**Live video broadcasting system design**

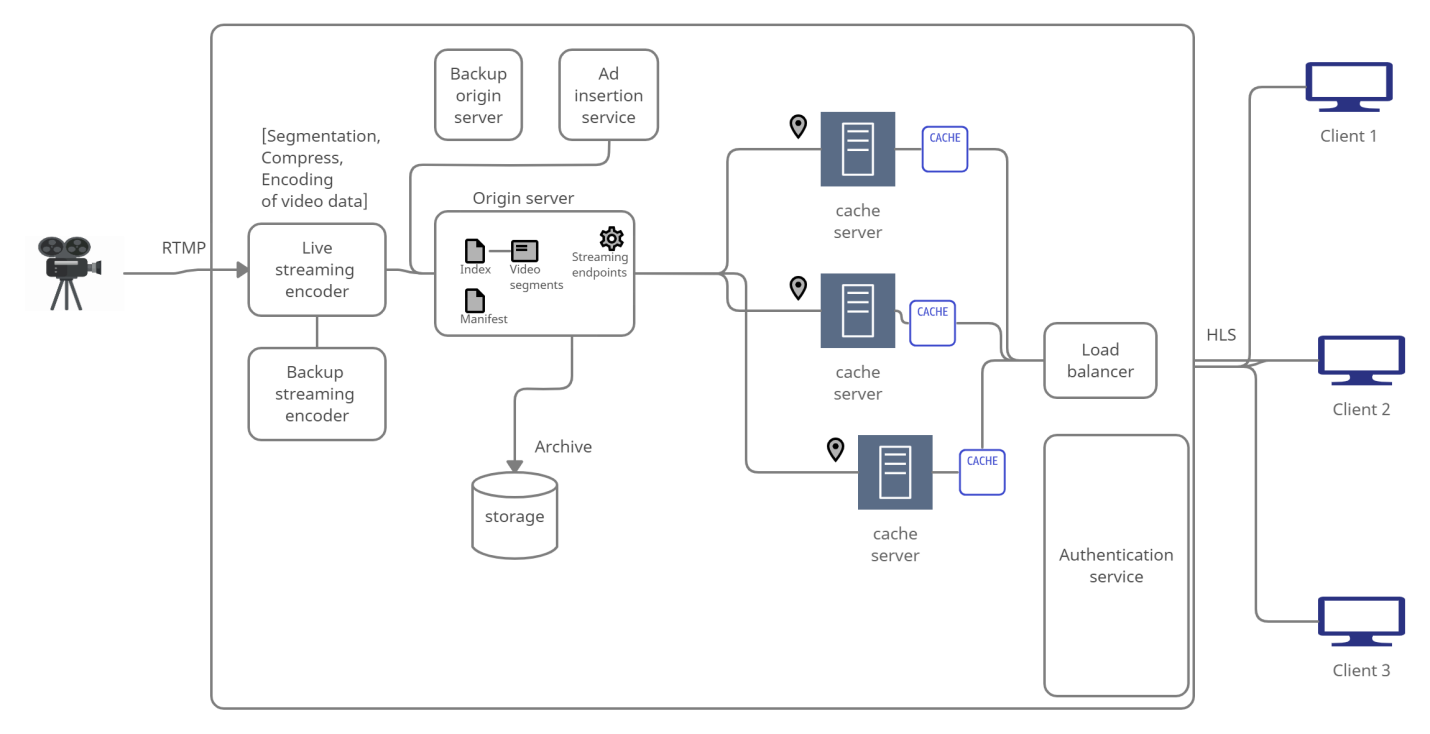
**Vasavi Yarramneni**

LIVE VIDEO BROADCASTING SYSTEM

# Requirements

* Request live video using endpoint streaming URL
* Dynamic ads/Cue points in live video
* Save/archive videos after streaming ends

# Architecture



RTMP – Real Time Messaging protocol

HLS – HTTP Live Streaming

From input sources the raw data is sent to live encoder using RTMP stream.

***Live encoders:***

It converts the raw data into video segments with length say, 10sec each and compresses and encodes the video segments.

Encoder and server should use UTC format, else timestamp issues might occur.

At the end of live streaming, the encoder should send the EOS signal.

We have a Primary and backup stream. If one stream goes down, the player can switch to the backup stream and continue seamlessly. Primary and backup streams ingested from different physical locations.

***Origin server:***

Here we get the video segments and an index file and manifest file which have meta-data about video segments order etc. The server side program pushes the data to a streaming end point URL (ingest URL) given for a streaming pipeline.

We maintain a backup origin server which uses sync-failover strategy.

***CDN caching:***

When more number of requests comes from users, the load balancer routes the requests to appropriate servers. First, it checks with the geographically nearest cache server which can serve the request or not. If that server is busy the request is routed to next appropriate server geographically distance wise and which is not busy.

Using the CDN cache servers, the latency is also decreased due to decrease in RTT (Round Trip Time).

HLS (HTTP Live streaming):

Transmits the video to the clients device where it decodes and decompresses the video segments.

***Dynamic Ad request Failover strategies:***

Say, a 120-second ad break but the ads that were returned from service are only 80 seconds?

* Displays video content that you’ve created (static content) as slate to fill the break.
* Displays live stream content to fill the break.

# Request Workflow

1. Client requests for the live stream using streaming endpoint URL
2. The request goes to the edge servers
   1. If content already present in cache, it is served to user directly.
   2. Else, the edge servers make a request to the origin server for the content.
3. The origin server sends the content or video segment to the client along with index and manifest file using which the order of video segments is determined.
4. The client device will decode the encoded video segment and will play it using video player in client device.