NFC in Android

Public

MobileKnowledge
October 2015

Index

**NXP** Training Mobile Knowledge
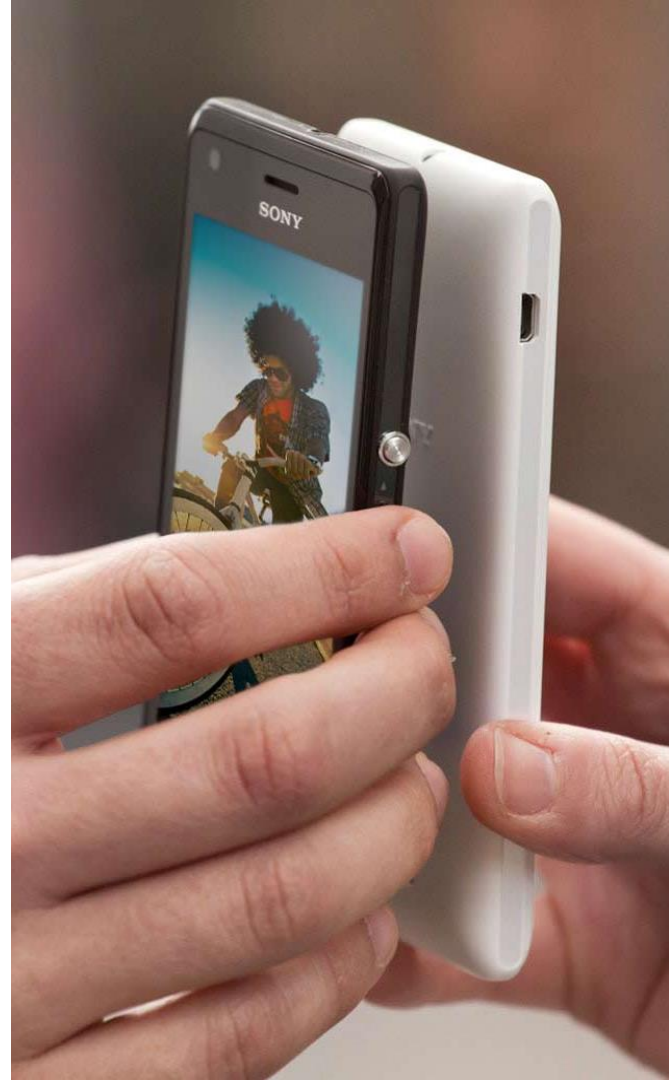
What is a Mobile OS?

► A mobile Operating System, is an operating system specifically designed to run on mobile devices such as mobile phones, smartphones, PDAs, tablet, computers and so on.

► The Mobile OS is the software platform on top of which other programs, called applications, can run on mobile devices. It provides hardware abstraction to these applications.

Introduction to Android and NFC
Android in mobile devices

► Android is a Mobile Platform and its software stack includes:
  - The Operating System based on Linux
  - The middleware that allows apps to talk to a network or to one another
  - Key applications that the phones will run

► Android is a very popular mobile platform as it is multi-platform, open-source and free.
  - +1.5M apps on the PlayStore
  - +1.5B downloads from the PlayStore every month
  - +1M devices activated worldwide every day
  - +450K publishers

► Android is the Mobile Platform used by many phone manufacturers
  - Samsung, Sony, LG, HTC, Motorola, Huawei, Lenovo, Oppo, Xiaomi, etc.

Smartphone OS market share 2014

- Android
- iOS
- Windows Phone
- Blackberry
- Others
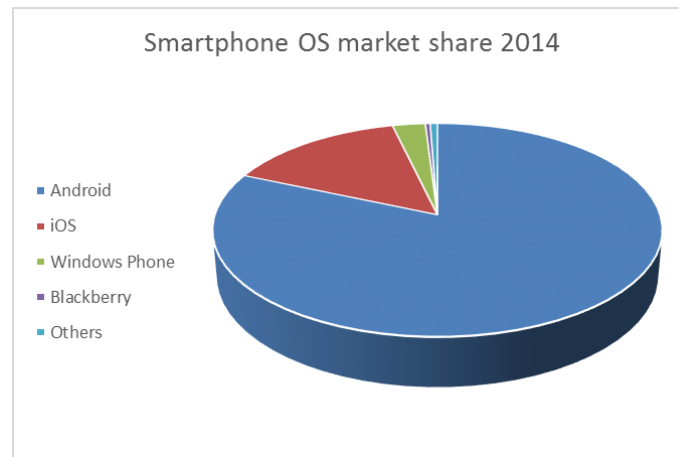
Source: International Data Corportation

Android Versions
Brief history …

► 2003: Android Inc. was founded in California (USA)

► 2005: Google acquired startup Android Inc. to start Android platform

► 2007: Open Handset Alliance announced

► 2008: First Android phone released (HTC Magic) together with SDK 1.0 and Android Open Source Project (AOSP)

Version	Code Name	Release Data	API
6.0	Marshmallow	October, 2015	23
5.1	Lollipop R1	March, 2015	22
5.0	Lollipop	November, 2014	21
4.4.x	KitKat	October, 2013	19
4.1.x	Jelly Bean	July, 2012	16
4.0.x	ICS	December, 2011	15
3.x	Honeycomb	Feb, 2011	11
2.3.3–.7	Gingerbread	February, 2011	10
2.3.0–.2	Gingerbread	December, 2010	9
1.0	Apple Pie	September, 2008	1

HCE

Android Beam

NFC Support
R/W & P2P

NFC API
Extensions
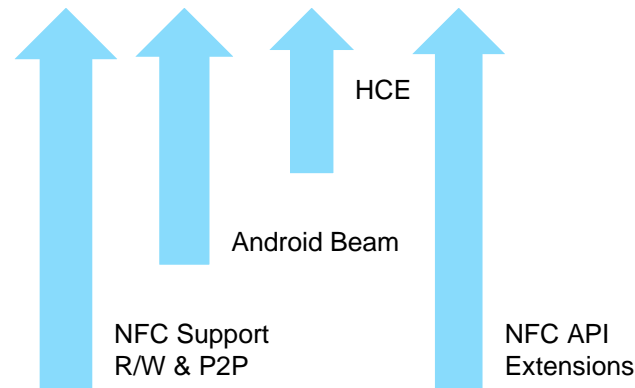
Introduction to Android and NFC
## Android everywhere

NFC connected devices
Market update - some key figures

▶ 1.2 billion smartphones shipped in 2014

▶ Smartphone's share expected to continue growing from 67% in 2014 to > 80% or even higher in coming years

▶ 850 million NFC handsets shipped between 2012 and 2014

▶ 3 in 4 mobile phones to come with NFC by 2018

▶ > 5 billion NFC handsets will be shipped between 2013 and 2018

▶ NFC-enabled CE devices and tags growing exponentially, IoT wave coming.

* Updated list of NFC phones and tablets available in the market:
http://www.nfcworld.com/nfc-phones-list/

11.0B+ NFC-enabled devices shipping 2013-2018

Shipments (Mpcs)

Legend: NFC Handsets — CE Devices — Tags
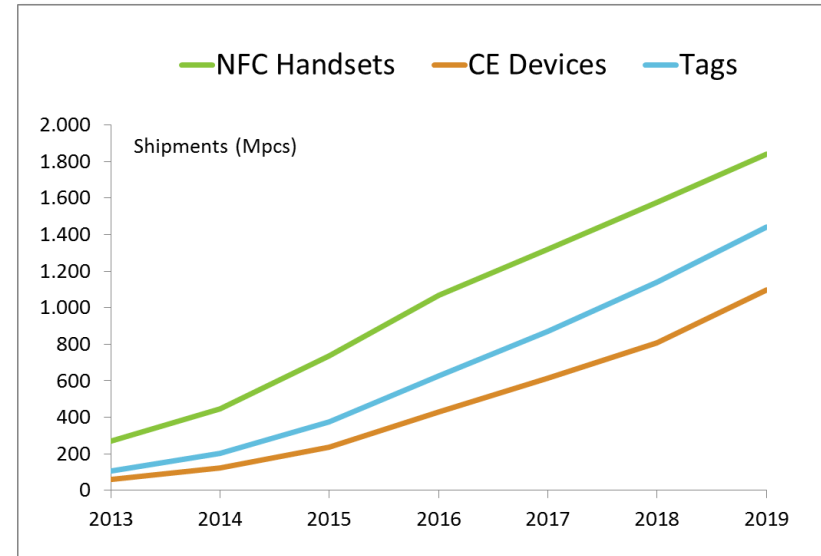
Sources: ABI Research, Sep'14

Android programing

Development environment
Android Studio

► Android Studio is the official IDE by Google
  ▪ http://developer.android.com/sdk/index.html

► Android Studio is an all-in-one installation
  ▪ Android Studio IDE
  ▪ Android SDK tool
  ▪ Latest Android API Platform – Android 6.0 (Marshmallow)
  ▪ Latest Android API emulator system image - Android 6.0

► Android Studio is multi-platform
  ▪ Windows, MAC, Linux

► Advanced GUI preview panel
  ▪ See what your app looks like in different devices

Last Updated: 06/10/2015

Training Mobile Knowledge
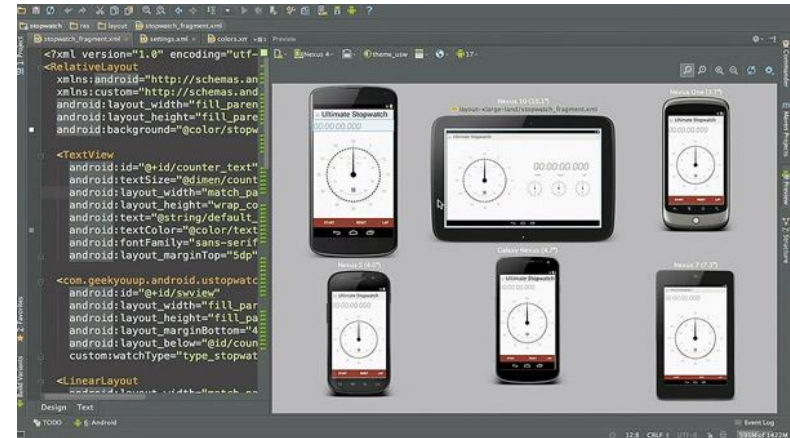
Android programing

► Android developers website
   ▪ http://developer.android.com/index.html

► Applications written in Java programing language

► Android security constraints:
   ▪ Each application runs in its own process.
   ▪ Each application is assigned a unique Linux user ID; by default, files of that application are only visible to that application.

► Android APK: Application PacKage file
   ▪ File format used to distribute and install applications

NFC in Android

NFC Technology
Read/Write mode

Card Emulation

Peer to Peer

Read/Write

Reads / Writes data from any
tag or contactless card

NFC in Android

► Card Emulation mode "supported"
  ▪ HCE supported since Android KitKat

► Read/Write mode supported
  ▪ Passive NFC Forum Tags
    ❖ Tag Type 1: Topaz
    ❖ Tag Type 2: MIFARE Ultralight & NTAG (simple dedicated API)
    ❖ Tag Type 3: FeliCa
    ❖ Tag Type 4: MIFARE DESFire
  ▪ Proprietary NXP NFC Tags
    ❖ MIFARE Classic (simple dedicated API)
    ❖ ICODE

► Peer to Peer mode supported

► Android NFC developer's guide
  ▪ http://developer.android.com/guide/topics/connectivity/nfc/index.html

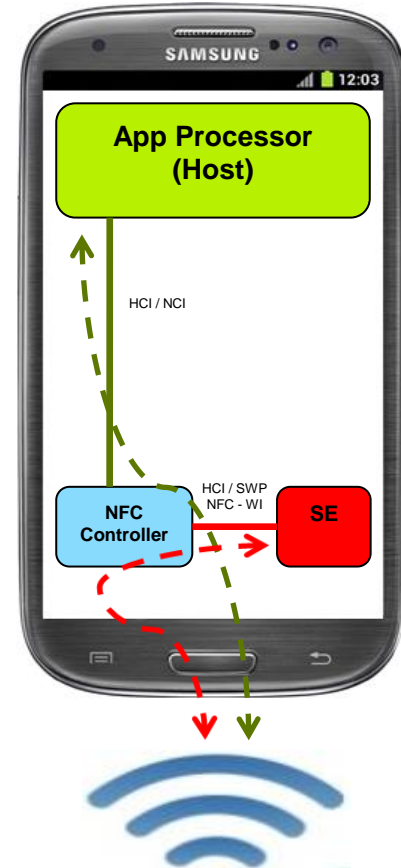# Card Emulation Mode

Card Emulation Mode
Configurations

► Secure Element
- Proven high-secure and tamper-resistant microcontroller in the device
  - Same family of product used for mass market solutions: payment cards, e-Passports…
- A specific IC to handle and store sensitive data
  - Non-Volatile Memory, Security CPU, Crypto co-processors
- Protected against attacks and tampering by cryptographic keys
  - Only authorized entities can access the SE
- Secure IC validated by third party certification, i.e. Common Criteria

► Host Processor
- Main processor of the device in which the OS and applications reside
- Sensitive information is stored in the Host Processor or in the Cloud
  - More memory available via host versus secure element
- Application/service providers and end users get (more) control
- "more-simple-but-less-secure" card emulation

► The NFC Controller forwards each APDU according to its Routing Table

App Processor
(Host)

HCI / NCI

NFC
Controller

HCI / SWP
NFC - WI

SE

Card Emulation Mode
Emulating NFC Applications

► How to develop my Android NFC application
  ▪ Indicate where the application will be emulated
    ❖ To be indicated in the androidmanifest.xml file (ON or OFF Host APDU Service)
  ▪ In HCE exchange APDUs with the NFC Reader
    ❖ Based on Android Services (NFC application available even if not in the foreground)
  ▪ In SECE communicates with the SE using APDUs
    ❖ SEEK for android based on Open Mobile API
    ❖ NFC Extras library
      – Communication restricted to applications signed with a key in /etc/nfcee_access.xml
  ▪ Application logic will be NFC application dependent

► https://developer.android.com/guide/topics/connectivity/nfc/hce.html

► http://nelenkov.blogspot.com.es/2012/08/accessing-embedded-secure-element-in.html

► http://seek-for-android.github.io/

NFC Frontend solutions

► Robust, flexible options
► Supported by NFC Reader Library
► Power-saving passive mode
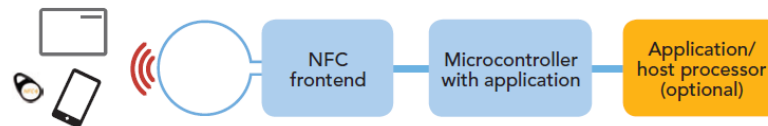
| High-performance NFC frontends	
CLRC663	High-performance multi-protocol NFC ready frontend
MFRC631	High-performance ISO/IEC 14443 A/B frontend
MFRC630	High-performance MIFARE frontend
SLRC610	High-performance ISO/IEC 15693 frontend
Standard-performance NFC frontends	
PN512	Full NFC Forum-compliant frontend
MFRC523	Standard 3V ISO/IEC 14443 A/B frontend
MFRC522	Standard 3V MIFARE frontend
High-performance frontends delivering full NFC Forum compliance	
PN5180	High-performance multi-protocol full NFC Forum-compliant frontend (available 2015)

NFC controller solutions
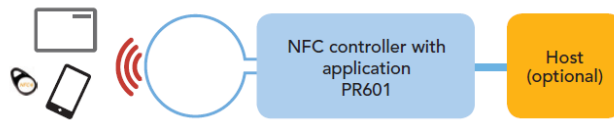
PN7120

Integrated firmware

- Linux, Windows, Android environments
- Pre-loaded with NFC Forum's NCI interface, to support full OS
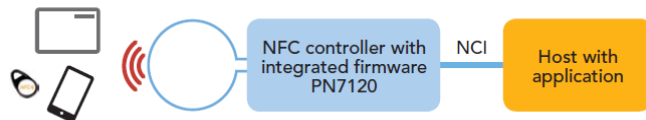
PR601

Customizable firmware

- For use with own-developed software
- Integrated LPC1227 microcontroller
- Supported by NFC Reader Library

Read/Write Mode

Read/Write Mode
## Operating on NFC Tags

► How to develop my Android NFC application

- Indicate NFC support and optionally Tag Technologies to capture
  - ❖ Defined in the project AndroidManifest.xml file
- Capture and filter tags tapped by the user
  - ❖ Based on Android Intents
- Get tag supported technologies and obtain the specific tag object
  - ❖ Supported card technologies can be obtained from the Intent
  - ❖ Android provides the classes and methods to manage all technologies
- Connect to the tag
- Exchange tag specific commands
  - ❖ Exchange read, write, ... commands according to your application logic
- Close the connection with the tag

► Sample application by Android developer's guide

- http://developer.android.com/guide/topics/connectivity/nfc/advanced-nfc.html
  - ❖ Read and write MIFARE Ultralight tag

Connected NFC Tag solutions

▶ Passive, NFC Forum type 2 tag

▶ Field-detection function

▶ Optional I²C interface

▶ NTAG F for battery-powered systems

▶ NTAG I²C for full bi-directional communication with host microcontroller

▶ Innovative energy-harvesting feature for low-power systems
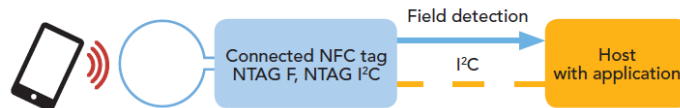
Connected NFC Tag solutions	
NTAG 216F	Passive NFC tag with field-detection output signal, 888 bytes
NTAG 213F	Passive NFC tag with field-detection output signal, 144 bytes
NTAG I²C 2k	Passive NFC tag with I²C interface, 1904 bytes
NTAG I²C 1k	Passive NFC tag with I²C interface, 888 bytes

NTAG I²C in short

Fully NFC Forum Type 2 Tag Compliant

Dual (NFC + I2C) Interface

Up to 1,9KByte of non-Volatile User Memory → **Version 1K** 888 bytes of User Memory | **Version 2K** 1904 bytes of User Memory

PASS-THROUGH mode for fast data transfer → 64 bytes SRAM Buffer for RF→ I2C and I2C→RF data transfer

Field-detection Feature → Open drain implementation | Configurable | It can signal to the uC
-RF has written new data in the SRAM buffer
-RF has read the data in the SRAM buffer

Energy Harvesting → To power external devices (e.g: MCU)
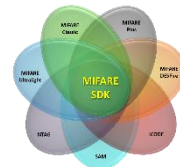
Very Small Footprint Package → SOT 902 (1,6*1,6*0,5 mm)

Read/Write Mode
MIFARE SDK

► Software development tool that lets developers create contactless applications for MIFARE, NTAG and ICODE products.

► Developers are able to benefit from an enormous reduction in development time.
  ▪ Developers focus on designing creative apps and the best GUI brand.
  ▪ Short time from idea to market

► Get rid of "complicated" datasheets and application notes
  ▪ Full command set support on Java level

► Comprehensive documentation: User Manual and Javadoc

► Source code examples to get familiar with the technology

► Talk to our experts on the MIFARE SDK Forum

https://www.mifare.net/es/productos/tools/mifare-sdk/
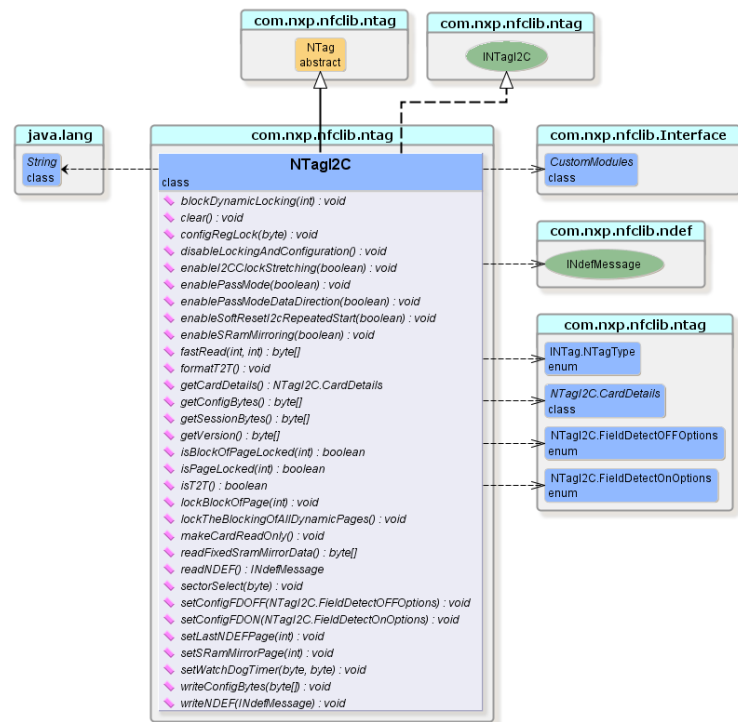
Peer-to-Peer Mode

Peer-to-Peer Mode

► Android NPP: since v2.3
  ▪ Exchange data between two Android devices (Google solution)
  ▪ Fast, easy and very intuitive (tap & touch)

► Android Beam: since v4.0
  ▪ Exchange data between two NFC devices (interoperable solution)
  ▪ More secure solution (Touch to Beam)
  ▪ Compliant with SNEP specification

► Android Beam: since v4.1
  ▪ Provides automatic connection handover mechanism to BT
  ▪ Suitable for media exchanging

Peer-to-Peer Mode
Operating on NFC Tags

► How to develop my Android NFC application

- Indicate NFC support
  - ❖ Defined in the project AndroidManifest.xml file
- Use of the following methods
  - ❖ setNdefPushMessage: it automatically beams the message when two devices are in close proximity
  - ❖ setNdefPushMessageCallback(): it calls a callback method to create the NDEF message just before sharing it

► Android Beam restrictions

- Device must be unlocked and touching the screen is mandatory for security reasons
- The application must be in the foreground
- Exchanging one unique NDEF message on each Android Beam session is allowed
- Android Beam only supports Put Request by SNEP

► Sample application by Android developer's guide

- http://developer.android.com/guide/topics/connectivity/nfc/nfc.html#p2p

Peer-to-Peer Mode
Communication modes

The ISO 18092 standard defines **two modes of communication**
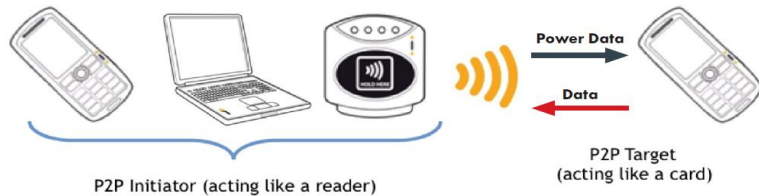
Active / Passive

Only one device generates the magnetic field.
In terms of RF, the initiator behaves like a
reader and the target behaves like an
emulated target

Active / Active

Both devices generate the magnetic field.
Higher bit rates and distances might be
reached, but the solution is more complex
and consumes more battery

Integrating NFC into my Android NFC Reader

PN7120
Best plug'n play full NFC solution

- ► Full **NFC Forum-compliant** controller
- ► Support NFC card emulation, reader/writer and peer-to-peer modes
- ► Compatible with **ISO/IEC 14443-A&B**, **FeliCa** and **ISO/IEC 15693** cards
- ► **Integrated firmware** with **NCI** interface
- ► **Android** and **Linux** software drivers
- ► Low power operation mode

Learn more about PN7120 in our dedicated webinar
PN7120: Best plug'n play full NFC solution

Product website PN7120: www.nxp.com/products/identification_and_security/nfc_and_reader_ics/nfc_controller_solutions/PN7120A0EV.html

PN7120 in a nutshell

Customer Benefits

- ► Low PCB footprint
- ► Low power consumption
- ► EMVCo 2.3.1a PCD analog and digital
- ► NFC Forum Device Requirements v1.3
- ► Full SW stack available for integration within Linux and Android 4.4.x and 5.x

Features

Ease of integration

- ► Direct connection to 5.5V device battery
- ► Flexible clock supply concept
- ► Supports both 1.8 and 3V connections to host controller
- ► Buffered output drivers to connect an antenna with minimum number of external components

Flexibility in use case supports

- ► Fully configurable polling loop with low power modes for automated device discovery
- ► Autonomous mode when host is shut down (host can be in a deep sleep mode and be awakened via IRQ pin by PN7120 when entering RF field)

RF communication modes

Reader/Writer modes

- ► NFC Forum tags Type 1, 2, 3, 4 and 5
- ► ISO/IEC 14443 Type A & B, R/W up to 848 Kbps
- ► ISO/IEC 15693 Tags (ICODE)
- ► FeliCa tags up to 424 Kbps
- ► MIFARE 1K/4K
- ► MIFARE DESFire
- ► Kovio ink printed tags

Card modes

- ► ISO/IEC 14443-A and B card emulation via host

P2P modes

- ► Active and passive initiator and target according to ISO/IEC 18092 at all data rates (106 kbps to 424 kbps)
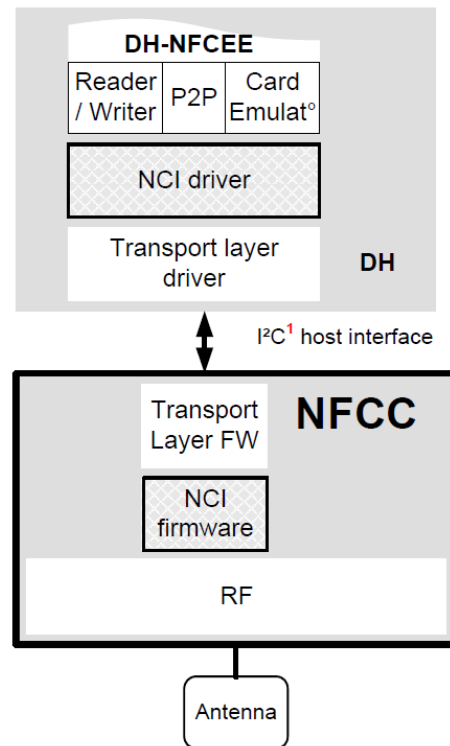
Interfaces

- ► I2C up to 3,4MBaud/s
- ► NFC Forum NCI 1.0 compliant protocol
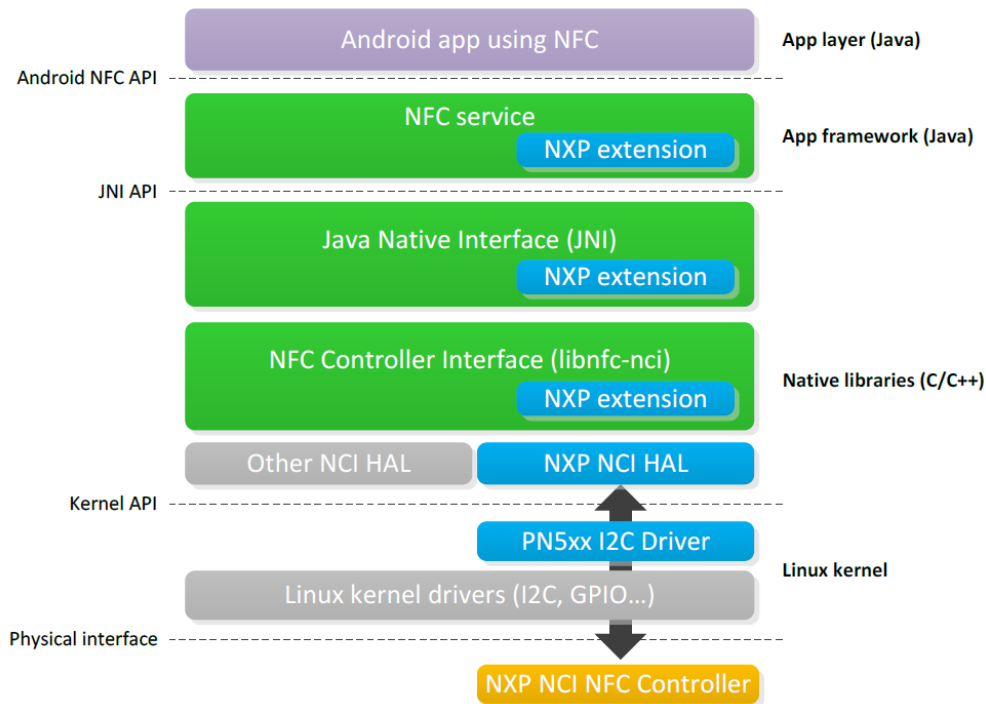
Package

- ► VFBGA49

PN7120 NCI Interface

▶ The NCI defined by the NFC Forum is the specification that defines a standard interface within an NFC device between an NFC controller and the device's main application processor.

▶ The NCI interface provides manufacturers with a standard interface they can use for whatever kind of NFC-enabled device they build

▶ NXP extends NCI interface with a proprietary extension to allow customers access to the entire functionality set defined by the PN7120.

DH-NFCEE

| Reader / Writer | P2P | Card Emulat° |

NCI driver

Transport layer driver **DH**

I²C[1] host interface

Transport Layer FW **NFCC**

NCI firmware

RF

Antenna

PN7120 SW integration in Android
Android NFC stack

► **NFC service:** API within the Android framework that provides access to the NFC functionality.

► **JNI**: Glue code between Java classes and Native classes (written in C/ C++)

► **Libnfc-nci:** Native library providing NFC functionality for which extension is added to support NXP proprietary features

► **NXP NCI HAL**: NXP hardware specific implementation supporting full capabilities

► **PN5xx_I2C driver:** kernel module allowing the access to NXP NCI based NFC Controller hardware resource.

Android app using NFC	App layer (Java)

Android NFC API

NFC service — NXP extension	App framework (Java)

JNI API

Java Native Interface (JNI) — NXP extension	

NFC Controller Interface (libnfc-nci) — NXP extension	Native libraries (C/C++)

Other NCI HAL	NXP NCI HAL

Kernel API

PN5xx I2C Driver	
Linux kernel drivers (I2C, GPIO...)	Linux kernel

Physical interface

NXP NCI NFC Controller

PN7120 SW integration in Android
Android porting guidelines

► PN5xx I2C Kernel mode driver
- Robust and mature communication with the NXP NCI NFC Controller
- Both the libnfc-nci stack and the PN5xx I2C driver are distributed by GitHub:
  ❖ Android libnfc-nci stack: https://github.com/NXPNFCLinux/android_nxp-nci
  ❖ PN5xx I2C driver: https://github.com/NXPNFCLinux/nxp-pn5xx

► AOSP Integration
- Merge NXP-NCI Android NFC package into the target AOSP source directory
- Add NFC to the build by modifying *device/brand/platform/device.mk* file

Integration details explained in dedicated Application Note in Docstore
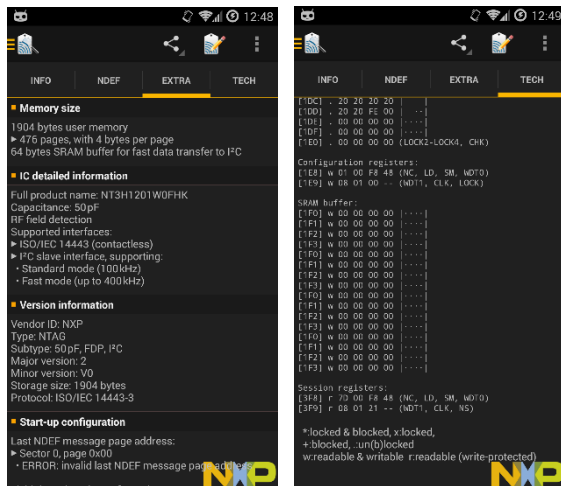
* AN11690 NXP-NCI Android porting guidelines
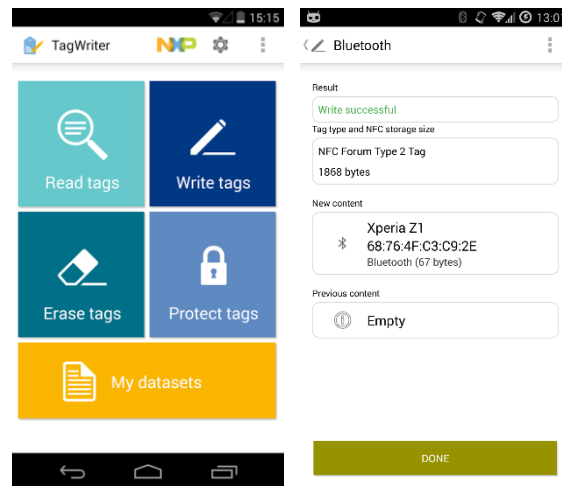
NFC Android apps by NXP

NFC Android apps by NXP
NXP TagInfo and NXP TagWriter

NXP TagInfo

The ideal tool to get detailed information about contactless ICs, explore the capabilities of NFC-enabled items and browse detailed information about the content stored on your NFC tags

NXP TagWriter

The ideal tool to store contacts, bookmarks, Bluetooth and WiFi Handover, etc. to any NFC-enabled items based on NDEF messages. Once data has been stored, it allows us to read the programmed data including options to launch applications based on the stored data.

NFC Android apps by NXP
NTAG I2C Demoboard

► App to be used together with the NTAG I2C Explorer Kit

  ▪ https://nxp-rfid.com/products/ntag/ntag-i2c-explorer-demonstration-and-development-kit/

► Operate on your NTAG I2C IC:

  ▪ Configure the IC using the SRAM by changing the LED color to light up on the NTAG I²C demonstration board

  ▪ Measure the speed of download and upload with NTAG I²C based on your NFC device

  ▪ Read & change the IC configuration by updating the NTAG I2C Session & Configuration Registers

  ▪ And more …

Conclusion

NFC in Android
Wrap up

► Android is the undeniable Mobile OS market leader and NFC is becoming a commodity
  ▪ It opens a huge new business opportunity for companies

► Android devices are the ideal way to communicate with NFC infrastructure readers
  ▪ Connected tags in Read/Write Mode
  ▪ NXP NFC Readers in Card Emulation & Peer-to-Peer Mode

► PN7120 is the best plug'n play solution to integrate NFC in your target NFC Reader device

► NXP NFC Android applications let you understand the NFC basics and interact with your NFC tags

MobileKnowledge
Thank you for your attention

► We are a global competence team of hardware and software technical experts in all areas related to contactless technologies and applications.

► Our services include:
  - Application and system Design Engineering support
  - Project Management
  - Technological Consulting
  - Advanced Technical Training services

► We address all the exploding identification technologies that include NFC, secure micro-controllers for smart cards and mobile applications, reader ICs, smart tags and labels, MIFARE family and authentication devices.

For more information

Eric Leroux
eric.leroux@themobileknowledge.com
+34 629 54 45 52