

Cross-Platform Player Strategy Worksheet

Plan the OTT device matrix: what to share, what stays native, and what to maintain — one page.

1. Device targets to cover

- Web browser (Chrome, Edge, Firefox, Safari).
- iOS / iPadOS and Android phones & tablets.
- Apple TV (tvOS), Fire TV, Android TV / Google TV.
- Roku, Samsung (Tizen), LG (webOS); consoles if in scope.

2. Share-or-build, per layer

- Content & backend: SHARED — one CMAF/cbcs encode, manifests, license + catalogue APIs.
- Business logic: SHARED library — sign-in, rows, history, entitlement, analytics rules.
- Application interface: per input idiom (TV remote / touch / mouse) — ~3, not per device.
- Player engine: NATIVE adapter per platform — the only part genuinely rewritten.

3. Player engine per platform

- Web: Shaka / hls.js / dash.js on MSE + EME (browser-only).
- Apple (iOS + tvOS): AVPlayer + FairPlay (one Swift player).
- Android + Fire TV + Android TV: ExoPlayer (Media3) + Widevine.
- Roku: SceneGraph Video node (BrightScript).
Tizen/webOS: AVPlay / webOS media.

4. Encode-once foundation (confirm)

- One CMAF (ISO/IEC 23000-19) package; no per-platform re-encode.
- cbcs Common Encryption (ISO/IEC 23001-7) — one encrypt, three DRMs.
- One HLS playlist + one DASH manifest over the same segments.
- Same keys issue Widevine, PlayReady, and FairPlay licenses.

5. Maintenance & certification

- OS-version matrix tracked per platform; minimum versions set.
- App-store re-certification cycles calendared (each store).
- DRM / security migrations watched (e.g. Samsung PlayReady 2026).
- Real-device lab of TVs & sticks; shared-core regression suite.

6. Team & QoE ownership

- Skills: web (JS), Apple (Swift), Android (Kotlin), Roku (BrightScript), HTML-TV (JS).
- One owner for the shared playback contract / descriptor.
- QoE instrumentation consistent across every client.
- Maintenance staffed as a standing line item, not a one-off.