NFC on Linux IVI systems

Samuel Ortiz
Intel Open Source Technology Center

September 20th, 2012

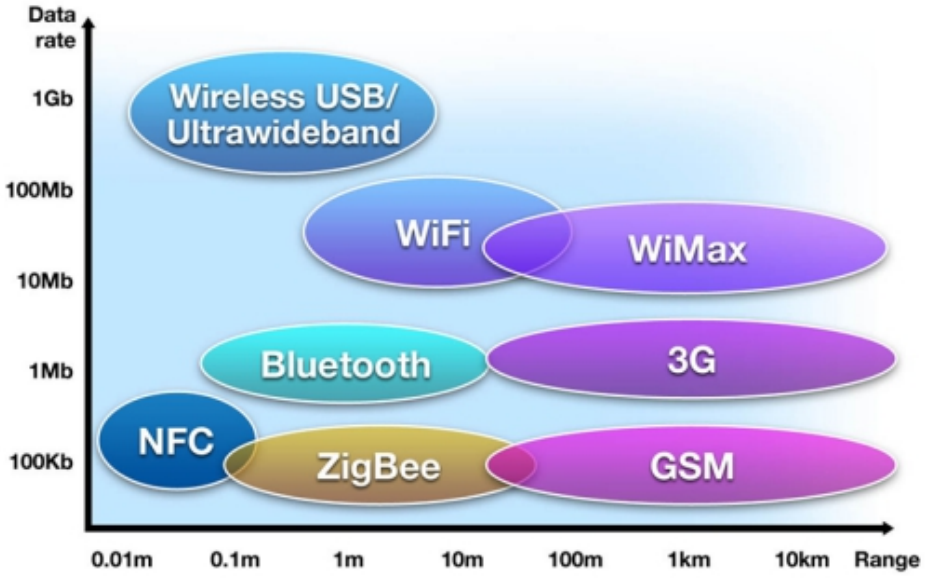
Agenda

- NFC basics
- NFC open source stacks
- The Linux NFC stack
- NFC Bluetooth Handover and IVI

NFC basics

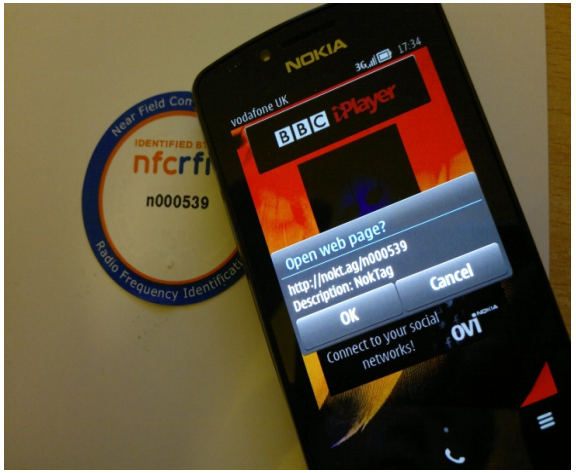
Near Field Communication

- A short range (< 5cm) wireless technology.
- Low throughput (< 500 kbps).
- Low cost.
- Not Bluetooth, not RFID.
- Partly standardized by the NFC Forum.
- “Tap-to-share” NDEFs.
- NFC tags and NFC devices.



Three NFC modes

- Reader
  - One device reads a tag.
- Peer to peer
  - Two devices talk to each others
- Card emulation
  - One device pretends to be a tag

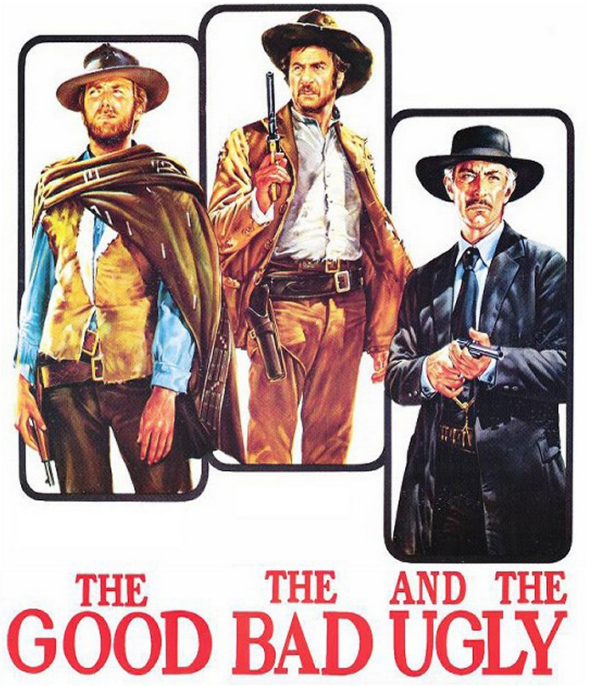


Use cases

- Very wide...
- Data exchange.
  - Playlists, URLs, business cards...
- Connection Handover.
  - Simplified Bluetooth pairing
- Payments, loyalty cards.
- Ticketing.
- Security, access control.
  - Key-less rental cars

NFC Open Source Stacks

- The Android bounty.
- Android as the single supported platform.
- No kernel support for NFC.
- No standard Linux distribution support.



Two stacks, same issues

- Two Android stacks.
  - libnfc-nxp, opennfc.
- 100% userspace, ad hoc kernel interface.
- Exclusive HW support.
  - NXP pn544, INSIDE microread: HCI only.
- No community, no source code repositories.
- Exclusive support, no visibility.
  - Google, INSIDE.

Other stacks

- nfcpy
  - Nice implementation, 100% python.
  - Sony sponsored.
  - No HCI or NCI support.
- libnfc
  - Academic project, LGPL licensed.
  - Only USB and UART devices supported.
  - Missing features.
  - SVN repository, community.

The Linux NFC stack

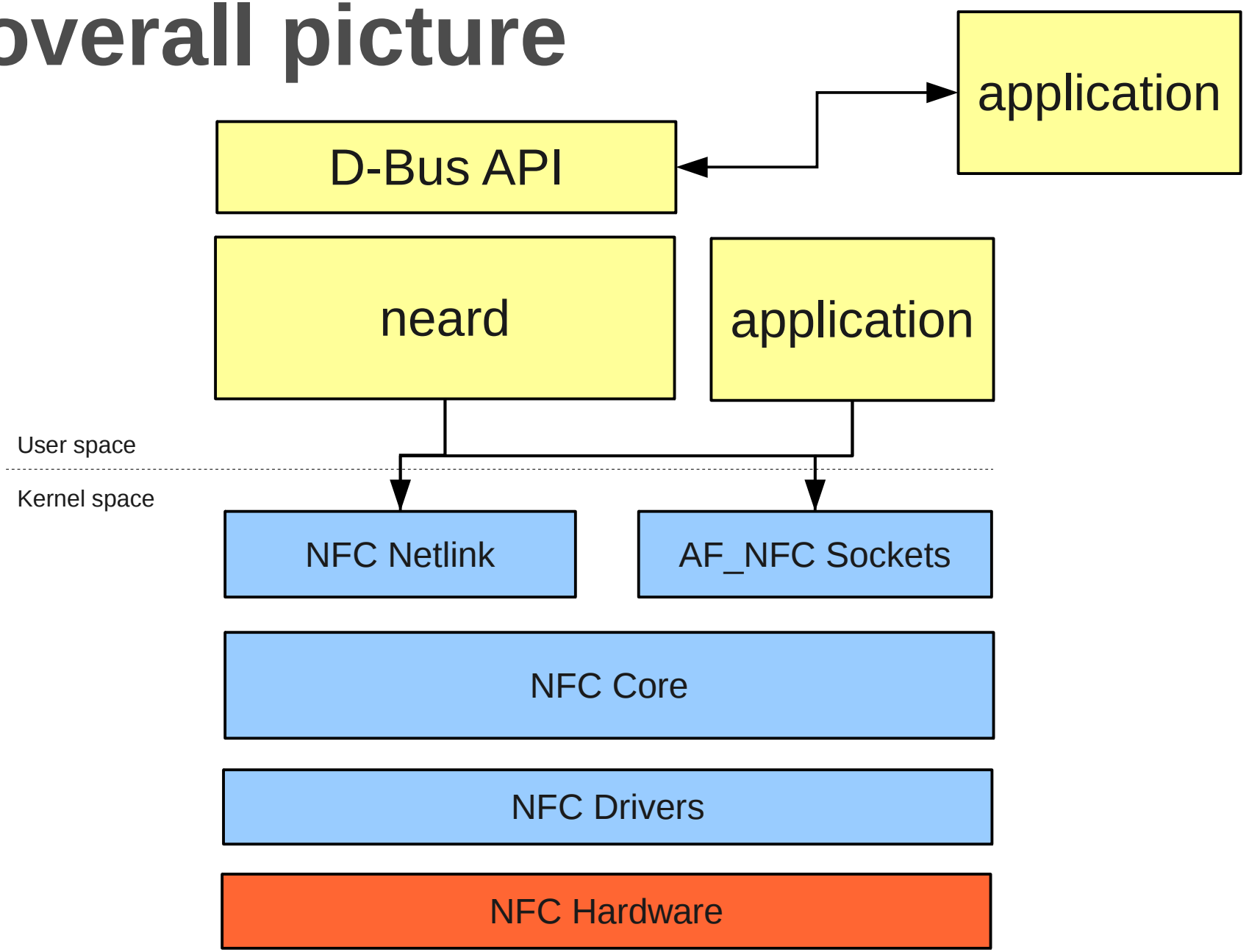
Yet another stack ?

- HW independence.
- NFC for non Android platforms.
- POSIX NFC APIs.
- Kernel/User space split.
- Consistent behavior and APIs.
- Open development process.

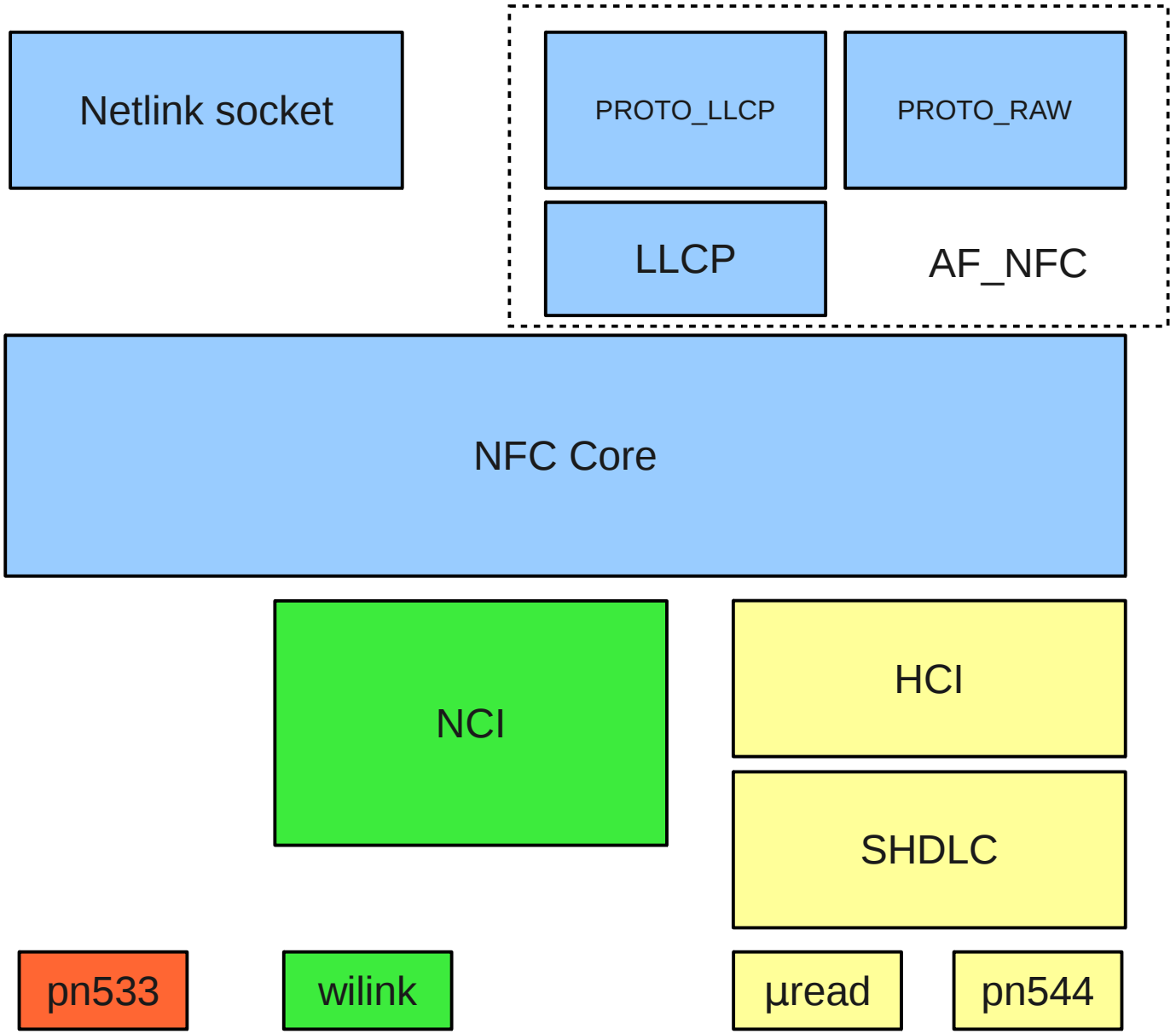
The Linux NFC stack

- The official NFC Linux kernel stack.
- Maintained by Intel.
- Hosted on git.kernel.org.
- GPLv2 licensed.
- 1.5 year old.
- Split between kernel and user spaces.
- Open development.

The overall picture

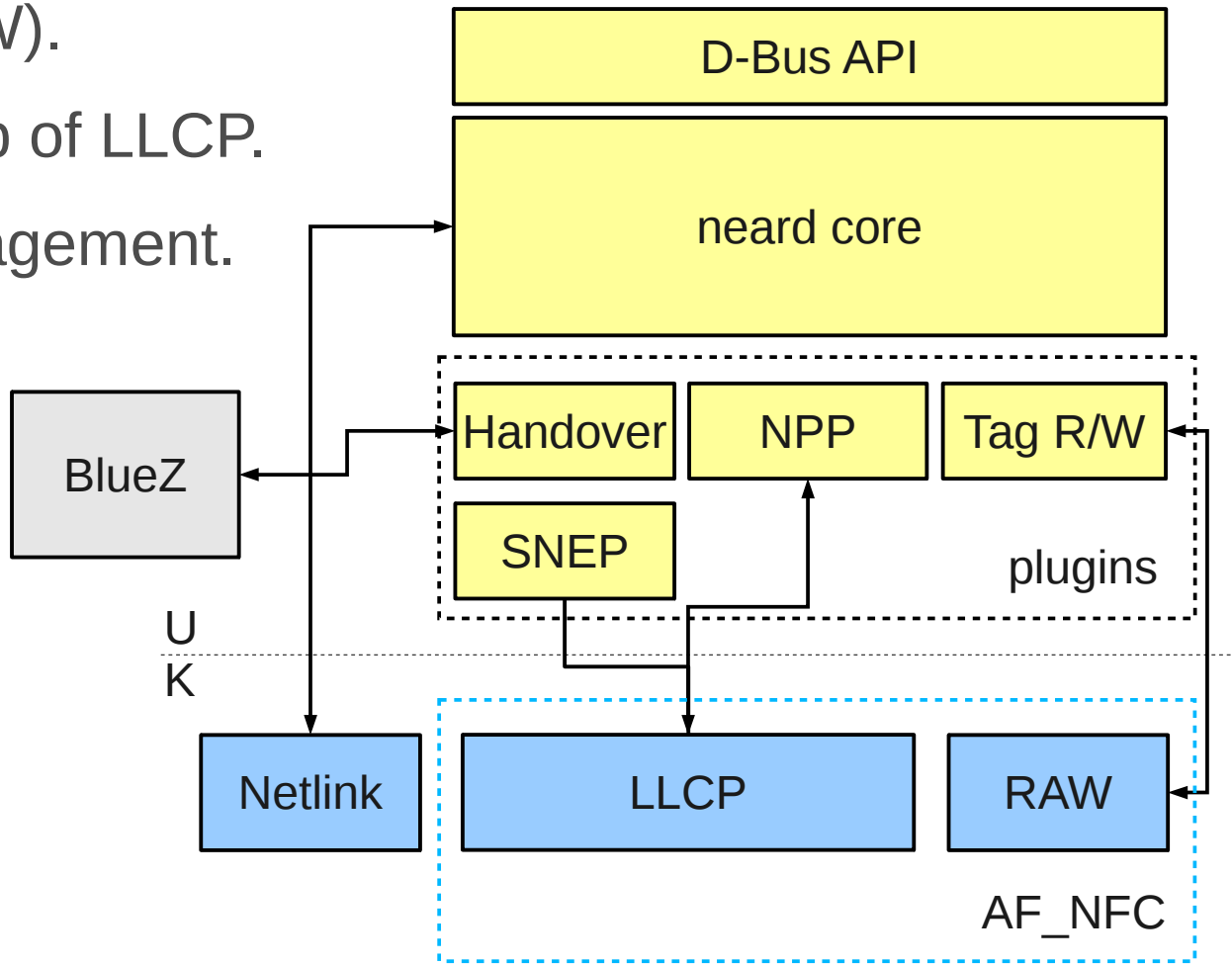


Kernel Architecture



The NFC daemon

- Tag specific handling (R/W).
- Transport protocols on top of LLCP.
- Adapter and targets management.
- NDEF parsing.
- Handover.
- D-Bus APIs.
- Plugin based.



Hardware and Features Support

	Supported Hardware
Linux	NXP pn544, NXP pn53x[1], TI nfcwilink
Android	NXP pn544
Inside Secure	Inside Secure microread
libnfc	NXP pn53x
nfcpy	NXP pn53x

	Interfaces	Tag R/W	LLCP	Handover	Card Emulation
Linux	HCI, NCI, USB	Yes	SNEP, NPP	Bluetooth	No
Android	HCI	Yes	SNEP, NPP	No[2]	Yes
Inside Secure	HCI	Yes	SNEP	Bluetooth, WiFi	Yes
libnfc	USB, UART	Yes	No	No	Yes
nfcpy	USB	Yes	SNEP	Bluetooth	No

[1] PN532 not supported yet

[2] Bluetooth handover supported with Jelly Bean

Plans

- Short term
  - Improve MIFARE support.
  - Secure Element and card emulation netlink API.
  - Service Discovery Protocol.
  - Wi-Fi Handover.
- Long term
  - Inside Secure microread support.
  - NFC monitor.
  - OBEX and IP over NFC.

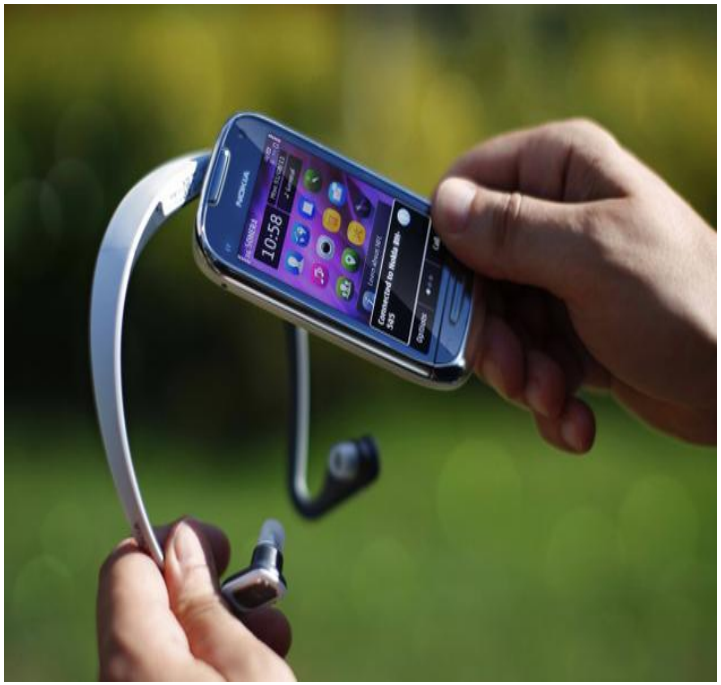
NFC Connection Handover

NFC Connection Handover

- Switching to an alternative carrier via NFC.
  - NFC is the credentials carrier.
- Static or Dynamic.
- Bluetooth or Wi-Fi.
- Requester or selector.

Bluetooh Handover and IVI

- Bluetooth Secure Simple Pairing.
- Simplified User Experience.
  1. Press a button
  2. Touch your dashboard
- No user input, no extra menus.
- Fully supported by neard 0.6.
  - “Press a button” ↔ Start a Handover request
  - BlueZ dependency



Questions ?

- NFC daemon

  http://git.kernel.org/?p=network/nfc/neard.git;a=summary

- NFC kernel

  http://git.kernel.org/pub/scm/linux/kernel/git/sameo/nfc-3.0.git

- Web site
  - https://www.01.org/linux-nfc
- Mailing list

  https://lists.01.org/mailman/listinfo/linux-nfc

- sameo@linux.intel.com