# OTT Scaling & Capacity-Planning Worksheet

Fora Soft Learn

Run the concurrency arithmetic for your own audience, then check the architecture against the levers that bend the cost curve. Engineering guidance — CDN egress is tiered, commit-dependent, and 95th-percentile billed, so confirm your own rate and date the assumption.

## 1 · THE ARITHMETIC — fill in for your peak (concurrency is the number that matters)

- ☐ **Peak concurrency.** The highest number of simultaneous streams you expect — NOT registered users, NOT average. Size everything to this. _____
- ☐ **Per-stream bitrate.** ~4 Mbps for 1080p on a modern codec. Lower with HEVC/AV1. _____ Mbps
- ☐ **Throughput = concurrency × bitrate.** 1,000,000 × 4 Mbps = 4 Tbps. Your figure: _____
- ☐ **Egress per viewer-hour.** bitrate × 3,600 s ÷ 8 = ~1.8 GB/hr at 4 Mbps. Hourly cost = concurrency × 1.8 GB × your $/GB. At $0.04/GB and 1M viewers ≈ $72,000/hr.

## 2 · LEVERS THAT BEND THE CURVE (tick what your design already does)

- ☐ Cache-hit ratio (origin offload) targeted above 85% — ideally into the 90s for live.
- ☐ Origin shielding enabled so dozens of edge misses collapse to ~1 origin fetch per object.
- ☐ Delivery portable across more than one CDN (resilience + regional reach).
- ☐ Content steering (ETSI TS 103 998 / HLS EXT-X-CONTENT-STEERING) to move viewers between CDNs mid-stream — no bespoke player code.

## 3 · TWO PLANES, TWO PLANS (size them separately)

- ☐ **Data plane** sized in bandwidth (Tbps, egress GB): encode, package, origin, CDN, player. Lever: cache offload.
- ☐ **Control plane** sized in requests/sec: sign-in, entitlement, billing, ads, analytics. Lever: autoscaling.
- ☐ Entitlement is one service every player calls — not logic copied into each app.
- ☐ Low-latency live (LL-HLS / LL-DASH) multiplies small-request load — budget for it before enabling.

## 4 · BEFORE THE BIG NIGHT (the readiness gate)

- ☐ Did we forecast PEAK concurrency, separately for steady catalog and live spike?
- ☐ Is the control plane provisioned AHEAD of the arrival curve (predictive autoscaling)?
- ☐ Do we degrade gracefully — waiting room, retry/backoff — instead of hard-failing?
- ☐ Did we load-test the whole platform at the target peak in rehearsal, not in production?
- ☐ Is there sub-second automated failover between CDNs if one degrades during the spike?

## THE ONE RULE OF SCALING

Concurrency, not catalog size, sizes an OTT platform. The video path (data plane) scales with bandwidth and is tamed by cache offload — a high cache-hit ratio, origin shielding, and more than one CDN with content steering. The decide-and-record path (control plane) scales with requests per second and is tamed by provisioning ahead of the arrival curve, because a live premiere's thundering herd arrives in the same sixty seconds. Size each plane separately, engineer for failure, and load-test at peak before the event — never in production on the biggest night.