



E0-245: ASP

Lecture 18: NFC

Dipanjan Gope



Module 2: Android Sensor Applications

- Location Sensors
 - Theory of location sensing
 - Package android.location
- Physical Sensors
 - Sensor Manager
 - Accelerometer
 - Gyroscope
 - Magnetometer
 - Sensor fusion
- NFC
- Multimedia
 - Camera
 - Microphone

References

- Greg Milette, Adam Stroud: Professional Android Sensor Programming, 2012, Wiley India
- [Google I/O 2011: How to NFC](#)

Popular Apps with NFC



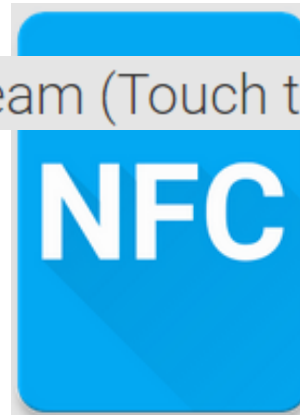
PAYMENT APPS



SuperBeam | WiFi Direct Share

PAIRING USING NFC

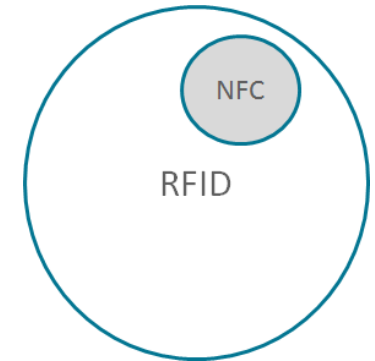
File Beam (Touch to Send)



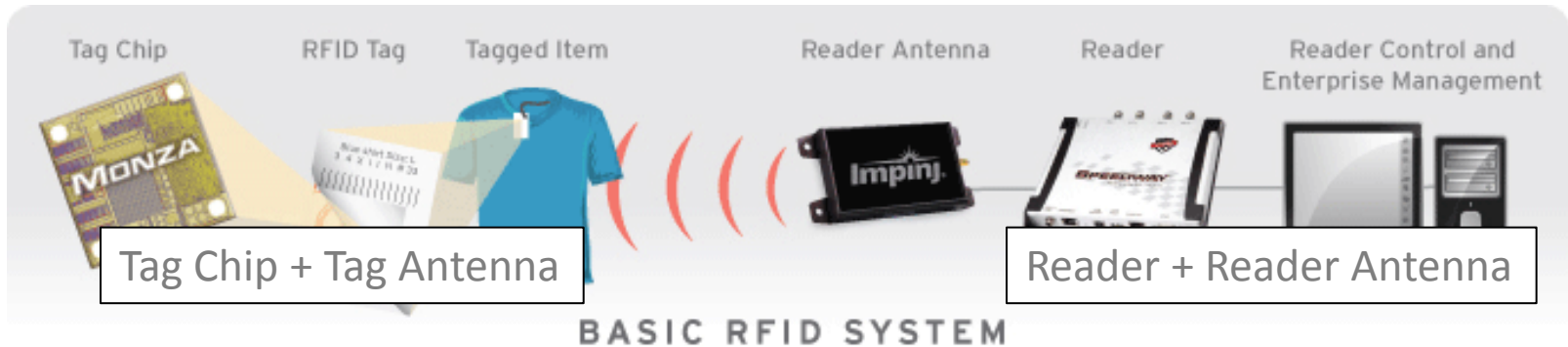
SHARING APPS

RFID vs NFC

- NFC is a special form of RFID



RFID



NFC



Reader (+ "tag")
Communicating
at $\sim < 4\text{cm}$

RFID vs NFC

- Range
 - NFC: 10cm
 - RFID: upto 100m
- Frequency
 - NFC: 13.56MHz
 - RFID: LF(125-134kHz – $r \sim 10\text{cm}$); HF(13.56MHz – $r \sim 30\text{cm}$); UHF(856-960MHz- $r \sim 100\text{m}$)
- Power
 - NFC: Device – active; Stickers – passive
 - RFID: Active or passive

History of NFC

- 1983: RFID patent
- 2004: NFC Forum
- 2006: NFC tag specifications
- 2006: Nokia 6131 with NFC support
- 2010: Samsung Nexus S Android mobile NFC

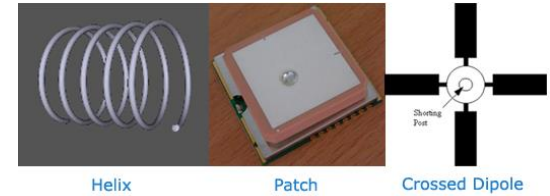
5 billion devices even before 2011

- Transit
- Credit cards
- Passports
- Physical access

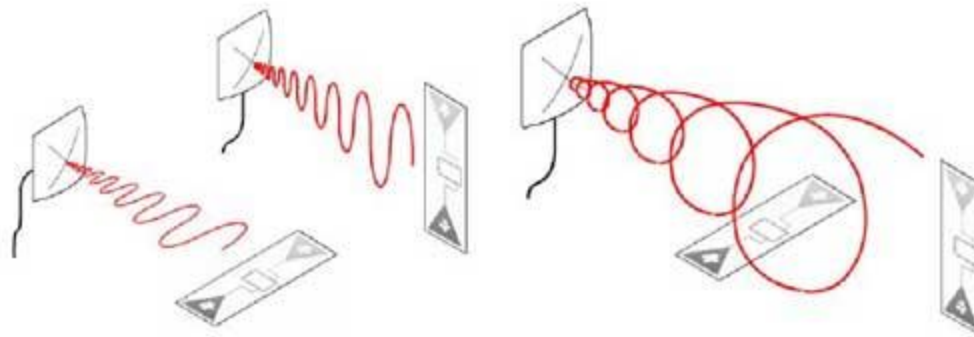
RFID UHF Antenna

Reader Antenna:

- Linear polarization (dipole antenna)
- Circular polarization (helix, cross-dipole, patch)
- Monostatic circular / Bistatic circular

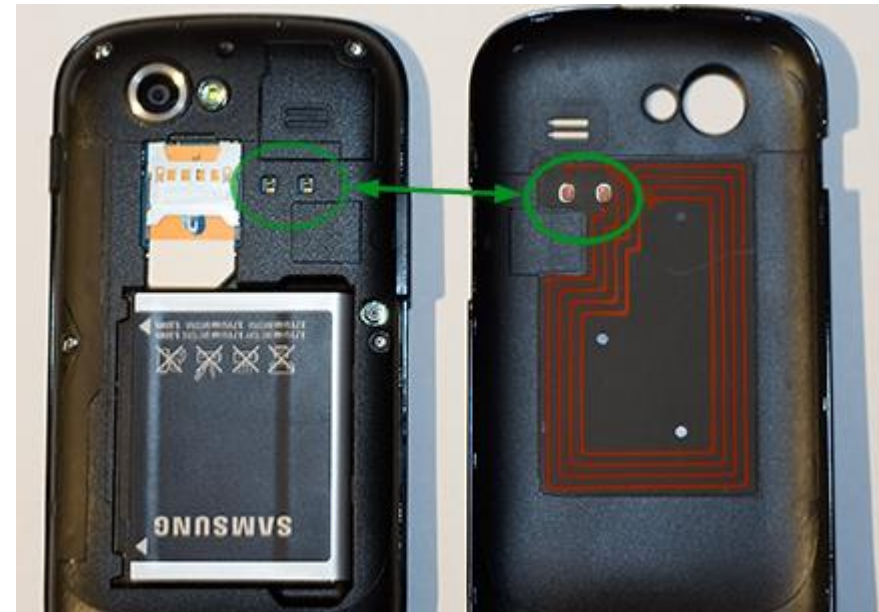


http://skyrfid.com/index.php?pr=RFID_Circular_Antenna_Types



NFC Antenna

- 13.56MHz – 22m wavelength
- Not an antenna – but an inductor



NFC tag
9

NFC modes

- Reader/writer mode
 - smart posters
 - v-cards
- Peer-to-peer
 - exchange data
 - exchange money
 - bluetooth/wifi connection setup
- Card emulation mode
 - payments
 - ticketing
 - access
 - identity
 - loyalty

NFC (peer-to-peer) vs Bluetooth

	NFC	Bluetooth
Connection Setup	Low Friction (No discovery or pairing – 0 click)	High Friction (12 sec discovery + UI for pairing)
Range	Low (1-4cm)	High (10s of meters)
Data Rate	Low (106-414kbps)	High (>3Mbps)

NFC (reader-tag) vs QR code

	NFC Tags	QR code
User Experience	High (0-click)	Low (align and click)
Cost	\$0.3	\$0
Size	10-30mm	20mmx20mm
Security	Can be encrypted	Open

NFC Data Format

- NDEF: NFC Data Exchange Format
- NDEF message vs. NDEF record



Ref: [2]

Payload length: octet or 8 bits

Payload type: URI/MIME/Smart poster/Signature

- Type Name Format – byte array

- e.g. NdefRecord.TNF_ABSOLUTE_URI or TNF_WELL_KNOWN (text)

Payload Identifier – byte array

NFC Tags

- Type 1: Topaz
 - Type 2: MIFARE Ultralight
 - Type 3: Sony Felica
 - Type 4: MIFARE Desfire
-
- Android: peer to peer

NFC tags

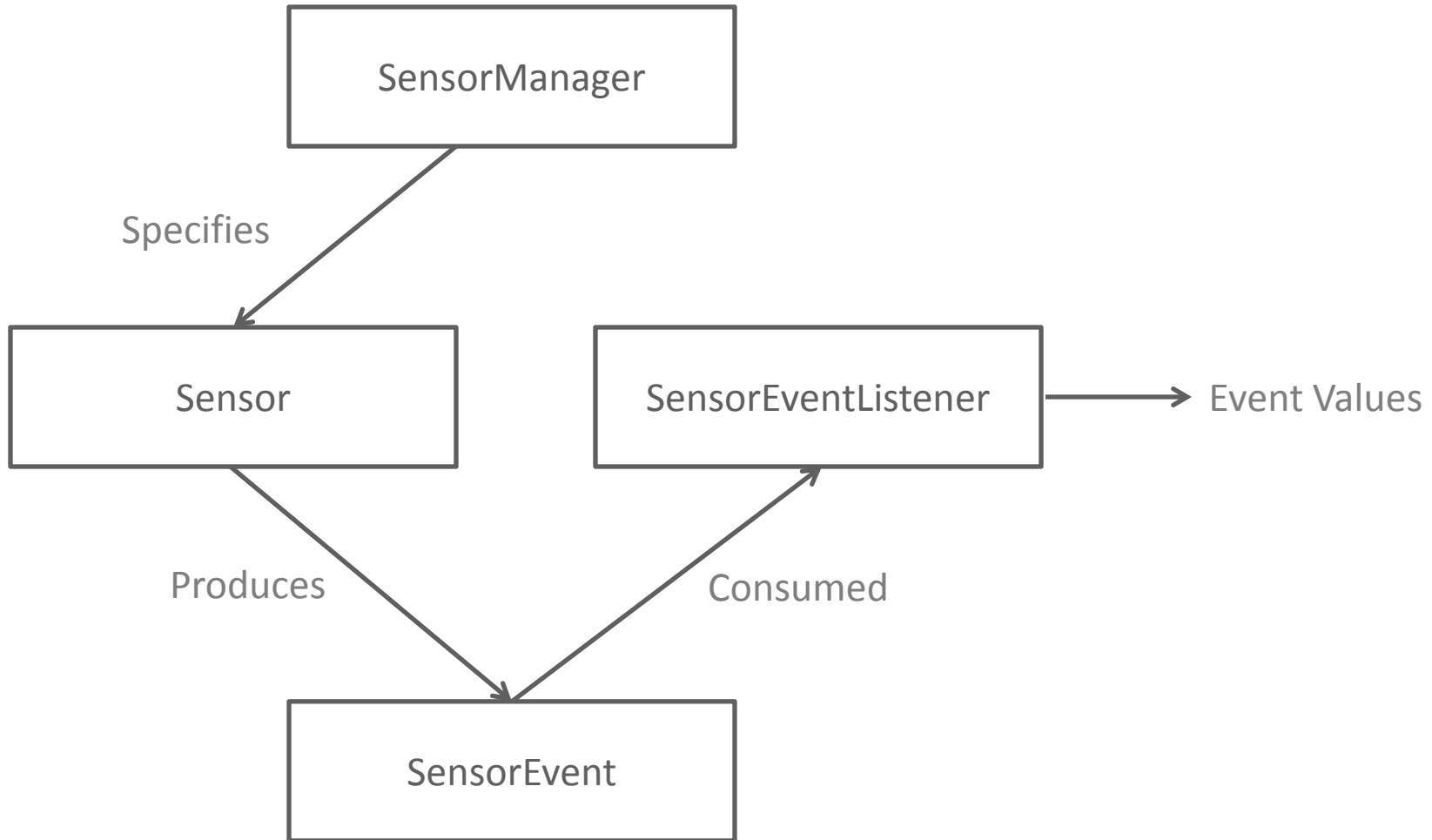
NFC FORUM TYPE	POPULAR PRODUCTS OF THIS TYPE	OPERATIONS SPECIFICATIONS	REWRITE CAPABILITIES	AVAILABLE MEMORY	COMMUNICATION SPEED	PRICE RANGE (PRICE PER UNIT)
1	Broadcom Topaz	ISO 14443A	User rewritable; can be marked as read-only by user	96 bytes, expandable to 2KB	106kbit/s	Low (~\$1-2 USD)
2	MIFARE UltraLight	ISO 14443A	User rewritable; can be marked as read-only by user	48 bytes, 144 bytes is common, expandable to 2KB	106kbit/s	Low (~\$1-2 USD)
3	Sony FeliCa	JIS X 6319-4	Manufacture pre-configured to be read-only or re-writable.	variable, theoretical 1MB	212kbit/s or 424kbit/s	High (~\$8-10 USD or higher)
4	NXP DESFire, NXP SmartFX	ISO 14443A, ISO 14443B	Manufacture pre-configured to be read-only or rewritable.	4KB for DESFire, up to 32KB for SmartFX	Up to 424kbit/s	Medium-High (~\$3-4 USD)

Security and Privacy

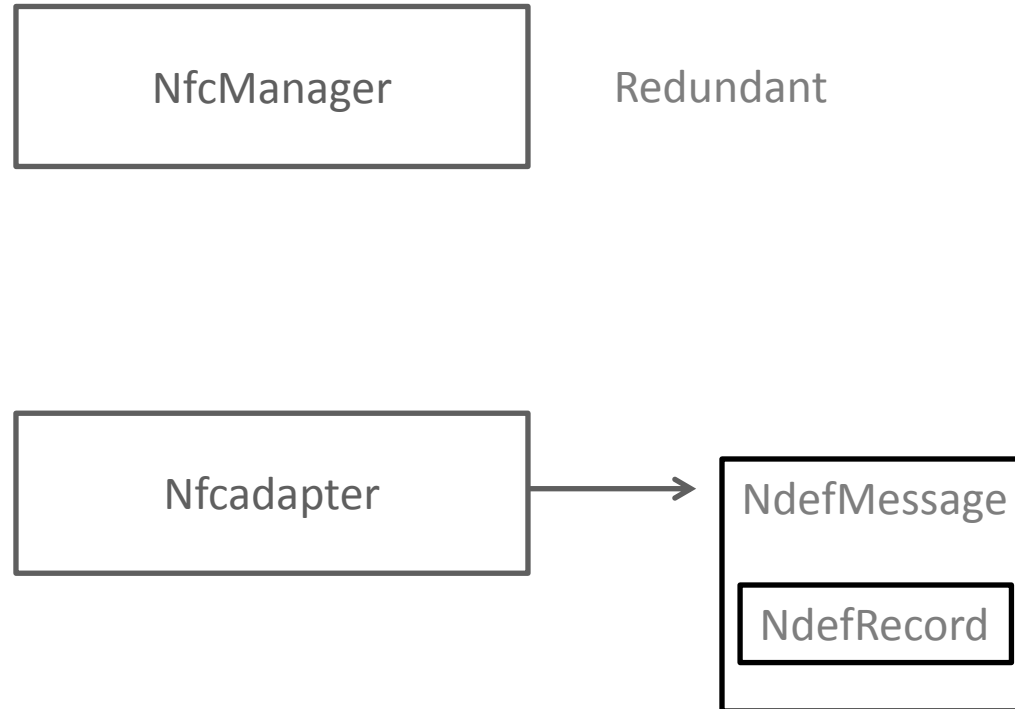
- Proximity
- Screen off turns NFC off
- Apps must be in foreground to access NFC
- No default link level encryption

Lets Code...

Package: Android.hardware



Package: Android.nfc Classes



Manifest File

```
<uses-permission android:name="android.permission.NFC"></uses-permission>
```

```
<!-- Handle notes detected from outside our application -->  
<intent-filter>  
  <action android:name="android.nfc.action.NDEF_DISCOVERED" />  
  <category android:name="android.intent.category.DEFAULT" />  
  <data android:mimeType="text/plain" />  
</intent-filter>  
</activity>
```

Reading NDEF

- Part 1: Dispatch
 - NDEF data is dispatched to only 1 activity
 - Type of first NDEF record is used for dispatch

Ref: [2]

Reading NDEF

- Part 2: Dispatch Types



Ref: [2]

Reading NDEF

- Part 3:Receiving

AndroidManifest.xml:

```
<!-- Handle notes detected from outside our application -->
<intent-filter>
    <action android:name="android.nfc.action.NDEF_DISCOVERED" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:mimeType="text/plain" />
</intent-filter>
</activity>
```

MyActivity.java:

```
protected void onResume() {
    ...
    NdefMessage msg =
        getIntent().getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES)[0];
    Byte[] payload = msg.getRecords()[0].getPayload();
    ...
}
```

Ref: [2]

Writing NDEF Tags

```
String text;
NdefRecord textRecord = new NdefRecord(NdefRecord.TNF_MIME_MEDIA,
    "text/plain".getBytes(), text.getBytes());
NdefMessage textMessage = new NdefMessage(new NdefRecord[] {textRecord});
Tag tag = getIntent().getExtra(NfcAdapter.EXTRA_TAG);
Ndef ndef = Ndef.get(tag);
ndef.writeNdefMessage(textMessage);
```

Ref: [2]

Writing NDEF P2P

- **Foreground Activities** can register an NDEF payload for P2P push

Register the payload in advance

```
adapter.enableForegroundNdefPush(this, ndefMessage);
```

- **Ice Cream Sandwich preview**

Register interest in P2P in advance, push the payload live

```
public interface NdefPushCallback {  
    public NdefMessage createMessage();  
}  
adapter.registerForegroundNdefPush(this, callback);
```

Ref: [2]

Summary

TAG

P2P

READ

```
if (NfcAdapter.ACTION_TAG_DISCOVERED.equals(action)
    || NfcAdapter.ACTION_NDEF_DISCOVERED.equals(action)) {
    Parcelable[] rawMsgs = intent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);
    if (rawMsgs != null) {
        msgs = new NdefMessage[rawMsgs.length];
        for (int i = 0; i < rawMsgs.length; i++) {
            msgs[i] = (NdefMessage) rawMsgs[i];
        }
    } else {
        ...
    }
}
```

WRITE

```
ndef.writeNdefMessage(message);
```

```
mNfcAdapter.enableForegroundNdefPush(MainActivity.this, getNoteAsNdef());
```

```
adapter.registerForegroundNdefPush(this, callback);
```

Google StickyNote Application

