Cross-platform NFC Tag UX

Application Document

Version 1.1

2022-12-06

[CPUX]

NFC Forum™

Contents

Figures

Tables

1 Introduction

NFC Forum Tags can be used in many applications, such as providing information to a mobile device or triggering an action. To ensure the interoperability of NFC Forum devices the NFC Forum has defined specific types of NFC Forum Tags. Data are stored on these tags using a common data format – the NFC Data Exchange Format (NDEF). Using NDEF, NFC Forum Tags can store data such as URL addresses, Wi-Fi or Bluetooth pairing information, vCard, iCal and others, including custom data. The data on the NFC Forum Tag can either be used directly by the mobile operating system or can be forwarded by the mobile operating system to applications that have registered support for the corresponding data type. A frequent example is an URI that can be opened in the mobile browser on the device.

Mobile operating systems such as Android or iOS provide different support for acting on data types that are read from NFC Forum Tags and provide different mechanisms for handling NDEF data. These differences make it difficult for developers that are developing applications for multiple mobile operating systems to select the features that work across the targeted platforms and still achieve a common user experience.

This Cross Platform NFC Tag UX Application Document assists developers in overcoming some of these issues.

1.1 Objectives

This document provides recommendations for service developers who want to achieve a similar user experience on different mobile operating systems when they use NFC Forum Tags in their applications. The recommendations are based on the corresponding functionality and user experience of the Android and iOS mobile operating systems.

1.2 Scope

The recommendations take into account the current versions of these mobile operating systems at the time of writing. The current version of Android is Android 11; however, the mechanisms described in this document apply also to earlier versions, starting from Android 6. The current version of iOS is version 14; however, NFC tag reading has been available since iOS 11 (‘Core NFC’ framework), Background Tag Reading was introduced in iOS 12, and AppClips in iOS 14.

1.3 Audience

The targeted audience for this document is software developers and service providers who want to use NFC Tags for their applications and services on multiple mobile operating systems.

1.4 Applicable Documents or References

[AndroidNFCDev] https://developer.android.com/guide/topics/connectivity/nfc

[AndroidAppLinks] https://developer.android.com/training/app-links/verify-site-associations

[AndroidDeepLinks] https://developer.android.com/training/app-links/deep-linking

[CH]	Connection Handover (CH) Technical Specification, NFC Forum
[iOSAppClips]	https://developer.apple.com/documentation/app_clips
[iOS_BTR]	https://developer.apple.com/documentation/corenfc/adding_support_for_background_tag_reading
[iOSCoreNFC]	https://developer.apple.com/documentation/corenfc
[iOSCustomSchemes]	https://developer.apple.com/documentation/xcode/allowing_apps_and_websites_to_link_to_your_content/defining_a_custom_url_scheme_for_your_app
[iOSUniversalLinks]	https://developer.apple.com/ios/universal-links/
[NDEF]	NFC Data Exchange Format (NDEF) Technical Specification, NFC Forum
[NFCSpecs]	NFC Forum specification overview https://nfc-forum.org/our-work/specification-releases/specifications/
[RFC 2046]	N. Freed, N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types" RFC 2046, Innosoft, First Virtual, November 1996.
[RFC3986]	Uniform Resource Identifier (URI): Generic Syntax, RFC 3986, T. Berners-Lee, R. Fielding, L. Masinter, March 1997, Internet Engineering Task Force
[RTD]	Record Type Definition (RTD) Technical Specification, NFC Forum
[RTD_URI]	URI Record Type Definition Technical Specification, NFC Forum
[REG]	NFC Forum assigned-numbers-register https://nfc-forum.org/our-work/specification-releases/specifications/nfc-forum-assigned-numbers-register/

1.5 Administration

The NFC Forum Cross-platform NFC Tag UX Application Document is supported by the Near Field Communication Forum, Inc., located at:

401 Edgewater Place, Suite 600
Wakefield, MA, 01880

Tel.: +1 781-876-8955
Fax: +1 781-610-9864

https://www.nfc-forum.org/

The NFC Forum, Inc. maintains this Application Document.

1.6 Trademark and Logo Usage

The Near Field Communication Forum's policy regarding the use of trademarks and logos is described in the NFC Forum Brand Identity Guidelines and N-Mark Usage Guidelines, which can be found on the NFC Forum website.

1.7 Intellectual Property

The Cross-platform NFC Tag UX Application Document has not been submitted to an IPR Election pursuant to the NFC Forum IPR Policy, and therefore may contain elements that are subject to intellectual property rights of third parties. This document is therefore PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY, COMPLETENESS AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.

Without limiting the statement above, NFC FORUM ASSUMES NO RESPONSIBILITY TO COMPILE, CONFIRM, UPDATE OR MAKE PUBLIC ANY THIRD PARTY ASSERTIONS OF PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS THAT MIGHT NOW OR IN THE FUTURE BE INFRINGED BY AN IMPLEMENTATION OR USE OF THIS APPLICATION DOCUMENT IN ITS CURRENT, OR IN ANY FUTURE FORM. IF ANY SUCH RIGHTS ARE DESCRIBED IN THIS APPLICATION DOCUMENT, NFC FORUM TAKES NO POSITION AS TO THE VALIDITY OR INVALIDITY OF SUCH ASSERTIONS, OR THAT ALL SUCH ASSERTIONS THAT HAVE OR MAY BE MADE ARE SO LISTED.

1.8 Abbreviations

Table 1 contains the definitions of the abbreviations and acronyms used in this document.

Table 1: Abbreviations

Abbreviation	Description
AAR	Android Application Record
API	Application Programming Interface
MIME	Multipurpose Internet Mail Extensions
NDEF	NFC Data Exchange Format [NDEF]
NFC	Near Field Communication
OEM	Original Equipment Manufacturer
RTD	Record Type Definition [RTD]
TNF	Type Name Format
URI	Uniform Resource Identifier [RFC3986]
UX	User eXperience

1.9 Glossary

NDEF payload

The application data carried in an NDEF record.

NDEF payload type

An identifier that indicates the type of the NDEF payload.

NDEF record

An NDEF record contains a payload described by a type, a length, and an optional identifier.

NFC Forum Tag

A contactless tag or (smart) card supporting NDEF.

2 Introduction to NFC Forum Tags

Tags are integrated circuits that store data and that can be read by NFC-enabled devices. By standardizing tag types and formats, the NFC Forum is promoting interoperability across the NFC market. The NFC Forum has specified five different tag types ('NFC Forum Tags'), of which four are currently mandated to be supported by NFC Forum devices (Tag Types 2, 3, 4 and 5).

Figure 1: NFC Forum Devices and NFC Forum Tags

To abstract from the different technologies and command sets of the different tag types, NFC Forum has defined the NFC Data Exchange Format (NDEF), which is used to store data on all NFC Forum tag types. With support from mobile operating systems, developers can read and write NDEF from or to a tag without caring about the specific technology used by the tag.

2.1 NDEF

NDEF is a binary data format that is supported by all NFC Forum Tags. The basic NDEF structure is the NDEF message, which is composed of one or more NDEF records. Each NDEF record has a header that contains metadata about the NDEF record and a payload that contains the content of the record. The metadata in the header supports the interpretation of the payload and contains information that includes the length of the NDEF record and the type of the data in the payload.

NDEF message boundaries are determined by marker bits in the NDEF record header that indicate whether an NDEF record is the first or last record in the NDEF message (an NDEF message does not have its own header field).

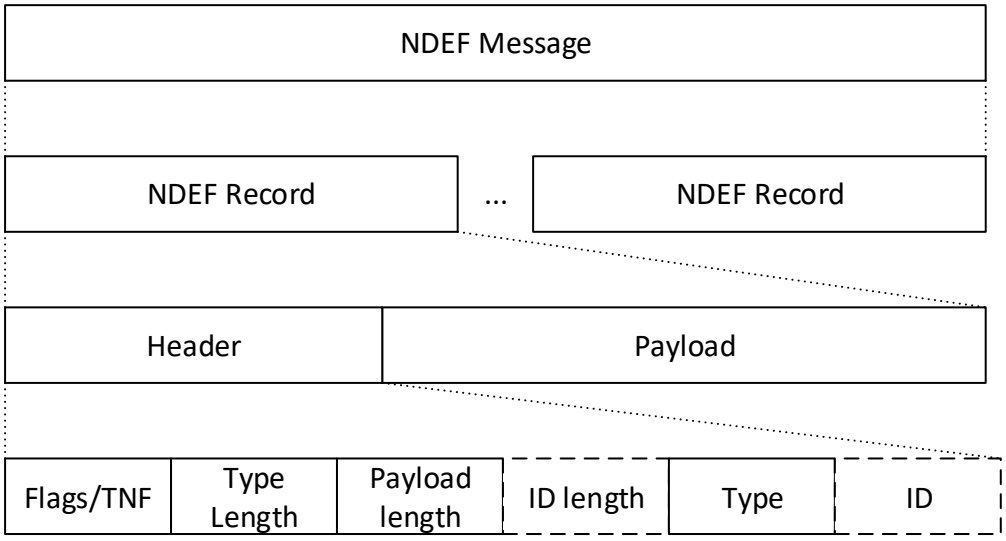

Figure 2: NDEF structure

The payload of an NDEF record can be one of several different data types. The data type of each record is identified in its NDEF record header. NDEF supports different type formats (identified by the TNF field in the NDEF record header), including MIME types and URIs, but also well-known data types that are defined by the NFC Forum.

NFC Forum Tags contain a single NDEF message. NFC Forum Tags might be read-only, or they might allow writing a new NDEF message.

Mobile platforms that support NFC provide APIs to create and parse NDEF messages. They also provide ways to read and write NDEF messages from or to NFC Forum Tags.

More information is available in the NDEF specification (see [NDEF]). The type format for NFC Forum well-known types is defined in [RTD]. NDEF records using well-known types are defined in separate specifications. For example, the NDEF record for storing an URI is defined in [RTD_URI] and uses the well-known type ‘U’. A list of all well-known types is available as part of the NFC Forum assigned numbers register [REG].

3 Basic Differences in User Experience

When they touch an NFC Forum Tag, mobile platforms can automatically detect whether the tag contains an NDEF message and, if one exists, can automatically read the message. Depending on the content of the message, a mobile platform will either handle the tag itself or will start an application to handle the message.

A mobile platform might first start an application that handles reading and writing from or to a discovered NFC Forum Tag. The user experience depends somewhat on the application.

Although the user experience is generally similar over different mobile platforms, there can be notable differences:

- In iOS the NFC reader functionality cannot be disabled by the user, whereas in Android it can be switched off in the settings. Different Android OEMs might choose different default values for the corresponding setting. Therefore, service developers are encouraged to check the current setting and ask the user to enable NFC in case it is not active (the current status can be checked using the isEnabled() method of the NfcAdapter class). If the NFC function is disabled, it will improve the user experience if the app explains to the user why it is now necessary to enable the NFC function and then guides the user to the right settings menu to enable the NFC function (by using the intent ACTION_NFC_SETTINGS). The app can also improve the user experience if it listens with the ACTION_ADAPTER_STATE_CHANGED broadcast receiver for the event when the user has enabled the NFC function and then returns automatically back to the app.
- Android always provides some sort of user feedback when a tag is detected (for example, vibrate). iOS only provides feedback if it supports the content on the tag. If there is no feedback, it is difficult for the user to distinguish whether there was a problem reading a tag (which could potentially be solved simply by touching again), or whether the tag read was successful but the content of the tag is not supported.
- iOS always shows a notification that the user needs to tap before taking any action based on an NDEF message. Android only shows a dialog if there is an ambiguity about which application is meant to handle the message (see the descriptions below related to ‘application selection dialog’).
- Android processes NFC Forum Tags only when the screen is unlocked. But iOS can perform background tag reading while showing the lock screen; if the tag content allows an action, the corresponding notification is shown on the lock screen.
- When it comes to automatic processing of NDEF messages by the platform, Android has wider support than iOS for the NDEF record types that it can handle. For example, Android natively supports static connection handover, as defined in [CH]. As a result, using Android, a user can establish a Bluetooth connection or join a Wi-Fi network by touching an NFC Forum Tag. Android also provides support for starting an application based on an NDEF record with a MIME type (such as an NDEF record that contains a vCard). The corresponding NFC Forum Tags will be ignored by iOS.

4 Use Case: Application Start

4.1 Use Case Description

NFC Forum Tags, each containing an NDEF message, are deployed to start an application on a mobile device. After the mobile device touches the tag, the application is started automatically. Ideally, this happens without requiring any additional user interaction, before or after the touch, until the application has started. Depending on the NDEF message on the NFC Forum Tag, the application can be started in a specific state.

If the relevant application is not installed, the corresponding page in the application store is opened to allow the user to install a new copy.

4.2 Android Mechanisms

Android supports multiple methods of starting an application; the method to be employed is determined by the NDEF message that the Android device reads after it touches the NFC Forum Tag.

In the Android procedure ‘intents’ have been set up according to the NDEF content on the NFC Forum Tag. Applications, or, more precisely, activities, can register for specific intents by using intent filters. If only one application has registered to handle an intent, that application is invoked automatically when the device touches the tag. In that case the user does not have to specifically select the application to use.

On the other hand, if more than one application is available to handle the intent, Android displays a dialog in which the user needs to select one application among the ones that can handle the intent. This procedure is called the ‘application selection dialog’. The dialog also allows the user to choose whether the selected application will become the new default application for the corresponding content, thereby avoiding future invocations of the dialog for that intent.

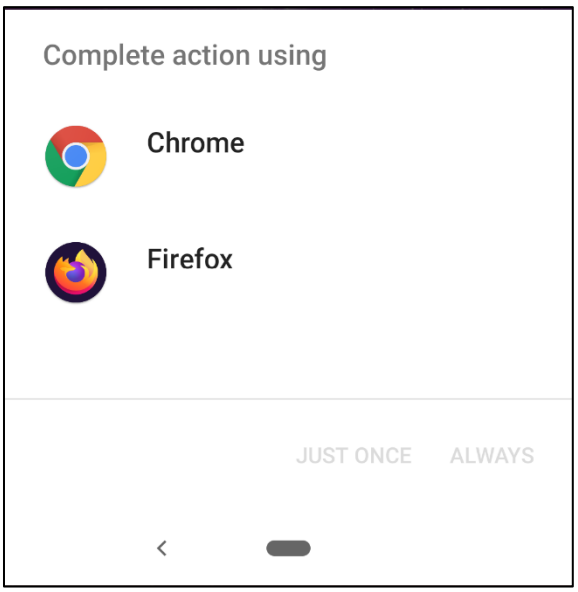

Figure 3: Example of an Application Selection Dialog

Android defines three intents that can be created based on the reading of the NDEF message from an NFC Forum Tag: ACTION_NDEF_DISCOVERED, ACTION_TECH_DISCOVERED and ACTION_TAG_DISCOVERED.

This section focuses on the ACTION_NDEF_DISCOVERED intent. The other two intents are less specific, and it is therefore more likely that multiple applications will be available to handle these intents. In these cases, it is more likely that a user interaction will be needed for the selection of the application. More information about the described mechanisms can be found at [AndroidNFCDev].

In general, applications registering for NFC events need to define intent filters that are as specific as possible for the targeted NFC application. Such filters avoid both unnecessary filter operation at the called activity to determine the specific NFC tags that the application is interested in, as well as unnecessary conflicts with other NFC applications. They also reduce the invocation of application selection dialogs and improve the end user experience.

Figure 4 shows how some example intent filters are set up for different use cases.

```
<!-- Intent filter to listen only for NFC Forum tags providing an URL of
https://nfc-forum-org -->
<intent-filter>
    <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="https" android:host="nfc-forum.org"/>
</intent-filter>

<!-- Intent filter to listen only for NFC Forum tags providing the URL
https://nfc-forum-org/join/ -->
<intent-filter>
    <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="https" android:host="nfc-forum.org"
     android:path="/join/"/>
</intent-filter>

<!-- Intent filter to listen only for NFC Forum tags providing an URL
starting with https://nfc-forum-org/our-work/ -->
<intent-filter>
    <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="https" android:host="nfc-forum.org"
     android:pathPrefix="/our-work/"/>
</intent-filter>

<!-- Intent filter to listen only for NFC Forum tags providing an URL of
a pdf-document at https://nfc-forum-org -->
<intent-filter>
    <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="https" android:host="nfc-forum.org"
     android:pathPattern=".*.pdf" />
    <data android:scheme="https" android:host="nfc-forum.org"
     android:pathPattern=".*.PDF" />
    <data android:scheme="https" android:host="nfc-forum.org"
     android:pathPattern=".*.Pdf" />
</intent-filter>

<!-- Intent filter to listen only for NFC-Forum tags providing a phone
number to US -->
<intent-filter>
    <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="tel" android:sspPrefix="+1"/>
    <data android:scheme="tel" android:sspPrefix="001"/>
</intent-filter>

<!-- Intent filter to listen only for NFC Forum tags providing an e-mail
to @nfc-forum.org -->
<intent-filter>
    <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="mailto"
     android:sspPattern=".*@nfc-forum.org.*"/>
</intent-filter>
```

Figure 4: Example Intent Filters

4.2.1 MIME Type

An application can register a MIME type in the Android manifest by using an intent filter. To be application specific, the registration uses a top-level media type of ‘application’ and a sub-level media type that is specific for the application. Otherwise, and when there are multiple applications that can handle the media type, Android will show an application selection dialog to allow the user to select one of the available applications.

The NDEF message needs to start with an NDEF record whose TNF field value is 0x02 (Media-type, as defined in [RFC 2046]) and an NDEF payload type value that is equal to that of the selected MIME type. For this mechanism to work, it is important that this NDEF record be the first record in the NDEF message.

4.2.2 URIs / Deep Links

Registration of an URI is very similar to that for the MIME type: the application registers an intent filter for a specific URI. The NDEF message needs to start with one of the following:

- An URI NDEF record (an NDEF Record that uses the NFC Forum Well Known Type “U” defined in [RTD_URI]) with an NDEF payload type that is equal to the selected URI.
- A SmartPoster NDEF record (an NDEF Record that uses the NFC Forum Well Known Type “SP”) that contains an URI NDEF Record, as defined in [RTD_URI]. The NDEF payload of the URI NDEF record must be equal to the selected URI.
- An NDEF record whose TNF field value is 0x03 (Absolute URI, as defined in [RFC3986]) and an NDEF payload type that is equal to the selected URI. The payload itself can be application specific.
- An NDEF record that uses an NFC Forum External Type, as defined in [RTD] (TNF field value is 0x04). Android creates an URI based on the URN in the type field. The resulting UL has the following format:
  `vnd.android.nfc://ext/<domain_name>:<service_name>`.

In practice, the first two options are the most relevant.

Android will show an application selection dialog if there is more than one application that can handle the URI scheme.

Android also supports the use of custom URI schemes. If a custom scheme is employed, it needs to be assigned a unique name, to avoid an application selection dialog with different applications of which only one single application works correctly with the given custom URI.

4.2.3 Android Application Record

The Android Application Record (AAR) is a specific NDEF record that contains the package name of the application to be started. The AAR can be located at any position in the NDEF message, since Android searches the whole message to find an AAR.

One notable difference from intent filters is that the AAR alone cannot start a specific activity, since the package name is bound to an application and not a specific activity.

NOTE Section 4.4.1 contains an option that achieves this by combining the AAR with an URI record.

If no application with the specified package name is installed, Android will open the application store to allow the user to download the corresponding app.

4.2.4 App Links

Android App Links allow an app to register as the default handler for a given link (see [AndroidAppLinks]). App Links are available in Android 6.0 or later. The main purpose of this approach is to avoid any dialog in which the user needs to select an application to handle the link.

The link must be an HTTP or HTTPS link pointing to a domain to which the application owner has access. Upon installation of the application, Android will verify that the application is entitled to be the default handler for the website by checking a specific file on the website (the 'Digital Asset Links' file). This file contains information on the applications that are allowed to use this mechanism.

App Links cannot be used directly in intent filters related to NFC Forum Tags, as they require an intent filter with the `android.intent.action.VIEW` action (they do not apply to intent filters with the `ACTION_NDEF_DISCOVERED` action).

To be usable for NFC Forum Tags, the application must:

- Include an App Link that establishes the application as the default handler for the specified URL pattern, as defined in [AndroidAppLinks].
- Include corresponding `ACTION_NDEF_DISCOVERED` intent filters for the Activities that are to be started upon reading an NFC Forum Tag.

The NDEF on the NFC Forum tag must be formatted using one of the three options described above in 4.2.2 for URIs.

After reading a corresponding NFC Forum Tag, Android will create the corresponding `ACTION_NDEF_DISCOVERED` intent. As the application is the default handler for the included link, Android will start the activity of that application that best matches the link. If there is no other application, Android will open the link in the default browser.

See Figure 5 for an example of intent filters in the Android manifest. This example contains a main activity that has the intent filter for the App Link, and two sub-activities that can be started via NFC.

```
<activity android:name=".MainActivity" android:launchMode="singleTask">
(…)
  <intent-filter android:label="testAppLink" android:autoVerify="true">
    <action android:name="android.intent.action.VIEW"/>
      <category android:name="android.intent.category.DEFAULT" />
      <category android:name="android.intent.category.BROWSABLE"/>
      <data
              android:scheme="https"
              android:host="yourdomain.com" />
   </intent-filter>
(…)
</activity>

<activity android:name=".SubActivity2" android:launchMode="singleTask">
(…)
  <intent-filter android:label="testAppLinkNFC">
    <action android:name="android.nfc.action.NDEF_DISCOVERED" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="https"
              android:host="yourdomain.com"
              android:pathPrefix="/target2" />
    </intent-filter>
(…)
</activity>

<activity android:name=".SubActivity1" android:launchMode="singleTask">
(…)
  <intent-filter android:label="testAppLinkNFC">
    <action android:name="android.nfc.action.NDEF_DISCOVERED" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="https"
              android:host="yourdomain.com"
              android:pathPrefix="/target1" />
  </intent-filter>
(…)
</activity>
```

Figure 5: Example Intent Filters for Main Activity and Sub-activities

If there is no application to handle the link, Android will open the link in a browser.

4.3 iOS Mechanisms

The mechanism in iOS to start an application based on touching an NFC Forum Tag is called ‘Background Tag Reading’. This section provides an overview of the mechanism.

4.3.1 Background Tag Reading

iOS scans in the background and, if an NFC Forum Tag is discovered, it reads the NDEF message. iOS then parses the NDEF message for an URI NDEF record. This is an NDEF Record that uses the NFC Forum Well Known Type “U”, as defined in [RTD_URI]. The system processes the first URI NDEF record it detects. This procedure is described in more detail in [iOS_BTR].

The payload of the URI NDEF record is an URI that the system processes either by opening the application that supports the URI scheme (e.g. an URI starting with ‘mail://’ will be handled by the default email application) or by directly opening a specific application if the URI has been registered by that application as an ‘Universal Link’.

The following sections describe the different options for the ways the URI can be processed by iOS.

Background tag reading is supported on iPhone XS and later.

Earlier phone models, such as iPhone 7, iPhone 7 Plus, iPhone 8, iPhone 8 Plus and iPhone X, can, instead, use the NFC Tag Reader. The NFC Tag Reader can be started via the Control Center after it has been added to the Control Center in the Settings app. Once it is added, a tap on the NFC Tag Reader button in the Control Center starts a search for NFC Forum Tags and shows a corresponding message. When an NFC Forum Tag is found, the content processing is the same as with background tag reading.

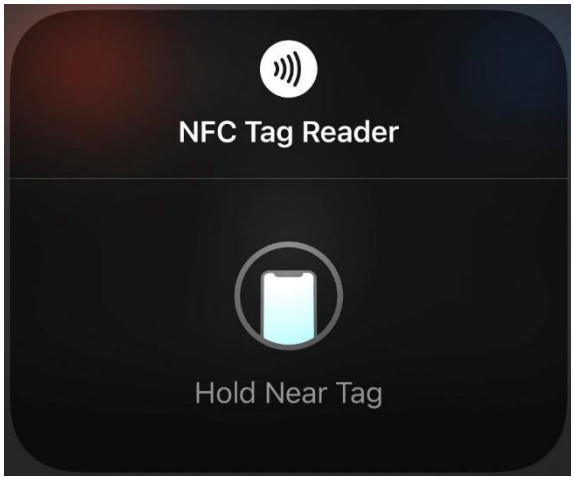

Figure 6: NFC Tag Reader

4.3.2 Universal Links

The payload in the URI record of the read NDEF message can be registered by an application as a Universal Link.

Universal Links are standard HTTP or HTTPS links that can be used to connect to content inside an application. If the application is not installed, iOS opens the URL in the default browser, allowing the corresponding website to handle the processing. An application can navigate to a specific view by parsing the Universal Link and redirecting the user to the corresponding view.

A Universal Link is a link that is registered by an application after iOS has verified that the application both is associated with the corresponding domain and has the right to handle the link. iOS accomplishes this verification by checking a specific file in the corresponding domain (the Apple-App-Site-Association file – AASA). More details on this process can be found at [iOSUniversalLinks].

4.3.3 App Clips

App Clips were introduced in iOS 14. An App Clip is a lightweight version of an application that offers a small subset of the functionality of the ‘full’ application. App Clips do not need installation. Instead they can be started, for example, by touching a banner in the browser, by a QR code, or via NFC.

For NFC initiation the mechanism works the same as described for the Universal Links above. In fact, the same URI can be used to start an App Clip and, after the user has installed it, the full application. The main difference is that the App Clip must be registered in the Apple-App-Site-Association file.

URIs can be configured as ‘advanced App Clip experiences’ in App Store Connect. Each advanced App Clip experience is associated with a URL that can be stored on the NFC Forum Tag. Such URLs are called Invocation Links. More information about developing App Clips can be found in [iOSAppClips].

4.3.4 Custom URL Schemes

iOS also allows applications to register for custom URL schemes, as described in [iOSCustomSchemes]. Even though [iOS_BTR] mentions that background tag reading does not support custom URL schemes, current iOS versions actually allow applications to be started by using a custom URL scheme read from an NFC Forum Tag.

Because it is possible for different apps to declare the same URL Scheme, this mechanism can potentially be used by malicious apps for a number of different attack scenarios, such as hijacking sensitive data from other apps. For this reason, Apple strongly recommends employing Universal Links instead of custom URL schemes.

If custom URL schemes are used, they can be handled only if the corresponding application is already installed. If not, iOS only displays a notification that no usable data were found on the NFC Forum Tag. In this case a forward to the application store cannot be implemented.

If the application is installed and a link using the custom scheme is executed for the first time, the user needs to confirm in an additional dialog that the corresponding application is to be opened.

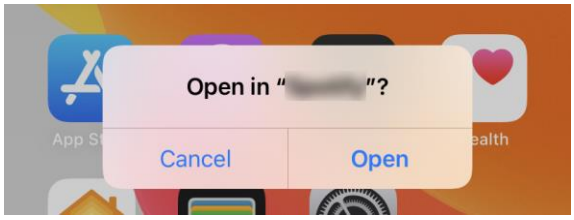

Figure 7: iOS Confirmation Dialog using Custom Schemes

4.4 Recommendations

In comparison to the mechanisms described in the previous sections, this section outlines options that provide similar user experiences for starting an application on both iOS and Android.

But one difference in user experience between iOS and Android that cannot be overcome is that in iOS a notification is shown after touching the NFC Forum Tag and the user needs to tap on this notification to start the application. In Android an application selection dialog is shown only when there are multiple applications that can handle the NDEF content; otherwise the application is started directly.

4.4.1 Option 1: URI and AAR Records

This option uses an NFC Forum Tag with an URI NDEF record as the first record in the NDEF message. The URI NDEF record contains an 'http' or 'https' link. In addition, the NDEF message contains an Android Application Record (AAR) as one of the following NDEF records.

The link in the URI NDEF record can be registered as a Deep Link by the Android application and as a Universal Link or Invocation Link by the iOS application.

If the application is already installed:

- iOS will show a notification that the user needs to tap to start the application that has registered the link.
- Android opens the application that is specified in the AAR without any additional dialog. Since the AAR takes precedence over the link in the URI NDEF record, this applies also when there are other applications installed that could handle the link. The link is evaluated as well and can determine which Activity is started from the application specified in the AAR.

  From a pure user experience point of view, the AAR can be seen as a replacement for registering the link as an App Link, as both avoid the application selection dialog.

If the application is not installed:

- iOS opens the corresponding website, which can redirect the user to the application store, as described for option 2 in the next section.
- Android will, based on the information in the AAR, open the corresponding entry in the Play Store to allow the user to install the application.

Option 2 described in the next section does not use an AAR; instead the link is registered as an App Link on Android. Comparing the two options:

- There is no difference for the processing on iOS.
- For Android, option 1 is simpler to implement as there is no need to manage the 'Digital Asset Links' file on the associated domain. But there is no link verification, so there might be ways for a malicious application to hijack the communication – however, this requires the installation of the malicious application with an identical package name to that of the original application.

4.4.2 Option 2: URI Record

This option relies solely on an URI NDEF record as the first record in the NDEF message. The URI NDEF record contains an ‘http’ or ‘https’ link. The link is registered as App Link by the Android application and as Universal Link or Invocation Link by the iOS application.

The difference from option 1 is that the processing on Android as an App Link is used instead of the AAR record.

If the application is installed:

- iOS shows a notification the user must tap in order to start the application that has registered the link.
- Android opens the application that registered the link without any additional dialog.

If the application is not installed, both systems will open the corresponding website in the default browser. The website is expected either to directly redirect the user to the corresponding application store or to offer a way to trigger this redirect, such as displaying corresponding banners.

One option to implement such a redirect is to use a client-side redirect based on JavaScript. The mobile operating system can be determined by the User-Agent request header of the http request.

A variant of this option is to register the link in the URI NDEF Record as an URI by the Android application (instead of an App Link) and as a Universal Link by the iOS application. However, this case leads to significant differences in user experience if multiple applications that can handle the URI are installed on an Android system, because Android will then show the application selection dialog.

5 Use Case: Opening an URI by a Default Application

In this case the NFC Forum Tag is deployed with an NDEF message that contains an URI. After the tag is touched by the mobile device, the default application for the URI scheme starts automatically and processes the URI. If the URI is an http link, the processing includes the browser loading the corresponding webpage. Ideally this happens without requiring additional user interaction (before or after the touch) until the webpage has fully loaded.

One main difference from the use case described in Section 3 is that this service is not employing its own application but relies on default applications that are preinstalled on the mobile phone.

5.1 Android

As described in 4.2.2, Android supports URIs on NFC Forum Tags (using an URI NDEF record, URI type or SmartPoster NDEF record). The NDEF record with the URI must be the first NDEF record in the NDEF message.

In Android the support for specific URI schemes depends on the default applications installed by the device manufacturer. However, the schemes listed in Table 2 are supported on all Android devices.

Table 2: Schemes Supported on all Android Devices

Scheme	Example
Website URL (HTTP/HTTPS)	https://www.example.com
Email	mailto:user@example.com
SMS	sms:+14085551212
Telephone	tel:+14085551212
Geolocation	geo:48.323922,008.967393
Navigation	google.navigation:?q=Address%0ACity%0ACountry

Website URLs either are forwarded directly to the browser or, if there are multiple applications installed that support such links, an application selection dialog is displayed. Similarly, an email URI is handled by the default email application.

For other URI schemes, such as telephone or SMS, Android opens an application that displays the possible actions (indicated under “New tag collected”). Figure 8 shows an example screenshot after an NDEF message is read with a telephone URI.

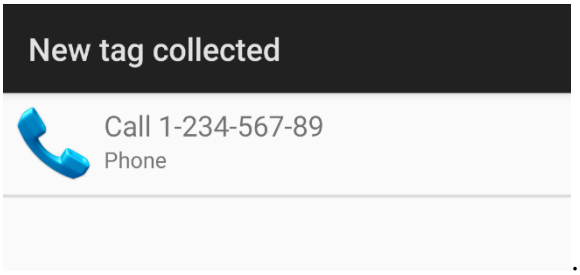

Figure 8: Example ‘New tag collected’ Action

For this type of action Android parses the full NDEF message and lists the possible actions for every NDEF record it finds. For example, if the NDEF message contains five URI NDEF records, the possible actions for all of them will be listed. URI NDEF records with content for which no action can be derived are shown as ‘Unknown tag type’. See Figure 9.

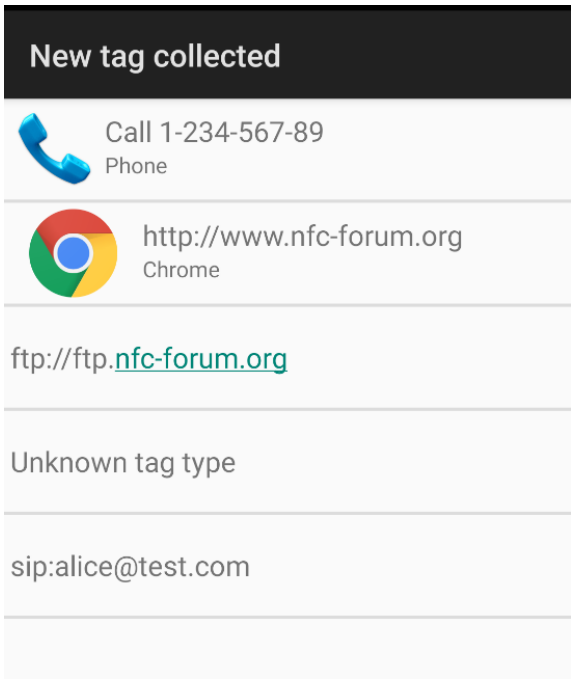

Figure 9: ‘New tag collected’ Action that has Multiple URI Records

As mentioned above, this procedure only applies if the first NDEF Record triggers the application. If for example the first URI NDEF record contains a website URL, this address will be automatically processed by the browser and no list will be shown. However, if the URI NDEF record that contains the website URL is the second NDEF record and the first is another URI NDEF record (for example containing a telephone scheme), then the list includes actions covering both records. Again, records that are not understood are listed as ‘Unknown tag type’.

5.2 iOS

iOS supports URIs on NFC Forum Tags encoded in an URI NDEF record (an NDEF Record that uses the NFC Forum Well Known Type “U”, as defined in [RTD_URI]). iOS will process the first URI NDEF record in the NDEF message.

Table 3 lists the URI schemes that are supported by iOS devices (as described in the site [iOS_BTR]).

Table 3: Supported URI Schemes

Scheme	Example
Website URL (HTTP/HTTPS)	https://www.example.com
Email	mailto:user@example.com
SMS	sms:+14085551212
Telephone	tel:+14085551212
FaceTime	facetime://user@example.com
FaceTime Audio	facetime-audio://user@example.com
HomeKit Accessory Setup	X-HM://12345

5.3 Recommendations

To enable support on Android and iOS the following rules need to be followed:

- The NDEF message contains an URI encoded in an URI NDEF record (that is, an NDEF Record that uses the NFC Forum Well Known Type “U”, as defined in [RTD_URI]).
- This URI NDEF record is the first record in the NDEF message.
- The URI uses a scheme that is supported by both platforms (for instance, http, https, mailto, sms or tel).

iOS will always show a notification that the user needs to tap before opening the default application. Android might show an application selection dialog when there are multiple applications installed that support the URI scheme.

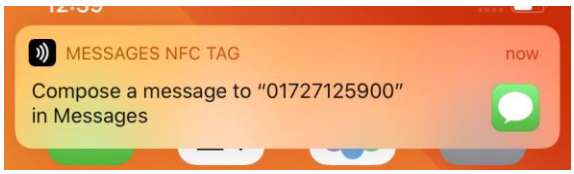

Figure 10: iOS Notification Example

When a link to a webpage (http or https) is used, both platforms open the default browser to process the link. However, the following difference in user experience can occur, depending on the mobile browser employed:

- If an NFC Forum Tag is touched multiple times, some mobile browsers will open a new tab only on the first touch and for all following touches will just reload the content in this tab.
- Other mobile browsers will open a new tab upon every touch.

Safari belongs to the first category, Chrome to the latter.

The behavior in other browsers can show other differences, depending on whether the URL contains parameters or not. In these browsers, if no URL parameters are present, the URL is usually opened in a new tab, whereas dynamic URLs trigger just a reload if a tab exists that was previously opened with the same URL.

6 Generic UX recommendations for Application SW developers

The way an application running on a mobile device is developed will impact the user experience with NFC Forum Tags and the adhesion to this technology.

The following recommendations apply regardless of the use case and will contribute to a good user experience

6.1 Short tap duration

6.1.1 Recommendation

The duration required for the NFC tap is intended to be as short as possible, to optimize the user experience. For most use cases and device configurations, it is best if a tap does not last longer than two seconds. If possible, a duration below one second is desirable (measured from the time the mobile device and the NFC Forum tag are brought close enough together to enable RF communication until the time the NFC transaction has finished).

6.1.2 Rationale

Even though some use cases or specific implementations might require the mobile device and the NFC Forum tag to be at contact for a longer time, the main scenario with NFC is a short tag between the two devices. A duration of two seconds is long enough for exchanging a reasonable amount of data for most use cases, even if the NFC Forum tag discovery is configured for a less frequent polling than for other devices.

6.2 Progress indicator

6.2.1 Recommendation

For data exchange longer than 1s, it is strongly advised that an identification mark showing the progress of the data exchange be provided (for example a progress bar or moving symbol). At the end of the exchange, the user needs to be informed to remove the mobile device from the NFC Forum tag.

6.2.2 Rationale

Without an indicator, the user does not know the status of the transfer and might remove the mobile device during the transfer process. Providing such feedback significantly increases the success rate of NFC communication, as the user can wait for feedback from the mobile device. Also, the user does not need to make a decision by himself to remove the mobile device from the NFC Forum tag but instead will be guided by the application running on the mobile device.

6.3 Avoid screen rotation

6.3.1 Recommendation

It is recommended that applications requesting the user to tap a NFC Forum tag avoid unintended screen rotation of the application during this period.

6.3.2 Rationale

Taping an NFC Forum tag with a mobile device requires a movement which might induce a rotation of the mobile device screen. To be able to properly read the screen, the user might try to move the device again which might interrupt the NFC data exchange.

6.4 Error management

6.4.1 Recommendation

It is essential that applications detect and report errors during NFC data exchanges. The application also needs to recommend an action by the user when an error is detected. It is especially important to inform the user whether it makes sense to retry the tap (for example in case of a communication error) or whether this would not resolve the issue (for example in case the read NDEF message contains erroneous or unexpected content).

6.4.2 Rationale

Providing information on errors during the data exchange might help the user to understand and fix what is going wrong. It helps to avoid restarting the same exchange multiple times with the same results and being confused.

6.5 Re-use of existing data formats and protocols

6.5.1 Recommendation

Open data formats already exist for a wide variety of use cases. Such open standard data formats can be used and implemented by anyone. The NDEF format allows re-use of existing data formats if they have an associated mime type or URI (see [NDEF]). In addition, the NFC Forum has developed own data formats for specific NFC use cases (so-called Record Type Definitions).

It is essential that an application re-use existing data formats whenever applicable. The same applies to existing protocols that are designed or can be used for NFC communication.

6.5.2 Rationale

Using existing data formats and protocols improves interoperability in the NFC ecosystem and the adoption of NFC technology. It also allows to re-use existing solutions for creating and consuming data and simplifies data exchange with other (non-NFC) applications.

Data formats and protocols defined by the NFC Forum can be found at [NFCSpecs]. Especially relevant in this context are the specifications in the category Record Type Definition Technical Specifications. For some use cases, the NFC Forum has also published application documents that often include concrete examples of the data to be exchanged (for example regarding connection handover to Bluetooth or Wi-Fi). Application documents can be also found at [NFCSpecs] in the corresponding category.

A. Exhibit A

No items have been included in Exhibit A.

B. Revision History

Table 4 outlines the revision history of the Cross-platform NFC Tag UX Application Document.

Table 4: Revision History

Document Name	Revision and Release Date	Status	Change Notice	Supersedes
Cross-platform NFC Tag UX	Version 1.0, April 2021	Final	Initial Release	-
Cross-platform NFC Tag UX Application Document	Version 1.1, December 2022	Final	Editorial update.	Version 1.0, April 2021