 ISO/IEC JTC 1/SC 29/WG 3 N00369

**ISO/IEC JTC 1/SC 29/WG 3**

**MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title: Requirements Coverage of MPEG-I Scene Description**

**Status:** Approved

**Date of document:** 2021-10-15

**Source:** ISO/IEC JTC 1/SC 29/WG 3

**Expected action:** ACT

**Action due date:** 2021-10-15

**No. of pages:** 12 (with cover page)

**Email of Convenor:** young.L@samsung.com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 3**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC 1/SC 29/WG 3 N** **00369**

**Online – October 2021**

|  |  |
| --- | --- |
| **Source** | **Systems** |
| **Title** | **Requirements Coverage of MPEG-I Scene Description** |
| **Editor** | **Thomas Stockhammer** |

# Scope

This document is primarily provided for the following purpose

* Summarize the requirements for the MPEG-I Scene Description standard development
* Reference the functionalities in glTF2.0 that address the requirements
* Identify the gaps in glTF2.0 in order to fulfil the requirements
* Provide reference to Drafts of ISO/IEC 23090-14 to document how MPEG-I Scene Description extensions address the gaps
* Provide a continuous project status on the requirements coverage

# Approved Requirements Output Documents

This document addresses the coverage of glTF2.0 against the following requirements:

* N19511 Requirements for MPEG-I Phase 2

# glTF as a Baseline

During MPEG#128, a document was produced

* N18869 Considerations on Working Draft for Scene Description

Based on this document and further discussions, it was agreed during MPEG#128, that MPEG-I Scene Description is developed as an extension to Khronos' glTF2.0 specification. This specification can be accessed here: <https://github.com/KhronosGroup/glTF/blob/master/specification/2.0/README.md>

N18869 contains an initial evaluation on how Khronos' glTF2.0 specification addresses MPEG-I Scene description requirements and what are the potential necessary extensions.

This document uses the above as baseline but continues updates on the requirements and their fulfilment by either glTF2.0 technologies or by developed and agreed MPEG extensions are documented.

The latest additions reflect the requirements based on the status of the DIS for 23090-14 and the TuC in SC29WG3\_N00264 and SC29WG3\_N00208, respectively.

# Initial Architecture

The initial architecture when the requirements were developed is shown below. The interfaces are referred to in the below requirements.



# General Requirements on Scene Description

The following table provides these columns

* Number: the requirements number as documented in N19511
* Fulfillment gltf2.0: Does gltf2.0 fulfill this and how
  + Pink: completely fulfilled by glTF2.0, no MPEG work necessary
  + Yellow: partially fulfilled by glTF2.0, but needs MPEG extension
  + Orange: partially fulfilled by glTF2.0, but prefer/ask Khronos to do extension
  + Blue: not fulfilled by glTF2.0, needs MPEG extension
* Status MPEG Extensions: documents the basic principles for the MPEG extensions and the status in the latest DIS.
* Suitability as Test Scenario: Documents how this requirement could be demonstrated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Requirements N19511 | Fulfillment gltf2.0, including N18869 | Status MPEG Extensions | Suitability as Test Scenario |
| Phase 2a Requirements | | | | |
| General | | |  | 1 – must have  2 – quite important  3 – nice to have  4 – if there is lots of time  5 – unnecessary |
| 4 | If possible, the solution shall define interfaces to integrate existing scene description formats rather than define a new scene description format. | Gltf2.0 is an existing scene description. The design based on glTF can be used as a blueprint for developing similar extensions to other existing scene description formats such as USD and OSG. |  | 1 basic test. Use gltf. |
| 5 | The solution shall reuse existing interfaces/API definitions (also from other SDOs) whenever possible and appropriate | Gltf2.0 basic functionalities to build APIs on top of it. Relevant ones will be defined by MPEG but re-using also concepts defined in W3C for HTML-5. | Addressed in DIS, APIs between presentation engine and media access functions defined. | 1 basic test. Use gltf. |
| 6 | It shall be possible to indicate object information in the scene description format such that one can derive a consistent relationship between object map information in a video bitstream and object information signaled in the scene description format | This requirement is primarily addressed by the media access function.  In addition, information in the scene description takes precedence over information in the media and shall always be present. The scene description may refer to object information in the stream, but in this case, it requires that the information if provided through a well-defined buffer to the presentation engine. | Addressed in DIS as part of the media access function and the buffer management. | 1 add a V-PCC/MIV object to a gltf scene.  Add a 360 object to a gltf scene  Combine a V-PCC/MIV object with a 360 object. |
| 7 | The information signaled in the scene description format shall be consistent with presentation layer information signaled in the video/audio bitstream at the codec layer as well as information signaled in the media formats at the systems level via ISOBMFF, DASH, etc… In case of contradicting information, the scene description format shall take precedence | This needs to be well-documented in the standard and requires detailed and consistent information for each MPEG media that is added to Scene description. It is a matter of well-defined extensions.  The MPEG-I architecture has been designed to properly decouple the media format as it is stored or delivered from the actual media data that is used for rendering. The Presentation Engine expects media data to be in raw graphics/audio formats (e.g. vertex buffers, color buffers, normal buffers, audio sample buffers, …). Any decoding and processing of the actual media to reformat it for rendering is done in the processing pipeline and is guided by the Media Access Function based on information received from the scene graph.  The following figure shows how the pipelines can be constructed to feed raw data into the buffers, which will be used for rendering by the Presentation Engine.  A close up of a screen  Description automatically generated | Addressed in DIS as part of the media access function and the buffer management. | 2  See above under 6 |
| 8 | It shall be possible to associate Information in the media bitstream targeted for the scene description with a processing model. | Based on the reference design, it requires that metadata in the bitstream is loaded into buffers for consumption the presentation engine with well-defined syntax and semantics.  Buffers may be static or dynamic. | Under development. Basic functions defined in DIS, but needs more work especially on integrating MPEG-defined media formats such as OMAF, PCC, etc.  Also dynamic metadata, for example carried in SEI messages such as recommended viewports would need processing models.  If processing model is not provided, such metadata is expected to be “lost”/”ignored” in the MAF processing. | 1  See test scenarios under 6. |
| 9 | The scene description shall enable the option to describe the scene using geographical coordinate systems. | glTF2 has currently no support for geographical coordinate systems. | Under development in TUC as part of the media access function and the buffer management. | 3  Test scenario tbd. |
| 10 | The scene description shall enable modular rendering, i.e. smaller portions of the scene can be independently accessed and rendered. | The current design permits to only access media that is in the viewport of the scene. The MAF design allows to optimize the access to data according to the current user pose and viewport. | Addressed in the in the DIS as part of the media access function and the buffer management. | 3  Test scenario tbd, but requires multiple objects at different viewports. For example two room scenario, where you can walk through. |
| Reference Scene Description Selection | | |  |  |
| 11 | The scene description shall support audio, video and other media formats standardised by MPEG. | glTF2 has currently no support for audio or video media formats. However, glTF supports several still image formats. | Partially addressed in DIS. Different approaches taken   * Use glTF to describe 2D buffers. Buffers are circular * Audio extensions also added.   Additional information in TUC and needs more work, for example to directly integrate 3D objects such as PCC or 360. | 1  For each of the three approaches a test scenario is desired.  Some overlap with the one in 6  Definitely need an A/V scenario. |
| 12 | The scene description shall enable the support of other visual or audio media formats. | With addressing 2b, media other than MPEG can be supported by adding such media to MPEG containers.  By setting the up the MAF pipelines, one can also use other codecs. As long as the raw format is supported, this is independent of the codecs.  Glft itself supports several 3D image formats (Alembic, etc.) | Mostly addressed in the DIS.  Some improvements on terminology expected, e.g.  *Raw* format: (2D, 360, Point cloud, mesh, etc.,)  *Coded* format: (compressed 2D or raw format, e.g. V-PCC, G-PCC, Draco, Alembic, OpenVDB, …)  *Track* formats: time-synced and packaged for delivery, accessed by MAF (MP4, CMAF, etc.) | 3  Combining MPEG media with for example Draco may be a nice test scenario. |
| 13 | The scene description shall support definitions to indicate how sub-graphs and objects are related in terms of their temporal, spatial and logical relationships | This requirement is partially supported, since with exception of animations, all nodes of a scene graph are assumed to be active at time 0 and there is no concept of scene updates in glTF2. Regarding the logical relationships between media elements within a scene, means of interactivity and possible constrains to it should be defined. | Partially supported in gltf by nodes (spatially and logically).  Scene description updates are under development and partially addressed in DIS. More work in EEs and TuC. | 4 |
| 14 | The scene description shall support composition of digital representations of natural and synthetic objects. | Gltf permits to add different objects through nodes and buffers into a scene representation. Objects can be then be rendered by a gltf renderer. |  | 1  Likely covered by one of the above scenarios. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 | The scene description shall support synchronisation between objects and attributes in the scene. | This is only supported through animations at the moment. MPEG needs to provide appropriate extensions to address this issue. | Addressed in the DIS. Timed animation is supported | 1  Likely covered by one of the above scenarios. |
| 16 | The scene description shall support spatial and temporal random access. | glTF supports spatial random access through node hierarchy,  glTF2 does not support temporal random access into the scene, since there is no notion of scene timeline or the ability to update and there is no inherent support for timed media in glTF2.  MPEG needs to provide appropriate extensions to address this issue. | Under development in DIS, but more work is carried in EEs. | 2 - 3  May be an extension to one of the above scenarios. |
| 17 | The scene description should support information to enable a renderer to perform path tracing. | glTF2 supports path tracing extensions.  Physically Based Rendering is collectively used to denote a set of rendering techniques that aim at providing an accurate rendering of the scene that is indistinguishable from a photograph of the real scene. These techniques usually rely on precise calculations of the light transfer in the scene by estimating the reflection and refraction of light, coming from all light sources in the scene, on the objects available on the scene. For these techniques to work properly, there is need to properly describe the materials of the objects and how these materials interact with light as well as the ability to describe the geometry orientation of the objects themselves. In addition, there is need to describe light sources in the scene.  Physically Based Rendering material feature originally came from an extension, but it was made part of the GLTF 2 standard. Values are stored in a material asset that gets applied to the GLTF file's geometry; some values vary per pixel of a texture map, and some are constants that get used equally everywhere the material is assigned.  As for the shader that then interprets the material values, there are GLTF extensions that allow custom user-created shaders, but that's not part of a standard GLTF 2 workflow. Normally, the PBR shader that interprets those material values is outside the GLTF file, supplied by the app reading it. The idea of the PBR approach is to standardize one flexible "ubershader" implementation across many different apps--so the final output should look fairly similar everywhere.  The core functionalities are here:   * pbrMetallicRoughness material model * normalTexture, emissiveTexture, and occlusionTexture map support for these materials * A non-normative BRDF implementation   The extensions are here:   * [KHR\_materials\_clearcoat](https://github.com/KhronosGroup/glTF/blob/master/extensions/2.0/Khronos/KHR_materials_clearcoat/README.md) * [KHR\_materials\_pbrSpecularGlossiness](https://github.com/KhronosGroup/glTF/blob/master/extensions/2.0/Khronos/KHR_materials_pbrSpecularGlossiness/README.md) * [KHR\_materials\_unlit](https://github.com/KhronosGroup/glTF/blob/master/extensions/2.0/Khronos/KHR_materials_unlit/README.md) * [KHR\_lights\_punctual](https://github.com/KhronosGroup/glTF/blob/master/extensions/2.0/Khronos/KHR_lights_punctual/README.md)   In addition, the following vendor specific extensions are also available:   * EXT\_lights\_image\_based which stores light probes from the scene in an environment map * ADOBE\_materials\_thin\_transparency to represent materials with transparency   Ray tracing is one of the techniques to achieve PBR. It is usually implemented in render passes that iterate over each pixel and cast light rays ending in that pixel to determine which objects will be reflected on that pixel by performing a hit test on the scene geometry. The way the rays reflect on the surfaces of these objects is determined by the object’s material characteristics. The above tools and extensions allow for an accurate implementation of ray tracing on a scene described by glTF. |  | 3 |
| 18 | The scene description shall support sub-graph representation that allows modular rendering e.g. leafs in the scene description tree can also be packaged and referenced individually from a parent scene description and container. | glTF2 supports modular design of leaves by buffers.  The following diagram shows the structure of glTF 2.0:  A picture containing electronics, control  Description automatically generated  It starts with a **scene** node which in turns has a hierarchy of nodes (a set of disjoint trees). This is exactly the definition of a scene graph, albeit with a limitation of nodes having a single parent for simple parsing and rendering.  The example assets that are provided show how a user can navigate the scenes, e.g. walking inside a conference room or living room, where other 3D assets/objects may also be placed. If in doubt, effort should be done to define **test scenarios** to validate capabilities of the technology. |  | 3 |
| 19 | The scene description shall support references (e.g. URLs) to external media resources in place of embedded file references. | glTF2 supports external media by reference.  Extensions for MPEG are needed to address different delivery use cases. | Addressed in DIS using MAF and APIs from GLTF to MPEG media delivery. | 1  Likely covered by one or several of the above scenarios |
| 20 | The scene description shall support a mechanism to safely customize behavior for nodes like camera, texture, geometry, audio, and object placement nodes through sandboxed, validated domain specific shaders or scripts for these nodes without affecting the functionality or forcing changes to the root node graph or other node types; i.e. provide a mechanism to safely extend the scene description. | Gltf2.0 support this. The following statements can be found in the glTF 2.0 specification:  *The GL Transmission Format (glTF) is an API-neutral runtime asset delivery format. glTF bridges the gap between 3D content creation tools and modern 3D applications by providing an efficient,* ***extensible****, interoperable format for the transmission and loading of 3D content.*  *glTF solves these problems by providing a vendor- and runtime-neutral format that can be loaded and rendered with minimal processing. The format combines an easily parseable* ***JSON scene description*** *with one or more binary files representing geometry, animations, and other rich data.*  *Extensibility. While the initial base specification supports a rich feature set, there will be many opportunities for growth and improvement. glTF defines a mechanism that allows the addition of both general-purpose and* ***vendor-specific*** *extensions.* |  | 4 |
| I-s Interface: Presentation engine interface | | |  |  |
| 21 | It shall be possible to update the whole scene-graph, a sub-graph, or a node in the scene description | glTF2 does not come with a scene update mechanism.  MPEG needs to provide appropriate extensions to address this issue | Partially supported in the DIS, work ongoing in EEs. | 1  Need to define a scenario. This is part of the current discussion |
| 22 | It shall be possible to correctly render a 6DoF Presentation after a random access in time | glTF2 does neither have support for a timing model nor scene updates through time and as such every glTF2 is considered a random access point in time.  MPEG needs to provide appropriate extensions to address this issue | Partially supported in the DIS, work ongoing in EEs. | 2  Likely covered by extension from above |
| 23 | It shall be possible to perform timed scene description updates | glTF2 does not come with a scene update mechanism.  MPEG needs to provide appropriate extensions to address this issue | Partially supported in the DIS, work ongoing in EEs. | 1  See 21 |
| 24 | It shall be possible to associate a scene description update with the corresponding scene description | glTF2 does not come with a scene update mechanism.  MPEG needs to provide appropriate extensions to address this issue | Partially supported in the DIS, work ongoing in EEs. | 2  See 21 |
| 25 | It shall be possible to use a scene description as the entry point to a 6DoF presentation | Gltf2.0 supports exactly this. |  | 1  Likely covered by many test scenarios |
| I-m Interface: Media access Interface | | |  |  |
| 26 | It shall be possible to access timed and non-timed, 2D and 3D media (meshes, point clouds, audio elements, …), stored locally or over the network | glTF2 has support for buffers and images that fetch their content from a local file system or over the network. However, there is no support for timed media.  MPEG needs to provide appropriate extensions to address this issue | Addressed in DIS using MAF and APIs from GLTF to MPEG media delivery. | 1  Likely covered by many test scenarios |
| 27 | It shall be possible to pre-fetch media that the presentation engine expects to be used in the presentation | glTF2 has support for buffers and images that fetch their content from a local file system or over the network. However, there is no support for timed media.  MPEG needs to provide appropriate extensions to address this issue | Addressed in DIS using MAF and APIs from GLTF to MPEG media delivery. | 3  Nice to have, but not relevant. We know that MPEG delivery can fulfill this. |
| 28 | It shall be possible to retrieve media depending on the desired level of detail | Retrieving different levels of detail of geometry and texture is generally possible in glTF2 through pre-processing the glTF2 scene. These operations are not supported by the glTF2 scene itself.  MPEG needs to provide appropriate extensions to address this issue | Partially addressed in DIS using MAF and APIs from GLTF to MPEG media delivery (for example your viewing position).  More work may be needed. | 3  Would be nice, but requires likely some work. |
| 29 | It shall be possible to retrieve and access referenced media partially in time and space | Currently, glTF2 references lack any attributes to indicate time or space points in the referenced media. Several MPEG media components offer partial access capabilities, as such means to describe such partial access in time and space should be defined.  MPEG needs to provide appropriate extensions to address this issue | Partially addressed in DIS using MAF and APIs from GLTF to MPEG media delivery (for example your viewing position). This allows to use viewport-dependent streaming as provided by MPEG media-  More work may be needed. | 3  Would be nice, but requires likely some work. |
| 30 | It shall be possible to describe position, orientation, and visual/acoustic characteristics when rendering referenced media | Supported for visual objects through gltf2.0.  glTF2 has no support for audio nodes.  MPEG needs to provide appropriate extensions to address this issue  MPEG needs to provide appropriate extensions to address this issue | Addressed in DIS for audio extensions. | 1  Covered by above scenarios |
| 31 | It shall be possible to synchronize media objects/resources and media components of a single object | glTF2 has no support for timed media nodes.  MPEG needs to provide appropriate extensions to address this issue | Addressed in DIS using MAF and APIs from GLTF to MPEG media delivery. | 1  Covered by above scenarios |
| 32 | Audio elements shall be rendered consistently with their corresponding visual elements, if such visual elements exist. | glTF2 has no support for audio nodes.  MPEG needs to provide appropriate extensions to address this issue | Addressed in DIS using, audio extensions, MAF and circular buffers. | 1  Covered by above scenarios |
| 33 | The specification shall enable synchronization of audio and video of users and the scene. | glTF2 has no support for audio from the scene or from the user.  MPEG needs to provide appropriate extensions to address this issue  May also relate to real-time | Addressed in DIS using, audio extensions, MAF and circular buffers. Real-time support for users is likely not be part of phase 2a. | 1  Covered by above scenarios |
| Phase 2b | | | | |
| Reference Scene Description Selection | | | | |
| 77 | The scene description should support nodes and attributes in order to implement natural laws of light, energy propagation and physical kinematic operations. | glTF2 supports these extensions. See Requirement 17. |  |  |
| 78 | The scene description should support nodes and attributes in order to implement natural laws of acoustic energy propagation and physical kinematic operations. | The support for acoustic characteristics of materials and nodes is non-existent in glTF2.  MPEG needs to provide appropriate extensions to address this issue. | Basic principles addressed for audio extensions in DIS  More advanced features are part of MPEG-I audio available for phase 2b. |  |
| 79 | The scene description should support description of ray-traced camera parameters for rendering | glTF2 supports ray tracing extensions. See requirement 17. |  |  |
| 80 | The scene description shall support parametric models for use in rendering environmental acoustic behaviour (e.g. reverberation, occlusion and directivity). | glTF2 has no support for audio in the scene.  MPEG needs to provide appropriate extensions to address this issue. | Basic principles addressed for audio extensions in DIS  More advanced features are part of MPEG-I audio available for phase 2b. |  |
| I-l Interface: Local capture Interface | | |  |  |
| 81 | It shall be possible to discover and configure local capture modalities | glTF2 itself has no support for local modalities but OpenXR provides APIs to do that.  MPEG needs to provide appropriate extensions to address this issue | Left to phase 2b. Needs work |  |
| 82 | It shall be possible to adjust the presentation based on local capture modality availability | glTF2 itself has no support for local modalities but OpenXR provides APIs to do that.  MPEG needs to provide appropriate extensions to address this issue | Left to phase 2b. Needs work |  |
| 83 | It shall be possible to reference media objects that are captured locally using different capture modalities | glTF2 itself has no support for local modalities but OpenXR provides APIs to do that.  MPEG needs to provide appropriate extensions to address this issue | Left to phase 2b. Needs work |  |
| 84 | It shall be possible to provide feedback through available actuators | glTF2 itself has no support for local modalities but OpenXR provides APIs to do that.  MPEG needs to provide appropriate extensions to address this issue | Left to phase 2b. Needs work |  |
| I-i Interface: User inputs Interface | | |  |  |
| 85 | It shall be possible to discover user interactivity modules | glTF2.0 supports this, but interaction with timed media still needs to be defined.  MPEG needs to provide appropriate extensions to address this issue | Left to phase 2b. Needs work |  |
| 86 | it shall be possible to define custom interactivity procedures based on input from the user or from the user’s devices and sensors | glTF2.0 supports this, but interaction with timed media still needs to be defined.  MPEG needs to provide appropriate extensions to address this issue | Left to phase 2b. Needs work |  |
| Export | | |  |  |
| 87 | The scene description shall support information to enable a renderer to output raster data (image, and video), volumetric data (point clouds, meshes, arrays of voxels, and reflectance fields) and audio. | glTF 2.0 supports this requirement for all mentioned visual formats. An extension is required to extend the support or audio. | Audio support in DIS.  Raster data support in DIS  Volumetric data partially supported in DIS, more work is needed. |  |
| 87 | The scene description shall support a scriptable export output node for asynchronously exporting (as a file stream or buffer) any or all parts of the scene description connected to a node into a simpler or flattened representation. | glTF 2.0 is already a simple flat graph representation. Tools and scripts are available to convert it into a binary representation. |  |  |