

Hybrid stack architecture pattern card

One page. Pick the hybrid pattern that matches your product, then build the stack.

1. The four canonical patterns

Pattern A — Interactive top, broadcast tail

Live shopping, auctions, town halls, watch parties.

Interactive WebRTC or WHEP for host + on-camera buyers (≤ 50 publishers).

Broadcast LL-HLS or LL-DASH over CMAF for the passive audience (1k-10M).

Why this pattern Stream-to-chat lag > 10 s kills conversion; pure WebRTC at 50k viewers is 3-10 $\times$ the cost of LL-HLS over a commodity HTTP CDN.

Pattern C — Real-time core, recording tail

Conferencing, telemedicine, e-learning.

Interactive WebRTC SFU end-to-end (every live participant is interactive).

Broadcast HLS recording for VOD playback after the session.

Why this pattern WebRTC is the only fit for the live meeting; replay on commodity HTTP is $\sim 100\times$ cheaper than keeping a live SFU alive for replays.

Pattern B — Broadcast main, moderator preview

Premium sports, live news, regulated broadcast.

Interactive Private WHEP endpoint for the control room (10-100 viewers).

Broadcast LL-HLS or LL-DASH for the public viewer tier (millions).

Why this pattern Plain HLS would add 2-3 s of blind spot for the producer; a private WHEP preview removes it for the bounded audience that needs it.

Pattern D — MoQ trialled in parallel with LL-HLS

Future-leaning live (sports, betting, interactive).

Interactive Opt-in slice on Media over QUIC (draft-ietf-moq-transport-17, Jan 2026).

Broadcast LL-HLS over CMAF as the load-bearing path for everyone else.

Why this pattern MoQ is still an Internet-Draft and carries interop risk; parallel A/B builds operational muscle while LL-HLS keeps the load-bearing path proven.

2. The handoff inside the origin — package once, deliver many

The interesting engineering in a hybrid stack lives in the middle, at the moment the contribution stream becomes one or more delivery streams. Discipline: transcode the incoming RTMP/S / SRT / WHIP stream once into a CMAF (ISO/IEC 23000-19) fragmented MP4 source. From that single source, emit an HLS manifest, a DASH manifest, and (if needed) a WHEP egress endpoint. The audio and video bytes are identical on every leg — only the manifest format, the transport, and the framing differ. This is the inside-the-origin pattern of every modern hybrid stack.

3. The role-not-audience cost split

Pure WebRTC for everyone

50,021 concurrent SFU sessions at 4 Mbps.

Hourly cost on commercial WebRTC media-server tier: $\sim$ \$3,000 to \$9,000.

Latency for everyone: sub-second (overkill for the 50,000 in chat).

Hybrid WebRTC + LL-HLS

21 SFU sessions + 50,000 cacheable LL-HLS viewers.

Hourly cost: $\sim$ \$900 (LL-HLS CDN) + \$10 (SFU for 21 people) = $\sim$ \$910.

Latency by role: sub-second for 21 on camera, 2-4 s for everyone else.

Cost ratio: ~ 3 to 10 times in favour of the hybrid stack at the same product experience.

4. Re-architecting a pure-WebRTC build to a hybrid stack

Step 1 — Split the audience. Identify the smallest set of users who genuinely need sub-second latency (host, on-camera buyers, moderators, producer).

Step 2 — Add a CMAF + LL-HLS surface. At the origin, subscribe to the host's track from the SFU and re-encode it into a CMAF fragmented MP4 source. Emit an LL-HLS manifest. Deploy behind a commodity HTTP CDN.

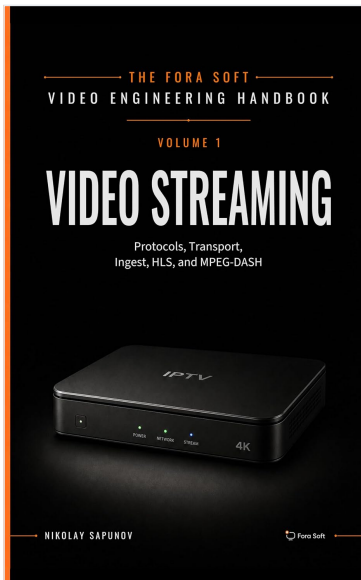
Step 3 — Keep chat real-time. WebSocket sidecar. The chat is not on the media path; do not give it a WebRTC-shaped latency budget.

Step 4 — Keep WebRTC for the interactive ring. Do not delete the SFU. The host and on-camera buyers still publish and subscribe through it.

Step 5 — Measure the bill. The SFU now serves 20-50 concurrent publishers and subscribers instead of 50,000; the LL-HLS fleet serves the 50,000 on commodity HTTP. The change is a 6-10 $\times$ reduction in egress cost at the same audience and the same product experience.

Companion to: 'The Switching Protocols Reality: Hybrid Stacks'

www.forasoft.com/learn/video-streaming/articles-streaming/hybrid-protocol-stacks



GO DEEPER · VOLUME 2 OF 7

Video Streaming: A Complete Guide to Video Delivery

From Network Protocols and Transport to Ingest, HLS, and MPEG-DASH

By Nikolay Sapunov, CEO at Fora Soft · 1,124 pages · Kindle & paperback on Amazon

Read it on Amazon →

amazon.com/dp/B0HH8F4R82

This card matches your product to one of four hybrid patterns and shows the package-once origin and cost split. Volume 2 explains the delivery layer underneath: ingest over RTMP, SRT and WHIP, CMAF packaging, LL-HLS and MPEG-DASH, and the latency budget for live.

WHAT YOU WILL FIND INSIDE

- Live latency budgets end to end
- CMAF packaging for HLS and DASH
- Ingest with RTMP, SRT and WHIP
- LL-HLS for large passive audiences

THE FULL HANDBOOK · 7 VOLUMES · 9,000+ PAGES · ALL ON AMAZON

Vol. 1 · Video Encoding — pixels and color to H.264, HEVC, AV1, quality metrics and AI

Vol. 2 · Video Streaming I — network transport, contribution and ingest, HLS, MPEG-DASH, CMAF

Vol. 3 · Video Streaming II — ABR, CDNs, players, WebRTC, DRM and the economics of delivery

Vol. 4 · Audio for Video I — PCM, codecs, loudness, accessibility, real-time voice processing

Vol. 5 · Audio for Video II — jitter buffers, lip sync, Atmos, quality metrics, neural codecs

Vol. 6 · AI for Video I — models and inference, computer vision, speech, search, generative video

Vol. 7 · AI for Video II — real-time pipelines, agents, serving, cost, compliance, verticals

Paying WebRTC prices for an audience that mostly watches?

We will split your audience by role, design the hybrid WebRTC and LL-HLS stack and estimate the hourly cost before you build. 250+ video products shipped since 2005.

Book a 30-min call →

WhatsApp →

Email us →