Hướng dẫn tích hợp mạng quảng cáo

để kiếm tiền từ ứng dụng Android & iOS

Ad Monetization · Mediation · eCPM Optimization

Hoan Lê - The Masters Reach (TMR Team)

Từ việc chọn đúng ad format, tích hợp SDK, đến tối ưu mediation —
để ứng dụng tạo ra doanh thu mà không làm mất người dùng.

Mở đầu: App miễn phí kiếm tiền bằng cách nào?

Hơn 95% ứng dụng trên App Store và Google Play hiện nay được phát hành miễn phí — nhưng điều đó không có nghĩa là chúng không tạo ra doanh thu. Quảng cáo trong app (in-app advertising) chiếm hơn 50% tổng doanh thu của ngành ứng dụng di động toàn cầu, vượt qua cả in-app purchase và subscription.

Tuy nhiên, tích hợp quảng cáo không chỉ đơn giản là copy-paste vài dòng code rồi ngồi đợi tiền về. Nếu làm sai — đặt quảng cáo sai thời điểm, chọn sai format, hoặc bỏ qua bước tối ưu mediation — bạn sẽ vừa kiếm được ít tiền hơn mức có thể, vừa đẩy người dùng rời bỏ ứng dụng.

Bài viết này sẽ hướng dẫn bạn từ nền tảng đến nâng cao: chọn đúng ad format cho từng loại app, tích hợp SDK đúng cách, khai báo những gì cần thiết để lên store, và quan trọng nhất — cách tăng eCPM lên 20-40% bằng mediation mà phần lớn developer đang bỏ qua.

1. Hiểu các Ad Format trước khi tích hợp

Mỗi ad format có đặc điểm riêng về trải nghiệm người dùng, eCPM, và ngữ cảnh phù hợp. Chọn đúng format quan trọng hơn chọn đúng mạng quảng cáo.

Format	eCPM tương đối	UX impact	Phù hợp với
Banner	Thấp ($0.2–$1.5)	Rất thấp — hiển thị liên tục ở đầu/cuối màn hình	News, content, utility app. Không phù hợp với game.
Interstitial	Trung bình ($2–$8)	Trung bình — toàn màn hình, xuất hiện ở điểm chuyển tiếp	Game giữa các level, app sau khi hoàn thành tác vụ
Rewarded Video	Cao nhất ($8–$25)	Tích cực — user chủ động chọn xem để nhận phần thưởng	Game (extra lives, coins), app có virtual goods
Native	Trung bình ($1–$5)	Thấp — hòa vào UI, trông như nội dung tự nhiên	News app, social app, content feed
App Open	Trung bình ($2–$6)	Thấp–trung bình — xuất hiện khi user mở hoặc resume app	Utility app, productivity app
Rewarded Interstitial	Trung cao ($5–$15)	Nhẹ — toàn màn hình nhưng user có thể bỏ qua sau 5 giây	Game, app dùng thường xuyên

1.1 Nguyên tắc vàng khi chọn format

- **Revenue không phải là chỉ số duy nhất:** Một banner eCPM $0.5 hiển thị liên tục có thể làm giảm retention 15–20%, kéo theo mất doanh thu dài hạn từ chính user đó.
- **Rewarded video là format tốt nhất cho hầu hết trường hợp:** User chủ động, không bị cưỡng ép — đây là lý do format này vừa có eCPM cao nhất vừa ít tác động tiêu cực nhất đến retention.
- **Interstitial cần impression cap:** Không nên hiển thị quá 1 interstitial mỗi 3–5 phút. Vượt ngưỡng này, người dùng sẽ gỡ app thay vì tiếp tục dùng.
- **Đừng dùng banner trong game:** Banner liên tục trong khi chơi game là một trong những lý do phổ biến nhất khiến game bị review 1 sao.

1.2 Mapping format theo loại app

Loại app	Format ưu tiên	Format thứ hai	Tránh
Casual game	Rewarded Video	Interstitial (giữa level)	Banner trong gameplay
Hyper-casual game	Interstitial + Rewarded	App Open	Native (không phù hợp UX)

Loại app	Format ưu tiên	Format thứ hai	Tránh
Utility / Tool	Interstitial nhẹ + App Open	Banner ở màn hình kết quả	Rewarded (ít context để trao thưởng)
News / Content	Native + Banner	Interstitial khi chuyển bài	Rewarded Video (không có context)
Social / Community	Native	Interstitial khi mở profile	Banner liên tục
Education	Rewarded (mở nội dung premium)	Interstitial giữa bài học	Banner trong bài học

---

2. Tích hợp mạng quảng cáo đầu tiên: Google AdMob

AdMob là điểm khởi đầu tốt nhất: fill rate cao toàn cầu, tài liệu chi tiết, và miễn phí không giới hạn. Khi đã tích hợp AdMob thành công, bạn có thể thêm các mạng khác qua mediation.

2.1 Đăng ký và lấy thông tin cần thiết

Trước khi viết code, bạn cần:

- Đăng ký tài khoản AdMob tại admob.google.com
- Tạo App trong dashboard AdMob (Android và iOS là hai app riêng)
- Lấy App ID dạng: ca-app-pub-XXXXXXXXXXXXXXXX~XXXXXXXXXX
- Tạo Ad Unit cho từng format muốn hiển thị — mỗi Ad Unit có một ID riêng

> Không dùng Ad Unit ID thật khi đang phát triển. Dùng test Ad Unit ID của Google để tránh bị phát hiện invalid traffic và khoá tài khoản AdMob.

2.2 Android — Tích hợp AdMob SDK

Bước 1: Thêm dependency vào build.gradle

```
dependencies {
    implementation 'com.google.android.gms:play-services-ads:23.+'
}
```

Bước 2: Khai báo trong AndroidManifest.xml

```
<manifest>
  <uses-permission android:name="android.permission.INTERNET"/>

  <!-- Android 13+: cần để đọc Advertising ID -->
  <uses-permission
    android:name="com.google.android.gms.permission.AD_ID"/>

  <application>
    <!-- AdMob App ID — thay bằng ID thật của bạn -->
    <meta-data
      android:name="com.google.android.gms.ads.APPLICATION_ID"
      android:value="ca-app-pub-XXXXXXXXXXXXXXXX~XXXXXXXXXX"/>
```

```
  </application>
</manifest>
```

Bước 3: Khởi tạo SDK trong Application class

```
class MyApp : Application() {
    override fun onCreate() {
        super.onCreate()

        MobileAds.initialize(this) { initializationStatus ->
            // SDK đã sẵn sàng, có thể bắt đầu load ad
            val statusMap = initializationStatus.adapterStatusMap
            for ((adapter, status) in statusMap) {
                Log.d("AdMob", "Adapter: $adapter, Status:
${status.initializationState}")
            }
        }
    }
}
```

Bước 4: Load và hiển thị Banner Ad

```
// Trong layout XML
<com.google.android.gms.ads.AdView
    android:id="@+id/adView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    ads:adSize="BANNER"
    ads:adUnitId="ca-app-pub-3940256099942544/6300978111"/>

// Trong Activity/Fragment
val adView = findViewById<AdView>(R.id.adView)
val adRequest = AdRequest.Builder().build()
adView.loadAd(adRequest)
```

Bước 5: Load và hiển thị Rewarded Ad

Rewarded ad cần được pre-load trước khi hiển thị để tránh delay làm gián đoạn trải nghiệm người dùng:

```
private var rewardedAd: RewardedAd? = null

// Load ad (gọi ngay sau khi user cần, trước khi họ click nút xem ad)
private fun loadRewardedAd() {
    val adRequest = AdRequest.Builder().build()
    RewardedAd.load(
        this,
        "ca-app-pub-3940256099942544/5224354917", // Test ID
        adRequest,
        object : RewardedAdLoadCallback() {
            override fun onAdLoaded(ad: RewardedAd) {
                rewardedAd = ad
            }
            override fun onAdFailedToLoad(error: LoadAdError) {
                rewardedAd = null
            }
        }
```

```
    )
}

// Hiển thị ad khi user click nút "Xem quảng cáo để nhận thưởng"
private fun showRewardedAd() {
    rewardedAd?.let { ad ->
        ad.show(this) { rewardItem ->
            // User đã xem hết — trao phần thưởng
            val rewardAmount = rewardItem.amount
            grantUserReward(rewardAmount)
        }
        // Load ad tiếp theo ngay lập tức
        loadRewardedAd()
    } ?: run {
        // Ad chưa load xong — disable nút hoặc thông báo user
    }
}
```

2.3 iOS — Tích hợp AdMob SDK

Bước 1: Thêm SDK qua CocoaPods

```
# Podfile
pod 'Google-Mobile-Ads-SDK'
```

Chạy pod install sau đó mở file .xcworkspace (không phải .xcodeproj).

Bước 2: Khai báo trong Info.plist

```
<!-- AdMob App ID -->
<key>GADApplicationIdentifier</key>
<string>ca-app-pub-XXXXXXXXXXXXXXXX~XXXXXXXXXX</string>

<!-- Bắt buộc để hiển thị ATT prompt — cần có trước iOS 14.5 -->
<key>NSUserTrackingUsageDescription</key>
<string>Cho phép chúng tôi hiển thị quảng cáo phù hợp
 hơn với sở thích của bạn. Số lượng quảng cáo
 không thay đổi.</string>

<!-- Cho phép HTTP nếu một số ad network cần (nên hạn chế) -->
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
</dict>

<!-- SKAdNetwork IDs — xem Section 3 để biết danh sách đầy đủ -->
<key>SKAdNetworkItems</key>
<array>
  <dict>
    <key>SKAdNetworkIdentifier</key>
    <string>cstr6suwn9.skadnetwork</string>
  </dict>
</array>
```

Bước 3: Khởi tạo trong AppDelegate.swift

```
import GoogleMobileAds
import AppTrackingTransparency
```

```
@main
class AppDelegate: UIResponder, UIApplicationDelegate {

    func application(_ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions:
        [UIApplication.LaunchOptionsKey: Any]?) -> Bool {

        // Khởi tạo AdMob SDK
        GADMobileAds.sharedInstance().start(completionHandler: nil)
        return true
    }
}

// Gọi ATT prompt đúng thời điểm (sau onboarding)
// Trong ViewController phù hợp:
func requestTrackingPermission() {
    if #available(iOS 14.5, *) {
        ATTrackingManager.requestTrackingAuthorization { status in
            // Bắt đầu load ad sau khi có kết quả
            DispatchQueue.main.async {
                self.loadBannerAd()
            }
        }
    } else {
        loadBannerAd()
    }
}
```

Bước 4: Load và hiển thị Rewarded Ad trên iOS

```
import GoogleMobileAds

class GameViewController: UIViewController {

    private var rewardedAd: GADRewardedAd?

    func loadRewardedAd() {
        let request = GADRequest()
        GADRewardedAd.load(
            withAdUnitID: "ca-app-pub-3940256099942544/1712485313",
            request: request
        ) { [weak self] ad, error in
            guard let self = self else { return }
            if let error = error {
                print("Rewarded ad failed: \(error.localizedDescription)")
                return
            }
            self.rewardedAd = ad
        }
    }

    func showRewardedAd() {
        guard let ad = rewardedAd else {
            print("Ad not ready")
            return
```

```
        }
        ad.present(fromRootViewController: self) { [weak self] in
            let reward = ad.adReward
            print("User rewarded: \(reward.amount) \(reward.type)")
            self?.grantReward(amount: reward.amount.intValue)
            // Pre-load next ad
            self?.loadRewardedAd()
        }
    }
}
```

---

3. Khai báo bắt buộc: SKAdNetwork và Permissions

Trước khi submit app lên store, một số khai báo là bắt buộc. Thiếu các khai báo này có thể khiến app bị reject hoặc quảng cáo không hoạt động đúng.

3.1 Android — Khai báo đầy đủ AndroidManifest.xml

Dưới đây là template AndroidManifest.xml hoàn chỉnh cho app tích hợp AdMob và Meta Audience Network:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android">

  <!-- Quyền mạng — bắt buộc -->
  <uses-permission android:name="android.permission.INTERNET"/>
  <uses-permission
    android:name="android.permission.ACCESS_NETWORK_STATE"/>

  <!-- Android 13+: Advertising ID -->
  <uses-permission
    android:name="com.google.android.gms.permission.AD_ID"/>

  <!-- Package visibility Android 11+ -->
  <queries>
    <package android:name="com.android.vending"/>
    <intent>
      <action android:name="android.intent.action.VIEW"/>
      <category android:name="android.intent.category.BROWSABLE"/>
      <data android:scheme="https"/>
    </intent>
  </queries>

  <application>
    <!-- AdMob App ID -->
    <meta-data
      android:name="com.google.android.gms.ads.APPLICATION_ID"
      android:value="ca-app-pub-XXXXXXXXXXXXXXXX~XXXXXXXXXX"/>

    <!-- Cho phép cleartext traffic nếu mạng nào đó cần -->
    <!-- <meta-data android:name=
      "android:usesCleartextTraffic" android:value="true"/> -->
  </application>
</manifest>
```

3.2 iOS — Danh sách SKAdNetworkIdentifier

Mỗi ad network có một hoặc nhiều SKAdNetwork ID riêng. Thiếu ID của mạng nào, mạng đó sẽ không điền quảng cáo hiệu quả và không nhận được conversion data.

> *Khi dùng mediation platform như AppLovin MAX hoặc AdMob Mediation, họ thường cung cấp script hoặc danh sách tự động cập nhật để generate toàn bộ SKAdNetworkItems. Ưu tiên dùng công cụ này.*

```
<key>SKAdNetworkItems</key>
<array>
  <!-- Google AdMob -->
  <dict><key>SKAdNetworkIdentifier</key>
        <string>cstr6suwn9.skadnetwork</string></dict>

  <!-- Meta Audience Network -->
  <dict><key>SKAdNetworkIdentifier</key>
        <string>v9wttpbfk9.skadnetwork</string></dict>

  <!-- AppLovin -->
  <dict><key>SKAdNetworkIdentifier</key>
        <string>ludvb6z3bs.skadnetwork</string></dict>

  <!-- Unity Ads -->
  <dict><key>SKAdNetworkIdentifier</key>
        <string>4468km3ulz.skadnetwork</string></dict>

  <!-- ironSource -->
  <dict><key>SKAdNetworkIdentifier</key>
        <string>su67r6k2v3.skadnetwork</string></dict>

  <!-- Pangle (TikTok) -->
  <dict><key>SKAdNetworkIdentifier</key>
        <string>238da6jt44.skadnetwork</string></dict>

  <!-- Mintegral -->
  <dict><key>SKAdNetworkIdentifier</key>
        <string>kbd757ywx3.skadnetwork</string></dict>

  <!-- Liftoff (Vungle) -->
  <dict><key>SKAdNetworkIdentifier</key>
        <string>gta9lk7p23.skadnetwork</string></dict>

  <!-- Kiểm tra tài liệu từng mạng để lấy danh sách đầy đủ -->
</array>
```

> Danh sách SKAdNetwork IDs thay đổi thường xuyên khi Apple cập nhật hoặc mạng thêm ID mới. Kiểm tra lại tài liệu từng mạng trước mỗi lần release lớn.

---

4. Mediation — Vũ khí tăng eCPM mà nhiều developer bỏ qua

Đây là phần quan trọng nhất trong bài viết. Nếu bạn chỉ đọc một section, hãy đọc section này.

4.1 Tại sao một mạng quảng cáo không bao giờ là đủ?

Khi chỉ dùng AdMob, fill rate của bạn sẽ không bao giờ đạt 100% — tức là trong một số trường hợp không có quảng cáo để hiển thị, bạn mất doanh thu. Quan trọng hơn, AdMob đặt giá cho ad slot của bạn dựa trên đấu giá nội bộ của họ — không có áp lực cạnh tranh từ bên ngoài.

Mediation giải quyết cả hai vấn đề: kết nối nhiều ad network cùng lúc, để chúng cạnh tranh với nhau. Network nào trả cao nhất sẽ được hiển thị.

4.2 Waterfall vs In-app Bidding

Có hai mô hình mediation khác nhau về cơ bản:

Waterfall (mô hình cũ):

Bạn định nghĩa thủ công thứ tự ưu tiên: nếu AdMob không có ad với giá trên $3, thử Meta; nếu Meta không có, thử Unity; và tiếp tục xuống dưới. Vấn đề: thứ tự này cố định, không phản ánh được biến động giá thực tế theo thời gian.

In-app Bidding (mô hình mới — khuyến nghị):

Tất cả các mạng đấu thầu đồng thời trong thời gian thực (dưới 100ms). Mạng trả cao nhất thắng và hiển thị quảng cáo. Kết quả: eCPM trung bình tăng 20-40% so với waterfall theo số liệu của AppLovin và ironSource.

> Nên dùng In-app Bidding từ đầu nếu có thể. AppLovin MAX và ironSource LevelPlay là hai mediation platform hỗ trợ bidding tốt nhất hiện nay và miễn phí hoàn toàn.

4.3 Tích hợp AppLovin MAX làm mediation layer

AppLovin MAX là mediation platform miễn phí, hỗ trợ in-app bidding, và có network adapter cho hầu hết các mạng lớn (AdMob, Meta, Unity, ironSource, Pangle...).

Android — Thêm dependency:

```
dependencies {
    // AppLovin MAX SDK
    implementation 'com.applovin:applovin-sdk:+'

    // Các network adapters — thêm mạng nào bạn muốn tích hợp
    implementation 'com.applovin.mediation:google-adapter:+'
    implementation 'com.applovin.mediation:facebook-adapter:+'
    implementation 'com.applovin.mediation:unityads-adapter:+'
    implementation 'com.applovin.mediation:ironsource-adapter:+'
}
```

Khởi tạo AppLovin MAX SDK:

```
class MyApp : Application() {
    override fun onCreate() {
        super.onCreate()

        AppLovinSdk.getInstance(this).mediationProvider = "max"
        AppLovinSdk.getInstance(this).initializeSdk { sdkConfig ->
            // SDK và tất cả network adapter đã sẵn sàng
            // Bắt đầu load ad
        }
    }
}
```

Load Rewarded Ad qua MAX:

```
private val rewardedAd = MaxRewardedAd.getInstance("YOUR_AD_UNIT_ID", activity)

init {
    rewardedAd.setListener(object : MaxRewardedAdListener {
        override fun onAdLoaded(ad: MaxAd) {
            // Ad sẵn sàng — enable nút "Xem quảng cáo"
        }
        override fun onAdDisplayed(ad: MaxAd) {}
        override fun onAdHidden(ad: MaxAd) {
            // Load lại ngay sau khi ad đóng
            rewardedAd.loadAd()
        }
        override fun onAdClicked(ad: MaxAd) {}
        override fun onAdLoadFailed(adUnitId: String, error: MaxError) {
            // Retry với backoff
        }
        override fun onAdDisplayFailed(ad: MaxAd, error: MaxError) {}
        override fun onRewardedVideoCompleted(ad: MaxAd) {
            // Bắt đầu trao thưởng — không đợi onUserRewarded
        }
        override fun onUserRewarded(ad: MaxAd, reward: MaxReward) {
            // User đã xem đủ — trao thưởng chính thức
            grantUserReward(reward.amount)
        }
    })
    rewardedAd.loadAd()
}
```

4.4 Các demand source nên thêm vào mediation

Network	Thế mạnh	Ghi chú
Google AdMob	Fill rate cao, reach rộng toàn cầu	Thường là mạng base, luôn phải có
Meta Audience Network	eCPM cao, targeting tốt	Cần app đã có lượt tải nhất định
Unity Ads	Mạnh với gaming, rewarded video tốt	Ưu tiên nếu app là game
ironSource	Rewarded + offerwall, gaming tốt	Cùng hệ sinh thái với LevelPlay mediation
Pangle (TikTok)	eCPM cạnh tranh ở SEA, trẻ hoá audience	Phù hợp thị trường Việt Nam
Mintegral	Mạnh ở Châu Á, playable ads tốt	Cân nhắc nếu target user Asian market

> *Không cần tích hợp tất cả ngay lập tức. Bắt đầu với AdMob + Meta là đủ. Thêm dần các mạng khác và theo dõi eCPM sau mỗi lần thêm để đo lường tác động thực tế.*

---

5. Test quảng cáo đúng cách trước khi release

5.1 Tại sao phải dùng test mode?

Google, Meta và hầu hết các ad network có hệ thống phát hiện invalid traffic tự động. Nếu bạn hoặc team click vào quảng cáo thật trong lúc test, hệ thống sẽ phát hiện và có thể:

- Khóa tài khoản AdMob vĩnh viễn
- Withhold payment (giữ tiền) cho tháng đó
- Đánh dấu tài khoản cần review — ảnh hưởng đến tất cả app

5.2 Cách bật test mode

Cách 1 — Dùng test Ad Unit ID (đơn giản nhất):

Google cung cấp sẵn test Ad Unit IDs chạy quảng cáo giả:

Format	Android Test Ad Unit ID	iOS Test Ad Unit ID
Banner	ca-app-pub-3940256099942544/6300978111	ca-app-pub-3940256099942544/2934735716
Interstitial	ca-app-pub-3940256099942544/1033173712	ca-app-pub-3940256099942544/4411468910
Rewarded Video	ca-app-pub-3940256099942544/5224354917	ca-app-pub-3940256099942544/1712485313
Native	ca-app-pub-3940256099942544/2247696110	ca-app-pub-3940256099942544/3986624511
App Open	ca-app-pub-3940256099942544/9257395921	ca-app-pub-3940256099942544/5575463023

Cách 2 — Đăng ký thiết bị test (dùng real Ad Unit ID):

```
// Android: Lấy test device ID từ log sau khi chạy app lần đầu
// Tìm dòng: "Use RequestConfiguration.Builder().setTestDeviceIds"
// Sau đó đăng ký:
val testDeviceIds = listOf("YOUR_DEVICE_ID_FROM_LOG")
val config = RequestConfiguration.Builder()
    .setTestDeviceIds(testDeviceIds)
    .build()
MobileAds.setRequestConfiguration(config)
```

5.3 Checklist test trước khi dùng real Ad Unit ID

1. Ad hiển thị đúng vị trí và kích thước trên nhiều màn hình khác nhau
2. Banner không che nội dung quan trọng (nút, form input)
3. Interstitial xuất hiện đúng điểm chuyển tiếp, không trong giữa action
4. Rewarded ad có thể skip sau khoảng thời gian quy định
5. Phần thưởng được trao sau khi xem hết rewarded ad
6. App không crash khi ad load fail (test bằng cách tắt wifi)

7. App không bị freeze hay lag khi ad đang load
8. Giao diện ad không bị tràn ra ngoài safe area trên iPhone có notch

---

6. Theo dõi và tối ưu: eCPM, Fill Rate, ARPDAU

Sau khi release, việc theo dõi đúng chỉ số giúp bạn quyết định khi nào cần thêm mạng, khi nào cần điều chỉnh tần suất, và format nào đang hoạt động tốt.

6.1 Các chỉ số cốt lõi

Chỉ số	Công thức	Benchmark tham khảo	Hành động khi thấp
eCPM	Revenue / Impressions × 1000	Game: $3–$15 | Utility: $1–$5 | News: $0.5–$2	Thêm mạng vào mediation, bật bidding
Fill Rate	Ads served / Ads requested × 100%	Nên đạt trên 85%	Thêm demand source, kiểm tra cấu hình mediation
ARPDAU	Revenue / Daily Active Users	Casual game: $0.02–$0.08 | Utility: $0.005–$0.02	Tối ưu ad placement, tăng session length
Impression per DAU	Impressions / DAU	Game: 5–15 | Utility: 2–6	Điều chỉnh tần suất và trigger hiển thị ad
CTR	Clicks / Impressions × 100%	Banner: 0.2–0.5% | Native: 0.5–1.5%	Thử nghiệm vị trí và kích thước ad khác nhau

6.2 Ví dụ tính ARPDAU thực tế

Giả sử casual game của bạn: 10.000 DAU, doanh thu AdMob ngày hôm qua là 80 USD. ARPDAU = 80 / 10.000 = $0.008. Benchmark casual game là $0.02–$0.08 — vậy bạn đang thấp hơn thị trường.

Bước tiếp theo: kiểm tra Fill Rate (nếu thấp → thêm mạng), kiểm tra Impression per DAU (nếu thấp → user ít xem ad, xem lại ad placement), kiểm tra eCPM (nếu thấp → bật mediation bidding).

6.3 Tần suất hiển thị — Cân bằng doanh thu và retention

Đây là điểm nhiều developer mắc sai lầm nhất: tăng tần suất quảng cáo để tăng impression, nhưng retention giảm mạnh hơn mức tăng doanh thu.

- **Interstitial:** Không hiển thị quá 1 lần mỗi 3–5 phút session time. Thêm cooldown tối thiểu 180 giây giữa hai lần hiển thị.
- **Rewarded Video:** Không cần giới hạn cứng — user tự chọn. Nhưng đừng đặt phần thưởng quá thấp làm user cảm thấy bị ép xem quá nhiều.
- **Banner:** Refresh tự động mỗi 30–60 giây là chuẩn. Đừng rút ngắn refresh interval vì tăng impression nhưng giảm eCPM do traffic kém chất lượng.
- **App Open:** Giới hạn 1–2 lần mỗi phiên. Hiển thị mỗi lần user mở app sẽ gây khó chịu nhanh chóng.

7. Khai báo trên store khi submit

7.1 Google Play — Data Safety Form

Kể từ năm 2022, Google yêu cầu tất cả app phải khai báo dữ liệu thu thập trong mục Data Safety trên Play Console. Khi tích hợp AdMob và các ad network, bạn cần khai báo:

- Location: Approximate location — nếu mạng dùng để target theo vị trí
- Device or other IDs: Advertising ID — dùng để phân biệt thiết bị
- App activity: App interactions — thông tin về cách user dùng app
- Personal info: nếu mạng nào thu thập email hoặc thông tin cá nhân

> *AdMob tự động thu thập một số dữ liệu không cần khai báo thêm. Tuy nhiên, kiểm tra trang Data Safety guidance của Google để đảm bảo không thiếu khai báo.*

7.2 App Store — Privacy Nutrition Labels

Apple yêu cầu khai báo chi tiết về dữ liệu thu thập trong App Privacy section khi submit. Với AdMob và các ad network, thường cần khai báo:

- Device ID — Advertising ID — dùng cho advertising
- Precise or Coarse Location — nếu app request location permission
- Purchase History — nếu dùng cho re-targeting quảng cáo
- Crash Data và Performance Data — từ Firebase/Analytics SDK

> Khai báo sai hoặc thiếu trên App Store có thể dẫn đến app bị reject. Apple có quyền yêu cầu bổ sung khai báo sau khi app đã lên store.

Kết luận: Lộ trình theo giai đoạn DAU

Không có một cấu hình quảng cáo nào phù hợp cho mọi giai đoạn tăng trưởng. Dưới đây là lộ trình khuyến nghị dựa trên quy mô:

Giai đoạn	DAU	Stack khuyến nghị	Mục tiêu chính
Mới launch	< 500	AdMob đơn lẻ, chỉ Rewarded + Banner	Validate UX, đo retention trước khi tối ưu revenue
Tăng trưởng	500–5.000	AdMob + Meta qua AdMob Mediation, thêm Interstitial	Tăng fill rate, so sánh eCPM giữa 2 mạng
Scale	5.000–50.000	AppLovin MAX với In-app Bidding (AdMob + Meta + Unity)	Tối ưu eCPM qua bidding, A/B test ad placement
Mature	> 50.000	MAX hoặc LevelPlay với 5+ networks, floor pricing	Maximize ARPDAU, phân tích LTV theo nguồn user

Ba điều cần nhớ khi xây dựng chiến lược monetization qua quảng cáo:

- **Retention trước, revenue sau:** Một app giữ chân được user 7 ngày sẽ kiếm nhiều tiền hơn gấp nhiều lần so với app chỉ giữ được 1 ngày dù có nhiều quảng cáo hơn.
- **Mediation là khoản đầu tư bắt buộc khi scale:** Tăng 20–40% eCPM chỉ bằng cách bật in-app bidding mà không cần thay đổi trải nghiệm người dùng.
- **Đo lường liên tục:** ARPDAU, Fill Rate, và eCPM cần được theo dõi ít nhất hàng tuần. Một mạng quảng cáo có thể giảm giá đột ngột — mediation giúp bạn không bị phụ thuộc vào một nguồn duy nhất.

Bài viết liên quan: Hướng dẫn tích hợp mạng quảng cáo và tracking cho ứng dụng Android & iOS — User Acquisition, MMP Attribution, và tối ưu CPI/ROAS.