
- maemo

Free and open

- OpenMoko, OpenEmbedded

Free, open and standardized

- Nothing yet! (FSO has a chance!)



GPL

Thank you!

Mobile@FrosCon:

- Freie Software für das iPhone
  - Ralph Pöllath
  - 11:15 Uhr, HS 4
- Openmoko 2008
  - Michael Lauer
  - 14:00 Uhr, HS 1