



# The Ad Monetization Foundation

How to Build, Optimize, and Scale  
Ads in Mobile Games and Apps

**100+**

Studios Supported

**€2B+**

Revenue Generated

**€250M+**

From Ad Monetization

FREE GUIDE

# The Ad Monetization Foundation

---

How to Build, Optimize, and Scale Ads  
in Mobile Games and Apps

**100+**

Studios Supported

**€2B+**

Total Revenue Generated

**€250M+**

From Ad Monetization

## INTRODUCTION

# Building Ad Monetization That Scales, Not Just “Works”

If you're reading this, you've likely discovered what most mobile developers learn the hard way: ad monetization is deceptively complex.

On the surface, it seems straightforward. Integrate a few SDKs, turn on some ads, and watch the revenue roll in. But within weeks, sometimes days, the cracks begin to show. Fill rates plateau. eCPMs stagnate. Users complain about poor ad experiences. And despite following “best practices” from outdated blog posts, your revenue per user remains stubbornly below industry benchmarks.

The reality is that effective ad monetization operates at the crossroads of product design, technical implementation, user psychology, and market dynamics. A single misconfiguration in your mediation setup can cost you 20–30% of potential revenue without triggering any obvious warnings. An incorrectly implemented consent management platform can cut your European earnings in half. Treating all geos identically leaves money on the table in your highest-value markets.

This guide covers the ten core steps required to build a sustainable, scalable ad monetization foundation. These aren't theoretical concepts. They're the exact frameworks we use when auditing apps that are underperforming or preparing products for scale.

You'll notice something different about this guide: it doesn't promise that ad monetization is easy, and it doesn't pretend that reading a document is a substitute for experience. Instead, it gives you the knowledge to understand what's actually happening in your monetization stack, identify where your setup might be leaving revenue behind, and make informed decisions about whether to build this expertise in-house or work with specialists who do this every day.

Most app developers never optimize beyond their initial setup. They choose networks based on name recognition, deploy default mediation configurations, and assume that revenue performance is simply a function of traffic. By the time they realize their monetization infrastructure is underperforming, they've already left hundreds of thousands (or millions) on the table.

*The good news? Once you understand the system, the improvements compound quickly. Small optimizations in network competition, placement strategy, and compliance configuration routinely generate 30–50% revenue lifts. Proper segmentation and testing can double monetization efficiency in key markets.*

Whether you're preparing to launch your first monetized product or scaling an existing app with millions of users, this guide will help you avoid the most expensive mistakes and understand what truly drives ad revenue performance.

**Let's get started.**

#### ABOUT US

## Why Work With Us?

Reading this guide will give you a solid foundation. Implementing it correctly and continuously optimizing based on your specific user behavior, market dynamics, and product evolution requires dedicated expertise.

We've supported over 100 studios in launching and growing more than 100 new games, generating over €2 billion in total revenue. Our monetization strategies alone have directly brought in over €250 million. We know which configurations matter and which are just noise. We catch issues before they impact revenue, optimize faster than in-house teams building this knowledge from scratch, and deliver measurable results within weeks, not quarters.

**If you're serious about maximizing your ad revenue without the trial-and-error, let's talk.**

# Core Steps to Launching and Scaling Ad Monetization in Mobile Games and Apps

---

- 1 Select Ad Formats That Align With Product Design and User Behavior
- 2 Define Ad Placements as Part of UX, Not Just Monetization
- 3 Use Mediation to Introduce Competition for Your Inventory
- 4 Build a Diversified Ad Network Stack
- 5 Maintain SDKs With a Controlled Update Strategy
- 6 Apply COPPA Settings Only When Legally Required
- 7 Implement app-ads.txt to Protect Demand and Revenue
- 8 Enable sellers.json Where Supported
- 9 Integrate a CMP to Preserve GDPR-Compliant Revenue
- 10 Balance Brand Safety Controls With Revenue Impact

## STEP 1

# Select Ad Formats That Align With Product Design and User Behavior

---

Most developers choose ad formats the wrong way: they look at industry benchmarks, see that rewarded video has the highest eCPMs, and immediately start cramming rewards into their product without considering whether it actually makes sense.

Here's what actually matters: different formats monetize attention in fundamentally different ways, and forcing the wrong format into your product will damage retention faster than it generates revenue.

Rewarded video works when users can meaningfully exchange time for value, such as extra lives, in-game currency, or power-ups. The exchange has to feel fair, and the reward has to matter within your game economy. When this alignment exists, rewarded video delivers the highest eCPMs and the strongest user acceptance among all formats.

Interstitials generate significant revenue when placed at natural transition points where users expect a pause – level completions, session exits, and menu transitions. Place them randomly or too frequently, and you'll watch your retention metrics collapse.

Banners produce lower eCPMs but contribute steady incremental revenue in products with longer sessions or utility-focused use cases. The mistake most developers make is treating banners as "harmless" and slapping them everywhere without considering how they degrade the user experience.

These are the most popular formats, but they're far from the only options. Emerging formats like audio ads or less often used formats such as offerwalls can work exceptionally well in games and apps where traditional formats fall short: games with complex economies or even streaming apps and podcast players. The catch is that these formats require expert knowledge to implement correctly. Misconfigure an offerwall, and you'll destabilize your entire economy. Poorly integrated audio ads can disrupt the user experience even worse than an interstitial.

Format selection should be driven by session length, session frequency, progression mechanics, and the structure of your in-app economy. From a technical perspective, plan your formats early. Introducing new formats later often requires changes to placement logic, mediation configuration, and core UX flows – work you don't want to be doing when you're trying to scale.

### COMMON MISTAKES THAT COST YOU REVENUE

- Choosing formats based on what other apps do instead of how your users actually behave
- Overusing interstitials without measuring the retention impact
- Adding rewarded ads without building a balanced reward economy first
- Treating banners as harmless background revenue and placing them where they destroy UX

### WHY THIS MATTERS

Format decisions lock in your monetization patterns for the long term. Reversing them later isn't just expensive - it's often impossible without damaging the product.

## STEP 2

# Define Ad Placements as Part of UX, Not Just Monetization

Once you've chosen your formats, placement strategy becomes the primary driver of performance. This is where most monetization strategies actually fail.

The core principle is simple: ads should feel like a natural part of the experience, not an interruption. When placement is done correctly, users barely notice the friction. When it's done poorly, they notice immediately - and leave.

Interstitials should only appear at natural breakpoints where users mentally expect a pause. End of level. Session exit. Major transition. Never mid-action, never on a timer, never because "enough time has passed."

Rewarded ads must always be user-initiated. The user chooses to watch. They understand exactly what they're getting in return. The value exchange is clear before they commit. Anything less than this erodes trust and tanks engagement.

Banner placements need to respect screen layout, device variation, and orientation changes. A banner that works on an iPhone 15 might completely obscure critical UI on an older Android device.

From an implementation standpoint, placement logic needs to be predictable, modular, and built to support experimentation. The moment you hardcode ad triggers too closely to rendering logic or specific game states, you've limited your ability to optimize in the future.

### COMMON MISTAKES THAT COST YOU REVENUE

- Triggering interstitials based on timers instead of user flow
- Showing rewarded ads before users understand their value
- Hardcoding placement logic that prevents A/B testing later
- Overloading early sessions before retention patterns stabilize

### WHY THIS MATTERS

Placement problems don't show up as immediate revenue drops. They show up as slowly declining engagement and retention metrics that you won't connect to your ad strategy until it's too late.

### STEP 3

## Use Mediation to Introduce Competition for Your Inventory

If you're relying on a single ad network, you're leaving at least 30-40% of your potential revenue on the table. It's that simple.

Mediation enables multiple demand sources to compete for the same impression in real-time. More competition means better fill rates and higher eCPMs. Without mediation, you're taking whatever price your single network offers - and they know it.

Modern mediation platforms offer a more sophisticated setup: bidding logic, waterfall optimization, network prioritization, and performance tracking. The most common solutions are Google AdMob, Unity LevelPlay, and AppLovin MAX. They're similar in concept but differ significantly in reporting depth, SDK behavior, bidding support, and operational complexity.

**Here's the critical part most developers miss: If you're planning to scale, implement mediation from day one.**

### COMMON MISTAKES THAT COST YOU REVENUE

- Treating mediation as a one-time setup and never touching it again
- Using default configurations without AB testing for potential improvements
- Adding too many networks without validating their actual performance
- Completely ignoring bidding configuration

### WHY THIS MATTERS

Simple mediation misconfigurations suppress competition and silently cap your eCPMs across massive traffic segments. You won't know it's happening until you audit the setup.

#### STEP 4

## Build a Diversified Ad Network Stack

With mediation in place, your results depend almost entirely on which networks you've integrated and how you've configured them.

A strong initial stack typically includes global networks with broad demand, performance-focused networks optimized for specific verticals, and regionally strong demand sources for your key markets. Common starting points include:

AdMob

Meta Audience Network

AppLovin

UnityAds

Liftoff Monetize

ironSource

Mintegral

DT Exchange

But here's what matters more than the names: network performance varies dramatically by platform, geography, ad format, and user profile. A network that crushes it for rewarded video in the US might be completely worthless for interstitials in Southeast Asia.

This means your network stack can't be static. As your traffic scales and your user distribution changes, your network configuration needs to evolve with it.

#### **COMMON MISTAKES THAT COST YOU REVENUE**

- Choosing networks based on brand recognition instead of actual data
- Using identical priority configurations across all countries
- Keeping underperforming networks active “just in case”
- Never reassessing your stack as you scale

#### **WHY THIS MATTERS**

Network performance is dynamic, not static. A setup that works today will quietly degrade over time if you’re not actively managing it.

### **STEP 5**

## **Maintain SDKs With a Controlled Update Strategy**

Ad SDK maintenance directly impacts stability, compatibility, and revenue performance. Ignore it, and you’ll pay for it.

Outdated SDKs lead to lower fill rates, reporting inaccuracies, crashes, and missed access to new ad formats or bidding features. But updating immediately every time a network releases a new version is unnecessary and introduces risk.

The right approach is a controlled update cadence – typically every few months or aligned with your major app releases. Each update should be isolated, tested, and validated before going live to avoid disrupting your mediation behavior or introducing crashes.

#### **COMMON MISTAKES THAT COST YOU REVENUE**

- Delaying SDK updates indefinitely because “it’s working fine”
- Updating SDKs without proper testing
- Ignoring release notes that mention changes to bidding or compliance behavior
- Updating multiple SDKs simultaneously with no rollback plan

### WHY THIS MATTERS

SDK issues usually manifest as gradual revenue drops or instability. By the time you notice something's wrong, root-cause analysis becomes extremely difficult.

## STEP 6

# Apply COPPA Settings Only When Legally Required

COPPA configuration has a massive impact on monetization, and many developers get it wrong.

When COPPA is enabled, ad targeting is heavily restricted, and some networks stop serving ads entirely. The result is substantially lower eCPMs - often 50-70% lower than non-COPPA traffic.

COPPA should only be enabled if your app is explicitly directed at children and they represent your primary audience. That's it. For mixed-audience apps or general-audience apps, enabling COPPA "just to be safe" is one of the most expensive mistakes you can make.

### COMMON MISTAKES THAT COST YOU REVENUE

- Enabling COPPA without analyzing your actual audience composition
- Applying COPPA globally instead of conditionally based on age gates
- Assuming all ad networks behave the same way under COPPA restrictions

### WHY THIS MATTERS

Incorrect COPPA settings can reduce your monetization by an order of magnitude. We've seen apps lose 70-80% of their ad revenue because they enabled COPPA when they didn't need to.

## STEP 7

# Implement app-ads.txt to Protect Demand and Revenue

The app-ads.txt file verifies which partners are authorized to sell your ad inventory. It's a simple text file, but getting it wrong has serious consequences.

A properly implemented app-ads.txt file improves trust with buyers, reduces fraud risk, and is actually required by some networks to serve ads at all. Only networks you directly work with should be included, and the file needs to be updated whenever your partner relationships change.

#### **COMMON MISTAKES THAT COST YOU REVENUE**

- Including reseller lines without understanding your supply paths
- Forgetting to update the file after removing networks
- Publishing malformed or incorrectly hosted files that buyers can't parse
- Assuming your mediation platform manages app-ads.txt automatically (they don't)

#### **WHY THIS MATTERS**

An incorrect app-ads.txt implementation can block premium demand sources from bidding on your inventory - and the warning signs aren't obvious.

### **STEP 8**

## **Enable sellers.json Where Supported**

sellers.json improves supply path transparency at the network level by allowing buyers to validate who they're actually buying from.

Networks like AdMob, AppLovin, and Liftoff Monetize support sellers.json. While your direct involvement is often minimal, verification should still be part of your complete monetization setup.

#### **COMMON MISTAKES THAT COST YOU REVENUE**

- Assuming sellers.json requires no verification on your end
- Failing to confirm that your seller IDs and seller types are correct
- Overlooking sellers.json when diagnosing demand issues

### WHY THIS MATTERS

Transparency increasingly influences advertiser bidding behavior and long-term demand quality. It's a small detail that affects bigger outcomes.

## STEP 9

# Integrate a CMP to Preserve GDPR-Compliant Revenue

In GDPR-regulated regions, consent management directly affects how much money you make from each user.

Without a properly integrated Consent Management Platform (CMP), users in these regions receive limited ads with heavily restricted targeting. The result is significantly lower eCPMs - often 40-60% lower than properly consented traffic.

A CMP ensures you're compliant with privacy regulations while preserving access to premium demand. But the implementation matters: consent signals must be correctly passed to all ad SDKs and mediation layers, or you're still leaving money on the table.

### COMMON MISTAKES THAT COST YOU REVENUE

- Using a CMP that doesn't integrate properly with your ad SDKs
- Passing consent signals inconsistently across different networks
- Treating consent rates purely as legal metrics instead of revenue drivers
- Ignoring opt-in rate trends by geography

### WHY THIS MATTERS

Consent mismanagement impacts revenue just as much as it impacts compliance. Get it wrong, and you lose both ways.

## STEP 10

# Balance Brand Safety Controls With Revenue Impact

Brand safety controls should protect your user experience and brand positioning without unnecessarily limiting demand.

Blocking sensitive ad categories might be appropriate depending on your audience and brand - gambling ads in a kids' app, for example. But excessive blocking reduces auction competition and lowers fill rates. The same applies to advertiser-level blocking and app-level blocking.

The right approach is to block selectively, monitor both performance and user feedback, and revisit your decisions regularly as your product and audience evolve.

### COMMON MISTAKES THAT COST YOU REVENUE

- Blocking entire categories without actually reviewing what creatives look like
- Maintaining static blocklists that grow over time but never get trimmed
- Overreacting to isolated user complaints

### WHY THIS MATTERS

Over-blocking reduces competition in your auctions. Less competition means lower eCPMs. It's that direct.

## CONCLUSION

# The Cost of Waiting

You now understand the ten steps that separate functional ad monetization from revenue-optimized ad monetization. But understanding the theory and executing it correctly are very different things.

Every week you delay optimization is revenue you'll never recover. Misconfigured mediation waterfalls. COPPA settings that don't match your audience. Consent management that's blocking premium demand in your most valuable markets. These aren't hypothetical problems - they're the exact issues costing most apps 30-50% of their potential ad revenue right now.

## PROVEN RESULTS

### Results That Speak for Themselves

We don't just write guides about ad monetization - we execute these strategies daily for some of the industry's leading studios:

#### **PLAYSTUDIOS +20% eCPM**

Increased eCPM by 20% by optimizing their competitor blocking strategy - without increasing churn

#### **Tactile Games +101% Ad ARPDAU**

Achieved a 101% ad ARPDAU boost through mediation optimization and strategic in-game placement improvements

#### **New Moon Production Offerwall Success**

Successfully integrated offerwall monetization into flagship farm simulation games

These aren't marginal improvements. They're step-function changes in revenue performance - achieved by teams who stopped treating ad monetization as a "set it and forget it" system and started managing it as the complex, dynamic revenue engine it actually is.

## Here's What We Do Differently

We don't waste your time with generic advice or industry best practices that may or may not apply to your product. We audit your existing setup, identify exactly where you're losing revenue, implement the fixes, and continue optimizing based on your specific performance data.

**Most importantly, we move fast. Our clients typically see measurable revenue improvements within the first 2-4 weeks - not quarters.**

## Ready to Stop Leaving Money on the Table?

If you're serious about maximizing your ad revenue without the trial-and-error learning curve, let's talk.

**Schedule a free consultation, and we'll show you exactly where your setup is underperforming and what we'd do to fix it.**

WHO THIS GUIDE IS FOR

# Whether you built the system, manage it, or own the numbers.

## You built the integration, but the config isn't your job

Mediation tuning, waterfall logic, and consent signal plumbing aren't engineering problems. They're monetization problems that happen to live in your codebase.

## You see the revenue, but can't tell if it's 70% or 100%

When ad revenue is "fine," it's hard to know what you're leaving behind. This guide gives you the framework to evaluate your real performance.

## You're the one person making monetization work

Useful as a self-audit checklist, a resource to pressure-test your setup, or a way to build the case for changes you already know need to happen.

**30-50%**

of potential ad revenue is left on the table  
by most apps. Even profitable ones.



[www.gamebizconsulting.com](http://www.gamebizconsulting.com)