KHRONOS GROUP

glTF Roadmap

CTTF Universal Textures and Second Generation PBR

Neil Trevett, Khronos President
NVIDIA VP Developer Ecosystems
ntrevett@nvidia.com | @neilt3d

W3C® Workshop on Web Games
Redmond, June 2019

Active Khronos Standards

HIGH PERFORMANCE 3D GRAPHICS

Vulkan, OpenGL ES, OpenGL, OpenGL SC, WebGL

Evolved OpenGL ES and WebGL from OpenGL. Vulkan cross-platform new generation GPU API. WebGL extensions in flight for compute shaders and multi-draw call batching

3D ASSET AUTHORING AND DELIVERY

glTF, COLLADA

Efficient, reliable, real-time transmission of modern 3D scenes and assets

PORTABLE XR – VIRTUAL AND AUGMENTED REALITY

OpenXR

Native cross-platform API for all XR functionality except rendering. Encouraging close cooperation with WebXR

PARALLEL COMPUTATION, VISION, MACHINE LEARNING AND INFERENCING

OpenCL, SYCL, SPIR, OpenVX, NNEF

Widely adopted Intermediate Representation for shaders and kernels

KHRONOS GROUP

Khronos is an open, member-driven industry consortium developing royalty-free standards, to harness the power of silicon acceleration for demanding graphics rendering and computationally intensive applications

glTF - The JPEG of 3D!

Audio	Video	Images	3D
MP3	H.264	JPEG	glTF
napster	YouTube	facebook	New market opportunities for 3D content creation and deployment!

glTF

- Compact to Transmit ✓
- Simple and Fast to Load ✓
- Describes Full Scenes ✓
- Runtime Neutral ✓
- Open and Extensible ✓

Efficient, reliable transmission
Bring 3D assets into 1000s of apps and engines

glTF

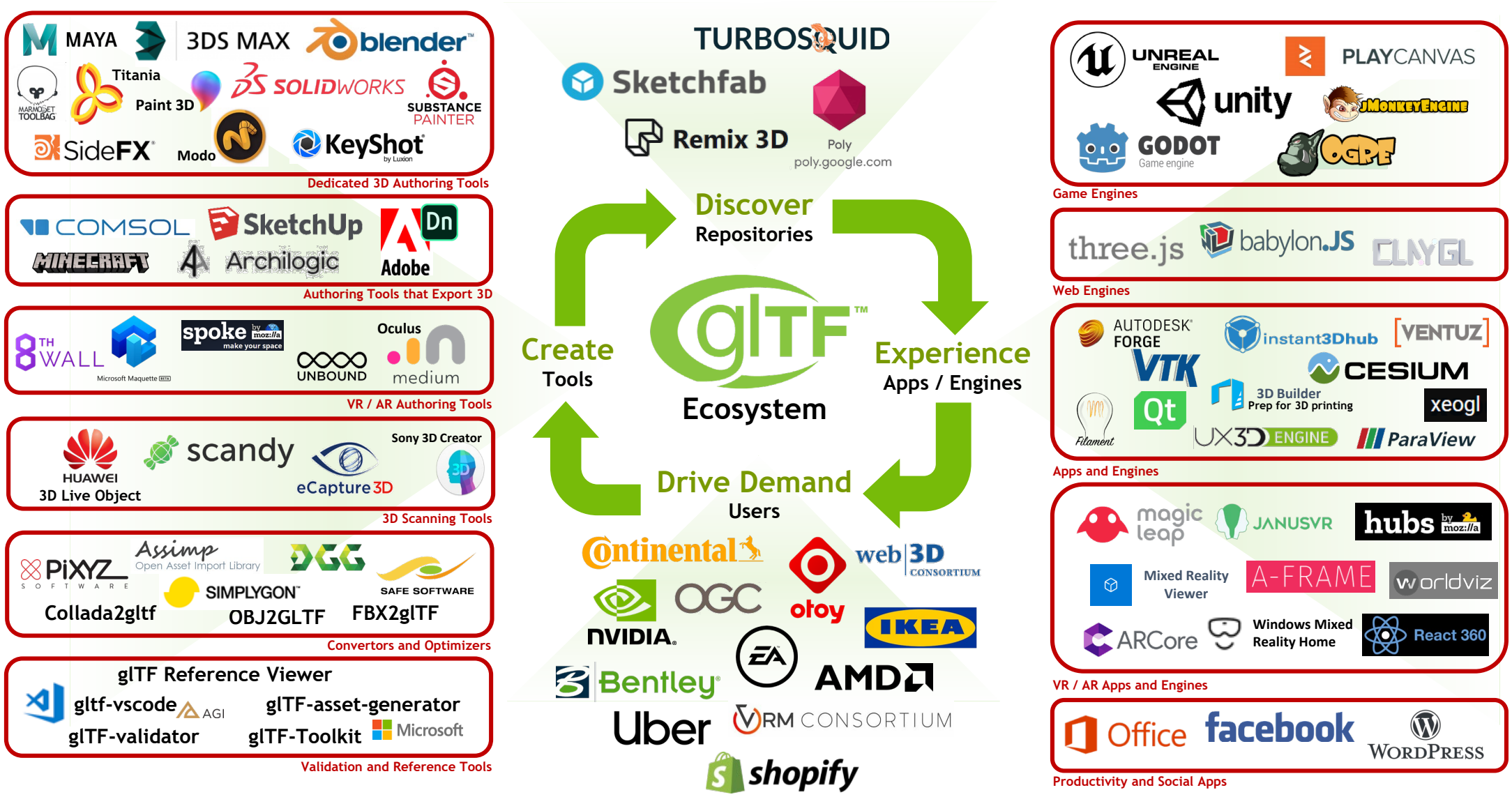
WebGL

glTF 1.0 - December 2015
Primarily for WebGL
Uses GLSL for materials

glTF spec development on open GitHub - get involved!
https://github.com/KhronosGroup/glTF

WebGL DirectX 12 Vulkan

glTF 2.0 - June 2017
Native AND Web APIs
Physically Based Rendering
Metallic-Roughness and Specular-Glossiness


MAYA
3DS MAX
blender
Titania
Paint 3D
SOLIDWORKS
SUBSTANCE PAINTER
SideFX
Modo
KeyShot
Dedicated 3D Authoring Tools
COMSOL
SketchUp
MINECRAFT
Archilogic
Adobe
Authoring Tools that Export 3D
8TH WALL
Microsoft Maquette
spoke
Oculus
UNBOUND
medium
VR / AR Authoring Tools
HUAWEI
3D Live Object
scandy
Sony 3D Creator
eCapture 3D
3D Scanning Tools
PiXYZ
Assimp
Open Asset Import Library
SIMPLYGON
SAFE SOFTWARE
Collada2gltf
OBJ2GLTF
FBX2glTF
Convertors and Optimizers
glTF Reference Viewer
gltf-vscode
AGI
gltf-asset-generator
glTF-validator
glTF-Toolkit
Microsoft
Validation and Reference Tools
TURBOSQUID
Sketchfab
Remix 3D
Poly
poly.google.com
Discover
Repositories
Create
Tools
glTF™
Ecosystem
Experience
Apps / Engines
Drive Demand
Users
Continental
web 3D CONSORTIUM
OGC
otoy
NVIDIA
IKEA
Bentley
AMD
Uber
VRM CONSORTIUM
shopify
UNREAL ENGINE
PLAYCANVAS
unity
JMonkeyEngine
GODOT
Game engine
OGRE
Game Engines
three.js
babylon.JS
CLAYGL
Web Engines
AUTODESK FORGE
instant3Dhub
VENTUZ
VTK
CESIUM
3D Builder
Prep for 3D printing
xeogl
Filament
Qt
UX3D ENGINE
ParaView
Apps and Engines
magic leap
JANUSVR
hubs
Mixed Reality Viewer
A-FRAME
worldviz
ARCore
Windows Mixed Reality Home
React 360
VR / AR Apps and Engines
Office
facebook
WordPress
Productivity and Social Apps

glTF 2.0 Scene Description Structure

.gltf (JSON)

Node hierarchy, PBR material textures, cameras

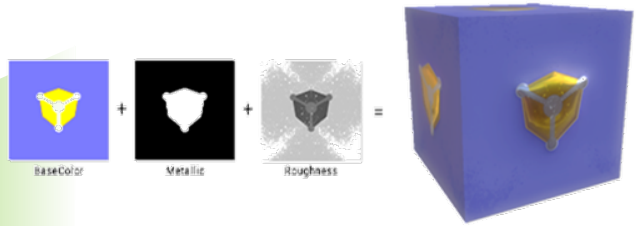
.bin
Geometry: vertices and indices
Animation: key-frames
Skins: inverse-bind matrices

.png
.jpg
...
Textures

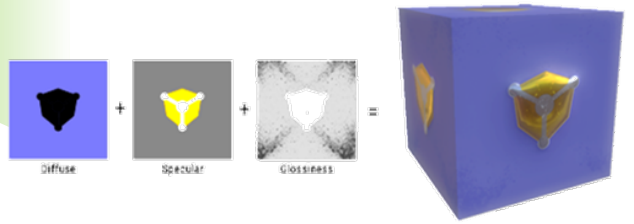
Geometry

Texture based
PBR materials

Mandatory Metallic-Roughness Materials



Optional Specular-Glossiness Materials



glTF Ecosystem Evolution

glTF 2.0 import/export
with Blender 2.8 Beta

Tools!

Striving for native glTF import and Export from every tool. Catalyzed Blender IO as exemplar

glTF 2.0 - June 2017

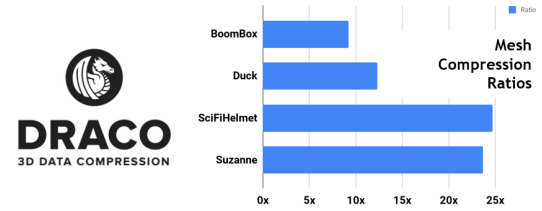
Consistency!

Avoid dialects at all costs! Sample viewer and Asset Validator in open source. Sample models and asset generator for unit tests

Sample Viewer for accurate
Ground Truth glTF renderings

Functionality!

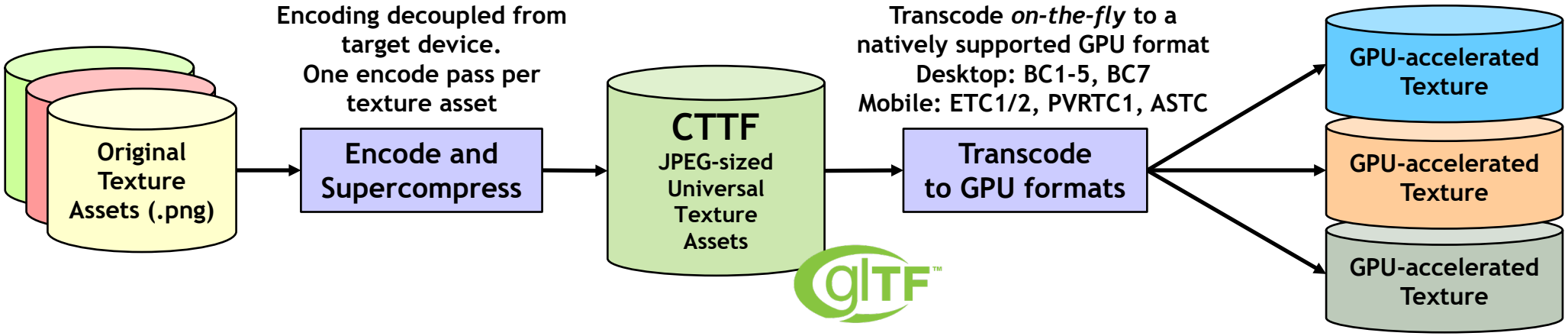
Balancing functionality versus complexity. glTF is extensible - only bring widely adopted extensions into core



glTF Mesh compression extension
provides up to 25x geometry compaction

CTTF Universal Textures for glTF

- **Fragmentation of GPU texture formats is significant issue for developers**
  - Ideally one compact set of texture assets could be natively accelerated everywhere
- **Binomial's 'Basis Universal' format contributed to glTF to enable CTTF Universal Textures**
  - 'Compressed Texture Transmission Format'
- **Binomial and Google recently open sourced 'Basis Universal' compressor and transcoder**
  - C++ and WebAssembly for native and Web stacks - it *already* works in all browsers
  - https://github.com/binomialLLC/basis_universal
- **CTTF extension for glTF will use KTX2 as a container format**
  - github.com/KhronosGroup/CTTF-Specification

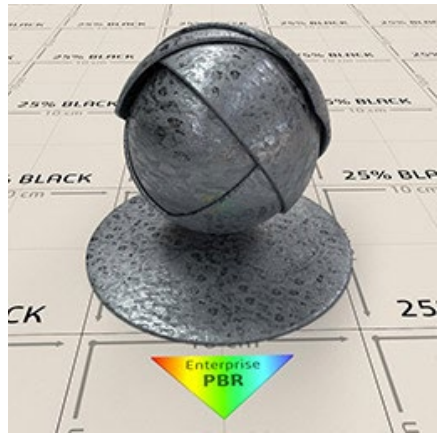


Next Generation glTF PBR Materials

- **Demand for advanced PBR for photorealistic assets**
  - Beyond current ‘Metallic-Roughness’ and ‘Specular-Glossiness’
  - E.g. Absorption/attenuation, clear coat, subsurface scattering, anisotropy
- **Extending Metallic-Roughness parameters**
  - Consistency and fallbacks for performance for any device
- **Inspiration from Dassault Systèmes Enterprise PBR Shading Model (DSPBR)**
  - https://github.com/DassaultSystemes-Technology/EnterprisePBRShadingModel/tree/master/gltf_ext
- **Wide industry collaboration for compatibility**
  - Dassault Systèmes
  - Google Filament
  - Microsoft BabylonJS
  - NVIDIA MDL
  - OTOY Octane

Join the GitHub Discussion!
https://github.com/KhronosGroup/glTF/issues/1442

Images from https://dassaultsystemes-technology.github.io/EnterprisePBRShadingModel/



VRM – Using glTF 2.0

“VRM” is a file format for handling 3D humanoid avatar (3D model) data for VR applications

It is based on glTF 2.0

Anyone is free to use it. In addition, a standard implementation (UniVRM) in c# that can import and export VRM file in Unity is released as open source

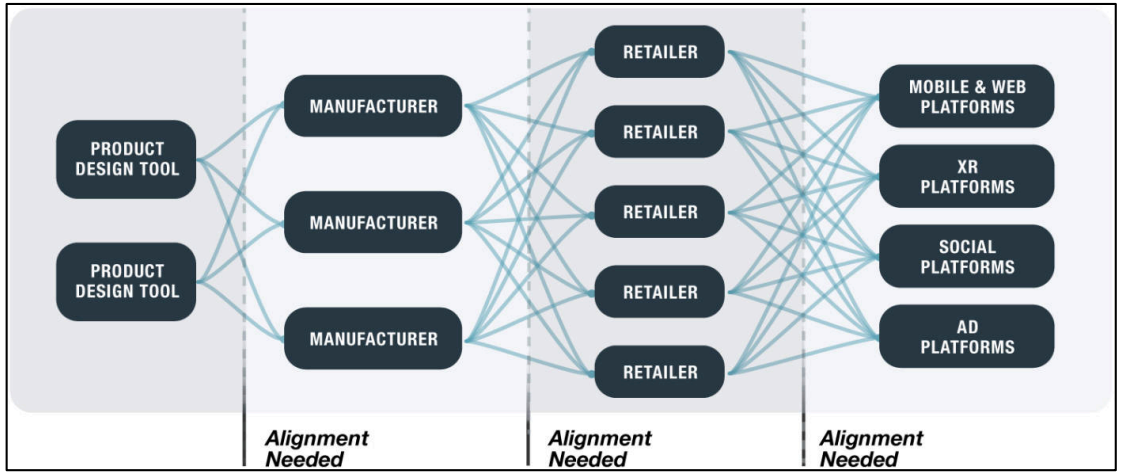
26 Companies based primarily in Japan

https://vrm.dev/en/

Khronos 3D Commerce Exploratory Group

Khronos 3D Commerce Exploratory Group

Over 70 Retail AND Technology companies making virtual 3D product representations possible on an industrial scale



Open to any company under NDA during exploratory phase

https://www.khronos.org/exploratory/3d-commerce/

Roadmap Discussions

- **Many of these topics are being discussed on GitHub**
  - https://github.com/KhronosGroup/glTF
  - Come and give your views!
- **Animation 2.0**
  - Advanced Avatars and Face emoji, with compression
- **LOD and Streaming**
- **Point Clouds (with compression)**
- **Cross-asset linking**
- **Enhanced Metadata**

glTF Roadmap is Driven by Developer Feedback and Requirements

NVIDIA MDL Physically Based Rendering

Resources

- glTF Home Page
  - https://www.khronos.org/gltf/
- glTF GitHub
  - https://github.com/KhronosGroup/glTF
- Universal Compressed Texture Transmission Format CTTF)
  - https://github.com/KhronosGroup/glTF-Texture-Transmission-Tools/milestone/1
- PBR 2.0 – advanced materials
  - https://github.com/KhronosGroup/glTF/issues/1442
- Khronos 3D Commerce Exploratory Group
  - https://www.khronos.org/exploratory/3d-commerce/
- More Information
  - www.khronos.org
  - ntrevett@nvidia.com
  - @neilt3d

KHRONOS® GROUP

KHRONOS GROUP

Background Materials

Focus on glTF Ecosystem Robustness

- **Khronos constantly working on improving ecosystem's consistency**
  - Rendering (reference viewer, reference environment)
  - Technical low-level issues (validator & asset generator)
- **If you are CREATING glTF Files**
  - Ensure generated files are validator clean
    - https://github.com/KhronosGroup/glTF-Validator
  - Help the community understand what your exporter supports
    - https://github.com/KhronosGroup/glTF/issues/1271
- **If you are LOADING glTF files**
  - Ensure loader can correctly load all sample models (integration tests)
    - https://github.com/KhronosGroup/glTF-Sample-Models
  - Ensure loader can correctly load all asset generator models (unit tests)
    - https://github.com/bghgary/glTF-Asset-Generator

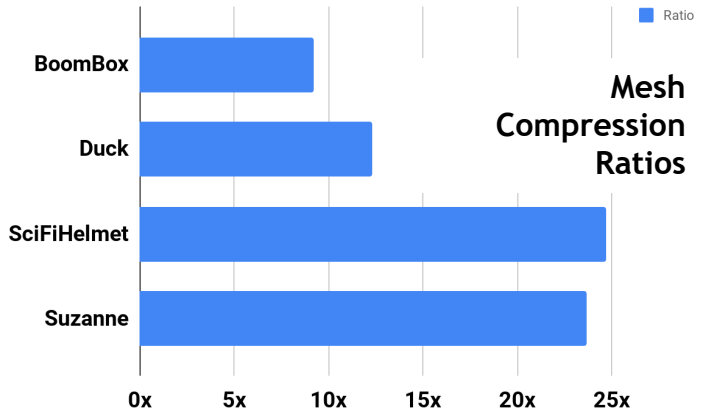
Users of glTF can help to keep glTF reliable and consistent!

glTF Sample Viewer

- **Generates accurate Ground Truth renderings of glTF Models**
  - https://github.com/KhronosGroup/glTF-WebGL-PBR/tree/reference-viewer
- **Not all glTF apps and engines need visual consistency**
  - But is critical to key use cases such as online retail
- **Headless mode, generates images**
  - Compare against offline path-traced renderings
  - For regression testing
- **Can be embedded in Visual Studio Code for live model previews**

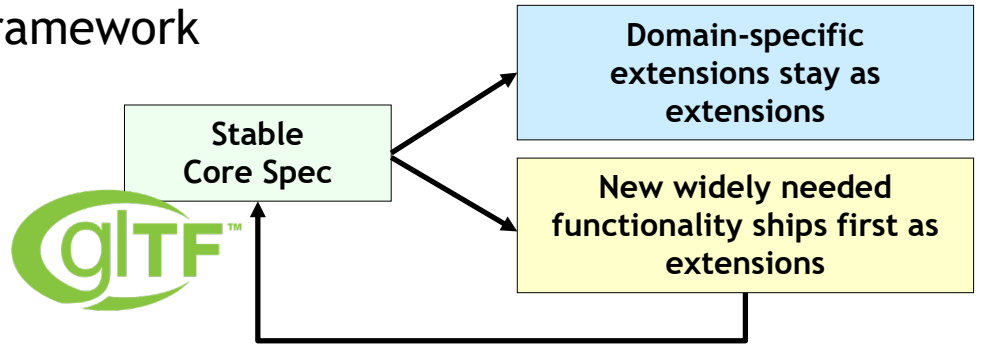
Draco glTF Mesh Compression Extension

- **Library for compressing and decompressing 3D geometric meshes and point clouds**
  - Draco designed and built for compression efficiency and speed - great fit with glTF!
  - https://github.com/google/draco
- **Draco glTF extension launched in February 2018**
  - https://github.com/KhronosGroup/glTF/blob/master/extensions/2.0/Khronos/KHR_draco_mesh_compression/README.md
- **Google has released Draco encoders and decoders in open source**
  - C++ source code encoder to compress 3D data
  - C++ and JavaScript decoders for the encoded data
  - https://github.com/google/draco/tree/gltf_2.0_draco_extension
- **Draco compression already in use**
  - glTF pipeline, FBX2glTF, AMD Compressonator, three.js, and glTF sample models



glTF Evolution Philosophy

- **glTF manages its roadmap very carefully – complexity is the enemy**
  - Mission #1: ensure widespread, consistent, reliable usage
- **Extension for your own use – create a Vendor Extension**
  - Register your PREFIX by submitting a GitHub Issue
- **Multiple applications from multiple vendors need an extension**
  - Create a multi-vendor "EXT" extension
- **Broadly applicable across all apps/platforms - propose a "KHR" extension**
  - Need at least two implementations
  - Discussed and agreed by glTF working group
  - Covered by Khronos Intellectual Property Framework
- **Always have a fallback to core spec**
  - Avoid breaking compatibility with broader ecosystem
  - If you choose to not have a fallback list your extension in *extensionsRequired*



Integrate extensions into new core spec only when:
1) Widespread need is confirmed by the industry
2) Widespread reliable implementation is enabled (e.g. open source)