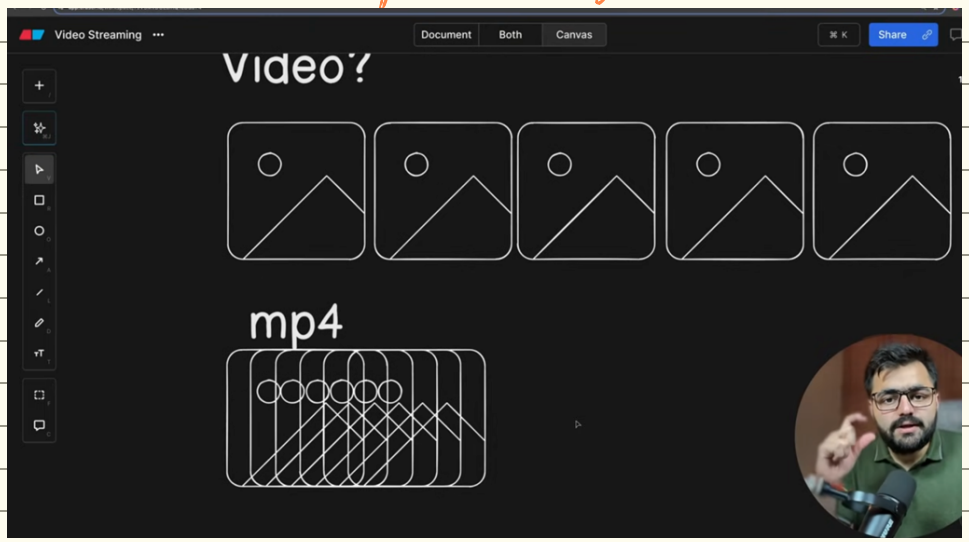
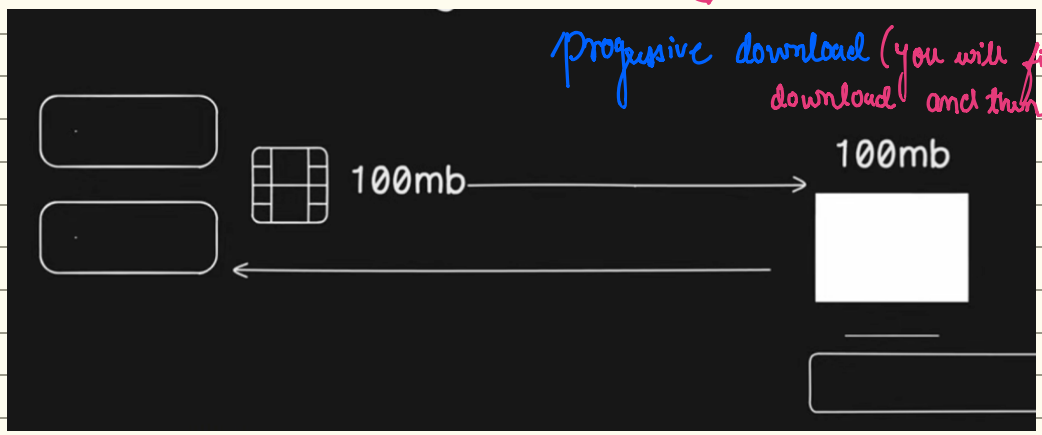
SYSTEM DESIGN

How video Streaming works on Scale.

→ **Videos:-** Video are sequence of photos and frames in motion



in early 2000s :- Streaming platforms



→ **Streaming :-**

RTMP
RTSP → Specialized Streaming protocol.

∵ RTMP:- Real time messaging protocol.

∵ RTSP:- Real time streaming protocol.

Low latency.

Live Streaming support.

Efficient use of Bandwidth.

(i). RTMP was developed by ADOBE

(ii). RTSP was developed by. RealNetworks.

→ Adaptive Bitrate Streaming:-

Adaptive Bitrate Streaming (ABR) is a video streaming technique that automatically adjusts the video quality in real-time based on the viewer's current network speed and device performance.

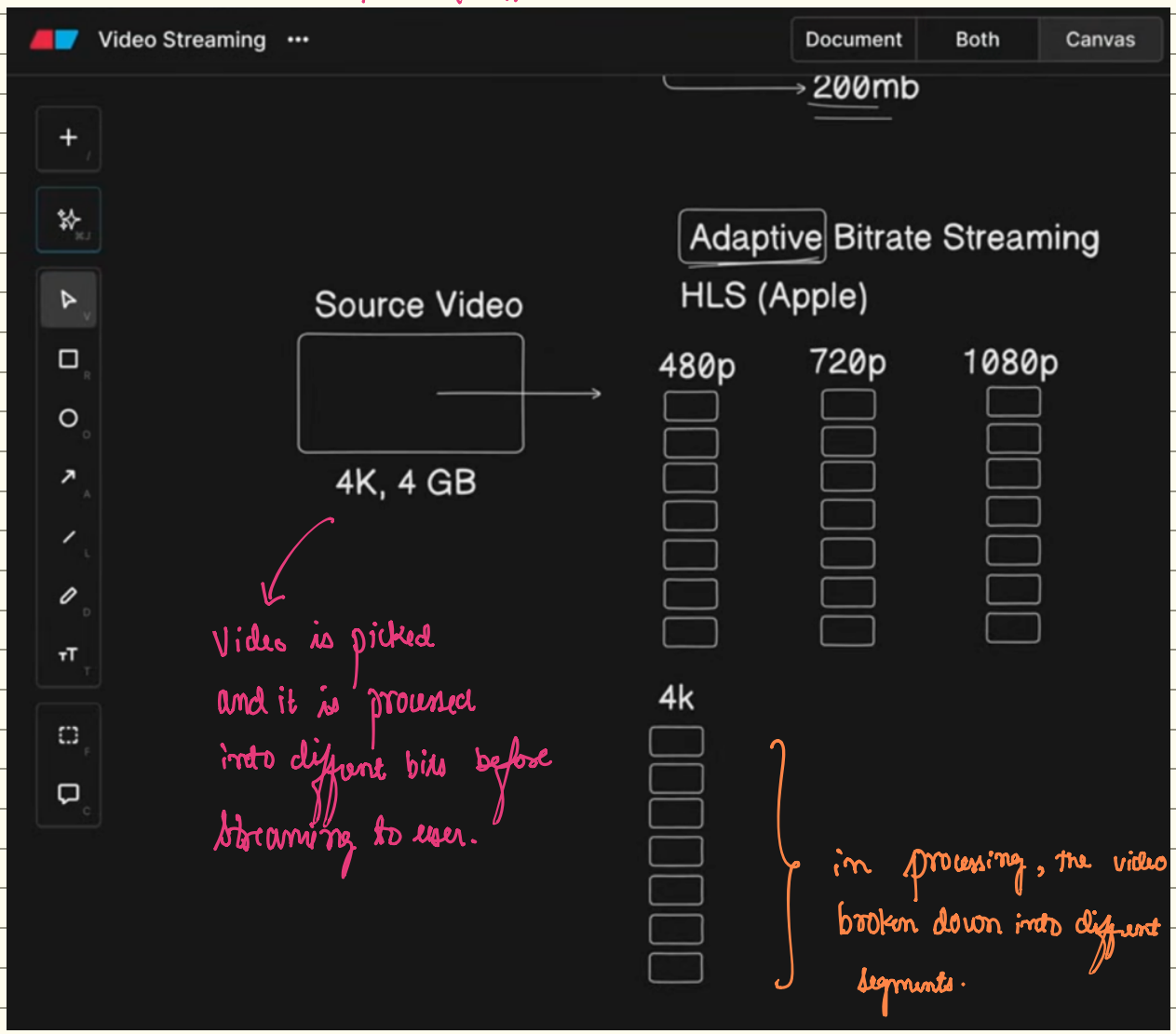
🎥 How It Works

When you stream a video:

1. The video is **divided into small chunks** (segments), each a few seconds long.
2. Each segment is **encoded in multiple quality levels** (e.g., 240p, 480p, 720p, 1080p).
3. The player (on your phone/laptop) continuously checks:
   - **Internet speed**
   - **Buffer health**
   - **CPU/device capability**
4. Based on current conditions, the player **dynamically switches** between quality levels.

So if your network slows down → video automatically shifts to lower quality (prevents buffering).
If your network improves → quality jumps back to HD/4K.

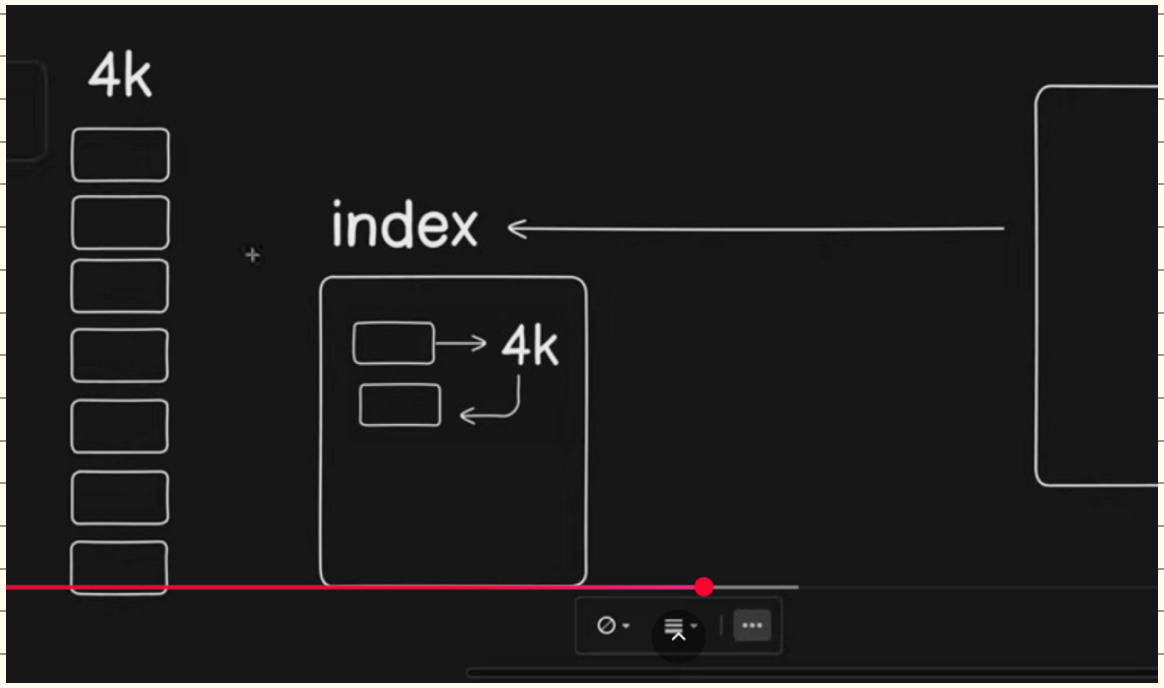
HLS (developed by Apple)



→ in processing, it encodes the video into diffent segments.

→ index.m3u8 :-

An index.m3u8 file is a manifest file used in HLS (HTTP Live Streaming).
It does not contain video data itself — instead, it lists and organizes the video segments that a video player should download and play.



The index file tracks which segment is of which format and where is its next segment.

Adaptive Bitrate Streaming

Learn how to use Adaptive Bitrate Streaming for long videos with ImageKit.

Copy page

Adaptive Bitrate Streaming (ABS) enables the optimum streaming video viewing experience for different types of devices over a broad set of connection speeds. This results in very little buffering, a fast start time and a good experience for both high-end and low-end connections.

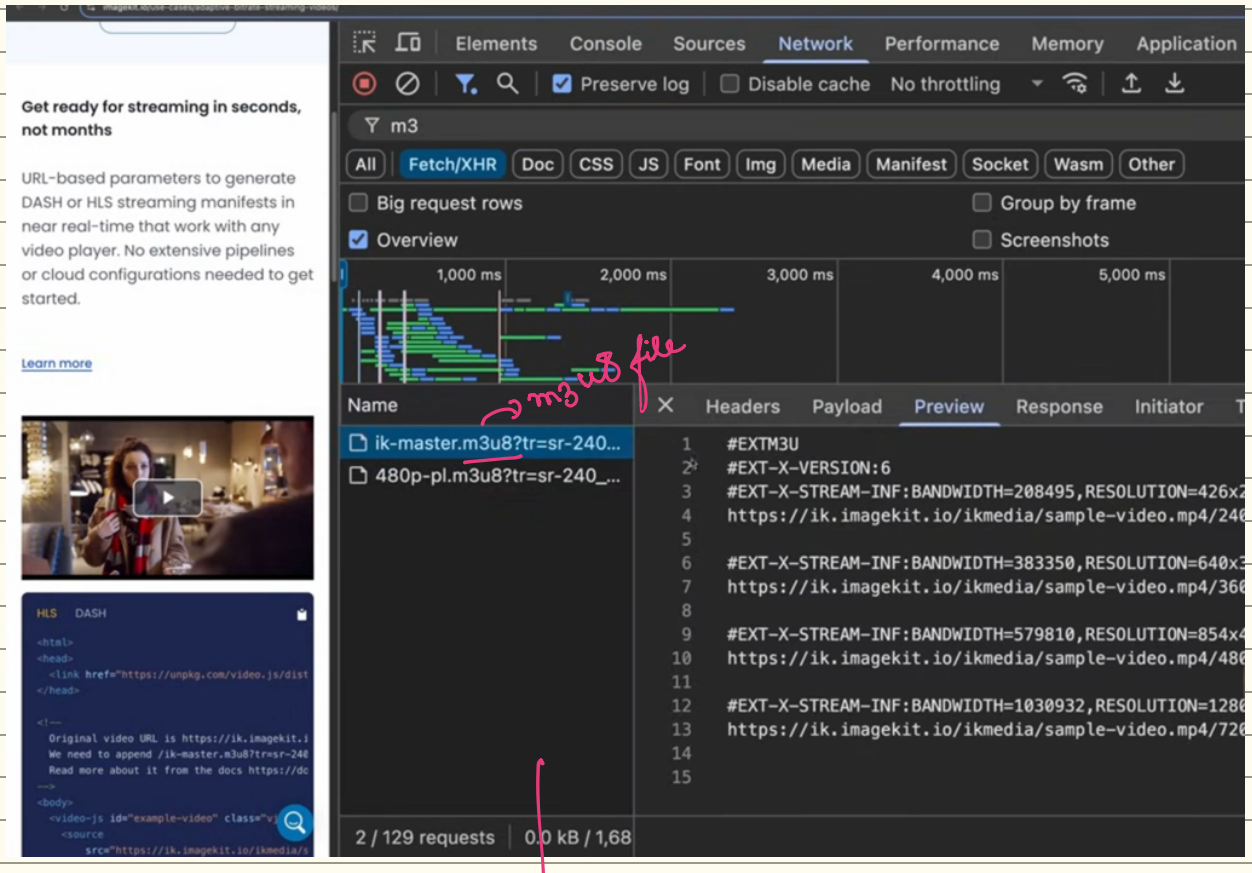
The client (e.g. a video player) loads the manifest file and then chooses an appropriate segment, usually starting from the lowest bit rate stream to speed up initial playback. If network throughput is greater than the bit rate of the downloaded segment, it requests a higher bit rate segment. The client continues to adapt to changes in network throughput to ensure a smooth playback experience. The exact algorithm of choosing which segment to load can vary from client to client, but fundamentally it remains the same -- choose the appropriate segment and adapt as network (or other device constraints) changes.

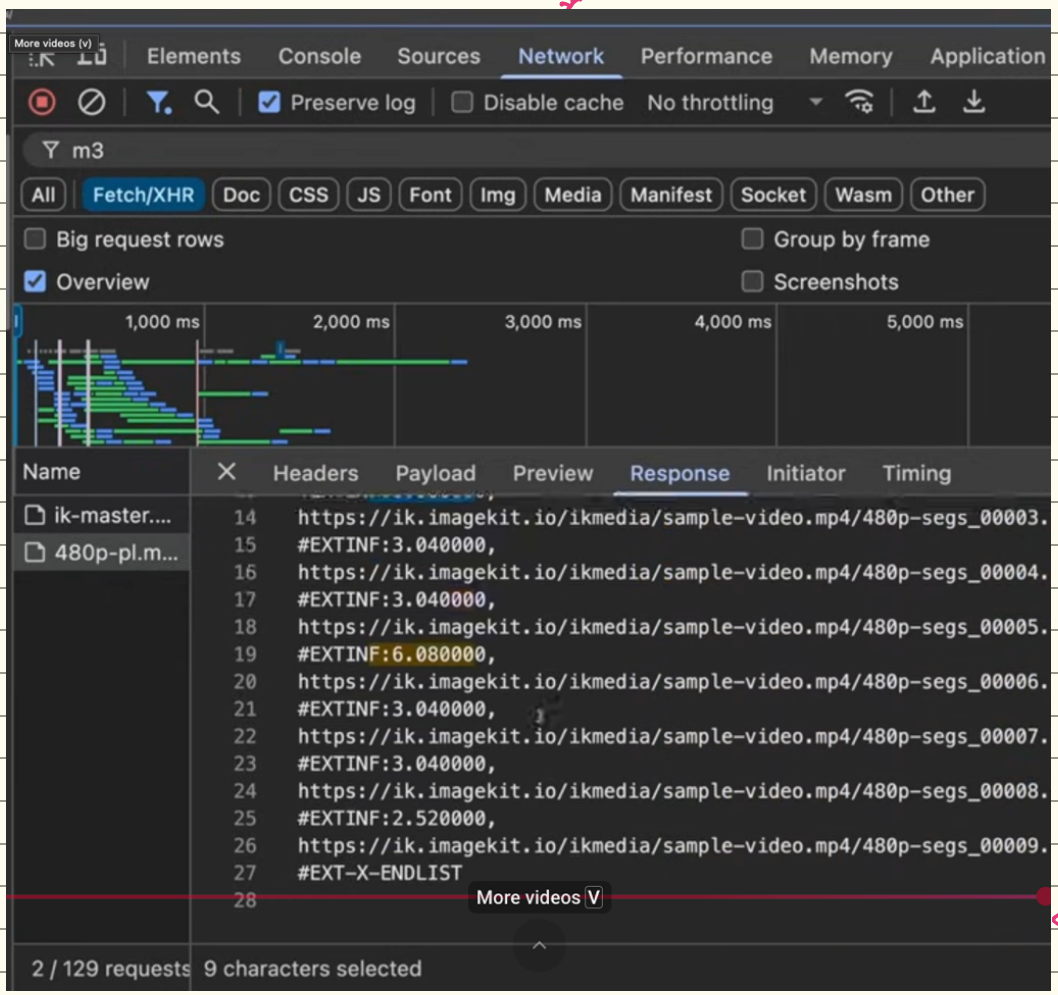
For ABS to work, the client needs a manifest file that contains information about segments of different variants at varying bitrates. ImageKit can generate and deliver all necessary variants and manifest files from a single source video that is accessible through your ImageKit's account. The original video can be hosted in the ImageKit Storage or external storage integrated with ImageKit. Extra storage created because of generated variants and manifest files is counted towards your media library storage.

ImageKit supports the following streaming protocols. Both leverage existing HTTP infrastructure including CDN caching.

- HTTP Live Streaming (HLS)
- Dynamic Adaptive Streaming over HTTP (MPEG-DASH) protocol

imagekit.io → has simplified Adaptive streaming.





Segments of a 480p file.