Clean Play Developer Implementation Guide

How to build, self-test, document, and maintain a Clean Play-ready mobile game

Clean Play Standards

August 6, 2026

Contents at a glance

Start here

This guide is for developers. It is not a substitute for the normative Standard 1.0 or an audit decision. Use it before implementation, before application, before every release, and when a service or monetization feature changes.

The fastest path is architectural: avoid ad networks, behavioral analytics, purchased random rewards, premium currency, manipulative retention, paid competitive advantage, hostile messaging, and opaque subscriptions from the beginning. Retrofitting them out later is harder.

The self-review loop

1. Inventory every client SDK, server service, remote flag, experiment, purchase, notification, event, account flow, and data field.
2. Map each item to one of the nine rules.
3. Remove or redesign obvious failures.
4. Capture the least sensitive evidence that proves the resulting behavior.
5. Test clean install, returning user, offline, account/guest, region, age, subscription, purchase, remote flag, and failure states.
6. Freeze the release scope and repeat after material changes.

Evidence pack structure

```
00-scope/
01-sdk-service-inventory/
02-data-flow-and-diagnostics/
03-ads-and-sponsorship/
04-accounts/
05-randomness/
06-events-timers-fomo/
07-currency-pricing/
08-competition-progression/
09-notifications/
10-subscriptions/
11-store-and-remote-config/
12-limitations-and-change-log/
```

Do not place passwords, signing keys, private cryptographic keys, customer databases, child data, or unrelated source code in the evidence pack.

CPS-01: No Disruptive Ads

Goal

Keep play uninterrupted and free from ad-network surveillance.

Build pattern

Remove third-party ad SDKs, banners, surprise interstitials, forced videos, progress-gated rewarded ads, fake close controls, and ad-like gameplay. Use direct prices, a normal premium unlock, static house promotion, or fixed sponsor credit with no tracking.

Minimum self-test

- test the current production-equivalent build;
- test all account, region, age, purchase, subscription, and remote-flag variants that can change the behavior;
- verify failure/offline states;
- verify store/server configuration, not only client UI;
- record expected and observed results.

Evidence checklist

SDK inventory; binary/dependency scan; representative play recording; store configuration; network sample; sponsor/house-promotion details.

Common trap

A supposedly optional rewarded ad still fails when it depends on an ad network, attribution, or user profiling.

CPS-02: No Tracking

Goal

Collect only what the product genuinely needs; do not profile players.

Build pattern

Avoid advertising IDs, cross-app attribution, fingerprinting, behavioral analytics, session replay, precise location, contact harvesting, and open-ended event telemetry. Prefer on-device logs, aggregate operational counts, short retention, and processors barred from independent reuse.

Minimum self-test

- test the current production-equivalent build;
- test all account, region, age, purchase, subscription, and remote-flag variants that can change the behavior;
- verify failure/offline states;
- verify store/server configuration, not only client UI;
- record expected and observed results.

Evidence checklist

Data-flow map; SDK/service inventory; privacy manifest/labels; permission use; event catalog; retention/deletion; vendor configuration/contract excerpts; network sample.

Common trap

A privacy-friendly vendor name does not pass by itself; actual configuration, fields, retention, reuse, and subprocessors control the result.

CPS-03: No Forced Accounts

Goal

Let people play basic or offline features without unnecessary identity collection.

Build pattern

Provide guest/local play by default. Limit accounts to genuine identity-dependent features such as cloud sync, multiplayer moderation, purchase restoration, or cross-device progress. Do not require marketing consent or social login.

Minimum self-test

- test the current production-equivalent build;
- test all account, region, age, purchase, subscription, and remote-flag variants that can change the behavior;
- verify failure/offline states;
- verify store/server configuration, not only client UI;
- record expected and observed results.

Evidence checklist

First-run recording; guest feature matrix; account data/retention; deletion flow; feature dependency rationale; parental/age controls where relevant.

Common trap

A device-local profile is not an account when it is not transmitted or used to identify the player externally.

CPS-04: No Loot Boxes

Goal

Do not sell or fund randomized rewards.

Build pattern

Remove paid gacha, mystery crates, paid keys, random card packs, paid spins, subscription-funded randomness, and ad-funded random rewards. Directly sell known content. Free procedural or earned randomness may remain when no purchase, ad, paid acceleration, or cash-out path reaches it.

Minimum self-test

- test the current production-equivalent build;
- test all account, region, age, purchase, subscription, and remote-flag variants that can change the behavior;
- verify failure/offline states;
- verify store/server configuration, not only client UI;
- record expected and observed results.

Evidence checklist

Economy diagram; purchase catalog; drop tables; acquisition paths; ad/subscription links; representative gameplay; server configuration.

Common trap

Publishing odds does not make paid randomness compliant.

CPS-05: No FOMO Mechanics

Goal

Do not punish absence or manufacture anxiety to force return or spending.

Build pattern

Remove punitive streak loss, fake countdowns, repeating “last chance” offers, permanent paid scarcity, monetized energy interruption, and threats of loss. Genuine events may use truthful dates, no punishment, and reasonable replay/return paths.

Minimum self-test

- test the current production-equivalent build;
- test all account, region, age, purchase, subscription, and remote-flag variants that can change the behavior;
- verify failure/offline states;
- verify store/server configuration, not only client UI;
- record expected and observed results.

Evidence checklist

Event calendar; offer configuration; timer logic; streak/energy rules; notification links; recordings across time/region/account states.

Common trap

A seasonal event is not automatically FOMO; the decisive question is whether design uses deception, punishment, or paid irreversible loss pressure.

CPS-06: No Premium Currencies

Goal

Show the real price of every purchase.

Build pattern

Price known items directly in local currency. Do not sell gems/tokens that obscure price, force mismatched bundles, create leftover balances, use multiple conversions, or expire purchased value. Earned-only gameplay points, score, and XP can remain.

Minimum self-test

- test the current production-equivalent build;
- test all account, region, age, purchase, subscription, and remote-flag variants that can change the behavior;
- verify failure/offline states;
- verify store/server configuration, not only client UI;
- record expected and observed results.

Evidence checklist

Store catalog; currency sources/sinks; conversion table; bundle math; expiration/refund behavior; screenshots in supported regions.

Common trap

A currency becomes problematic when real money enters the conversion chain, even if some units can also be earned.

CPS-07: No Pay-to-Win

Goal

Do not sell competitive superiority or relief from engineered frustration.

Build pattern

Keep purchases cosmetic, soundtrack/story/expansion content, or otherwise neutral to shared competition. Do not sell power, exclusive superior equipment, paid acceleration that dominates competition, or shortcuts around deliberately painful grind.

Minimum self-test

- test the current production-equivalent build;
- test all account, region, age, purchase, subscription, and remote-flag variants that can change the behavior;
- verify failure/offline states;
- verify store/server configuration, not only client UI;
- record expected and observed results.

Evidence checklist

Competitive rules; item/stat comparison; matchmaking; progression timing; purchase effect matrix; live-ops configuration; playtests.

Common trap

A single-player accessibility option available to everyone is not pay-to-win; selling it as relief may be exploitative.

CPS-08: No Hostile Notifications

Goal

Use notifications only for factual, requested, respectful information.

Build pattern

Avoid guilt, emotional dependency, false urgency, loss threats, repeated purchase pressure, manipulative timing, and default-on marketing. Use user-requested reminders, factual turn/security/download notices, category controls, and easy disablement.

Minimum self-test

- test the current production-equivalent build;
- test all account, region, age, purchase, subscription, and remote-flag variants that can change the behavior;
- verify failure/offline states;
- verify store/server configuration, not only client UI;
- record expected and observed results.

Evidence checklist

Full notification catalog; trigger logic; default settings; scheduling; localization; deep links; opt-in/disable flow; server templates.

Common trap

A technically true message can still be hostile when designed to exploit guilt or urgency.

CPS-09: No Hidden Subscriptions

Goal

Make recurring cost and cancellation unmistakable.

Build pattern

Show full price, cadence, trial conversion date/amount, included value, renewal, restore, and direct platform-management route. Do not hide recurring terms, use coercive cancellation screens, or require support when platform management exists.

Minimum self-test

- test the current production-equivalent build;
- test all account, region, age, purchase, subscription, and remote-flag variants that can change the behavior;
- verify failure/offline states;
- verify store/server configuration, not only client UI;
- record expected and observed results.

Evidence checklist

Store subscription configuration; paywall/trial/cancel recordings; localized price displays; restore flow; management link; renewal communications.

Common trap

On iOS/Android, the platform controls billing termination; the app must still make the management route clear and must not obstruct it.

Third-party services

A service is not permitted or prohibited by logo alone. Record:

- exact purpose and product feature;
- fields and identifiers sent;
- whether the provider acts only on your instructions;

- independent reuse, advertising, profiling, or cross-customer combination;
- retention and deletion;
- subprocessors and regions;
- contract/DPA terms;
- actual configuration;
- observed network behavior;
- fallback and removal plan.

Advertising networks, ad mediation, cross-app attribution, device fingerprinting, session replay, behavioral marketing analytics, and data brokers normally fail. Hosting, payment, transactional email, support, security, cloud save, multiplayer, and minimized crash processing can pass only when configured and contracted consistently with the standard.

Release checklist

Before every certified release:

- confirm package IDs, versions, store listings, and regions;
- diff SDKs and dependencies;
- diff privacy manifests/labels and permissions;
- diff purchase/subscription catalog;
- diff remote flags, experiments, events, streaks, energy, scarcity, and notifications;
- diff data fields, purposes, retention, and vendors;
- retest all affected rules;
- report material changes to Clean Play before release where practical;
- verify the public credential remains Active and in scope;
- update the in-app/static verification link if the credential changes.

Badge integration

- Website: use the credential-specific live badge and link to the exact registry page.
- App/store/press/print: use only the static **CHECK LIVE STATUS** asset.
- Show the credential ID and official .org verification route.
- Never bundle an Active badge or an API secret.
- Never describe the whole studio or catalog as certified.
- Stop mark use immediately when the registry is Suspended, Expired, or Revoked.

Application readiness test

You are ready to apply when you can answer every application question, provide the exact scope, explain every service and purchase path, produce a representative test build, provide evidence without exposing unnecessary secrets or people, and commit to material-change reporting.

Legal and scope note

This guide does not establish certification, legal compliance, platform approval, child-safety approval, accessibility conformance, or security certification. The adopted standard and final case decision control.

Formation-stage working material. Adoption, legal review, deployment, staffing, testing, and real records are required before production certification.