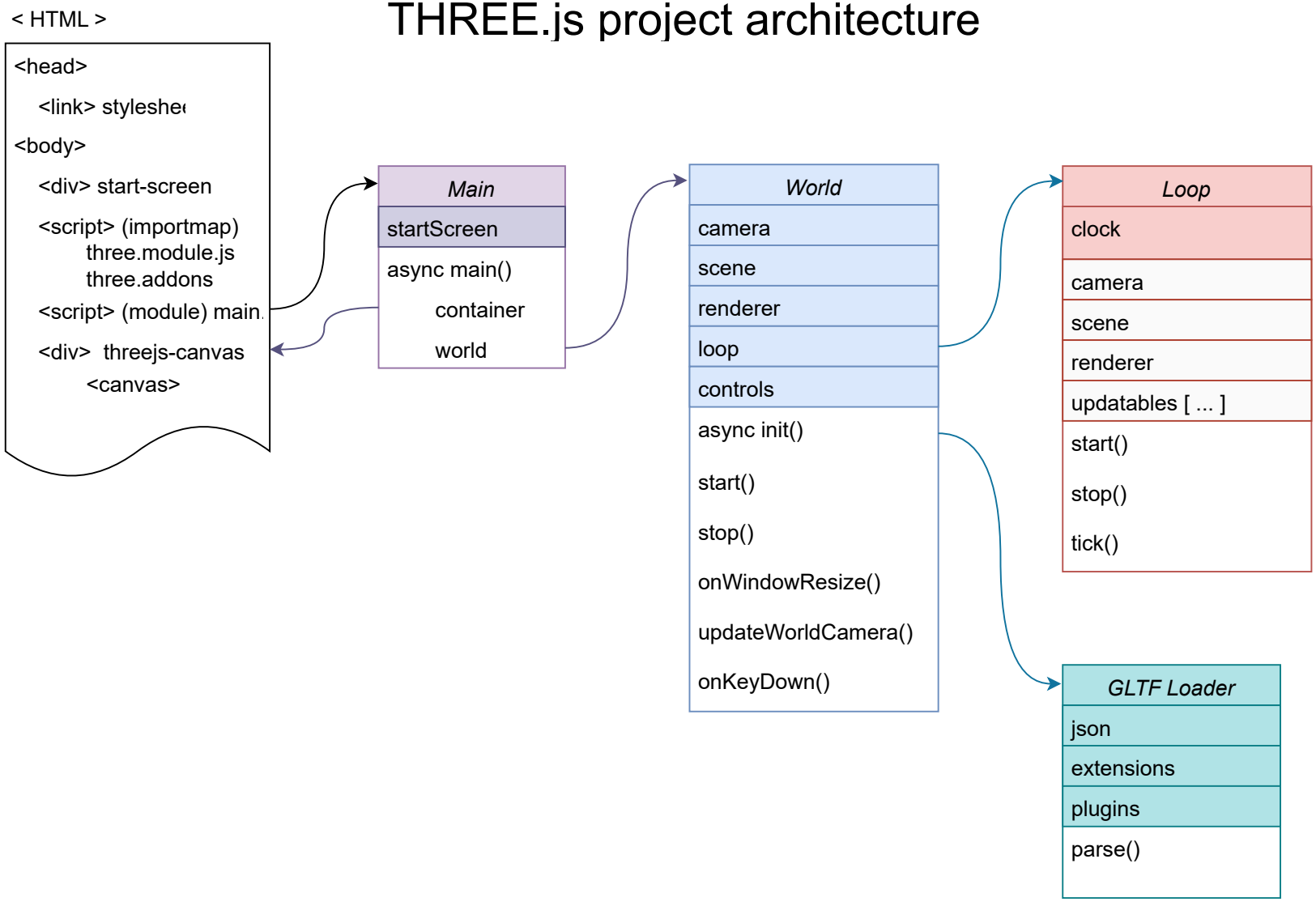
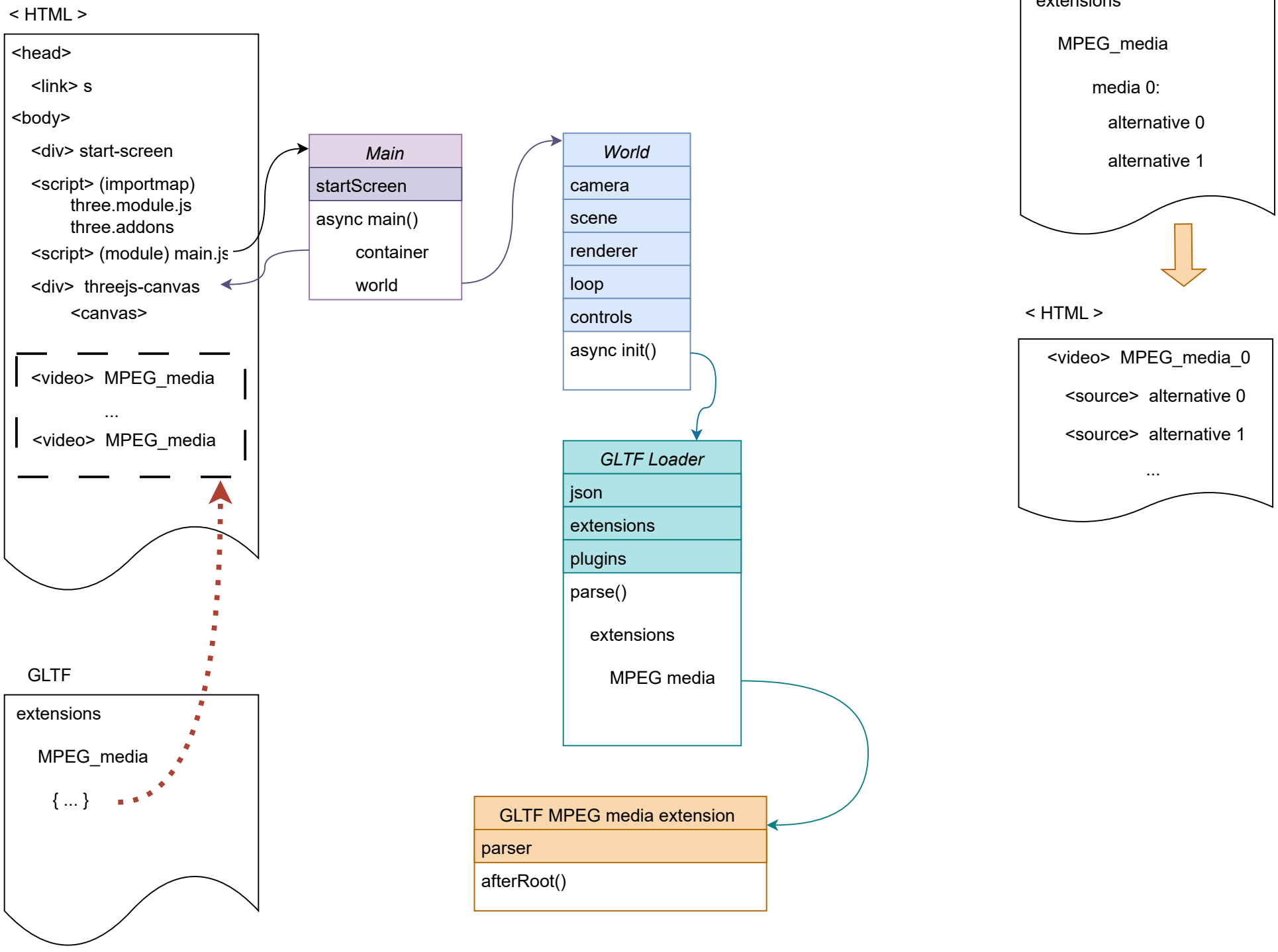
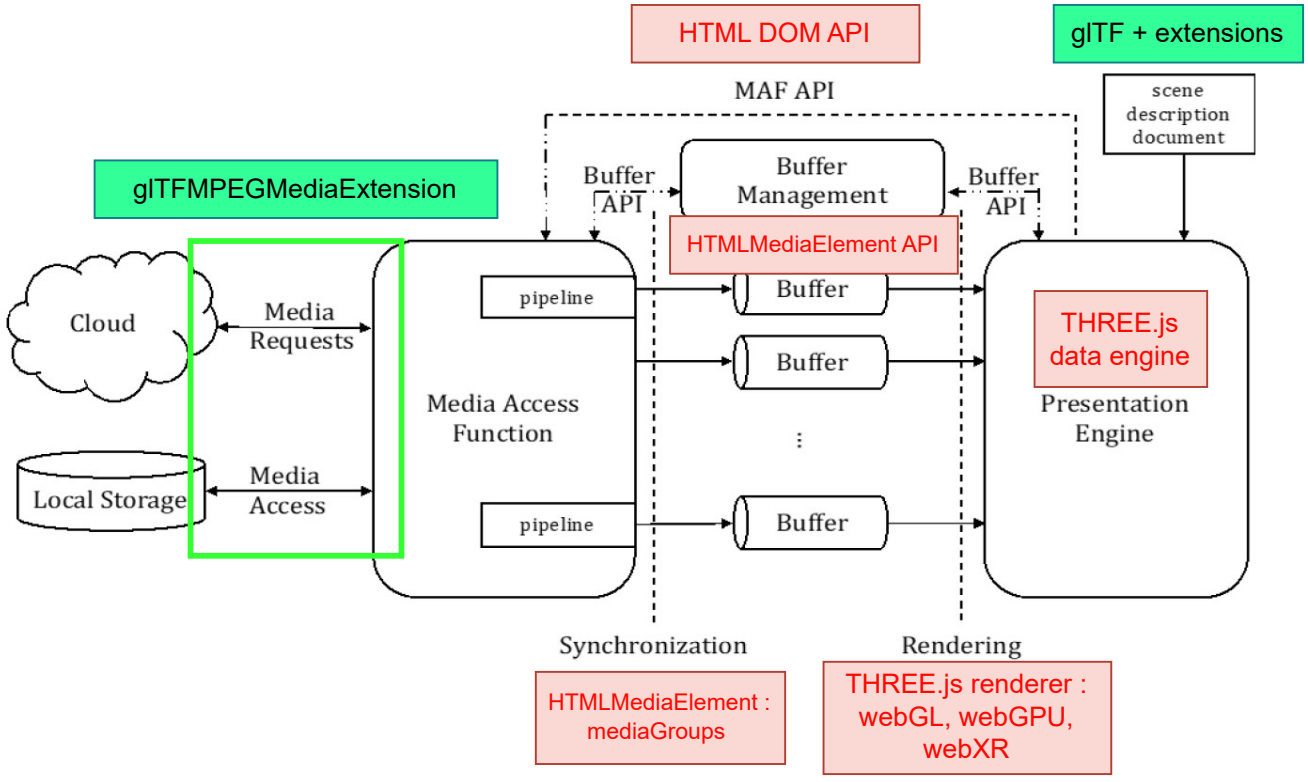
THREE.js project architecture



MPEG_media extension



in the context of the MPEG-I SD Architecture...



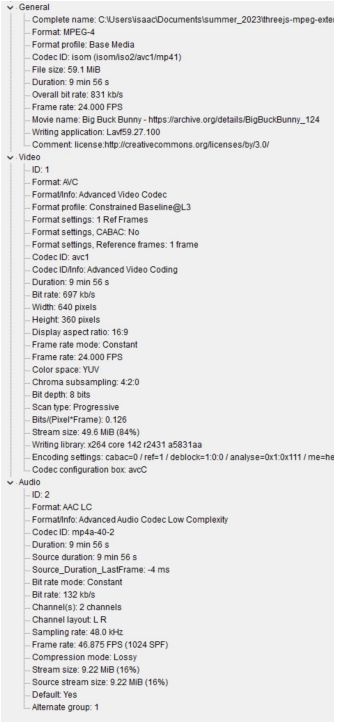
MPEG-I SD three.js extensions

existing infrastructure

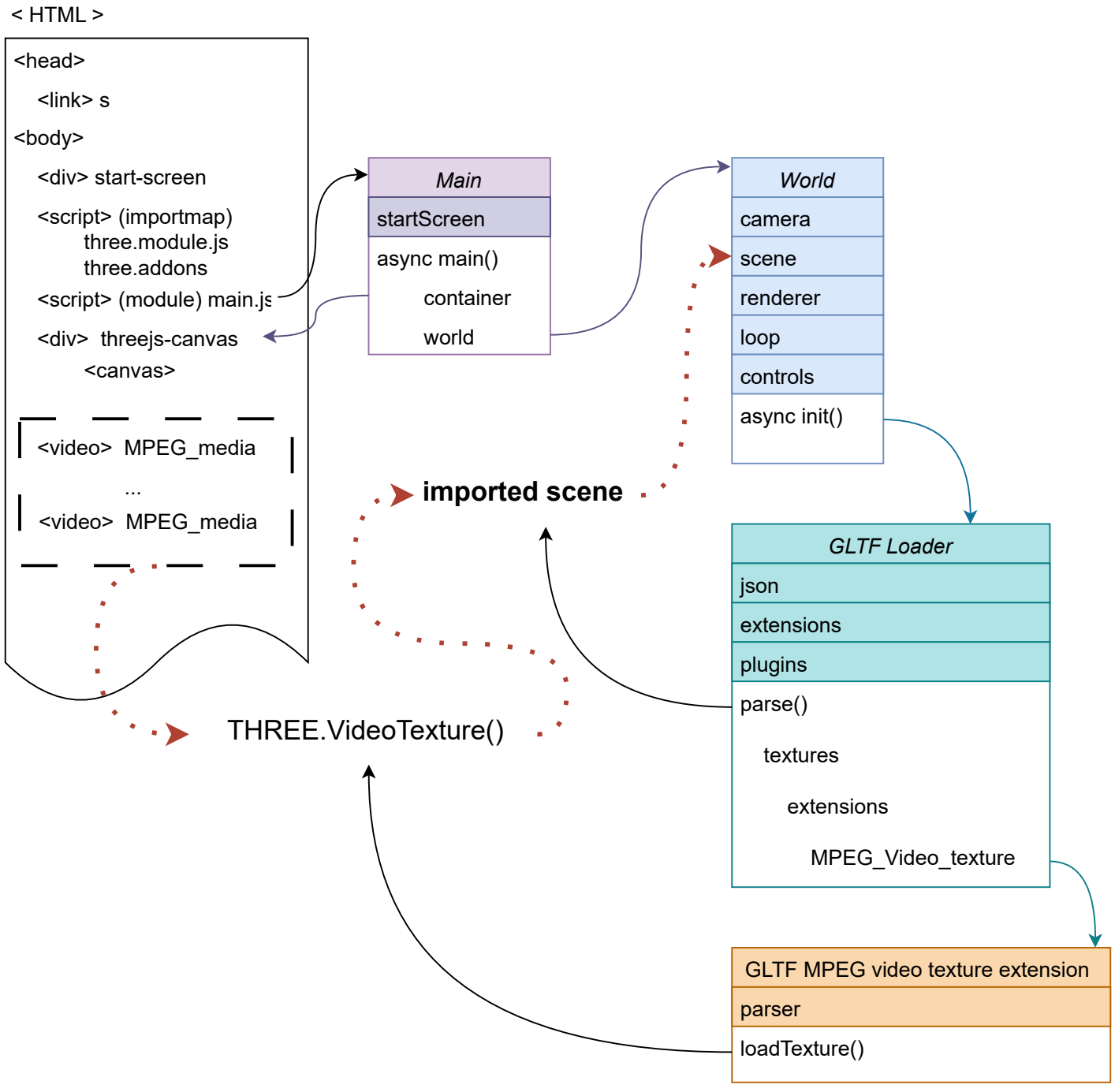
THREE.js mappings for MPEG-I SD media:

- Media Requests / Media Access:
  - handled by our MPEG_media THREE.js extension.
  - asynchronous requests / loads media referenced in the MPEG-I SD glTF extensions.
  - checks for playback compatibility (codec support) and embeds media.
- Media Access Function / MAF API:
  - access to MPEG-I SD media is managed via the DOM and the corresponding DOM API.
  - provides seamless integration with THREE.js VideoTexture and PositionalAudio components
  - permits bidirectional communication between "raw" media and the presentation engine (i.e. rendering, glTF export)
- Buffer Management / Buffer API:
  - individual buffers are managed via the HTMLMediaElement API. Specifically the <video> element and Media Source Extentions (MSE)
  - closely aligned with MPEG-I SD spec (referenced in the paper!)
  - allows the presentation engine to synchronize subsets of the requested media (via MediaGroups)
  - permits the user / presentation engine to specify playback at runtime (i.e. play, stop, seek)
- Presentation Engine
  - we use a THREE.js 'scene' to embed and render an immersive scene in the browser

media header ?



MPEG_texture_video extension (with embedded MPEG_media)



MPEG_audio_spatial extension (with embedded MPEG_media)