Beyond Traditional Mobile Linux

by Carsten “Stskeeps” Munk,
Mer project architect
http://www.merproject.org

Mobile Linux up to 2011

- Moblin, MeeGo, Maemo, LiMo, OpenEmbedded (Yocto, WebOS), OpenWRT, etc..
- OpenMoko-centric (QtMoko, FSO/SHR, etc.)
- Android (Replicant, Ophone, Baidu Yi, B2G, etc.)
- Familiar, Access Linux Platform, Ubuntu Mobile/MID, Mobilinux
- ... and many many more

What do most of them have in common?

- Many of them are now dead or zombie projects.
- Many were centric around specific vendors or specific devices.
- Many of them were wasted effort for the Mobile Linux community.

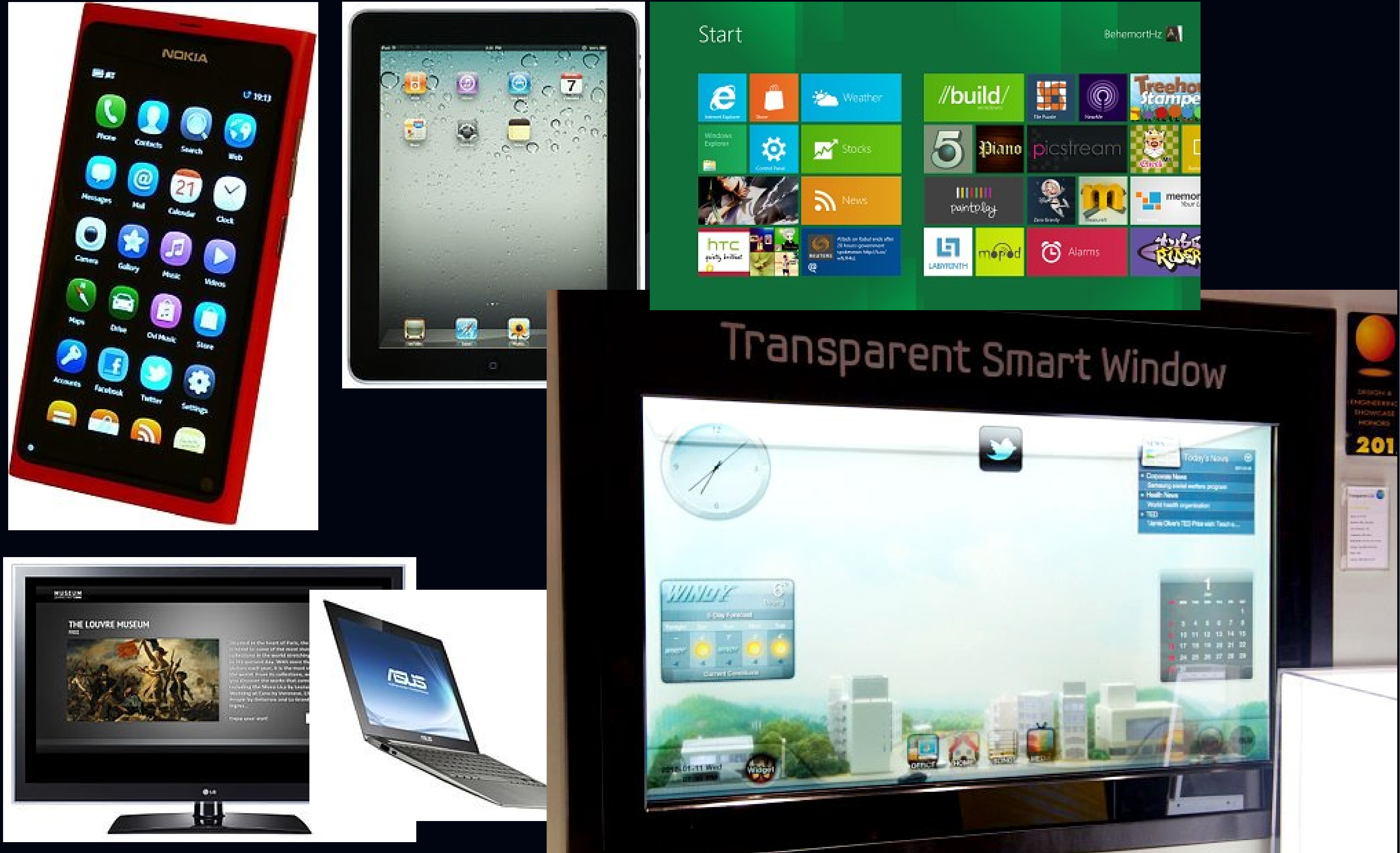
Mobile Linux in 2012

- OpenWRT, OpenEmbedded (Yocto)
- Android & Boot2Gecko
- Tizen, Mer, WebOS, Linaro efforts
- Intentionally not mentioning single-hardware/vendor OS'es, UI projects or open hardware
- Linux in general in all sorts of consumer devices
- Why not Fedora, Debian, Ubuntu, Slackware, etc..?

The world around us

If we were to interpret the world around us through what we see in popular Linux distributions and attitudes

There's just one problem about that..



This is not how real life looks like anymore.

- But but but, what about KDE, GNOME, all our projects centered around the PC as the primary form of computer usage?
- We're experiencing the beginnings of a paradigm shift in how people use computers.

> *“the notion of a major change in a certain thought-pattern — a radical change in personal beliefs, complex systems or organizations, replacing the former way of thinking or organizing with a radically different way of thinking or organizing”*

But..

- A lot of open source projects are built around this old paradigm – centered around the PC.
- Is this the end of user-facing Linux? Are all our efforts wasted?
- There's three options for projects in the face of a paradigm shift:
  - Perish and die
  - Evolve and adapt
  - Rebase and reinvent

Don't panic. This isn't the end.

- Remember origins of Desktop Linux?
  - Difficulties to get any kind of hardware support
  - Need to write tools and UI frameworks, etc..
- We're in a much better situation when it comes to mobile Linux
  - Because embedded Linux is in use everywhere
  - We have OSS UI frameworks that are suited for the post-PC experience
- It's time to be ahead instead of playing catch-up

Problems in mobile Linux

- It's fragmented to bits
- There's not properly open generic platforms to avoid this fragmentation
- Knowledge and skills are spread thin
- No common application stories

If we were to picture Mobile Linux as seen by discussions in it

While it should be more like

My personal top 5 disagreements seen in Mobile Linux

- RPM vs Debian packages
- Hardware adaptation, chipset politics and openness in those
- UI choices and frameworks
- Project openness and governance
- Views originating from a misunderstanding that mobile is same as desktop
- The only of these that real consumers remotely care about is the experience

But in the end, peple seem to end up with similar kind of stacks

- GCC, (e)glibc, GNU utilities or busybox, wayland, xorg, directfb, some middleware, apps, hardware adaptation.. etc
- So why aren't we working together on the parts that we actually all need?
- .. and importantly: keeping out the politics by removing some of the parts that cause it from the equation
- More code/effort, less politics, please!

How is this connected by the paradigm shift?

- There is a clear tendancy that we're moving towards many many different types of devices in our home and surroundings.
- Many many different types of devices will need to be made
- Bigger companies can't easily scale and diversify their portfolio
- But small to medium enterprises can put out many different products due to their numbers

And how is this connected to open source?

- A open source device has exactly the same base system burden as company does, in order to do a product
- A burden is time and time is money. There is not a unlimited supply of open source contributors anymore, so no unlimited time.
- A open source device will have to compete for their market share (also contributors) just like any company does

So why not work together and share the burden?

Encouragements

- Start projects to work on common cores, judging by the discussions and stacks:
- make a core that uses RPM, one that uses Debian packaging, another OE, etc.
- Leave out chipset politics and hardware specifics, leave out UIs and don't tie down things with trademarks.
- Be open so you can collaborate properly.
- Work on common app stories.

My own initiative

- The Mer Project – merproject.org
- An open, mobile-optimised, core distribution aimed at device manufacturers; powered by Qt/QML and HTML5 - openly developed, inclusive, and meritocratically governed.
- RPM-based, so if you want to start your own DEB-based core, you're welcome :)

What's in it for open hardware?

- Faster time to market (no need to remake an entire stack)
- Make open hardware easily that people would actually buy!
- The best way to ensure software freedom is to create it.
- Build the future!

Make open source projects that..

- Makes it a benefit for the user to have open hardware and not walled gardens
- Show to the world how you make open hardware, examples, reference
- Motivates repurposable hardware

On common app stories

- There is a clear tendancy towards HTML5 as a common app story.
- This is a benefit for open hardware as they don't need to provide more than runtimes to have an application story.
- Remember that just because your apps are HTML5, it doesn't mean the rest of your system has to be.

Thank you for listening

- Questions?

- If you'd like to contact me, mail carsten@merproject.org
- If you'd like to know more about Mer, please visit merproject.org or visit us and have a chat in #mer on irc.freenode.net