**Review Tracker**

| **Reviewer** | **Due** | **Status** | **Contact** |
| --- | --- | --- | --- |
| Initial draft shared with Khronos | 8/24 | * ~~Y~~ | Rachel Bradshaw or Devon Dean |
| Khronos marketing initial review | 8/30 |  | Rachel Bradshaw |
| Khronos initial review | 9/6 |  | Tim Lewis |
| glTF WG Chair sign off | 9/11 |  | Alexey Medvedev |
| Presenter sign-offs | 9/15 |  | Thomas Stockhammer and Imed Bouazizi |
| Neil review and sign off | 7/21 |  | Neil Trevett |
| Marketing final checks | 7/23 |  | Caster Team |
| Post private draft to website | 7/25 |  | James Riordon |
| Make public on website and share | 7/26 |  | James Riordon |
| Share with press | 7/26 |  | Caster Team |

Link: <https://www.khronos.org/blog/mpeg-extends-and-integrates-gltf-2.0-into-mpeg-i-scene-description-iso-iec-23090>

Short Link: https://khr.io/16k

Placeholder for hero image to blog


**Title: MPEG Extends and Integrates glTF 2.0 into MPEG-I Scene Description ISO/IEC 23090**

**Author:** *Thomas Stockhammer, Senior Director of Technical Standards, Qualcomm, Khronos Member, MPEG and 3GPP Working Group leader, and Metaverse Standards Forum Board of Directors Member and Dr. Imed Bouazizi, Director of Technical Standards, Qualcomm, Khronos Member, 3GPP and MPEG Working Group Contributor, and Metaverse Standards Forum Member*

Standards development is a highly collaborative undertaking within and across standards development organizations (SDOs). We are active in both 3GPP and MPEG working groups, and our work on mobile applications for 3GPP inspired us to tackle an emerging challenge within MPEG: interactive scene description for XR applications. Ultimately, our search for a scene description format that could be used for last-mile XR scene description and address a wide array of devices – including the lightweight, mobile AR devices of the future – led us to work closely with the Khronos Group, on the development of a set of glTF 2.0 extensions in MPEG-I Scene Description. The recently completed MPEG-I Scene Description ISO/IEC 23090 standard builds on glTF 2.0 using the MPEG vendor extension framework.

In a [glTF meet-up presentation](https://www.khronos.org/developers/linkto/gltf-2.0-extensions-in-mpeg-and-3gpp-real-time-exchange-formats-for-3d-experiences-vid), we shared the use cases and requirements that lead us to glTF, and detailed the architecture of the extensions. This blog summarizes that presentation, explaining the journey to incorporating glTF 2.0 into MPEG-I scene description and how content creators and developers can begin using this new standard to develop interactive real-time applications.

## Use Cases

Metaverse applications call for many people to be able to access and share dynamic scenes simultaneously. Ideally, those people should be able to have a similar experience irrespective of individual hardware, computing or network constraints. We approached this challenge in terms of two discrete use cases.

First, we looked at the need to access interactive 3D scenes in AR/VR/MR (XR) for **entertainment**, such as games, movies, theater, etc. This use case entails considerations such as:

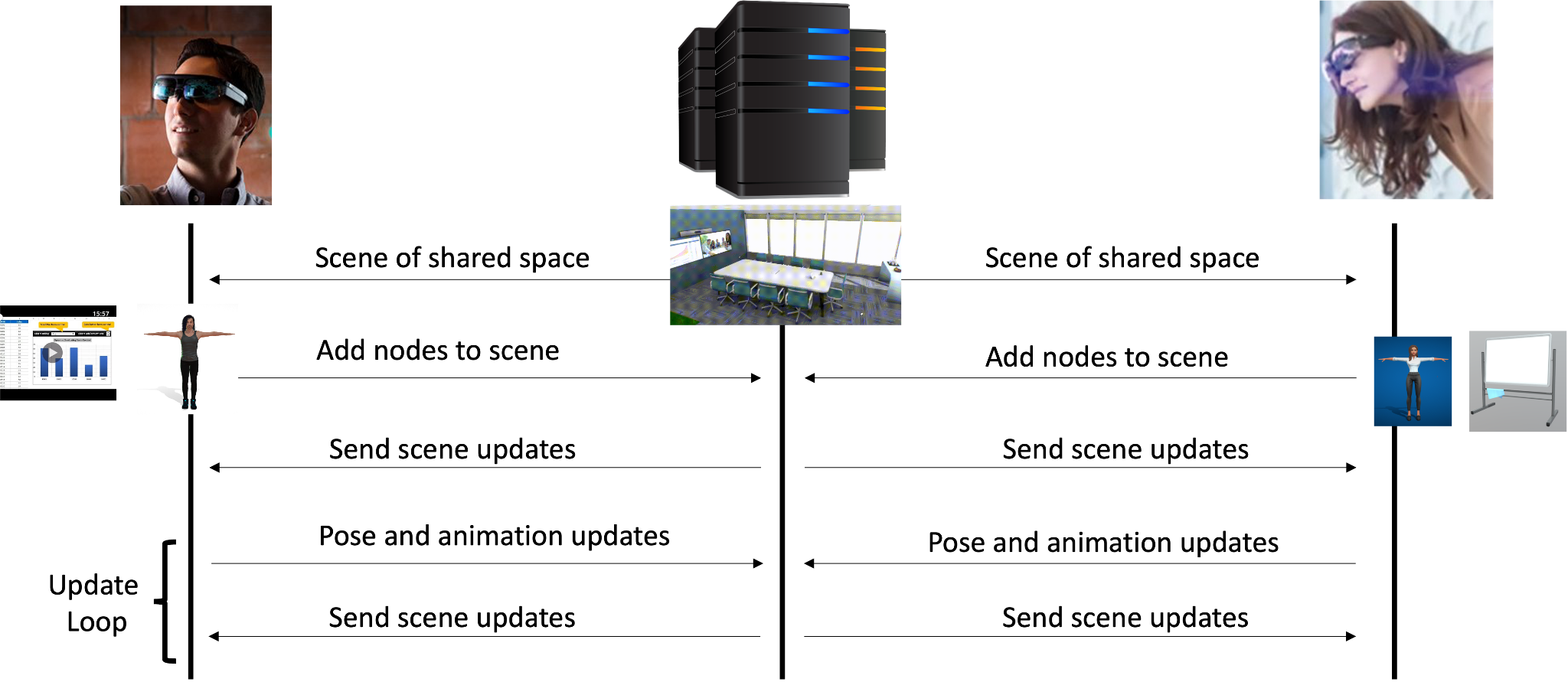
* Static and dynamic objects that need to be downloaded or streamed jointly and presented in parallel.
* Huge volumes and data sizes, stressing the network.
* Interactive and immersive experiences using a variety of input types, including controllers, body movements, hand motions, etc.
* Different media, including 3D and 2D visual information, audio, haptic information, etc.
* Diverse access mechanisms, including both download and smart streaming.
* Diverse hardware, from powerful gaming desktop computers to relatively limited mobile devices.

In addition, we looked at the emerging application of **XR calls and conferencing,** which aims to extend existing IMS and webRTC-based telephony services into immersive multi-user experiences with AR support. This application involves the creation of shared virtual or mixed reality spaces in which users can interact: remote users can join a meeting with realistic 3D avatars and contribute both 2D and 3D content.

## Procedure

Both use cases share the same high-level procedure:

1. Scenes are shared across different participants using a single, central scene composition function.
2. Participants and nodes are added to the scene.
3. The scene is dynamically updated as new users come in, new objects are added, and new assets are shared.
4. Participants can interact with animations, triggers/actions, avatars, and pose tracking; thus triggering scene updates to be distributed to all participants.



In order to facilitate these use cases across diverse hardware, we need an exchange format to dynamically describe the scene, contribute to the scene, and aggregate objects.

## Target Device

A close-up of a pair of glasses

Description automatically generated

Our target device is reference AR glasses, identified by 3GPP as an important device class for mobile AR services. This reference device is a very basic pair of AR glasses, with limited processing power, power consumption and weight. It’s a device users might feasibly wear for extended periods or on a regular basis. We want scenes both rich and lightweight to be accessible on this type of target device.

We expect that a distributed compute architecture including **split rendering** to be a major component of how devices like our reference device will access complex, dynamic scenes. Using this approach, one would operate a game engine on a network server, “pre-baking” the scenes into a format that the device can render. In the simplest form, a split-rendering architecture might just use a runtime application to do some post-processing.

A diagram of a computer

Description automatically generated

This approach splits the GPU power required to run the application. However, a real-time application must also operate in a latency-constrained environment due to the need to track body movements and enable interactivity and immersion. As a result, split-rendering workflows need a session description entry format that will ensure proper synced capabilities exchange and description of scenes and objects within rendered frames. Such an entry format is expected to be processed by an XR player end-point that can set up pipelines for delivery, decoding and rendering based on the information in the entry point document.

## Choosing glTF

The need for a flexible and dynamic scene description solution for immersive experiences has been identified early in the process of specifying technologies for immersive media in MPEG. In looking for a solution to this gap, MPEG reviewed existing technologies to identify formats with a history of being successfully deployed and widely used. The group then evaluated whether one of these formats would satisfy the identified requirements, or whether a new format needs to be defined from scratch.

We were particularly interested in the problem of last-mile distribution to consumer devices, which requires us to reduce complexity, flatten hierarchies, and compress components for efficient network use. An effective approach would need to be adaptive to devices and networks, supporting features like textures, light baking, and more.   
A screenshot of a computer

Description automatically generated

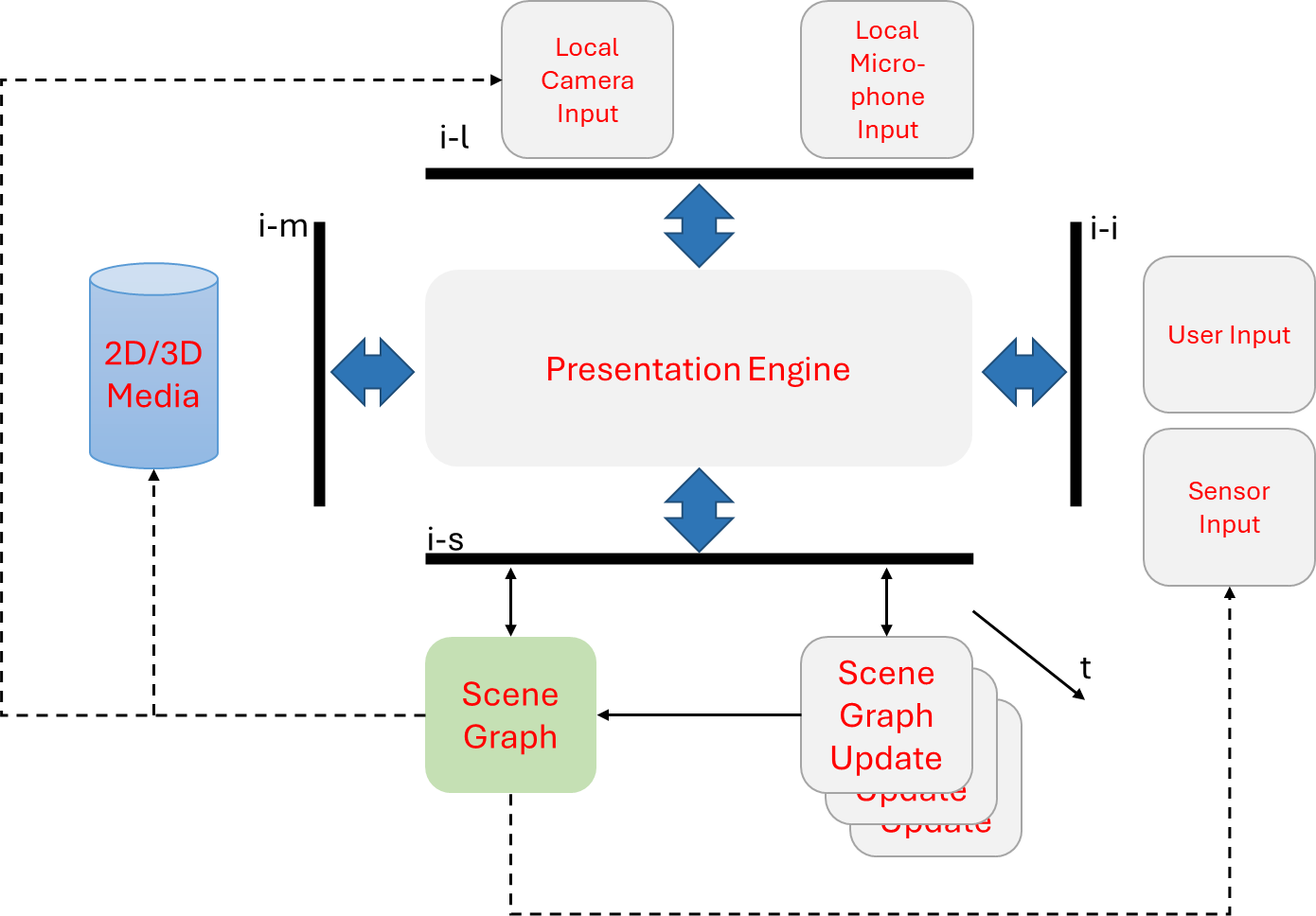
glTF stood out as an exemplary option. glTF 2.0 already provides a multitude of baseline functionalities: It standardizes the creation of 3D scenes and models and is recognized as an open standard. Often referred to as the "JPEG of 3D," it has many of the features required for our purposes. Our challenge was determining if we could extend glTF to integrate new types of real-time media elements, like dynamic point clouds, dynamic meshes, and 2D content. We wanted to develop something like the "HTML5 for XR and the Metaverse." Using existing codecs, HTML5 made media on the Web more mobile-friendly while incorporating real-time multimedia elements and supporting different consumption environments and geographical locations. We envisioned making MPEG more XR-friendly in a similar manner by extending glTF.

Crucially, extending MPEG to XR using glTF would create a vendor-neutral entry point for 3D experiences, so users don't need to rely on proprietary apps or platforms. A standards-based approach is flexible and distributable, enabling highly portable and interoperable shared experiences.

Now, through intensive collaboration with the Khronos Group, we've completed the first version of MPEG-I Scene Description ISO/IEC 23090, integrating glTF 2.0 within the MPEG vendor extension framework.

## MPEG Scene Description Architecture

The presentation engine is the core of MPEG-I scene description architecture. It is analogous to a browser in HTML applications: The presentation engine aims to offer a consistent experience across various platforms and devices.



The user’s entry point to an immersive experience is the scene description document, or scene graph. The presentation engine retrieves the scene graph and any subsequent scene updates from a separate interface called the S interface. For example, in our conference room use case, all interactions – users joining and leaving, sharing content, changing poses, etc. – are scene updates, which are then mirrored as the current status of the shared environment.

The presentation engine then communicates with a media access function, telling it which media and datasets it needs to load, including all their components. The scene graph contains descriptions of how to access all components. This information is then relayed to the media access function for media loading.

Additional interfaces address local interactions and interactive experiences, central to the immersive experience we aim to achieve.

A diagram of a software development

Description automatically generated

Above, we see a more detailed view of the interface between the presentation engine and the media access function. This interface is documented in the ISO/IEC specification 2390, Part 14, which defines the architecture, interfaces, and APIs, such as the media access function API and the buffer API. Our objective is to distinctly separate the media layers from the presentation engine. Ideally, the presentation engine should remain oblivious to mesh or video texture compression techniques like HEVC or AV1.

A diagram of a software flow

Description automatically generated

The media access function (MAF) API receives media descriptions and their locations, seeking the optimal coding or access path for the referenced media. It establishes a pipeline specific to that media format, ensuring the raw media is delivered to the presentation engine via a buffer interface. This interface then stores the decoded and decrypted media in a well-formatted buffer whose description is provided in the scene graph. The scene description provides an understanding of features like glTF accessors and buffers, ensuring that the presentation engine can effortlessly fetch and render data.

One of the core functionalities of the MAF API is optimization. Depending on factors like network changes or the user's viewpoint within a virtual scene, this function might decide not to fetch certain scene elements that are out of view or adjust the detail level of the fetched media. This API also allows you to provide timing, spatial, sync, shader and media consumption data.

A screen shot of a computer

Description automatically generated

For instance, a dynamic mesh might have a component requiring a video texture. The media access function gets a request to make that texture available, looks at alternatives, and constructs a media pipeline tailored to the specific access technology. The pipeline fetches and decodes the media, performing necessary post-processing like YUV to RGB conversions if required. Then, it delivers the media to the buffer for retrieval by the presentation engine.

## MPEG-I Scene Description ISO/IEC 23090 Phase One glTF Extensions

In Phase One, the goal of the MPEG Working Group was to introduce a dynamic, time-based dimension to glTF. This would support dynamic media forms such as video textures, audio, dynamic meshes, and point clouds.

A diagram of a system

Description automatically generated

In the illustration above, the three key glTF extensions are shaded in green and the dependent extensions in gray. Key extensions include:

* **MPEG\_Media:** defines access to externally referenced media.
* **MPEG\_accessor\_timed:** Informs the presentation engine that the accessor directs to a dynamic buffer. This extension prepares the presentation engine to expect that the metadata, including information like the number of vertices, might change from frame to frame.
* **MPEG\_buffer\_circular: a** dynamic buffer where concurrent read and write operations occur. The media access function writes into this buffer, while the presentation engine reads from it simultaneously.

The **MPEG\_Media** extension provides detailed media descriptions, playback instructions, and access methods. You can add as many alternative access pathways as you want to support alternative encodings or different media formats for a wide range of target devices. This flexibility means the media access function must intelligently select the most suitable alternative.

Diverse media sources can be accessed, ranging from dynamic streaming sets using DASH to real-time media through WebRTC, or even locally stored content. This system must address multiple challenges, including the decryption of encrypted content or conversion between formats. The objective is always to meet the presentation engine's expectations.

Circular diagram with text and words

Description automatically generated

The **MPEG\_buffer\_circular** and the **MPEG\_accessor\_timed** extensions work together to ensure synchronization and prevent conflicts. The MPEG\_buffer\_circular supports buffer management, telling the presentation engine to create a circular buffer with the recommended number of frames for the media pipeline. The goal is to prevent frame losses due to speed disparities between the reading and writing processes.

A diagram of a computer

Description automatically generated

Currently, glTF uses a static buffer. The **MPEG\_accessor\_timed** extension introduces support for dynamic buffers to enable real-time and streamed media. One can set different accessor parameters, offering flexibility based on potential data dynamics. For instance, in the case of a video texture, certain parameters, like video width and height, remain consistent even when adaptive streaming is in play. Thus, they're marked static. Conversely, in situations like a dynamic meshes, all parameters, such as number of vertices and faces, could change from frame to frame. **MPEG\_accessor\_timed**  gives you access to another buffer view that contains the dynamic parameters, allowing for the description of these frequent changes.

**MPEG\_accessor\_timed** has headers for both Accessor and bufferView fields that may change over time:

* Accessor information:
* componentType
* bufferView
* type
* normalized
* byteOffset
* count
* max
* min
* bufferView information:
* bufferViewByteOffset
* bufferViewByteLength
* bufferViewByteStride

Use of these dynamic buffers is optional, adding adaptability without unnecessary complexity.

Video textures is one of the first extensions that uses this concept: the timed accessor tells the presentation engine that it needs to initiate a dynamic buffer in order to access the video texture. If a device or system doesn't support video textures, it can default to a regular static image placeholder.

## MPEG\_audio\_spatial

The MPEG Working Group also created a glTF extension for spatial audio, enabling more immersive experiences. In the current extension, supported formats include mono object-based sources and HOA (Higher Order Ambisonics).

Audio sources can be linked with effects; the current version of the spec supports reverb.

MPEG\_audio\_spatial allows you to track the user’s position with an “audio listener” node. We recommend that the “audio listener” be added as either a child node of the scene camera or an extension to the camera. This allows MPEG\_audio\_spatial to adapt based on the user's movements and provides realistic sound orientations.

## 

## Additional Phase One glTF Extensions

Phase One development on MPEG-I Scene Description ISO/IEC 23090 also included various dependent glTF extensions, supporting the functionality of the primary extensions:

* **MPEG\_scene\_dynamic** - indicates that the scene description document will be updated.
* Updates are provided through the JSON patch protocol.
* Patch sample is an atomic update operation (all patch operations part of one transaction).
* Consistency/validity of scene after application of a patch is the responsibility of the author.
* **MPEG\_viewport\_recommended** - provides dynamically changing information including translation and rotation of the node and the camera object, as well as the intrinsic camera parameters of the camera object.
* **MPEG\_animation\_timing** - provides alignment between MPEG media timelines and animation timelines defined by glTF 2.0.

## 

## Demo

The MPEG Working Group is working on open sourcing a Unity Player implementation of MPEG-I Scene Description ISO/IEC 23090. You can view a brief demo of this implementation below.

<iframe width="560" height="315" src="https://www.youtube.com/embed/U9VgiYORNrA?si=3lULoMwNYGWqFpEN&amp;clip=UgkxzzdD-PfRUttoM7oigc6CL73BfWa3Hcuj&amp;clipt=EPCcgwEY8JOGAQ" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>

The Working Group is also working on a Blender implementation so you can author and export scenes using MPEG-I Scene Description.

## 

## Phase Two and Future glTF Extensions

A diagram of a software project

Description automatically generated with medium confidence

In Phase Two, the MPEG Working Group is developing additional glTF extensions designed to enrich the user experience by incorporating new features to bridge any existing gaps between MPEG-I and glTF and our target XR use cases. Phase Two introduces support for:

* Augmented Reality (AR)
* Enhanced interactivity
* Dynamic lighting
* Haptic feedback
* 6 Degrees of Freedom (6 DOF)
* Improved audio

**MPEG\_scene\_anchor** and **MPEG\_node\_anchor** ensure smooth functionality in Extended Reality (XR), allowing AR users to anchor the virtual scene to their real world, making it feel like it's part of their room or office.

The intention is to have the scene description communicate to the receiver about specific "trackables" in the environment. These might be familiar features like a floor but could also be application-specific elements such as a QR code. This guidance helps anchor the virtual scene to the real world.

**MPEG\_scene\_interactivity** and **MPEG\_node\_interactivity** operate based on behaviors that link triggers to actions. Triggers could be events like a collision, proximity user input, or the visibility of an object. Actions could involve activating or transforming objects, starting animations, changing materials, or initiating haptic feedback.

The MPEG Working Group is also working on integrating avatars via **MPEG\_node\_avatar**. At their core, avatars are essentially meshes, potentially dynamic or animated. Avatars enable interactivity. They act as your digital twin, allowing you to use your hands to interact with objects such as turning on a TV, opening a door, or using a light switch. It's essential to segment the avatar so that specific body parts, like hands or fingers, can be recognized for interactivity. This extension will leverage glTF-based avatar representation formats developed by the VRM Consortium and others, and enable the integration of such avatars into interactive scenes.

We are also developing dynamic lighting via **MPEG\_dynamic\_light\_texture\_based.** This feature allows for dynamic lighting that is more like a video than a static source: coefficients like reflectance can change over time. In AR applications, this extension will allow you to monitor the current light conditions and adjust your rendering accordingly.

A collage of images of clouds and a computer screen

Description automatically generated

The MPEG haptics extensions, **MPEG\_haptic** and **MPEG\_material\_haptic**, support various features like stiffness, friction, and temperature – our aim is to be inclusive of both current and future haptic responses. Our system supports diverse haptic experiences based on triggers in the virtual world, such as the texture of a virtual object. For instance, running your hand over a digital wall might offer specific haptic feedback based on its texture, whether smooth or rough.

The MPEG Working Group is also enhancing our audio support. We aim to integrate a 6 DOF audio solution for a more immersive sound experience, moving beyond our current spatial audio offerings. This system will account for factors like acoustic materials, offering something akin to a PBR for audio.

## Conclusion

glTF has proven to be an effective and easily extensible 3D format for MPEG to integrate interactive scene descriptions for XR applications with the recently completed MPEG-I ISO/IEC 23090 standard. The MPEG working group at ISO looks forward to expanding our cooperation with the Khronos Group to continue to extend glTF to bring together the worlds of interactive 3D and streaming media.

## Resources

Developers interested in further exploring MPEG-I Scene Description ISO/IEC 23090 are encouraged to explore the following resources:

* [MPEG-I Scene Description ISO/IEC 23090 Khronos Extensions](https://github.com/KhronosGroup/glTF/tree/main/extensions/2.0/Vendor)
* [GitHub Repository](https://github.com/MPEGGroup/glTF) - provide feedback, report issues, or suggest new features
* [MPEG-I Scene Description Whitepaper](https://www.mpeg.org/wp-content/uploads/mpeg_meetings/140_Mainz/w22138.zip)
* [MPEG-I Reference Software](https://www.5g-mag.com/reference-tools)
* [MPEG-I Open Source Unity Implementation](https://github.com/5G-MAG/rt-xr-unity-player)
* [5G-MAG: XR Integration into 5G](https://5g-mag.github.io/Getting-Started/pages/xr-media-integration-in-5g/)

You can also view the original presentation in its entirety on [YouTube](https://www.khronos.org/developers/linkto/gltf-2.0-extensions-in-mpeg-and-3gpp-real-time-exchange-formats-for-3d-experiences-vid).