

# Putting Linux in the mobile

Embedded Systems Scandinavia

Stefan Persson  
Ericsson Mobile Platforms



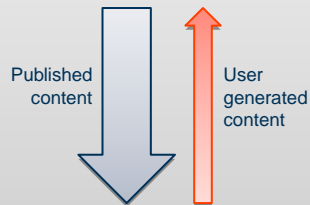
# Outline

- Why Linux in mobile platforms/phones?
- Design challenges putting Linux in mobile phones

# Why Linux?

## Web 1.0

"the mostly read-only web"  
250 000 sites

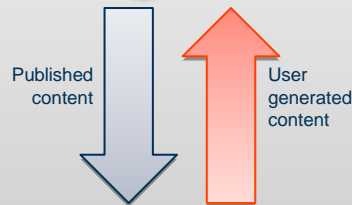


45 million global users

1996

## Web 2.0

"the widely read-write Web"  
80 000 000 sites

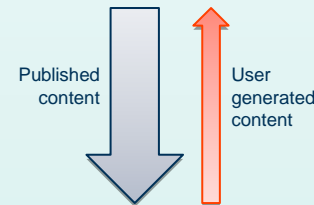


1 billion+ global users

2006

## Closed OS

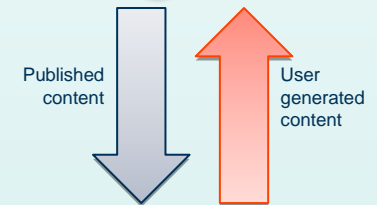
"one company & it's  
resources"



2006

## Linux

"worlds' all Linux developer"



2010

Power transition towards users/developers



One way channels of communication make for great monologues but.....

....

**POOR CONVERSATIONS**



**Empower  
people and  
give them  
a voice**

# COMMUNITY



# Why Linux is a good choice?

From a technical perspective

- Good execution model with real processes
  - gives reliable and memory efficient systems
- Good Tools and Development environments
  - Get quicker and more stable results
- Open source and communities:
  - don't spend time and money reinventing the wheel
  - 3<sup>rd</sup> part know-how, drivers, education, support, applications, etc
- Scalability: Linux runs on anything
  - not only for smart and feature phones



# Outline

- Why Linux in mobile platforms/phones?
- Design challenges putting Linux in mobile phones

# Design challenges at EMP

With Linux on mobile platforms/phones

- Migrating from “old” OS to Linux while maintaining backwards compatibility AND still take advantage of goodies in Linux
  - Keep low power consumption
  - Keep RAM usage low
- Licensing aspects: GPL, LGPL, etc. vs. IPR and 3<sup>rd</sup> party components
- However biggest Linux software challenging is in the application framework area!



# Examples of technical challenges encountered with Linux



# Access-Application

## Problem

### Dependencies

- Tight coupling between both sides
- Functions spread on both sides (for workload balance)

## Solution

### Re-design

- Create a clear separation and interface between each side
- "Move" features around, to where they have a logical meaning

# HW Drivers

## Problem

### Inexistence or performance

- Some drivers missing:  
MemoryStick, Battery charging & supervision, Analog joystick, Thumb wheel, etc
- Drivers were not optimized for EMP's hardware

## Solution

### Re-design

- Create missing drivers
- Update and optimize under-performing drivers

# Sound

## Problem

### Quality and performance

- Direct porting gave low quality performances and audio quality

## Solution

### ALSA

- Embrace the Advanced Linux Sound Architecture

BUT ALSA is not yet very popular/adopted in the Open Source community

# Security

## Problem

### Models and philosophies

- Going from a closed to an open platform
- Choice of different security models and solutions in Linux

## Solution

### Select the best one

- Review existing models
- Mix 'n' match according to needs and/or requirements

# Testing

## Problem

### OS and test-case dependency

- New paradigms
- New behaviors
- New tools
- New systems

## Solution

### Rewrite and update

- Adopt new test tools
- Update use-cases
- Update test suite and reference applications

# IPC – Inter Process Communication

## Problem

### Different architectures and models

- Coexistence of different communicating OS's
- Different communication protocol on each OS
- Even if they can coexist, there is a performance impact: direct port of Linux was 12 times slower at the process communication level

## Solution

### Improve and adapt, optimize

- First round of improvement brought communication speed down to 3 times slower
- Further improvements are expected to result in same (or better) than the original mono-OS solution

# Other areas to consider

Address space  
(single vs. multi)

Flash Memory  
Partitioning

Power Management

Memory  
Management

Startup / Boot  
loader

File system

Loaders

**ERICSSON**



**TAKING YOU FORWARD**