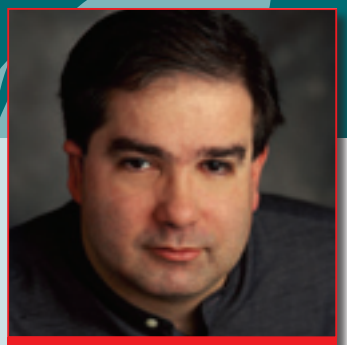
Open Phone, Closed Phone



Bill Weinberg

Reach into your pocket or your purse and grab hold of that indispensable companion, your cell phone. Stare fixedly at its little screen and keypad, and try to picture it running Linux. What do you see?

Some pretty heavy hitters have already done their own imagining, and in 2005 almost two dozen phone models shipped with 'Linux Inside' from familiar companies like Motorola, NEC, Panasonic and Samsung, and less well-know mainland Chinese manufacturers like Datang, e28, Haier, Huawei and ZTE. Among them, they shipped over ten million Linux-based handsets during 2005. Motorola led the charge with family of a dozen smart phones in China, while NEC and Panasonic did their share shipping millions of mid-tier 'feature phones' to Japanese NTT DoCoMo subscribers.

For many free software types, the concept of an 'open phone' is tantalising. Just put Linux on a mobile phone and hackers and enlightened end-users could customise skins and ring-tones, add and improve applications, upgrade the kernel, tweak performance, and variously hack the devices to their hearts' content, all without the annoying intervention of wireless operators. Slot in a wireless card to enable WiFi or Bluetooth and you wouldn't even need those bothersome operators with their monthly minutes, roaming charges and boggling line item charges.

The vision of handset manufacturers, however, is not shaped by end-users, but by their immediate customers: wireless carriers and operators. Wireless network owners present mobile phone OEMs with requirements specifications that rival the Tokyo telephone directory in heft, often containing over 5,000 unique line items. While many of these specs relate to end-user features like battery life or dialling behaviour, many more focus on radio spectrum usage and power output, security, and emergency services availability. Nowhere in technical tomes will you find requirements for open OS and software stacks, to say nothing of end-user hackability.

The vision of handset manufacturers **is not shaped by end-users, but by their immediate customers:** *wireless carriers*

So why are leading phone suppliers turning to Linux in increasing numbers for their shiniest and sexiest telephony toys? First and foremost, handset OEMs look to Linux and Open Source to cut development and deployment costs. Linux lets device OEMs choose for a greater range of silicon, platform providers, and middleware and applications ISVs. Lower or zero cost for deployment is also very attractive, improving margins in a highly competitive, high-volume marketplace.

Second, mobile phone suppliers are seeking to satisfy carrier/operator service delivery requirements. Today, most carriers and operators deploy their value-added services (voice-mail, IM/MMS, web services, games) through a mix of fixed native applications and downloaded apps written in Java or to specialised 'sandbox' APIs like Qualcomm Brew. These virtualised delivery platforms do offer predictability and security, but have not lived up to their promise of performance and portability. So, phone manufacturers seek to position Linux user-space applications as the next 'can opener' for value-added services delivery, with native run-time performance and more options for integration and optimisation.

Third, handset OEMs are indeed interested in courting developers, just not necessarily the ones who want to hack their wares. Phone manufacturers want to cultivate an ecosystem of 'post-platform' software developers, that is, ISVs, integrators and individuals who will build or re-target applications to run on a family of Linux phones, much as those same developers do for SymbianOS or for Windows CE handsets. virtual

For 'pre-platform' development, phone OEMs look upstream, working with .orgs like OSDL's Mobile Linux Initiative, ARM, ARM licensees (like FreeScale, Intel and TI) and indirectly with kernel.org and Open Source projects. While software developers at mobile device OEMs may participate in open source software projects, their employers seek stability, deploying derivative commercial platforms from MontaVista, PalmSource and others, or their own carefully controlled home-grown mobile distributions.

'Open and Free' or 'Open for Business' - does a new digital divide yawn before us? The dilemma is not new - several dozen mass market consumer electronics devices have been publicly hacked: TiVO, Linksys WG54T, even the Apple iPOD - in some cases, with full approval or encouragement of the device manufacturer, in others to their corporate dismay. A case in point is Harald Welte's open-ezx.org, founded to "to gather information about the Linux-based Motorola EZX phone platform... to provide a 100% free software stack for those phones, especially... to avoid any proprietary filesystem and/or device drivers." Motorola's chief mobile architect responded to Welte's challenge, commenting that his company had no immediate plans for native Linux application support, due to carrier concerns about network security, and interoperability. Another example is of course TiVo, which helped to create a developer and hacker community around its platforms, but did not open every aspect of device operation.

In most situations, unauthorised re-flashing, reprogramming and tweaking are tantamount to cracking the plastic cover and voiding manufacturer warranties. In a few cases, any sort of hack is actually illegal, in the U.S. at least. Tweaking software radio drivers violates Federal Communications Commission regulations and defeating DRM mechanisms puts hackers squarely at odds with the Digital Millennium Copyright Act and other statutes.

So, are we likely ever to see a 100% open phone? Do we really need or want one? The market votes with its chequebooks and credit cards. Unless the great mass of end-users perceive a real benefit in deep customisation of phones and other intelligent devices, or ISVs, carriers and operators see real advantages and business opportunities in a more open phone definition, closed devices will continue to dominate, whether or not they are based on open source platforms. The debate, however, has just begun in earnest. To join in, participate in the OSDL Mobile Linux Initiative. You can also send the author your 'wishlist' for an open phone definition at <mli_discussion@lists.osdl.org>

Bill Weinberg is the OSDL's representative to the Mobile Linux Initiative