# Clean Play Certified - Badge, Registry, Credential, and Enforcement Master Specification

Design, technical stack, issuance, integration, status, security, and enforcement

Clean Play Standards

August 6, 2026

# Contents at a glance

**Document type:** Design-to-technology master specification
**Applies to:** Clean Play Standards 1.0: Mobile Games
**Public mark:** Clean Play Certified
**Source of current truth:** the public Clean Play registry

---

# 1. The simple public model

The public sees only:

1. **Clean Play Standards** — the organization.
2. **Clean Play Standards 1.0: Mobile Games** — the rulebook.
3. **Clean Play Certified** — the mark a specific game may use after passing an audit.

The program does not market separate pledge, member, signatory, verified, approved, bronze, silver, gold, or pilot labels. Application and case statuses are internal operating language, not public trust marks.

### Approved claim

**[Product] is Clean Play Certified to Clean Play Standards 1.0: Mobile Games for the exact platforms, package identifiers, versions, dates, and limitations shown in the public registry.**

### Claims that are not allowed

- “Clean Play approved company.”
- “Every game from this developer is certified.”
- “Guaranteed safe, private, ethical, or accessible.”
- “Government approved.”
- “Accredited” unless accreditation is later obtained and the precise scope is stated.
- “Permanent certification.”
- “Certified by blockchain.”

Certification is product-specific, scope-specific, time-limited, conditional, and revocable.

---

# 2. What the certification actually proves

An Active registry record means that Clean Play Standards completed its defined review process, the identified product met all applicable requirements at the decision point, the certification-mark license is in force, and no later status action has made the credential inactive.

It does **not** prove that:

- every future build is unchanged;
- the game has no bugs or security vulnerabilities;
- every player will have the same experience;
- the game satisfies every law or every accessibility, child-safety, privacy, or platform rule;
- the organization continuously observes every live session;
- a badge copied from somewhere is authentic.

The registry page must make the tested scope visible.

---

# 3. Certification status model

Only four statuses are public:

| Status | Meaning | Mark use |
|---|---|---|
| **Active** | Credential is current and the exact scope is authorized. | Allowed under the license. |
| **Suspended** | Authorization is temporarily paused while a serious issue is reviewed or corrected. | Must stop. |
| **Expired** | The credential reached its end date without successful renewal. | Must stop. |
| **Revoked** | Authorization was withdrawn after a final adverse decision or serious misuse. | Must stop immediately. |

Internal case statuses such as draft, submitted, evidence required, review in progress, remediation, decision review, and approved pending contribution are never displayed as certification.

### State transitions

```
Application → Audit → Independent decision
                            │
                            ├── Fail / Withdrawn → no credential
                            │
                            └── Pass → contribution or waiver → Active
                                                                  │
                                                                  ├── renewal → Active
                                                                  ├── deadline missed → Expired
                                                                  ├── urgent concern → Suspended
                                                                  └── final adverse action → Revoked
```

No public badge is issued before the independent decision, conflict checks, scope freeze, administrative completion, and authorized issuance.

---

## 4. The three badge uses

### 4.1 Dynamic web badge

Use on websites that can load a live SVG from Clean Play.

Example:

```
<a href="https://cleanplaystandards.org/verify/CPS-2026-000001"
   aria-label="Verify the current Clean Play certification status for Example Game">
  <img
    src="https://cleanplaystandards.org/api/v1/credentials/CPS-2026-000001/badge.svg"
    alt="Clean Play Certified — live status for Example Game"
    width="180" height="240" loading="lazy">
</a>
```

The endpoint renders the current state from the registry. It sends Cache-Control: no-store for current-status views. If the credential is unavailable, malformed, or cannot be checked, the safe response is **STATUS UNAVAILABLE — VERIFY**, never Active.

A signed domain-embed token may reduce casual hotlinking, but it is only a deterrent. HTTP Referer, Origin, CORS, DNS, and an HMAC query token do not prove that the party displaying an image owns a domain or is the certified app.

### 4.2 Static product and app-store badge

App-store screenshots, press kits, printed materials, and native binaries cannot change after publication. Their badge must therefore be status-neutral:

**CLEAN PLAY CERTIFIED — CHECK LIVE STATUS**

It includes the credential ID or verification short URL. A QR code may be used only when it resolves directly to the official .org verification page. Static artwork must never permanently say Active.

### 4.3 In-app certification screen

The simplest trustworthy mobile implementation is:

- a bundled status-neutral mark;
- the product name and credential ID;
- a plain-language scope statement;
- a button that opens the official verification URL;
- optional anonymous retrieval of current public status, cached briefly and never used to block launch or gameplay.

The game must not send player, device, account, advertising, or behavioral identifiers merely to show certification status.

A signed token stored in a game binary is copyable and replayable. A client secret embedded in a mobile app is not secret. Therefore, neither proves that a running app is genuine. The official registry and package/store scope remain the public proof system.

---

## 5. Visual system

### 5.1 Relationship between organization logo and certification mark

- **Organization logo:** open square frame containing a mint play triangle, with the words Clean Play Standards.
- **Certification mark:** a distinct rounded badge using the same symbol and palette, prominently labeled Clean Play Certified.
- The organization logo may identify Clean Play itself. It must not be used by applicants as proof of certification.

### 5.2 Core palette

| Token | Use | Hex |
|---|---|---|
| Deep Ink | primary background/text | #061626 |
| Secondary Ink | panels and navigation | #0B2338 |
| Clean Mint | mark accent and Active strip | #25D2A2 |
| Soft Mint | light backgrounds | #E7F8F2 |
| White | reverse text | #FFFFFF |
| Warning Amber | review/attention | #D97706 |
| Suspension Red | suspended/revoked signals | #B42318 |
| Neutral Gray | expired/unavailable | #667784 |

Status must always be written as text and may additionally use an icon or color. Color alone is never the status signal.

### 5.3 Geometry

- Standard badge ratio: **3:4**.
- Master digital artboard: **360 × 480 px**.
- Rounded outer corners: approximately **11% of badge width**.
- Clear space around the badge: at least the height of the play triangle’s shortest side.
- Minimum web width: **120 CSS px** for the full badge; 96 px for compact use only.
- Recommended web width: 160–220 CSS px.
- Minimum print width: **25 mm** for the full badge.
- Do not add shadows, gradients, outlines, rotations, or unapproved taglines.

- Never separate “Clean Play” from “Certified” or recreate the mark in another font.

### 5.4 File formats

- Web and product UI: SVG preferred; PNG at 1×, 2×, and 3× when SVG is unsuitable.
- Store screenshots: PNG.
- Print: SVG or press-ready PDF.
- Never distribute JPEG as the master mark because compression damages edges and small text.

### 5.5 Accessible alternative text

Dynamic example:

Clean Play Certified. Current status Active for Example Game. Credential CPS-2026-000001. Select to verify.

Static example:

Clean Play Certified mark for Example Game. Check current status at cleanplaystandards.org/verify/CPS-2026-000001.

---

## 6. Issuance workflow

1. Application is accepted and assigned a case ID.
2. Product scope is frozen: legal developer, product name, platforms, store URLs, package/bundle identifiers, current version and build, territories, and exclusions.
3. Conflict and independence checks are recorded.
4. Evidence is collected in the private case store.
5. Reviewer tests all nine requirements and documents findings.
6. Applicant resolves blocking findings and submits replacement evidence.
7. A qualified decision-maker who did not perform the audit reviews the complete case.
8. Pass/fail decision is recorded with date, rationale, and scope.
9. Only after a pass, the maintenance contribution or waiver is completed.
10. Issuance operator prepares a credential draft.
11. A second authorized person checks the scope, dates, URLs, status, and mark license.
12. The issuance service signs the credential and writes the Active public registry record in one controlled transaction.
13. Applicant receives approved mark assets and exact usage instructions.
14. Public registry becomes the source of truth.
15. Evidence is retained only for the defined period and then securely deleted.

The same person should not perform intake, the full audit, the final certification decision, and unrestricted credential issuance.

---

## 7. Public registry record

Every credential page must show:

- credential ID;
- public status in text;
- legal organization and public developer name;
- exact product name;
- platform or platforms;
- App Store/Google Play/public store URLs;
- bundle identifiers or package names when publication is appropriate;
- certified version/build or version range;
- standard and interpretation version;

- audit completion date;
- issue and expiry dates;
- material limitations or exclusions;
- last status update;
- link to the signed credential;
- how to report suspected misuse.

The registry should not expose private applicant contacts, internal evidence, reviewer notes, security findings, personal data, or confidential code.

### Verification URL

```
https://cleanplaystandards.org/verify/CPS-YYYY-NNNNNN
```

The page title and visible domain must make phishing harder. Verification pages should not redirect to applicant-controlled domains before showing the result.

---

## 8. Signed credential

### 8.1 Purpose

The signed credential provides issuer authenticity and tamper evidence for the credential content. It does **not** replace live status checking.

### 8.2 Format

Launch format: compact JWS using **ES256 / P-256** with a public key exposed through JWKS.

Illustrative payload:

```
{
  "iss": "https://cleanplaystandards.org",
  "sub": "clean-play-product:CPS-A-2026-EXAMPLE01",
  "credential_id": "CPS-2026-EXAMPLE01",
  "standard": {
    "code": "CPS-MG-1.0",
    "name": "Clean Play Standards 1.0: Mobile Games",
    "version": "1.0"
  },
  "product": {
    "name": "Example Game",
    "organization": "Example Studio LLC",
    "store_urls": ["https://apps.apple.com/example"]
  },
  "scope": {
    "platforms": ["iOS"],
    "package_ids": ["com.example.game"],
    "version_scope": "2.4.1 (241)",
    "regions": ["United States"],
    "limitations": []
  },
  "audit": {
    "decision_id": "D-EXAMPLE",
    "completed_at": "2026-08-01T16:00:00Z",
    "decision_digest": "64-character-lowercase-sha256-value",
    "public_summary": "Passed all nine requirements for the listed scope."
  },
  "issued_at": "2026-08-06T12:00:00Z",
  "expires_at": "2027-08-06T12:00:00Z",
  "status_at_issuance": "active",
  "status_url": "https://cleanplaystandards.org/api/v1/credentials/CPS-2026-EXAMPLE01/status",
  "registry_url": "https://cleanplaystandards.org/verify/CPS-2026-EXAMPLE01"
}
```


The JWS header contains alg, typ, and a kid that resolves to the correct public key.

### 8.3 Key controls

- Generate keys outside source control.
- Never commit a private key, production API token, service-account secret, or signing seed.
- Store the active private key in a protected runtime secret for the beta; move to stronger managed key custody as scale and risk increase.
- Restrict issuance to a dedicated route and role.
- Require two-person approval before signing.
- Log preparation, approval, signing, status changes, and key use.
- Maintain an offline recovery procedure and encrypted backup of authorized key material.
- Rotate keys on schedule and immediately after suspected compromise.
- Keep retired public keys available so historical signatures can be validated.
- Reissue or revoke affected credentials after a compromise, following a documented incident decision.

### 8.4 Live status beats signature

A valid signature means the issuer signed the data and it was not altered. It does not mean the certification is still Active. Every verifier must inspect the current status endpoint or registry page.

---

## 9. API surface

The deployable Worker uses these canonical public and applicant routes:

```
GET  /api/v1/health
GET  /api/v1/config
GET  /.well-known/jwks.json
GET  /api/v1/transparency
GET  /api/v1/registry
GET  /api/v1/credentials/{id}
GET  /api/v1/credentials/{id}/status
GET  /api/v1/credentials/{id}/jws
GET  /api/v1/credentials/{id}/badge.svg
GET  /api/v1/credentials/{id}/badge-static.svg
POST /api/v1/applications
GET  /api/v1/applications/{id}/status
GET  /api/v1/applications/{id}/portal
POST /api/v1/applications/{id}/messages
GET  /api/v1/applications/{id}/evidence
POST /api/v1/applications/{id}/evidence
POST /api/v1/applications/{id}/access/rotate
POST /api/v1/applications/{id}/contribution/checkout
POST /api/v1/complaints
POST /api/v1/contact
POST /api/v1/webhooks/stripe
```

Administrative routes are protected behind Cloudflare Access plus an allowlist and role checks. They cover assignments, findings, decisions, issuance, status changes, complaints, evidence access, deletion, and audit logs.

### Status response

```
{
  "credentialId": "CPS-2026-EXAMPLE01",
  "status": "active",
  "asOf": "2026-08-06T12:00:00Z",
  "expiresAt": "2027-08-06T12:00:00Z",
  "verificationUrl": "https://cleanplaystandards.org/verify/CPS-2026-EXAMPLE01"
}
```

No public status request needs a player identifier, device fingerprint, advertising ID, account token, or behavioral event.

---

## 10. Cloudflare implementation

### Public website and static assets

- Cloudflare Worker with Static Assets serves the site and API from one deployable project.
- HTML, CSS, JavaScript, SVG, icons, legal pages, resources, and downloads are static assets.
- The website is framework-free to reduce dependencies and operational overhead.
- No advertising analytics, session replay, or cross-site trackers are included.

### D1

D1 stores application metadata, hashed applicant access keys, applicant-safe case messages, findings, decisions, contributions/waivers, credentials, status history, complaints, contacts, administrative roles, and audit events. Confidential evidence bytes are not stored in D1.

### R2

A private R2 bucket stores applicant evidence. Object keys are random and case-scoped. There is no public bucket URL. Downloads require an authorized route, role check, and audit event.

### Cloudflare Access

The admin area is protected by Access and a server-side administrator allowlist. Access identity alone is not sufficient to authorize every action; application roles and separation-of-duties rules still apply.

### Turnstile

Turnstile protects public submission forms. Server verification is mandatory. A client widget alone is not a security control.

### Payments

Payment is requested only after a recorded passing decision. Stripe or another processor may host checkout. The payment webhook updates administrative status; it never creates a passing audit result by itself.

### Email

Transactional messages should be routed through a configured provider. Email must not contain confidential evidence. Portal access codes are shown once, stored as hashes, rotatable, and never logged in plaintext.

---

## 11. Website, portal, and admin behavior

### Public website

The public site contains:

- Home
- The Standard
- Certification Process
- Apply
- Applicant Portal
- Public Registry
- Credential Verification
- Good/Bad Examples
- Developer Resources
- Governance
- Transparency
- About
- Complaints
- Contact
- Terms, Privacy, Cookie/Storage, Accessibility, Vulnerability Disclosure, and Claims pages

### Application form

A human-friendly seven-section form captures:

1. Organization and contact
2. Product identity and platforms
3. Monetization and purchases
4. Ads, tracking, SDKs, accounts, notifications, and subscriptions
5. Evidence inventory and build access
6. Conflicts, related parties, and declarations
7. Review, consent, and submission

Browser autosave stays on the device. Server submission is explicit. Evidence upload occurs through the private portal after a case is created.

### Applicant portal

The applicant can see:

- case identity and scope;
- current internal progress in plain language;
- requested evidence;
- secure uploads;
- applicant-visible case messages;
- sanitized findings and remediation requests;
- decision outcome;
- post-pass contribution/waiver status;
- issued credential and mark instructions;
- renewal and material-change reporting.

### Administrator console

Authorized staff can manage:

- application queue;
- conflict screening;
- reviewer assignment;
- evidence requests and access;
- findings;
- independent decisions;
- credential preparation and second-person approval;
- suspensions, expirations, and revocations;
- complaints and appeals;
- deletion schedules;
- payments and waivers;
- audit logs and exports.

The admin interface is an operational tool, not a public CMS and not a substitute for documented procedures.

---

## 12. Threat model and controls

| Threat | Why it matters | Primary controls |
|---|---|---|
| Screenshot or copied badge | Image alone can be stolen | status-neutral static mark, credential ID, registry, license, monitoring, enforcement |
| Fake verification page | Users may trust a look-alike | official .org URL, prominent domain, signed credential, education, search monitoring |
| Dynamic-badge hotlink | Unapproved site can embed the endpoint | optional domain token as friction, visible product name/ID, registry scope; never call it proof |
| Stale Active badge | Cached image may outlive status | no-store, short edge cache where used, fail-safe unavailable state, live registry |
| Copied client token | Mobile binaries can be inspected | no secret-based proof model; registry and exact package/store scope |

| Threat | Why it matters | Primary controls |
|---|---|---|
| Signing-key compromise | Attacker could forge credentials | protected secret, issuance role, two-person approval, logging, rotation, incident plan |
| Admin takeover | Status or evidence could be changed | Access, allowlist, MFA, least privilege, audit logs, recovery, separation of duties |
| Evidence disclosure | Applicants may submit sensitive material | private R2, minimization, role-gated downloads, retention/deletion, no public URLs |
| Insider bias or conflict | Trust collapses if decisions are bought | conflict records, independent decision, founder-affiliated restrictions, no reviewer commission |
| Payment appears to buy approval | Undermines credibility | pass decision before payment, waivers, public fee policy, audit fee independent of outcome |
| Unauthorized mark use | Public deception | certification-mark license, monitoring, notice workflow, platform IP reports, counsel escalation |
| Service outage | Users cannot verify | graceful “status unavailable,” backups, restore tests, public incident notice; never default Active |

# 13. Enforcement and legal protection

## Mark license

Every certified product receives a narrow, nonexclusive, nontransferable, revocable license tied to:

- one legal entity;
- one named product;
- defined platforms/package identifiers;
- an Active credential;
- approved artwork and claims;
- an obligation to report material changes;
- immediate cessation after suspension, expiry, revocation, withdrawal, or termination.

## Monitoring

Monitor app stores, developer sites, press releases, social posts, and image search for the mark and key phrases. Record screenshots, dates, URLs, package identifiers, and archived evidence before contacting the user.

## Escalation

1. Confirm registry status and license scope.
2. Preserve evidence.
3. Send a clear correction/removal notice with a deadline.
4. Contact the host, marketplace, or platform through its applicable intellectual-property reporting channel.
5. Escalate serious or repeated cases to counsel.
6. Publish a status note only under the program’s documented transparency and fairness rules.

Trademark/certification-mark procedures are not the same as copyright/DMCA procedures. No document should promise that Apple, Google, a host, or a court will remove a listing immediately.

### Registration strategy

Before filing:

- complete federal, state, app-store, domain, web, and common-law clearance;
- decide whether the word mark, logo, and certification mark require separate applications;
- define the goods/services and certification characteristics accurately;
- adopt the certification rules and control process;
- obtain trademark counsel for filing and enforcement strategy.

Until registration is complete, describe the mark accurately and do not use the registration symbol.

---

## 14. Renewal and material changes

### Normal renewal

- Default credential term: one year unless the board adopts another term.
- Send renewal notices 90, 60, and 30 days before expiry.
- Reconfirm scope, SDK inventory, monetization, notifications, accounts, subscriptions, and store versions.
- Select risk-based retesting plus mandatory checks for every changed area.
- No renewal decision by silence.

### Material-change triggers

The certified developer must report changes involving:

- ad or analytics SDKs;
- account requirements;
- subscriptions, IAP, premium currency, loot/random reward systems, or pricing;
- notifications or engagement loops;
- online communication or user content;
- company ownership or responsible legal entity;
- package/bundle identifiers or store listing;
- major engine or backend changes affecting data or monetization;
- a regulator, platform, or security incident relevant to the nine requirements.

Clean Play decides whether the change is administrative, requires targeted review, or requires a full re-audit.

---

## 15. Acceptance tests before launch

Do not issue the first credential until all of the following pass:

1. An Active registry page renders the exact product scope.
2. Suspended, Expired, and Revoked states render clearly in text and color.
3. A status change propagates to the dynamic badge and public page within the adopted service target.
4. Static artwork says Check Live Status.
5. A valid ES256 credential verifies with the published JWKS.
6. A modified credential fails signature validation.
7. A valid but revoked credential still verifies cryptographically while the status endpoint correctly says Revoked.
8. A retired public key remains available for historical validation.
9. Private evidence has no anonymous URL.
10. Unauthorized admin users cannot read evidence or issue credentials.
11. An issuer cannot approve their own prepared credential without the second-person control.

12. Application, complaint, contact, and portal endpoints enforce validation, rate limits, and Turnstile where appropriate.
13. Backup and restore are tested.
14. Evidence deletion is tested and logged.
15. An expired portal code cannot be used and plaintext codes do not appear in logs.
16. A copied static mark never produces a false Active result without the registry.
17. Website pages are keyboard usable, responsive, and free of known advertising trackers.
18. The public claims page explains scope and limitations.
19. Complaint and appeal routes work before the first badge is issued.
20. Counsel has reviewed the certification agreement, mark license, claims, privacy, terms, and formation posture.

---

## 16. Launch recommendation

Use the server-hosted registry architecture, not NFTs or transferable tokens. Begin with:

- one standard;
- one certification mark;
- one public registry;
- one product credential;
- a dynamic web badge;
- a status-neutral static badge;
- ES256 signed credentials;
- a Cloudflare Worker/D1/R2 backend;
- a small, independent audit and decision process;
- strict mark licensing and public corrections.

This is simpler for developers, more revocable, more transparent, and more honest about what technology can and cannot prove.

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