Latency, availability

Tuesday, 30 September 2025 10:26 PM

Target (Latency, Availability)	Core Delivery Choices	Core Data Choices	Core Compute/Infra	Trade-offs	Good For
Ultra-low, Very high	Multi-CDN, HTTP/3, edge caching + ISP appliances, pre-warm hot titles	Cassandra RF=3 (multi-region, LOCAL_QUORUM), Kafka RF≥3 acks=all, Redis global	Active-active multi-region, stateless services, autoscale, health-based geo DNS	Highest cost/ops complexity; consistency & cache invalidation are hard	Live sports, tier-1 VOD
Low, High	Single CDN + regional POPs, origin shielding, ABR 2–6s segments	Cassandra RF=3 single region + async x-region DR, Kafka RF=3, Redis per-region	Multi-AZ; warm standby in 2nd region	Lower cost than above, slower failover; some data loss risk on region loss	Mainstream VOD, large apps
Moderate, High	CDN with longer TTLs, fewer POPs, no ISP appliances	Primary DB with read replicas (MySQL/Postgres), limited Cassandra/Kafka	Multi-AZ; backup & restore tested; canary deploys	Cheaper; higher tail latencies; limited write scalability	Content sites, SaaS dashboards
Low, Moderate	Heavy edge caching, single CDN, short TTL manifests	Single-region Cassandra/Redis; Kafka RF=2	Single-region K8s; blue/green; no cross-region	Great p95 latency; region outage = downtime	Startups optimizing UX over HA
Moderate, Moderate	Basic CDN, standard HLS/DASH, longer segments (4–10s)	Single primary DB + cache; minimal streaming infra	Single-AZ+burst HA (auto-recreate)	Lowest complexity; higher jitter; maintenance windows	MVPs, internal tools with users
High, Very high	No edge tuning needed; downloads/batch	Strongly replicated DB (e.g., Spanner/Dynamo), Kafka RF≥3	Active-active control plane; strict SLOs for durability not latency	Rock-solid availability; sluggish UX	Financial ledgers, compliance systems
Ultra-low, Low	Pure edge compute, peer-to-peer/torrent-style fan-out, LL-HLS	In-memory only (Redis), eventual/async persistence	Single region/POP, no DR	Blazing fast when up; outages acceptable	Experiments, gaming tournaments
High, Low	Direct origin, no CDN	Single DB, nightly backups	Single instance/zone	Minimal cost & ops; frequent downtime	Prototypes, internal batch jobs