

Мост WebRTC → HLS — decision sheet

Пять продакшен-паттернов рядом, бюджеты задержки и дерево решений 2026 года.

A. Пять паттернов моста — в одной таблице

Паттерн	Композиция	Live/Batch	Glass-to-glass	Гибкость макета	Стоимость	Лучший случай
1. Пер-трековая запись	Нет — пер-трековые файлы (.mjr / WebM).	Batch	Не live; файл через минуты-часы.	Нет до пост-процессинга.	Самая низкая	Телемед compliance, архивы тренингов, депозитии.
2. Server-side композитор	Мост-воркер; GStreamer / FFmpeg с compositor и hlssink2.	Live	2-5 с LL-HLS · 8-15 с classic.	Ограничена; изменения в коде.	Низкая на масштабе	Высоковолюмные стандарт-макеты, спорт-беттинг, live shopping 10k+ зрителей.
3. Headless-браузер	Мост поднимает headless Chrome с HTML/CSS-шаблоном.	Live	3-5 с LL-HLS · 8-15 с classic.	Безграничная; CSS/JS.	Средняя	Кастомные макеты, бренд-оверлеи, переходы, SaaS-конференции.
4. WebRTC → RTMP → live-энкодер	Мост композитрует; downstream-энкодер (MediaLive, Mux, IVS, Wowza) делает HLS.	Live	6-12 с типично.	Ограничена со стороны моста.	Низкая на мосту; энкодер отдельно	Команды, у которых уже работает кластер live-энкодеров.
5. Управляемый мост SaaS	Вендор скрывает всё; оператор выбирает шаблон.	Live	3-8 с типично.	Шаблоны вендора.	Самая высокая поминутно	Time-to-market; небольшой и средний объём; стандартные макеты.

B. LL-HLS Паттерн 2: бюджет задержки по стадиям (intra-DC, 200-мс части, буфер 2 части)

Захват у спикера 30 мс · pacing энкодера 30 · RTP спикер→SFU 40 · SFU→мост intra-DC 5 · декод моста (1 кадр) 33 · compositor 8 · кодирование 33 · закрытие 200-мс части 200 · origin→CDN edge 80 · буфер плеера (2 части) 400 · декод у зрителя 100 ⇒ Итого ≈ 959 мс. Удвоение буфера до 4 частей даёт ≈ 1,4-1,6 с; classic HLS с 4-с сегментами и 3-сегментным буфером ≈ 14-16 с.

C. Дерево решений (по порядку)

- B1. Запись смотрит живая аудитория или только хранится в архиве? → Только архив: [Паттерн 1 — Пер-трековая запись](#). Пер-трековые файлы; пост-процесс в HLS-VOD или MP4.
- B2. Live broadcast — сотни или тысячи зрителей? → Сотни: [Паттерн 5 — Управляемый мост / WHIP напрямую](#). HLS пропускаем, если хватает.
- B3. Тысячи+ зрителей — стандартный или кастомный макет? Стандарт на масштабе → [Паттерн 2 — GStreamer-композитор](#). Кастом → [Паттерн 3 — Headless-браузер \(HTML-шаблон\)](#). Уже есть кластер live-энкодеров → [Паттерн 4 — WebRTC → RTMP → live-энкодер](#).

D. Пять ловушек, в которые попадает каждая команда

- Пиньте мост на верхний слой — иначе congestion control у SFU молча просадит разрешение broadcast-хвоста.
- webrtcbin из GStreamer не заменяет SFU — мост должен говорить на goom-протоколе SFU, прежде чем RTP попадёт в pipeline.
- Cloudflare WHIP-in + HLS-out — не один шаг (середина 2026) — нужно идти через RTMP в Cloudflare Stream Live, если нужен HLS.
- RTMP-хоп в Паттерне 4 добавляет ~1,5 с поверх сегментации downstream-энкодера. Закладывайте 6-12 с glass-to-glass.
- Паттерн 3 избыточен для стандартных макетов — headless Chrome это 1,5-2 ГБ RAM на комнату. Паттерн 2 — треть от того, если макет не меняется.