Solution Implementation Summary

Media
360-Degree Video Distribution

intel.

Intel Advanced 360 Video

Intel Corporation
Visual Infrastructure Division

William Cheung
Platform Solution Architect

Gang Shen
Software Architect

Dusty Robbins
Segment Manager

Introduction

This solution implementation document presents how an end-to-end immersive video implementation can be constructed for low-latency content distribution. In contrast to traditional cloud broadcast implementation, the reference implementation can reduce the end-to-end broadcast delay significantly via tight integration of immersive video stitching and encoding at ingestion points in the network and application of Open WebRTC in the delivery stage. Being 5G MEC ready, the implementation also showcases how the new generation of wireless and edge technologies can enhance the overall immersive experience for end users.

The primary audiences for this document are architects and engineers planning to implement their own immersive video solution. Readers should use this document as a demonstration of how low-latency immersive video can be created and distributed over 5G edge technology.

It is important to note that the details contained herein are a reference implementation example for immersive video streaming. Intel does not aim to promote or recommend any specific hardware, software, or supplier mentioned in this document. In addition, Intel does not aim to tie customers to any specific software and hardware stack.

Solution Overview

Intel Corporation's immersive video reference solution was developed to demonstrate that immersive video content, in high visual fidelity, can be streamed with high network efficiency and with glass-to-glass latency under two seconds. In contrast to traditional broadcasting solutions that rely on cloud and content delivery network (CDN) infrastructure for content processing and delivery, this solution is designed to use the latest edge-based technologies, such as low-latency 5G RAN and Intel® Xeon® processors with local edge compute and Kubernetes container-based application orchestration in a private 5G network.

The solution is implemented on a complement of Intel software and hardware technologies that are available for commercial trials and service deployments. Software components include industry innovations such as Intel Open WebRTC Toolkit (OWT), Scalable Video Technology (SVT), OpenNESS, and Open Visual Cloud 360 (OVC360).

The implementation will support third party production and client software while maintaining standards compliance with both 3GPP and MPEG-I.

Figure 1 describes the traditional approach currently used to capture, process, and deliver 360 immersive video experiences. This solution incorporates on-premises servers for processing the captured content and cloud infrastructure for transcoding, tiling, and packaging the video content for distribution. Content is distributed through a supporting CDN, such as Akamai, to a variety of targeted client devices. For live, high-resolution (i.e. 8K) 360 video, typical end-to-end latencies are about 15-20 seconds, which is sufficient for global distribution of live events using this type of content.

Table of Contents

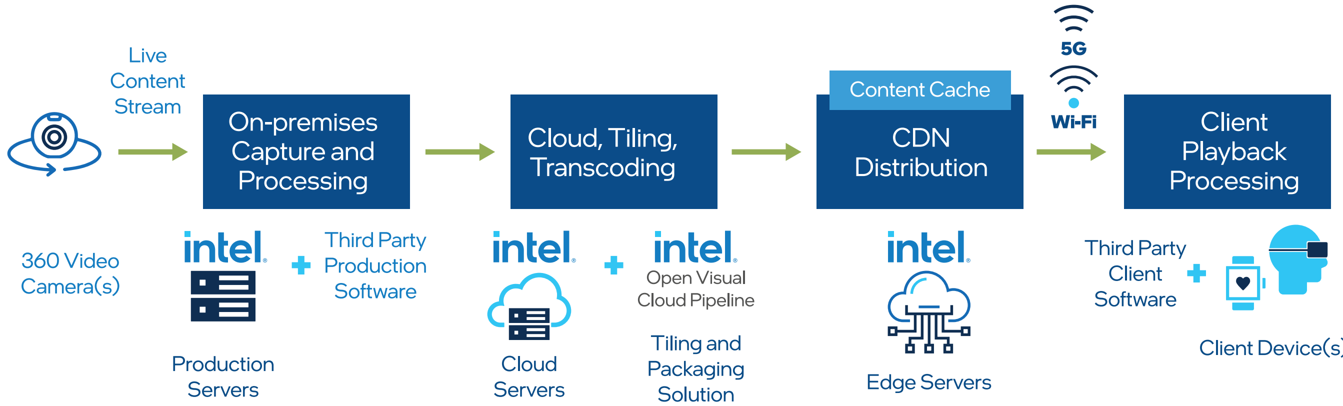

Figure 1. Traditional live immersive video end-end pipeline (non-broadcast configuration).

However, interactive and on-location live content consumption usages would demand much lower end-to-end latency. By utilizing immersive video processing and delivery software that are highly optimized for Intel® Server platforms on the 5G edge infrastructure, delays in various stages of the pipeline can be greatly reduced, thereby meeting the ultra-low-latency requirement of these usages and other future enhancements.

Figure 2 compares the immersive video workload architecture using a traditional cloud and CDN mechanism with one that utilizes 5G edge.

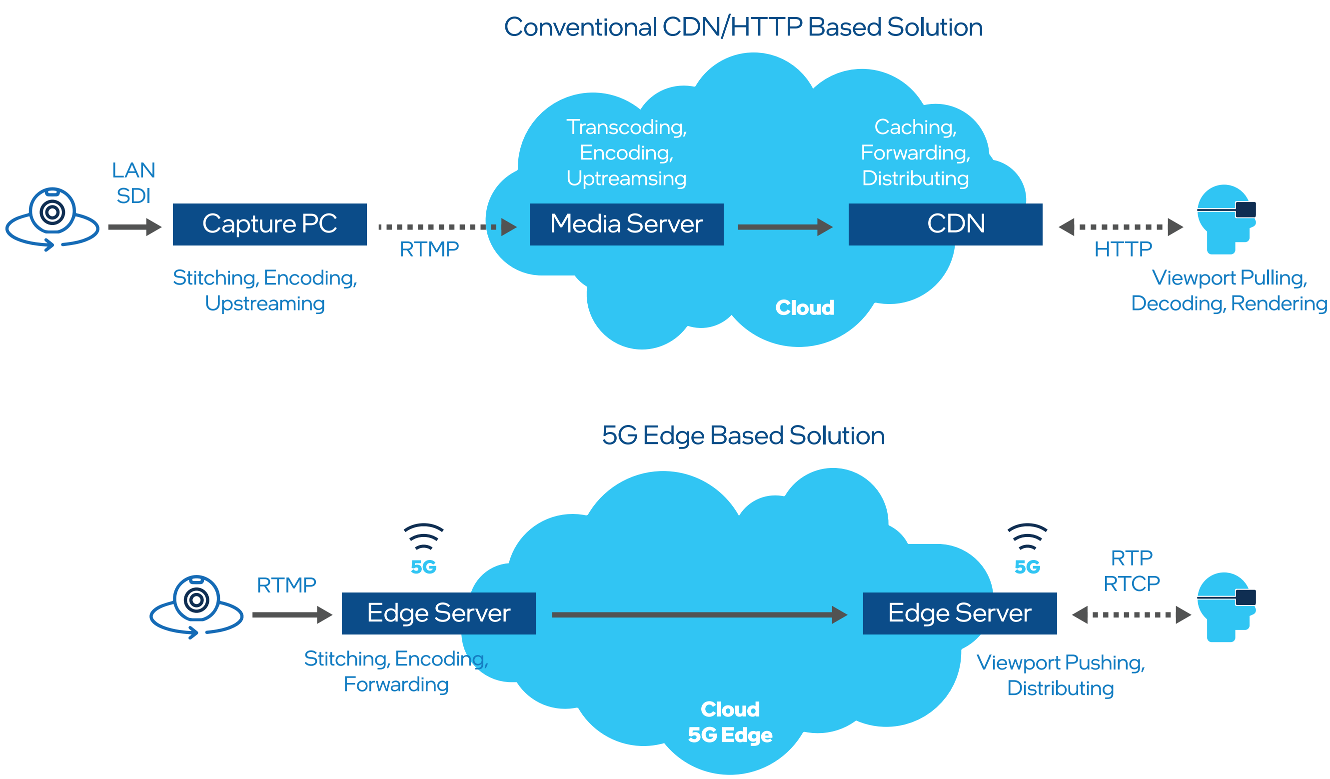

Figure 2. Public cloud and CDN architecture compared to 5G edge implementation.

It should be noted that while we are highlighting a 5G edge implementation, the conventional CDN solution can also be placed on the network edge or telecommunications service provider's 5G edge. When implemented in this fashion, it is possible for the CDN approach to achieve a greatly improved motion-to-high-quality (MTHQ) latency.

It should be also noted that edge server (or server clusters) at ingestion points can provide remote production capabilities, with extensible computational resources that are unmatchable for Capture PC and Media Server in conventional CDN/HTTP based solutions.

Table 1 below summarizes the implementation's hardware specifications. Media functions are intended to be deployed in edge platforms, such as OpenNESS (or other Kubernetes-based edge platforms). Options for this implementation can support different network layouts (e.g., on-premises, public) and inclusion of more powerful platform technologies for media distribution should support for more users be desired.

Table 1. Hardware Specifications

Server	Specification	Function
2× Intel® Server	• Intel® Server Board S2600STBR • Dual Intel® Xeon® Platinum 8280 Processor, 2.70 GHz, 38.5 MB cache, 28 cores, 205W • Memory: 8 × 32 GB (total 256 GB) 2933 MHz DDR4 • 2 × 10 GbE ports via Intel® Ethernet Controller X557-AT2 • Storage: 240 GB SSD, SATA 6Gb/s Intel D3-S4510 Series	Media Ingest Function (MIF)
1× Intel® Server	• Intel® NUC NUC8i7HVK Mini PC • Intel® CoreTM Processor i7-8809, 3.10 GHz, total 4 cores • Memory: 2 × 16 GB (total 32 GB) 2400 MHz DDR4 • 2× 1GbE ports via Intel® Ethernet Connection i219-LM and i210-AT • Storage: 1024 MB SSD M2 SSD	Media Control Function (MCF)
1× Intel® Server	• Intel® NUC NUC8i7HVK Mini PC • Intel® CoreTM Processor i7-8809, 3.10 GHz, total 4 cores • Memory: 2 × 16 GB (total 32 GB) 2400 MHz DDR4 • 2 × 1GbE ports via Intel® Ethernet Connection i219-LM and i210-AT • Storage: 1024 MB SSD M2 SSD	Media Distribution Function (MDF)
1× Intel® NUC	• Intel® NUC NUC6i7KYK Mini PC • Intel® CoreTM Processor i7-6770HQ, 2.60 GHz, total 4 cores • Memory: 2 × 16 GB (total 32 GB) 2133 MHz DDR3 • 1 × 1GbE ports via Intel® Ethernet Connection I219-LM • Storage: 1TB SSD M2 SSD 6 Gb/s	Linux Client
2 x Cameras	• Kandao Obsidian R • Insta360 Pro II	Video Input
2 x Phones	• Huawei Mate30 • Samsung S10	Video Consumption

In the reference solution, we utilize several software elements that provide the essential ingredients for end-to-end immersive video. See Figure 3 for an overview of the software architecture used.

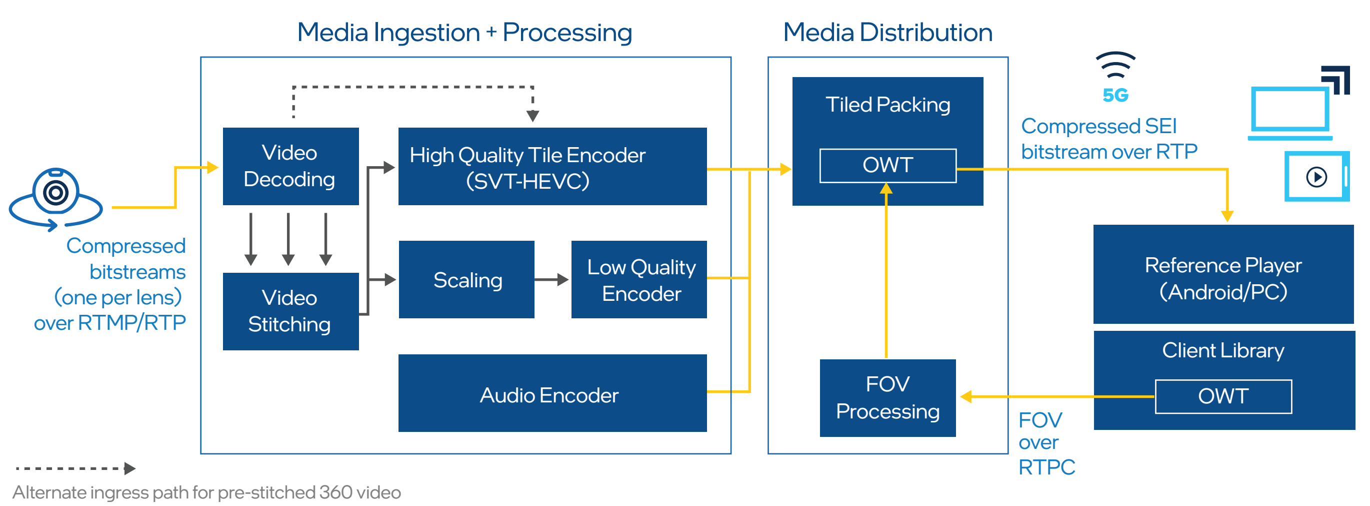

Figure 3. Software architecture for low latency immersive video.

For media ingestion and processing, the solution relies on Intel Scalable Video Technology (SVT) for efficient transcode. In addition, an Intel-developed immersive video solution is utilized for video stitching and tiling. The Intel Open WebRTC Toolkit (OWT) is utilized for connecting and distributing the forwarded video streams. All components are encapsulated and orchestrated by OWT framework. For consumption, a reference player is used on the endpoints. Media ingestion, processing, and distribution are served through functions, or microservices, on the network. Cameras and cellphones are considered user equipment and will connect to those microservices when needed.

To minimize the burden on the underlying network, tiling of the processed video is used to minimize the amount of information being streamed to the client at any moment. With this approach, content users are only provided with their field of view; end device movements are tracked and utilized to keep the field of view (FOV) content current. Tiling in the Motion Constraint Tile Sets (MCTS) utilizes Intel SVT-HEVC, making tiles independently decodable from other tiles and key for supporting viewport-dependent streaming. The Intel 360SCVP library provides OMAF-compliant video packing format for the FOV content in 360-degree video.

The solution includes three major functional software modules running on docker images:

- MIF (Media Ingestion Function) handles acceptance and ingestions of different types of cameras, as well as media processing functions such as stitching and transcoding. It is intended to be placed on the edge server and close to media capture devices (cameras), so that immediate error corrections and stable network connections can be available.
- MDF (Media Distribution Function) provides viewport-based distribution and delivery, which repacketizes and transmits media content according to each individual client's viewport. The client devices will send viewport information constantly to MDF. The exchange of viewport and FOV content is via RTP/RTCP over UDP.
- MCF (Media Control Function) handles sessions management and signaling between MIF, MDF, and UEs (including cameras and cellphones). It facilitates the establishment of the media transport sessions.

As noted earlier, these services can be deployed on edge platforms such as OpenNESS or any 5G MEC edges.

The table below highlights the primary software components used in the implementation:

Table 3. Software components.

Function	Product
Operating system (Servers)	CentOS 7.6.1810 and Ubuntu 18.04.2
Open WebRTC Toolkit (OWT)	https://github.com/open-webrtc-toolkit/owt-server v4.3.1
Scalable Video Technology (SVT)	https://github.com/OpenVisualCloud/SVT-HEVC v1.5.0
Intel Open Visual Cloud – 360SCVP Library	https://github.com/OpenVisualCloud/Immersive-Video-Sample.git Commit ID: 9ce286edf4d5976802bf488b4dd90a16ecc28c36
Libxcam	https://github.com/intel/libxcam Commit ID: dd4874ab1df0d5d6d193dc2116372d4ba916f8b3
OpenNESS	OpenNESS 20.03

The implementation utilizes both CentOS v7.6.1810 and Ubuntu v18.04 on the servers. OpenNESS (v2020.03) is used on the edge platform.

The major software libraries used include:

1. Intel Open WebRTC Toolkit (OWT). A versatile WebRTC server toolkit optimized on Intel® Architecture.
2. Intel SVT-HEVC. A parallel and scalable HEVC encoder optimized in Xeon processors.
3. Intel Libxcam. An open-source library of extended camera features, image processing and analysis optimized on IA.
4. Intel Open Visual Cloud – 360SCVP Library. A MPEG-I OMAF compliant 360-degree video format library.

Test Results

To understand the performance of the video pipeline, it is important to set up a complete solution from end to end with an appropriate network layout. The test results collected are based on an on-premises network layout, though it is possible to deploy the solution on either on-premises or public networks with the correspondent configurations.

Figure 4 shows the hardware on-premises test setup. The setup relies on an Intel Tofino Ethernet Switch, which specifies different network segments for ingestion and distribution with VLANs. For the ingestion network (10.10.10.X), two cameras are connected to two MIFs respectively. For the distribution network (30.30.30.X), four cellphones and two Linux client players are connected to one MDF. The MCF is in the network segment (20.20.20.X), which connects both ingestion and distribution networks and functions (MIF and MDF). All functions are running in dockers as microservices.

Wi-Fi, 5G, and cables were used as connections between MDF and the cellphones, where cables and 5G were used as connections between cameras and MIF.

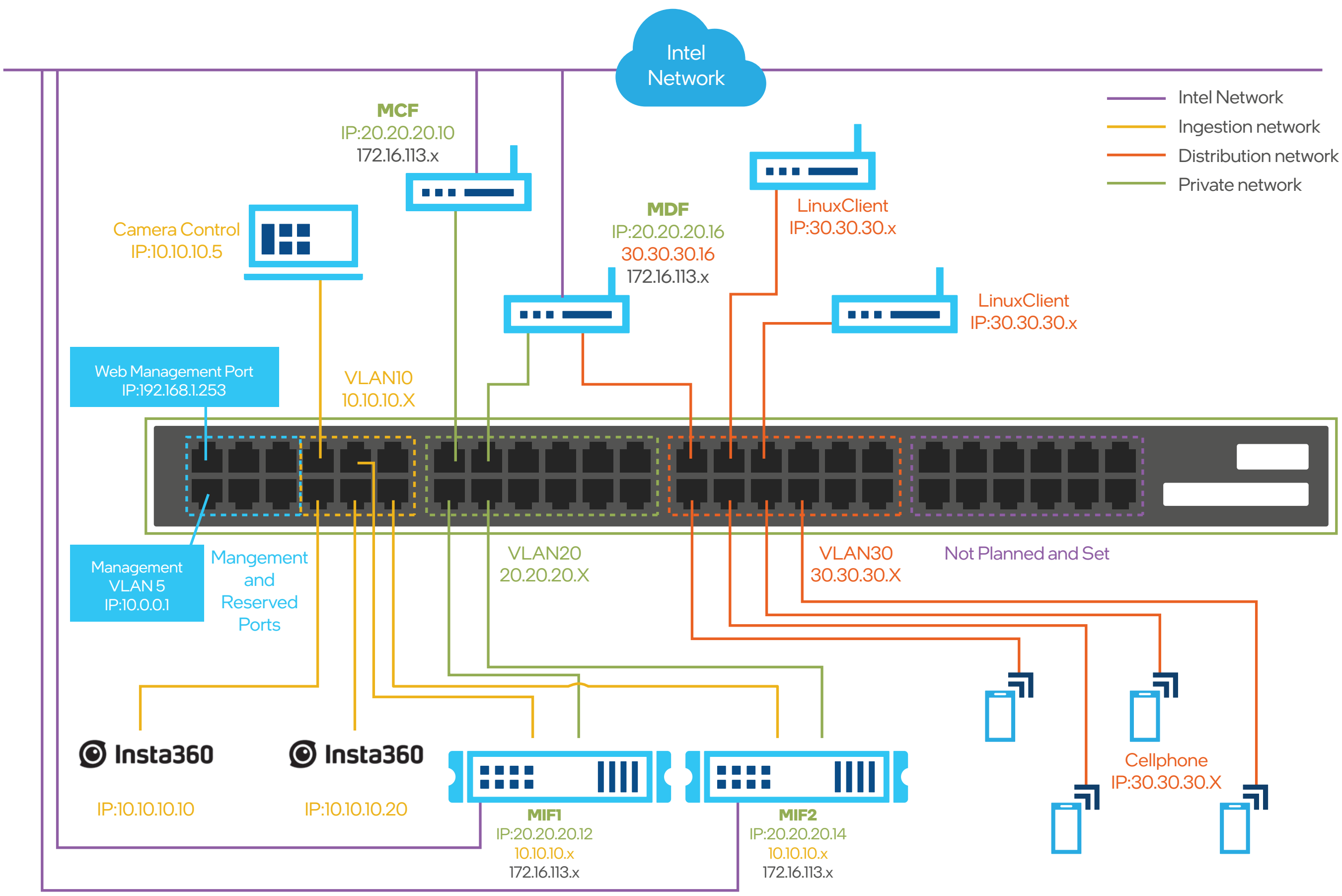

Figure 4. Hardware Setup

Scenarios

The testing scenario simulates an event, such as a concert or sporting activity, that is being captured by multiple high definition cameras and input devices, ingested into the network/cloud, processed in real time, and distributed to multiple end users for an immersive viewing experience.

Results

The following results were obtained in a December 2020 test, in field of view mode, and obtained through a Type-C to Ethernet adapter:

End to End Latency (ms)

		End-to-end Latency Breakdowns			
System Setup	Video Source	**8K File**	**8K File**	**Insta360**	**Insta360**
	Cell Phones	**Mate30**	**S10**	**Mate30**	**S10**
Media Ingestion Function (MIF)	LiveStreamIn	516.215	516.394	541.299	503.128
	FFMpeg Decoder	500.124	499.669	568.032	568.068
	360 Stitching	27.01	26.828	28.455	28.558
	SVT HEVC Encoder	323.306	323.735	420.493	406.324
Media Distribution Function (MDF)	VideoPacketizer	2.013	2.095	1.87	1.992
Server total latency (Sub Total)	Server (MIF+MDF)	1368.668	1368.721	1560.149	1508.07
Android Client	RTP Packet	40	47	35	79
	Decoder	85	34	72	34
	Renderer	13	14	13	14
Client total latency	Client	138	95	108	127
E2E total latency (Total)	CAM+ Server + Client	N/A	N/A	1745ms ~ 1856ms	1745ms ~ 1856ms

MTHQ (ms)

		MTHQ Latency Breakdowns			
System Setup	Video Source	**8K File**	**8K File**	**Insta360**	**Insta360**
	Cell Phones	**Mate30**	**S10**	**Mate30**	**S10**
Media Ingestion Function (MIF)	Tile selection/GOP5	74.81	88.63	89.67	71.33
Android Client	RTP Packet	40	47	35	79
	Decoder	85	34	72	34
	Renderer	13	14	13	14
MTHQ	Tile_selection + Client	212.81	183.63	197.67	198.33

The test results show the solution can achieve 1.75s~1.86s end-to-end (or glass-to-glass) latency, and a MTHQ latency of approximately 200ms. The GOP (group of picture) size may affect the MTHQ latency; GOP size 5 is used in testing.

In addition, the CPU utilization of MDF (which is a NUC in testing) is as low as 3.1% for one client. The outbound bandwidth of MDF occupied by one client is about 11 Mbps. The total number of supported clients is bounded by CPU capability and total network bandwidth. It can be extrapolated from the measurements that a MDF (a NUC in the above configuration) can support about 30 clients simultaneously.

These results show the implementation can deliver real-time, 8K 360-degree video for use in live broadcasting. This would be suitable for scenarios such as sporting events and concerts.

Next Steps

To learn more about the technologies mentioned in this paper, please visit the following links:

- To learn more about Intel's Visual Cloud technologies, visit https://www.intel.com/content/www/us/en/cloud-computing/visual-cloud.html.
- To learn more about Open WebRTC Toolkit, visit https://github.com/open-webrtc-toolkit.
- To learn more about OpenNESS, visit https://www.openness.org/.
- To learn more about Scalable Video Technology, visit https://01.org/svt.
- To learn more about reference implementations for Visual Cloud usages, such as Immersive Video, visit https://01.org/openvisualcloud.

References

Name	Reference
Heavy Reading: Producing Live 8K, 360-Degree Streaming Media Events	https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/hr-producing-live-8k-360-degree-streaming-media-events.pdf
Business Brief: Lightning-Fast Video Retrieval Delivers Better 8K VR Experiences	https://www.intel.com/content/dam/www/public/us/en/documents/case-studies/tiledmedia-360vr-business-brief.pdf
MPEG-I OMAF	https://mpeg.chiariglione.org/standards/mpeg-i/omnidirectional-media-format
MPEG HEVC MCTS	https://mpeg.chiariglione.org/standards/mpeg-h/high-efficiency-video-coding/n16499-working-draft-1-motion-constrained-tile-sets
ETSI GS MEC 002 V2.1.1	https://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/02.01.01_60/gs_MEC002v020101p.pdf
W3C WebRTC	https://www.w3.org/TR/webrtc/
Intel Open WebRTC Toolkit (OWT)	https://01.org/open-webrtc-toolkit, https://github.com/open-webrtc-toolkit
Intel SVT-HEVC	https://01.org/svt/overview, https://github.com/OpenVisualCloud/SVT-HEVC
Intel Visual Cloud – 360SCVP Library	https://github.com/OpenVisualCloud/Immersive-Video-Sample/blob/master/src/doc/Immersive_Video_Delivery_360SCVP.md
Intel LibXCam	https://github.com/intel/libxcam

Acronyms and Abbreviations

Abbreviation	Description
CAM	Camera
E2E	End-to-end
FOV	Field of View
GOP	Group of Picture
MCF	Media Control Function
MCTS	Motion Constraint Tile Sets
MDF	Media Distribution Function
MEC	Multi-access Edge Computing
MIF	Media Ingestion Function
MTHQ	Motion-To-High-Quality
NUC	Next Unit of Computing
OMAF	Omnidirectional Media Format
OWT	Intel Open WebRTC Toolkit
OVC	Intel Open Visual Cloud
RTC	Real-time Communication
RTCP	RTP Control Protocol
RTP	Real-time Protocol
UDP	User Datagram Protocol
VLAN	Virtual LAN

0621/MH/PDF/347196-001US