<symbio>

Mobile Computing

Juha-Matti Liukkonen, Nov 17, 2010

Contents

- Mobile Computing revolution
- Structural impact of device evolution
- A look into Mobile Linux

Mobile Computing revolution

Pocketable power

- Advances in technology make computers mobile
  - Low-power processors, displays, wireless network chipsets, …
  - Laptops outsell desktop computers

    iSuppli, Dec 2008
  - High-end smartphones = mobile computers

    Nokia terminology
- Handheld consumer products
  - 2010 high-end: 1 GHz, 512 MB RAM
  - 2011 high-end: 2x1.5 GHz, 1 GB RAM

New use cases

- The power and capacity of modern handheld devices enable new use cases
  - Users are always online, GBs of storage capacity
  - 3G, LTE, WiFi download speeds for fast data access
  - Rich and intuitive user interfaces

Note the emphasis on IP connectifity – voice calls are no longer the main use

- Battery technology remains a limitation
  - Key to success = device (components) must sleep whenever possible

Apple’s game changer

- In 2007, Apple change the mobile world with the iPhone
  - Touch user interface, excellent developer tools, seamless services integration, …
  - Modern operating system, shared with iPod and Mac product lines
  - Caught “industry regulars” with their pants down
  - Nokia, Google, Samsung, et al going for Linux, Microsoft recently introduced WP7

iPad “killed the netbook”

- In 2010, Apple introduced another mobile game changer
  - iPad = basically, a scaled-up iPhone with a 10″ touch screen
  - Bigger case = can fit bigger battery, for ~10 hours of intensive use
  - 7,5 million sold as of Sep 30, 2010
  - Phenomenal netbook sales growth fizzled

Apple, 2010

NPD, Morgan Stanley Research, 2010

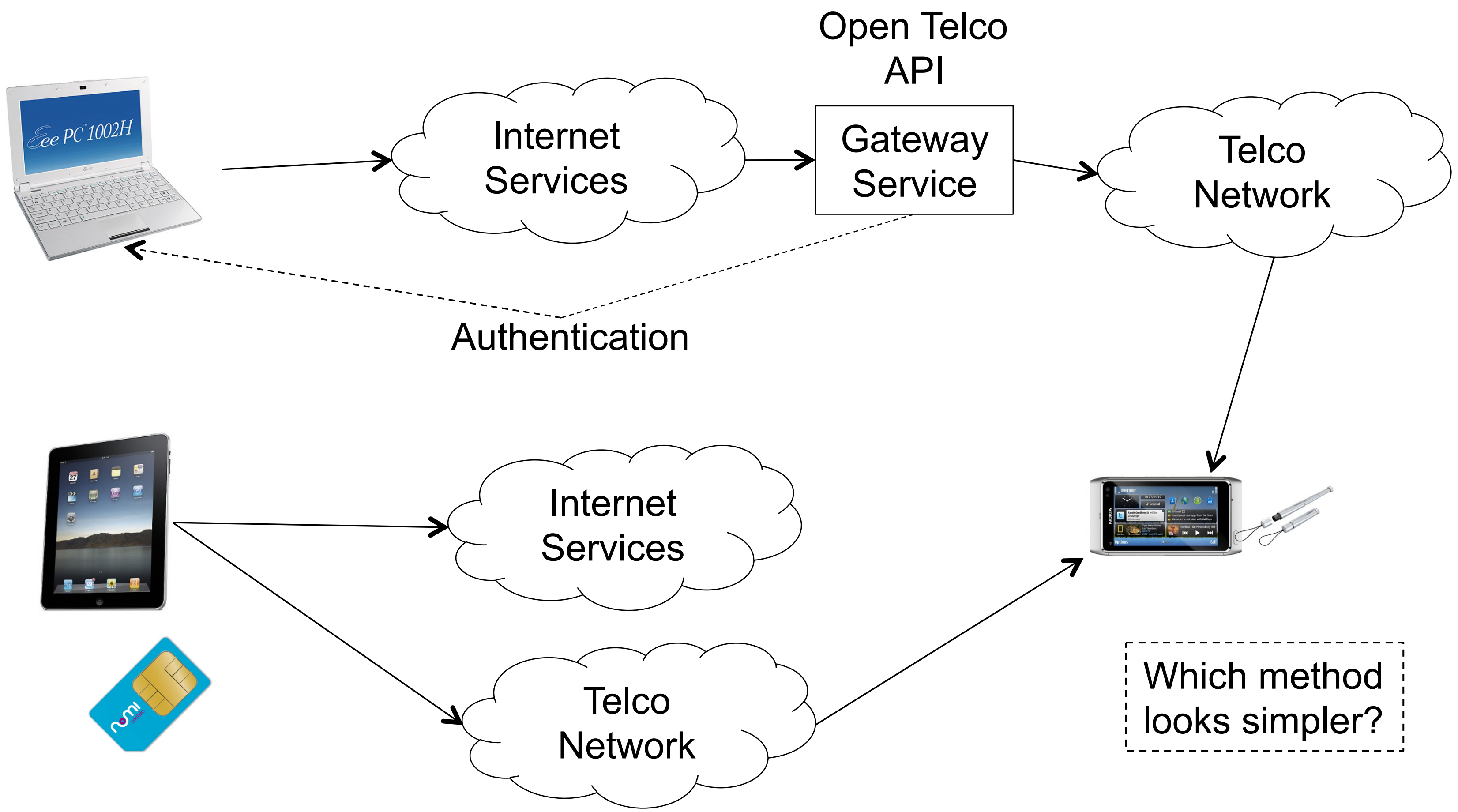
App ecosystems

- Powerful mobile computers can run variety of software
  - Dynamic availability of applications to provide added value over device lifetime
  - Devices become multi-purpose and adaptable
  - Voice call functionality a secondary feature
- New software design challenges
  - New user interaction models
  - Preservation of battery

Nokia was ahead of its time with the Communicator concept back in 1996.

Structural impact of device evolution

Telecom network access



Which method looks simpler?

Proliferation of IP-connected devices

- Users love their adaptable devices
  - New apps, new content keep users entertained
  - Always online supports social networking
- Businesses love their adaptable devices
  - Devices can be adapted and repurposed via software
  - Portable data entry and viewing terminals with live GPS & IP network access (NFC is coming)
- New Cloud based business models

Structural impact: networks

- Number of always connected IP devices explodes
  - Wireless network role as an IP bearer increases, traditional telco function decreases
  - Traffic volume increases, per-packet/per-MB cost must come down
  - Latency requirements increase with device interactivity
  - Lessons learned by AT&T at iPhone launch

We should also be concerned about the IPv4 address space.

Structural impact: the Cloud

- Always online = when network is available
  - In effect, "always online" = traditional intermittent net presence, with shorter disconnected intervals
- Heavy processing must be offloaded
  - Device components must still sleep when possible, to achieve decent use times
- Synchronizing multiple devices
  - Keeping master copy of data in the cloud

Mobile devices are also easier to lose or break than traditional computers. It's good to have your key data hosted elsewhere.

Structural impact: open source

- Open source a strong force in the new device environment
  - Apple iOS: core OS is open source
  - Google Android: open source
  - MeeGo: open source
  - Symbian: open source (on paper)
  - Microsoft WP7: proprietary
- Platforms become commodities

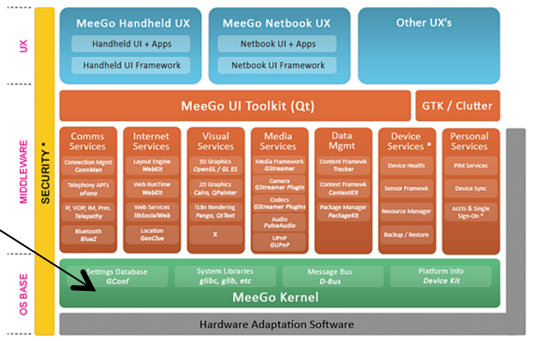
Opportunities & challenges

- New business opportunities
  - Refactoring the wireless network technology
  - Re-thinking distributed application architecture
  - Finding new uses for mobile devices
- Engineering challenges
  - New wireless network technology is needed
  - Maturing the cloud services
  - Security considerations

A look into Mobile Linux

Linux Distribution

- Linux is the operating system kernel
  - Deals with hardware abstraction



- A *distribution* is a managed collection of software, including the kernel
  - Device drivers, middleware, user applications
  - Comes with distributor-defined default settings and applications
  - Often optimized for specific use(s)
  - E.g. Ubuntu, Red Hat Enterprise Linux, Maemo

Mobile Linux distributions

- Maemo
  - Nokia’s Linux distribution for Internet tablets and high-end smartphones
  - Powers the N770, N800, N810, N900
- Android
  - Google’s Linux distribution for Internet tablets and smartphones
  - Powers many HTC devices, Nexus One, etc.

Mobile Linux distributions

- OpenEmbedded
  - Open source project
  - Best suited for custom adaptations to very small devices
- MeeGo
  - New kid on the block
  - Combines Intel’s Moblin netbook Linux and Nokia’s Maemo Linux
  - First MeeGo devices out in fall 2010

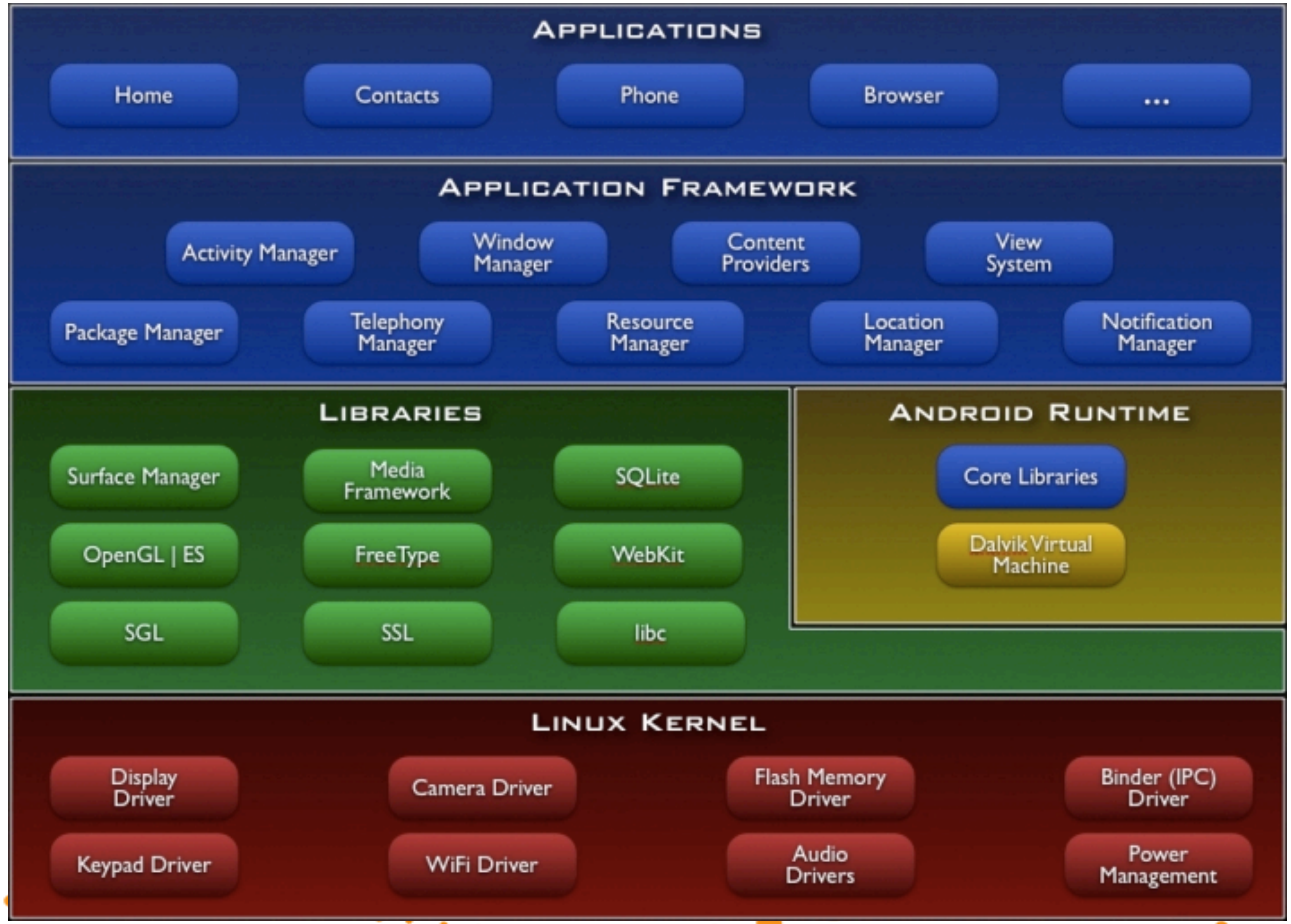
Android details

- Uses custom Linux kernel
  - Google maintains a set of Android patches
- Applications developed using Java
  - Google’s custom Dalvik Java VM
- 5 versions in active use
  - 1.5, 1.6, 2.0, 2.1 and now 2.2
  - Used in various smartphones by HTC, Google, Motorola, LG, etc.

There is also a Native Development Kit (NDK) for building native Linux applications.

The devices have a bit different resolutions and feature sets.

Android architecture



Android points of interest

- Custom C library
  - C library = system calls (interface to kernel), POSIX & ANSI standard library routines
  - Linux standard is glibc, which is a bit bloated
  - Android has a stripped down libc
  - Compatibility issues for generic Linux code
- Custom application installation
  - Apps bundled into .apk "Android packages"

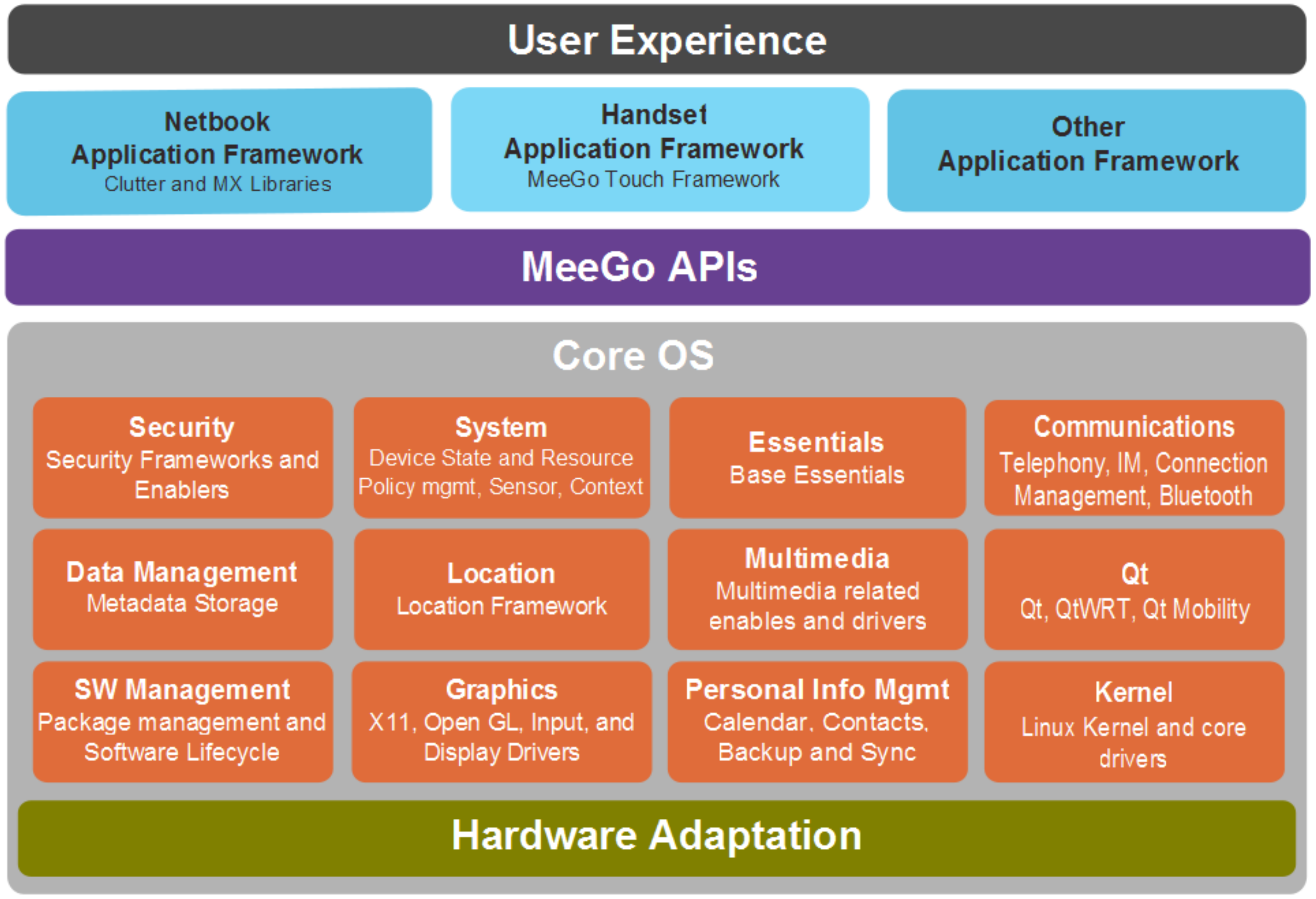
Android points of interest

- Programming model
  - Activity
    - Impements an application view
  - Service
    - Background program with no UI
  - Broadcast receiver
    - Listens for e.g. battery notifications
  - Content provider
    - Shares data from an app

MeeGo details

- Uses standard Linux kernel
- Applications developed using Qt, C/C++
- Replaces both Moblin from Intel, and Maemo from Nokia
- 1.1 release Oct 2010
  - In practice, Beta quality right now
  - Product quality release Apr 2011

MeeGo architecture


User Experience
Netbook Application Framework
Clutter and MX Libraries
Handset Application Framework
MeeGo Touch Framework
Other Application Framework
MeeGo APIs
Core OS
Security
Security Frameworks and Enablers
System
Device State and Resource Policy mgmt, Sensor, Context
Essentials
Base Essentials
Communications
Telephony, IM, Connection Management, Bluetooth
Data Management
Metadata Storage
Location
Location Framework
Multimedia
Multimedia related enables and drivers
Qt
Qt, QtWRT, Qt Mobility
SW Management
Package management and Software Lifecycle
Graphics
X11, Open GL, Input, and Display Drivers
Personal Info Mgmt
Calendar, Contacts, Backup and Sync
Kernel
Linux Kernel and core drivers
Hardware Adaptation

MeeGo points of interest

- Mostly regular Linux
  - Glibc, gstreamer, ALSA, etc.
  - Not based on any existing distribution, but uses rpm & zypper for package management
- User interface modules separated from base platform
  - Different user interaction models for different use scenarios
- Qt is the primary application interface

MeeGo points of interest

- UX modules
    - Handheld: touchscreen (meegotouch toolkit on top of Qt)
    - Netbook: keyboard/mouse
    - Connected TV: remote control
    - In-Vehicle Information: touchscreen, joystick
- Reference applications for each UX model
    - System vendors can customize as needed

MeeGo points of interest

- Stable API
  - Any MeeGo application can run on any MeeGo certified system (given the same CPU arch.)
  - Main part of API is Qt (Core, Gui, Mobility, …)
  - Also: gstreamer, sqlite, ALSA, D-BUS interfaces to various frameworks, etc.
- Goal is to encourage an App Store ecosystem rivaling Apple

<symbio>

Why MeeGo is interesting to us

- Only credible challenger to Google-controlled Android
- Backed by Nokia -> direct impact to Finnish software development scene
- Innovative architectural solutions
- Aims to become the "industry standard" Linux for modern embedded systems

You can participate in building MeeGo: go to meego.com and become active!

SERIOUS ABOUT SOFTWARE