

28/09/2021

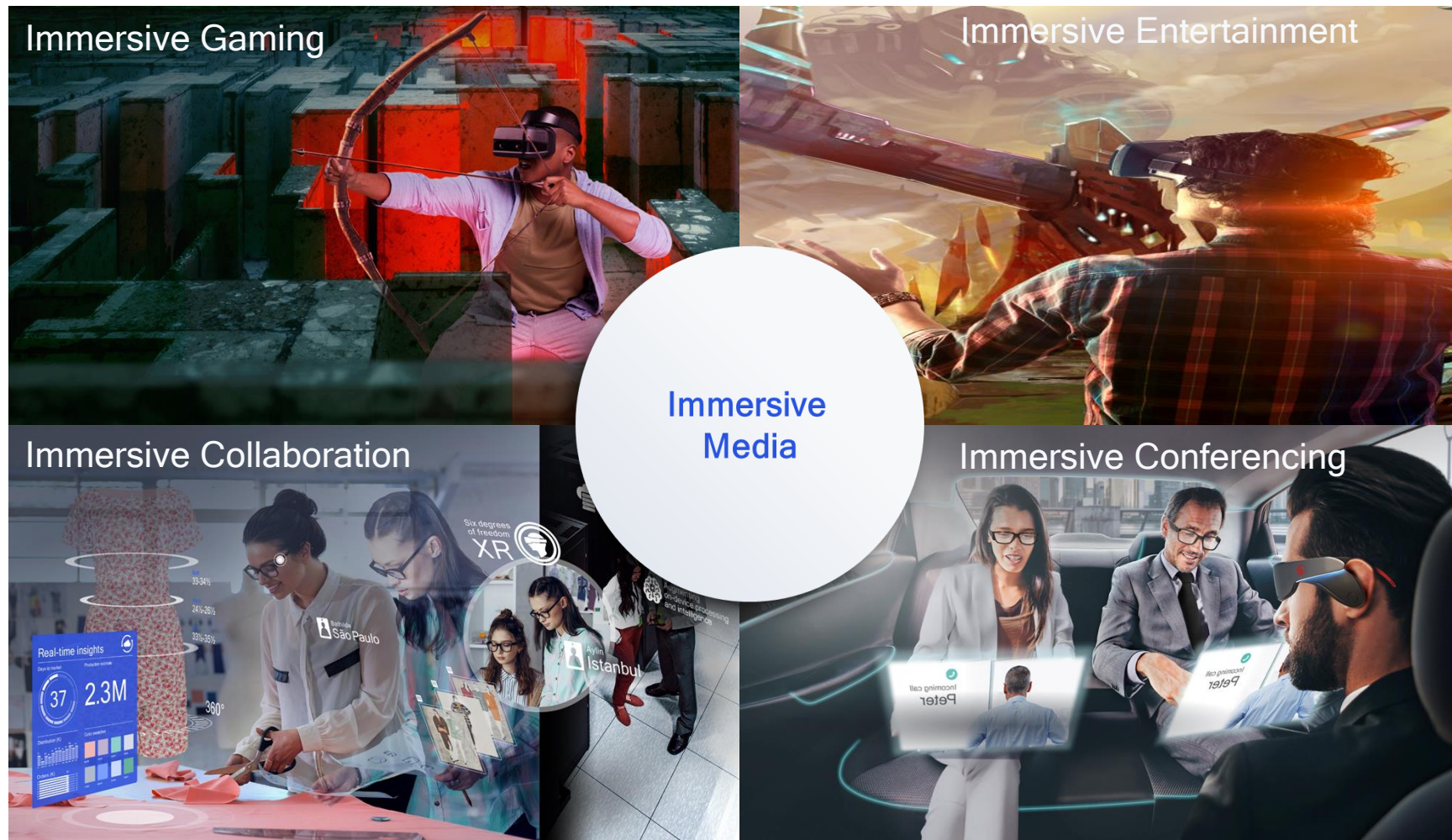


MPEG-I Scene Description

Imed Bouazizi

Lukasz Kondrad

MPEG-I Target Use Cases



Last Mile vs Interchange



Last Mile

- Low Complexity
- Flat Hierarchy
- Compressed components
- Adaptive and network friendly
- Support for Texture/Light Baking

Interchange

- High fidelity
- Superstructure
 - Hierarchical
 - Distributed
 - Preserves author's intents/choices
 - Documents authoring process
- Lossless
- Preserves asset's metadata/versioning



Last Mile

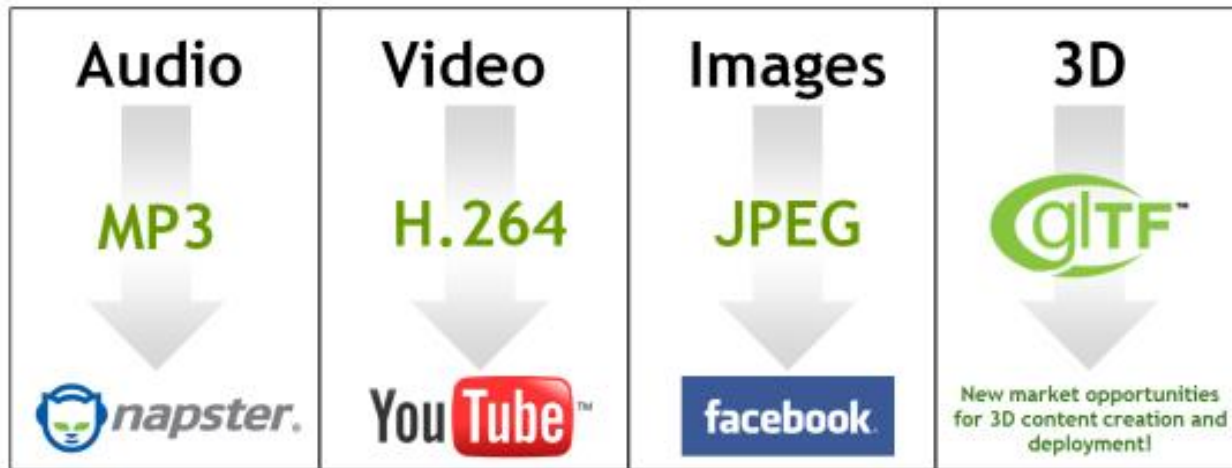


Interchange

Towards the MPEG of 3D



glTF - The JPEG of 3D!



Time Dimension



Dynamic 6DoF Scenes



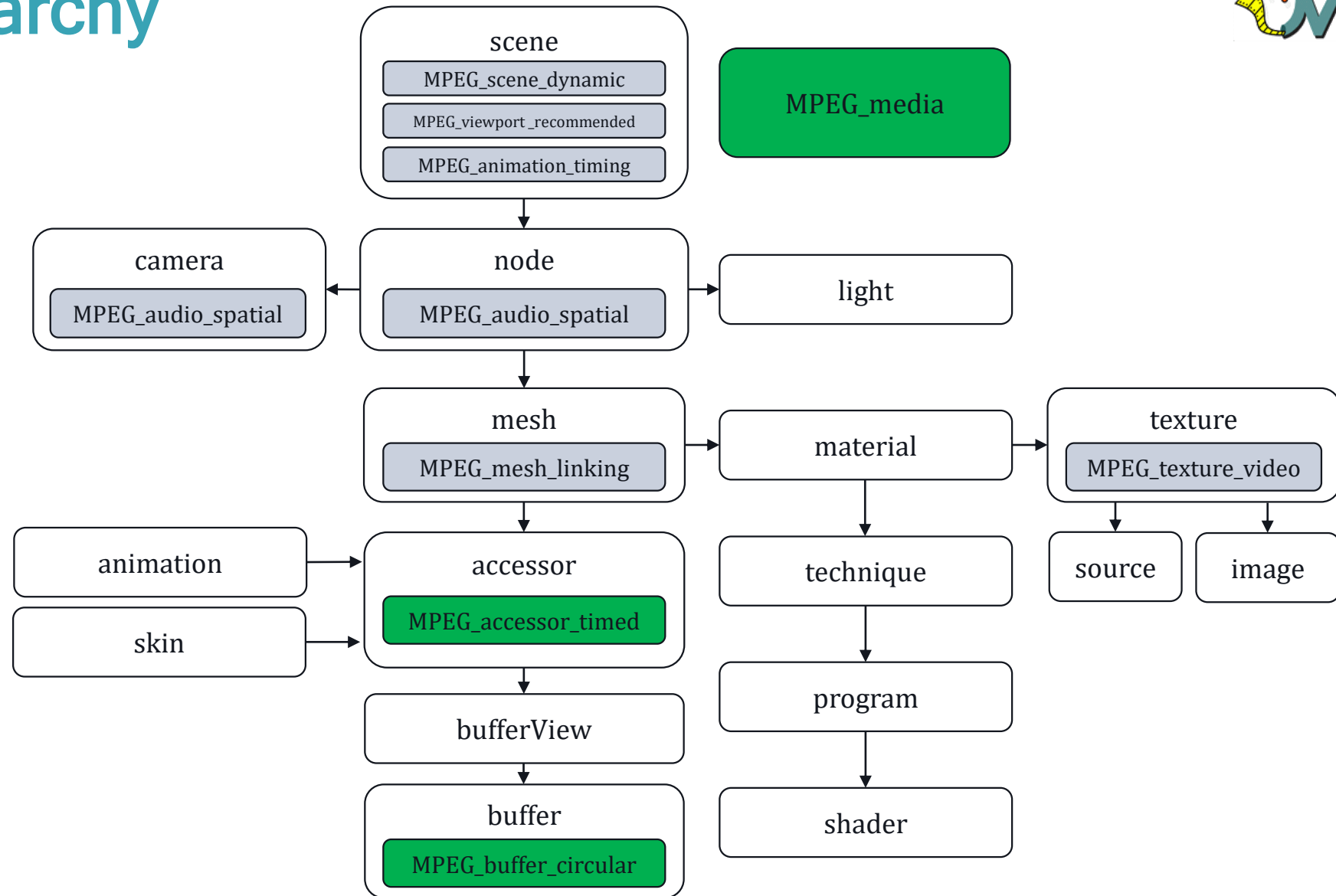
Background and Requirements

- MPEG decided to extend an existing format rather than starting from scratch
 - No BIFS, MPEG-7, LAsER, etc. anymore
- glTF was recognized by MPEG as the best candidate for a baseline
 - Suitable for distribution/last-mile
 - Wide support with importers/exporters and renderers available in all environments
- The following gaps were identified:
 - No support for Audio
 - No support for timed media (dynamic meshes/point clouds, video textures, ...)
 - No support for scene updates
 - No decoupling of media access from rendering
- MPEG defined extensions to address these gaps
 - All extensions use the **MPEG_** namespace
- Key Requirements
 - Describe the expected buffer format of data after decoding/processing
 - Support alternative ways to access the data (local storage, network storage, live streaming, real-time communication, ...)
 - Permit metadata changes for dynamic media (e.g. # vertices/points of a dynamic object)
 - Decouple Rendering from Media Access

Node Hierarchy



 Core Extensions



External Media References

MPEG_media extension

- Top-level extension to glTF 2.0
- Allows referencing all types of media
 - Timed and non-timed
 - Compressed and non-compressed
 - MPEG and non-MPEG
- It supports different types of delivery
 - DASH & CMAF
 - WebRTC
 - HLS & CMAF
 - Local Storage (ISO BMFF, MP4)
- Orthogonal Functions: encryption, etc.
- This extension decouples Media Access Function from Presentation Engine in the Scene Graph



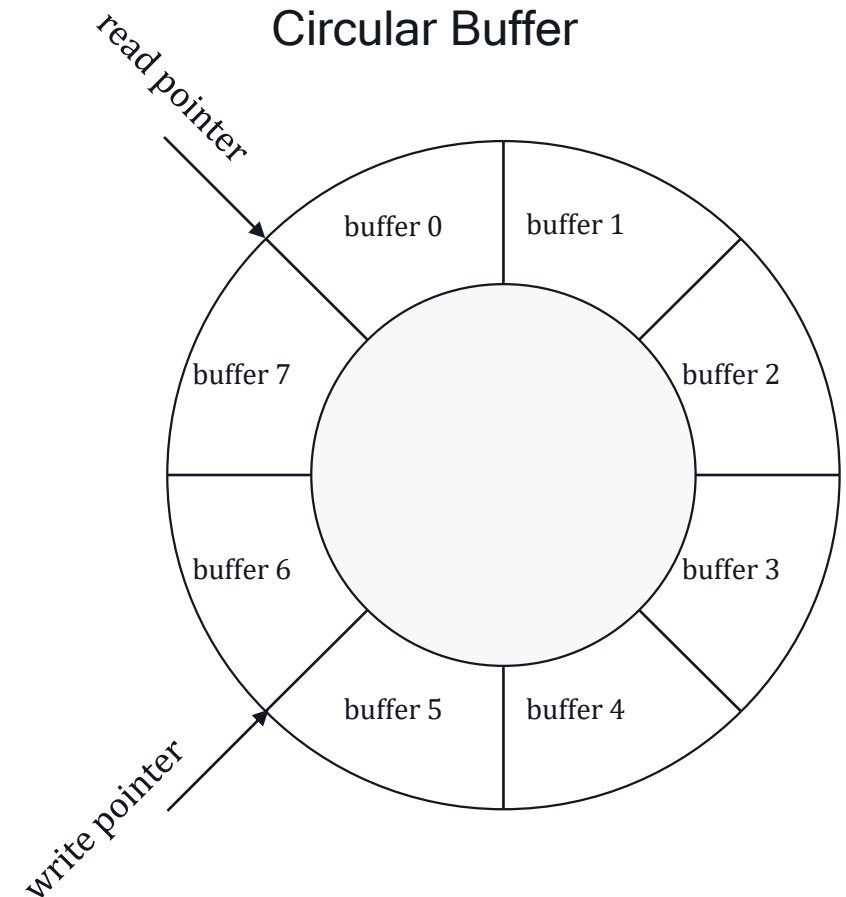
```
{
  "extensions": {
    "MPEG_media": {
      media: [
        {
          "name": "source 0",
          "renderingRate": 25.0,
          "timeOffset": 0.0,
          "autoplay": "true",
          "loop": "true",
          "alternatives": [
            {
              "mimeType": "application/dash+xml",
              "uri": "https://www.foo.com/manifest.mpd",
              "tracks": [
                {
                  "track": "#track=1"
                }
              ]
            }
          ]
        }
      ]
    }
  }
}
```

Describing how to access data

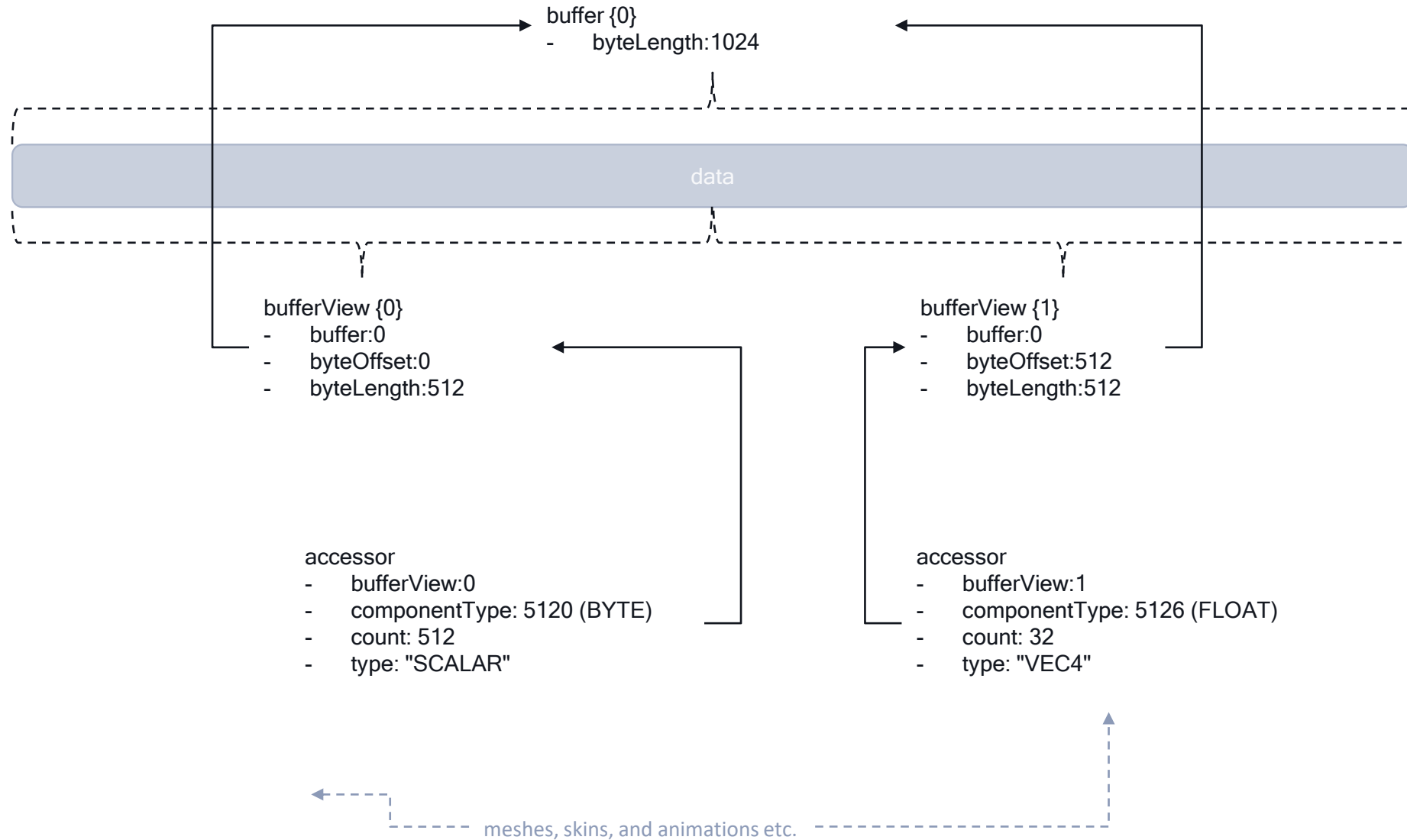


MPEG_accessor_timed and MPEG_buffer_circular extensions

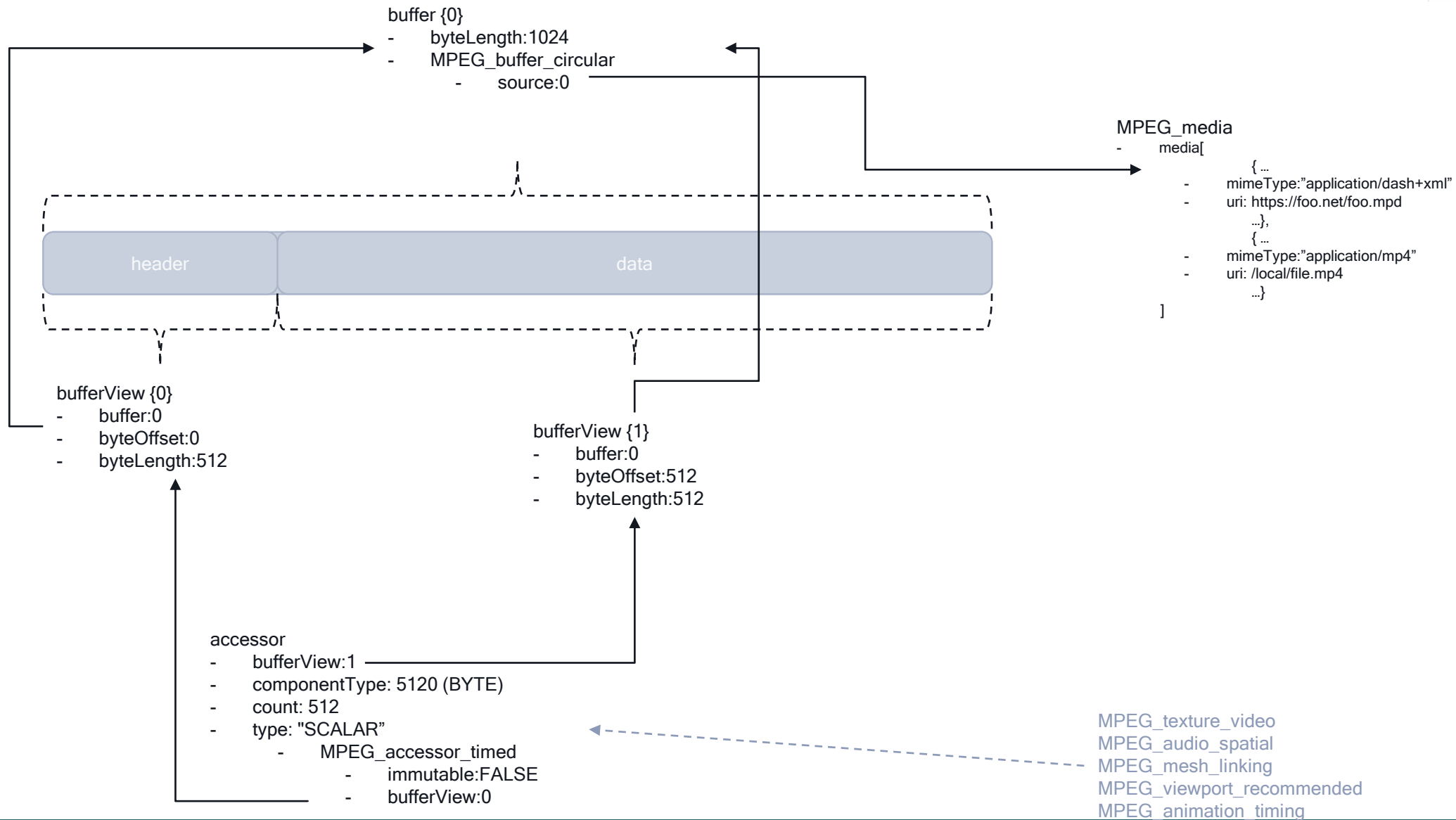
- glTF accesses data through accessors
 - They define the components of the data and their data types (e.g. a VEC3 of floats)
 - Semantics are provided by the referencing attribute/property (e.g. position)
 - The **accessor** points into a **bufferView**, which defines how the data is packed in the referenced **buffer**
 - No support for timed data
- MPEG_accessor_timed
 - Extension to accessor
 - Used to access all types of dynamic and timed media (audio, visual, volumetric, ...)
 - Backwards compatible: in case of no support, fallback to static data
- MPEG_buffer_circular
 - Extension to buffer
 - Dynamic variable-size swap chain buffer for exchange of media data for rendering
 - Acts as the interface between the Presentation Engine and MAF. All requested data through MAF API is delivered through a Buffer or Circular Buffer
 - Header is used to propagate metadata such as timestamps
 - Circular Buffer references MPEG_media



Static Buffer Usage



Dynamic Buffer Usage



Timed accessor header information

- Mutable Information in Buffer View and Accessor
- Accessor information that may change over time
 - componentType
 - bufferView
 - type
 - normalized
 - byteOffset
 - count
 - max
 - min
- bufferView information that may change over time
 - bufferViewByteOffset
 - bufferViewByteLength
 - bufferViewByteStride

Video Textures



MPEG_texture_video extension

- Materials in the scene may make use of textures
- A texture in glTF 2.0 only supports references to images of format JPEG or PNG
- MPEG_texture_video adds support for dynamic textures such as atlases
 - Point into a timed accessor instead of an image
 - Keep the image pointer as fallback
 - To support dynamic atlases, texture coordinates themselves are dynamic and fed through a timed accessor as well

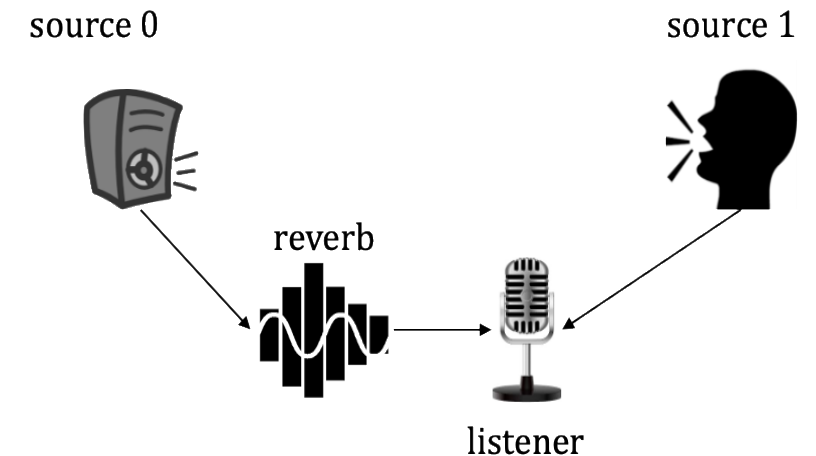
```
{
  "textures": [
    {
      "sampler": 0,
      "source": 1,
      "extensions": {
        "MPEG_texture_video": {
          "accessor": 2,
          "width": 2048,
          "height": 2048
        }
      }
    }
  ]
}
```

Spatial Audio



MPEG_audio_spatial extension

- glTF has no support for audio
- The MPEG_audio_spatial extension:
 - Audio Sources can be coupled to visual nodes to share the same transformations
 - Supports 3 types of nodes:
 - Audio Source: emits audio signals. Simple mono and HOA sources are supported
 - Audio Effect: a reverb zone effect is currently supported
 - Audio Listener: provides the position of the listener
 - The Audio Listener may be linked to the scene camera to allow for an immersive spatial experience. The listener will move together with the camera.
 - Actual rendering is not defined.
 - It is up to the Audio Rendering Engine to convert the signals that are received at the audio listener into a format that matches the actual speaker setup.
 - For example, binauralization is done for users wearing an HMD.

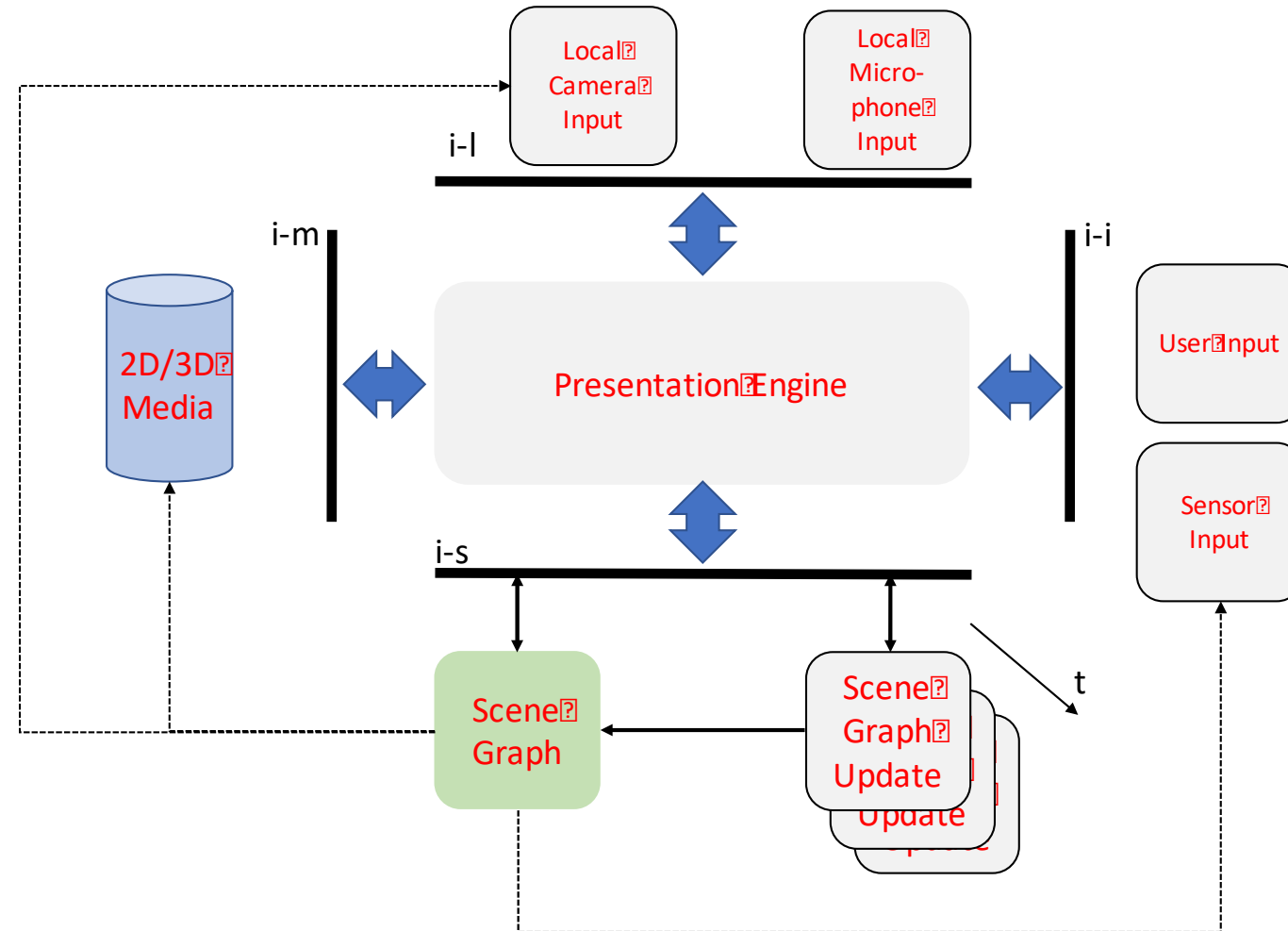


Other extensions

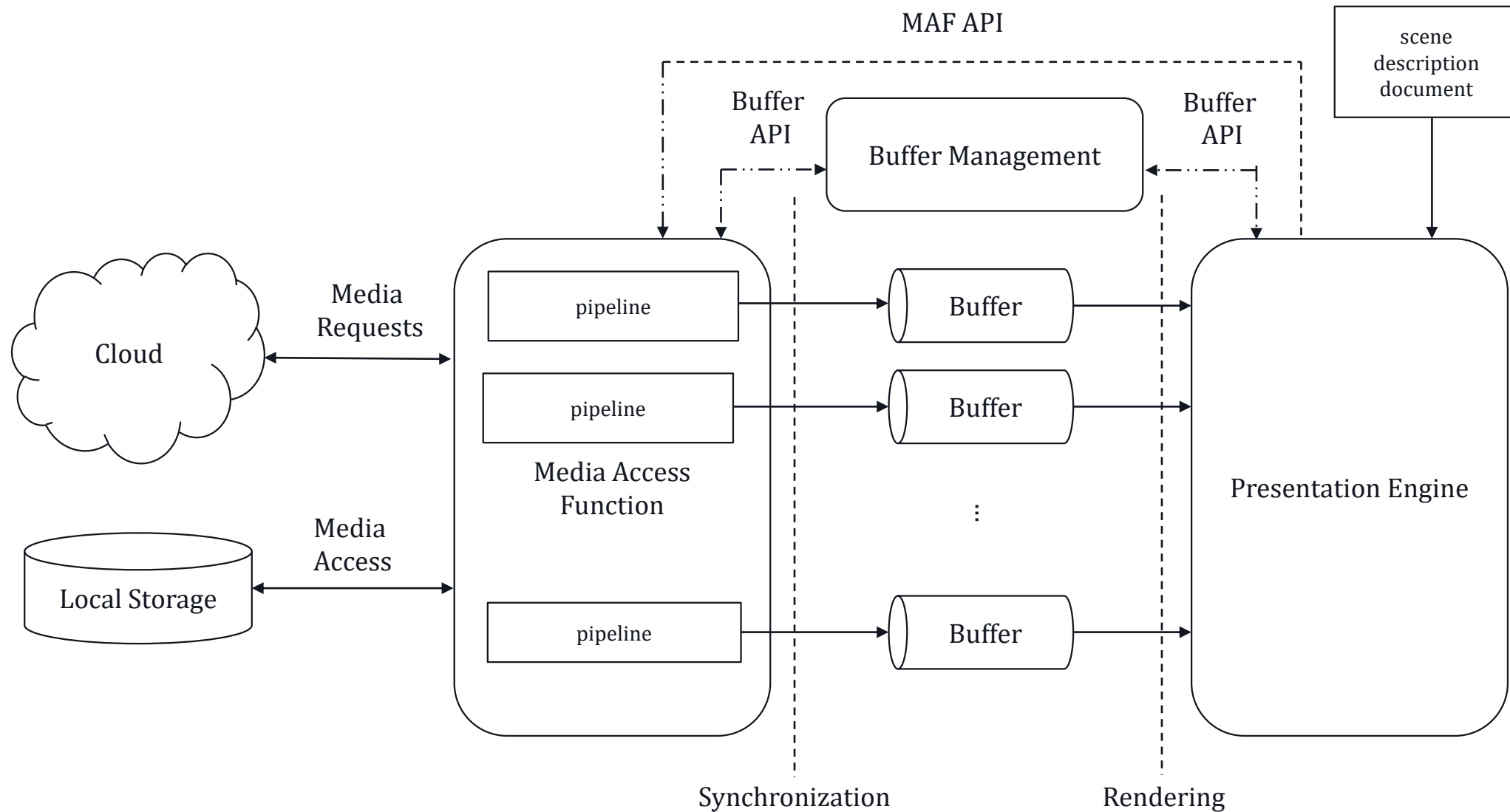


- **MPEG_viewport_recommended**
 - provides dynamically changing information which includes translation and rotation of the node which includes the camera object, as well as the intrinsic camera parameter of the camera object.
- **MPEG_animation_timing**
 - provides alignment between MPEG media timelines and animation timeline defined by glTF 2.0
- **MPEG_mesh_linking**
 - provides the possibility to link two meshes in a glTF asset, so called dependent and shadow meshes. The extension links the dependent mesh and the shadow mesh and provides with the data and information which is used to achieve the ability to animate the dependent mesh. The shadow mesh is present in the glTF assets to assist in achieving the ability to apply transformation onto the dependent mesh.
- **MPEG_scene_dynamic**
 - Provides the possibility to indicate that the scene description document will be updated
 - Updates are provided through 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

Scene Description in MPEG-I Architecture



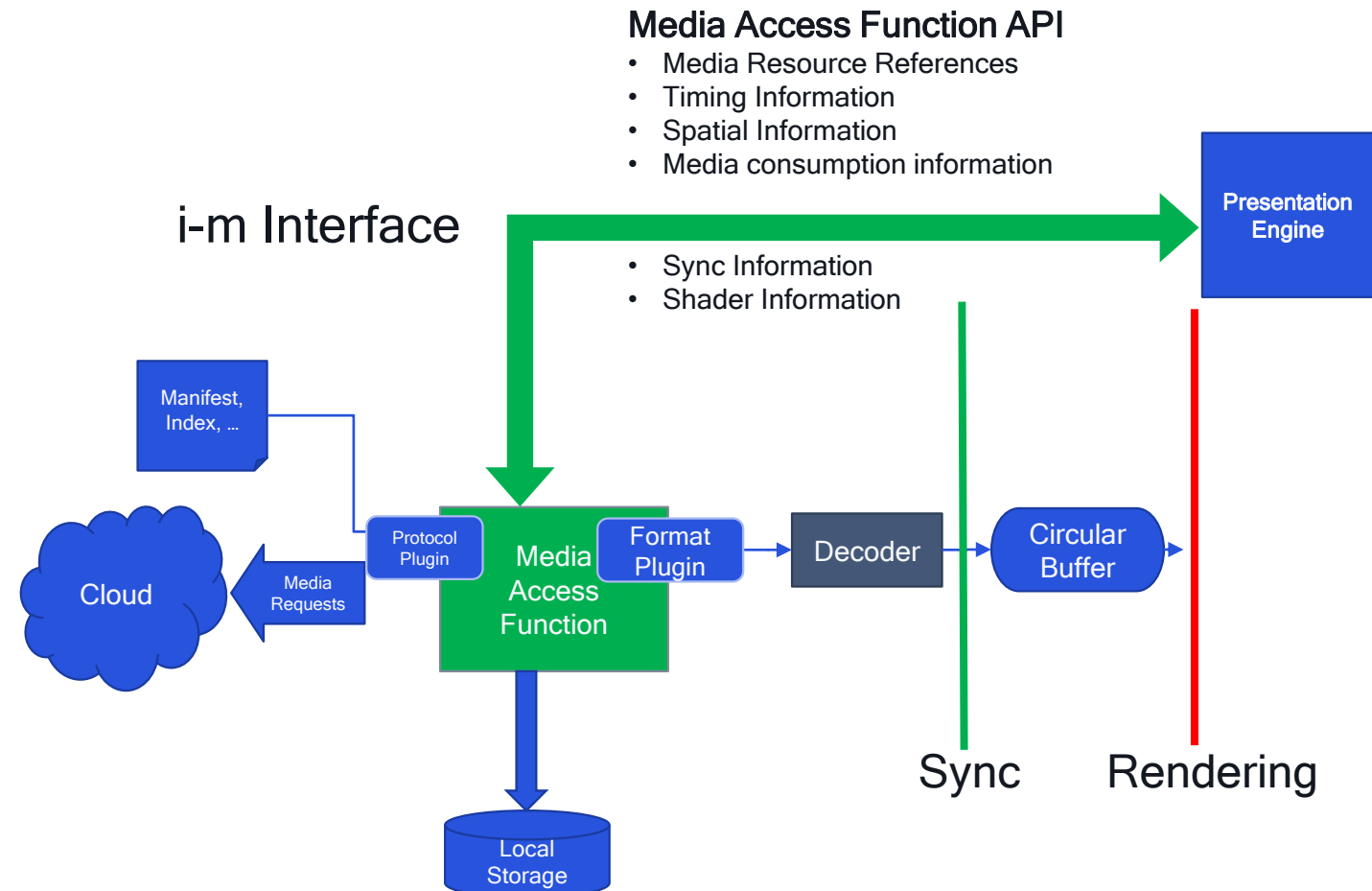
MPEG-I Scene Description Architecture



Media Access Function (MAF) API



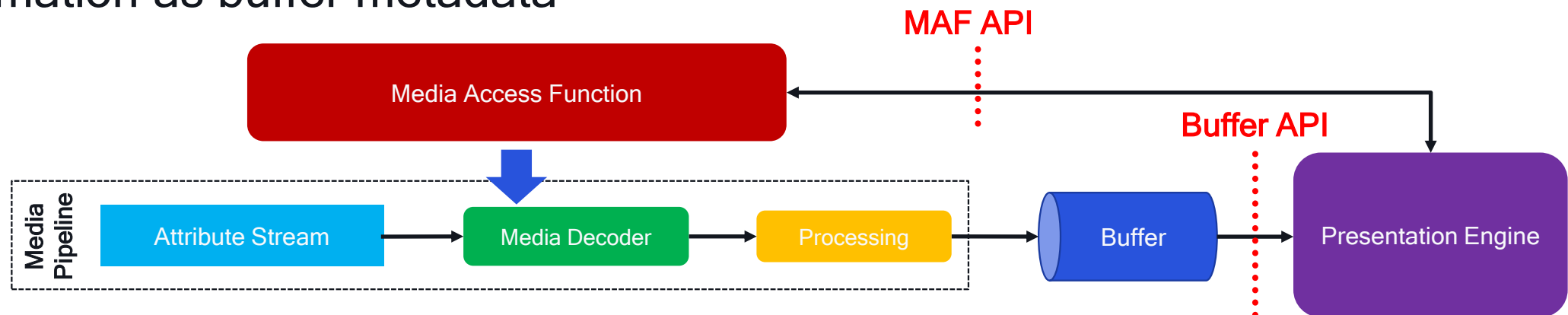
- Support for wide range of formats through Plugins
- Endpoint for the Media Access Function API
- Optimized Media Fetching
 - Random spatial and temporal access
 - Partial delivery matching Presentation Engine needs
 - Integration with Cloud and Edge media processing



Media Pipelines



- The MAF instantiates and manages Media Pipelines
 - A media pipeline typically handles content of an attribute/component of an object/mesh
 - It produces content in the format indicated by the glTF file
 - The formatted frame is then pushed into the circular buffer
- Media Pipelines are highly optimized and customized for the type and format of media that is being fetched
- Media Pipeline maintains sync information (time and space) and passes that information as buffer metadata





Thank you

For more information, visit:

www.mpeg-sd.org