






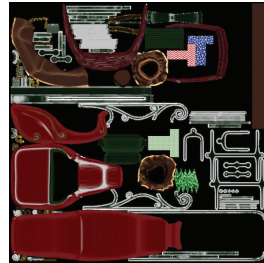
The Runtime Asset Format for GL-based Applications

July 2016

3D Needs a Transmission Format!

Audio	Video	Images	3D
MP3	H.264	JPEG	glTF™
 napster.			!

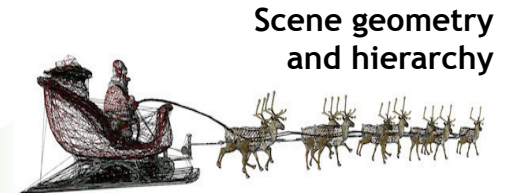
A widely adopted media format ignites previously untapped commercial opportunities



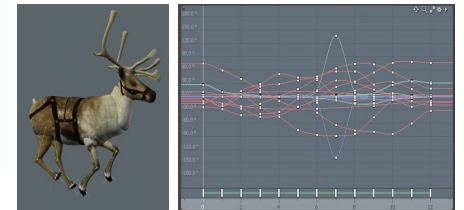
Materials and textures



glTF is used for NORAD's Santa Tracker Web Page



Scene geometry and hierarchy



Animations and skins

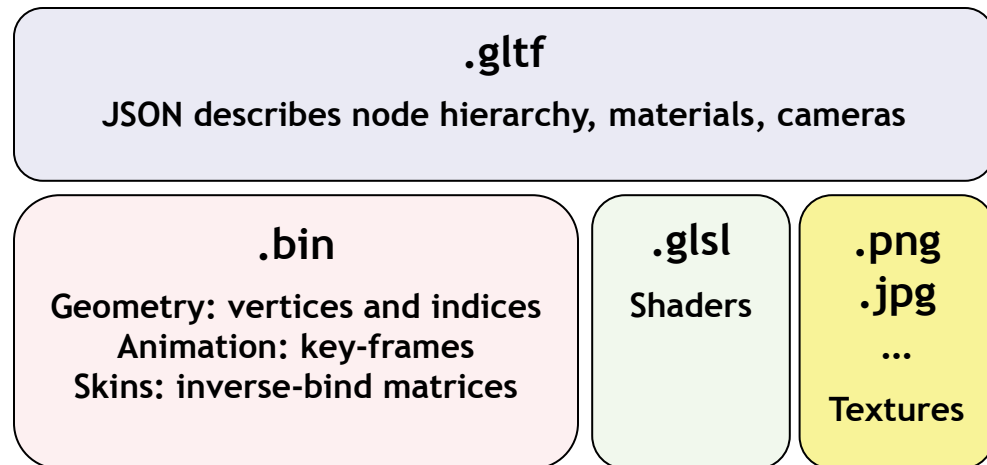
glTF = “JPEG for 3D”

- ‘GL Transmission Format’
 - Runtime asset format for WebGL, OpenGL ES, and OpenGL applications
- **Compact representation for download efficiency**
 - Binary mesh and animation data
- **Loads quickly into memory**
 - GL native data types require no additional parsing
- **Full-featured**
 - 3D constructs (node hierarchy, materials, animation, cameras, lights)
- **Runtime Neutral**
 - Can be created and used by any tool, app, or runtime
- **Flexible Extensibility**
 - E.g. payloads with compression and streaming



glTF Internals

- JSON describes node hierarchy
 - Includes cameras
 - References geometry, animations, skins, shaders, textures
- Vertices
 - Uses native typed array format
 - Includes key-frame animations and skinning
- Shaders
 - With extensions for materials
- Textures
 - Use existing standard image compression formats e.g. JPEG
- Extras
 - For app-specific data (metadata)



glTF Example

JSON Node (the truck)
with three children (sets of two wheels)

Visualization
of Node
Hierarchy

The screenshot displays the AGI Model Inspector interface. On the left, a 'Hierarchy' panel shows a tree structure with nodes: 'defaultScene', 'polyRenGeometry-mesh001Node', 'Geometry-mesh002Node', 'meshInst004Node', and 'meshInst009Node'. A red circle highlights this hierarchy. In the center, a JSON node definition is shown:

```
"Geometry-mesh002Node": {  
  "children": [  
    "Geometry-mesh001Node",  
    "meshInst004Node",  
    "meshInst009Node"  
  ],  
  "matrix": [  
    1,  
    ...  
  ]  
}
```

 A red circle highlights this JSON node. On the right, an 'Animation Player' panel shows three animations: 'animation_0', 'animation_1', and 'animation_2'. A red circle highlights this panel. The main view shows a 3D model of a truck. At the bottom left, 'Key Statistics' are visible: 'Node: Geometry-mesh002Node', 'Number of Draw Calls: 5', and 'Number of Rendered Primitives: 7860'.

Three
animations -
one for each
set of wheels

glTF Ecosystem



Blender DIRECT export

Tools

Export

Drag and Drop FBX -> glTF
(coming soon)

<http://gltf.autodesk.io/>



AUTODESK
Autodesk FBX -> glTF
AssImp
OBJ2GLTF
glTF Pipeline
COLLADA2GLTF
Cesium converter



Drag and drop COLLADA -> glTF
<http://cesiumjs.org/convertmodel.html>

Translators

Convert | Optimize

Validator

Validate



Request for Quotations (RFQ) to create glTF Validator is open!



Engines

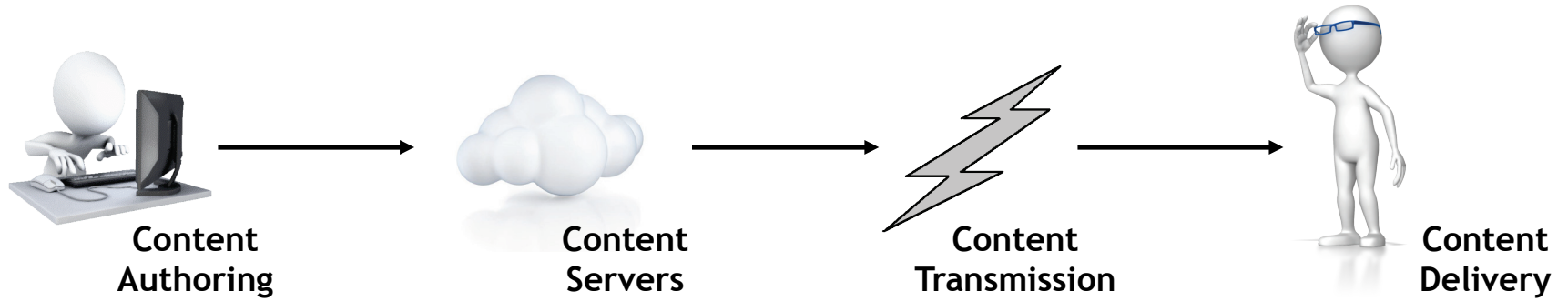
Import



glTF Ecosystem Page

<https://github.com/KhronosGroup/glTF#gltf-tools>

Khronos AR/VR Standards



WITHOUT Standards

Tools import/export custom 3D formats and so do not interoperate

Every service/app stores 3D assets in a custom format -> Silo'd content

Long download times and proprietary code to unpack received 3D assets

Apps have to be ported to each device and often don't use acceleration

WITH Standards

Mix and match tool pipelines through common 3D asset import/export

3D assets are easily understood and used by any application and device

3D assets packed into efficient formats with streaming and compression

APIs provide consistent access to graphics, compute and vision acceleration

Khronos standards useful for AR



AR/VR Will Need Many, Many Standards

3D is About to Go Social!

