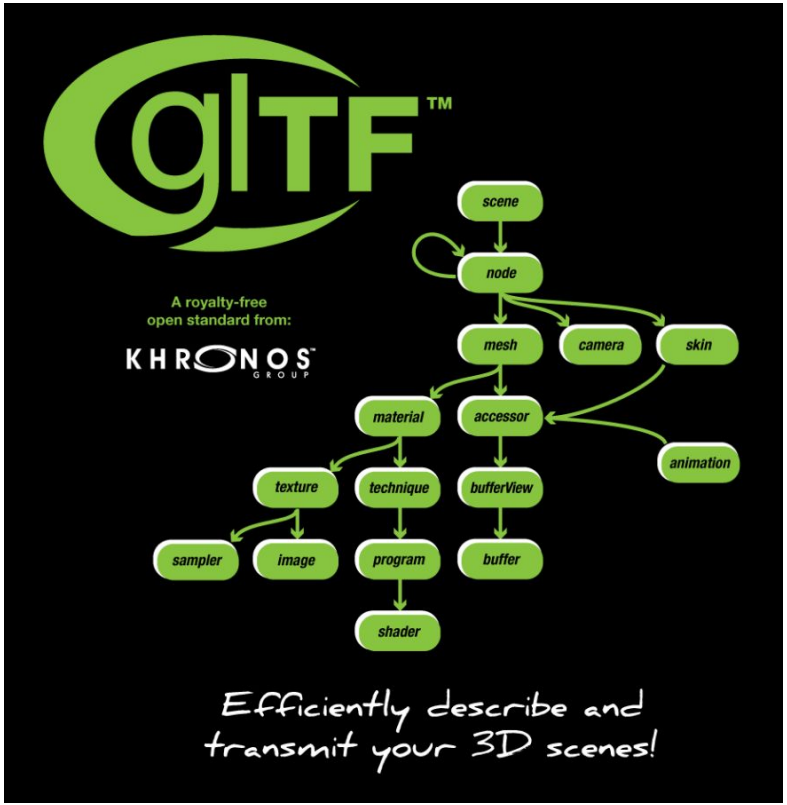


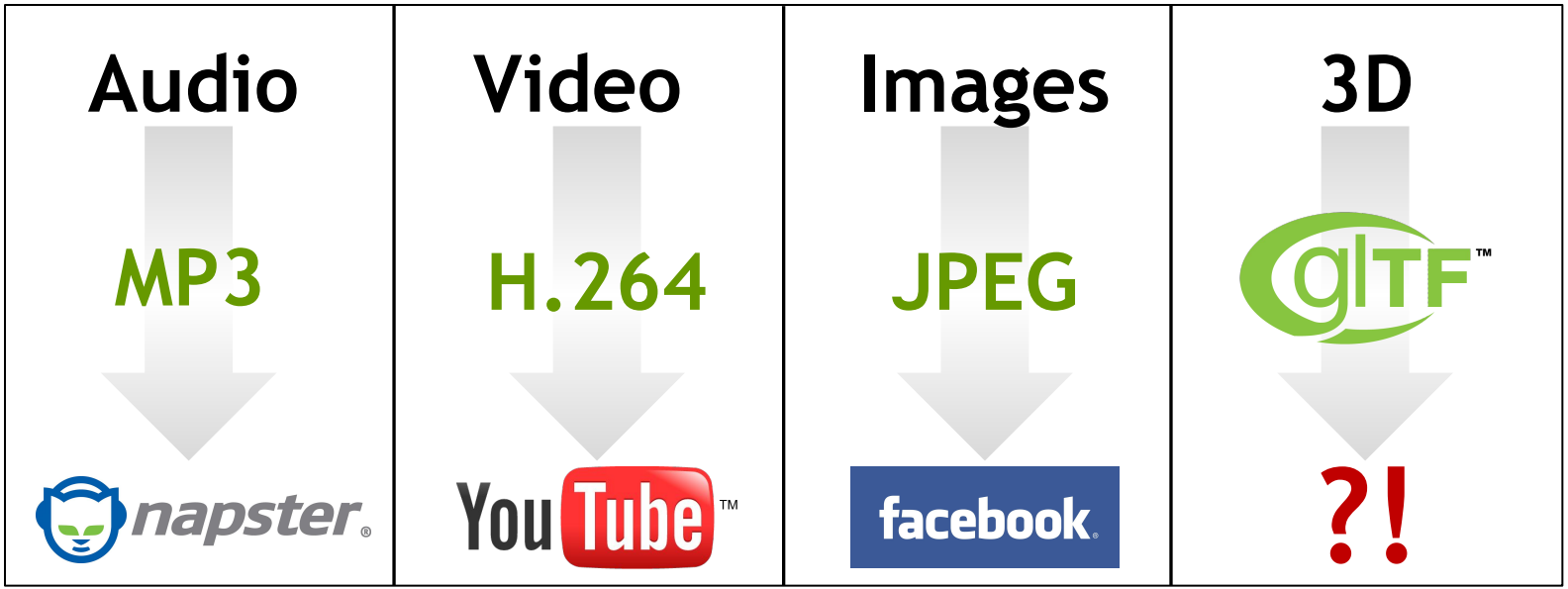
KHRONOS™ GROUP

glTF

Real-time 3D Asset Delivery

Tony Parisi, FormVR
glTF Specification Co-editor
Amanda Watson, Oculus
glTF WG member

3D: The Only Major Media Type Without a Standardized Delivery Format



...until now.

Compact to Transmit 

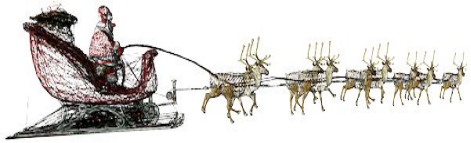
Fast to Load 

Describes Scenes 

Runtime Neutral 

Extensible 

glTF 1.X Scene Description Structure



.gltf

JSON describes node hierarchy, materials, cameras

.bin

Geometry: vertices and indices
Animation: key-frames
Skins: inverse-bind matrices

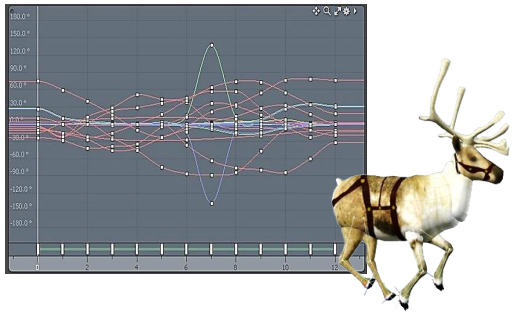
.glsl

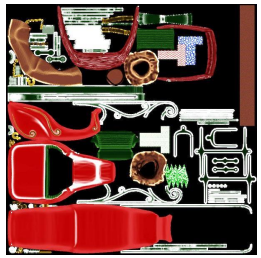
Shaders

.png
.jpg
...

Textures

NORAD's Santa Tracker





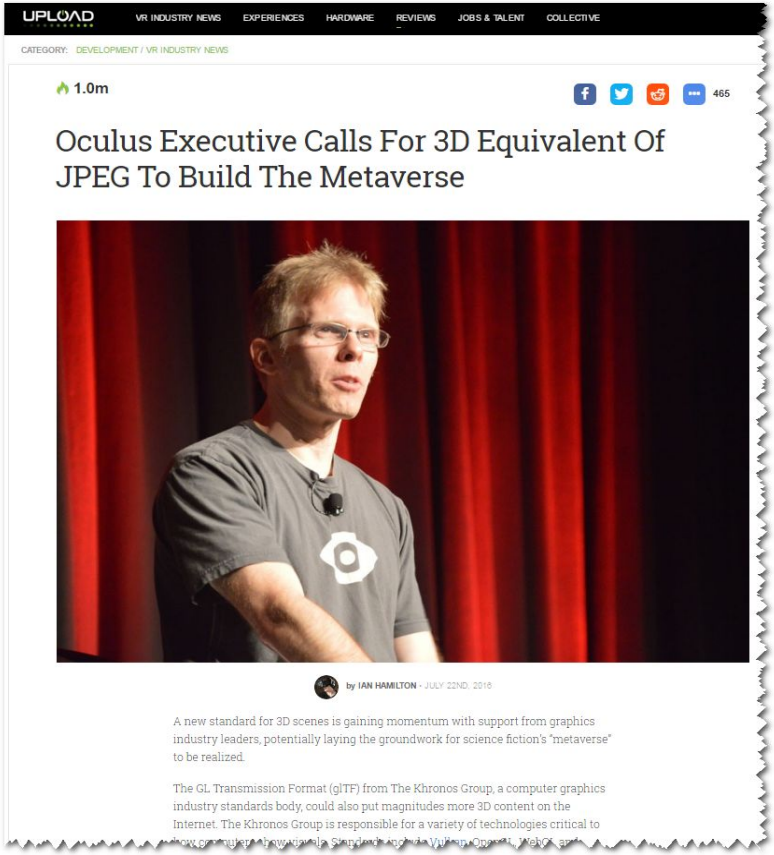
Timeline So Far

Original motivation:
standardized way to deliver
3D into WebGL applications

Changed new spec version from 1.0.1 to 1.1 as includes some simplifications e.g. animation

Will ratify V 1.1 when specification is proven by loader implementations

Strong Momentum



UPLOAD VR INDUSTRY NEWS EXPERIENCES HARDWARE REVIEWS JOBS & TALENT COLLECTIVE

CATEGORY: DEVELOPMENT / VR INDUSTRY NEWS

1.0m 465

Oculus Executive Calls For 3D Equivalent Of JPEG To Build The Metaverse

by IAN HAMILTON - JULY 22ND, 2016

A new standard for 3D scenes is gaining momentum with support from graphics industry leaders, potentially laying the groundwork for science fiction's "metaverse" to be realized.

The GL Transmission Format (glTF) from The Khronos Group, a computer graphics industry standards body, could also put magnitudes more 3D content on the Internet. The Khronos Group is responsible for a variety of technologies critical to

glTF™

oculus

web3D CONSORTIUM

MPEG

Adobe

blender™

OGC

box

Microsoft

Fraunhofer

NVIDIA

AREA Augmented Reality for Enterprise Alliance

otoy

CESIUM

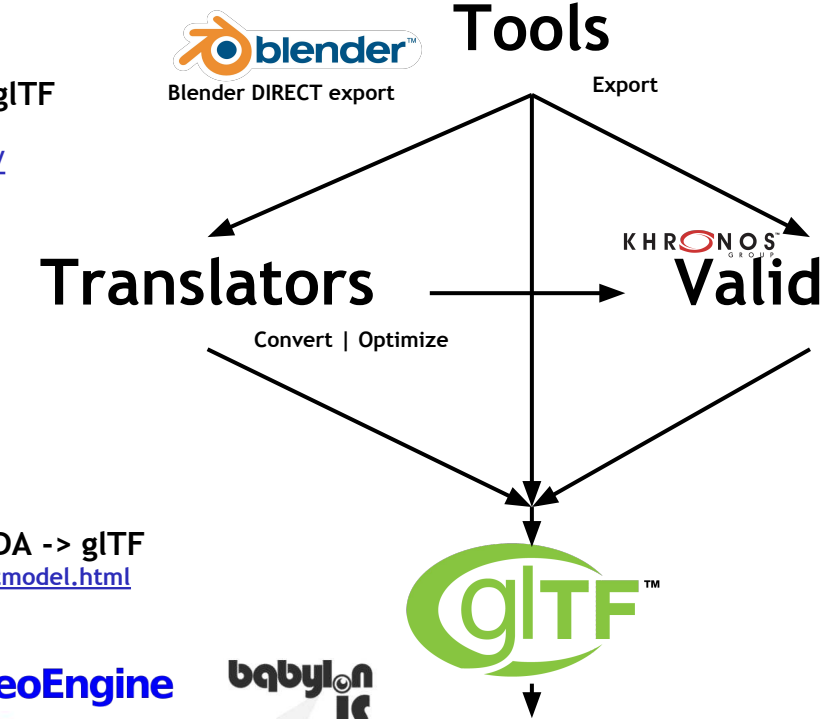
AUTODESK

Google

Publicly Stated Support for glTF

KHRONOS GROUP

glTF Ecosystem



Drag and Drop FBX -> glTF
(coming soon)
http://gltf.autodesk.io/

AUTODESK
Autodesk FBX -> glTF

blender
Blender DIRECT export

AssImp
OBJ2GLTF
glTF Pipeline
COLLADA2GLTF
Cesium converter

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

model/gltf+json MIME type
Approved by IANA!

glTF 1.1 Spec in Review and
glTF Validator in open source!
http://github.khronos.org/glTF-Validator/

xeoEngine
babylon JS
A-FRAME
three.js
NVIDIA
nvpro-pipeline
PEX
CESIUM

glTF Tools Page
https://github.com/KhronosGroup/glTF#gltf-tools

Oculus & glTF

"The world has long needed an efficient, usable standard for 3D scenes that sits at the level of common image, audio, video, and text formats... something at home on the internet, capable of being directly created and consumed by many different applications"

oculus

glTF™

models.json	.gltf
Ordered list of "surfaces"/materials	JSON describes node hierarchy, materials, cameras
models.bin	**.bin**
Geometry: vertices and indices (sorted, pre-filtered)	Geometry: vertices and indices Animation: key-frames Skins: inverse-bind matrices
.pvr, .ktx ... Textures	.glsl Shaders
	.png .jpg ... Textures

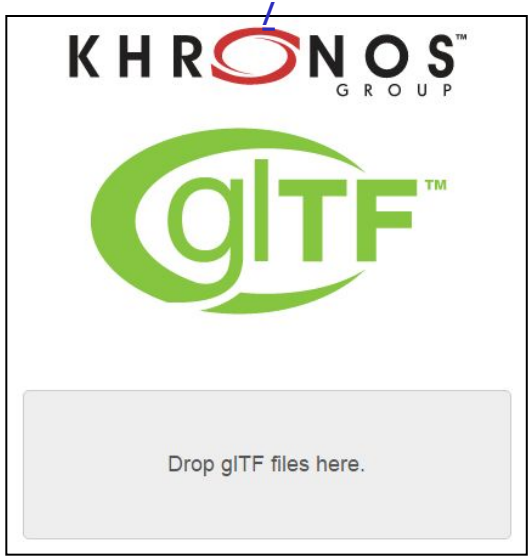
FBX2glTF: built with VR in mind

glTF 1.1 Validator

- **glTF 1.1 tightens specification**
  - For robust validation and interoperability
    https://github.com/KhronosGroup/glTF/issues/605
- **Validator in open source on GitHub**
  - Khronos Validator project RFQ awarded to Alexey Knyazev – doing awesome work!
  - Rigorous checking for correctly formed glTF files
  - Checks JSON syntax, all property details, GL parameter combinations etc. etc.
  - Built using Dart (easy API level integration)
  - Shipping today as client-side drag-n-drop and command-line wrapper
  - Client-side JavaScript library coming soon
  - Extensible – validation plugins for extensions – output can be integrated into the validation report

Please give us feedback on GitHub!

http://github.khronos.org/glTF-Validator/



‘glTF Next’

- **Targeting Draft Specification by Spring 2017**

- **Refactor so no API dependencies in core** *https://github.com/KhronosGroup/glTF/issues/733*
  - Extensions for API specifics: WebGL 2.0, Vulkan, DX12, Metal, GLSL, HLSL, SPIR-V, Metal C++
  - Compressed texture formats, compute?
- **Physically Based Rendering (PBR)** *https://github.com/KhronosGroup/glTF/issues/696*
  - Must be scalable from simple/fast to high-end/photorealistic renderers - Fraunhofer, NVIDIA MDL?
- **Format changes and improved efficiency** *https://github.com/KhronosGroup/glTF/issues/507*
  - BSPs, arrays, bounding boxes, spatial constructs, texture channels/usage, syntax cleanup
- **Animation enhancements - Morph targets, keyframes and interpolation types, compression**
- **Mesh Streaming - progressive mesh, quantization, LOD**

- **Metadata - for mixing advanced experiences – external spec**
- **Unstructured storage with accessors - Metadata to tie buffers to use cases**
- **Parametric Geometry and paths – OpenSubdiv**
- **Multipass – deferred rendering**
- **Mesh Compression - MPEG 3DGC (royalty-free), Fraunhofer SRC?**