



Mobilizing Open Source: Creativity, Contribution, and Community

David “Lefty” Schlesinger
Director, Open Source Technologies
ACCESS Co., Ltd.

www.access-company.com

TS-7101

What's It All About?

- A quick overview of the ACCESS Linux Platform
 - Why PalmSource (now part of ACCESS) made the decision to go with open source rather than continue on the proprietary operating system path
- How working with open source software in the context of a commercial product differs from “doing it all” in-house
 - New opportunities and new challenges, the importance of “best practices”
- What are the challenges to broad adoption of open source-based platforms?
 - How are companies like ACCESS and others working to address and respond to these?
- How do the various “mobile open source initiatives”, such as LiPS, the Linux Foundation, and others, impact on all of this?
- What is the outlook for open source platforms on mobile devices?

What's Good About Being Proprietary

- You can control your own fate
- High functionality, and tight integration, particularly when there's strong overall architectural vision
- (Vaguely) predictable timeframes, in theory anyway

What's Bad About Being Proprietary

- You have to do pretty much all the heavy lifting yourself
- You cannot rely much on the work of others, even when it's useful and germane
 - Incorporating open source, in most cases (i.e., code released under non-BSD-like licenses), is problematic
 - Thus, you find yourself needing to “reinvent the wheel” in many cases, rather than being able to leverage existing work

PalmSource Goes **Open** Source (and **Why**)

- Proprietary operating systems are at an inherent disadvantage
 - Driver development is idiosyncratic and difficult
 - Direct silicon vendor software support is scarce
 - Learning curves are high, many aspects are idiosyncratic
- Silicon suppliers typically test out new designs using Linux
 - Drivers are readily available
 - Time-to-market is reduced
 - Knowledge is wide-spread, expertise is increasingly common
- The potential exists to leverage the efforts of others, if those efforts are approached in the right way

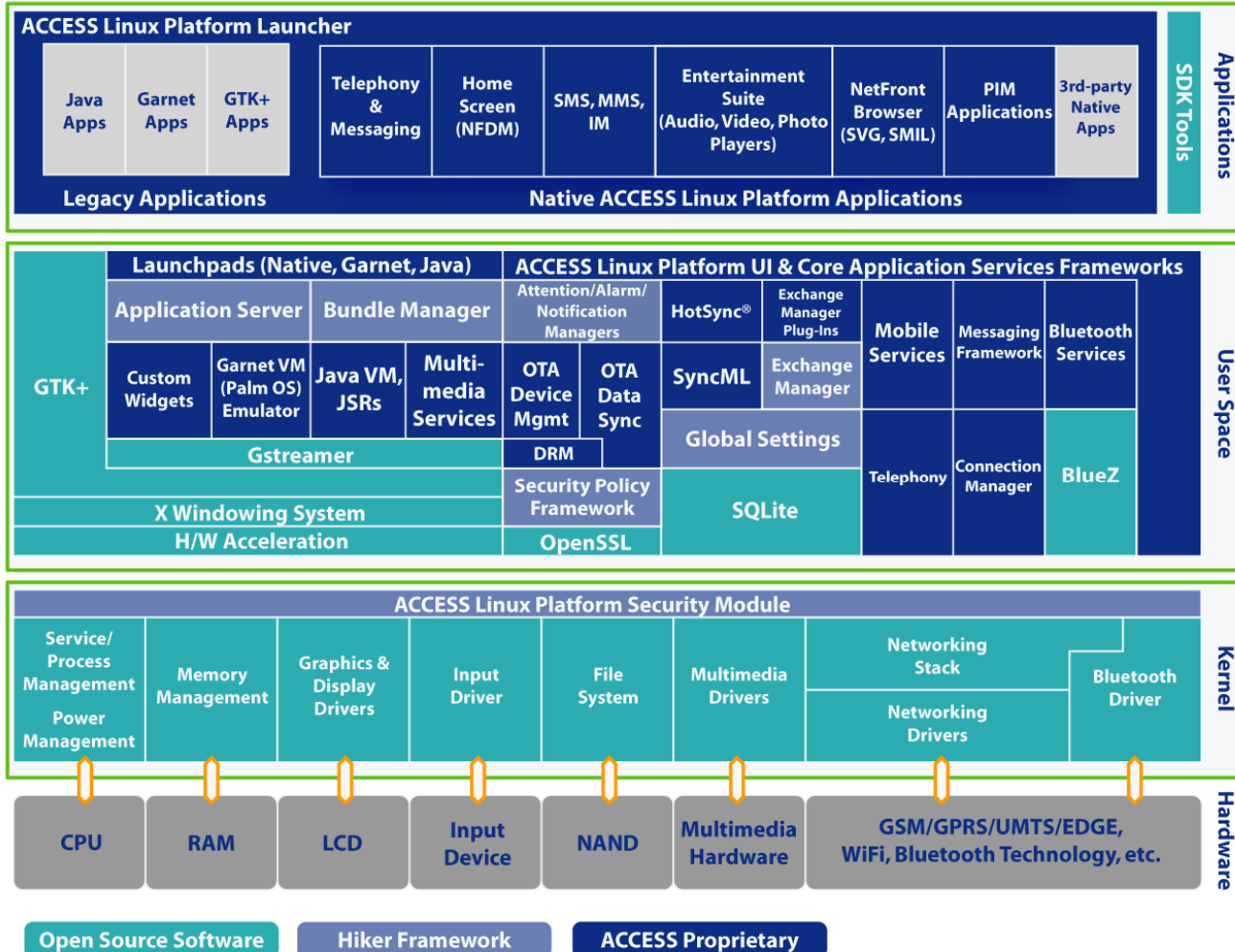
Advantages of Open Source for Mobile

- High Customizability
- Modularity and Flexibility
- Stability
- Existing mobile proprietary platform solutions have limitations
 - Customization is difficult
 - Pace of innovation is controlled
 - Complexities and support issues impede time to market
- Increasingly complex hardware and device requirements are raising the importance of open software platforms

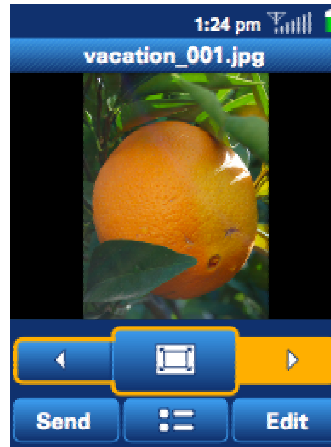
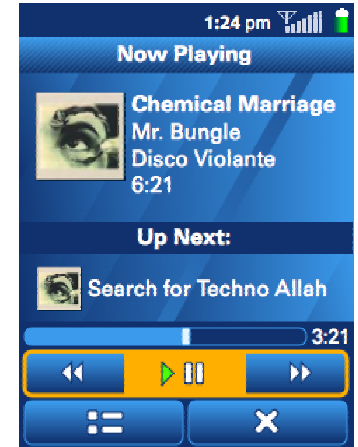
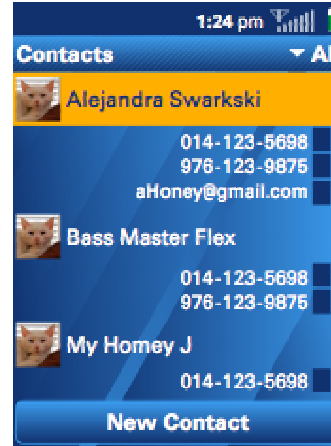
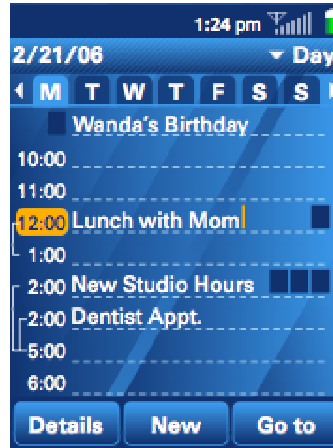
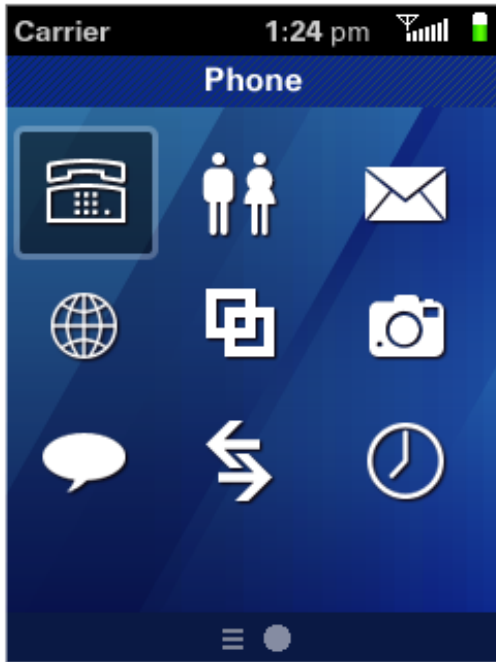
The ACCESS Linux Platform: Key Objectives

- An Open, Flexible, and Easy-to-Use commercial grade Linux-based software platform for smart mobile devices
- Combines best-in-class open source components with proven technologies and expertise from ACCESS and PalmSource
- Delivers a complete and fully customizable platform solution for smart mobile devices to handset vendors and mobile operators that allows competitive differentiation

The ACCESS Linux Platform



Easy, Consistent Mobile Experience





Opening Up: What's Different?



How Proprietary Software Gets Written

- You tell somebody what to do
- They do it (with any luck)

How Open Source Software Gets Written

- You tell somebody what to do
- They do what they **feel like**

What's Different About Open Source?

In the corporate world...

- People work to meet requirements
- People work according to schedules to meet deadlines
- People work on a project until it's finished (or cancelled)
- People work to meet specific (hopefully high) quality goals

In the open source world...

- People work on what they find interesting
- People work when, and as much, as they feel like
- People work on a project until they get bored
- Quality levels are often negotiable

Bottom line: You can't expect that the "open source community" is going to work to meet *your* goals, unless those goals are in the community's interest. (You can't necessarily expect it even then.)

Another Challenge: Licensing Madness

- The Open Source Initiative recognizes no fewer than 58 open source licenses, from “Academic” to “Zope”
- Some of them are written in language which is scarcely “legal”; some appear to make assertions contrary to, e.g., actual, copyright law
- There is a general lack of understanding of the meaning and implications of even the most widely-used open source licenses
 - This is true within the open source community as well
- Uncertainty makes companies nervous; it makes lawyers really nervous
- There is, happily, a trend to move in the direction of fewer (and hopefully, better written) open source licenses

Challenges on the Open Source Side

- There's little understanding, for the most part, of the issues which are peculiar to mobile devices
 - Regulatory requirements
 - Operator requirements
- Efforts on the part of companies like ACCESS, Nokia, and others are helping to better educate the community
- A phone is and isn't a "personal computer"
 - "Toaster" or "energy converter"?
 - Drills vs. holes

What Needs to Be Different?

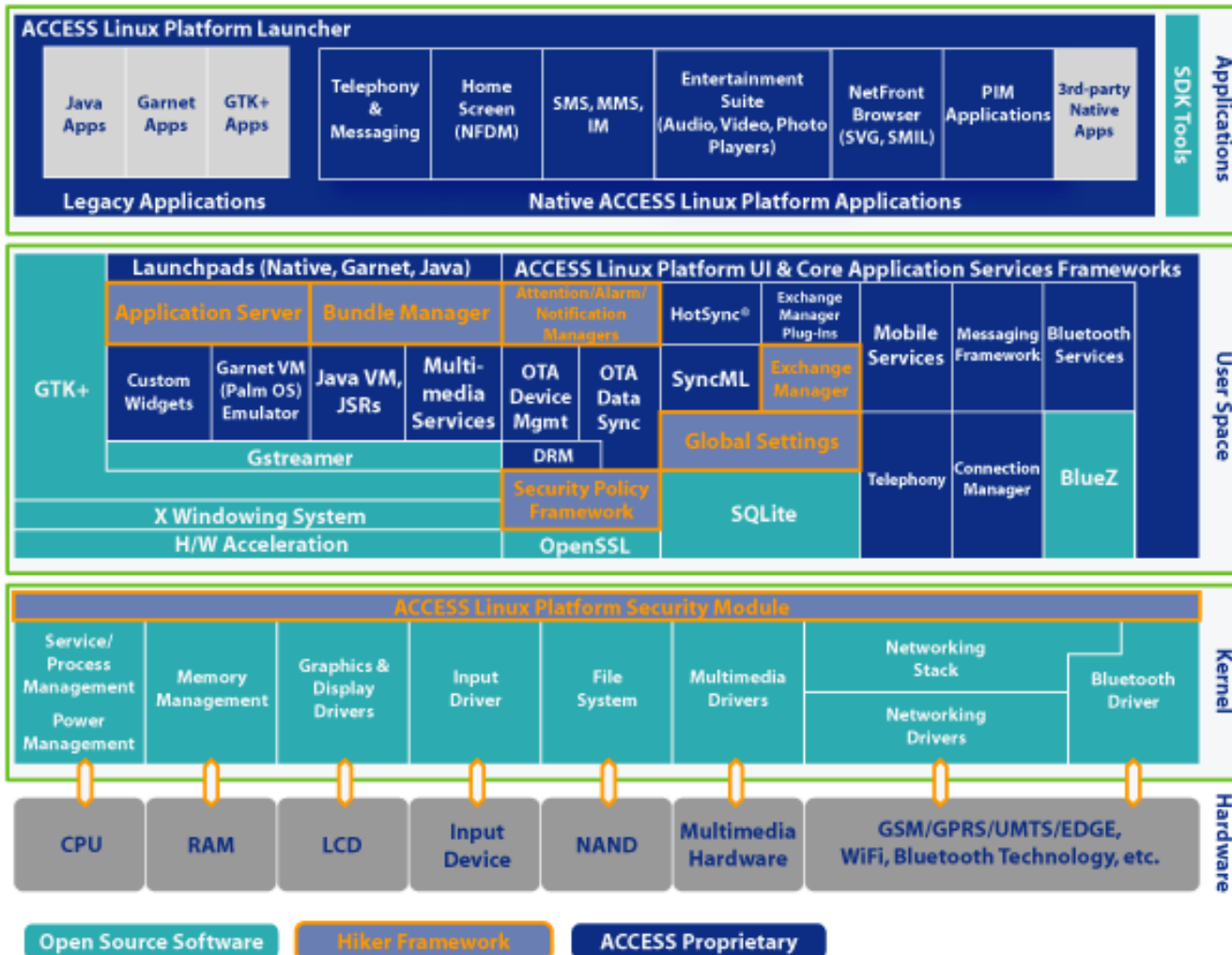
- People work on what interests them
- People work on that from what they stand to benefit in some way
- In order to engage the open source community in your effort, you need to make it interesting and rewarding

The “Hiker Project”

- ACCESS released a number of core middleware components as the “Hiker Project” (www.hikerproject.org)
- Hiker components
 - Manage the “application lifecycle”: Application Manager, Bundle Manager
 - Facilitate inter-task and task-to-user communication driven by event- and time-based stimuli: Notification Manager, Attention Manager, Alarm Manager
 - Enable transfer or use of structured data between applications and devices: Exchange Manager
 - Handle policy-based security: Security Policy Manager, Hiker Security Module
 - Manage preferences and setting: Global Settings
 - Some useful utility components: Abstract IPC, libsqlfs



The Hiker Framework



Hiker's Goals (Beyond Helping Get Our Work Done)

- To address some noteworthy gaps in existing open source stacks
- To reduce duplication of effort and fragmentation in these areas
- To engage with, reach out, and contribute to the open source community in a concrete way



The Need for “Best Practices”

- Licensing and copyright issues are complex
- Inadvertently creating “derivative works” is a significant risk to mixed proprietary/open source projects
- Participating in open source projects can be tricky
- Clear guidelines and direction need to be provided to all participants, not just engineers

Key “Best Practices” Areas

- Understanding the most important licenses, what they mean, and how they can impact design decisions
- Understanding the basics of copyright and intellectual property law
- Understanding how to participate in an open source project
 - This includes not only things like coding conventions, but also things like “how to behave on a mailing list”
- Understanding how (and when) to create a new open source project



Why Aren't All Phones Open?



What's Inhibiting Linux Adoption?

- Fragmentation: Every Linux system is different
- The Hobbyist Mentality (or “Linux: the Operating System for People Who Love to Write Operating Systems”)
- The Problem of “Good Enough” (or “Open Source Software: 80% as Good as the Last Guy Who Worked on It Needed It to Be”)
- The Problem of “Mostly Good Enough, Most of the Time” (or “Tsujiigiri [辻斬] as a Software Test Methodology”)
- It's just plain difficult (or “If it was easy, everybody'd do it”)

Fragmentation

- “They say their phone runs Linux!”
“How can they tell?”*
 - * Dorothy Parker, on being told of the death of American President Calvin Coolidge
- There is little to no commonality between the “Linux” on one phone and the “Linux” on another
 - Third-party opportunities are reduced, at best
- Most systems are closed, anyway...
 - Open source, but behind a locked (or very tiny) door

Tensions Between “Open”, “Free”, and “Proprietary”

- For the average consumer, all software might as well be proprietary
 - “Just download the sources, configure things, build it and get it installed. Oh, you might have to fiddle with it a little...”
 - Not an option for 99% of all end users
- Typical phone users don’t “get” “software freedom”
 - They’re not wrong, either...
 - The typical cell phone user doesn’t much care what OS his phone is running
 - Given the choice between having an entirely “free” system and listening to mp3s, most users will unhesitatingly pick mp3s
 - Freedom also means that you have to let other people be free to make choices that you might not...

Gaps in the Landscape

- Open source software stacks, when it comes to mobile, have significant gaps
 - e.g., telephony, power management, etc.
- Mobile devices have a very different use paradigm than desktop systems do
 - More “task oriented”, with more frequent interruptions
- Efforts like the Hiker application framework are meant to address some of those gaps
- Other areas—those above the “value line”—will get addressed in competing, and frequently proprietary, ways
 - The value line is continually rising!

“Good Enough” Isn’t Good Enough

- Your typical end user has a very low tolerance for cell phones which behave in unpredictable or quirky ways
- Testing of open source software is frequently spotty
- Commercial organizations, which are obliged to do extensive and structured testing anyway, can make significant contributions here
- Tools like Coverity can be very helpful

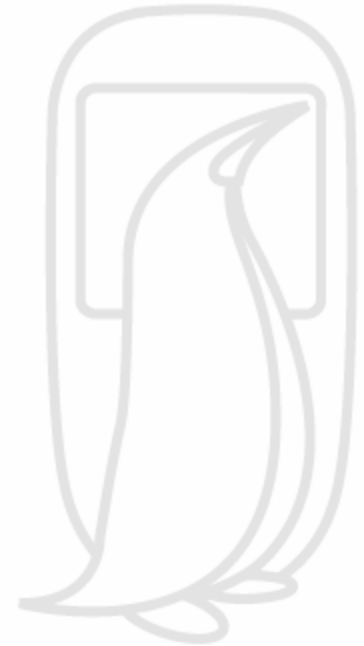


Who's Working to Address This?



The Linux Phone Standards Forum

- (Some) Members: ACCESS, France Telecom, OpenPlug, Telecom Italia SpA, ARM, MIZI, MontaVista, etc.
- Focus: Application-focused higher-level service APIs
 - e.g., telephony, application framework, messaging, etc.
- Current efforts: specification of mid-level APIs for a variety of “enablers”; identification of a standard hardware platform for developers



The Linux Foundation

- (Some) Members: ACCESS, IBM, Intel, Motorola, Nokia, Red Hat, Novell, Wind River, MontaVista, etc.
- Focus: Low-level services and facilities; broad requirements for specific application areas (e.g., “Carrier Grade Linux”, “Data Center Linux”, etc.)
- Current efforts: Propagation of the Linux Standards Base (LSB) as a basis for defining baseline system conformance; publication of guidelines, recommendations, and requirements in a variety of areas

The GNOME Foundation



- (Some) Members: ACCESS, Nokia, Red Hat, Collabora, Novell, etc.
- Focus: Adaptation of “GNOME-related” technologies to function more effectively in mobile settings
- Efforts: Mostly ad-hoc so far; in response to prodding by GNOME members, many improvements to, e.g., Cairo have been made to reduce floating-point dependencies

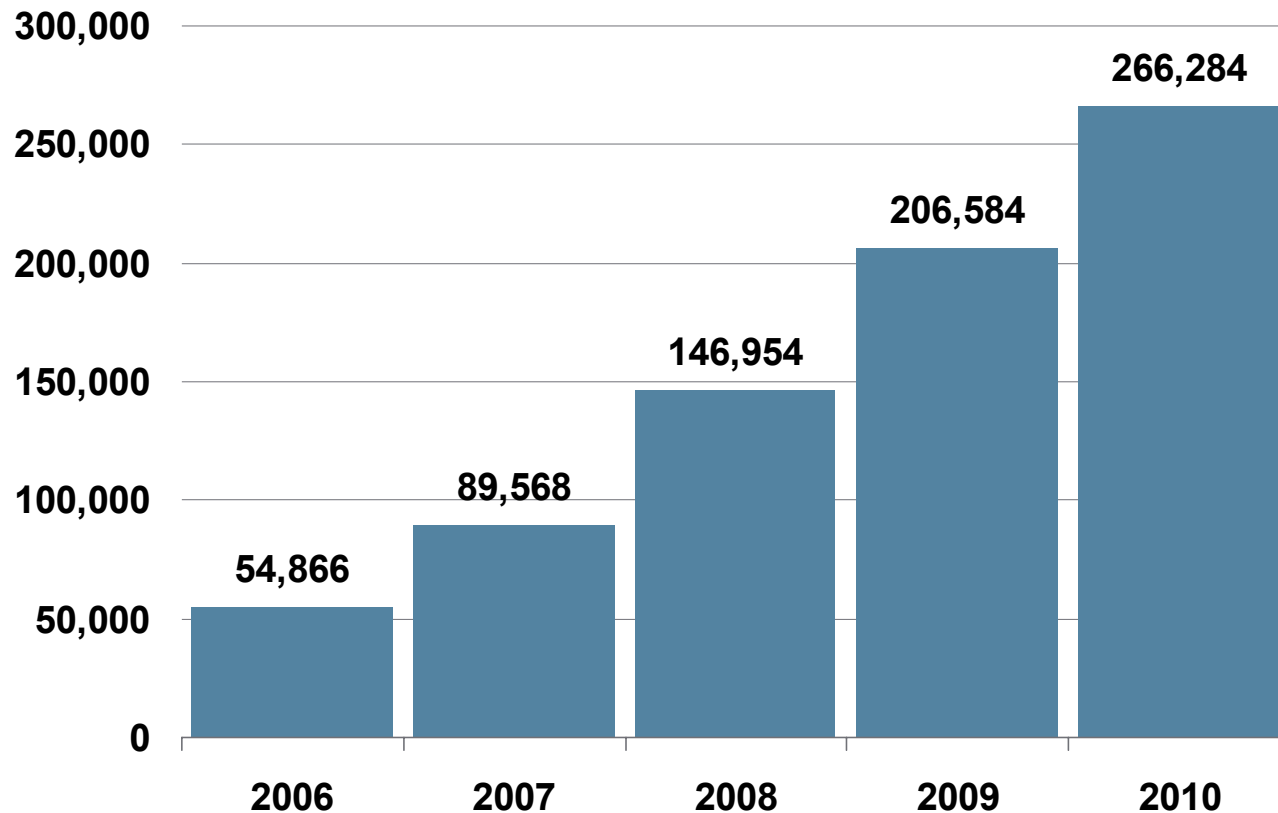


Where's It All Going?



Higher-End Phone Use Is Growing...

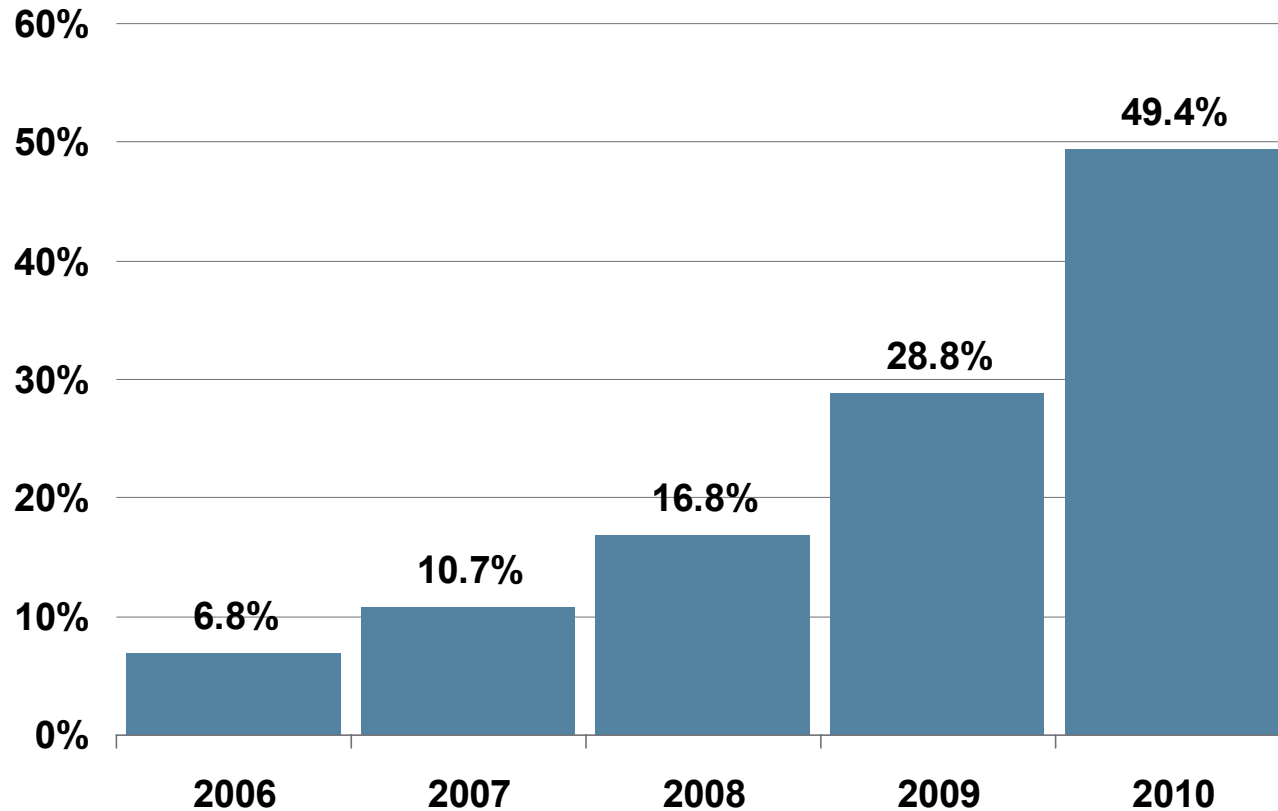
Smart Phone Shipments (000 units)



Source: In-Stat, 2006

Linux Is Gaining Marketshare...

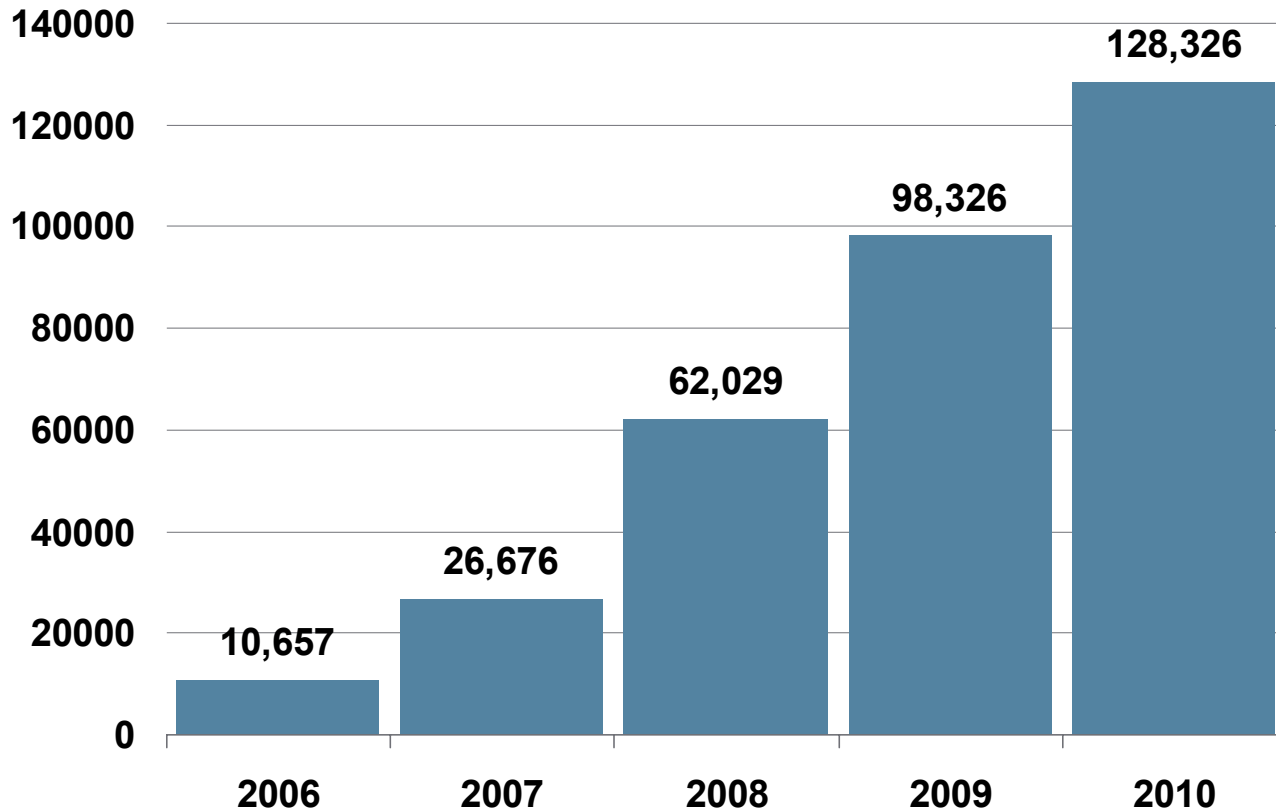
Linux Smart Phone Market Share



Source: Canalys, 2006

So the Prognosis Is Good...

Linux Smart Phone Shipments (000 Units)



Source: In-Stat, 2006

...But Simply Running Linux Isn't Enough

- ...or at least, it doesn't get you as far as you could go
- Most current “Linux phones” are based on previous generation foundations
- The real power of “smart” devices will be best realized through open developer ecosystems
- Third-party development creates more customization, more personalization, more utility...
- However, it creates its own set of requirements
 - Security
 - Commonality
 - Interoperability

A Bright Future!

- Companies like ACCESS and others...
- Working together in efforts like LiPS, the Linux Foundation and the GNOME Foundation...
- Reducing fragmentation in concrete ways...
- Enabling a thriving third-party developer ecosystem...
- Means more open source software in more people's hands, and more opportunities for third-party developers!



The content, including without limitation, descriptions of third party products, features, functions, specifications, release dates and programs and feature products is supplied "AS-IS" AND WITH NO EXPRESS OR IMPLIED WARRANTIES whatsoever made by ACCESS CO., LTD., PalmSource, Inc. or its third party suppliers or licensees. Product features, functions, specifications, programs and the development thereof may be modified, deleted or discontinued and any release date may be modified or cancelled, at any time and without prior notice, in ACCESS', PalmSource's or the applicable third party's sole discretion.

ACCESS and NetFront™ are trademarks or registered trademarks of ACCESS CO., LTD. in Japan and other countries.

ACCESS Linux Platform™, Garnet VM™, Palm OS®, Palm Powered™ and certain other trade names, trademarks and logos are trademarks which may be registered in the United States, France, Germany, Japan, the United Kingdom and other countries and are either owned by PalmSource, Inc. or its affiliates, or are licensed by PalmSource, Inc. from Palm Trademark Holding Company, LLC. These marks may not be used in connection with any product or service that does not belong to PalmSource, Inc. (except as expressly permitted by a license with PalmSource, Inc.), in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits PalmSource, Inc., its licensor, its subsidiaries or affiliates.

The registered trademark LINUX® is owned by Linus Torvalds, owner of the mark in the U.S. and other countries, and licensed exclusively to the Linux Mark Institute

All other trademarks are the property of their respective owners.



Mobilizing Open Source: Creativity, Contribution, and Community

David “Lefty” Schlesinger
Director, Open Source Technologies
ACCESS Co., Ltd.

www.access-company.com

TS-7101