A

Dissertation on (Major Project-II)

Smart Sensitive Content Management MPEG-DASH based HEVC tiling technique

Submitted in Partial Fulfillment of the Requirement
For the Award of Degree of

Master of Technology

In

Software Technology

By

Rahul Aggarwal
University Roll No. 2K16/SWT/513

Under the Esteemed Guidance of

Dr. Manoj Kumar
Department of Computer Science & Engineering

2016-2019

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
DELHI – 110042, INDIA

STUDENT UNDERTAKING

Delhi Technological University
(Government of NCT of Delhi)
Bawana Road, Delhi- 110042

This is to certify that the thesis entitled **"Smart Sensitive Content Management MPEGDASH based HEVC tiling technique"** done by me for the Major project-II for the achievement of **Master of Technology** Degree in **Software Technology** in the **Department of Computer Science & Engineering**, Delhi Technological University, Delhi is an authentic work carried out by me under the guidance of Dr. Manoj Kumar.

Signature:
Rahul Aggarwal
2K16/SWT/513

Above Statement given by Student is Correct.

Project Guide:
Dr. Manoj Kumar
Associate Professor
Department of Computer Science & Engineering, DTU

ACKNOWLEDGEMENT

I would like to express sincere thanks and respect towards my guide **Dr. Manoj Kumar, Department of Computer Science & Engineering, Delhi Technological University Delhi.**

I consider myself very fortunate to get the opportunity for work with him and for the guidance I have received from him, while working on this project. Without his support and timely guidance, the completion of the project would have seemed a far. Special thanks for not only providing me necessary project information but also teaching the proper style and techniques of documentation and presentation.

Rahul Aggarwal

M.Tech (Software Technology)

2K16/SWT/513

ABSTRACT

streaming technologies adaptive in nature are widely used for delivery of video multimedia content over the internet, some video may not be appropriate for all audience, so content provider like YouTube may place many restrictions like age restriction on such videos. Some factors that affect this are:

- Violence and disturbing imagery
- Nudity and sexually suggestive content
- Portrayal of harmful or dangerous activities

In such scenarios content provider have following options:

- Content is completely removed or only visible to certain audience which fulfil the requirement like age.
- Modify the content. For example, (blur the scene, cut few sequences from the video etc.).

Proposed solution will address this issue and provide the way to modify the content at client side in real time in one of most optimized form possible, which will have following advantages:

- Save lot of server space.
- Only MPD file need to be changed on new policy addition or change in policy content will remain unaffected.
- Delivery of Media content will be more flexible as it provides more options to apply different policies.

Smart Sensitive Content Management MPEG-DASH based HEVC tiling technique

A method for tagging a content having sensitive portion is disclosed. The method includes receiving, by a receiving module of a content tagging system, a video content having at least one sensitive portion. The video content is formed of a plurality of frames. The method includes encoding, by an encoding module, each frame of the video content in form of a plurality of tiles. Each tile is indicative of a sub-frame. The method includes categorizing, by a categorizing module, each tile into at least one of a many of defined categories of sensitive content. This method includes tagging, by a tagging module, at least one tile with at least one associated category of the sensitive content.

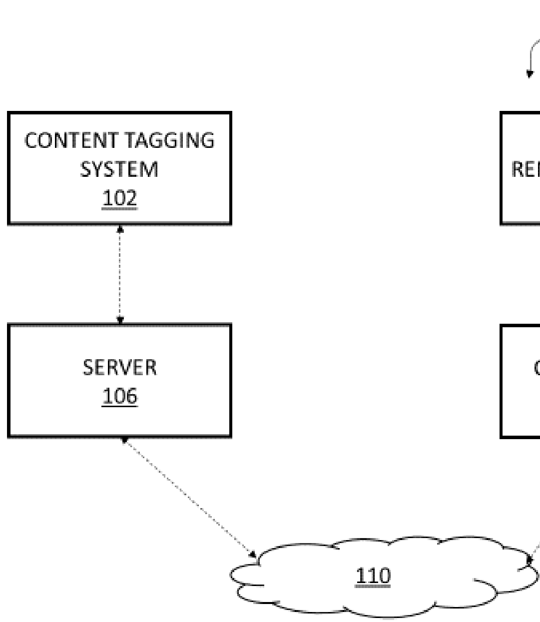


FIGURE 1

TABLE OF CONTENTS

Chapter 1: Introduction

Internet connectivity has allowed viewers across the globe to connect with content providers from different parts of the world to view all sorts of content. This flexibility of course offers a wider exposure to a viewer to understand and experience different cultures around the world. On the other hand, from the perspective of content providers, they also get to cater to a large pool of audience belonging to different geographic locations, which is at least a financial advantage for them.

It is relevant to ensure that such content is being responsibly rendered to the viewers so that it doesn't leave any undesired impression on them. For example, the content may include sensitive portion, such as violence and disturbing imagery, nudity and sexually suggestive content, and portrayal of dangerous activities. Accordingly, there are various factors, such as age restrictions, territorial laws, and content provider policies, which govern the type of content that can be rendered to a viewer. Therefore, it is not possible to render the same content to different viewers across the globe.

Now, in case the restrictions are to be imposed on such sensitive content based on the above-mentioned factors, the content providers either completely remove the content or selectively render the content to a predefined section of viewers meeting the mandatory requirements, such as age. In order to restrict the viewing of banned content, the content providers may censor the content, for example, by blurring a specific scene or by removing few sequences from the content.

As would be gathered, different viewing policies are enforced in different regions and for different age groups. In fact, even within the same territory, multiple viewing policies exist for different communities or age groups. Therefore, it is not possible to generate a common version of the content that can be rendered to different groups of viewers.

Therefore, for different set of viewing policies, the content providers will have to render different versions of the same content, which is a cumbersome and complicated task. Further, in case of any change in any viewing policy, the content providers will also have to remodel their entire rendering policies in order to accommodate the change. For example, the entire video content must be regenerated again. This would of course demand a lot of media server storage for regeneration of each version of content based on a viewing policy. Such regeneration of new versions of the content would also require a lot of manual intervention, even in case of a minor policy change, which poses inconvenience as well. Moreover, owing to significant manual intervention, possibility of error would be higher too.

In order to address the above-mentioned problem, we are proposing a **Smart Sensitive Content Management system based on MPEGDASH HEVC tiling technique, proposed technique included modification on server as well as at the client end.**

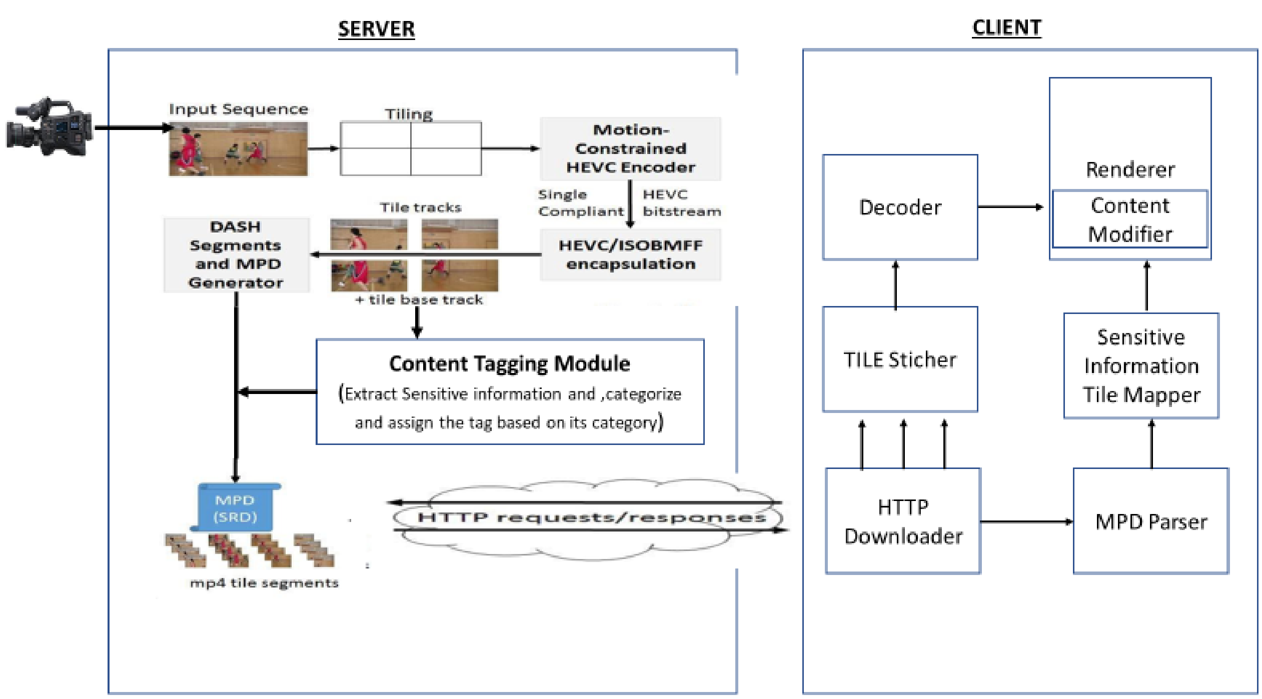


Figure 1.1 Client and server high level implementation diagram

Server uses the HEVC *motion-constrained* **tiling to encode the video and tag each tile with the help of content tagging system** to a particular sensitive category, The method includes receiving a video content having at least one sensitive portion. The video content is formed of a plurality of frames. The method includes encoding, by an encoding module, each frame of the video content in form of a plurality of tiles. Each tile is indicative of a sub-frame. The method includes categorizing, by a categorizing module, each tile into at least one of a many defined categories of sensitive content. This method includes tagging, by a tagging module, at least one tile with at least one associated category of the sensitive content and then content is represented in the form of MPEG which enable live or on-demand HTTP traffic.

In proposed technique tag sensitive information for each tile will be updated in Media presentation Description (MPD) file with a sensitive TAG for each tile in the content and this MPD is available for a client to download before the actual content. A playlist file indicative of a video content having sensitive portion tagged with at least one of a plurality of predefined categories of sensitive content. The plurality of categories is defined based on at least one of age-based restrictions, demography-based restrictions, and language-based restrictions.

At client-side, HTTP downloader will be responsible for any connection with server and for the downloading of MPD and media content in the form of different tiles. Client system includes a HTTP downloader module configured to receive a playlist file indicative of a video content having sensitive portion tagged with at least one of the plurality of predefined categories of sensitive content. The plurality of categories is defined based on at least one of age-based restrictions, demography-based restrictions, and language-based restrictions. The client system includes a decoding module in communication with the Stitching module and configured to decode each frame of the video content.

Each tile is indicative of a sub-frame and at least one tile is tagged with at least one associated category of the sensitive content. The Client rendering system includes a modifying module in communication with the decoding module and the sensitive tile mapper module and configured to modify at least one tile having sensitive content, based on a predefined set of viewing policies for a user. The client rendering system includes a rendering module in communication with the modifying module and configured to render the video content with tiles modified in-line with the predefined set of viewing policies.

We consider using HEVC tiles to solve problems of modification related to sensitive content as per the policy followed by client device which may be enforced by the content provider , government or the user itself. As we are using various multimedia terminology and technology in proposed method, we feel it is important to provide some details about the each one of them.

1.1 HEVC(H265) Tiling

The latest HEVC (High-Efficiency Video Coding Standard) MPEG / ITU-T encoding video standard that achieves a deceleration rate of over 50% using AVC (Advanced Video Coding) in the same quality. HEVC is for High Definition Video (UHD). By using the main profile and today's HEVC codecs, video streams are delivered typically at speeds of 15-20 Mbps. High definition videos (UHD) may not be viewable by all devices (due to the need for processing) or cannot stream on all networks (due to high speed). HEVC offers the concept of tiles to provide parallelism features of coders and decoders. The tiles allow different regions to encode independently, within a single frame, using a one decoder and efficient coding

Consecutive video frames are encoded, the tile in the frame only requires coded information that spans the same space as the reference frames, which effectively creates tile video tunnels . The tiles are stored in a standard format called ISOBMFF (ISO Base Media File Format) files and delivered independently. The client can then offer at least one tile tunnels at the same time based on user interaction. Adaptive HTTP based streaming technologies, such as MPEG DASH, enable live streaming over HTTP or special media streams. The resolution quality of the media that is delivered to the customer can be broadly tailored: user preferences (e.g. language, perspective), client capabilities (e.g., codec support), and network change.

This adaptability is controlled by the client, in which client continuously downloads and play small segments, quality media files and dynamically switches between other quality media files presented by HTTP servers. The player at client side only downloads the quality necessary to the situation, depends on the algorithm of adaptation. With MPEG-DASH, HEVC streaming has been demonstrated and discovered in the industrial technology world. In the academic world, the flow of streaming video for lectures or panoramic images has been explored. HEVC package concepts and responsive HTTP threads are unexplored and open to new threads.

The client can only adaptively stream one or more interesting tiles to reduce requirements of decoding. Although all the tiles are provided, new algorithms customization can be created that allow the consumer to choose different tile properties. Thus, reducing overall bitrate while maintaining the best quality of interesting tiles. Forcing the tiles to become independent with time limits the coding tools used and thus reduces the efficiency of coding. The use cases that consider adaptive streaming of HEVC videos based on tile tracks. In these use cases, a player may decide to play all tiles, possibly with different qualities, or it may also decide to download and play only some tiles (possibly just one). We consider that these use cases are the most relevant. The use case of building a complete bitstream from a single tile track (using extractors) is not considered in this contribution, since it can already be achieved with standard DASH dependencies description.

In the considered use cases, the aggregation of the downloaded media segments may form a complete bitstream (i.e. that there are NALU for all the tiles); or the downloaded media segments may only contain data for some tiles (of interest) and not all, if the DASH client decides to fetch only a subset of the content. It should therefore be pointed out that: - The reconstructed MP4 file (i.e. the one resulting from the aggregation of the tile media segments and the init segment, following the DASH dependency attributes) is conformant to the HEVCFF. - The HEVC elementary stream that can be extracted from the MP4 file may not be complete, and therefore may not respect the HRD model, but is conformant. Additionally, it will be treated correctly by proper HEVC decoders (e.g. decoders that handle packet loss: a 'missing' tile is a special case of a 'loss', simpler to handle that general NALU loss).

There are several options to author the MPD (even for the considered use case). We propose to restrict the options, without loss of features. About initialization segments, the main choice concerns the content of the initialization segment. There are the following options:

- Either all representations use the same initialization segment, and therefore combining some or all representations is simple since the (single) initialization segment can be used;
- or some representations (possibly all) use a different initialization segment. For example, each tile representation may have an initialization segment declaring only the tile track and its base track. Combining two tile representations therefore requires either: mixing initialization segments (which can be tricky) or providing an additional representation with the corresponding initialization segment. This latter case would require as many representations as there are possible tile combinations, which may not be reasonable for large tiling grids and several bitrates.

About SRD when using HEVC tile tracks in DASH, a representation pointing to the base tile track only could be used. Although that associated base tile track is fully conformant to ISOBMFF, it does not lead to a meaningful rendering on its own. We propose to mandate that when such representation is used, it is signaled as non-meaningful.

A Tile Representation is a Representation whose segments are conformant to ISO/IEC 14496-15 and whose media segments contain only media samples for one or more tile tracks, as defined in ISO/IEC 14496-15, i.e. whose type is "hvt1". - A Tile Base Representation is a Representation whose segments are conformant to ISO/IEC 14496-15 and whose media segments contain only media samples for the base tile track, i.e. whose type is "hev2" or "hvc2".

NOTE 1: A Tile Representation is dependent on a Tile Base Representation because ISOBMFF tile tracks depend on ISOBMFF base tile track for processing.

NOTE 2: An MPD MAY offer all tile tracks in one single Tile Representation, or each tile track in its own single Tile Representation, or a combination of tile tracks in a single Tile Representation.

NOTE 3: The above definitions apply to HEVC tile tracks (i.e. of type "hvt1") and to the associated tile base track. ISO/IEC 14496-15 allows the carriage of tile content in other types of tracks. MPD authors may decide to provide Representations using these other track types. This Annex does not constraint these Representations.

Constraints on initialization segments in an MPD, all Tile Representations depending on the same Tile Base Representation SHALL have the same initialization segment (i.e. declaring base tile track and all tiled tracks).

NOTE1: The media segments of those Representations will typically contain only media samples from a set of tile tracks.

For a given MPD time, the concatenation of a Tile Base Representation initialization segment, followed by the Tile Base Representation media segment for that time and the media segments for that time of one or more Tile Representations but identical initialization segment shall be a conforming HEVC file as defined in ISO/IEC 14496-15.

NOTE2: Given that all Representations use the same initialization segment and that this segment declares the tile tracks and the base track, the above rule makes it possible for a DASH client to select several Tile Representations and to combine them.

NOTE3: It is possible to offer different Tile Base Representations and different Tile Representations associated to these Tile Base Representations, for example to provide multiple resolutions. In this case, reconstructing a single complete HEVC bit stream is made by selecting all Tile Representations sharing the same initialization segment.

Constraints on Spatial Relationship Descriptions shall be used in Tile Representations or Tile Base Representations to indicate a DASH client that how the different Representations (and containing Adaptation Sets) are combined in the source content. SRD descriptors shall be used, and the Tile Base Representation and its dependent Tile Representations shall have the same source id parameter.

The Tile Base Representation shall have its width and height set to the values of the tile base track, i.e. the width and height of the full media. SRD descriptors shall be used on Tile Base Representations as follows: - an EssentialProperty SRD descriptor shall be used,

- this SRD's object_x, object_y, object_width and object_height shall be set to 0, NOTE: The use of 0 for object_width and object_height is meant to signal to the DASH client that this Representation alone does not lead to a meaningful presentation.

- this SRD shall contain the optional total_width and total_height parameters, specified as the values of the full video width and height, possibly in arbitrary units.

SRD descriptors shall be used on Tile Representations as follows: - a SuplementalProperty SRD descriptor should be used, - this SRD's object_x, object_y, object_width and object_height shall be set to the horizontal position, vertical position, width and height values of the tile in the reconstructed picture, in the same units as the Tile Base Representation SRD descriptor.

1.2 Protocol Stack of an HTTP adaptive system

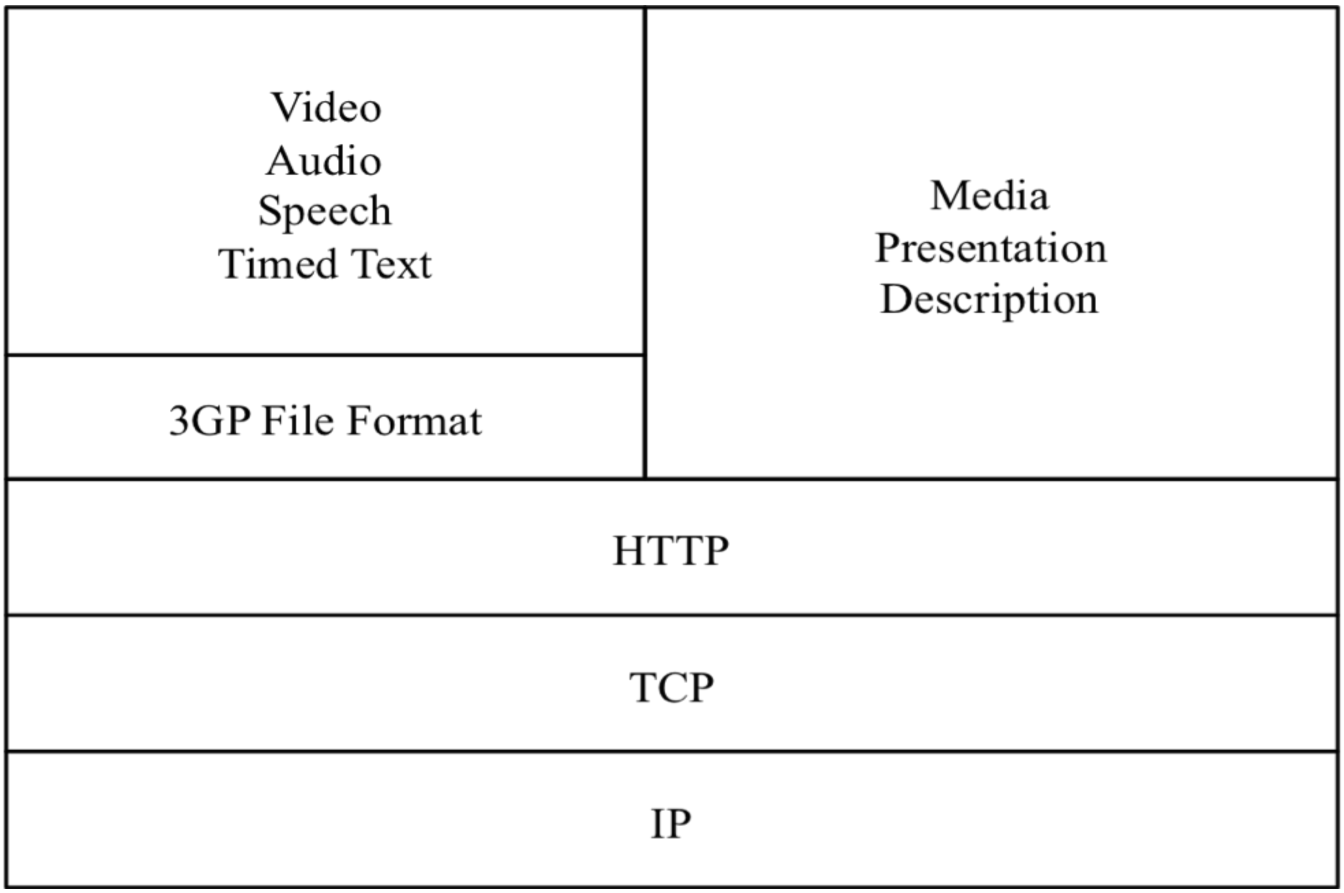


Figure 1.2 HTTP Protocol stack for adaptive system

1.3 File Formats:

- A video application always needs more than just the video bitstream.
  - Metadata, including timing information etc., to ease content exchange, editing, streaming, playback operations like seeking.
- Lots of today's video applications, e.g., all video streaming systems, are based on a file format.
- One of the most widely used standard file format is the ISOBMFF (ISO base media file format)
  - ISO/IEC 14496-12
- Each media codec typically has a codec-specific file format based on ISOBMFF, for carriage of media coded using that codec in ISOBMFF
- ISO/IEC 14496-15 includes
  - AVC file format
  - SVC file format
  - MVC file format
  - HEVC file format
  - Layered HEVC file format
  - File format for HEVC and layered HEVC tiled video

1.4 ISOBMFF typical box hierarchy

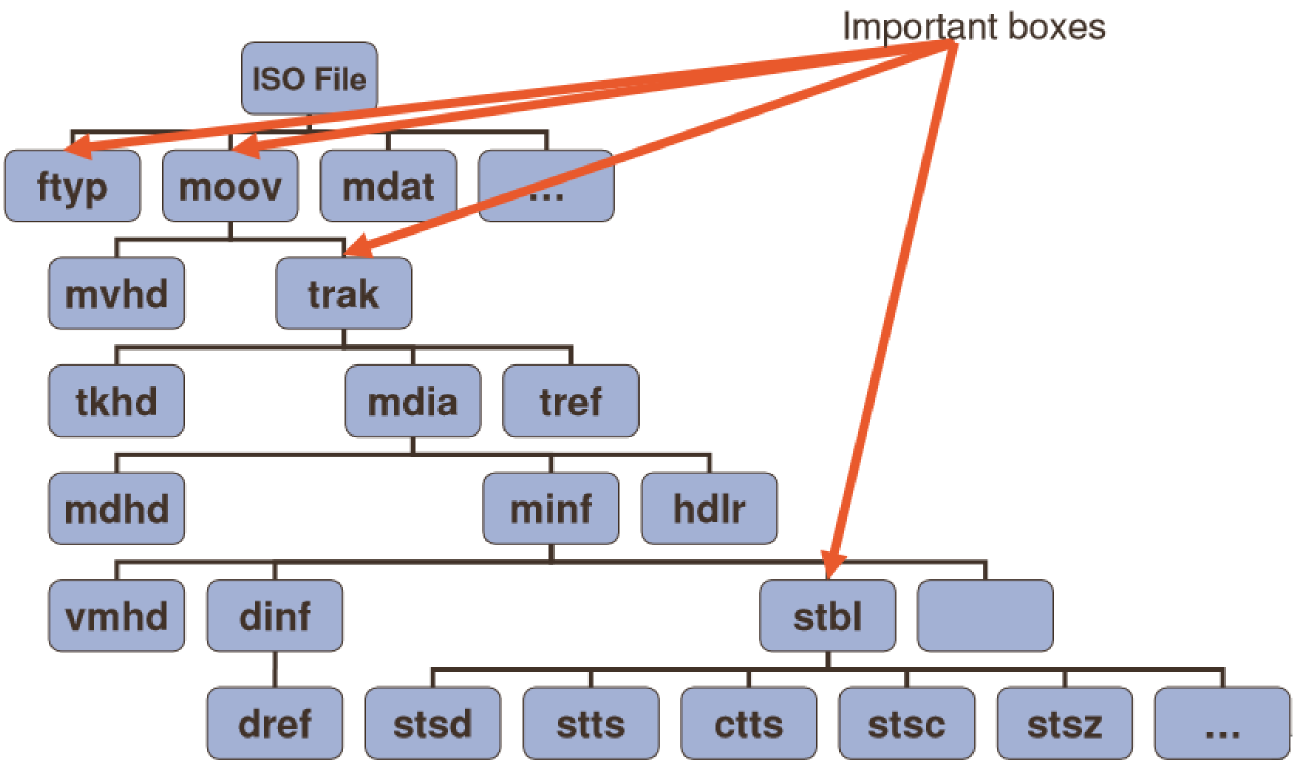


Figure 1.3 ISOBMFF box hierarchy diagram

Details of some ISOBMFF boxes

- `ftyp` (File Type): 1 per file
  - File type
  - File version
  - Compatibility with other ISO files
- `mdat` (Media Data):
  - Contains the media data
  - A file may have several, non contiguous
- `moov` (Movie):
  - Unique container for the metadata of a presentation
- `mvhd` (Movie Header):
  - Generic info about the movie
- `trak` (Track):
  - Container for Meta-date related to one stream
- `hdlr` (Handler)
  - Indicates the type of stream
- `dinf/dref` (Data Information/Data Reference)
  - Indicates the location of the data (current file or remote file)
- `stbl` (Sample Table)
  - Contains the meta data related to samples, sample per sample
- `stsd` (Sample Description)
  - Contains the decoder configuration for the elementary stream
- `stts` (Sample To Time)
  - DTS for each sample
  - Use a predictive coding scheme
- `stsz` (Sample To Size)
  - Size of each sample, run-length coded

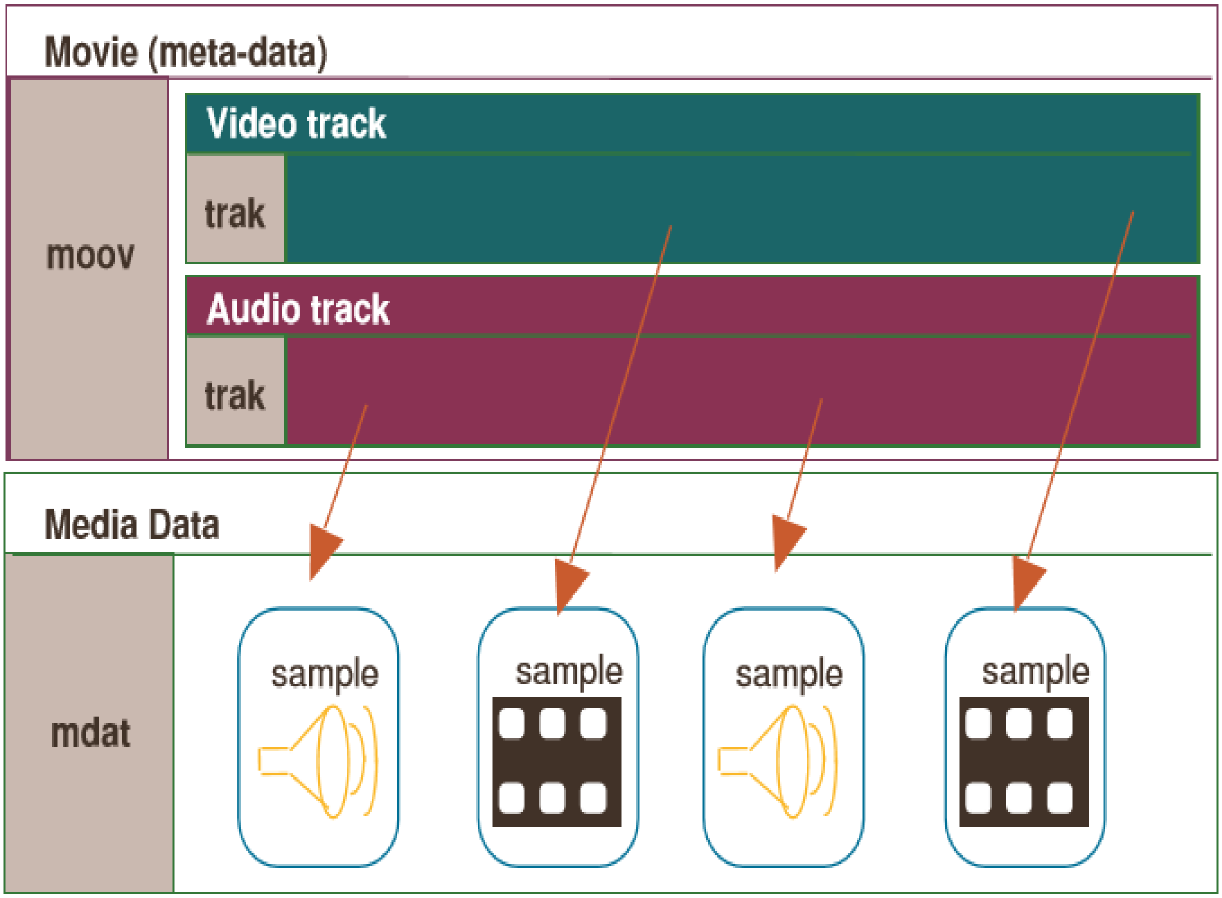


Figure 1.4 ISOBMFF example file

1.5 DASH Basics

1.5.1 DASH Streaming procedure

a) The client obtains the MPD of a streaming content, e.g., a movie.

   i. The MPD includes information on different alternative representations, e.g., bit rate, video resolution, frame rate, audio language, of the streaming content, as well as the URLs of the HTTP resources (the initialization segment and the media segments).

b) The client requests the desired representation(s), one segment (or a part thereof) at a time.

c) Based on information in the MPD and the client's local information, e.g., network bandwidth, decoding/display capabilities, and/or user preference, The client requests segments of a different representation with a better-matching bitrate.

   I. When the client detects a network bandwidth change.

   II. Ideally starting from a segment that starts with a random-access point.

1.5.1 Why adaptive streaming over HTTP?

- **Basic Approach: Adapts the video to the Internet without changing the Internet.**
- **Streaming realized by continuous Small Downloads.**
- Downloads in small portions to reduce bandwidth loss.
- Let's you monitor consumer and customer monitoring.
- Adapts to the dynamic environment and capabilities of the device.
- Adapts to dynamic situations everywhere through the Internet or home network.
- Adaptation to display resolution, processor and client memory resources.
- It simplifies the paradigm, "on any device, anytime, anywhere".
- **The quality of the Experience is improved**
- Enables quick start and search (as opposed to progressive download)
- Eliminates and Reduces freezes, rebuffering, skips and stutters
- **Based on HTTP**

- The name / addressing method is well understood
- Provides easy switching to all types of middle boxes (e.g. NAT, firewall)
- Enables cloud access, uses an existing HTTP caching infrastructure.
- Enables client-driven deployments
- Enables reuse of existing web technologies: authentication, authorization, etc.

1.5.2 DASH Basics

Providing the client with information on where and when the A / V experience can be found

1) **MPD**

- Provides cloud and HTTP-CDN delivery → HTTP-URLs and MIME Types.
- Allows your service provider to combine / splice different types of content into one media presentation.

2) **Periods**

- Provide service provider that allows the client / user to select media content components based on the user's preferences, profiles and capabilities of the interaction device using terms and other metadata.

3) **Adaptation Sets**

- Provide the same content with different encodings (bitrate, resolution, codecs)→ **Representations.**
- Provide scalable syntax and semantics that describe the properties of the presentation and presentation of the customization → **Descriptors.**
- Provide ability to access to the small content and plan your access properly → **Segments and Subsegments.**
- Ensure efficient signaling and address optimization → **Playlist, Templates, Segment Index.**

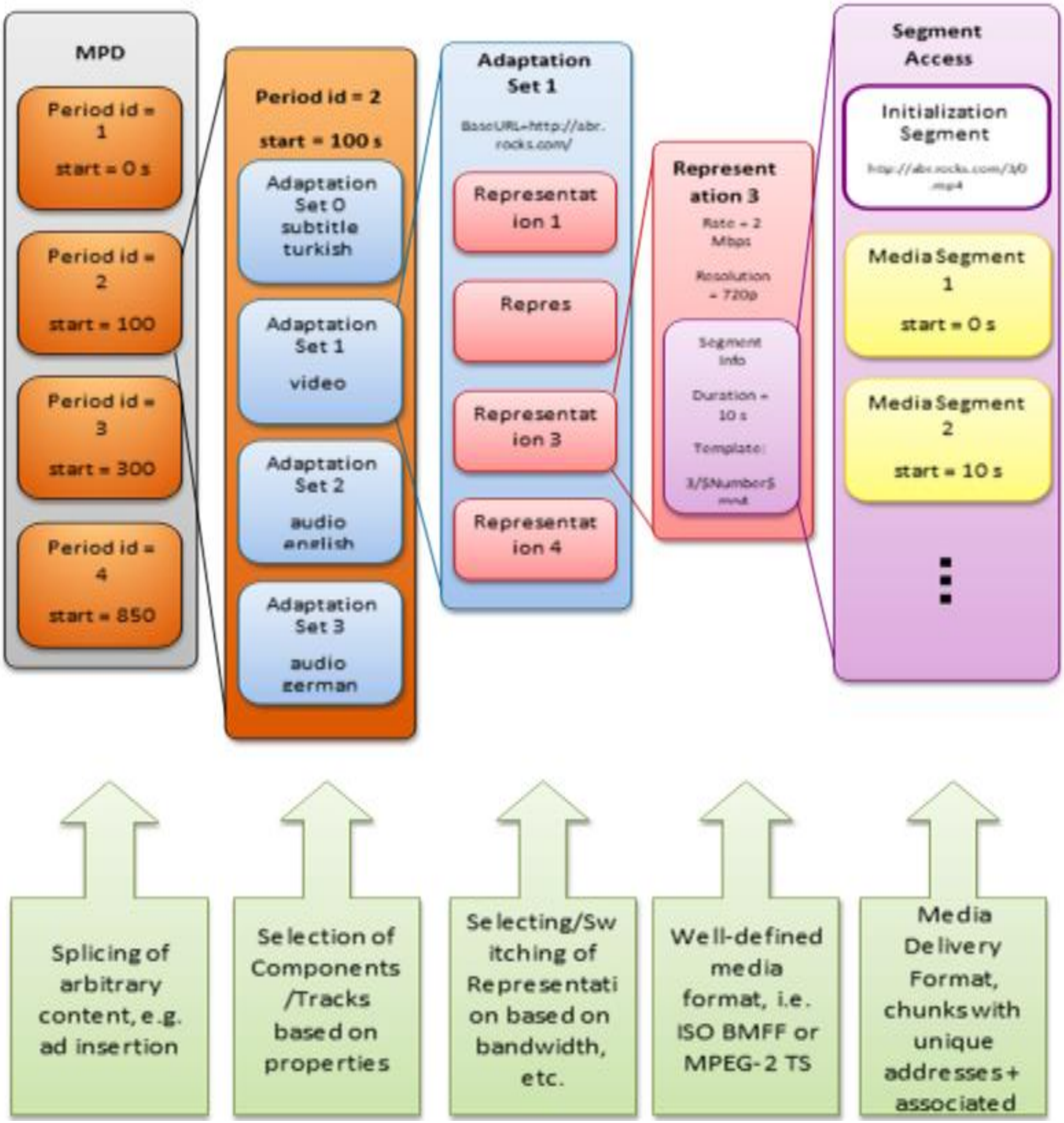


Figure 1.4 MPEG-DASH MPD structure

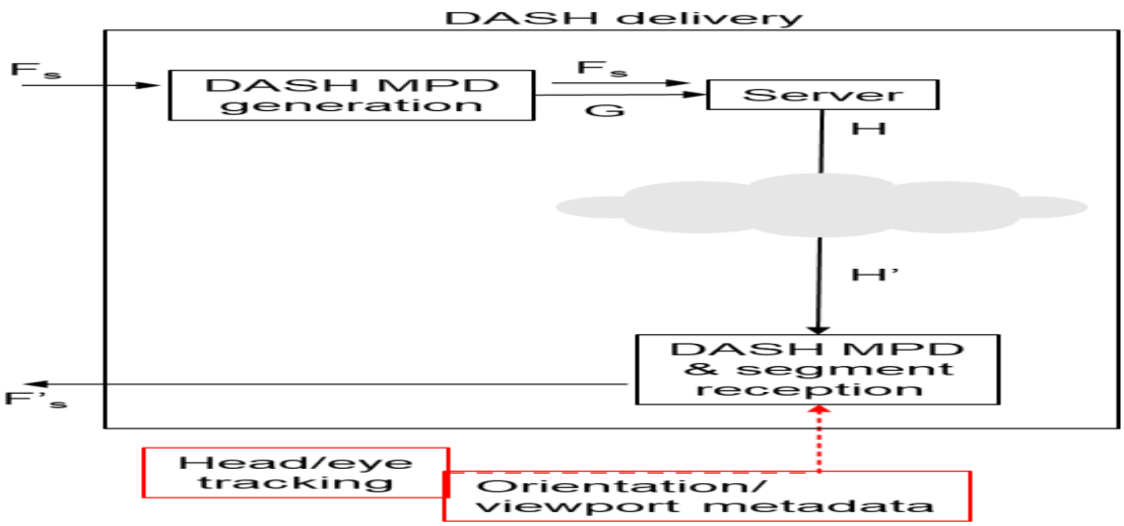


Figure 1.5 DASH Delivery system

- F_s/F'_s: Initialization and media segments
- G: MPD
- It additionally includes OMAF-specific metadata, such as information on projection and region-wise packing
- Basic OMAF DASH streaming procedure

  1. The client gets the MPD.
  2. The client obtains the current viewing orientation and gets the estimated bandwidth.
  3. The client chooses the Adaptation Set(s), the Representation(s) and requests the (Sub) Segments to match the client's capabilities, incl. OMAF-specific capabilities, and to maximize the quality, under the network bandwidth constraints, for the current viewing orientation.
  4. Repeat steps 2 and 3.

Chapter 2: Related Work

Several research reports have investigated the problem of transmitting high or high definition video in bandwidth restricted networks for various applications, such as panoramic video, streaming lectures, or mobile devices. In some references, authors explore an interactive stream of areas of interest for high definition videos in two ways: Streamed less resolution tile independent videos combined on the client-side to accommodate the ROI (Region of interest), or the Region of interest (ROI) transmission and coding dependencies are tracked in a one monolithic video. The methods proposed yield good results for reducing bandwidth, but on expense of client modifications for flow reconstruction.

We try to achieve the same purpose with method of tiling, but with no change in decoder and other standard multimedia components. In the following work (9), the authors consider using adaptive coding to differentiate ROI quality from different tiles. We will also be using an adaptive streaming approach, but the main difference between reference approach and the approach followed in proposed method is that the client makes an adaptive decision. In adaptive solution as it is used in today's adaptive flow techniques using a standard scalable HTTP server, MPEG-DASH.

Finally, in other following paper (10), the authors study the stitching of tiles with dissimilar qualities in the context of DASH adaptive streaming. In addition to other works [6], the creators propose a video-encoded approach so that it can be obtained from an arbitrary ROI compressed domain that meets different zoom factors.

This system uses H.264(AVC) standard P slices, related search and continuing memory motion compensated estimate. It has exciting features that can be scaled up in terms of coding, i.e. the client does not need special coding, but a specific decoder. This is due to the fact that It is centered on slices which in HEVC are less suitable than tile in HEVC. The proposed solution addresses similar situations to unmodified standard decoders of HEVC using HEVC tiles. Likewise, the work given in reference (11) is described in H.264 | AVC and amenable macro block positioning (FMO) are used to encode the secluded areas in the video.

FMO macro block positioning is like a tile with the troubles that it is not generally endorsed in deployed profiles and the ISOBMFF storage space is not displayed efficiently. Like shown in detail [2], in HEVC the image can be partitioned into rows and columns, and the area in the middle of two rows in a row and two columns in a columns is the tile. As mentioned above, tiles in HEVC have been launched primarily to improve the parallelism of both video codecs and decoders.[12] Discuss the various parallelism methods offered in HEVC, including tile content, and offer their efficiency in coding. This document indicates that parallelism has turn out to be a necessity for high definition video and that tiles are superior for ROI coding scenarios. This means that for 4k videos, tile-based achieve parallelism which can accelerate decoding.

However, these benefits are the cost of encoding due to a decrease in the probability of entropy encoding, a limitation of the intra-picture prediction, and a violation of the decoding filters on the tile boundaries. [12] Value coding shows an increase in the quantity of tiles and is reduced from 7% to 1% at 1080p resolution, but drops at a resolution and to an average of 2% at Ultra High Definition(UHD) resolution. It also suggests to use square tiles, as such tiles reduce their dependency on non-square tiles. This act does not include traffic restrictions. Several studies, such as [13], [14], have calculated the effect of some kind of MC(motion-constrained) tiling on HEVC for products or cloud remixing OR video conferencing, where numerous videos, such as each video input, are embedded in one video. These researches show that up to 3% of unlimited coverage is used for low resolution and simple 2x2 mesh, but only with a limited range of test chains and configurations.

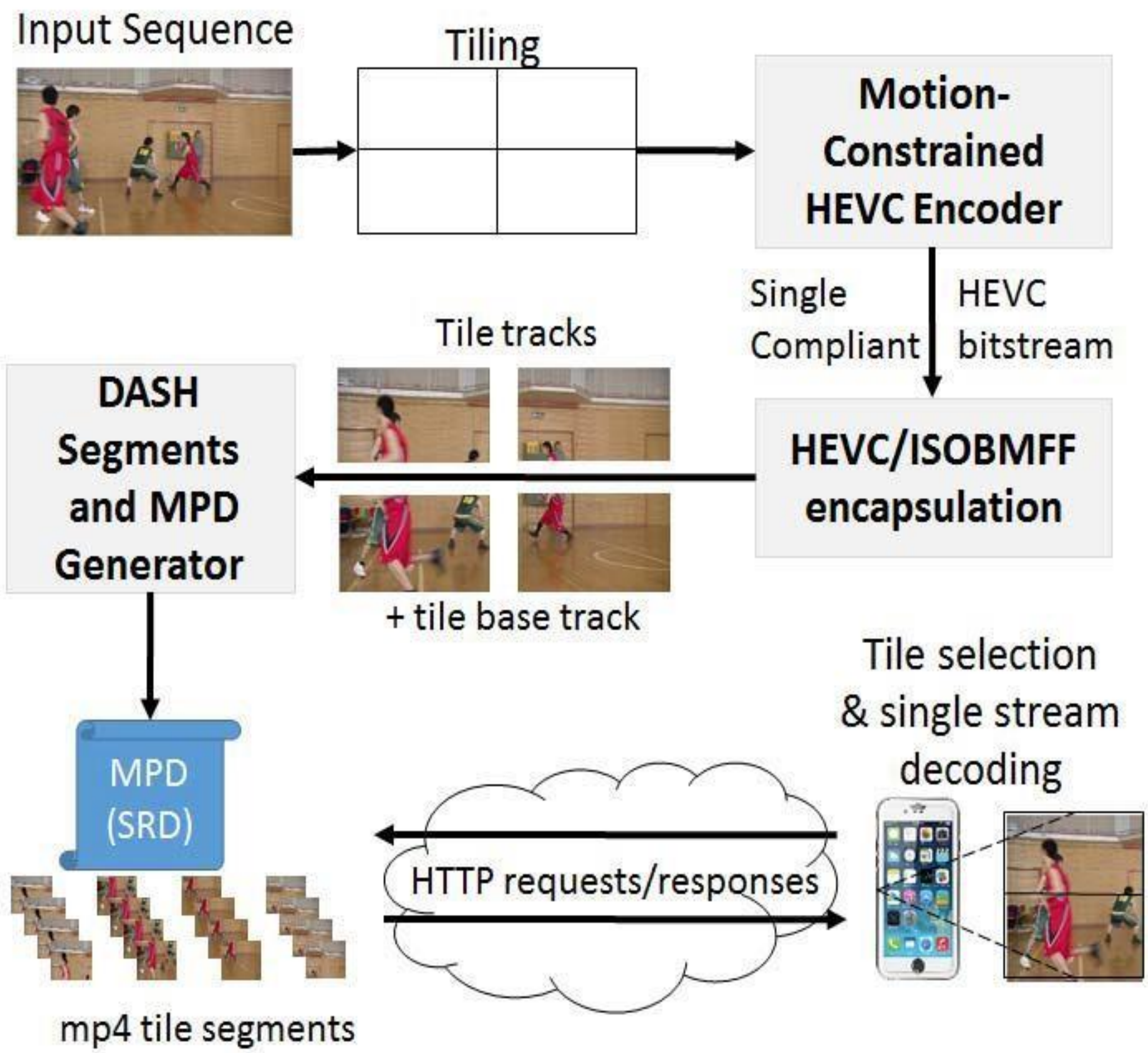


Figure 2.1 TILE based server and client system

Lastly, from a perspective of system [15], it considers various opportunities for HEVC content delivery, involving broadcast, file playback and streaming. It offers top-level HEVC content syntax, which includes the signaling elements to the tiles; and HEVCs to ISOBMFFs, compared to H.264 AVC, emphasizes that parameter sets can be saved in-band. with image data. This method is adaptable and allows easy assimilation into the DASH ecosystem. It facilitates adaptive streaming of encoded HEVC content which saved in the ISOBMFF, as shown by the GPAC Open Source Project [16] or [17]. [15] However, in addition to DASH as discussed in this document, the problem of delivering high-definition HEVC video using tiles cannot be resolved. Finally, using tapes, HEVC discusses panoramic streaming video, but not the best or DASH aspects of coding.

As per our research till now we did not find any papers which talk about handling the content sensitivity using HEVC tiling technique, Using tiling technique, we are not only making an efficient system which is fast but also flexible in terms of modifying the content and at the same time it saves lots of storage and video editing effort at server end.

Chapter 3: Problem Statement

3.1 Description of Problem

Some video may not be appropriate for all audiences, so content provider like YouTube may place many restrictions on such videos like age restriction on the video. Some factors that affect this are:

- Violence and disturbing imagery
- Nudity and sexually suggestive content
- Portrayal of harmful or dangerous activities

In such scenarios content provider have following option

a. content is completely removed or only visible to certain audience which fulfil the requirement like age.
b. Modify the content (Blur the scene, cut few sequences from the video)

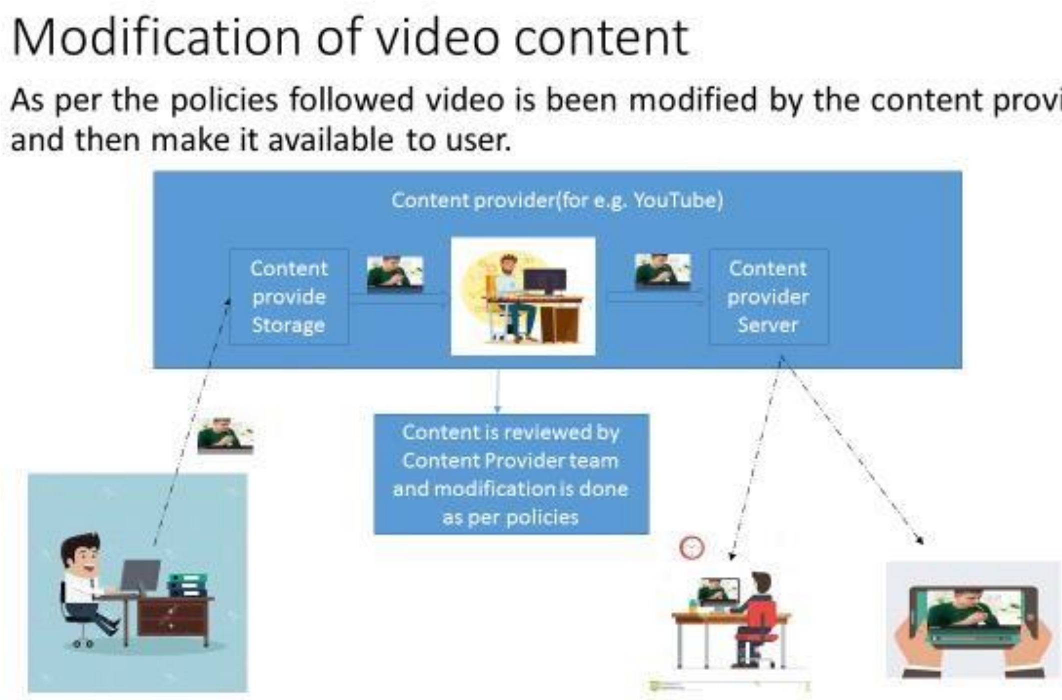


Figure 3.1 Present system for video content modification

Following are the problem with modifying the content as per fixed policies, as policies are defined and influenced by many factors such as: -

- Content provider policies
- Age restriction policies
- Policies forced as per country law
- All the above categories mentioned are subject to change any time depending upon different internal and external factors.
- Different policies exist for different region and different age groups, so one content is not enough to satisfy all the policies around world.

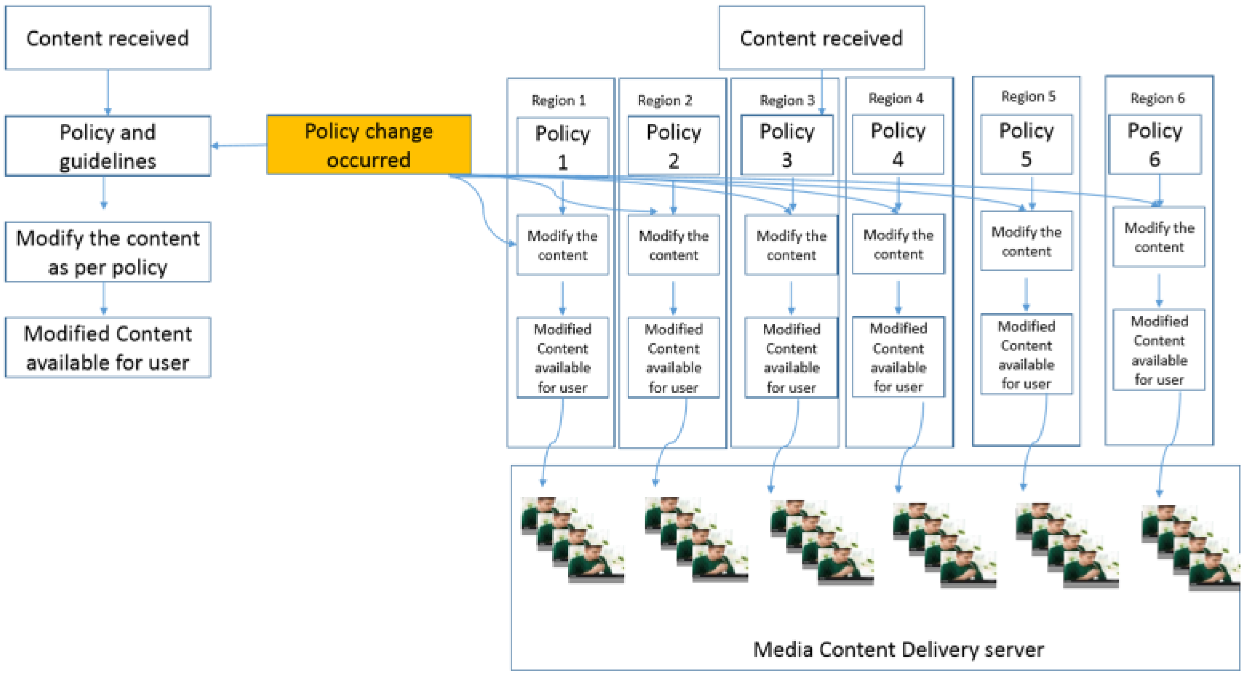
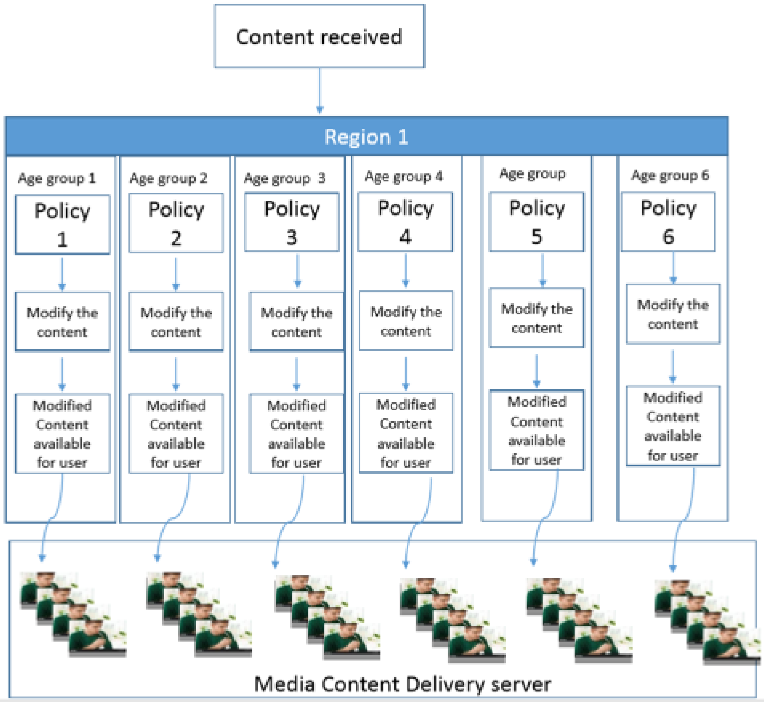


Figure 3.2 Present system of video content modification based on different policies

On Each policy change content need to be modified, also policies are influenced by region or country, One region policy can be different than other region policies, so content provider needs to keep content as per each policy



<u>Figure 3.3 present system of video content modification based on different policies</u>

<u>Even one region can have multiple policies applying to different age group or community, Content provider needs to modify the content for each policy and also in case of any change all these content will need to edited again</u>

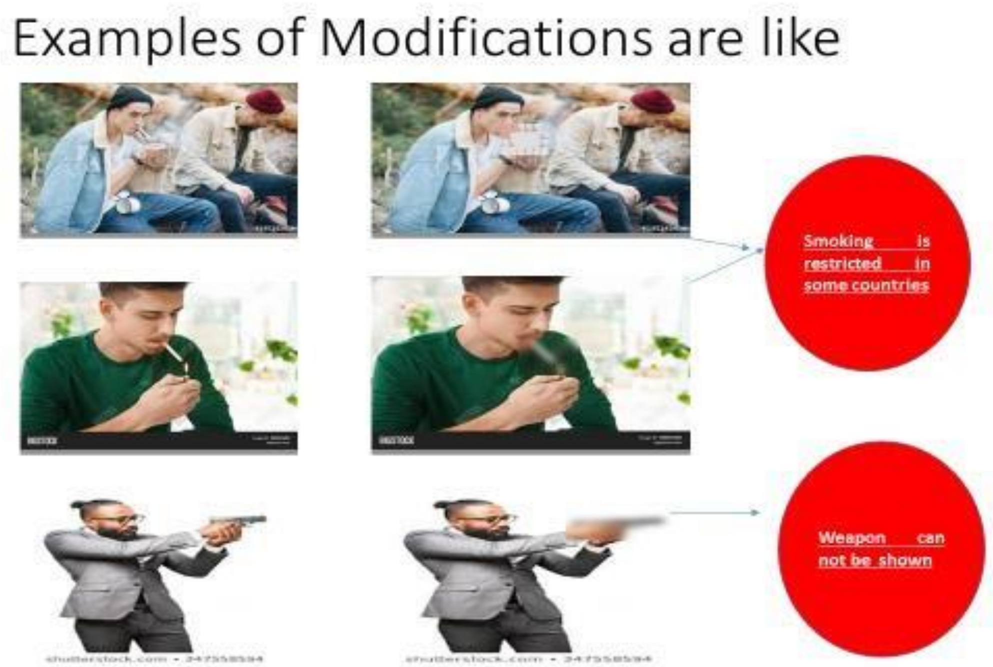


Figure 3.2 image of modification examples

Highlights of problems:

- **A whole video content needs to be regenerate again**
- **Lots of Media server storage is required for each content based on policy.**
- **A whole new content will be created each time new policy is in place or there is a change in existing policy.**

Chapter 4: Proposed System Design

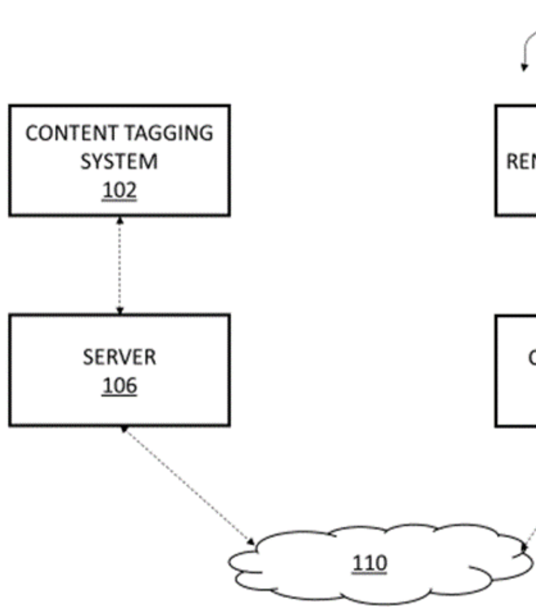


Figure 4.1 Content tagging high level architecture

Figure 4.1 demonstrates an ecosystem where, environment 100 is employing a content tagging arrangement 102 and a content rendering system 104, according to an embodiment of the proposed solution. The environment 100 may include a server 106 of a content provider in communication with a client device 108. In an embodiment, the server 106 and the client device 108 in communication with each other over a network 110. The network 110 may be a wired network or a wireless network. The server 106 is shown to be in communication with only one client device 108 in the proposed solution. The server 106 may be in communication with a plurality of client devices 108.

Further, the content tagging system 102 may be installed at the server 106 whereas the content rendering system 104 may be installed at the client device 108. In an embodiment, the content tagging system 102 may be configured to tag a content having at least one sensitive portion. Constructional and operational details of the content tagging system 102 are explained in the description. Figure 2. Further, the content rendering system 104 may be configured to render a content having tagged sensitive portion.

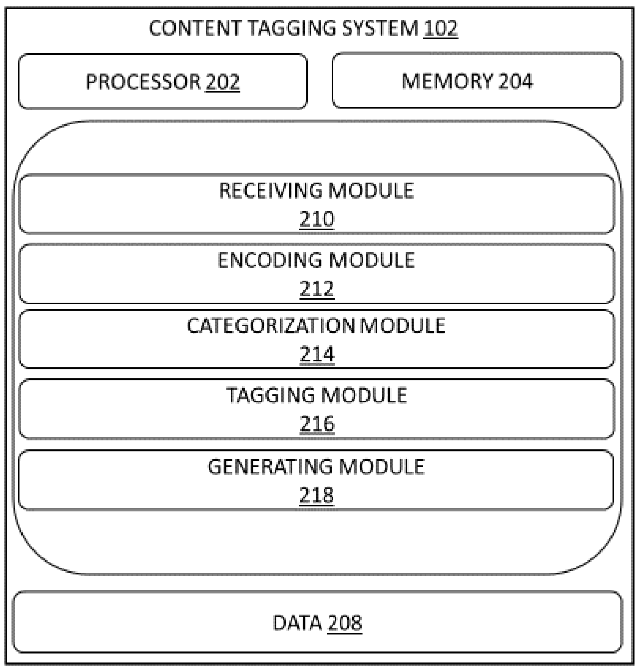


Figure 4.2 Content tagging high level Device system

Figure 4.2 illustrates a block diagram of the content tagging system 102, according to a proposed solution. The content tagging system 102 may include a processor 202, a memory 204, modules 206, and data 208. The modules 206 and the memory 204 are coupled to the processor 202. The processor 202 can be a single processing unit or several units, all of which could include multiple computing units.

The processor 202 may be implemented as one or more microprocessors, microcomputers, microcontrollers, digital signal processors, central processing units, state machines, logic circuitries, and/or any devices that manipulate signals based on operational instructions. Among other capabilities, the processor 202 is configured to fetch and execute computer-readable instructions and data stored in the memory 204.

Memory 204 can be any read-only environment known to the art, like SRAM(static random access memory) and DRAM(dynamic random access memory), and / or non-volatile memory only as ROM(read-only memory), programmable ROMs, flash drives, hard disks, optical drives and magnetic tapes. Plans, programs, objects, components, data structures, etc. that can perform specific tasks or perform specific types of data, such as modules 206. includes. Modules 206 can be implemented, for example, as a signal processor (s), state apparatus (s), logic circuits and / or any other device or module that manages signals based on operating guidelines.

In addition, components 206 can be realized in hardware, in the directions provided by the processor device, or in combination. The processor block may include a computer, a processor such as a 202 processor, a state machine, a logical array, or any other suitable device that can process instructions. A processor block can be a target processor block, which performs instructions that force a general-purpose processor to accomplish the necessary tasks. Another version of this release may be a machine-readable instruction (software) that performs any of the functions described during module / processor execution at module 206.

In an implementation, the modules 206 may include a receiving module 210, an encoding module 212, a categorizing module 214, a tagging module 216, and a generating module 218. The receiving module 210, the encoding module 212, the categorizing module 214, the tagging module 216, and the generating module 218 may be in communication with each other. Further, the information 208 serves, amongst other things, as a repository for storing data processed, received, and created one or several of the components 206.

In an embodiment, the receiving module 210 may be configured to receive a video content having at least one sensitive portion. In an embodiment, the video content may be formed of a plurality of frames. Further, the encoding module 212 may be configured to encode each frame of the video content in form of a plurality of tiles. Each tile may be indicative of a sub-frame. The coding module 212 can encode each frame in the form of multiple tiles using standard high-performance video encoding (HEVC) technique. HEVC encoding is a Moving Picture Experts Group (MPEG)/ITU-T capable of accomplishing more than 50% bit rate cut over the AVC (Advanced Video Coding) standard of the same quality. The standard technique is specifically intended for Ultra-High Definition UHD videos.

At this resolution, if main profiles are used and with today's HEVC codecs, video streams can usually be up to 15-20 Mbps. UHD or higher-resolution videos might not be viewable on all devices, for example due to the need to process them, or stream-able through on all networks, for example at high bit rates. The standard HEVC tiling method can be used in the 212 coding module to provide parallelism of encoders and decoders. Tile frame encoding allows you to encode different areas within a single frame using a single decoder and efficient encoding.

In one embodiment, the 212-encoding module can implement motion-constrained tiling of frames of the content. The 212 encoding Module can encode sequential images into video sequences, so the tile in the image is only needed if the image contains such coded information in the same space, which effectively creates video tile tunnels. The tile video tunnels can then be deposited in ISOBMFF(ISO Base Media File Format) files and are supplied separately.

The player can then choose to present tunnels one or more tile, based on user interaction. HTTP Adaptive streaming based technologies, such as MPEG DASH, allow live streaming over HTTP or on-demand providing media streams. In one embodiment, the quality of media provided to a customer may be broadly tailored to a user's favorites, such as language and perspective), codec support, and network change.

Alternatively, in one embodiment, the 214-categorization module can be configured to classify each tile, at least in one of the majorities of predefined categories of sensitive content. In one version, several categories can be defined for at least one age- based restrictions demographic and language restrictions. Tagging module 216 can be configured to mark at least one tile with at least one associated sensitive content category.

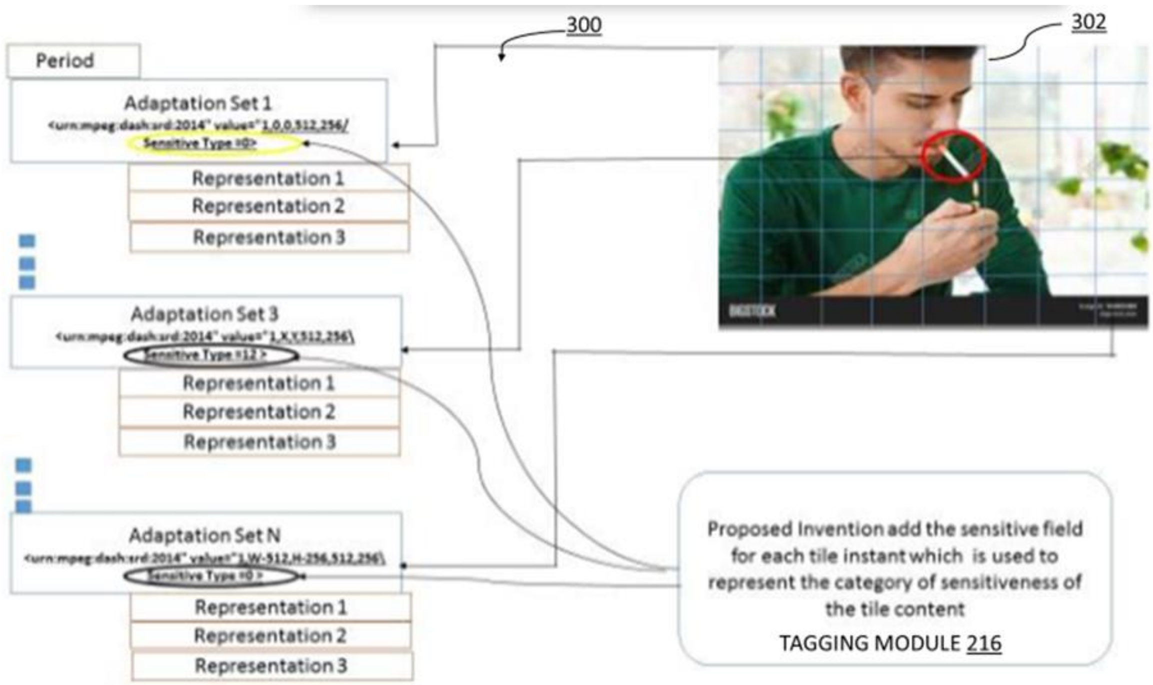


Figure 4.3 Content tagging Implementation in MPD

Figure 4.3 illustrates a block diagram 300 depicting tagging of the sensitive portion of a frame of the content by the tagging module 216, according to an embodiment of the current revelation. As would be valued by a person experienced in the art, Figure 3 is included for providing better understanding of the present disclosure and therefore, should not be construed as limiting in any way.

Referring to Figure 2 and Figure 3, the frame 302 is encoded in form of a plurality of tiles. As illustrated, the frame 302 is depicting a male smoking a cigarette. Further, the tagging module 216 may tag each tile of the frame 302 with at least one associated category of the sensitive content. For example, the tagging module 216 may tag the tile having the cigarette with a category of the sensitive content, namely, “smoking”. Therefore, the frame 302 is now tagged with the category “Smoking” indicative of the frame 302 having a smoking sequence.

Referring to Figure 2, once the content is tagged by the tagging module 216, the generating module 218 be designed to generate a playlist file indicative of the video content having the sensitive portion tagged with the associated predefined categories.

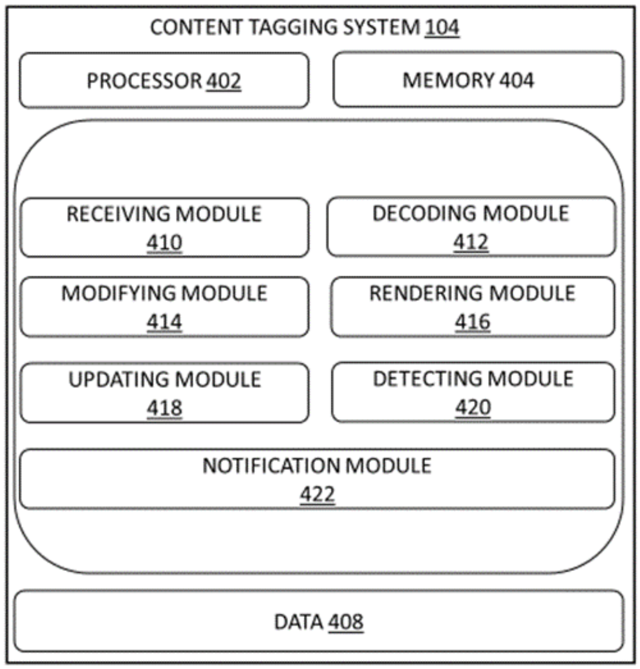


Figure 4.4 Content tagging high level Device system

Figure 4.4 illustrates a block diagram of the content rendering system 104, according to an embodiment of the present method. The content rendering system 104 include a processor 402, a memory 404, modules 406, and data 408. The modules 406 and the memory 404 are coupled to the processor 402. The constructional and operational characteristics of the processor 402, the memory 404, the modules 406, and the data 408 may be like the processor 202, the memory 204, the modules 206, and the data 208 of the content tagging system 102. Therefore, for the sake of brevity, those details are not explained again in the description of Figure 4.

In an embodiment, the modules 406 may include a receiving module 410, a decoding module 412, a modifying component 414, a rendering component 416, an updating component 418, a detecting module 420, and a notification module 422. In an embodiment, the receiving module 410, the decoding module 412, the modifying module 414, the rendering module 416, the updating module 418, the detecting module 420, and the notification module 422 may be in communication with each other.

In an embodiment, the receiving module 410 may be configured to receive a playlist file indicative of a video content having sensitive portion tagged with at least one of a plurality of predefined categories of sensitive content. In an embodiment, the receiving module 410 may receive the playlist file from the content tagging system 102.Upon receiving the playlist file, the decoding module 412 may be configured to decode each frame of the video content in form of the plurality of tiles. Further, the modifying module 414 may be configured to modify the at least one tile having sensitive content, based on a predefined set of viewing policies for a user. In an embodiment, the rendering module 416 may be configured to render the video content with tiles modified in-line with the predefined set of viewing policies.

In an embodiment, the updating module 418 may be configured to update the predefined set of viewing policies for the user. Based on the updated set of viewing policies, the modifying module 414 may be configured to re-modify the at least one tile having the sensitive content. In an embodiment, the detecting module 420 may be configured to detect, while the video content is being rendered to the user, presence of a sensitive portion in one of subsequent frames of the video content. Further, the notification module 422 may be configured to notify the user of the presence of the sensitive portion in the subsequent frame, before the sensitive portion is rendered to the user.

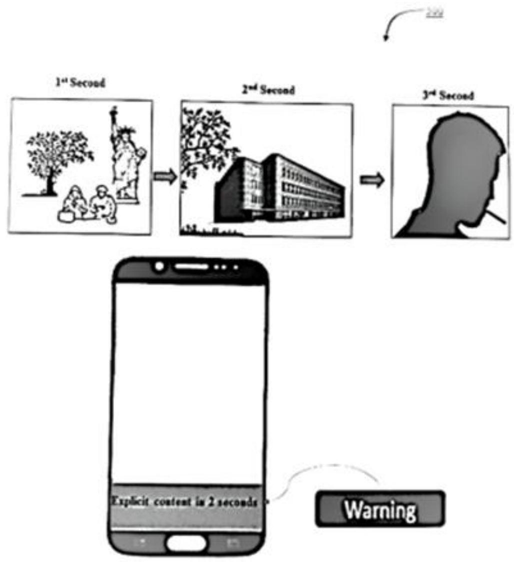


Figure 4.5 example of notifying a user of the presence of sensitive portion

Figure 4.5 illustrates a block diagram 500 depicting an example of notifying a user of the presence of sensitive portion in a subsequent frame of the content, according to an embodiment of the present disclosure. As illustrated, the block diagram 500 includes 3 frames of the content, for example, at the 1^{st} , 2^{nd} and the 3^{rd} second of the content. Further, the third frame of the content includes sensitive portion. Therefore, when the user is viewing the content, the detecting module 420 may detect the presence of the sensitive portion in the 3rd frame of the content. Accordingly, the notification module 422 may notify the user of the presence of the sensitive portion in the upcoming frames, for example, by prompting a message "Sensitive content in next 2 seconds". Referring to Figure 4, in an embodiment, the content rendering system 104 may receive a user instruction, while playing the content. The user instruction may be indicative of a tile of a frame having a sensitive content. In such an embodiment, the content rendering system 104 may prompt the user to indicate a category of the sensitive content for associating the corresponding tile.

Upon receiving the user's indication, the content rendering system 104 may transmit such information to the content tagging system 102. Thereafter, the content tagging system 102 may tag such tiles of the frames of the content with the user-selected category of sensitive portion. Accordingly, the content rendering system 104 may render the subsequent portions of the content, based on the tagging done based on the instruction.

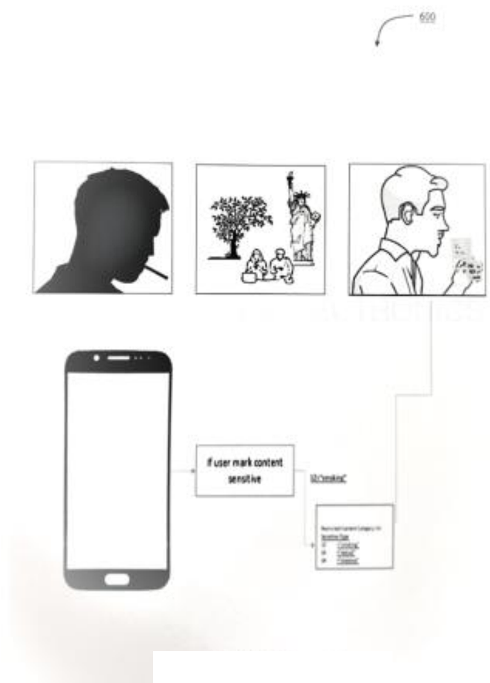


Figure 4.6 Example of the tagging of sensitive portion

Figure 4.6 illustrates a block diagram 600 depicting an example of the tagging of sensitive portion of the content, based on user instruction, according to an embodiment of the present disclosure. As illustrated, while watching the content, the user may select a tile of the frame for being tagged with a category of "Smoking". Subsequently, any subsequent portion of the content showing smoking is automatically tagged by the content tagging system 102, and then censored by the content rendering system 104.

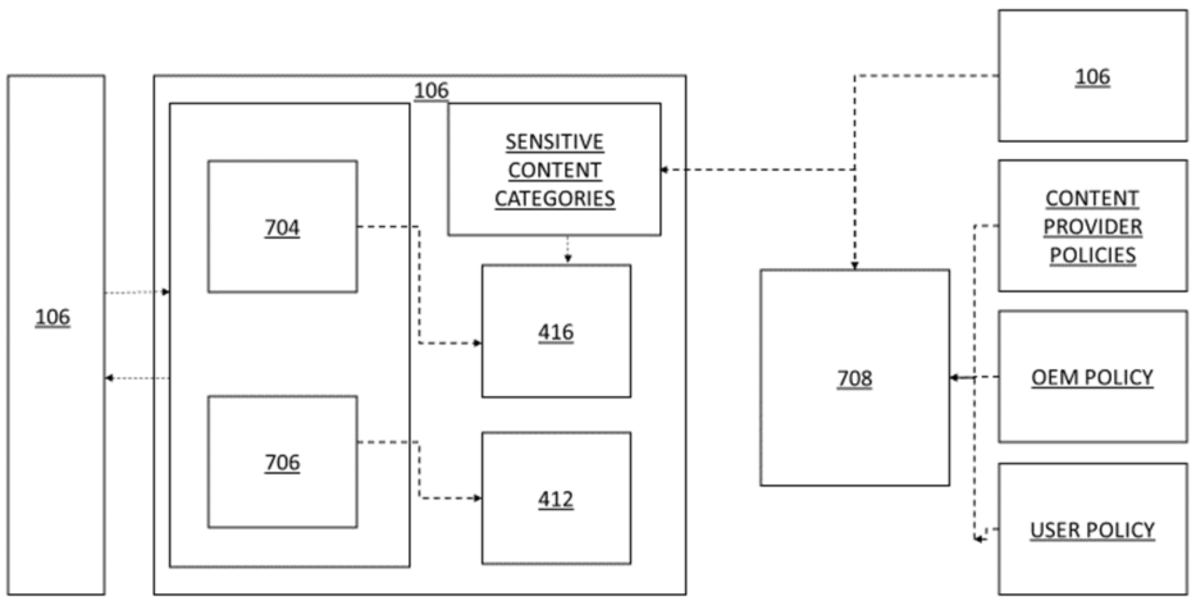


Figure 4.7 operations of the content tagging system

Figure 4.7 illustrates another environment 700 depicting operation of the content tagging system 102 and the content rendering system 104, according to another embodiment of the present disclosure. As illustrated, the server 106 is shown to be in communication with the client device 108. In an embodiment, the client device 108 may include, but is not limited to, an adaptive streaming engine 702, the decoding module 412, and the rendering module 416. The adaptive streaming engine 702 may further include, but is not limited to, a playlist parser module 704 and a tile stitching module 706.

In an embodiment, the adaptive streaming engine 702 may request the server 106 to share a playlist. In response, the server 106 may provide the client device 108 with the playlist. The playlist parser module 704 may retrieve information related to tiling of the frames of the content. For example, the information may relate to categories of sensitive content associated with each tile of each frame.

In an embodiment, the adaptive streaming engine 702 may request the server 106 for sharing specific tiles of a frame. Further, the tile stitching module 706 may stitch the received tiles and share the stitched tiles with the decoding module 412. The decoding module 412 may decode the tiles and share the decoded tiles with the rendering module 416. The rendering module 416 may then render the content based on at least one sensitive content category. Further, a user equipment 708 may stream the content, based on at least one of content provider policies, OEM policies, and user policies.

Therefore, the adaptive streaming engine 702 may download and parse the playlist of the content from the server 106, informing the client device 108 about the sensitive portion of each tile. Such information can then be matched with viewing policies and only the tiles that do not fall within the purview of any category of sensitive content may be played normally. Further, the tiles with sensitive portion may be censored based on the viewing policies before being shown to the user.

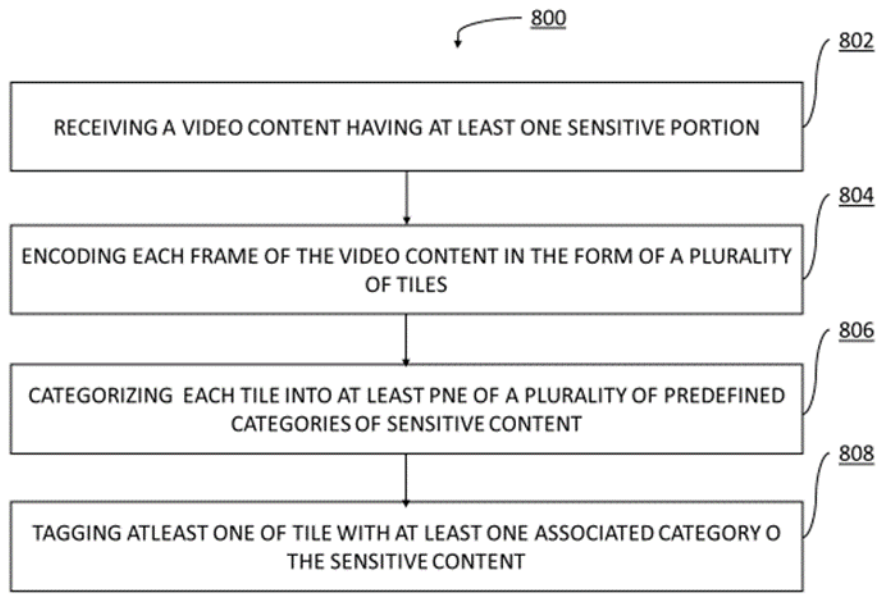


Figure 4.8 Server system for generating sensitive tiling content

Figure 4.8 illustrates a method 800 for tagging a content having sensitive portion, under an embodiment of the current disclosure. In an embodiment, the approach 800 might be a computer-implemented method 800. In an embodiment, the method 800 may be executed by the processor 202. Further, in brief, details of the present disclosure that are explained in details in the description Fig 4.1, Fig 4.2, Fig 4.3, Fig 4.4, Fig 4.5, Fig 4.6, Fig 4.7, and Fig 4.8 are not explained in detail in the description of Fig 4.8.

At the block 802, the method 800 includes receiving the video content having at least one sensitive portion. In an embodiment, the video content may be formed of the plurality of frames. In an embodiment, the receiving module 210 of the content tagging system 102 may receive the video content.

At the block 804, the method 800 includes encoding each frame of the video content in form of the plurality of tiles. Each tile may be indicative of a sub-frame. In an embodiment, the encoding module 212 may encode each frame in form of the plurality of tiles. At the block 806, the method 800 includes categorizing each tile into at least one of the pluralities of predefined categories of sensitive content. In an embodiment, the categorizing module 214 may categorize each tile into at least one of the plurality of predefined categories.

At the block 808, the method 800 includes tagging the at least one tile with the at least one associated category of the sensitive content. In an embodiment, the tagging module 216 may tag at least one tile. In an embodiment, the method 800 may include generating the playlist file indicative of the video content having the sensitive portion tagged with the associated predefined categories. In an embodiment, the generating module 218 may generate the playlist file.

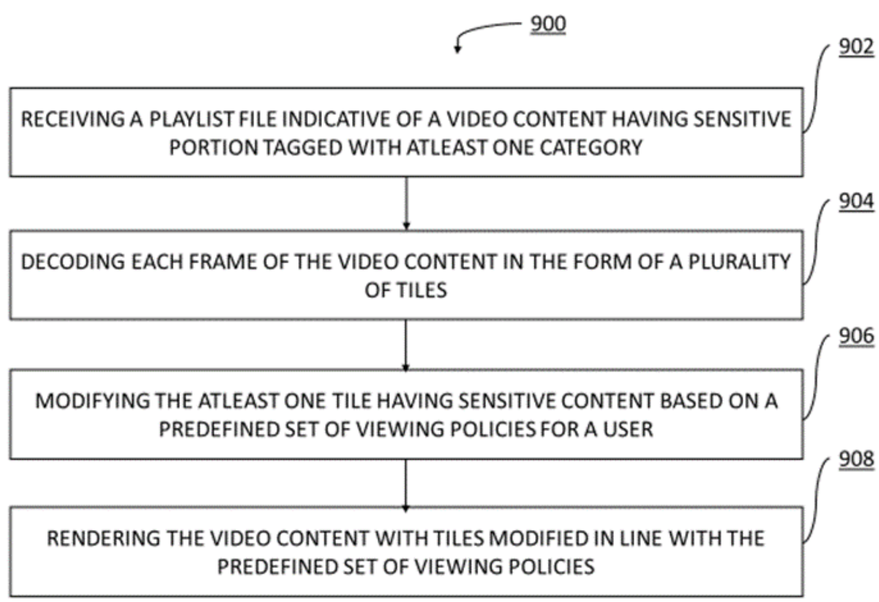


Figure 4.9 Client system for rendering sensitive tiling content

Figure 4.9 illustrates a method 900 for rendering content having sensitive portion, under an embodiment of the current disclosure. In this embodiment, the approach 900 might be a computer-implemented method 900. In an embodiment, the method 800 may be executed by the processor 402. Further, in brief, details of the present disclosure that are explained in detail in the description of Fig 4.1, Fig 4.2, Fig 4.3, Fig 4.4, Fig 4.5, Fig 4.6, Fig 4.7, and Fig 4.8 are not explained in detail in the description of Fig 4.9.

At a block 902, the method 900 includes receiving a playlist file indicative of a video content having sensitive portion tagged with at least one of a plurality of predefined categories of sensitive content. In an embodiment, the method 900 may include receiving the playlist file generated by the content tagging system 102. Further, the plurality of categories may be defined based on at least one of the age-based restrictions, the demography-based restrictions, and the language-based restrictions. In an embodiment, the receiving module 410 of the content rendering system 104 may receive the playlist file. At a block 904, the method 900 includes decoding each frame of the video content in form of the plurality of tiles. Each tile may be indicative of a sub-frame. Further, at least one tile may be tagged with at least one associated category of the sensitive content. In an embodiment, the decoding module 412 may decode each frame of the video content.

At a block 906, the method 900 includes modifying the at least one tile having the sensitive content, based on a predefined set of viewing policies for the user. In an embodiment, the modifying module 414 may modify at least one tile.

At a block 908, the method 900 includes rendering the video content with tiles modified in-line with the predefined set of viewing policies. In an embodiment, the rendering module 416 may render the video content. In an embodiment, the method 900 may include detecting, while the video content is being rendered to the user, presence of a sensitive portion in one of the subsequent frames of the video content. In an embodiment, the method 900 may include notifying the user of the presence of the sensitive portion in the subsequent frame, before the sensitive portion is rendered to the user.

Chapter 5: Implementation Details

Present system of any media content consists of manual content management as per the policy followed and then segmentation and conversion into the DASH content which consist of Media segment and the MPD file, which can be supplied to client on request. Client will make an HTTP GET request to request the content from server, there is no modification done at client side on sensitive content , it is considered to be server job, One positive aspect is it will keep the client simple and light but the problem it encounters is that the content need to be modified at every change in policy or as different policy need to be followed for different geographical region as a result , same content needs to be modified differently and different copies need to be maintained at server side.

On comparison of present and proposed system, Proposed solution instead of modifying the content at server end, only extracts the sensitive information, categorize each tile with particular sensitive tag, and update the sensitive information to the MPD file, which client can download.

Once the client downloads the MPD file, it can parse all the information about the available tiles and there characteristics which include sensitive category tag as well, So once MPD parsing is done client has all the knowledge of tiles and their sensitive category as well , which can be used by the client for conditional downloading ,rendering or decoding of a particular tile.

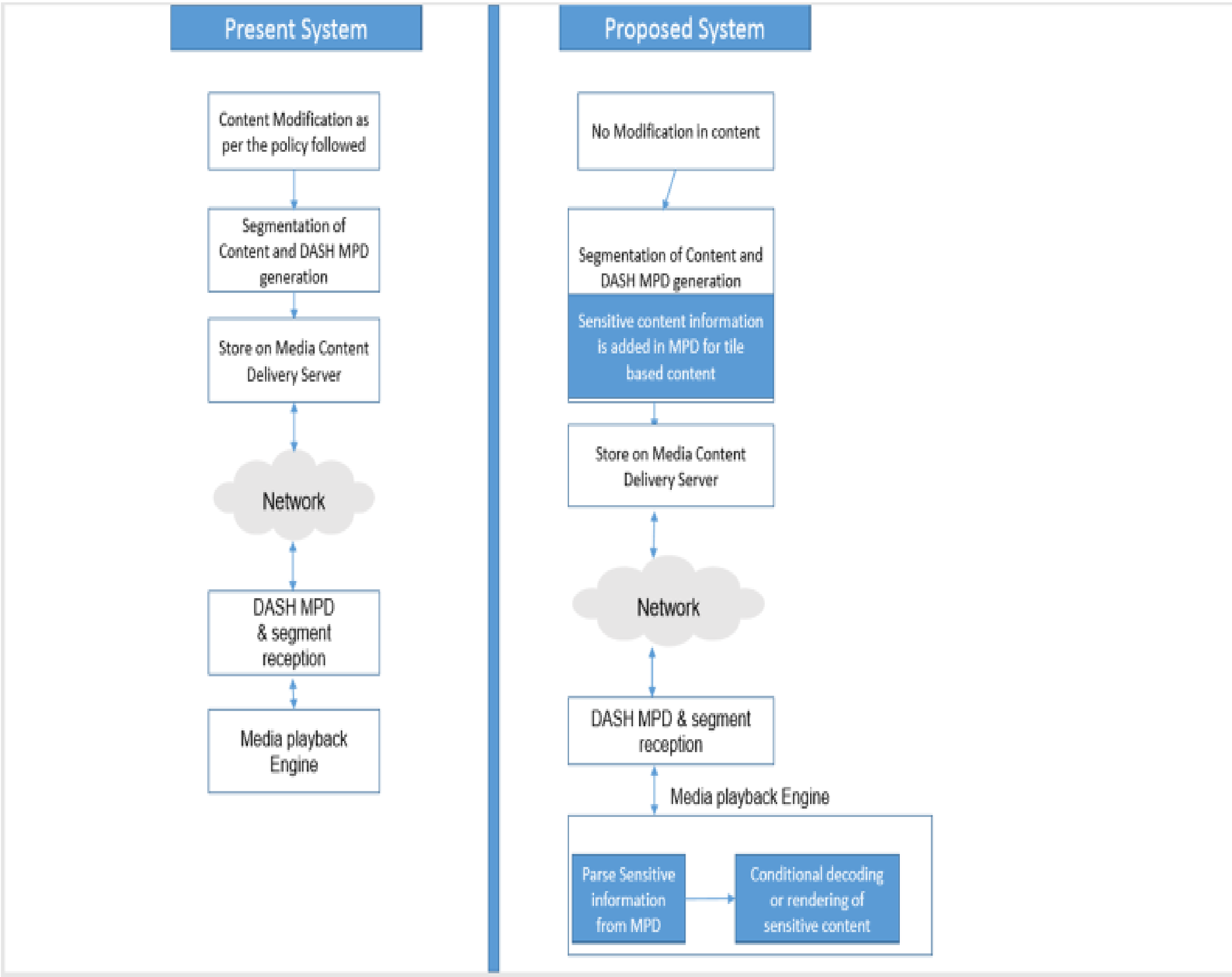


Figure 5.1 Flow diagram of Present and Proposed Solution

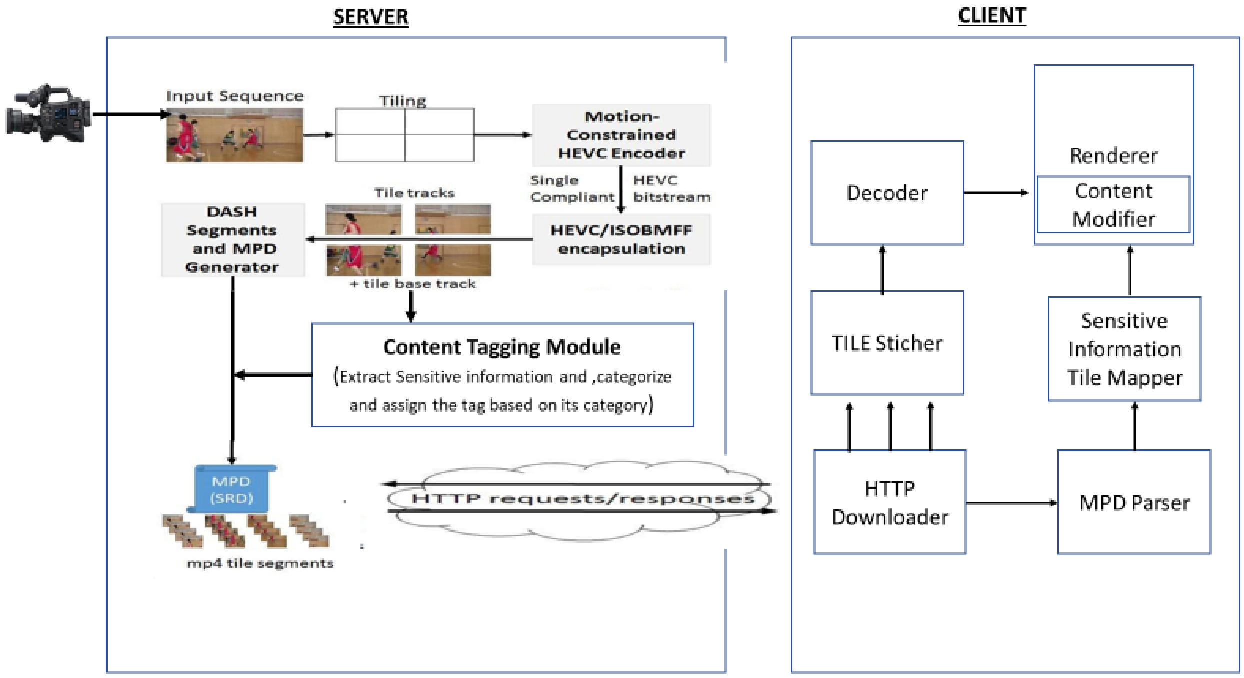


Figure 5.2 Client and Server High level Block Diagram

Proposed solution does not modify media content directly, it just analyses the content and TAG that information in MPD file under sensitive attribute which is mapped to spatial boundaries of the tile content.

To implement the proposed solution we required client server based solution where server just need to support normal HTTP based request response mechanism, So client will request MPD first and once MPD is received from server, client will parse the MPD files and request for set of tiles to make the video frame, As client already parsed the MPD it also know if the requested tile will come under normal or sensitive category.

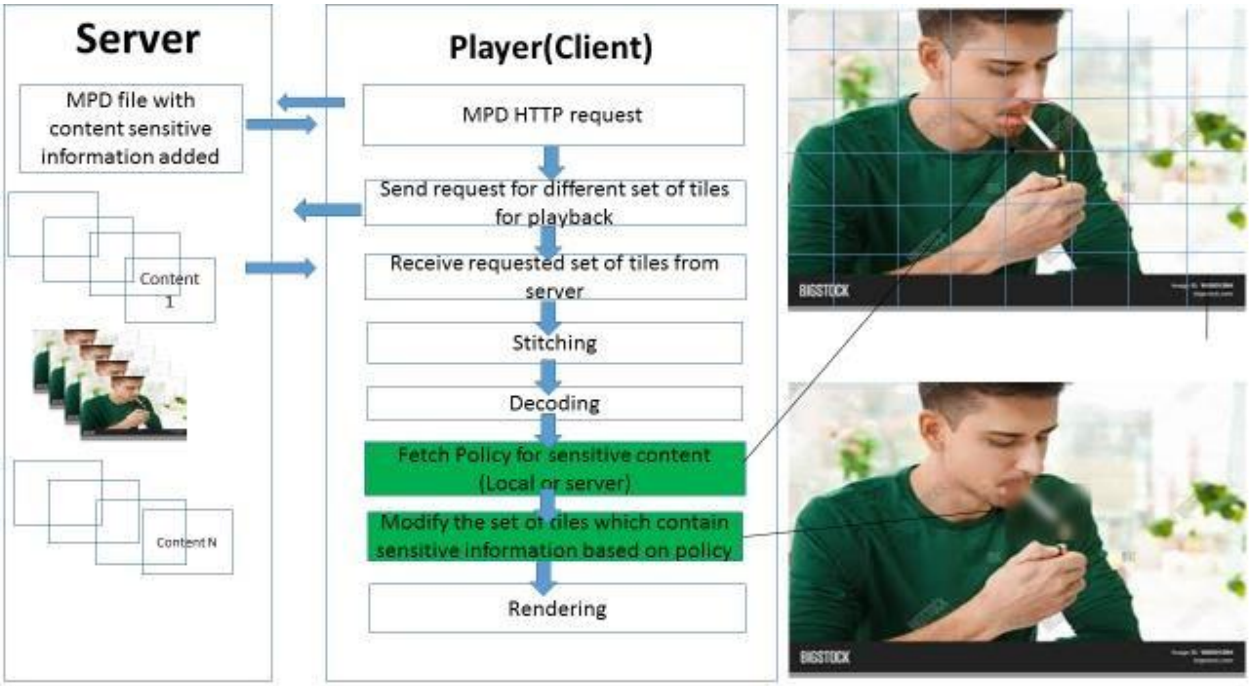


Figure 5.3 Client and Server low level implementation flow

Client will send the HTTP request to server to download all the desired HEVC tiles either in one connection or in multiple connections, As the client server interaction is HTTP based , server will just respond to client request and send all the tiles requested, Once set of tiles are received from server, client will stitch all the tiles frame by frame to make a complete frame with the help of stitcher module and pass it to a decoder, A decoder will produce the RAW video frame for rendering module.

Client is flexible enough to impose the sensitive policies at decoder level by not stitching the tiles which fall under restricted category or impose those restriction at rendering end where rendering module will use the restricted tile information and decide which tile to render normally and which tile need to be modified as per the sensitive attribute in MPD list.

The preparation of the content has been done with Kvazaar , FFmpeg and MP4Box version [GPA18] which is a implementation by GPAC

The content creation process starts by taking a video recording. This can be any kind of video in any format. We start by decoding the video to raw YUV-format. A more extensive explanation of the YUV color space can be found [Mic18]. For now, it is important to know what YUV means which is explained as below: -

- Y stands for the luminance, which is the amount of light, leaving from a point in any direction being that it is emitted, passes through or reflected from a surface.
- U stands for the horizontal axis in the color space representing the blue plane.
- V stands for the vertical axis in the color space representing the red plane.

The conversion of the video to a raw YUV video is because of Kvazaar. Kvazaar expects input in YUV420p format at 8 bit depth. Understanding that 4:2:0 stands for only half of the bits going to the color space while double the bits go to the luminance is enough for the YUV content. Furthermore, represents the 8 bit depth the amount of bits used to save the luminance and color.

To convert the video from any video in any container to a raw YUV420p 8 bit depth video, FFmpeg can be used.

- -i defines the input video.
- -c:v is a bit more complicated as the -c stands for defining the codec and :v is a way of defining that this codec is specifically meant for video. The encoder that will be used is rawvideo.
- -pix fmt defines the pixel format; as previously explained, this has to be YUV420p.
- <output video.yuv> is the last parameter which is the name of the output video. $ ffmpeg -i <input_video > -c:v rawvideo -pix_fmt yuv420p < output_video.yuv >

Once the raw video has been created, it can be used as input for Kvazaar. Kvazaar will take the raw video as input and create a HEVC tiled video, Next step is using MP4Box, provided by the team of GPAC themselves. Even though this toolkit is one with many more features [GPA18], MP4Box will only be used to create a MP4 file which has all tiles split up in different tracks from the created HEVC in the previous step, the input video.hvc will be split up by means of tiles. Each tile will be assigned to a MP4 stream.

However, there will be a track 1 that contains all non-VCL NAL units of the created MP4. This means that the generated MP4 file will have amount of tiles + 1-streams. This extra track will contain all non VCL NAL units needed to process the tiles, each track contains one tile region, besides track 1 which contains only non-VCL NAL units giving a total of amount of tiles + 1 tracks to the generated MP4.

Last step in the generation of tiled HEVC MPEG-DASH content is the creation of the MPEG-DASH content from the created HEVC tiled MP4 with MP4Box, GPAC will generate the MPD and Media tile content as an output, The MPD will contain as many adaptations sets as there are tile tracks and tile base tracks in the source file.

Apache server is setup and the generated content is placed on location which can be accessed by client through HTTP. Android based Exoplayer is chosen for the development of HTTP based tiled video playback, As Exoplayer already have MPEG-DASH playback capability, but Exoplayer does not have any support for HEVC tiled playback, so Following changes are made in Exoplayer to support the HEVC tiled playback

a) MPD parser is modified to parse SRD (spatial representation Description) information, which enables player to know it is a tile-based playback.
b) HTTP Downloader module is modified to support multiple connection download, to download multiple tile tracks parallelly.
c) Tile Stitch module is written from the scratch and integrated into Exoplayer, which convert the multiple tile data into a single video frame and feed to HEVC decoder.
d) OPENGL based rendering module is been written, which enable the proper rendering of tiles on display.

Once the player is capable of playback of tile HEVC streams with above changes,

Following changes are done to implement the sensitive content management solution :

a) MPD parser is modified to parse the sensitive tag in MPD parser, so client can fetches the sensitive information of each tile.
b) Create the mapper functionality which map the sensitive tag with each tile.
c) Conditional rendering is implemented in opengl renderer, which fetch the sensitive information TAG and restrict or modifies the restricted tile by creating a blur effect for the tiles lie in restricted category.

Chapter 6: Conclusion

Proposed solution offers a comprehensive approach of tagging and rendering the content based on the presence of sensitive portion. The proposed approach eliminates the need of manual intervention and high server storage cost in media content management. The tiling of the frames of the content for being tagged with a respective category of the sensitive portion allows for saving of storage space and offers efficient following of the viewing policies. Adaptation of the content markup 102 and the content rendering system 104 is controlled by the client, to which the client continuously downloads and play the small media files segments and dynamically shifts between different media files of distinct quality provided by HTTP servers. The user can download only the required quality, depending on the condition, based on the configuration of the adaptation.

Therefore, integrating the closed concept with adaptive HTTP flow allows new streaming potentials. The user can only adaptively stream one or more interesting tiles to lower decoding requests. Although all tiles are provided, the new configurations can be made to allow the client to choose between different tile qualities. This reduces total bitrate though retaining the top quality of interesting tiles. Therefore, the proposed approach saves a lot of server storage space. Further, only MPD files need to be changed in order to accommodate any change in the viewing policies. Accordingly, the content will remain unaffected by the change in the viewing policies.

Moreover, delivery of media content will be more flexible as the present disclosure provides multiple options for applying different policies. Therefore, the present disclosure offers the content tagging system 102, the content rendering system 104, the method 800, and the method 900 that are comprehensive, flexible, accurate, and intelligent.

References

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth, and M. Zhou, “An overview of tiles in HEVC,” *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 969–977, Dec. 2013.

[3] I. Sodagar, “The MPEG-DASH standard for multimedia streaming over the Internet,” *IEEE Multimedia*, vol. 18, no. 4, pp. 62–67, Apr. 2011.

[4] J. Le Feuvre, J.-M. Thiesse, M. Parmentier, M. Raulet, and C. Daguet, “Ultra high definition HEVC DASH data set,” in *Proc. 5th ACM Multimedia Syst. Conf.*, New York, NY, USA, 2014, pp. 7–12. [Online].

[5] N. Quang Minh Khiem, G. Ravindra, A. Carlier, and W. T. Ooi,“Supporting zoomable video streams with dynamic region-of-interest cropping,” in *Proc. First Annu. ACM SIGMM Conf. Multimedia Syst.*, New York, NY, USA, 2010, pp. 259–270. [Online].

[6] A. Mavlankar and B. Girod, “Spatial-random-access-enabled video coding for interactive virtual pan/tilt/zoom functionality,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 5, pp. 577–588, May 2011.

[7] M. Inoue, H. Kimata, K. Fukazawa, and N. Matsuura, “Interactive panoramic video streaming system over restricted bandwidth network,” in *Proc. Int. Conf. Multimedia*, New York, NY, USA, 2010, pp. 1191– 1194. [Online].

[8] F. Dai, Y. Shen, Y. Zhang, and S. Lin, “The most efficient tile size in tile-based cylinder panoramic video coding and its selection under restriction of bandwidth,” in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2007, pp. 1355–1358.

[9] K. Q. M. Ngo, R. Guntur, and W. T. Ooi, “Adaptive encoding of zoomable video streams based on user access pattern,” in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, New York, NY, USA, 2011, pp. 211–222. [Online].

[10] H. Wang, V.-T. Nguyen, W. T. Ooi, and M. C. Chan, “Mixing tile resolutions in tiled video: A perceptual quality assessment,” in *Proc. Netw. Oper. Syst. Support Digit. Audio Video Workshop*, New York, NY, USA, 2013, pp. 25:25–25:30. [Online].

[11] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, “Isolated regions in video coding,” *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 259–267, Apr. 2004.

[12] C. C. Chi *et al.*, “Parallel scalability and efficiency of HEVC parallelization approaches,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1827–1838, Dec. 2012.

[13] Y. Sanchez, R. Globisch, T. Schierl, and T. Wiegand, “Low complexity cloud-video-mixing using HEVC,” in *Proc. IEEE 11th Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2014, pp. 213–218.

[14] C. Feldmann, C. Bulla, and B. Cellarius, “Efficient stream-reassembling for video conferencing applications using tiles in HEVC,” in *Proc. Int. Conf. Adv. Multimedia (MMEDIA)*, Venice, Italy, 2013, pp. 130–135.

[15] T. Schierl, M. M. Hannuksela, Y.-K. Wang, and S. Wenger, “System layer integration of high efficiency video coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1871–1884, Dec. 2012.

[16] J. Le Feuvre, C. Concolato, J.-C. Dufourd, R. Bouqueau, and J.-C. Moissinac, “Experimenting with multimedia advances using GPAC,” in *Proc. 19th ACM Int. Conf. Multimedia*, New York, NY, USA, 2011, pp. 715–718.

[17] X. Wang *et al.*, “DASH(HEVC)/LTE: QoE-based dynamic adaptive streaming of HEVC content over wireless networks,” in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Nov. 2012, p. 1.[18] Y. Sánchez, R. Skupin, and T. Schierl, “Compressed domain video processing for tile based panoramic streaming using HEVC,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 2244–2248.