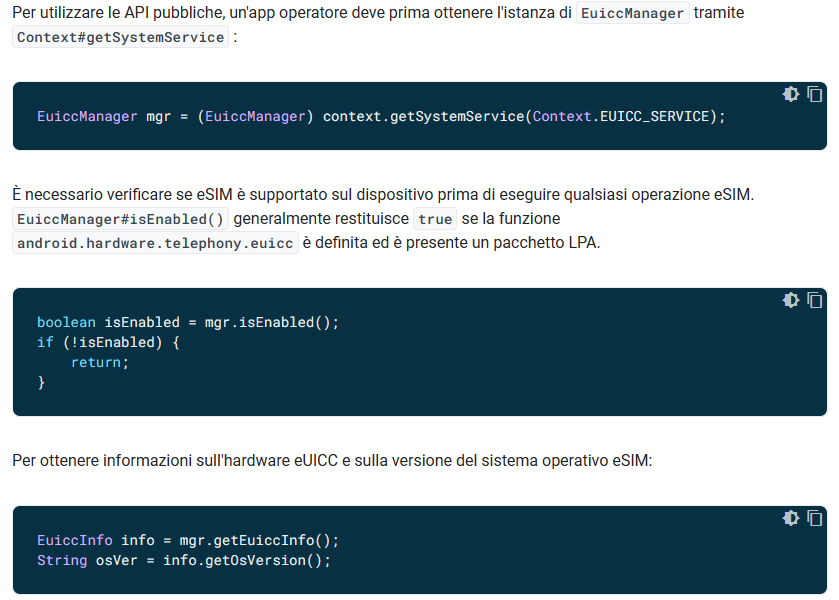
**CLASSI FONDAMENTALI**

-> **EuiccManager**: interfaccia all’eUICC e alla eSIM. È il punto di ingresso principale per le app per interagire con LPA (Local Profile Assistant)1. Infatti, le relative API richiedono un’app LPA correttamente implementata per funzionare.  
-> **EuiccService**: classe utile per connettersi ai servizi LPA.  
-> **EuiccCardManager**: classe che fornisce funzioni ES10x basate su GSMA RSP v2.02. Il chiamante delle relative API deve essere un LPA.

*1 Un’app LPA è responsabile della gestione dei profili degli abbonati all’interno della rete mobile, inclusa la creazione, l’aggiornamento e la cancellazione dei profili degli abbonati. Può essere anche usato per il controllo dell’accesso alla rete e per la sicurezza dei dati degli abbonati.*

*2 Il protocollo RSP (Remote SIM Provisioning) permette di aggiornare le informazioni contenute sulla SIM card di un dispositivo mobile tramite una connessione di rete sicura, senza la necessità di rimuovere fisicamente la SIM card dal dispositivo. Le funzioni ES10x basate su GSMA RSP, quindi, sono funzioni utilizzate per la gestione delle credenziali di sicurezza degli utenti (e.g. l’autenticazione, l’autorizzazione, la gestione delle chiavi, la crittografia).*

**EuiccManager**



**Scaricamento di un abbonamento creato da codice di attivazione o codice QR**  
Avviene mediante l’API downloadSubscription(), che fa uso del callback PendingIntent (il quale deve essere BroadcastReceiver). In particolare:  
-> **downloadSubscription()**, oltre a consentire di gestire i download di abbonamenti all’interno di un’applicazione Android, fornisce funzionalità per gestire la fatturazione degli abbonamenti stessi (e.g. verifica dello stato dell’abbonamento, richiesta di rinnovo dell’abbonamento).  
-> **PendingIntent** è un meccanismo di Android che consente a un’applicazione di richiedere l’esecuzione di un’azione in un momento successivo (e.g. quando l’utente fa clic su un’attività o una notifica).  
-> **BroadcastReceiver** è un componente del sistema Android che consente alle applicazioni di ricevere e rispondere a messaggi e notifiche provenienti dal sistema o da altre applicazioni. In altre parole, funge da ascoltatore per determinati eventi che si verificano all’interno del sistema Android (e.g. avvio del dispositivo o ricezione di un messaggio di testo). In particolare, quando si verifica un evento per cui un BroadcastReceiver è stato registrato, il sistema Android invia un messaggio di intent all’applicazione. Quest’ultima può quindi elaborare il messaggio di intent e prendere le opportune azioni in base al tipo di evento che si è verificato.  
-> Un **messaggio di intent** è un oggetto utilizzato per comunicare informazioni tra diverse componenti dell’applicazione o tra diverse applicazioni all’interno del sistema Android. In pratica, rappresenta un’azione da compiere o un evento che è avvenuto / deve avvenire nel sistema. Inoltre, include informazioni come il tipo di azione da eseguire, i dati da trasferire, il destinatario dell’intent, ma anche le autorizzazioni richieste dall’azione e l’URI dell’intent.

Immagine che contiene testo

Descrizione generata automaticamente