

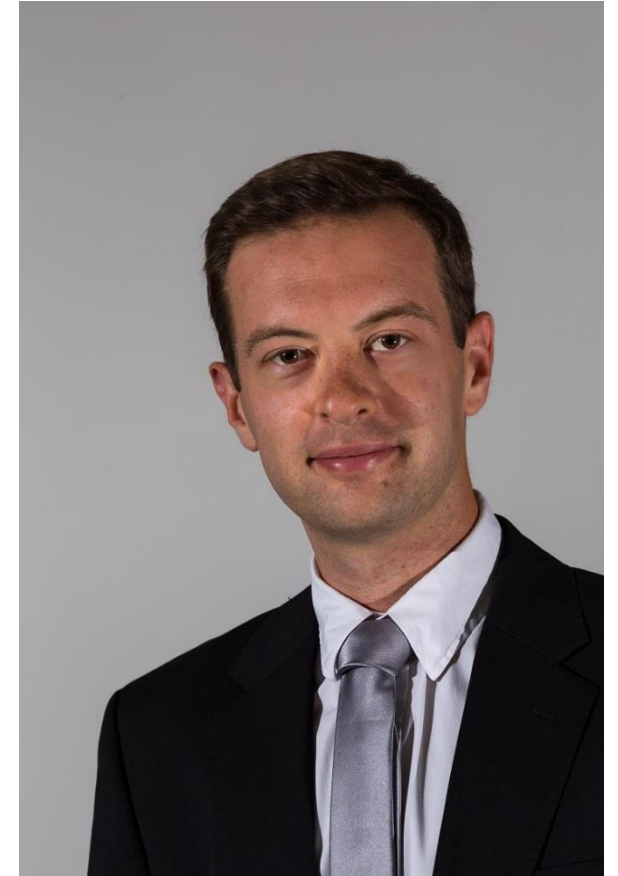
Environnements Logiciels pour l'Informatique Mobile

Introduction à Android



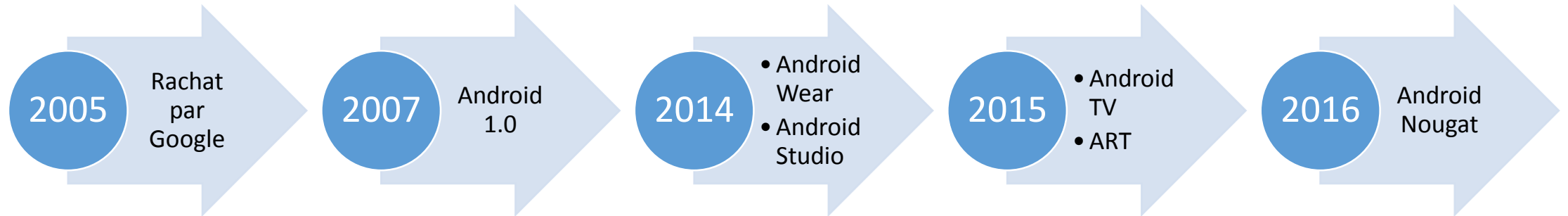
Présentation

- Polytech'Nice-Sophia 2012 (IAM)
- 5 ans chez Sopra-Steria
 - Développement
 - Architecture
 - Projets innovants
 - Formation
- gregory.marro@soprasteria.com



Un peu d'histoire ...

Une histoire qui commence en 2007



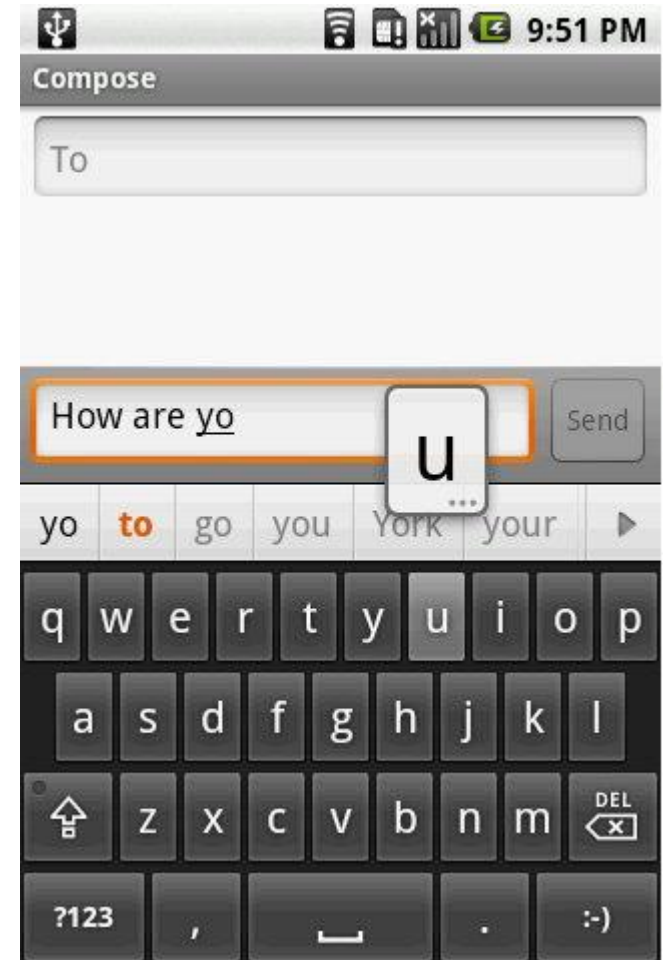
Android 1.0/1.1

- Lancé avec le HTC Dream fin 2008
- La 1.1 arrive en février 2009
- Apport :
 - Barre de notifications
 - Android Market
 - Widgets
 - Gestion de l'appareil photo



Android 1.5 « CupCake »

- Lancé le 30 Avril 2009
- Apport :
 - Clavier tactile
 - Presse papier
 - Enregistrement de vidéos



Android 1.6 « Donut »

- Lancé le 15 septembre 2009
- Apport :
 - Augmentation de la définition maximale à 1280x720 pixels
 - Prise en charge réseau CDMA



Android 2.0/2.1 « Eclair »

- Lancé le 26 octobre 2009
- Apport :
 - Gestion de plusieurs comptes Google sur un périphérique
 - Prise en charge Bluetooth 2.1
 - Google Maps
 - Synthèse vocale



Android 2.2/2.3 « Froyo »

- Lancé le 20 mai 2010
- Apport :
 - Améliorations graphiques
 - Gestion des densité de pixels jusqu'à 320ppi
 - Support du NFC, VOIP, caméra frontale
 - Synthèse vocale



Android 3.0 « Honeycomb »

- Lancé le 22 février 2011
- Réservée aux tablettes
- Apport :
 - Améliorations dans l'interface adaptée aux tablettes
 - Amélioration du multitâche



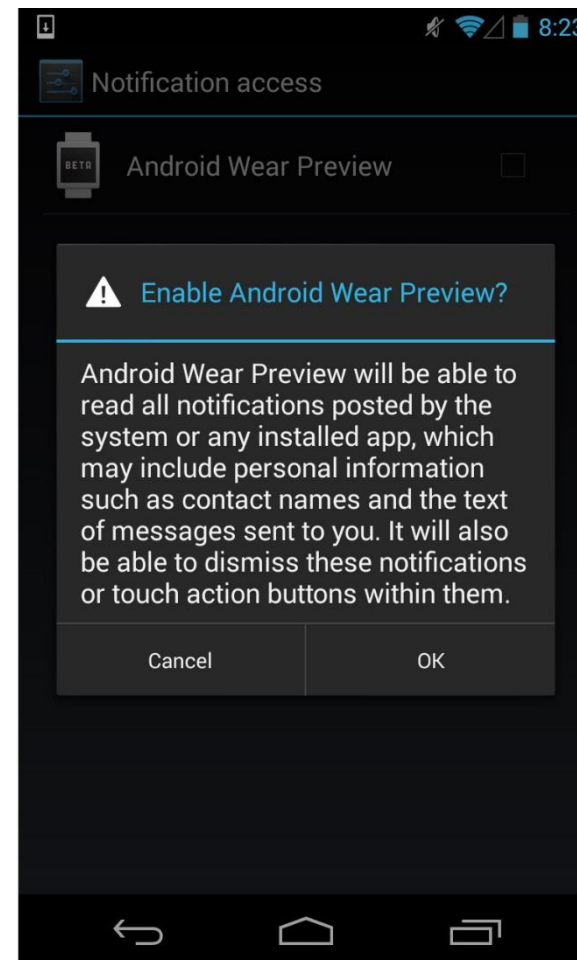
Android 4.0 « Ice Cream Sandwich »

- Lancé le 19 octobre 2011
- Smartphones et tablettes
- Apport :
 - Chiffrement des données
 - Dictionnaires de clavier
 - Gestion de la data
 - ...



Android 4.1/4.2/4.3 « Jelly Bean »

- Lancé le 27 juin 2012
- Apport :
 - Google Now
 - Clavier Swipe
 - Android Wear
 - Permissions multi-utilisateurs
 - OpenGL ES 3.0



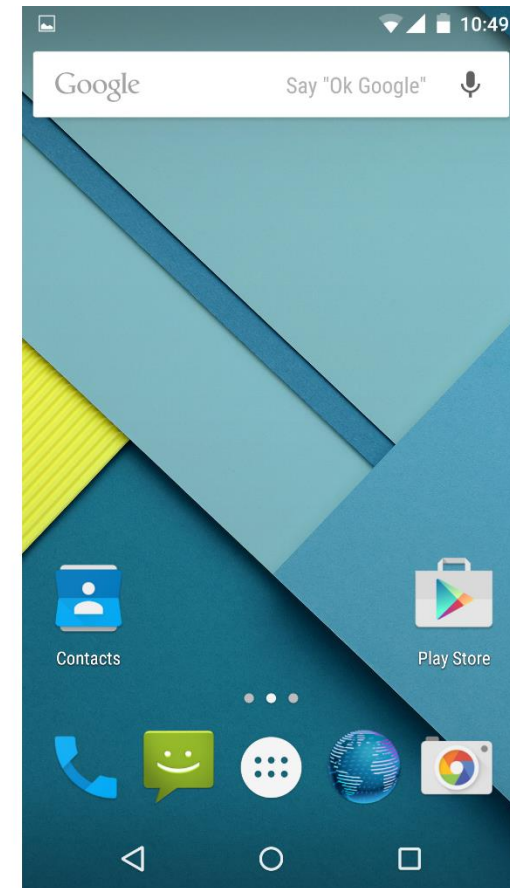
Android 4.4 « KitKat »

- Lancé le 31 octobre 2013
- Apport :
 - Diminution de la fragmentation
 - Nouveau style
 - Intégration plus poussée de Google Now



Android 5.0/5.1 « Lollipop »

- Lancé le 15 octobre 2014
- Apport :
 - Introduction du Material Design
 - Amélioration des notifications
 - Remplacement de Dalvik par ART
 - Support 64 bits
 - Android TV
 - Android Auto



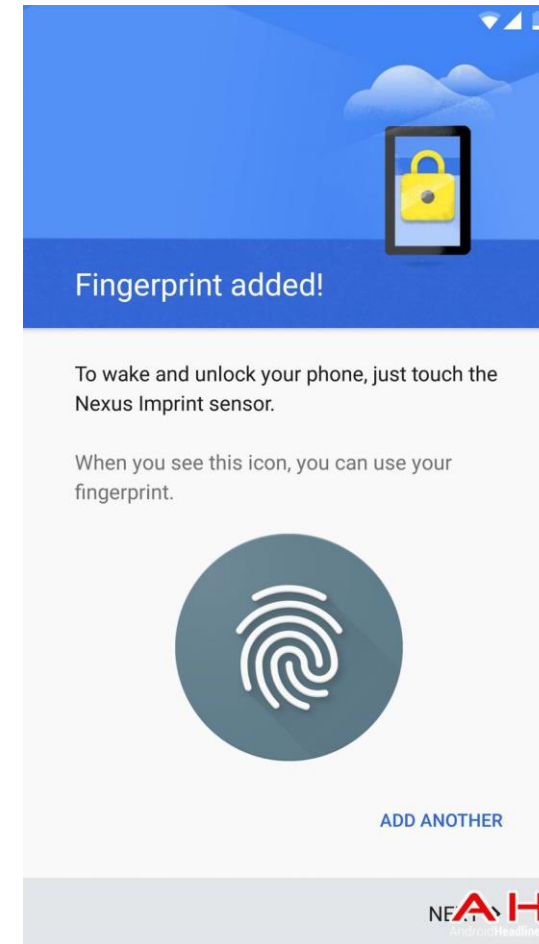
Android 6.0 « Marshmallow »

- Lancé le 5 octobre 2015
- Apport :
 - Autonomie améliorée avec Doze
 - Gestion des autorisations
 - Android Pay
 - Capteur d'empreinte digital



Android 7.0 « Nougat »

- Lancé le 22 août 2016
- Apport :
 - Multi-fenêtre
 - Centre de notifications
 - Capteur d'empreinte digital



Au cœur de l'OS

- 5 couches :
 - Noyau Linux (en 3.X depuis Android 4.0)
 - Bibliothèques logicielles (OpenGL, SQLite ...)
 - Machine virtuelle (Dalvik puis ART)
 - Android SDK (Framework)
 - Applications
- Open-source
- Compatibility Program



ART vs Dalvik

ART

- Introduit en 2015 en 4.4
- Compilation anticipée
- Amélioration du Garbage Collector

Dalvik

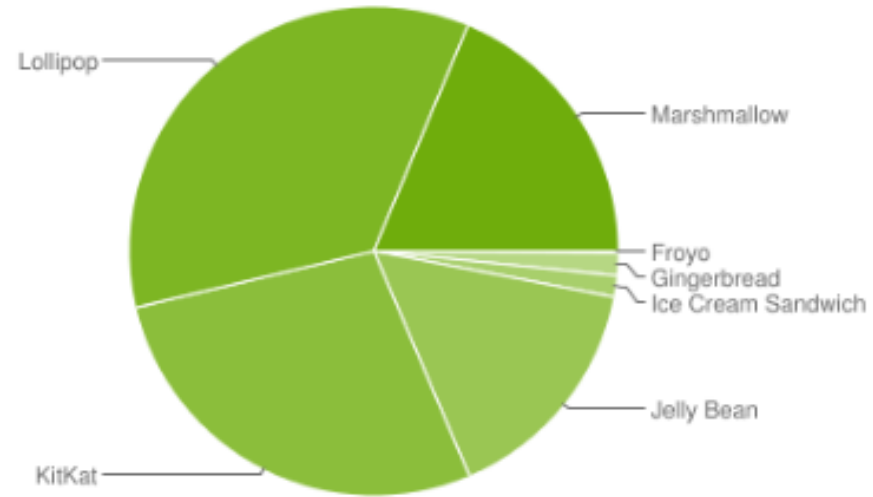
- Plus supporté actuellement
- Compilation JIT
- Orienté pour les périphériques avec de faibles capacités

Aujourd'hui

- 85% des parts de marché sur Smartphone
- Plus de 2 millions d'applications
- Couvre :
 - Smartphones
 - Tablettes
 - TV
 - Wearables

Aujourd'hui

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.5%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.4%
4.1.x	Jelly Bean	16	5.6%
4.2.x		17	7.7%
4.3		18	2.3%
4.4	KitKat	19	27.7%
5.0	Lollipop	21	13.1%
5.1		22	21.9%
6.0	Marshmallow	23	18.7%



L'environnement de développement

Le langage utilisée



+

<xml />

=



Java ?

- Java 8 (dans Android 7.0)
- Sans System.out/System.err
- Pas de AWT ou Swing
- Pas de java.beans (ou presque)



XML ?

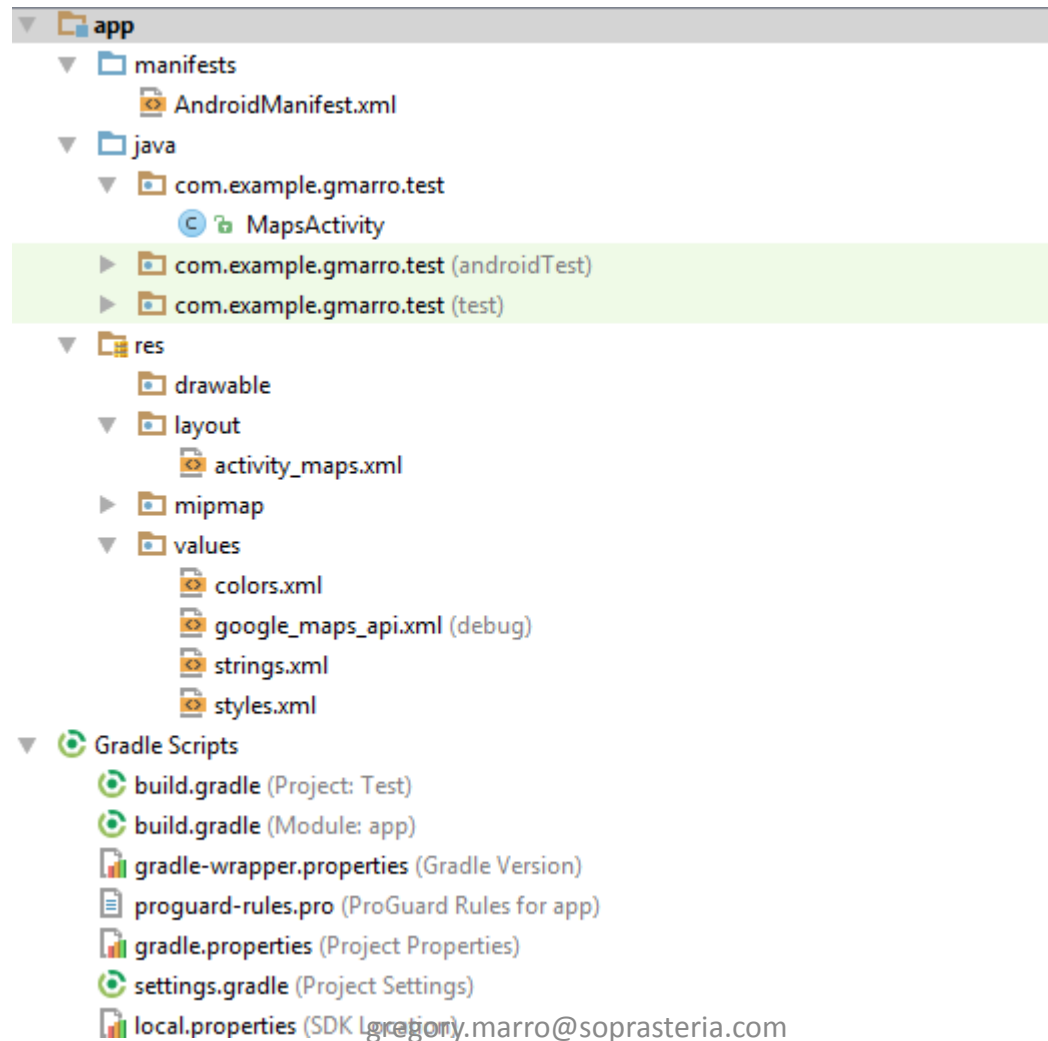
- Utilisation de namespace android
- Encodé par défaut en UTF-8
- Valeurs particulières pour les variables (@+id/valeur)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```


L'environnement de développement



Arborescence d'un projet Android



AndroidManifest.xml

- Descripteur de l'application
- Contient notamment:
 - Les permissions requises
 - Les icones de l'application
 - Le thème de l'application
 - La version minimum du SDK
 - Les points d'entrées de l'application
 - ...

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.gmarro.test">

    <!--
        The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
        Google Maps Android API v2, but you must specify either coarse or fine
        location permissions for the 'MyLocation' functionality.
    -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <!--
            The API key for Google Maps-based APIs is defined as a string resource.
            (See the file "res/values/google_maps_api.xml").
            Note that the API key is linked to the encryption key used to sign the APK.
            You need a different API key for each encryption key, including the release key that is used to
            sign the APK for publishing.
            You can define the keys for the debug and release targets in src/debug/ and src/release/.
        -->
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/google_maps_key" />

        <activity
            android:name=".MapsActivity"
            android:label="@string/title_activity_maps">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

Les Activity

- Point d'entrée d'un « écran »
- Hérite au minimum de `android.app.Activity` :
 - Method `onCreate()`
- Partie implémentation

Les Services

- Ne possède pas de vue
- Déclenché selon des évènements :
 - Démarrage de l'OS
 - Appel, SMS ...
 - Lancement d'application
- Exécution en arrière plan

Les broadcast et Intent receivers

- Permet l'écoute d'évènements système ou d'une application
- Déclenchement d'une action
- Souvent utilisé dans les services

Content providers

- Permet d'accéder a des ressources sur le périphériques
 - Contacts
 - Agenda
 - Photos
 - ...

Le dossier « res »

- Ressources de l'application :
 - Images
 - Icones
 - Externalisation de valeurs :
 - Fichiers de langage
 - Fichiers de couleur
 - Fichiers de propriétés
 - Fichiers de style
 - Fichiers de layout
- Présence également des ressources en fonction de la densité

Les layouts

- Contient les descriptions des vues de l'application
- Possibilité d'utiliser un éditeur WYSIWYG
 - Déconseillé pour une gestion fine et poussée
- Plusieurs approches :
 - LinearLayout
 - RelativeLayout
 - FrameLayout
 - TableLayout

Les layouts

LinearLayout

- Horizontal ou vertical
- Les éléments s'empilent les uns après les autres :
 - Listes

RelativeLayout

- Les éléments sont positionnés par rapport à leur parent ou leurs pairs :
 - Layout_below
 - Layout_toRightOf
 - Layout_centerVertical
 - ...

Les Views

- Composants Android ou personnalisés :
 - Button
 - TextView
 - WebView
 - ListView
 - ...
- L'éditeur WYSIWIG permet de les lister !

Le simulateur

- Peut être relativement lent
- Installation de HAXM pour améliorer les performances
- Utilisation d'un périphérique en mode développeur
 - Penser à installer les pilotes !

Hello World

TD : Réalisation d'un Hello World à base d'une « Empty Activity »

Objectif :

- Identifier les différents éléments qui composent l'application
- Centrer le texte au milieu de l'écran (horizontalement et verticalement)
- Ajouter un bouton qui remplace « Hello World » par « Hello » + votre prénom
- Récupérer l'espace disque total/libre grâce à `android.os.StatFs` et l'afficher dans un composant dédié

Bloqueur d'appel/SMS

Créer une application qui :

1. Possède un bouton qui active/désactive la réception de SMS et Appels
2. Liste les appels/SMS manqués
3. Envoi une notification si un appel/SMS est reçu 3 fois de suite