

Contents

Video Streaming System Design	1
Overview	1
1. Video Ingestion	1
2. Transcoding & Encoding	1
3. Chunking & Packaging	1
4. Storage & CDN Distribution	2
5. Playback & Adaptive Streaming (ABR)	2
6. Client-Side Buffering & Analytics	2
Optional: Offline Playback	2
Optional: Live Streaming Support	3
Architecture Diagram	3

Video Streaming System Design

Overview

This document outlines the architecture and data flow of a modern video streaming platform (e.g., Netflix, Prime Video). It covers ingestion, processing, adaptive delivery, playback, analytics, and optional features like offline and live streaming.

1. Video Ingestion

Studio/content provider uploads high-quality source video (ProRes, MXF) via secure CMS. This is often done via a content management system (CMS) and stored in cloud object storage. Uploads are authenticated and may use accelerated protocols for large files.

Key Technologies: - CMS: Internal tools, Brightcove, Mux, or custom UI - Storage: AWS S3, GCP Cloud Storage, Azure Blob - Upload Protocols: HTTPS, Aspera, FTP/SFTP - Authentication: OAuth2, IAM, signed URLs

2. Transcoding & Encoding

Raw video is transcoded into multiple resolutions and bitrates (e.g., 240p–4K) and encoded with modern codecs. Audio is encoded at multiple bitrates, and subtitles/captions are attached. This enables adaptive streaming for different devices and networks.

Key Technologies: - Transcoder Tools: FFmpeg, AWS Elemental, Bitmovin, Mux Video, Shaka Packager - Codecs: H.264, H.265, VP9, AV1 - Audio: AAC, AC-3, Dolby Atmos - Containers: MP4, MPEG-TS, fMP4

3. Chunking & Packaging

Video/audio streams are split into small chunks (2–6s) and packaged for streaming protocols like HLS and MPEG-DASH. Each quality level has a manifest file describing available chunks and

bitrates.

Key Technologies: - Packaging Tools: Shaka Packager, Bento4, AWS MediaPackage - Streaming Protocols: HLS (.m3u8), MPEG-DASH (.mpd)

4. Storage & CDN Distribution

All chunks and manifests are uploaded to cloud storage and distributed via CDN. CDN nodes cache video chunks near viewers to reduce latency and improve reliability.

Key Technologies: - Storage: S3, GCS, Azure Blob - CDNs: Akamai, CloudFront, Fastly, Netflix Open Connect - Access Control: Signed URLs, tokens

5. Playback & Adaptive Streaming (ABR)

When a user presses “Play”: - The player fetches the manifest file. - Based on bandwidth and screen, it chooses the best bitrate/resolution and downloads the first chunks. - While playing, it monitors bandwidth and buffer, upgrading or downgrading quality as needed. - DRM licenses (Widevine, FairPlay, PlayReady) may be fetched to decrypt protected content.

Key Technologies: - Client Players: HLS.js, Shaka Player (web); ExoPlayer (Android); AVPlayer (iOS); MSE - ABR: Adaptive Bitrate logic in player SDKs - DRM: Widevine, FairPlay, PlayReady; Google Widevine SDK, FairPlay Server, DRM Today

6. Client-Side Buffering & Analytics

The player buffers ahead to avoid pauses. If the network drops, it may pause, downgrade resolution, or retry fetching. Playback events are logged and sent to analytics services for monitoring quality of experience.

Key Technologies: - Buffer Management: Integrated in playback SDKs - Playback Metrics: Startup time, rebuffer ratio, resolution switches - Analytics Platforms: Mux Data, Conviva, New Relic, Datadog, Prometheus/Grafana

Optional: Offline Playback

Users can download DRM-encrypted chunks for offline viewing, with expiry and device restrictions.

Key Technologies: - Encryption: DRM-encrypted chunks - Storage: SQLite + file system - Player SDKs: Platform-specific with offline support

Optional: Live Streaming Support

Live video is ingested via RTMP and transcoded for adaptive delivery. Supports normal and low-latency modes, with live CDN distribution.

Key Technologies: - Ingest Protocols: RTMP → AWS MediaLive, OBS - Latency Modes: Normal (~30s), Low Latency (~3–5s) - Live CDNs: AWS IVS, Akamai Live, Wowza

Architecture Diagram

You can edit this diagram by uploading the PNG to [Excalidraw](#).

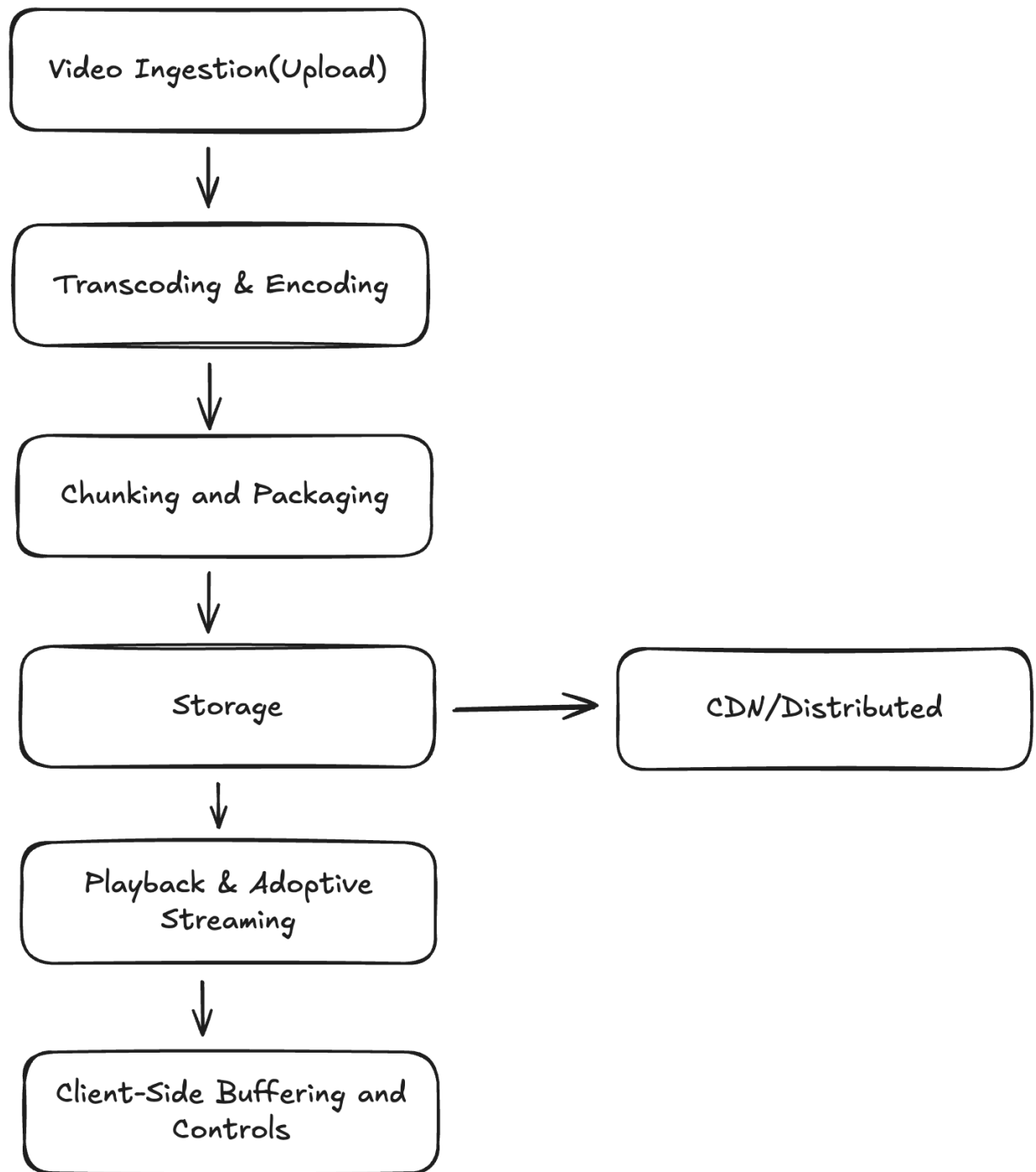


Figure 1: Video Streaming System Diagram