

WebTransport vs WHIP vs MoQ — 2026 cheat sheet

The three modern low-latency contribution paths, side by side, with the spec citations and the 2026 status flag.

WebTransport

W3C WD (25 Mar 2026) · draft-ietf-webtrans-http3-15 (2 Mar 2026, WGLC)

Layer: transport. Bytes in, bytes out.
 Runs on: HTTP/3 + QUIC + UDP.
 Modes: reliable streams (in-order, retransmitted) + unreliable datagrams.
 NAT traversal: none. Strict client → server.
 Media built in: no — pair with WebCodecs or MoQ.
 Baseline support: yes since Safari 26.4 (Mar 2026).
 Use when: browser-first ingest, you control the server, sub-second target.
 Risk: UDP/443 filtering, draft can still change.

WHIP (over WebRTC)

IETF RFC 9725 · Proposed Standard (Mar 2025)

Layer: contribution signalling. Wraps a WebRTC PeerConnection.
 Runs on: HTTP/1.1 or HTTP/2 + ICE + DTLS-SRTP + UDP.
 Modes: WebRTC's RTP for media; SCTP for data channels.
 NAT traversal: yes — ICE + STUN + TURN.
 Media built in: yes — full WebRTC stack with codec negotiation.
 Production-ready: yes; OBS 30+, Cloudflare, AWS IVS, Mux, Millicast.
 Use when: sub-second contribution today with no draft-track risk.
 Risk: TURN bandwidth costs; SDP and ICE complexity.

Media over QUIC Transport

draft-ietf-moq-transport-17 (2 Mar 2026)

Layer: media transport. Pub/sub on top of QUIC or WebTransport.
 Runs on: raw QUIC (native apps) or WebTransport (browser).
 Modes: streams + datagrams; intermediate relays for CDN-style fan-out.
 NAT traversal: not needed — client-server like WebTransport.
 Media built in: yes — but you choose the container (LOC, WARP, ...).
 Status: WG draft, ships in Cloudflare moq-rs, Ant Media, mengelbart/moqtransport.
 Use when: 12-24 month build window, willing to track drafts.
 Risk: spec churn; relay ecosystem still maturing.

Which protocol wins on which criterion (2026)

Criterion	WebTransport	WHIP / WebRTC	MoQ
Steady-state contribution latency	≈ 65 ms	≈ 95 ms	≈ 80 ms
Session setup time	1 RTT (QUIC 0-RTT possible)	1-3 RTT (ICE checks)	1 RTT
Connection migration on Wi-Fi→cellular	Yes (QUIC)	No (renegotiate)	Yes (QUIC)
Head-of-line blocking across streams	No	No	No
Reachability on networks that block UDP/443	Fails (no TCP fallback)	TURN-over-TCP fallback	Same as carrier
Browser support	Baseline since Mar 2026	Baseline since 2019	Depends on WT + WebCodecs
Standards maturity	WD + WGLC	Proposed Standard RFC	WG draft
Right answer in 2026 for...	Custom ingest you build	Production today	Aggressive 2027 bet

Decision tree (read top to bottom)

- Do you need sub-second latency from a browser this quarter?
 → Yes: use WHIP (RFC 9725). The ecosystem is there.
- Are you building a custom browser ingest stack and OK to operate a server?
 → Yes: use WebTransport + WebCodecs. Skip WebRTC entirely.
- Do you need CDN-style fan-out from a single live origin to millions?
 → Yes, with a 12+ month timeline: bet on MoQ over WebTransport.
- Network filters UDP/443 (some enterprise, some carriers)?
 → Plan a WebSockets or WHIP fallback path in advance.

Key spec citations (read these, not blog summaries)

W3C WebTransport WD, 25 Mar 2026 — www.w3.org/TR/webtransport/
 draft-ietf-webtrans-http3-15, 2 Mar 2026 — defines extended CONNECT, stream type 0x54, Session ID.
 draft-ietf-webtrans-overview-12, Jan 2026 — umbrella protocol framework.
 RFC 9725 (WHIP), Mar 2025 — Proposed Standard. The contribution path that ships today.
 draft-ietf-moq-transport-17, 2 Mar 2026 — §1.1: "MOQT runs over QUIC and WebTransport."
 RFC 9000 (QUIC), May 2021 — the transport under all of the above.
 RFC 9114 (HTTP/3), Jun 2022 — the HTTP layer WebTransport is carried on.
 RFC 9220 — Extended CONNECT and the :protocol pseudo-header.
 RFC 9221 — QUIC datagrams; mandatory for WebTransport.