Pour envoyer des notifications depuis une application sur Raspberry Pi vers un téléphone en utilisant Firebase Cloud Messaging (FCM), voici un guide étape par étape. Cela implique d'utiliser l'API FCM pour envoyer des notifications push à des appareils mobiles.

**Prérequis :**

1. Un projet Firebase.
2. Un Raspberry Pi avec Raspbian installé.
3. Un environnement de développement Python sur Raspberry Pi.
4. Un téléphone avec une application capable de recevoir des notifications FCM.
5. La bibliothèque firebase-admin pour envoyer des notifications depuis Python.

**Étapes détaillées :**

**1. Créer un projet sur Firebase**

* Allez sur Firebase Console.
* Créez un nouveau projet ou sélectionnez un projet existant.
* Activez Firebase Cloud Messaging dans la section "Cloud Messaging" du projet.
* Téléchargez le fichier google-services.json ou la clé privée (à partir de la console Firebase > Paramètres du projet > Comptes de service > Générer une nouvelle clé privée).

**2. Préparer l'application mobile pour FCM**

* Intégrez Firebase Cloud Messaging dans votre application mobile (Android ou iOS).
  + Suivez les instructions sur la documentation officielle de Firebase pour Android ou iOS pour configurer votre projet avec Firebase.
  + Vous devez récupérer un **token FCM** (un identifiant unique pour chaque appareil) pour pouvoir envoyer des notifications vers cet appareil.

**3. Configurer votre Raspberry Pi**

* Assurez-vous que votre Raspberry Pi a Python installé.
* Installez les bibliothèques nécessaires avec pip :

Bash

sudo apt update

sudo apt install python3-pip

pip3 install firebase-admin

**4. Configurer Firebase Admin SDK**

* Sur votre Raspberry Pi, créez un fichier Python, par exemple send\_notification.py.
* Ajoutez le fichier de clé privée Firebase que vous avez téléchargé dans le projet Firebase (par exemple firebase-adminsdk.json).

**5. Écrire le script Python pour envoyer des notifications**

Voici un exemple de code Python pour envoyer des notifications FCM depuis votre Raspberry Pi vers un appareil mobile :

Python

import firebase\_admin

from firebase\_admin import credentials, messaging

# Initialisez Firebase Admin SDK avec la clé privée

cred = credentials.Certificate('chemin/vers/votre/firebase-adminsdk.json')

firebase\_admin.initialize\_app(cred)

def send\_notification(token, title, body):

# Créez un message FCM

message = messaging.Message(

notification=messaging.Notification(

title=title,

body=body

),

token=token,

)

# Envoyez le message

try:

response = messaging.send(message)

print('Notification envoyée avec succès:', response)

except Exception as e:

print('Erreur lors de l\'envoi de la notification:', e)

# Exemple d'utilisation

device\_token = 'votre\_token\_fcm' # Le token FCM de votre appareil mobile

send\_notification(device\_token, 'Titre de la notification', 'Corps de la notification')

**6. Tester l'envoi de notifications**

* Exécutez le script Python sur votre Raspberry Pi :

bash

Copier

python3 send\_notification.py

* Si tout est configuré correctement, vous devriez recevoir une notification sur votre appareil mobile.

**Quelques remarques :**

* Vous devez obtenir le **token FCM** de l'appareil mobile, généralement dans l'application mobile lors de l'inscription à FCM. Ce token est unique pour chaque appareil.
* Vous pouvez aussi envoyer des notifications à un groupe d'appareils en utilisant des "topics" FCM.
* Assurez-vous que votre téléphone a bien l'application qui écoute et gère les notifications FCM.

C'est tout ! Vous pouvez maintenant envoyer des notifications à partir de votre Raspberry Pi vers votre téléphone avec Firebase Cloud Messaging.

**application native**

Une **application native** est une application mobile qui est **développée spécifiquement pour un système d'exploitation mobile particulier** (comme **Android** ou **iOS**) en utilisant les **outils de développement et langages natifs** fournis par le fabricant du système d'exploitation. Elle est donc **optimisée pour une plateforme spécifique** et est installée directement sur l'appareil.

### Caractéristiques principales d'une application native :

1. **Langages de programmation natifs :**
   * **Android :** Les applications natives pour Android sont généralement développées en **Java** ou **Kotlin**.
   * **iOS :** Les applications natives pour iOS sont développées en **Swift** ou **Objective-C**.
2. **Accès aux fonctionnalités natives de l'appareil :**
   * Les applications natives ont un accès direct aux fonctionnalités du matériel du téléphone, comme la **caméra**, le **GPS**, le **microphone**, les **notifications push**, etc.
   * Elles peuvent également utiliser des **API spécifiques au système d'exploitation** pour une meilleure intégration avec l'OS.
3. **Performances optimisées :**
   * Les applications natives sont généralement plus performantes que les applications web ou hybrides, car elles sont directement compilées pour l'architecture du système d'exploitation.
   * Elles bénéficient aussi d'une meilleure gestion des ressources, ce qui les rend plus réactives.
4. **Distribution via les stores d'applications :**
   * Les applications natives sont téléchargées et installées via les **stores officiels** comme le **Google Play Store** pour Android ou l'**App Store** pour iOS.
   * Elles sont soumises à des processus de validation (par exemple, des revues de l'application avant sa publication sur le store).
5. **Expérience utilisateur optimale :**
   * Elles offrent généralement une **expérience utilisateur fluide** et sont optimisées pour les interfaces graphiques et les interactions de chaque plateforme.
   * Elles respectent les **directives de conception** spécifiques aux plateformes (par exemple, **Material Design** pour Android ou **Human Interface Guidelines** pour iOS).

### Avantages d'une application native :

* **Performances supérieures** grâce à un accès direct aux ressources matérielles de l'appareil.
* **Meilleure intégration** avec les services et fonctionnalités de l'OS (notifications push, géolocalisation, caméra, etc.).
* Une **expérience utilisateur plus fluide** et spécifique à chaque plateforme.
* Accès aux **dernières fonctionnalités et API** fournies par le système d'exploitation.

### Inconvénients :

* **Développement spécifique à chaque plateforme**, ce qui nécessite de maintenir deux bases de code distinctes si l'on veut une version Android et une version iOS.
* Cela peut entraîner des **coûts de développement et de maintenance plus élevés** comparé à des solutions comme les applications web ou hybrides.

### Exemple :

* **WhatsApp**, **Instagram**, et **TikTok** sont des applications natives, car elles sont développées spécifiquement pour Android et iOS, avec une expérience utilisateur adaptée à chaque plateforme.

En résumé, une **application native** est une application qui est **conçue et optimisée** pour un système d'exploitation spécifique (Android ou iOS) en utilisant les outils et langages natifs de la plateforme. Elle offre de meilleures performances et une intégration plus poussée avec les fonctionnalités de l'appareil, mais nécessite des efforts de développement distincts pour chaque plateforme.

**Fonctionnement de FCM**

Firebase Cloud Messaging (FCM) est un service de **messagerie push** proposé par Google, permettant d'envoyer des notifications ou des messages aux utilisateurs de vos applications mobiles ou web. Il fonctionne en utilisant les **serveurs Firebase** pour envoyer des messages vers des appareils cibles via les **serveurs de Google**. Voici une explication détaillée de son fonctionnement :

**1. Architecture et Composants de FCM :**

**a. Serveur d'envoi (Back-end / Raspberry Pi ou autre serveur) :**

* Vous avez un **serveur** (dans votre cas, le Raspberry Pi ou une autre application back-end) qui est chargé de **générer et envoyer** des notifications.
* Ce serveur communique avec **Firebase Cloud Messaging** via l'API FCM pour envoyer les messages aux appareils mobiles.

**b. Firebase Cloud Messaging (FCM) :**

* FCM est la **passerelle** entre votre serveur et les appareils cibles. Une fois qu'une notification est envoyée par votre serveur à Firebase, FCM se charge de la **délivrer aux appareils cibles**, en passant par les **serveurs de Google**.

**c. Appareil mobile (Client) :**

* L'appareil mobile (Android, iOS ou PWA) est configuré pour recevoir des notifications via FCM.
* L'application mobile enregistre un **token FCM unique** qui identifie de manière unique cet appareil auprès de Firebase. Ce token est utilisé pour adresser spécifiquement une notification à cet appareil.

**2. Processus de fonctionnement de FCM :**

**a. Inscription de l'appareil (Client) :**

* Lorsqu'un utilisateur ouvre l'application pour la première fois, l'appareil **s'enregistre auprès de Firebase** pour recevoir des notifications.
* L'application mobile contacte Firebase et obtient un **token FCM unique** pour cet appareil. Ce token est stocké sur l'appareil et est également envoyé à votre serveur (via l'application), afin que le serveur puisse l'utiliser pour cibler cet appareil lors de l'envoi de notifications.

**b. Envoi d'une notification depuis le serveur :**

* Lorsque vous voulez envoyer une notification, votre **serveur (Raspberry Pi ou autre back-end)** envoie une requête à Firebase, en utilisant l'API FCM.
* La requête inclut des informations importantes :
  + **Token FCM** : le token de l'appareil cible auquel la notification doit être envoyée.
  + **Contenu de la notification** : le titre, le corps du message, les données supplémentaires (si nécessaire), etc.
  + **Type de message** : Notification simple (avec titre et corps) ou message de données (avec données personnalisées).

**c. FCM délivre le message à l'appareil :**

* Une fois la requête reçue par Firebase, **FCM prend en charge la gestion de l'envoi** de la notification au périphérique cible via les serveurs de Google.
* Si l'appareil est en ligne (connecté à Internet), le message est immédiatement envoyé et l'utilisateur recevra la notification.
* Si l'appareil est **hors ligne**, FCM va **mettre en file d'attente** le message et le livrer dès que l'appareil est de nouveau en ligne.

**d. Réception de la notification sur le mobile :**

* Lorsque l'appareil reçoit la notification, une **notification système** est affichée à l'utilisateur, si c'est une notification classique.
* Si le message est un message de données (pas une notification simple), l'application peut le traiter en arrière-plan (par exemple, pour mettre à jour l'interface utilisateur, ou pour démarrer une action).

**3. Types de messages FCM :**

Il y a deux principaux types de messages que vous pouvez envoyer via FCM :

**a. Notification Messages (Messages de notification) :**

* Ce sont des messages qui apparaissent directement dans la barre de notifications de l'appareil.
* Exemple : un message simple contenant un titre et un corps, comme "Vous avez un nouveau message".
* Lorsque l'utilisateur clique sur la notification, l'application peut être ouverte, ou une autre action peut être effectuée.

**b. Data Messages (Messages de données) :**

* Ce sont des messages contenant des **données personnalisées**, mais qui ne génèrent pas de notifications visibles immédiatement.
* Ce type de message est utilisé pour envoyer des données en arrière-plan, permettant à l'application d'effectuer des actions sans intervention directe de l'utilisateur.
* Exemple : mettre à jour des informations de l'application ou synchroniser des données.

**c. Messages combinés (Notification + Data) :**

* Vous pouvez aussi envoyer des messages combinés qui incluent à la fois des notifications visibles et des données.
* Exemple : Vous pouvez envoyer une notification et, en même temps, des données supplémentaires pour mettre à jour une partie de l'application.

**4. Sécurité et Authentification :**

* Pour envoyer des notifications via l'API FCM, vous devez utiliser un **token d'accès** sécurisé. Firebase s'appuie sur un **clé serveur** (disponible dans la console Firebase) pour authentifier les requêtes envoyées depuis votre serveur.
* Vous devez aussi vous assurer que l’application mobile enregistre correctement l'appareil dans le projet Firebase, ce qui garantit que seuls les utilisateurs autorisés peuvent recevoir des notifications.

**5. Gestion des tokens FCM :**

* Les **tokens FCM** peuvent être révoqués ou actualisés automatiquement par Firebase.
* Il est essentiel de **gérer** ces tokens pour s'assurer que les notifications sont envoyées aux bons appareils. Par exemple, si un utilisateur désinstalle l'application, vous devez retirer le token de votre base de données.

**En résumé, voici le flux complet :**

1. **L'utilisateur ouvre l'application mobile.** L'appareil enregistre son token FCM auprès de Firebase.
2. **L'application mobile envoie le token au serveur.** Le serveur enregistre ce token pour l'utiliser plus tard.
3. **Lorsque vous voulez envoyer une notification**, votre serveur envoie une requête à Firebase en incluant le token de l'appareil et le message à envoyer.
4. **Firebase délivre la notification** à l'appareil cible via ses serveurs.
5. **L'appareil mobile reçoit la notification** (ou les données) et l'affiche à l'utilisateur ou effectue une action en arrière-plan.

FCM est un service très puissant qui simplifie la gestion des notifications push à grande échelle, tout en permettant une grande flexibilité dans la manière dont les notifications sont envoyées et reçues.