**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG2017/N17327**

**January, Gwangju, Korea**

|  |  |
| --- | --- |
| **Source** | **Requirements** |
| **Status** | **Approved** |
| **Title** | **Use cases and requirement for NBMP (v3)** |
| **Authors** | Kyungmo Park, Rufael Mekuria |

1. **Introduction**

MPEG has developed various technologies for multimedia coding and transport, such as AVC/HEVC, 3D audio, MPEG-2 TS, ISOBMFF, DASH, and MMT. These technologies have been widely adopted and are heavily used by various industries in various applications, such as digital broadcasting, audio, and video streaming over the Internet, in mobile terminals, etc.

In order to develop standardized and efficient solutions for network-based media processing (NBMP), especially given the recent increase in demand for distribution of MPEG media in next generation network environments such as 5G, MPEG evaluates and addresses the current limitations of available standards in the MPEG media distribution area including taking considerations of processing units in networks and challenges in emerging network environments into account.

1. Interfaces for media processing functions in networks/cloud
2. Supplementary information for media processing
3. A format for standardization of chaining and composition of network based media processing functions
4. NBMP will also provide API to media processing
5. These specifications will together compose the NBMP framework

To enable interoperable distributed media processing in these networks and cloud environments, NBMP defines interfaces between media processing functions. These interfaces will allow configuration of the media processing functions and efficient exchange of media data. Second, these interfaces will facilitate a better tradeoff between resources (e.g., bandwidth, compute, storage) in this environment by having supplementary information useful for the media processing functions.  Further, NBMP will define formats for chaining and composing such media processing functions and provide an API for the media processing. These functionalities together compose the NBMP framework for interoperable network-based media processing.

NBMP is a framework that allows service providers and end users to describe media processing operations that are to be performed by the network. NBMP describes the composition of network-based media processing services out of a set of network-based media processing functions and makes these network-based media processing services accessible through Application Programming Interfaces (APIs).

An NBMP media processing entity performs media processing tasks on the input media data and the related metadata. NBMP also provides Control Functions that are used to compose and configure the media processing. In addition, NBMP provides transport methods for communication between media source and media processing entities.

This document contains use cases and draft requirements for potential NBMP standards addressing the needs of network-based emerging applications.

1. **Objectives**

In this document, a network-based media processing system is defined as follows:

1. **NBMP system:** A system for processing that is performed across processing entities in the network. It consists of media source entity, media processing entity and media sink entity.
2. **Media Source Entity:** the capture entity which provides the raw media content to be processed, for example, a digital camera, a microphone, an encoder, or persistent storage.
3. **Media Processing Entity:** performs a media processing task(s) on the input media, which may be the output of a prior media processing entity.
4. **Media Sink Entity:** consumes the output of media processing entity through existing delivery methods, for example through download, DASH, MMT, or other means. This entity is not standardized through this work.
5. **Scope**

The NBMP framework will define interfaces between the Media Source and the Media Processing Entity and among the Media Processing Entities, that will allow users to access the framework, configure the media processing with, upload/stream media data to the network for media processing, and utilize the processed media and the resulting metadata in real-time or in a deferred way.

The media and metadata formats that are used between Media Processing Entities in a media processing pipeline are also within scope. Workflow description that is used to orchestrate media processing entities and to compose media processing services into a pipeline of media processing entities is also in scope. NBMP format can be transmitted through existing delivery method, is also in scope, for example, NBMP format via HTTP.

Figure 1 depicts the NBMP architecture that will be used as a reference architecture to scope the NBMP work:

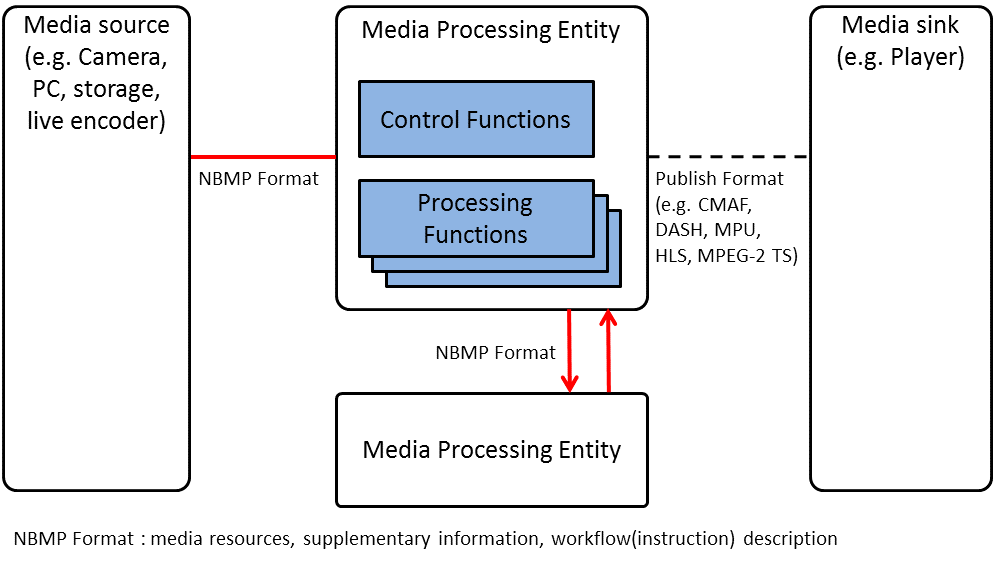


Figure Potential framework for Network-Based Media Processing system

The definition of NBMP format such as media resources, supplementary information, workflow(instruction) description will be described with the following aspect.

1. **Media Resources (M)**
   * Audio/ Visual (raw data) bit-stream, compressed stream, packaged stream
   * Encoded data, packaged data
2. **Supplementary information  (S)**
   * Metadata
   * auxiliary information (side information)
3. **Workflow description (describes the ingestion, processing steps, distribution/ end-to-end procedures, wish list from content provider)  (W)**
   * Java script
   * server manifest
   * Instruction for media processing
   * QoS requirement descriptor
   * QoS/QoE feedback management

The NBMP framework consists of two generic functions inside media processing entities: Processing Functions, and Control Functions.

1. **Processing Functions:** provide functionalities for media processing and analysis given control instructions from the control functions. These functions are considered to generate processing output information, which includes the following functions:
   * **Media processing function:** the core media processing function which performs processing of the input media that can generate output media or metadata. Examples of media processing are; content encoding, decoding, content encryption, content conversion to HDR, content trans-multiplexing of the container format, streaming manifest generation, frame-rate or aspect ratio conversion, and content stitching, etc.
   * **Analytics function:** provides functionalities for analysis of the logged information and makes the analysis report available via request.
2. **Control Functions:** provide functionalities for controlling and management of the media processing tasks and workflows, and how input media is processed into output media published into the media sink. A key control function is the “service/workflow manager” function.
   * **Service/Workflow function:** provides functionality for the composition of media processing workflows by chaining a set of media tasks. This includes matching the output of each media processing entity to the input of the succeeding media processing entity.
   * **Monitoring Function:** provides functionalities for monitoring the processing pipeline, and to guarantee media processing task execution, correctness, or to detect failures during media processing.
   * **Logging Function:** logs information about media processing and/or services
   * **Pub/Sub Function:** provides functionality for messaging and information exchange, including content retrieval and publishing. This may be used to trigger processing to start at one media processing entity after processing ends in a previous media processing entity.
   * **Security functions:** provides functionalities for ensuring user’s and content security, including user authentication, content encryption, and other functionalities, as necessary

Editor’s note: Add some text to describe the relation between functions and NBMP format

1. **Use cases**

The following section contains use cases of media processing applications that may benefit from network-based media processing.

* 1. ***Use case: Network-assisted media quality enhancement*** 
     1. ***Network-assisted VR stitching [1]***

High quality 360° video content is uploaded to the Cloud to provide high-quality Virtual Reality (VR) streaming service to the users. The service provider wants to save mobile data usage for users that are experiencing virtual reality through streaming and providing the high-quality VR content that is stitched with powerful computation environments in Cloud. Hence, the service provider wants to take advantage of the high processing capability of the Cloud. The high processing capability of the Cloud facilitates the machine learning and media processing in the Cloud to provide advanced VR streaming services. Here the advanced VR streaming service aims at offering high-quality VR experiences with low data consumption.

To reduce the data usage of mobile in downlink, a tiling and frame-packing based Field-of-View (FoV) VR streaming (alternatively called view-dependent VR streaming) are used. Here, the tiling and frame-packing based FoV VR streaming allow the server only to send video data that correspond to the user’s current viewport in high quality. A lower quality encoding of the whole panorama is streamed as a fallback as described in the following:

1. **Tiling approach:** A high-quality partition of the 360° scene and a low-quality whole panorama are sent to the users as separate video streams.
2. **Frame packing approach:** A high-quality partition of the 360° scene and a low-quality version of the whole 360° scene are frames packed together into a single frame packed video stream and sent to the user. During the packing operation, rotation may be applied.

In both the tiling and frame packing based FoV VR streaming, the equi-rectangular projection (ERP) is typically used as input format. Also, multiple encodings of each tile may be required to address different bandwidth and device capabilities and needs. This results in a significant amount of processing. Therefore, the service provider prefers to perform this VR processing in distributed networks. Compared to preparing the FoV VR content in a centralized manner, the use of distributed networks would significantly reduce processing time and save network bandwidth.

The service provider may want to apply advanced machine learning techniques to optimize its VR encoding configurations. This processing functionality can be provided to content providers as a configurable “Processing as a Service” offering.

* + 1. ***Network-assisted video up-scaling [2]***

A mobile user captures a video scene with her mobile phone. The video sequences captured by a mobile phone typically have an odd picture aspect ratio (PAR) due to a narrow width and/or height. When videos with such a PAR are played on a wider screen such as Desktop PC or Smart TV, the content will usually not cover the whole screen, which degrades the user experience. If traditional up-scaling is applied to match the video content to the presentation screen, the user experience might be even worse because it distorts the objects on the scene by applying a larger scaling factor on one dimension of the video.

A service provider wants to offer high processing capacity of the Cloud to provide intelligent video up-scaling feature to improve the use experience. The service provider may want to provide content adaptive up-scaling. The adaptive content up-scaling may be carried out either horizontally or vertically or on both dimensions. In intelligent up-scaling, the video scene is up-scaled so that the central objects of the video scene become more harmonious in terms of the ratio of central object(s) and background, the position of the central object(s) in the video background, etc. An example of such an up-scaling is shown in Figure 2, wherein the top, mid, and bottom row figures show the original captured video frames, traditional up-scaled video frames, and intelligently up-scaled video frames, respectively.

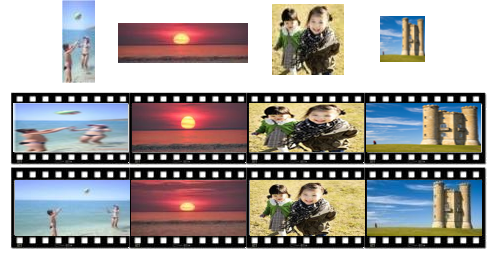


Figure Example of video (top), conventional up-scaling (middle), contents preserved up-scaling (bottom)

To provide advanced intelligent up-scaling, the service provider typically deploys Artificial Intelligence (AI) to improve the video quality and the resulting user experience. The service provider is being aware of the importance of learning the large data in machine learning. The service provider wants to deploy its unique advantages of the distributed networks or Cloud in the big database, high and parallel processing capabilities, and a complete set of APIs to improve the AI-based intelligent up-scaling. Being aware of the unique advantages of the Cloud or distributed networks in machine learning, the service provider expects that the Cloud-based and distributed networks based intelligent up scaling significantly improve the user’s quality of experience compared to the mobile device or personal computer-based approaches; hence further improves the business of the service provider.

Different users typically have different perceptions of the quality of the scaled videos, which is another important motivation of locating the scaling processing in distributed networks as close as possible to the end users. By processing the up-scaling in the distributed networks, the service provider not only facilitates the intelligent up-scaling to meet users’ requirements but also saves network bandwidth as the up-scaling processes are carried out at proximity from the user.

The distributed network ecosystem-based solution has its unique advantage regarding saving resources compared to the traditional broadcasting systems. In network-based media processing, the complete media processing functionalities could be provided as an ecosystem and used by various service providers in a shared manner, which will significantly save resources compared to the traditional broadcasting system where each of the service providers needs to have its own media processing functionality.

* + 1. ***Mobile edge encoding for the adaptive streaming [3]***

This use case proposes a new architecture of adaptive streaming service utilizing distributed encoding concept as shown in Figure 3. In this architecture, at the edge of the network instead of a simple cache, a mobile edge video encoder is used sequentially. A mobile edge video encoder generates a single version of video bitstream which can be easily converted into a version of bitstream with specific resolution and bit rate by a mobile edge video encoder. A streaming server populate a bitstream created by a original video encoder to the mobile edge video encoders. According to the available bandwidth of the last mile, a mobile edge video encoder create a version of bitstream adequate for available bandwidth.

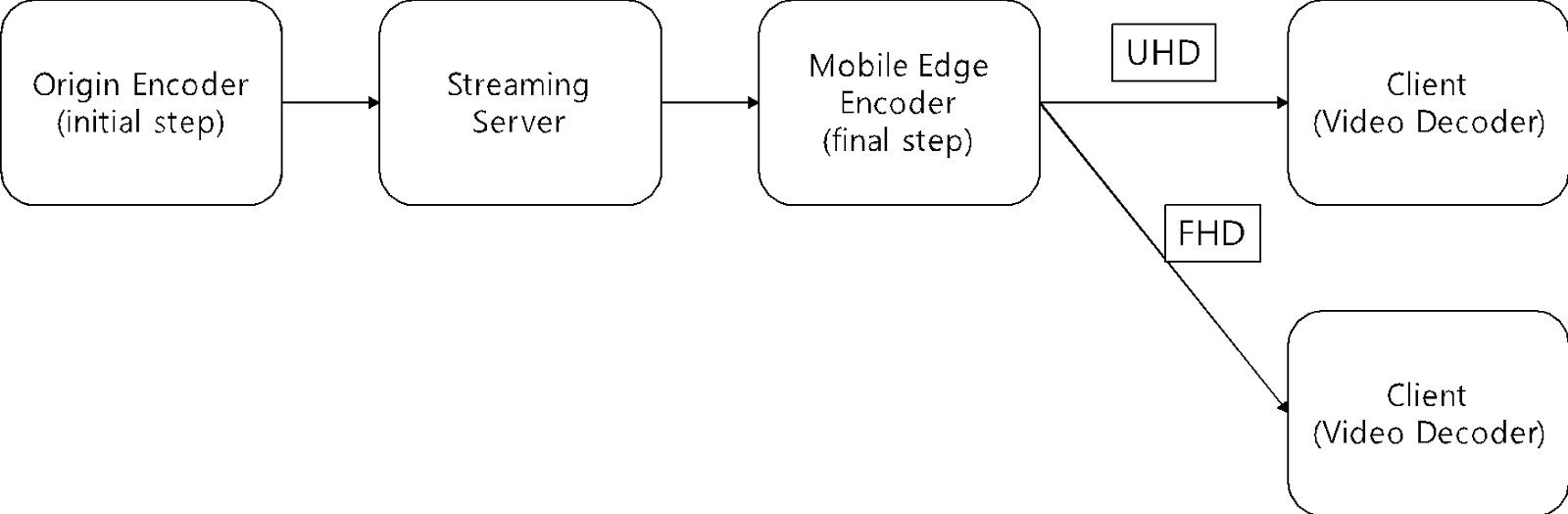


Figure Adaptive Video Streaming Service Architecture with Distributed Encoding

This architecture will allow the edge node to generate the bitstream fully utilizing available bandwidth of the last mile by using a mobile edge video encoder. Therefore, the quality of video service will be improved compared to the conventional adaptive bit rate streaming case. For example, if a mobile legacy encoder generated bitstreams for 1Mbps, 2Mbps, and 3Mbps respectively and available bandwidth at last mile is 1.5Mbps then a bitstream for 1Mbps will be delivered. So, only two third of available bandwidth is utilized in this case. However, if a mobile edge encoding architecture is used as shown in Figure 6, a mobile edge video encoder can generate a bitstream for 1.5Mbps. So, 100% of available bandwidth can be utilized in this case.

* + 1. ***Content-Aware Cloud Transcoding [4]***

This use case addresses Network delivery of Live or On-demand ABR video streaming applications.

Figure 4 shows an example of a media delivery network in which video content is prepared for live or on-demand ABR video delivery. Original video (S) is ingested and initially encoded into an initial stream (T) at an origin (A). The initial stream (T) is delivered to a node (B) within the media delivery network, down to the edge server, at which processing is applied in order to transform the initial stream (T) into ABR video representations (U1, U2, U3) that can be delivered to the end-user devices (C). Capped bit rates of these ABR representations may be defined by the service provider and given to A and B functions or adapted dynamically in function of available bandwidth in the delivery network, as described in this use case.



S

T

U1, U2, U3

U1

U2

U3

video

Initial encoding

Transcoding

**A**

**B**



T

Media source

Figure 4 Application example: Delivery of live or on-demand ABR video inside a media delivery network

In an evolution of ABR streaming, the format of ABR representations at a given capped bitrate could change dynamically in function of content characteristics. The determination of the most appropriate format may require some additional processing for the transcoders in the network. It may be more economically efficient to put this intelligence in the Origin encoder (A) and to guide the transcoding function with Metadata specifying the format of ABR representations dynamically. The high processing cost associated with the transcoding could be further reduced if the video coding process is assisted with block-level Metadata.

* + 1. ***Cloud-based 360 VR Stitching [5]***

360 VR stitching typically refers to a process of constructing a panoramic image, mono or stereoscopic, by combining imagery from multiple camera lens positions. The desired result is to create a visually seamless transition from one sensor’s perspective to another so that the appearance is that of one continuous image. The process consists of several steps. After any optional colour matching and de-noising has been applied to the per-sensor image files, the files, together with optional configuration, are ingested into the cloud processor (aka. Stitcher), in order to create left and right eye spherical/equirectangular images for use in viewing stereoscopic omnidirectional visual content using a viewing device such as a head-mounted display (HMD).





Figure input video from 8 fish-eye lenses and stitching several inputs in one panorama with overlapping areas (displayed as purple)

Due to the physical separation of these cameras/lenses and color calibration differences, some visible artifacts can occur in the interpolated (“seam”) areas of the resulting panoramic image. These artifacts tend to be more visible on objects that are closest to the camera, as the parallax difference between sensor angles is at its greatest for such objects. Hence, adjustment of seam locations can be possible to minimize these errors. Adjusting the seam locations, for instance, takes into account the width and position of the seams.

When necessary media data and metadata (i.e., calibration information for lenses, color space information, etc) is present, it is possible to perform the VR stitching in a cloud processing architecture.

Figure 6 represents how VR stitching processing function can be included in the NBMP architecture. It is envisioned that an NBMP Media Processing Entity can perform the VR processing workflow inside which VR stitching processing function can be part of the overall workflow. Having ingested the source media data (e.g., multiple fish-eye video/images) and the necessary configuration and control parameters, such a processing entity can stitch the media to an omnidirectional content and then stream it to the VR player based by utilizing a publish media format such as OMAF.

It is crucial that the necessary control and configuration information should be ingested to the processing entity to have a successful stitching process.

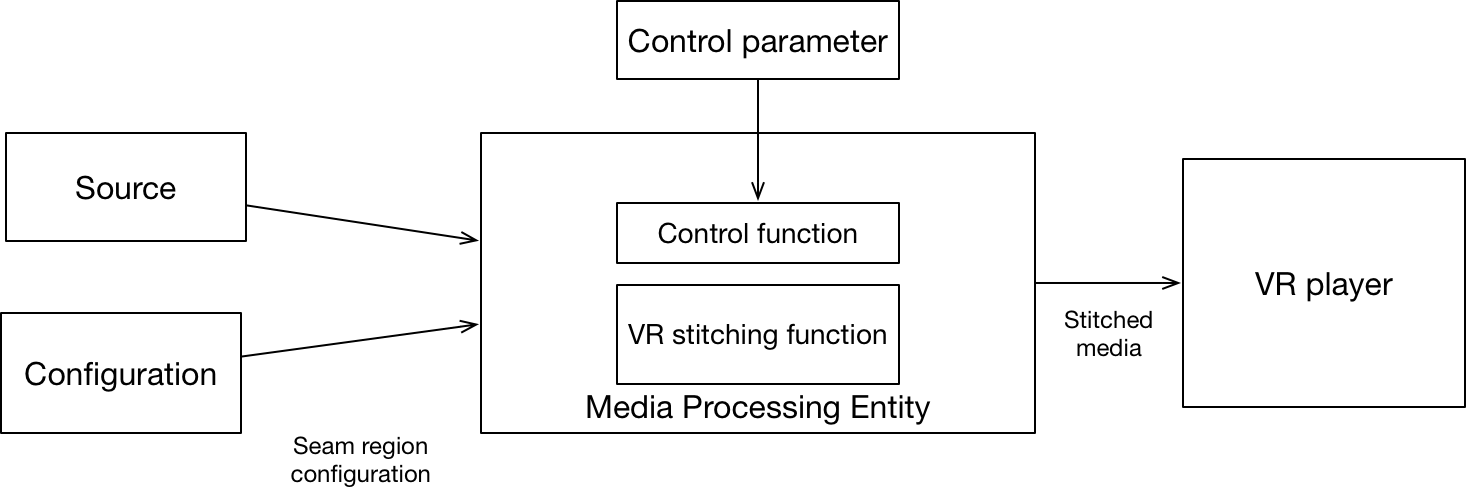


Figure 6 VR stitching mapped to NBMP architecture

* + 1. ***Cloud-based multiple capture device for VR streaming [6]***

It is common that a scene is captured with multiple cameras from multiple different perspectives simultaneously. The post-processing requires the knowledge of the relative global spatial position, e.g., the constellation of the cameras.

The process of determining the global or local location and orientation of a given camera is called camera pose estimation (aka. camera registration). One way of determining the camera locations is to use GPS data. However, the vision-based approach is more preferable when a certain level of the quality or accuracy is required, particularly, GPS does not provide orientation information directly.

The camera registration takes a vision-based approach to calculate the 2D-3D correspondences of given captured images or video frames and generate the global transformations (e.g., the positions and the rotations), plus the 3D point cloud with global geographic information.

Typically, there are two categories for vision-based camera pose estimation: 1) Structure from Motion for offline processing with typically an unordered set of images from multiple different cameras computed in the cloud with little to no time constraints; 2) visual SLAM works for real-time cases and processes continuous input frames with relatively small motion per camera. Both approaches can produce global geographic registered 3D data such as point cloud, depth maps, and registered camera localization per frame (or key frames). With such geographic registered information, some advanced video post-processing use cases can be realized.

There are many different camera types on the market. As an example, model types such as pinhole camera, fisheye camera, and panorama/omnidirectional camera, and camera setup such as single camera, and non-centric multi-camera devices can be listed. Processing a large amount of heterogeneous imagery data poses challenges to the vision-based camera pose estimation algorithms, the data management as well as the distributed computing infrastructure. A cloud-based solution should process not only the media data but also the metadata related to the capturing devices such as camera lens types, camera configurations, and other key sensor properties. Given the metadata, NBMP can cast the input imagery into a homogeneous internal data format to enable timely distributed processing.

The registration process projects the images from different lens models into a common 3D model, by their computed 2D features and then apply non-linear algorithms to estimate the correspondent locations and orientations. The process can be executed recursively to optimize the result or run in a timely fashion to deal with situations where cameras are non-stationary. After the first step, a local camera poses data can be determined. A global registration step can then start to translate the local camera poses to global camera poses. Global camera poses can be used alongside with the input images as a kind of timed data; or served statically and independently to input live video frames, assuming that the input cameras keep still. Also, the process can also output extra useful data such as 3D point cloud and depth maps.

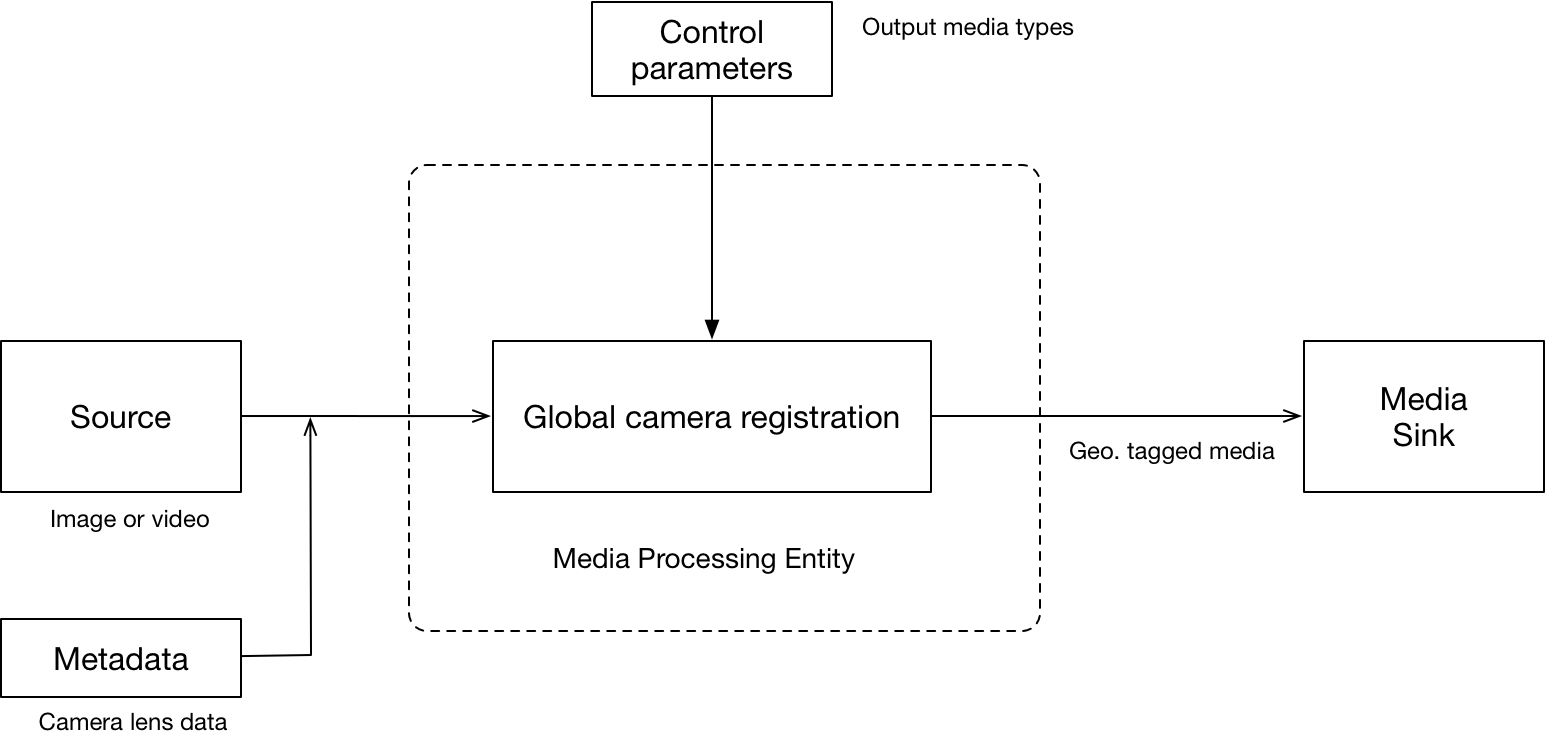


Figure 7 example : Multiple camera registration mapped to NBMP architecture

* 1. ***Use case: Network-assisted media distribution*** 
     1. ***Live Media Ingestion [7]***

Live encoders in the cloud or on-premise typically send their content to the media processing entity as soon as it becomes available. While the format typically follows formats like MPEG-2 TS or ISO Base Media, the transmission method (push-based manner (e.g., via the HTTP POST protocol)), auxiliary information and file format flavor often differ. Figure 8 shows the workflow of pushing content from a live encoder to a media processing entity that can provide additional processing such as for media streaming. NBMP should provide a base specification for the format of content ingestion from live encoders towards media processing entities.  This specification should include for example timed metadata to allow ad insertions or other meta-data like sub-titles, annotations and so on.  This could combine best of breed encoders with best in class processing entities enabling more advanced media services. A specification of the ingest format would make it easier for the service providers to deploy combinations of such functions in the cloud that are provided by different vendors. Specifically, the media ingest specification aims at the connection between the live encoders and processing entities.

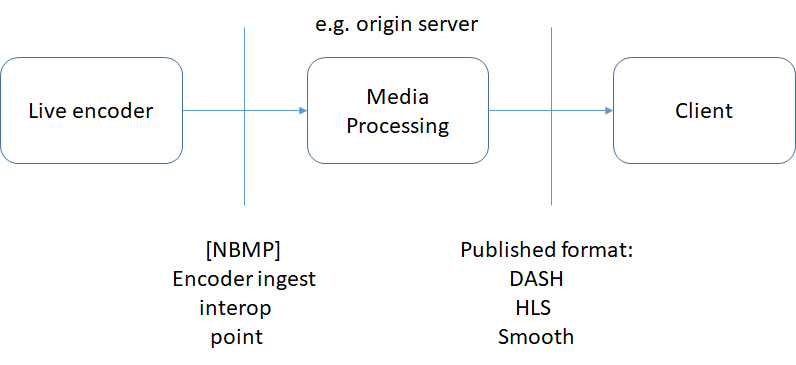


Figure Live encoder ingest workflow

* + 1. ***Online transcoding for media-aware caching [8]***

There have been three types of video transcoding strategies: In the first type，the content provider provides different versions（bit rates） of the video content stored in the video distribution device. When the user requests a video content from the server, the video distribution device sends the corresponding version to the client.

In the second type，the video provider sends only one video content with high bitrate to the video distribution device. The video distribution device then transcodes the high bitrate video content into different versions of lower bitrate contents before the users reveal their requests.

In the last type, the video provider sends the high-bitrate video content to the video distribution device, and the video distribution device transcodes the stored high bit rate video to a lower bitrate version according to the user’s request. The first two types require large storage at the distribution device, while due to the transcoding latency, the third type which facilitates instantaneous transcoding may suffer from a higher probability of video freeze，leading to the degraded user experience.

Caching is a technique to pre-fetched video content from the original content server to the distribution devices that are close to the edge users. Caching device will send the video content to users immediately if that content has already been cached. The original content server will send the remaining video content to the cache device, giving the user a lower transmission delay. As the cache server is closer to the client, the user experience can be improved as opposed to the case where the content is directly pulled from the original server.

By the idea of caching, in unit B in NBMP, we can pre-transcode only parts of the video content and store them in the memory. When the client requests reveal, the transcoded video content is first sent to the client, while the remaining video content will be served through real-time transcoding. So, with a probably not so high, the transcoding speed and the storage requirements will reduce dramatically, while user experience will degrade slightly, compared to the first approach, if the pre-transcoded video content is sufficient and the transcoding speed is fast enough.

Techniques concerning what part of the video content to be pre-transcoded and how to schedule the real-time transcoding process is therefore critical for higher user experience.

* + 1. ***Coded caching for media distribution [9]***

Code caching method is a promising caching technology which was proposed by Ali & Niesen recently. By careful design of content placement across end-user cache during off-peak hours, simultaneous multicasting opportunities are created to serve several different content requests. The coded caching method shows great potential in shifting peak-hour traffic to off-peak hours, which leads to considerable bandwidth saving in content delivery. We explain the method of coded caching by following example.

Suppose there are two media contents A for an interview of one basketball player1, and B for another interview of player2 which provided by the same provider. Two consumers are watching these interviews simultaneously at home. Conventionally, the end user facility may prefetch parts of the media content. However, once the cache misses, or one of the users change the player interview, the network needs to transport the whole new content to the users through the common delivery network.

The coded caching implementation means that caching facility only need to cache parts of the media content, let’s say A1(half of A) B1(half of B) for the first user and A2 B2 for the second user, in the placement phase. When in the delivery phase, the source only needs to deliver some coded XOR information compute from A&B base on the actual demands of the users. In this case, no matter what the choice of the users is, the server only need to transport a small size of coded media content. The caching process is shown in Figure 9.

To utilize such promising caching technology in NBMP, we propose to facilitate coded caching implementation by the Caching node in NBMP structure. As we can see, this coded cache method needs to implement near the end users, and it also needs some computational resources. Thus, firstly NBMP unit may serve as the user cache to store the bits and compute the raw information after receiving delivered information. Secondly, NBMP may also serve as the source which can generate coded information about the media content and compute the computing deliver messages base on the demands of the users.

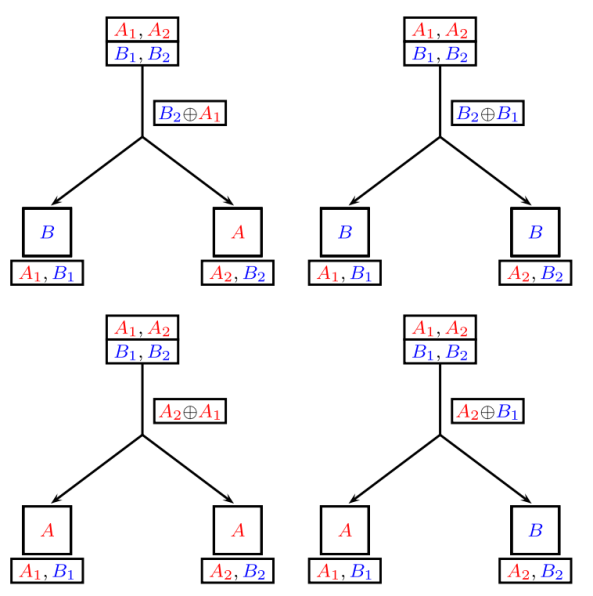


Figure example: Coded caching process

* + 1. ***Targeting caching for media distribution [10]***

NBMP node serves as family cache or dormitory cache rather than neighborhood cache. The promised range of NBMP cache serves several end users but not tens of hundreds of end users. Since that NBMP can pull precise popular contents or fragments from server or CDN nodes base on the local users’ interest, view history, and other user features.

For example, if a family like sports more than news, and they like viewing TV shows on the mobile screens, the NBMP node may pre-fetch 4K sports fragment and HD TV shows in the off-peak hours. In the peak hours, their content request may be served partially or totally by the local caching node (family cache), thus reducing content delay by avoiding access to the original content server, from which the content transport rate is often limited by the network bottleneck.

In other words, NBMP may serve as CDN end nodes. The difference is that NBMP can target precisely to the little number of users, so that improve the bit rate and increase the QoS.

* + 1. ***Large size caching [11]***

Since future media content like immersive media such as VR or point cloud always have a large scale of data volume, the end user cache cannot afford enough storage for such services. NBMP can provide a large cache near the consumer because the implementation of NBMP includes the utilization of NBMP nodes with computational and memory resource. This NBMP large size caching may serve dozens of the users with higher definition, longer fragments, lower time jitters media service.

* + 1. ***Multicast ABR Streaming [12]***

In recent years, network operators have invested in the construction of a wide range of IP multicast network for all types of live video business, which can greatly save server processing resources and network backbone bandwidth. Make the network multicast capability accessible to diverse OTT services, can greatly reduce CDN bandwidth rental costs, enhance playback quality through end-to-end service configuration on the network, and effectively solve the transient and unpredictable tidal effects of popular live OTT video services, including 4K/ 8K video, VR / AR streaming services.

Facing fierce market competition, service providers who are intent to leverage underlying network capabilities to provide optimal media distribution solutions. Multicast streaming with ABR is one of such network-based media processing and delivery solution, which enable dynamic configuration and conversion of media delivery procedure based on log statistics and analysis using existing standardized transport protocols and delivery formats.



Figure 10 Example of multicast streaming with ABR

The typical Multicast streaming with ABR use case and its end-to-end workflow in the NBMP system is shown in Figure 10 as an example.

1. Based on user’s channel request, corresponding audio/visual bit streams from Media Source are delivered as ABR unicast stream to the Media Sink.
2. M-ABR Agent in the Media Processing entity, which can be deployed on the Edge CDN node, and STB or RG, collect and report auxiliary information such as channel ID, bitrate, etc., to the M-ABR Controller.
3. M-ABR Controller (Controlling function), which is a Media Processing entity, generates hot channel list statistically based on logs information, updates and sends Multicast Channel Map configuration to the multiple network-distributed M-ABR Converters.
4. One of M-ABR Converter (Processing function), which is also a Media Processing entity, convert ABR unicast stream to ABR multicast stream after binding the hot channel to multicast configurations and optional multicast transmission protocol negotiation with the M-ABR Agent.
5. M-ABR Agent obtains metadata/service manifest (including multicast address assignment) via a unidirectional or bidirectional channel, depending on a network environment.
6. M-ABR Agent joins the multicast channel group to obtain ABR multicast stream.
7. ABR multicast stream converted to ABR unicast stream and delivered to the Media Sink
   1. ***Use case: Network-assisted media composition*** 
      1. ***Augmented Video streaming [13]***

A service provider wants to provide mixed video and AR experience to its users. In mixed video and AR, the source video sequence is overlaid with graphic images. The service provider may want to produce multiple versions of such mixed video and AR content by overlaying different graphics images on top of the video taking into consideration the user’s preferences. An example of such scenario is shown in Figure 11.

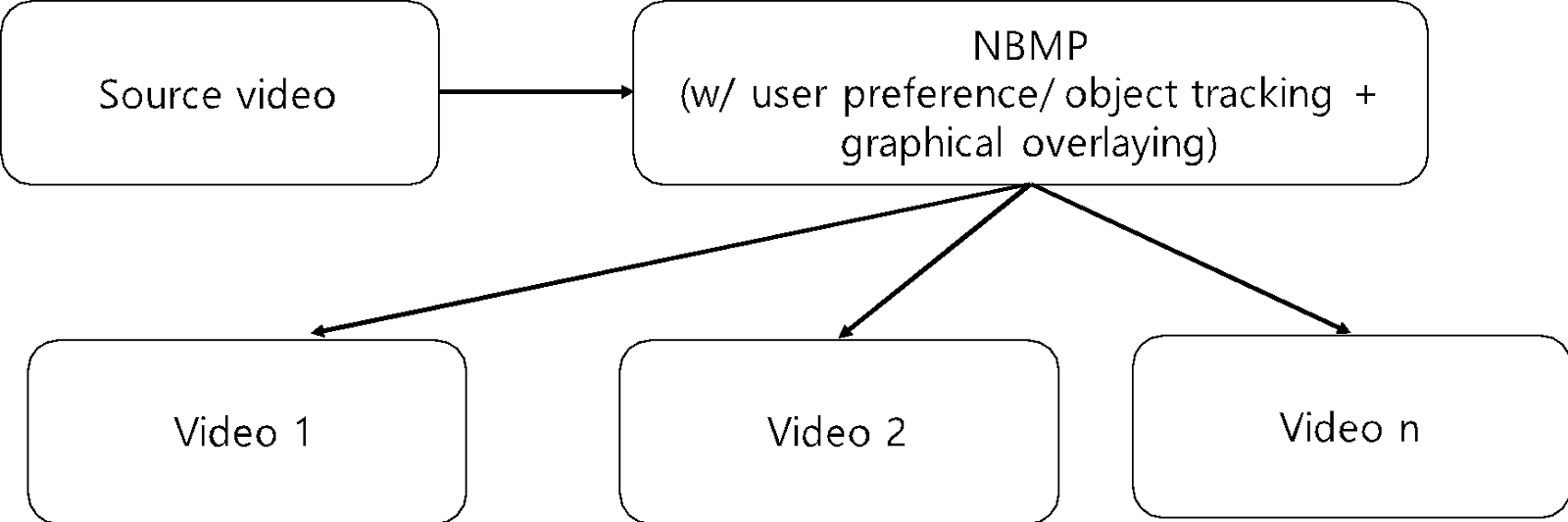


Figure Video and AR overlaying use case

Figure 12 shows an example of a volleyball sports video with overlapping graphics. In the example, the volleyball trajectory is augmented with moving graphics content that provides an augmented graphics trajectory of the moving ball. Multiple versions of such video and AR contents are prepared. The service provider then provides an appropriate version of the content to each user depending on that user’s preference.

The service provider wants to provide user aware AR experience. In a volleyball game, John is a fan of team A. In such a scenario, the service provider wants to add overlaying graphics by analyzing to cheer team A’s attack on team B. For the same volleyball game, Susan is a fan of team B. In such a scenario, the service provider wants to provide different graphics overlaying in support of team B.

To provide an advanced user-aware video and AR experience, the service provider typically deploys Artificial Intelligence (AI) to achieve the required effects. The service provider deploys its unique assets of high processing power in the cloud to analyze the video content and the user’s profile and offer a tailored user experience.



Figure example of sports video and graphics overlapping

The network-based media processing solution has its unique advantages regarding resource efficiency compared to the traditional broadcasting systems. When the media processing functionality is located in distributed networks or the Cloud, the complete media processing functionalities could be provided as Media Processing as a Service to various service providers in a shared manner, which will significantly save resources compared to traditional media processing systems, where each of the service providers needs to develop its own media processing functionality.

* + 1. ***Intelligent user-centric broadcasting [14]***

In traditional broadcasting of sports games, e.g., soccer and American football, multiple cameras are placed at various locations of the field. Video editing tools are typically used to edit multiple video scenes to produce a uniform video scene to broadcast to all viewers. In a user-centric broadcasting scenario, the service provider wants to upload multiple scenes to the Cloud and broadcast user-centric broadcasting service by utilizing the end user’s preferences. Many times a football fan might love a certain team, and he/she may be a fan of a certain football star. In user centric broadcasting, the service provider wants to provide a service so that a user who is fan of a star could continuously follow the player within a sub-scene window, which is overlapping with the main scene window, and switch between the sub-scene to the main scene upon request. In user-centric broadcasting, the service provider wants to deploy AI to analyze the user’s preference and edit the video streams based on the user’s preference automatically.

To provide an intelligent user-centric broadcast experience, the service provider typically deploys Artificial Intelligence (AI) to customize the user experience. The service provider is aware of the complexity of processing big data and fine-tuning the machine learning models accordingly and wants to deploy distributed networks or Cloud processing of big data to improve the user-centric broadcasting experience. Being aware of the unique advantages of the Cloud or distributed networks for machine learning, the service provider expects that the Cloud-based and distributed networks based intelligent user-centric broadcasting significantly improve the user’s quality of experience compared to the central computer-based approaches; hence further improves the business of the service provider.

In user-centric broadcasting, the service provider wants to provide user aware broadcasting service by taking into account that different users or different groups of users have their own preferences. This is another important motivation for residing the processing and editing of the video streams in the distributed networks close to the users.

The distributed network ecosystem-based solution has its unique advantage regarding saving resources compared to a traditional broadcasting system. When the media processing functionality resides in distributed networks or in the Cloud, the complete media processing functionalities could be provided as an ecosystem and used by various service providers in a shared manner, which will significantly save resources compared to the traditional broadcasting system where each of the service providers needs to has its own media processing functionality.



Figure User-centric broadcasting

* + 1. ***Interactive media services [15]***

Multiple clients can interact with the processing entity in NBMP. The client can upload the media content to the processing entity, which can process and integrate the media content from the server and the multiple clients, and send the processed media content to different clients.

Figure 14 shows an example of the system for multi-client interacting based on processing entity. Original video content is ingested and initially encoded into an initial distributing stream at an origin server. Then the initial stream is delivered to a processing entity within the media delivery network, up to the edge server. The processing entity in NBMP can simultaneously receive the media resources from the media source server and the clients. The processing entity can process and integrate the media content, and then the processed media content can be delivered to the corresponding clients. Different versions of media content need to meet the requirements of different devices.



Figure example: the multi-client interacting through processing entity

The following use cases appear to be feasible in a multi-client interaction scenario.

* + 1. ***Customized media composition [16]***

Service providers may want to translate and service the video for better user experience. For example, there is a service that provides old monochrome video as color video or provides aerial images taken by a drone as map images so that the user can easily recognize them. Before that, many professionals needed to work for these services.

However, with the development of AI technique, this video translation can be done automatically and in real time.

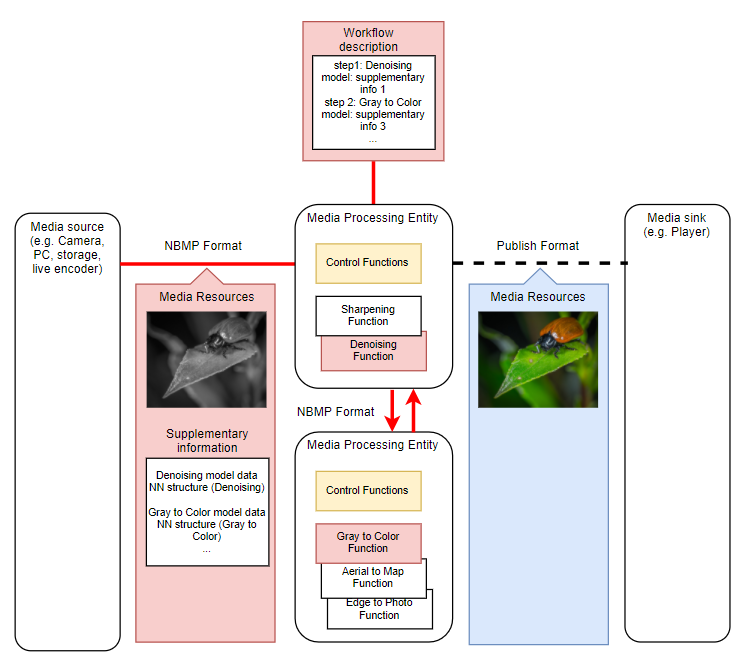


Figure Video translating scenario using NBMP and AI technique when a media source requests a Gray to Color translation

Figure 15shows a example of video translating where a media source requests a Gray to Color translation. First, the media source sends supplementary information along with the black and white image according to the NBMP format. The supplementary information contains Neural Network structure (NN structure) and pre-emulated model data for the media processing. The workflow description contains the desired sequence of processing and configuration information about each processing. In this example, the address of the pre- emulated model, which is used for media processing, is described in workflow description. In the example of figure 1, denoising function and Gray-to-color function use the pre- emulated model and neural network structure (NN structure) which is included in the supplementary information. The raw data of pre-emulatedmodel and NN structure is delivered through supplementary information.

The Media Processing Entity processes the requested operations in the order described in the workflow description. In the example shown in Figure 1, first the Denoising operation is performed, and the Gray to Color translation is performed. The control of this operations is processed by the Control function. The Media Processing Entity uses the pre-emulated model stored in the supplementary information to process the operations. The media processing entity sends the processed image to the media sink in publish format.

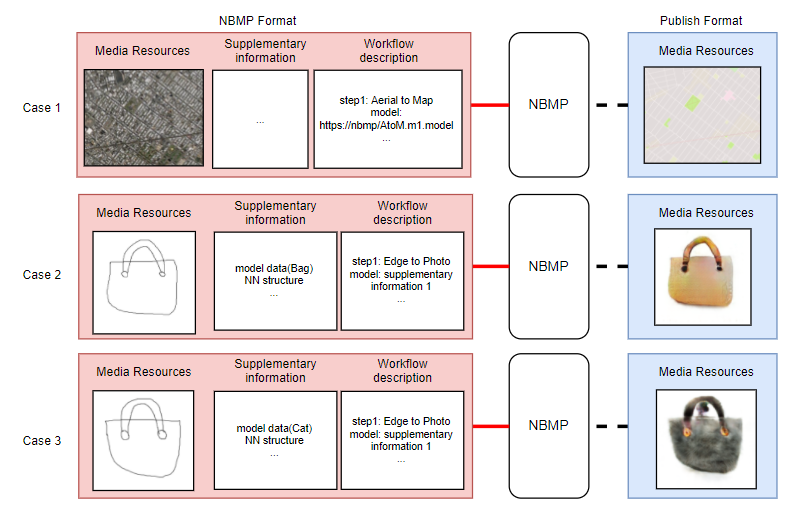


Figure Video translation example according to supplementary information and workflow description in the proposed use case.

Figure 16 shows that different kinds of video translation are performed as the contents of workflow description change. Case 1 shows the case when aerial to map function is selected as media processing function, and case 2 and case 3 show the case when Edge to Photo function is selected. In case 1, the pre-emulated model for processing function is provided by a network. In case 2 and case 3, it is provided by supplementary information. Case 2 and case 3 show why the pre-emulated model is designated in the workflow description. Because, as shown in case 2 and case 3, the results vary greatly depending on the pre-emulated model used even though both processing function is the Edge-to-Photo.

* 1. ***Immersive media handling***
     1. ***Network aggregated point cloud media [17]***

Shortly naturally captured point clouds will become as commonplace as mobile photography taken by mobile devices today.  Technologies similar to mobile captured point clouds (3D scanning for 3D printing) already exist in some state of the art mobile devices, and it can be predicted that there will be many various devices utilizing different technologies to capture and create point cloud media data at different resolutions and qualities.

When different point cloud media data of the same object or scene is captured using different capture devices and technologies, service providers may want to aggregate media data concerning the same object or scene to enhance and improve the data for the object or scene.

Aggregation of the point cloud media data for a certain object or scene in the cloud (e.g., an NBMP server) allows for the various applications:

1. Increasing the number of points for increased point resolution
2. Elimination of noise and redundant points
3. More accurate and diverse texture/reflective properties of the point cloud under different lighting conditions
4. Interactive point cloud media data for objects and scenes which change attributes depending on user interactivity

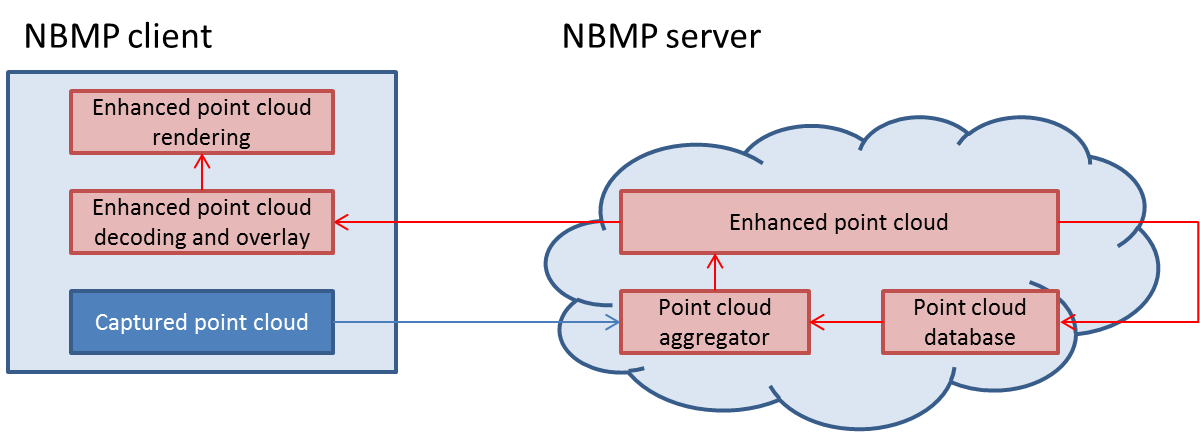


Figure the typical use case for network aggregated point cloud media.

Figure 17: proposed use case where client captured point cloud media data is aggregated by the NBMP server, updating the original point cloud media database with enhanced attributes

1. An NBMP client, such as a mobile device, captures an object or scene as encoded point cloud media data.
2. The media data is then sent to the NBMP server, where the point cloud aggregator decodes and recognizes (possibly using artificial intelligence) the object or scene.
3. The aggregator requests and fetches relevant data for the same or similar objects which are already contained in the point cloud database.
4. The aggregator then merges captured data and original data from the database to form an enhanced point cloud (the properties of which are the result of the various possible applications listed above).
5. The enhanced point cloud is then sent to the NBMP client, which decodes, overlays and renders the point cloud media according to its specific application, such as AR or MR.
   * 1. ***Network Pre-rendering [18]***

Full 6DoF immersive content allows the viewer to navigate a selected location within a 3D content space. The 6DoF content provides dynamic and immersive experiences to the users, such as views of the greatest plays from every angle or pivotal play from the viewpoint of a user’s favorite player. Seeing key plays up close from new perspectives will redefine what it means to watch a sports game.

One way to capture such content is through Intel’s freeD solution which captures 360 videos by using 38 5K ultrahigh-definition cameras resulting in data sizes of ~ one terabyte for a 15- to a 30-second clip. In this solution, volumetric content is pre-rendered by the high-performance computers and sent as 2D video to the users. MPEG-I would enable delivery of volumetric video to the users directly so that each user can select a view-point dynamically. When a live sports game, such as Super Bowl, is offered as 6DoF immersive content, a user may, for instance, watch the game from the perspective of his favorite player. In some cases such as a mobile terminal, it would be difficult to receive and render a complete 6DoF volumetric video content due to the lack of processing power, battery power consumption limits, and network conditions. Therefore, 6DoF content would be sent to a mobile terminal after a remote pre-rendering step on the network depending on the user’s selected viewport, the network conditions, and/or the device capabilities.

Pre-rendering of 6DoF scenes to match users’ capabilities and network conditions is a key use case for NBMP. With the proliferation of GPU clouds, this task can be performed with high efficiency in the network and help save resources at the end-device and in the access network.

In addition to flattening the scene to a 2D video, the network may offer other options instead. It can for instance produce a 360-degree video from a selected location in the scene, or it may just simplify the 6DoF scene into a less complex 6DoF scene that is sent to the receiver.

In summary, the 6DoF pre-rendering use case will:

1. allow receivers with low processing power or GPU capabilities to consume the 6DoF content
2. reduce the bandwidth needs of the access network
3. enable content providers to reach a wide variety of receiver devices

The realization of this use case in NBMP can happen through deploying a media processing entity at the network with the following capabilities:

1. High-performance GPU(s)
2. Proximity to the end receiver
3. Sufficient bandwidth to receive the 6DoF content
4. A 6DoF pre-rendering task that supports the input format

The NBMP media processing entity may run a set of media processing tasks to perform the 6DoF pre-rendering. The first task will receive and compose the scene graph and the referenced media resources promptly. The scene graph is used both by the control function and the pre-renderer function. The control function also receives feedback from the receiver both as static information (such as device capabilities) and as dynamic information (such as user and head movement and available bandwidth measurements). This information is used to instruct the pre-renderer to produce the required views in the requested output format.

Figure 18 depicts this NBMP processing pipeline.



Figure 18 Remote pre-rendering in NBMP

* 1. ***Additional use cases***
     1. ***Media processing for Vehicles [19]***

On the Internet of Vehicles scenario, the existing technology is the car to get around the traffic information and some of the surrounding vehicles simple information, in the local information processing, to guide the safety of the car driving. However, this information is often limited, single car's processing capacity is limited. On the Internet of Vehicles, many vehicles can be connected by Wi-Fi, LTE-U or other ways for free information exchange. A processing unit B may be added to these devices to improve the processing capability of vehicle status information, traffic information, etc., to obtain more reliable and effective results to guarantee safe driving.

As shown in Figure 19, on the Internet of Vehicles, the unit B (monitoring & control infrastructure) may be established as a roadside unit, and the vehicle in the area covered by the roadside unit transmits its own position, speed, acceleration, steering information and so on. At the same time，the roadside unit side itself can also collect traffic information through the sensor, such as the front may be pedestrians crossing the road. Roadside units also have stored some local information, such as the map information. A number of adjacent roadside units can communicate with each other and exchange traffic information. Also, the upper side of the roadside unit may have an upper layer information processing and control unit, and the roadside unit may upload information to the upper unit, and the upper unit may also send information to the roadside unit. Road-side-unit(RSU) can process information from all above (vehicles, adjacent roadside units, upper layer unit, local storage, and calculate the control information used to guide each car's safe driving, such as to remind the driver in front of the car may break, slow down, or broadcast traffic information to all cars.

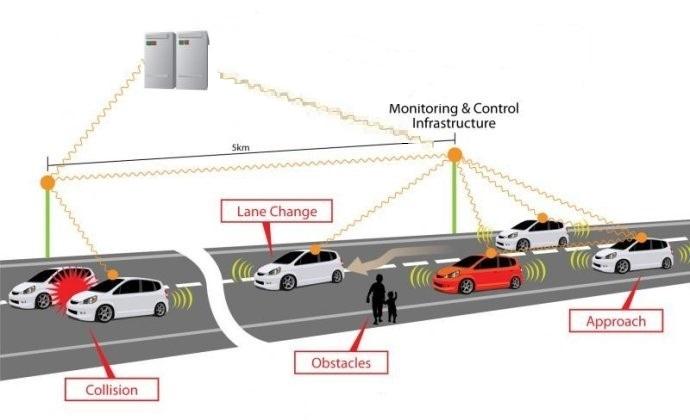


Figure NBMP as monitoring & control infrastructure on Internet of Vehicles

1. **Requirements for network-based media processing**
   1. ***General requirements (G)***
   2. NBMP shall support NBMP format.
   3. NBMP shall provide the APIs to support the network-based media processing.
   4. NBMP shall have advantages in terms of compression, computation, bandwidth, or storage efficiency over existing usage of MPEG standards. (G)
   5. NBMP shall support media processing distributed across processing units. (G)
   6. NBMP shall support media processing with processing units that are interconnected with each other in a network. (G)
   7. NBMP shall provide means for network-based media processing without affecting the compliance with existing decoding standards in the end-device. (G)
   8. NBMP shall support any type of media content (be media agnostic), including the existing MPEG codecs (which may produce very low to very high data rates) and MPEG formats such as ISO/IEC 13818-1, ISO/IEC 14496-12, ISO/IEC 23008-1 and ISO/IEC 23009-1. (G)
   9. NBMP shall support delivery for applications that use common components (e.g., ISOBMFF). (G)
   10. NBMP shall support delivery over an IP-based connection. (G)
   11. NBMP shall support delivery over HTTP and RTP. (G)
   12. NBMP shall support delivery over UDP and TCP. (G)
   13. NBMP shall support streaming. (W)
   14. NBMP shall support file delivery. (W)
   15. NBMP shall support progressive download.(W)
   16. NBMP shall support delivery using existing MPEG protocols (e.g. ISO/IEC 23008-1). (W)
   17. NBMP shall support delivery of media using existing standardized delivery formats. (W)
   18. NBMP shall support push-based streaming, e.g., over unidirectional or multicast channels. (W)
   19. NBMP shall support push-based progressive download, e.g., over unidirectional or multicast channels. (W)
   20. NBMP shall support hybrid delivery environments, such as multiple transmission channels, possibly of different types. (W)
   21. NBMP shall support multipath delivery. (W)
   22. NBMP shall support both unidirectional and bidirectional communication environments. (W)
   23. NBMP shall support seamless use of heterogeneous network environments including broadcast, unicast, multicast, storage, peer-to-peer, and mobile.(W)
   24. NBMP should support distribution of media processing workload among multiple physically-distributed processing units (G)
   25. NBMP should tolerate failures of distributed media processing units (G)
   26. NBMP shall support the compliance to the existing standards without affecting, e.g., container format and decode in end devices etc. (G)
   27. NBMP shall support customize broadcasted streams in a user-centric way without affecting the compliance with the existing standards, e.g., container format and decoding in end devices etc. (G)
   28. NBMP deployments shall integrate seamlessly with existing CDN deployments and enable the optimized serving of cached content (G)
   29. ***Requirement for media resource (M)***
   30. NBMP shall support the identification of conformance points of each content component (e.g., elementary stream). (M)
   31. NBMP shall support to upload locally generated multimedia content to the processing unit in the network (M)
   32. NBMP shall support storage of multiple components for a single application. (M)
   33. NBMP shall support storage of content that uses common components. (M)
   34. NBMP shall support to create time and space overlapping sub-video streams from the main video stream. (M)
   35. NBMP shall support pre-transcoded video content for efficient media caching in the network (M)
   36. NBMP shall support media aware cache by media processing of cached NBMP data. (M)
   37. NBMP should support media processing to ingest the media resource to the media processing entity.
   38. NBMP should support media processing of non-video contents, such as cloud gaming, point-clouds, 3D graphics, and multiple sensory data. (M)
   39. NBMP shall support the media processing of video streams with overlay graphics (W)
   40. ***Requirement for supplementary information (S)***
   41. NBMP shall support content component identification. (S)
   42. NBMP shall support clock recovery (e.g., PCR) (S)
   43. NBMP shall enable content synchronization at capture (from different sources) and playback (on different screens) (S)
   44. NBMP encoder shall provide means to signal NBMP metadata (e.g., side information). (S)
   45. NBMP shall support in-band or out-of-band carriage of format information such as bit-rate, resolution, codec type, frame rate, dynamic range, color space, sub-sampling type, segment duration. (S)
   46. NBMP shall support to collect user preferences to configure the media creation procedures (S)
   47. NBMP shall support the metadata for targeting cache (S)
   48. NBMP shall support interfaces to receive congestion notification from other network elements and enable appropriate congestion control (S)
   49. NBMP should support functionalities for adaptive media processing based on awareness of context, service, application, user’s history, location, client device, and processing resources. (S)
   50. NBMP should support exchange of messages among media processing entity (S)
   51. NBMP should support signaling messages of media service for optimal use of network resources in distributed network entities. (S)
   52. NBMP should support generating metadata as opposed to only the processed media (S)
   53. NBMP should support signaling of information such as camera lens calibration, camera rig information and orientation and geographic location data to the media processing entity. (S)
   54. NBMP should support signaling of metadata of multiple capture devices such as the camera pose and the relationship information of capture device to the media processing entity. (S)
   55. NBMP should support the metadata and auxiliary information for dynamic configuration of media delivery.
   56. NBMP should support signaling of media distribution information such as identification of streams, required bitrate and URI assignment to the media processing entity. (S)
   57. NBMP should support the feedback information based on the user preference and device capability to a media processing entity (S)
   58. ***Requirement for workflow description and API (W)***
   59. NBMP shall support transcoding for adaptive streaming in the network. (W)
   60. NBMP shall support a format for requesting media processing of time segments of media data for adaptive bit-rate streaming. (W)
   61. NBMP should support dynamic configuration (e.g., merging) of content components during delivery. (W)
   62. NBMP shall support low latency delivery (e.g., to support conversational applications, live content, etc.). (W)
   63. NBMP shall support use of different QoS types and levels.(W)
   64. NBMP shall support low-complexity format conversion for delivery and storage. (W)
   65. NBMP shall support relaying received content stored on storage devices.(W)
   66. NBMP shall support control of content relay and retransmission.(W)
   67. NBMP shall support dynamic media processing during delivery.(W)
   68. NBMP should support signaling, delivery, and utilization of content using multiple protection and rights management tools.(W)
   69. NBMP should support content encryption using encryption methods (W)
   70. NBMP should support seamless change between content rights management schemes.(W)
   71. NBMP should support load balancing among media processing units (W)
   72. NBMP should support adaptive scheduling mechanisms for dynamic scheduling of media processing tasks on distributed units (W)
   73. NBMP should support migration of the media processing tasks among distribution units during the processing (W)
   74. NBMP should support a logically centralized unit to manage the whole media processing distribution environment (W)
   75. NBMP should support real-time metrics for available resources on distributed media processing units, such as number of clusters, available memory, available processors, etc. (W)
   76. NBMP should support metadata and/or API accounting for total queuing delay until the start of media processing (W)
   77. NBMP should support media processing that is either batch or event-driven (W)
   78. NBMP should support media processing that is either delay-sensitive or delays tolerant (W)
   79. NBMP shall support to configure and execute processing of VR content to change encodings and frame packing variants (W)
   80. NBMP shall provide the APIs to enable the configuration and execution of VR processing in distributed networks (W)
   81. NBMP shall support to collect network traces and user consumption logs (S) and to apply machine learning techniques to determine or fine tune the VR processing procedures (W)
   82. NBMP shall provide the APIs to allow content providers to configure and execute graphics overlaying on their own video content securely. (W)
   83. NBMP should support video analysis techniques on the video stream, e.g., to perform object recognition and tracking (W)
   84. NBMP shall provide the APIs to configure and run up-scaling in the cloud (W)
   85. NBMP shall provide the APIs to collect feedback and analyze user preferences (W)
   86. NBMP shall provide the APIs to enable the configuration and execution of customization processing of the broadcasted streams. (W)
   87. NBMP shall provide the APIs for the collection and analyze user feedback and preferences, e.g., by measurement of consumption. (W)
   88. NBMP shall support distribution of pre-transcoding of video content (W)
   89. NBMP should support targeting caching which related to user preference. (W)
   90. NBMP should support large-scale caching for considering large-scale media distribution. (W)
   91. NBMP should be able to make use of mobile’s capability to use multiple Radio Access Technologies and shall support multi-path delivery (W)
   92. NBMP shall support advanced QoE reporting to enable optimized rate adaptation (W)
   93. NBMP shall support dynamic content offloading to different channels and to broadcast channels (W)
   94. NBMP shall support low-latency streaming with high bitrate allocation – even higher for low-latency compressed or raw media data streaming  (W)
   95. NBMP shall support low-latency real-time stitching fully synchronized media data delivery for real-time stitching (to reduce buffering requirements) (W)
   96. NBMP should support functionalities for filtering, modifying and mixing of media content. (W)
   97. NBMP should support signaling of video quality metrics for media processing (W)
   98. NBMP should support adaptive QoS-aware media processing to differentiate delay-sensitive and less delay-sensitive services (W)
   99. NBMP should provide task completion metadata and/or accounting API (usage and duration). (W)
   100. NBMP should support the APIs for dynamic configuration of media delivery (e.g., unicast to multicast).
   101. NBMP should support the description for multicast streaming with ABR to enable network assisted media distributions.
   102. NBMP should support pre-rendering function in the media processing entity. (W)
   103. NBMP should support configuration information about the required processing resources (e.g., required GPU acceleration).
   104. NBMP should provide the model of workflow for providing information about a certain functional behavior of media processing pipelines.
2. **References**

[1] m40680, NDVC use case on VR processing in the Cloud, Hobart MPEG

[2] m40681, NDVC use case on intelligent video editing, Hobart MPEG

[3] m41741, Distributed Video Coding for Network Based Media Processing Architecture, Macau MPEG

[4] m42108, [NBMP] Additional Use-case for NBMP, Gwangju MPEG

[5] m42091, [NBMP]Supplementary user case for 360 VR stitching, Gwnagju MPEG

[6] m42093, [NBMP]Proposed new use case: Cloud-based multiple camera registration, Gwangju MPEG

[7] m41889, [NBMP] live media ingest specification, Macau MPEG

[8] m40443, Additional use case and requirement for network distributed video coding: Cache, Hobart MPEG

[9] m40443, Additional use case and requirement for network distributed video coding: Cache, Hobart MPEG

[10] m40443, Additional use case and requirement for network distributed video coding: Cache, Hobart MPEG

[11] m40443, Additional use case and requirement for network distributed video coding: Cache, Hobart MPEG

[12] m42058, [NBMP] M-ABR Use Case and Requirements for NBMP, Gwangju MPEG

[13] m40682, NDVC use case on video and AR, Hobart MPEG

[14] m40681, NDVC use case on intelligent video editing, Hobart MPEG

[15] m40442, Additional use case and requirement for network distributed video coding: multi-client interacting, Hobart MPEG

[16] m42206, [NBMP] Use case for NBMP: video translation using Deep Learning, Gwnagju MPEG

[17] m41812, Additional use-cases for Network-Based Media Processing (NBMP), Macau MPEG

[18] m42134, Usecase for NBMP, Gwangju MPEG

[19] m40442, Additional use case and requirement for network distributed video coding: multi-client interacting, Hobart MPEG