

Плеер — единственная часть платформы, которой касается зритель. Закройте эти решения до релиза. Инженерное руководство; поведение библиотек плеера меняется от версии к версии — сверяйте с актуальной документацией.

1 · МЕДИА-КОНВЕЙЕР (основа)

- ☐ **Выберите движок, не пишите свой** — web: Shaka / hls.js / dash.js; iOS/tvOS: AVPlayer; Android/Fire TV: Media3.
- ☐ **MSE кормит плеер** — сегменты добавляются в SourceBuffer; media element играет из него (W3C MSE).
- ☐ **Сопоставьте кодеки с устройством** — не выбирайте rendition, который декодер устройства не тянет.
- ☐ **Вытесняйте просмотренное** — держите окно, а не весь тайтл, иначе QuotaExceededError на TB.

2 · АДАПТИВНЫЙ БИТРЕЙТ (ABR)

- ☐ **Выберите семейство ABR** — throughput, buffer-based (BOLA) или гибрид (большинство плееров в 2026).
- ☐ **Начните консервативно, затем растите** — быстрый первый кадр важнее идеальных первых 3 секунд.
- ☐ **Гасите переключения** — заметная чехарда качества хуже стабильного, чуть более низкого уровня.
- ☐ **Уважайте лимиты устройства** — ограничьте качество тем, что реально тянут экран и декодер.

ТЕСТ — MEDIA ELEMENT ЭТО НЕ ПЛЕЕР

Тег video в HTML5 (и встроенный просмотр на телефоне) играет один файл фиксированного качества и, на устройствах Apple, нативный HLS — и больше ничего. Разбор манифеста, переключение ABR, тюнинг буфера, восстановление сегментов, multi-DRM через EME и QoE-биконы добавляет движок плеера поверх. Почти никто из OTT-сервисов не пишет этот движок с нуля: web-команды оборачивают Shaka, hls.js или dash.js (или лицензируют коммерческий), а мобильные используют AVPlayer и ExoPlayer/Media3. Настоящая инженерия — это интеграция, тесты на каждом устройстве и тюнинг (buffering goal, семейство ABR, лестница восстановления), а не математика переключения, которая живёт в разделе Video Streaming. Тест готового плеера: на нестабильной мобильной сети он стартует менее чем за две секунды, молча восстанавливается после потерянного сегмента и сообщает о каждом stall, который не смог спрятать. Сверяйте поведение библиотек (Shaka bufferingGoal, режимы ABR dash.js, наименования Media3) с актуальной документацией перед запуском.

3 · БУФЕР + ВОССТАНОВЛЕНИЕ

- ☐ **Задайте buffering goal** — ~20-30 с для VOD, меньше для live; это секунды плохой сети, которые вы прячете.
- ☐ **Retry с backoff** — большинство edge-ошибок CDN временные; повторите запрос, прежде чем сдаваться.
- ☐ **Fallback rendition + gap jump** — перекачайте в другом качестве; перепрыгните дырку в таймлайне.
- ☐ **Fatal — в последнюю очередь** — перезагрузка из манифеста до экрана ошибки. Никогда не fatal на первом 404.

4 · DRM + ИЗМЕРЕНИЯ

- ☐ **Управляйте DRM через EME** — Widevine / PlayReady / FairPlay одним путём EME (W3C EME).
- ☐ **Encrypt once, licence many** — схема cbcs Common Encryption (ISO/IEC 23001-7) кормит все три DRM.
- ☐ **Заложите licence round-trip** — DRM добавляет задержку до первого кадра; измерьте её.
- ☐ **Шлите четыре метрики QoE** — время старта, ребуферинг, средний битрейт, ошибки.