

# Browser DRM Integration Cheat Sheet — EME on every browser

EME is the standard adapter, not a DRM. The CDM (proprietary, in the browser) decrypts; your license server issues keys. No single CDM covers the web, so plan multi-DRM. Engineering guidance; confirm browser and CDM support live, it changes.

## 1 · THE EME HANDSHAKE (the browser DRM flow)

- encrypted event** -> browser hands the player encryption metadata (initData).
- requestMediaKeySystemAccess** -> pick a Key System; create MediaKeys; setMediaKeys. HTTPS only.
- createSession + generateRequest** -> CDM emits a sealed **message** (license request).
- POST to your license server** -> pass the response to the CDM with **update()** -> CDM decrypts, plays.

## 2 · ROBUSTNESS -> SECURITY LEVEL -> RESOLUTION

- SW\_SECURE\_\*** -> Widevine L3 (software) -> SD / 480p. Widely supported.
- HW\_SECURE\_CRYPTO** -> L2 -> up to HD.
- HW\_SECURE\_DECODE / HW\_SECURE\_ALL** -> L1 (hardware) -> HD and 4K.
- Request the tier the content needs, after detecting support.** Too high = playback fails.

## BROWSER DRM INTEGRATION READINESS — BEFORE YOU SHIP

Encrypt the catalogue once with CBCS Common Encryption, then wire three license paths — Widevine, PlayReady, FairPlay — usually through one multi-DRM vendor. In the player, negotiate MediaKeys before choosing format/codec, detect the CDM and its robustness, request the robustness the content tier requires (not a fixed maximum), and serve over HTTPS. Keep a fallback for older Safari (WebKitMediaKeys, com.apple.fps.1\_0) and provision the FairPlay server certificate. Decide entitlement on your server, never in JavaScript. Test on real browsers and devices, not just one desktop. The 2024 EME draft adds encryption-scheme detection and getStatusForPolicy (HDCP query) — useful but uneven; verify support. Confirm current browser and CDM behaviour; it changes.

## 3 · BROWSER -> CDM -> KEY SYSTEM

- Chrome / Firefox** -> Widevine -> com.widevine.alpha (desktop = software L3).
- Edge (Windows)** -> PlayReady (+ Widevine) -> com.microsoft.playready (SL3000 HW path).
- Safari (macOS / iOS)** -> FairPlay -> com.apple.fps. Needs a server certificate.
- Baseline** -> Clear Key (org.w3.clearkey) is test-only — it protects nothing.

## 4 · COMMON MISTAKES

- One-DRM encode** -> Safari (or Chrome) shows a black screen. Cover all three.
- EME = protected?** No. EME is the API; the CDM + license policy are the protection.
- Max robustness everywhere** -> fails on software-only desktop CDMs.
- Trust logic in the player** -> any viewer can edit JS. Entitlement lives on the server.