The State of Android
Near Field Communication 2010

Jonas Estberger
Bespoke Code
Introduction

Scope
With Android version 2.3 (Gingerbread) Google introduced Near Field Communication (NFC) support. This report is intended to give the reader a understanding of current situation and what might be expected within the near future regarding Android and NFC.

Terminology
NFC – Near Field Communication
API – Application Programming Interface
NDEF - NFC Data Exchange Format

Hardware Support
As of now there is only one Android phone on the market that supports NFC and that is Google's own Nexus S.

Nexus S
The phone is manufactured by Samsung and is very similar to the Samsung Galaxy S. This leads to strong speculations that Samsung will include NFC in their future products if there is market incentive.

There has been speculations about the write capability of the Nexus S hardware and that a later release might add support for writing to tags as well.

Software Support
Google released Android version 2.3, called Gingerbread, in December 2010. The NFC API only supports reading of tags, not writing.

The API states support for NFC Data Exchange Format (NDEF) messages encoded according to NFC Forum Type 2 specifications [1].

The NFC scanning is user-optionally always on. Any application may listen for an ACTION_TAG_DISCOVERED intent to take actions once a tag has been read.

This means that the user does not need to start any specific application before physically scanning a tag. This is a major improvement over QR-codes that need a special application to run in the foreground.

Below are two tables that describes the currently supported NDEF messages [2].

<table>
<thead>
<tr>
<th>Record Type Definitions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTD_ALTERNATIVE_CARRIER</td>
<td>RTD Alternative Carrier type.</td>
</tr>
<tr>
<td>RTD_HANDOVER_CARRIER</td>
<td>RTD Handover Carrier type.</td>
</tr>
</tbody>
</table>

www.bespokecode.se jonas.estberger@bespokecode.se
<table>
<thead>
<tr>
<th>Type Name Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TNF_ABSOLUTE_URI</td>
<td>Indicates the type field contains a value that follows the absolute-URI BNF construct defined by RFC 3986.</td>
</tr>
<tr>
<td>TNF_EMPTY</td>
<td>Indicates no type, id, or payload is associated with this NDEF Record.</td>
</tr>
<tr>
<td>TNF_EXTERNAL_TYPE</td>
<td>Indicates the type field contains a value that follows the RTD external name specification.</td>
</tr>
<tr>
<td>TNF_MIME_MEDIA</td>
<td>Indicates the type field contains a value that follows the media-type BNF construct defined by RFC 2046.</td>
</tr>
<tr>
<td>TNF_UNCHANGED</td>
<td>Indicates the payload is an intermediate or final chunk of a chunked NDEF Record.</td>
</tr>
<tr>
<td>TNF_UNKNOWN</td>
<td>Indicates the payload type is unknown.</td>
</tr>
<tr>
<td>TNF_WELL_KNOWN</td>
<td>Indicates the type field uses the RTD type name format.</td>
</tr>
</tbody>
</table>

When looking into the framework source code we come across the following declarations that might provide the supported types of cards. Please note that this information is not passed to applications but only shows up in logs from the lower JNI layer. [4]

```
/* Name strings for target types */
#define TARGET_TYPE_ISO14443_3A     "Iso14443-3A"
#define TARGET_TYPE_ISO14443_3B     "Iso14443-3B"
#define TARGET_TYPE_ISO14443_4      "Iso14443-4"
#define TARGET_TYPE_ISO15693        "Iso15693"
#define TARGET_TYPE_MIFARE_UL       "MifareUL"
#define TARGET_TYPE_MIFARE_1K       "Mifare1K"
#define TARGET_TYPE_MIFARE_4K       "Mifare4K"
#define TARGET_TYPE_MIFARE_DESFIRE  "MifareDESFIRE"
#define TARGET_TYPE_MIFARE_UNKNOWN  "Unknown Mifare"
#define TARGET_TYPE_FELICA          "Felica"
#define TARGET_TYPE_JEWEL           "Jewel"
#define TARGET_TYPE_UNKNOWN         "Unknown Type"
```

Further investigation finds three different modes of communication. P2P, Tag and LLCP. Currently only Tag is documented. [4]

www.bespokecode.se  jonas.estberger@bespokecode.se
Use Cases

**Business Card**

Scanning a VCARD automatically brings up the *Tags* application included with Android 2.3 as shown to the left in the picture below. Tapping the “Import ...” brings up the *Contacts* application and let the user make further actions.
**Action Tag**

Scanning the tag should start some sort of action. Here is an example of a tag containing sms-action, phone number and text.

Please note that the text is not correctly parsed as message text but is still in the to-field.

**The Future**

If there is enough incentive Android might get tag write/modify capabilities. [3]

There is more functionality than is presently documented. P2P and LLCP communication is very likely to get introduced during 2011.

**References**