

# 2018

## Rapport de projet Géolocalisation par TAGs NFC



 UNIVERSITÉ  
PARIS  
SUD  
IUT DE CACHAN



Mehdi BASSOU  
Antoine GERGES-HANNA

# Sommaire

Objectifs du projet .....	2
Matériel à disposition.....	3
Tags utilisés .....	4
Changement de TAG .....	5
Création de l'IHM .....	5
Détection d'un TAG .....	7
Le NDEF .....	8
Intents .....	9
Construction du code .....	9
Fichier Manifest .....	9
Fichier MainActivity .....	10
Modification de l'icône .....	11

## Objectifs du projet

L'objectif du projet est de développer une application ANDROID permettant à un visiteur de s'orienter dans l'IUT. On dispose de TAG (étiquettes) NFC qui seront placés dans l'enceinte de l'établissement. Quand un visiteur passe son téléphone avec l'application d'ouverte, une flèche apparaît afin de le situer ainsi qu'une boîte de dialogue pour le notifier précisément du bâtiment dans lequel il se trouve.

## Matériel à disposition

Nous avons à notre disposition des TAGS NFC vierges de la série Mifare Classic 1k. Nous auront donc à écrire sur ces TAGS les infos qui nous intéressent.



Figure 1 : Etiquette ou TAG NFC

Afin d'écrire sur les TAGS, on utilise l'application mobile « NFC Tools » disponible sur le Play store.

Il nous est possible de Lire les informations d'un TAG, ainsi que d'écrire différents types d'informations

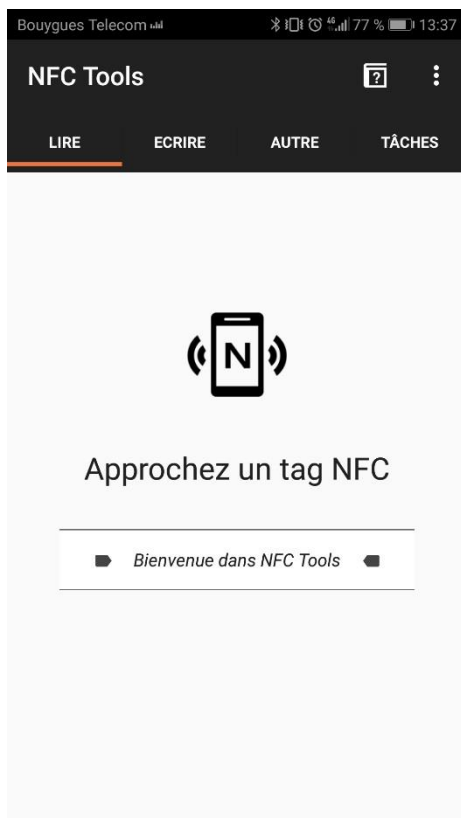


Figure 2 : Application NFC Tools (mode lecture)

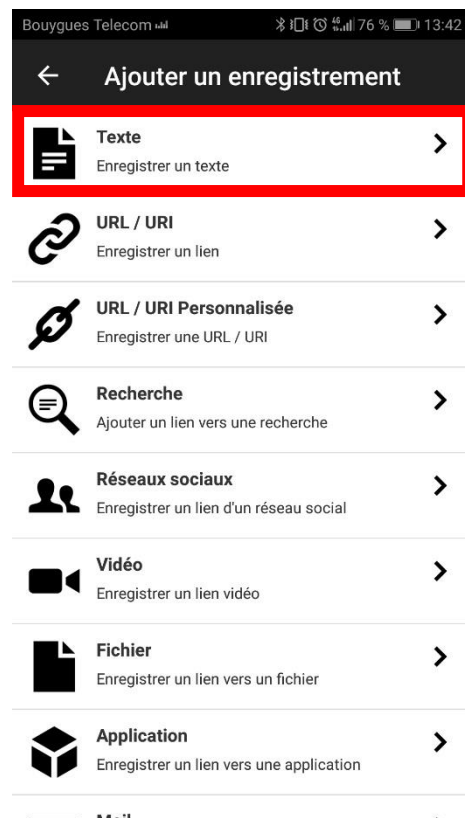


Figure 3 : Infos d'écriture

On va s'intéresser à l'écriture de texte sur les TAG. Nous allons tout d'abord écrire les infos de différents emplacements au sein de l'IUT (Entrée, Bâtiment G...). L'idée est que notre application détectera alors le texte inscrit et réagira en affichant la position de l'utilisateur.

## Types de TAG NFC



```
graph TD; A[Types de TAG NFC] --> B[Ultralight]; A --> C[Mifare Classic 1k]
```

### Ultralight

- Compatible avec la majorité des systèmes Android
- Pas cher
- Pensé pour l'inscription d'URL courtes et de texte
- Mémoire disponible : ~ 168 octets

### Mifare Classic 1k

- Compatible avec les systèmes Mifare mais avec peu d'appareils mobiles
- Supporte la cryptographie
- Mémoire disponible : ~ 716 octets

## Changement de TAG

Après avoir réalisé nos premiers tests de détection des tags, nous avons remarqué que les Tags de la série Mifare Classic 1k n'étaient pas compatibles avec la plupart des téléphones mobile. Nous avons donc décidé de changer de type de TAG avec ceux de la typologie *Ultralight*. En effet, ces TAGS sont compatibles avec la quasi-totalité des dispositifs Android.

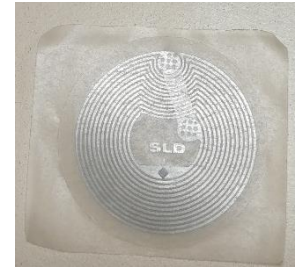


Figure 4 : TAG Ultralight

## Création de l'IHM

On utilisera le logiciel Android Studio pour développer l'IHM.



La 1ere version de l'application devrait se présenter comme ceci :



Figure 5 : 1ere version attendue

La flèche rouge sera initialement invisible, lorsque l'utilisateur placera le dos de son téléphone contre le TAG NFC situé à l'entrée de l'IUT, la flèche passera en mode visible. L'idée est alors la suivante : placer les TAGS à différents endroits de l'IUT ainsi que des flèches dans le plan. Il faudra donc associer chaque flèche à un TAG.

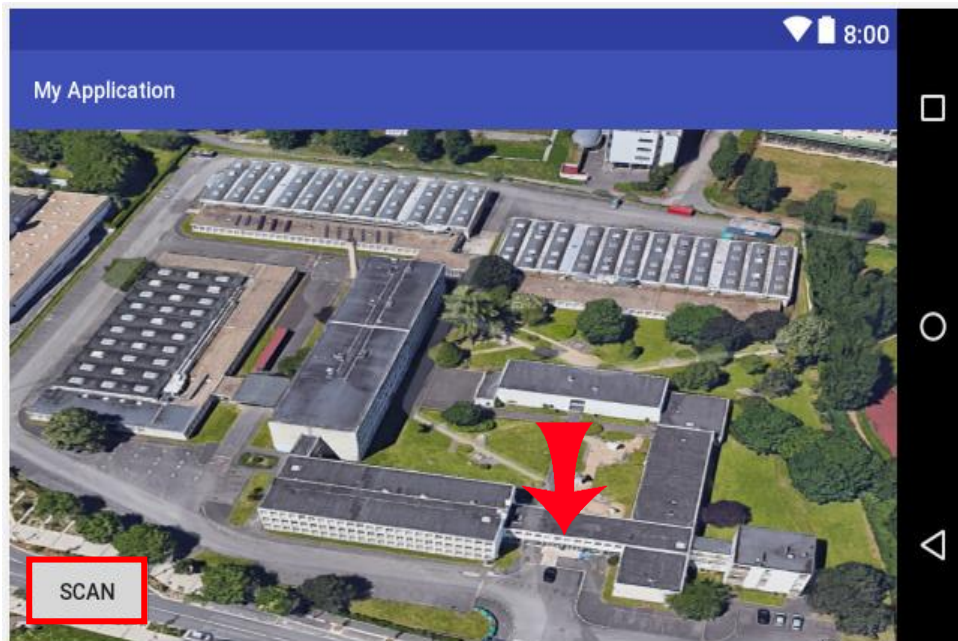


Figure 6 : 1ere Version de l'application



Figure 7 : 2eme version de l'application

## Détection d'un TAG

L'image ci-dessous nous montre comment le système Android réagit lors de la détection d'un TAG :

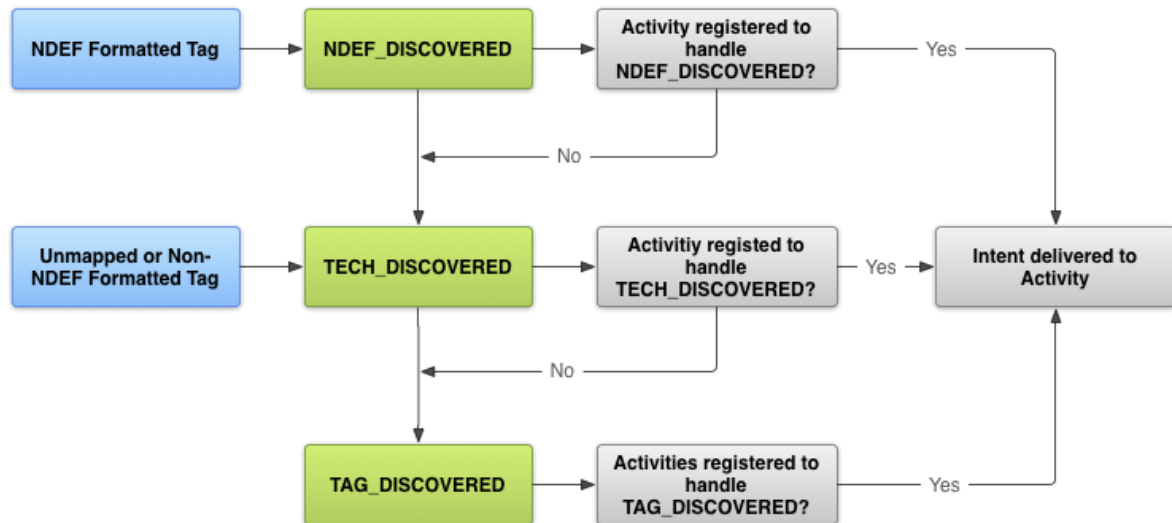


Figure 8 : Traitement d'un TAG NFC

Le traitement d'un TAG se fait en 3 étapes :

- Android essaye de lancer l'activité filtrant l'action NDEF\_DISCOVERED. Les activités qui filtrent cette action ont la plus haute priorité. Si le système détecte un tag avec le support NDEF, cela déclenche un Intent NDEF\_DISCOVERED. On rappelle qu'un Intent est un message permettant la communication entre les composants applicatifs. Il permet notamment de transmettre des informations d'une activité à une autre.
- Si le système ne trouve aucune activité pour l'Intent précédent, il essaie alors de démarrer les activités avec la deuxième plus haute priorité. Ces activités filtrent l'ACTION\_TECH\_DISCOVERED. Cet Intent se lance quand un tag est détecté et que des activités sont enregistrées pour la technologie spécifique présente dans le tag.
- Si aucune activité n'est trouvée dans le processus ci-dessus, l'Intent TAG\_DISCOVERED est enclenchée.



## Le NDEF

Le NDEF (NFC Data Exchange Format) est une norme de standardisation d'échanges entre 2 dispositifs NFC. L'idée est de lancer l'application lors de la détection d'un TAG NDEF.

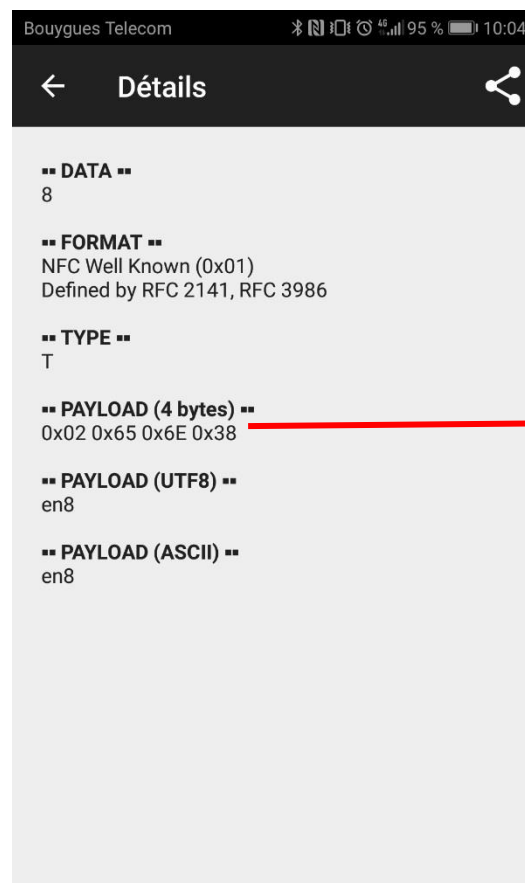
MB=0	ME=0	CF=0	SR=1	IL=0	TNF=01h
NFC Forum well-known type [NFC RTD]					
Type length = 01h					
Payload length = 10h (16 bytes)					
Type = "T"					
02h: UTF-8 with two-byte language code					
"en" ISO code for "English"					
"Hello, world!" UTF-8 string					

Figure 9 : Exemple enregistrement NDEF

On rappelle que nous écrivons le texte présent sur les tags avec l'application mobile **NFC Tools**. En écrivant sur les tags avec différentes applications, nous avons remarqué que nous nous retrouvions avec le même type de paquet de données. On en a donc conclu que ce type d'encodage était une norme (le « en » signifie english). Il restait donc à décoder le contenu du paquet afin de le traiter dans le code.



Figure 10 : Icône de l'application NFC Tools



- 0x02 : Start of Text
- 0x65 : e
- 0x6E : n
- 0x38 : 8

Figure 11 : Détail du texte du tag n°8

## Intents

Un *Intent* est un message asynchrone qui permet à des composants de l'application communiquer avec d'autres fonctions d'*Android*.

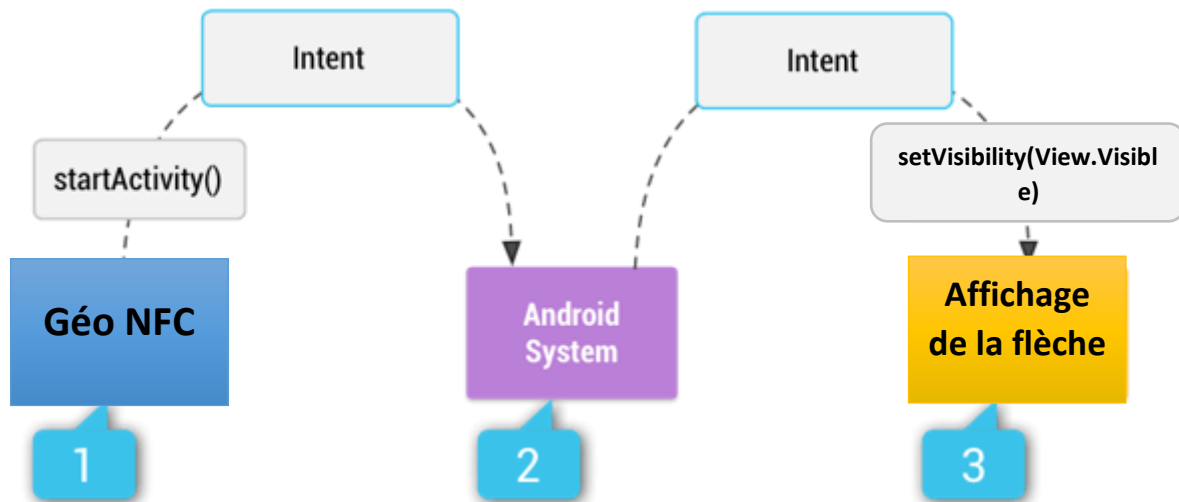


Figure 12 : Principe de fonctionnement d'un Intent

## Construction du code

### Fichier Manifest

Le fichier Manifest indique au système qu'un composant qui veut se lancer existe déjà. L'application doit déclarer tous ses composants, notamment les activités, dans un fichier AndroidManifest.xml. Ce fichier comprend également les informations concernant les permissions requises par l'application, le niveau minimum de l'API, les caractéristiques logicielles et matérielles et les bibliothèques requises.

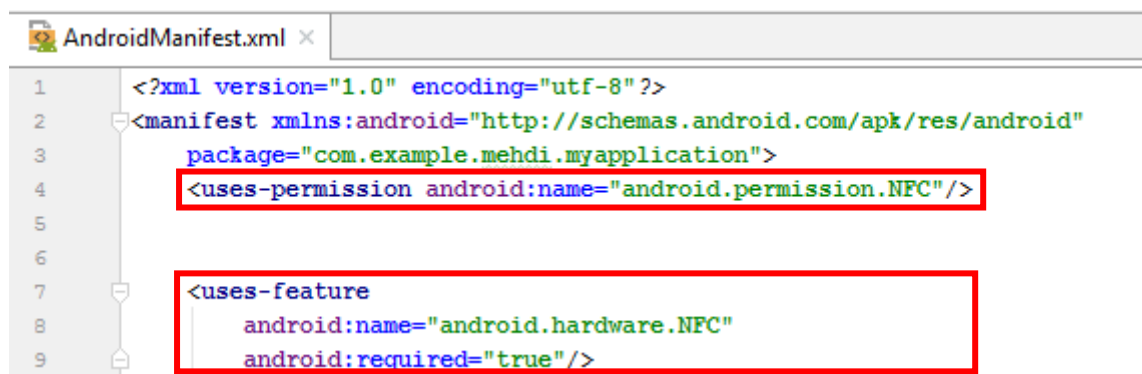


Figure 13 : Fichier AndroidManifest.xml

## Fichier MainActivity

Le but va être maintenant de vérifier si l'appareil possède le NFC et si ce dernier dispose de la fonction, de savoir si elle est activée. Sur le code ci-dessous, on vérifie dans le premier if si notre *nfcAdapter* est égal à *null*, auquel cas on part du principe que l'appareil ne prend pas en charge le NFC. Ensuite le *else if* est pour le cas où le *nfcAdapter* est différent de null et est activé. Enfin le *else* pour le cas où le NFC n'est pas activé.

```
// vérifie si l'appareil supporte le NFC et s'il est activé
if(nfcAdapter == null){
    Toast.makeText( context: this, text: "NFC n'est pas disponible", Toast.LENGTH_LONG).show();
}else if(nfcAdapter.isEnabled()){
    Toast.makeText( context: this, text: "NFC activé", Toast.LENGTH_LONG).show();
}else{
    Toast.makeText( context: this, text: "NFC non activé", Toast.LENGTH_LONG).show();
}
```

On notifie de chaque cas via le *Toast.makeText*.

## Modification de l'icône

Sur Android Studio, les applications développées ont cette icône par défaut et portent le nom « My application ».

Les parties relatives aux informations de l'icône se trouvent dans le Manifest.

Nous allons modifier ces informations afin d'y ajouter notre propre icône et nommer notre application.



Icône par défaut  
(format round)



Icône par défaut  
(format square)



The screenshot shows the AndroidManifest.xml file in Android Studio. The XML code is as follows:

```
13
14 <application
15     android:allowBackup="true"
16     android:icon="@mipmap/test icon"
17     android:label="Geo NFC"
18     android:roundIcon="@mipmap/test icon"
19     android:supportsRtl="true"
20     android:theme="@style/AppTheme">
21
```

Red boxes highlight the following attributes:

- `android:icon="@mipmap/test icon"` (points to "Chemin de l'icône")
- `android:label="Geo NFC"` (points to "Nom de l'application")
- `android:roundIcon="@mipmap/test icon"` (points to "Chemin de l'icône au format round")

Below the code, the text "Partie application du manifest" is written.

Afin que l'icône de notre application s'adapte à la plupart des types de téléphone, on rajoute l'icône dans les différents dossiers « mipmap » placés dans le dossier « res » de l'application avec des résolutions différentes (on se référera à la résolution des icônes d'origine).

<https://books.google.fr/books?id=DM8vr6LbgjIC&pg=PA204&lpg=PA204&dq=écriture+texte+tag+nfc&source=bl&ots=bWhwhMmBTX&sig=6v4Sl8spB9RuTh7cVNlrXhcdz9o&hl=fr&sa=X&ved=0ahUKEwihtdSRpMHbAhUCEVAKHUX1BHE4ChDoAQiJATAE#v=onepage&q=écriture%20texte%20tag%20nfc&f=false>

<https://www.youtube.com/watch?v=1jgY-7mG8rA> (resize image)

<https://developer.android.com/guide/topics/connectivity/nfc/nfc>