



Google
Developers



Up Close and Personal: NFC & Android Beam

Adding magic sharing to your applications

Robert Hamilton
Produktchef

Martijn Coenen
Software Engineer





What is NFC

...for those of you who don't know yet

NFC

The basics

- Near Field Communication
- Short-range (<5cm) wireless technology
- Low data-rates (106 - 424 kbit/s)
- No connection setup



NFC

How do you interact with it?



Passive tags



Powered NFC devices



5cm...

...isn't that very limiting?

- Short range is the feature!
- Touching physical objects for digital interaction
- Beaming virtual objects



What can you do with NFC?

Almost anything...

- Payments
- Ticketing and voucher redemption (Clipper Card, Oyster Card, PasMo, SUICA)
- Real-world gaming
- Home automation
- Location-based tagging
- Map a real object to a URL

The bridge between the physical and digital world
The limits of NFC are the limits of your imagination



NFC is coming

Don't wait too long!

1,000,000 new NFC devices per week

- NFC tags appearing in the wild
- More and more peripherals have NFC
- More and more Android devices carry NFC on-board!



NFC in Android

Simple, instant tag interactions

- Supports all common passive tag types
- APIs for formatting tags, reading and writing
- Immediately launch the appropriate application



Android Beam

Magic sharing

- The idea behind Beam: share what's shown on screen
- Bring two NFC-capable devices in range
- Either of the two can touch to send





Android Beam Demos

Magic sharing



Android Tag Demos

Instant gratification

New in J: NFC-initiated Bluetooth pairing

No more PINs

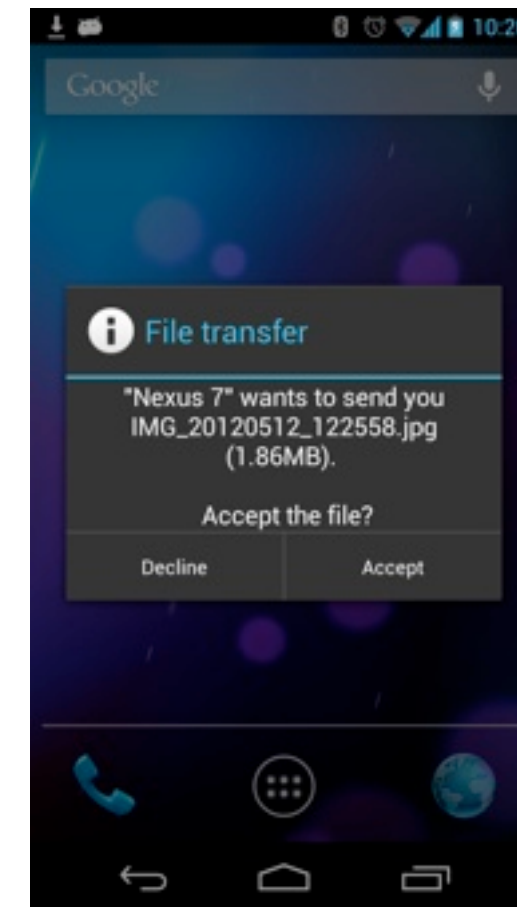
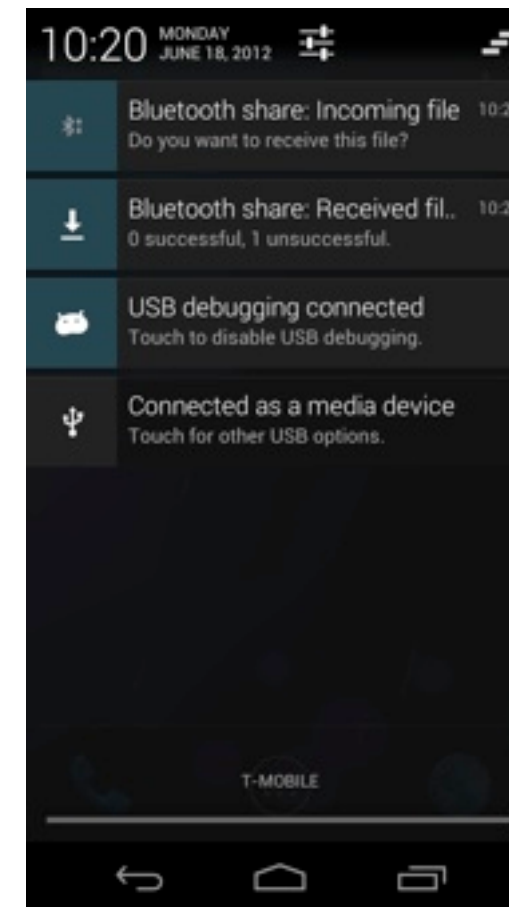
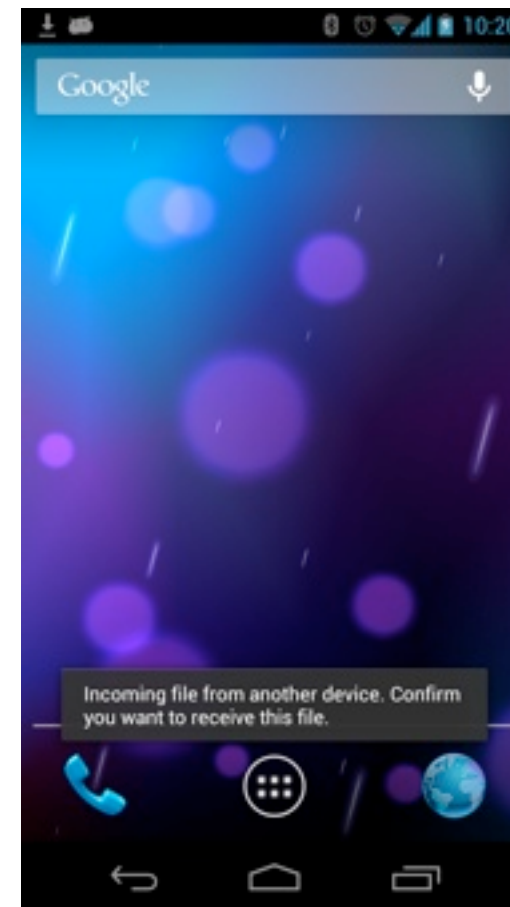
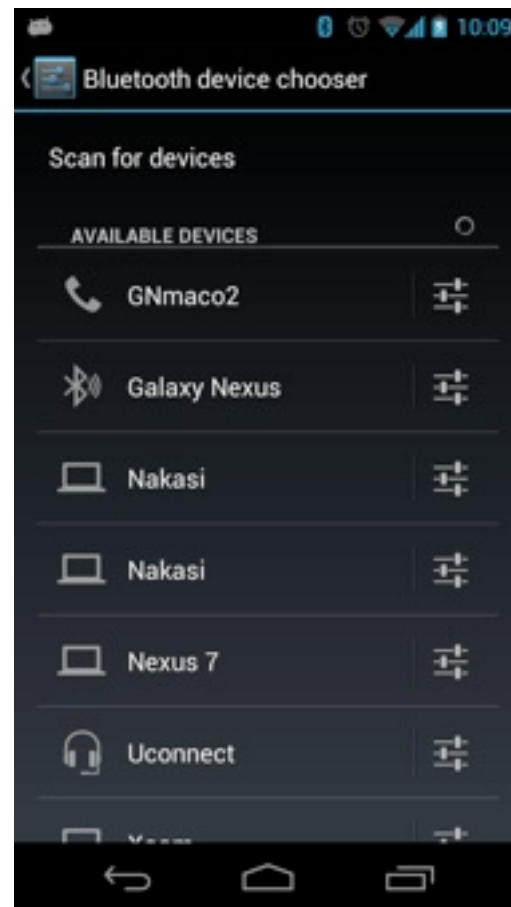
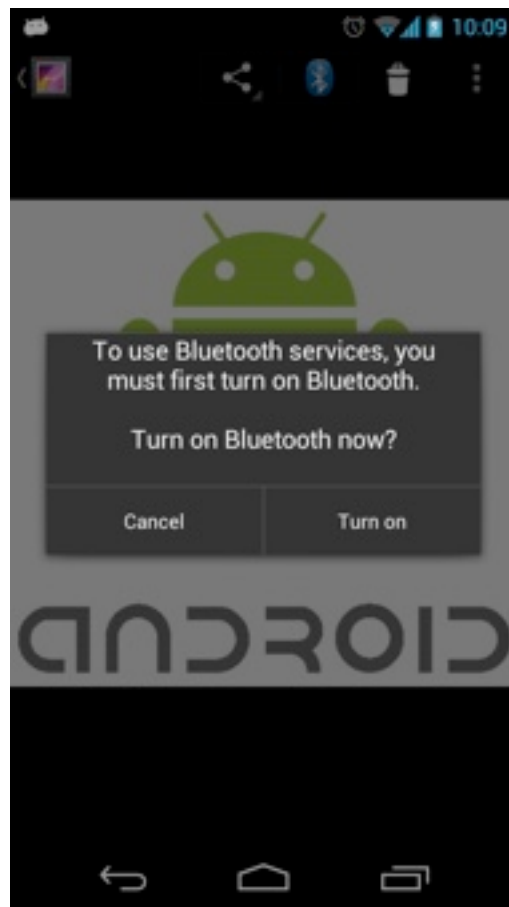
- Tap once to pair and start play
- Tap once again to stop



New in J: Android Beam for media

We can go faster!

Sending a file over Bluetooth is hard work!



New in J: Android Beam for media

We can go faster!

- Send photos, videos and files with Android Beam
- Initiate with NFC but move data over Bluetooth





Using NFC in your applications

Just a few lines of code!

NFC in your applications

Wait...why should you care?

- Users love NFC
- NFC can be a differentiator for your app
- Really simple to implement



NFC in your applications

Opportunities

- Think of new and exciting applications
 - Deploying NFC tags to interact with objects or locations
 - Using NFC tags to control the behavior of your phone or tablet
- Adding Android Beam support to your existing application



How to integrate Android Beam

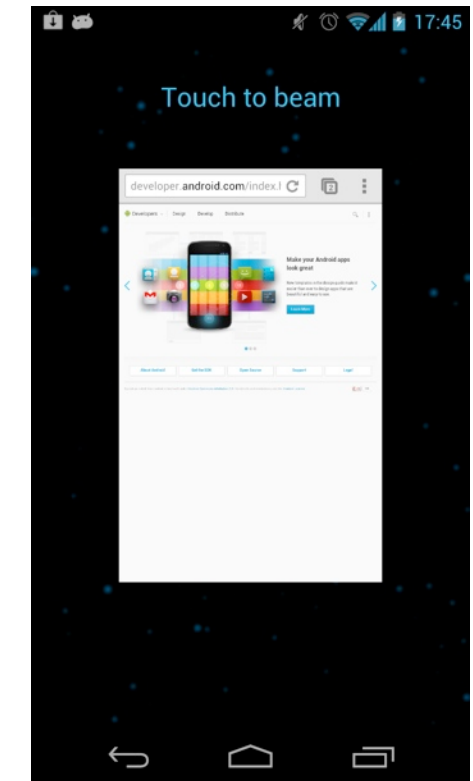
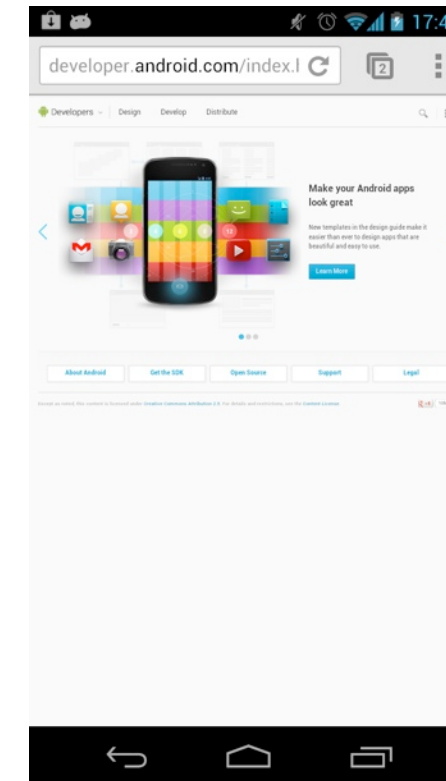
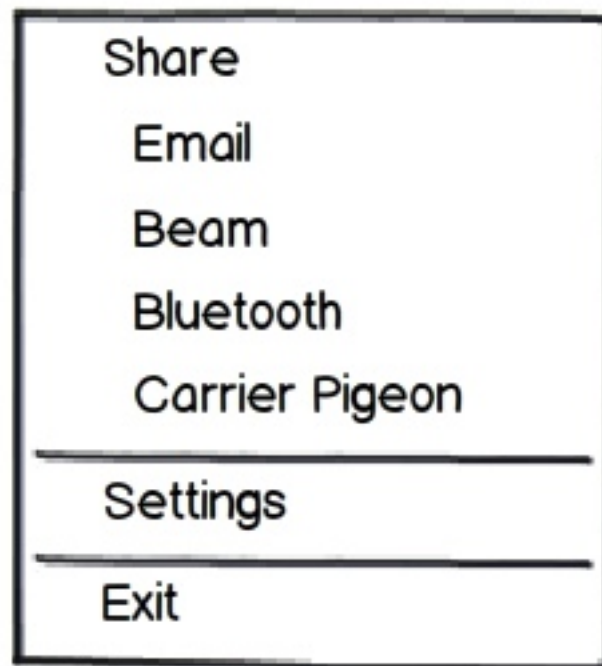
What to share?

- Share what's shown on screen
- Alternatively share "context" or "settings"



That's not Beam!

Don't chase the magic away

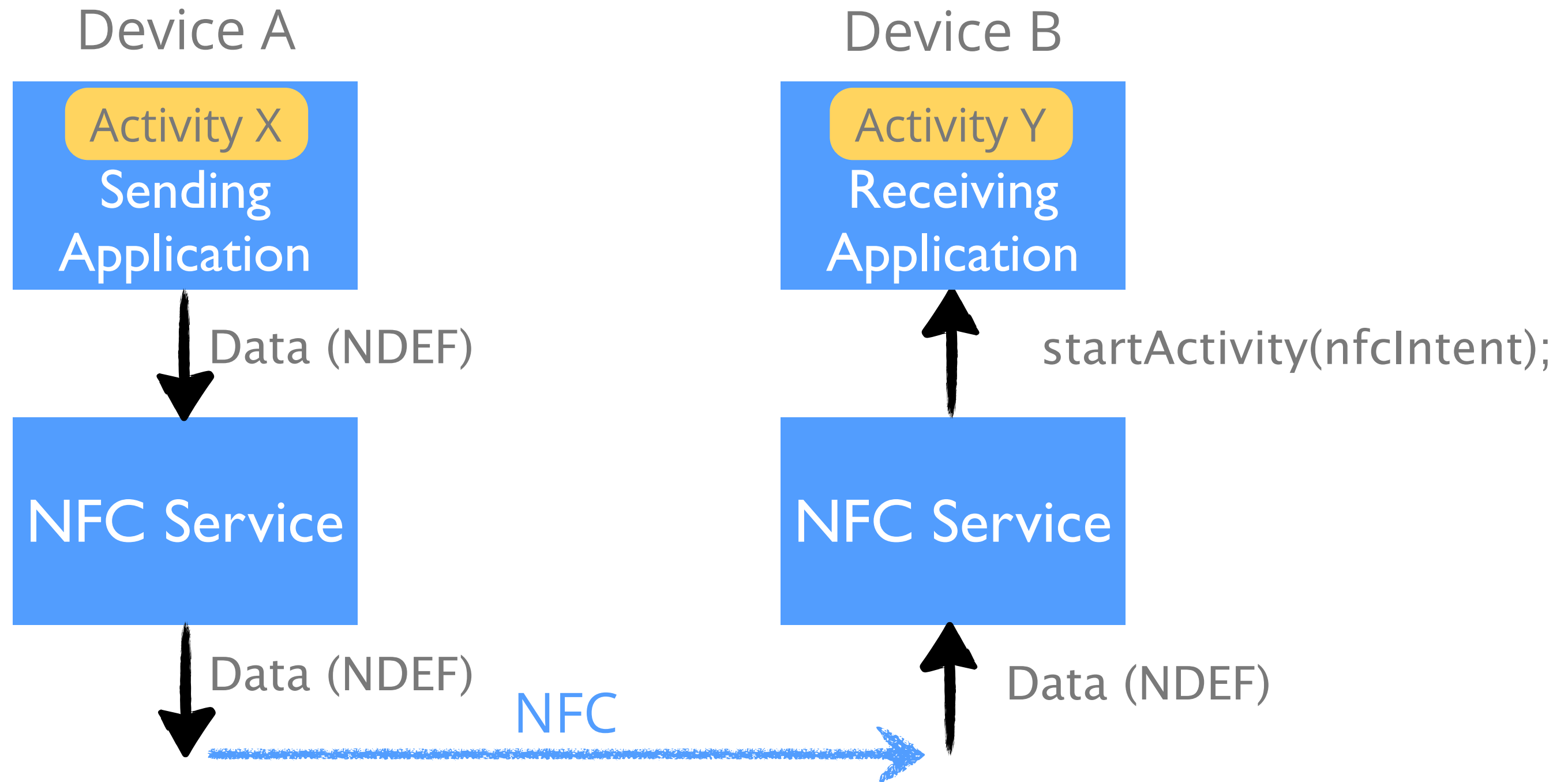


You can tell it's Beam when "It Just Works"



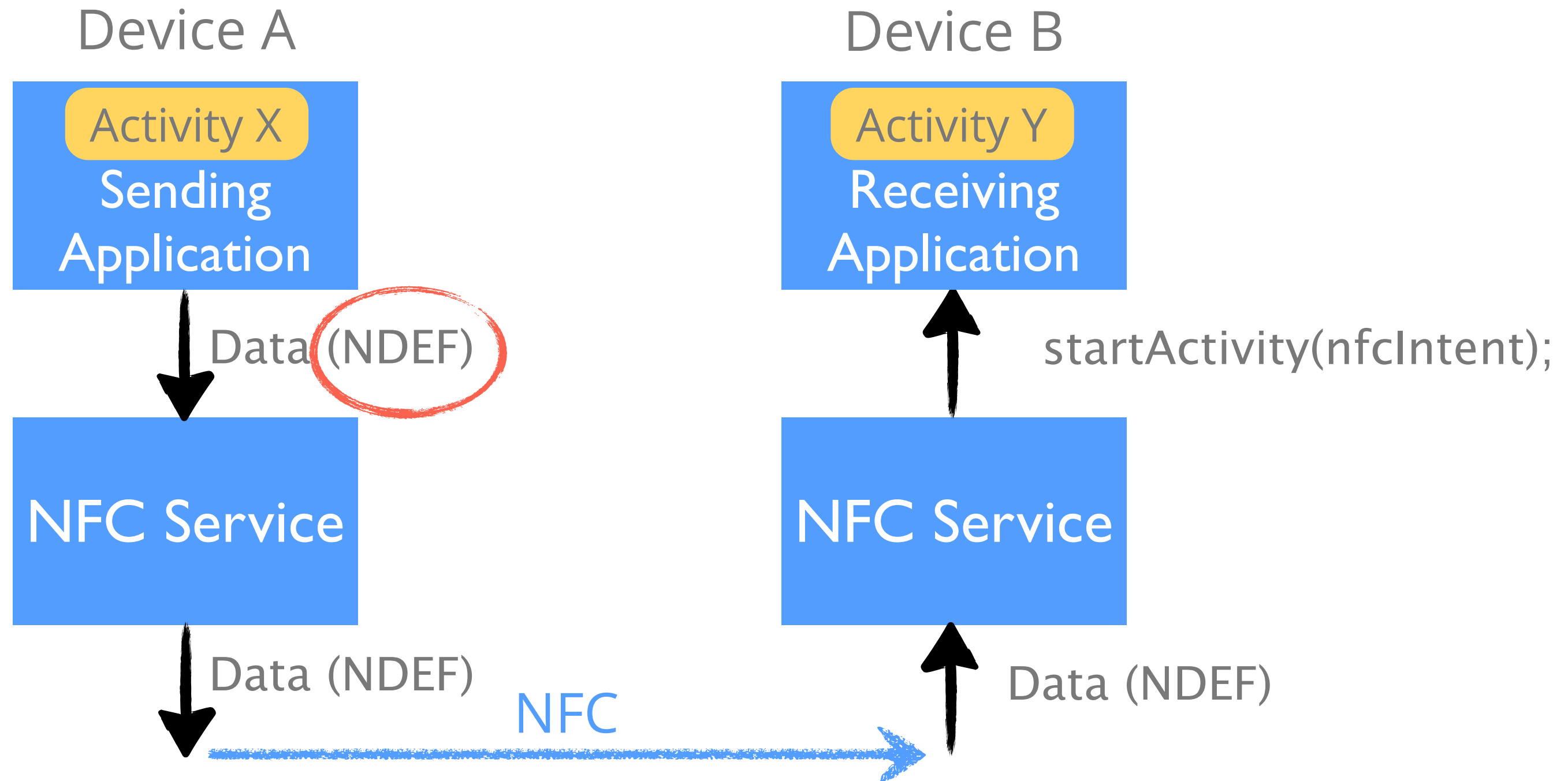
How to integrate Android Beam

Android Beam data flow



How to integrate Android Beam

Android Beam data flow



How to build an NDEF message

NFC Data Exchange Format

NDEF Message

NDEF Record 1

- Type Name Format
- Type
- ID
- Payload

NDEF Record 2

- Type Name Format
- Type
- ID
- Payload

Type Name Format == 0x01 (TNF_WELL_KNOWN)

Type == 0x55 => This is an Uri record

Type == 0x54 => This is a text record

Type Name Format = 0x02 (TNF_MIME_MEDIA)

Type field contains the mime-type string



How to build an NDEF message

Use the helpers!

```
// Creates a URI record
NdefRecord uriRecord = new NdefRecord(NdefRecord.TNF_WELL_KNOWN,
    NdefRecord.RTD_URI, "id".getBytes(),
    "http://www.google.com".getBytes());

// Easy way: creates a URI record
NdefRecord uriRecord = NdefRecord.createUri("http://www.google.com");
// Creates a record with a custom mime type and payload
NdefRecord mimeRecord = NdefRecord.createMime("application/vnd/mime",
    new byte[] {0x00, 0x01});

// Create a message from one of these records
NdefMessage ndefMessage = new NdefMessage(uriRecord);
```



How to build an NDEF message

Existing or custom mime-type?

- Existing mime-types: other apps may deal with your data
- Custom mime-types: if you want only your application to deal with it



How to build an NDEF message

Android Application Records

- Guaranteed delivery to a specified package
- Even if the package does not support NFC at all, it will still be launched
- Opens up the Play store if the package is not installed



How to build an NDEF message

Using Android Application Records

```
// Creates a URI record
```

```
NdefRecord uriRecord = NdefRecord.createUri("http://www.google.com");
```

```
// Creates an Android Application Record
```

```
NdefRecord aarRecord = NdefRecord.createApplicationRecord(  
    "com.example.apps.myapp");
```

```
// Create a message that delivers the URI to "com.examples.app.myapp"
```

```
NdefMessage ndefMessage = new NdefMessage(uriRecord, aarRecord);
```



How to pass the NDEF message

Use the setNdefPushMessage... APIs

```
NfcAdapter.setNdefPushMessage(NdefMessage msg, Activity activity,  
    Activity...);
```

OR

```
NfcAdapter.setNdefPushMessageCallback(  
    NfcAdapter.CreateNdefMessageCallback callback, Activity  
    activity, Activity...);
```



How to pass the NDEF message

Use the setNdefPushMessage... APIs

```
public class X extends Activity
    implements NfcAdapter.CreateNdefMessageCallback {

    protected void onCreate() {
        NfcAdapter adapter = NfcAdapter.getDefaultAdapter(this);
        // Set the callback
        adapter.setNdefPushMessageCallback(this, this);
        // Alternatively, use the static set function:
        // adapter.setNdefPushMessage(msg, this);
    }
}
```



How to pass the NDEF message

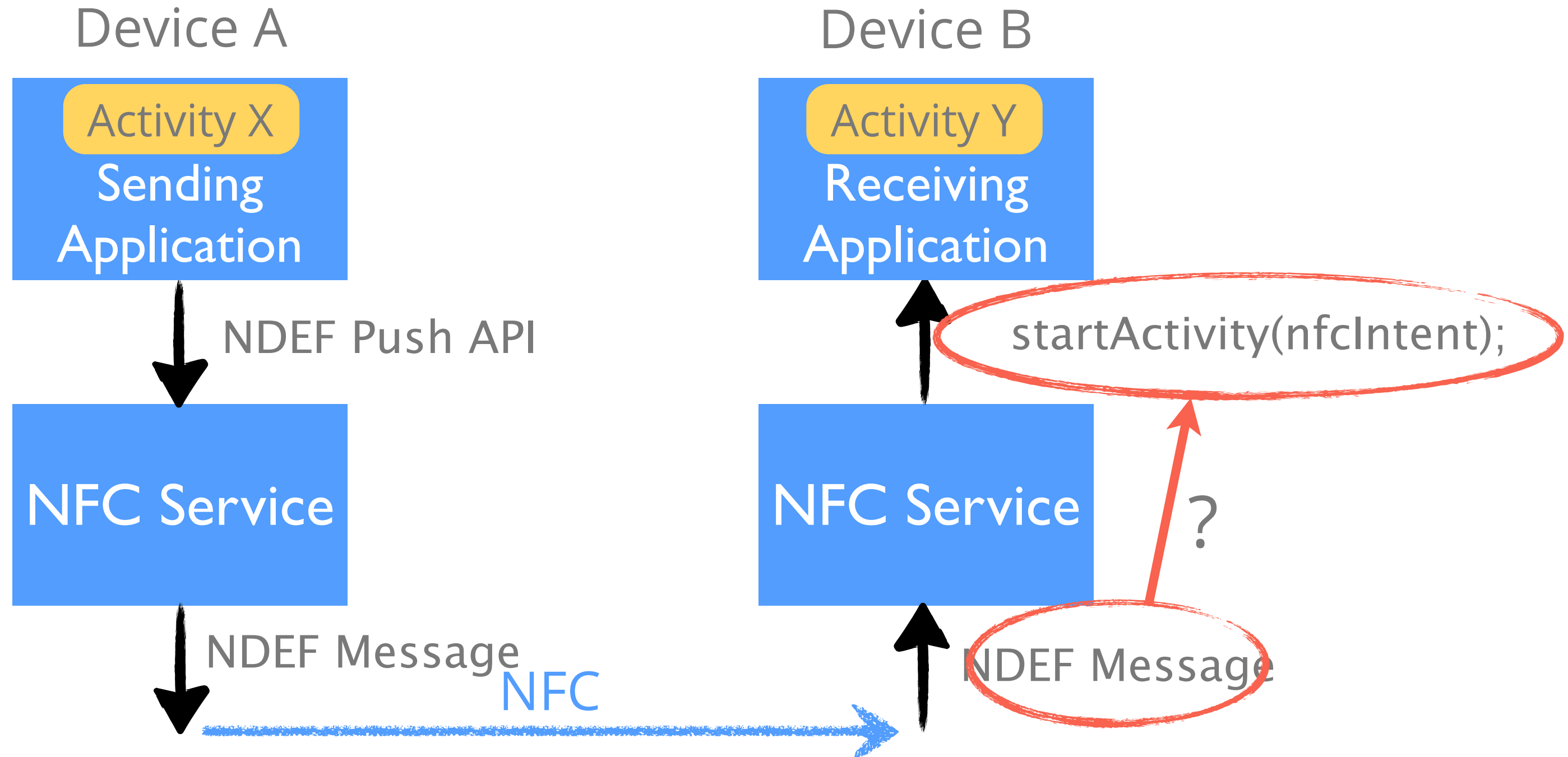
Use the setNdefPushMessage... APIs

```
// Called when the NFC link comes up, don't block or sleep!  
public NdefMessage createNdefMessage() {  
    // Create NDEF message msg with payload  
    byte[] payload = new byte[] ({0x00, 0x01});  
    NdefRecord mimeRecord = NdefRecord.createMime(  
        "application.vnd/mime", payload);  
  
    return new NdefMessage(mimeRecord);  
}
```



How NDEF is delivered

Translating NDEF to an Intent



How NDEF is delivered

Looking at the first record

NDEF Message

NDEF Record 1

- Type Name Format
- Type

- ID
- Payload

NDEF Record 2

- Type Name Format
- Type

- ID
- Payload

```
Intent nfcIntent = new Intent(
    NfcAdapter.ACTION_NDEF_DISCOVERED);

nfcIntent.setType(...); // OR, intent.setData(...);

nfcIntent.putExtra(
    NfcAdapter.EXTRA_NDEF_MESSAGES,
    ndefMessages);

startActivity(nfcIntent);
```



How NDEF is delivered

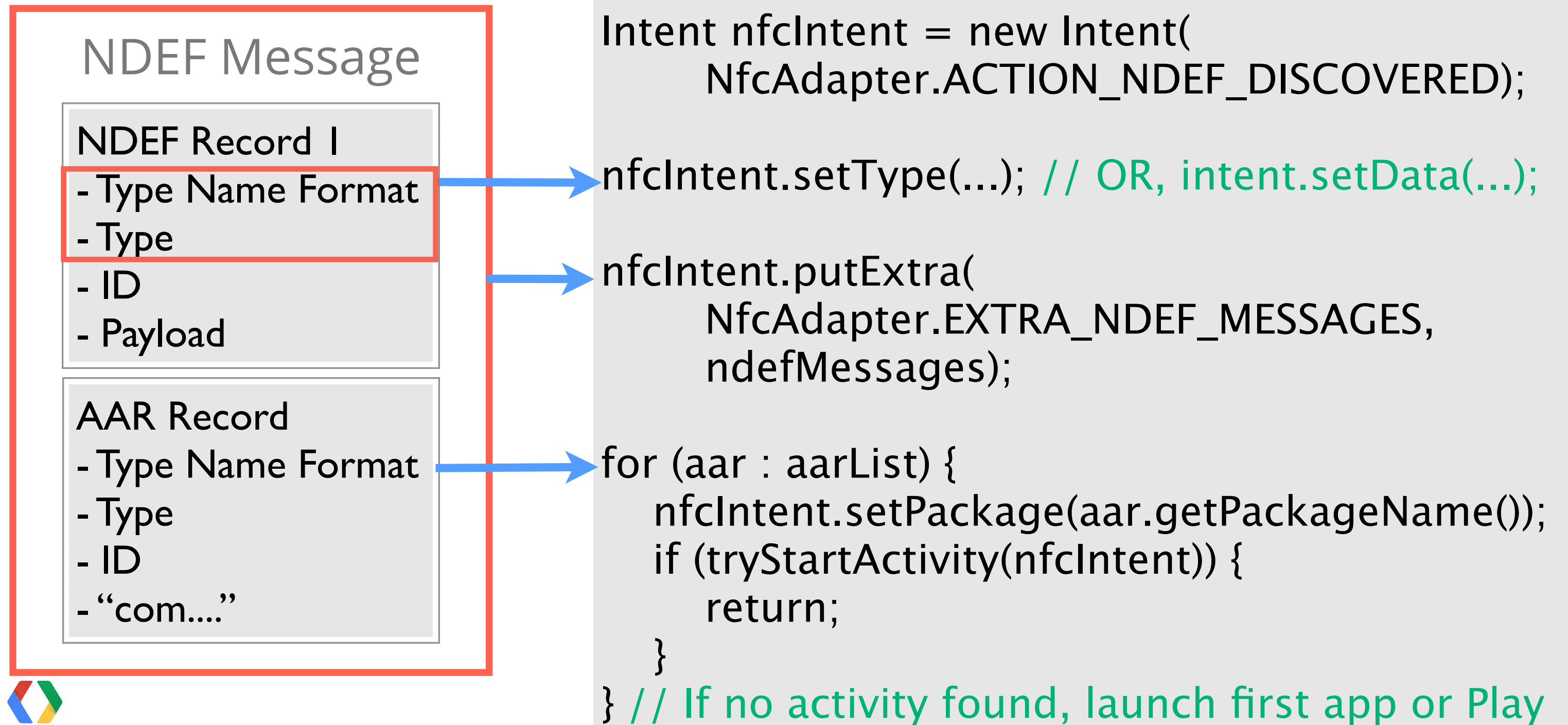
Mapping of NDEF Record TNF/Type to intent type

NDEF record	Type Name Format	Type	Payload	Intent type / data
URI	TNF_WELL_KNOWN (0x01)	RTD_URI (0x55)	" http://www.google.com "	http://www.google.com
Text data	TNF_WELL_KNOWN (0x01)	RTD_TEXT (0x54)	"My Text"	text/plain
Custom mime-type	TNF_MIME_MEDIA (0x02)	"application/vnd.mine"	"1011011000101110"	application/vnd.mine
NFC Forum External Type	TNF_EXTERNAL (0x04)	"android.com:my type"	"1011011000101110"	vnd.android.nfc://ext/android.com:mytype



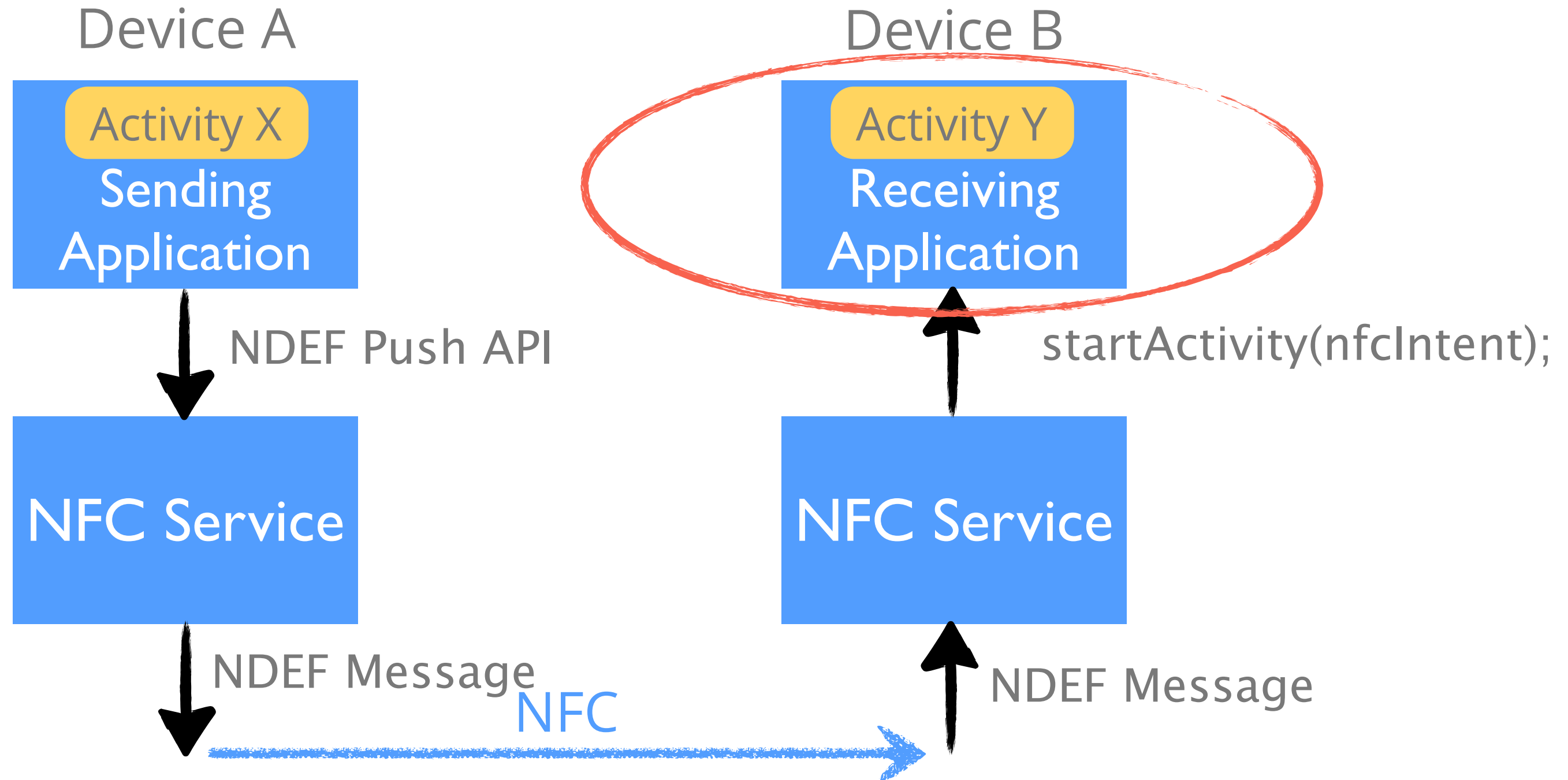
How NDEF is delivered

If Android Application Records are in the message



How NDEF is delivered

Filtering for the right intent



How NDEF is delivered

Inside <Activity> of AndroidManifest.xml

```
<Activity android:name="Y">
<!-- filter for mime-type -->
<intent-filter>
  <action android:name="android.nfc.action.NDEF_DISCOVERED" />
  <category android:name="android.intent.category.default" />
  <data android:mimeType="application/vnd.mine" />
</intent-filter>

<!-- filter for Uri http://www.google.com/nfc -->
<intent-filter>
  <action android:name="android.nfc.action.NDEF_DISCOVERED" />
  <category android:name="android.intent.category.default" />
  <data android:scheme="http" android:host="google.com" android:path="/nfc"/>
</intent-filter>
```



How NDEF is delivered

In Activity Y.java

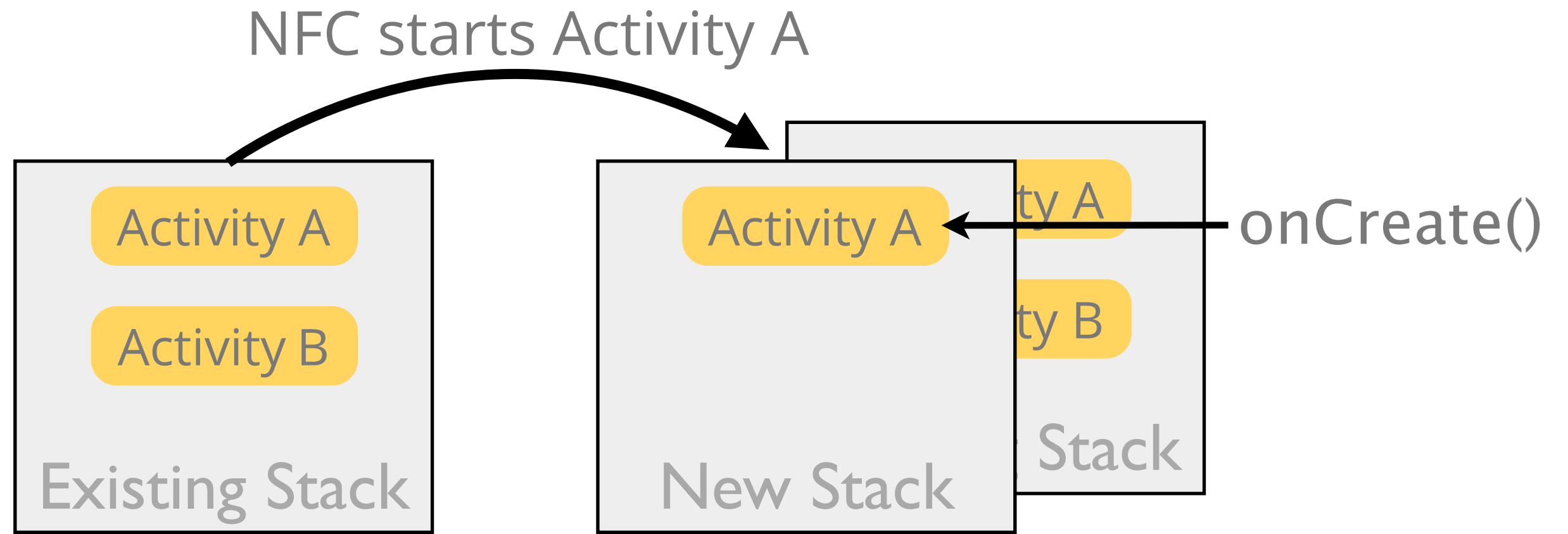
```
protected void onCreate() {
    Intent launchIntent = getIntent();
    String action = launchIntent.getAction();
    if (action.equals(NfcAdapter.ACTION_NDEF_DISCOVERED)) {
        // Get the first NdefMessage
        NdefMessage msg = (NdefMessage) intent.getParcelableArrayExtra(
            NfcAdapter.EXTRA_NDEF_MESSAGES)[0];
        // Get the payload of the first record
        byte[] payloadData = msg.getRecords()[0].getPayload();
        // Process payload (on different thread if needed)
    }
    ...
}
```



How NDEF is delivered

Receiving Pro Tips: Task stacks

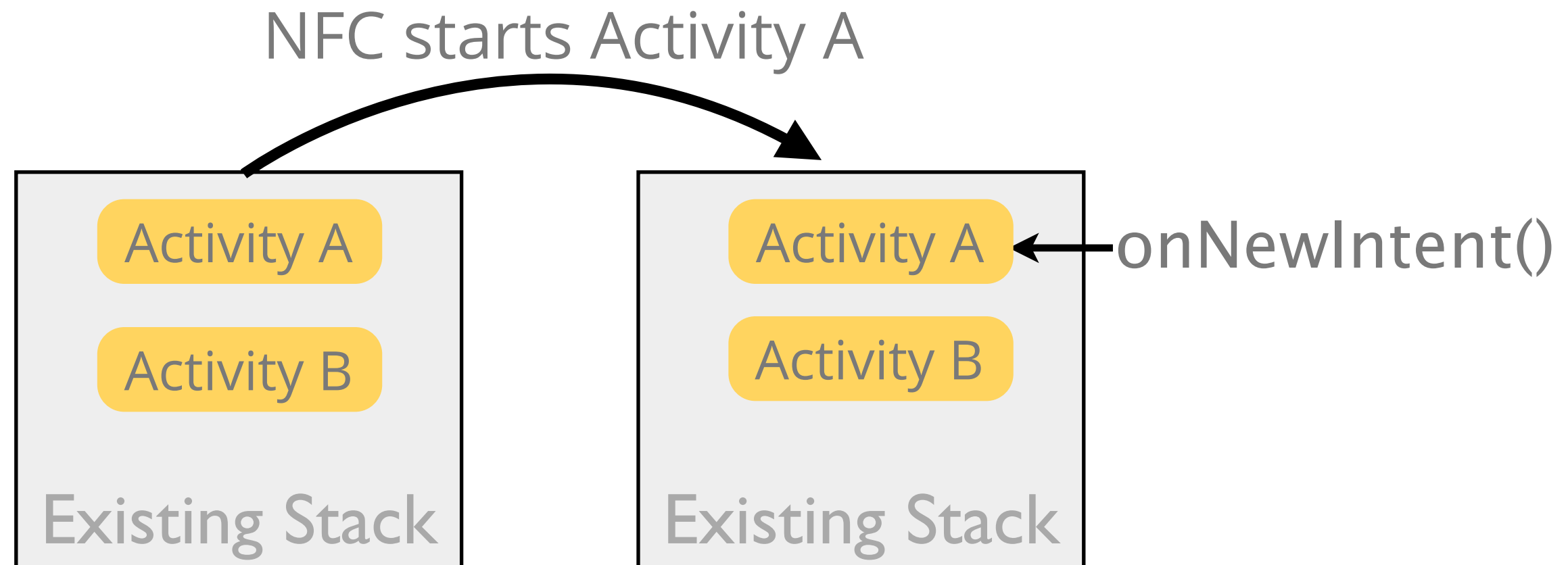
Every incoming NFC interaction by default launches in a new task stack



How NDEF is delivered

Receiving Pro Tips: Multiple NFC interactions with a single Activity

```
<activity android:name="A" android:launchMode="singleTask">  
  <intent-filter>...</intent-filter>  
</activity>
```



How NDEF is delivered

Receiving Pro Tips: Multiple NFC interactions with multiple activities

- What if you have a flow of activities that have the same NFC intent filter?
 - Would popup the Activity Chooser
- Use `launchMode="singleTask"` for the entry activity,
- Use `launchMode="singleTop"` for subsequent activities,
- Use foreground dispatch





The new NFC API in J

Incredibly simple media sharing

The new NFC API in J

Fast media sharing

It's **hard** to get a file from one device onto another...

Right?



The new NFC API in J

Fast media sharing

```
protected void onCreate() {  
    NfcAdapter adapter = NfcAdapter.getDefaultAdapter(this);  
  
    // Uri for a file  
    File myFile = ...;  
    Uri fileUri = Uri.fromFile(myFile);  
  
    // Uri for a content provider  
    Uri contentUri = Uri.parse("content://com.example.apps.myapp/content/1");  
  
    // Set the Uri to be Beamed  
    adapter.setBeamPushUris(new Uri[] {fileUri}, this);  
}
```



The new NFC API in J

Fast media sharing

WAIT...

Did I just show you how to send a file to another device
in 5 lines of code?



The new NFC API in J

How it works

Device A

Sending Application

`setBeamPushUris(Uri[]);`

NFC Service

Mac Address **NFC**

Device B

Receiving Application

`startActivity() with ACTION_VIEW Intent`

NFC Service

Stores binary data in file

Bluetooth

Binary data



The new NFC API in J

Fast media sharing

- You get all the UI and Bluetooth handling for free
- You can use it in combination with the NDEF API
 - Uri is tried first





Wrapping up

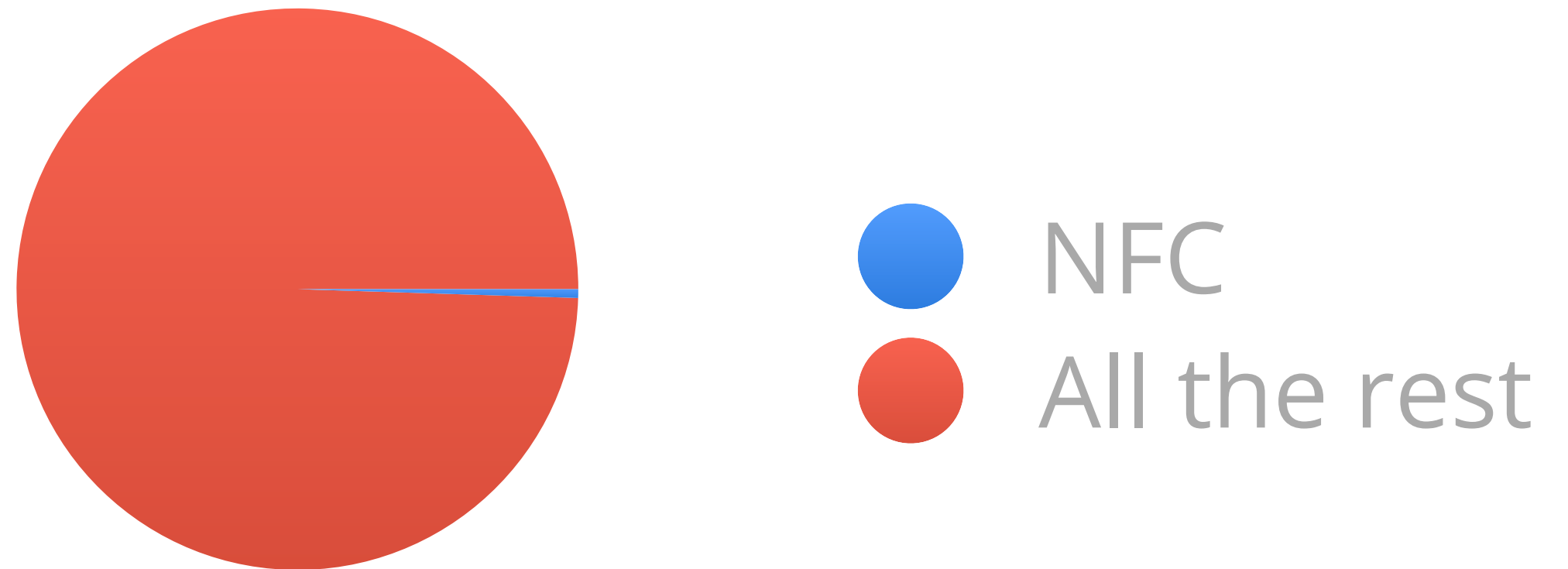
Things we're thinking about

- Support for two-way communication in Android Beam
- Customizing the Android Beam animation
- Extending the Beam media-sharing API to use WiFi-direct



By the way...please don't turn NFC off

Power consumption (screen on, unlocked)

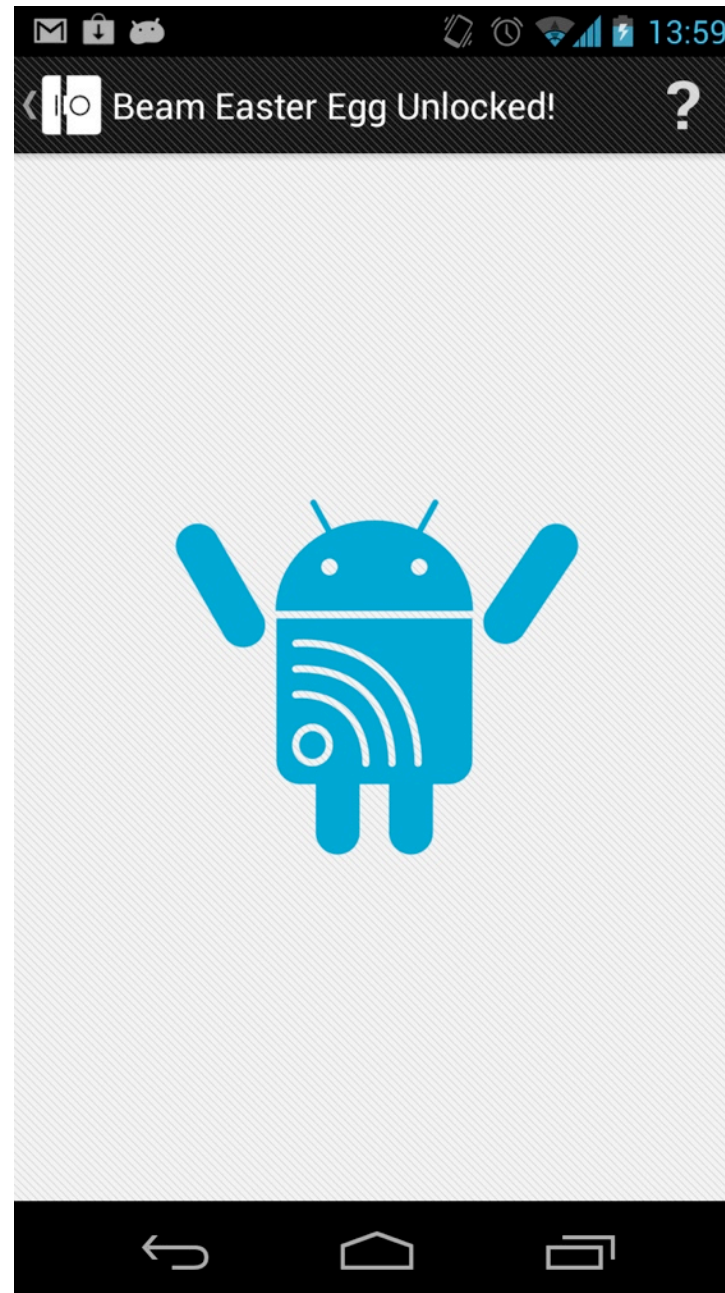


Summary

- NFC is here to stay
- Adding NFC to your apps is easy and adds value
- The new J API is awesome!



Recognize this?



Q&A



Thanks, and happy Beaming!



rhamilton@google.com

maco@google.com



Google
Developers