



## The state of WebGL and glTF

Patrick Cozzi | [@pjcozzi](#)  
3D Formats Working Group  
September 2015

# Khronos Connects Software to Silicon

Open Consortium creating  
ROYALTY-FREE, OPEN STANDARD  
APIs for hardware acceleration

Defining the roadmap for  
low-level silicon interfaces needed  
on every platform

Graphics, compute  
and vision processing

Rigorous specifications AND  
conformance tests for cross-vendor  
portability

*Acceleration APIs  
BY the Industry  
FOR the Industry*



Well over a **BILLION** people use Khronos APIs  
Every Day...

# Today's Topics

- What's new? What's upcoming?
  - Spec and ecosystem



**Today's Main Focus**

# A Few Recent WebGL Apps



[Marmoset Viewer](#)



[NASA Curiosity Rover](#)



[Red Bull X-Alps](#)

# Strong WebGL Ecosystem

- **Platforms**
  - Desktop, Android, and iOS
- **Engines**
  - Three.js, BabylonJS, SceneJS, xeoEngine, Pex, Blend4Web, Cesium
- **Tools**
  - Browser debuggers/profilers, Shader Editors  
WebGL Inspector, Web Tracing Framework
- **Education**
  - Books, online courses, and tutorials
- **Adoption across many domains, not just games**

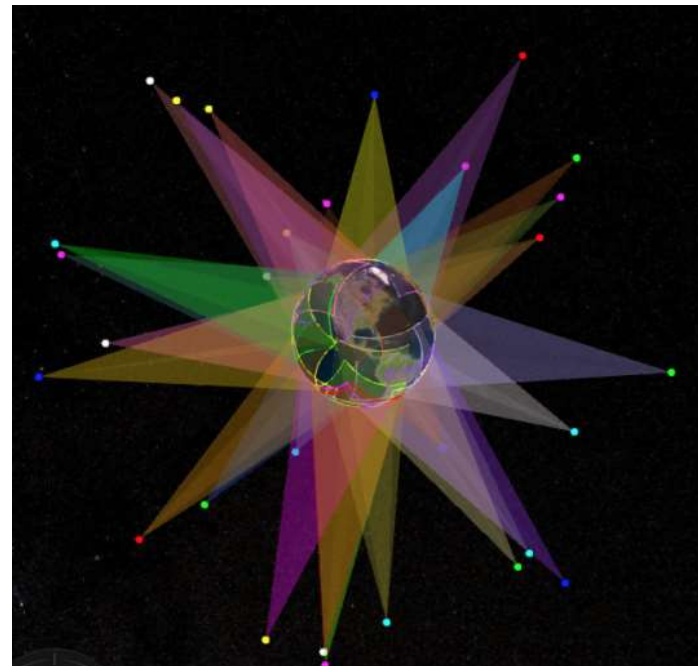
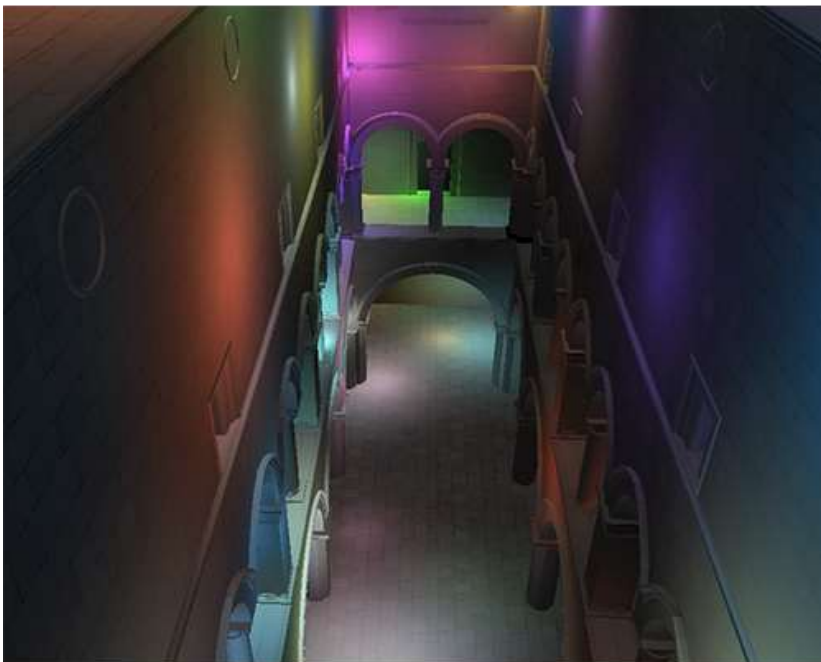


# WebGL 2.0

- **Brings most OpenGL ES 3.0 features to WebGL to enable**
  - Better visual quality
  - Better performance
  - New GPU algorithms
- **Draft spec is available**
  - <https://www.khronos.org/registry/webgl/specs/latest/2.0/>
- **Prototype implementations in Chrome and Firefox**
  - [https://www.khronos.org/webgl/wiki/Getting\\_a\\_WebGL\\_Implementation](https://www.khronos.org/webgl/wiki/Getting_a_WebGL_Implementation)
- **Chromium aiming to pass all known conformance tests by the end of the year**
- **Many WebGL 2.0 features are available today as extensions**
  - ANGLE\_instanced\_arrays, OES\_vertex\_array\_object, WEBGL\_draw\_buffers, ...

# Major WebGL 2.0 Features

- **WEBGL\_draw\_buffers (Multiple Render Targets)**
  - Deferred shading, Order-Independent Transparency (OIT), ...



# Major WebGL 2.0 Features

- Instancing
  - Less draw calls to draw lots of similar objects (instances)





# Major WebGL 2.0 Features

- Uniform buffers
- Transform feedback
- Multisampled Renderbuffers
- 3D textures
- NPOT textures
- More texture formats
- Occlusion queries
- Vertex array objects
- Sampler objects
- Sync objects
- Fragment depth
- Primitive restart
- ...

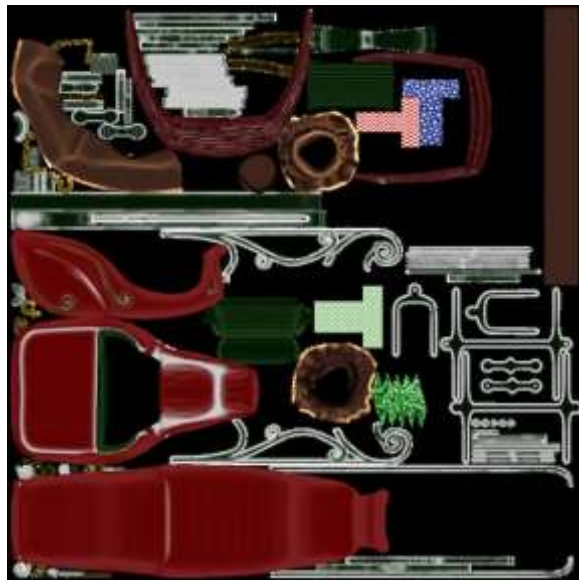


# Beyond graphics APIs, we need a runtime 3D Model Format

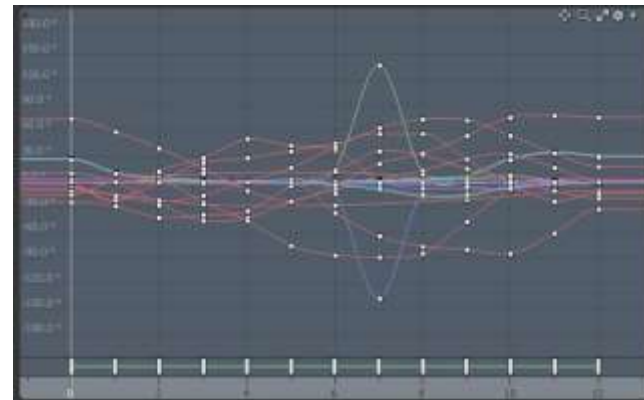
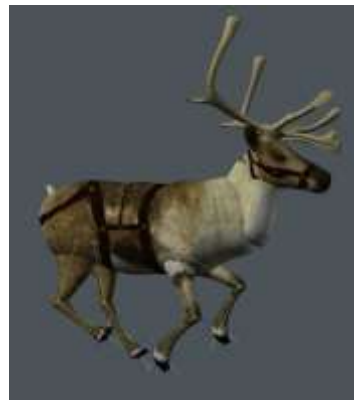
# What's in a 3D model?



Node hierarchy and geometry



Materials and textures



Animations and skins



# Native Modeling Tool Format Examples



↕  
.blend

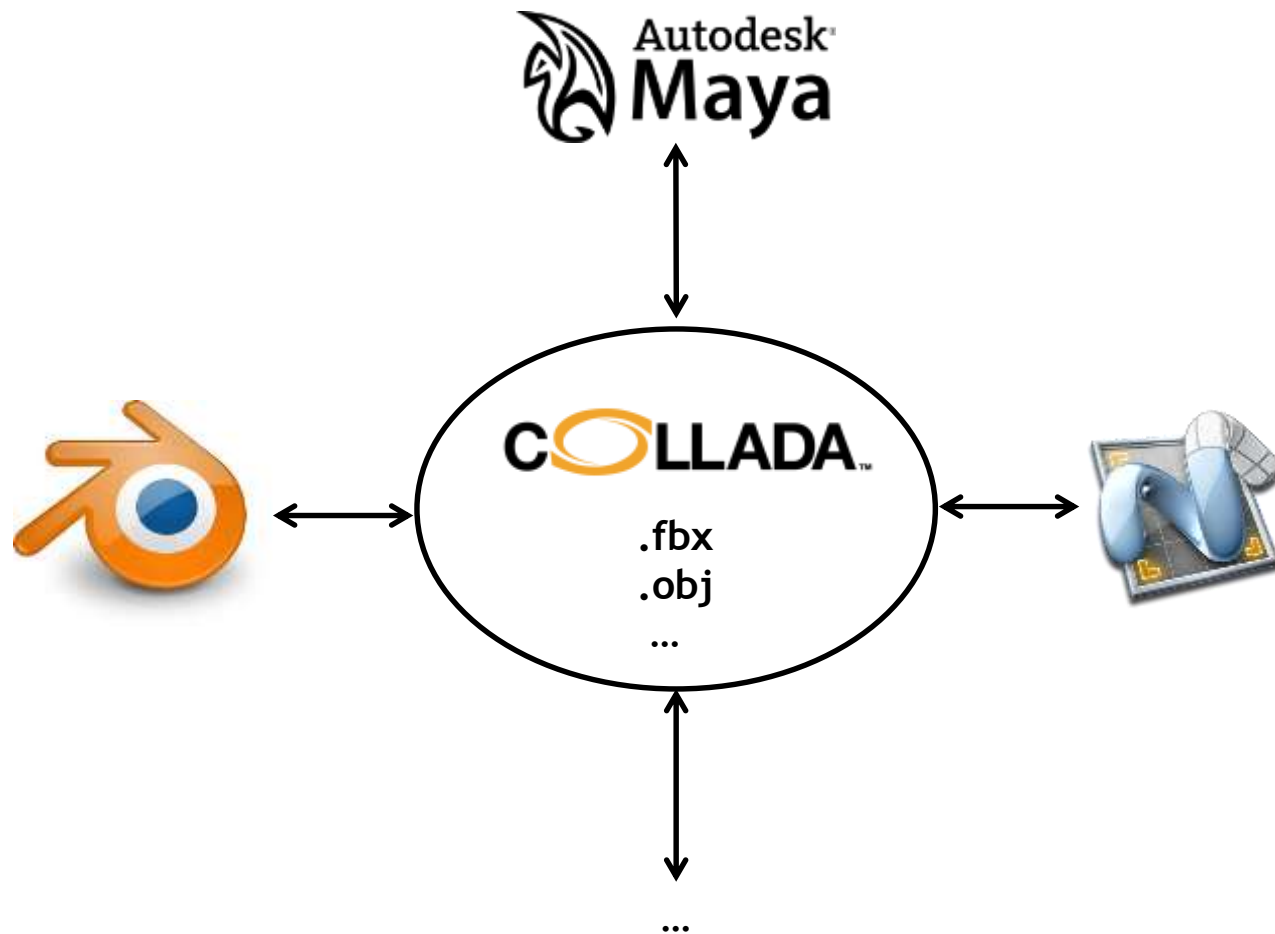


↕  
.ma / .mb

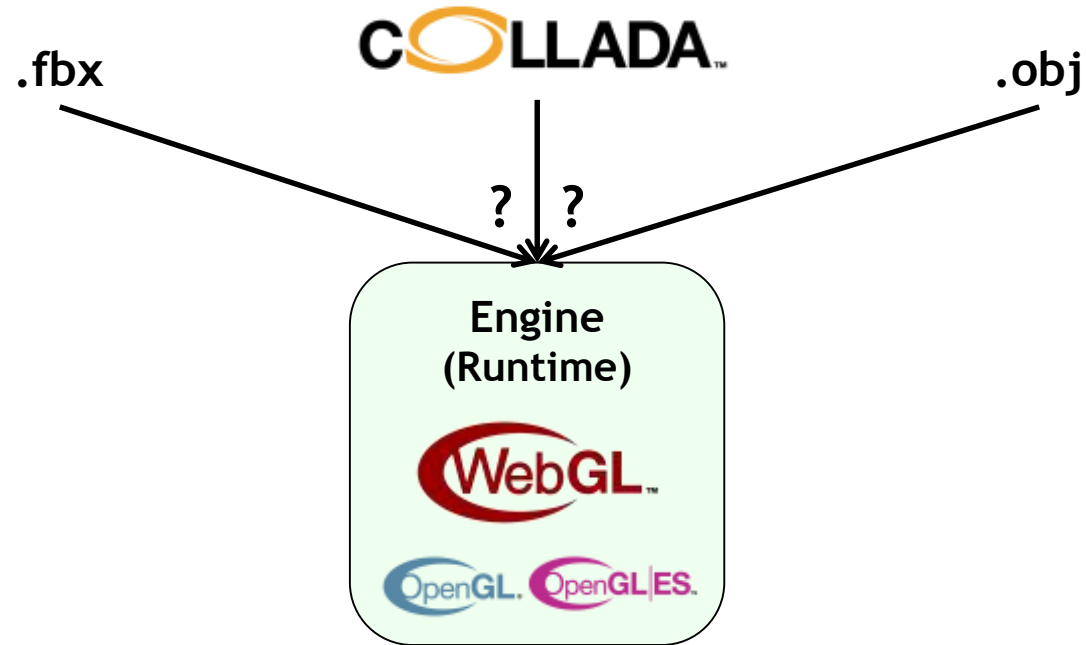


↕  
.lxd

# Interchange Formats



# Interchange Formats

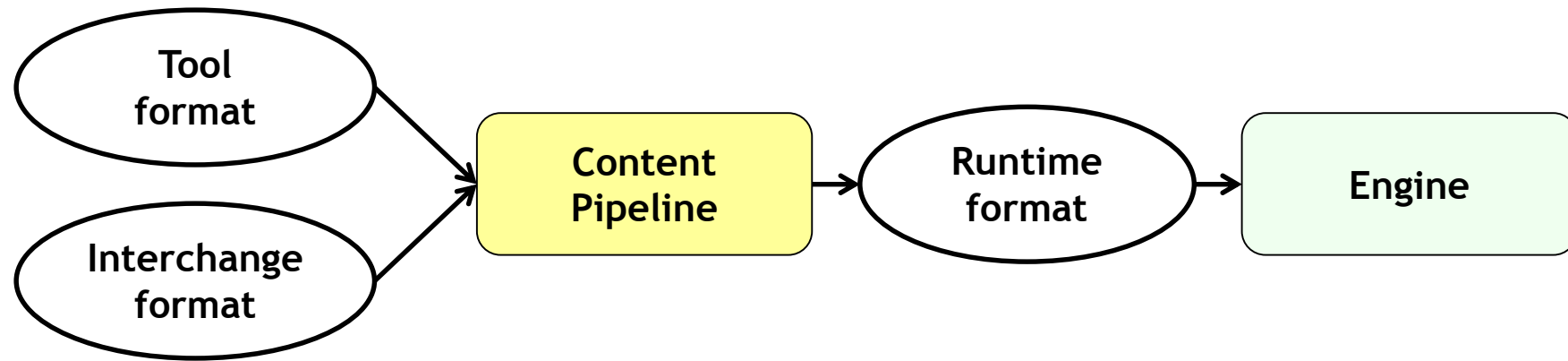


# Interchange Formats

- Target tools - not the graphics API
- Example: COLLADA
  - XML + image files
  - One index per attribute, not vertex
  - Unsigned int indices
  - Transform stack per node
  - Polygons and splines
  - Doesn't specify image file format
  - Lots of flexibility and indirection in animations and skins

# Runtime Format




- Optimized for use in a runtime engine





# 3D Needs a Transmission Format!

- Bridge the gap between tools and 'GL' based apps
  - Reduce duplicated effort in content pipelines
  - Enable richer 3D representation - OBJ, STL etc. too limited
  - Provide common publishing format for content tools and services

Audio	Video	Images	3D
MP3	H.264	JPEG	?
 <i>napster.</i>	 YouTube™	 facebook	!

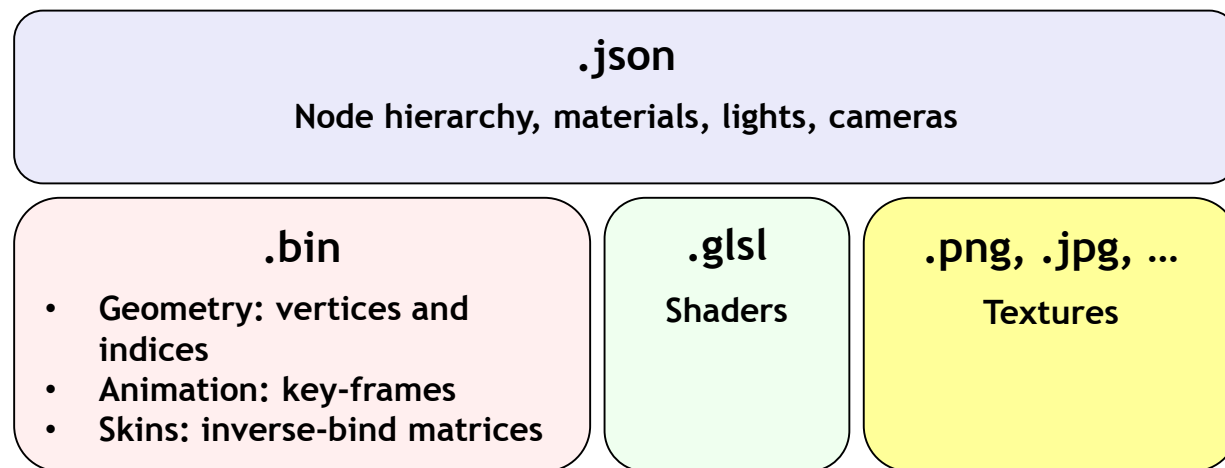
A widely adopted format ignites previously unimagined opportunities for a media type

# glTF = “JPEG for 3D”

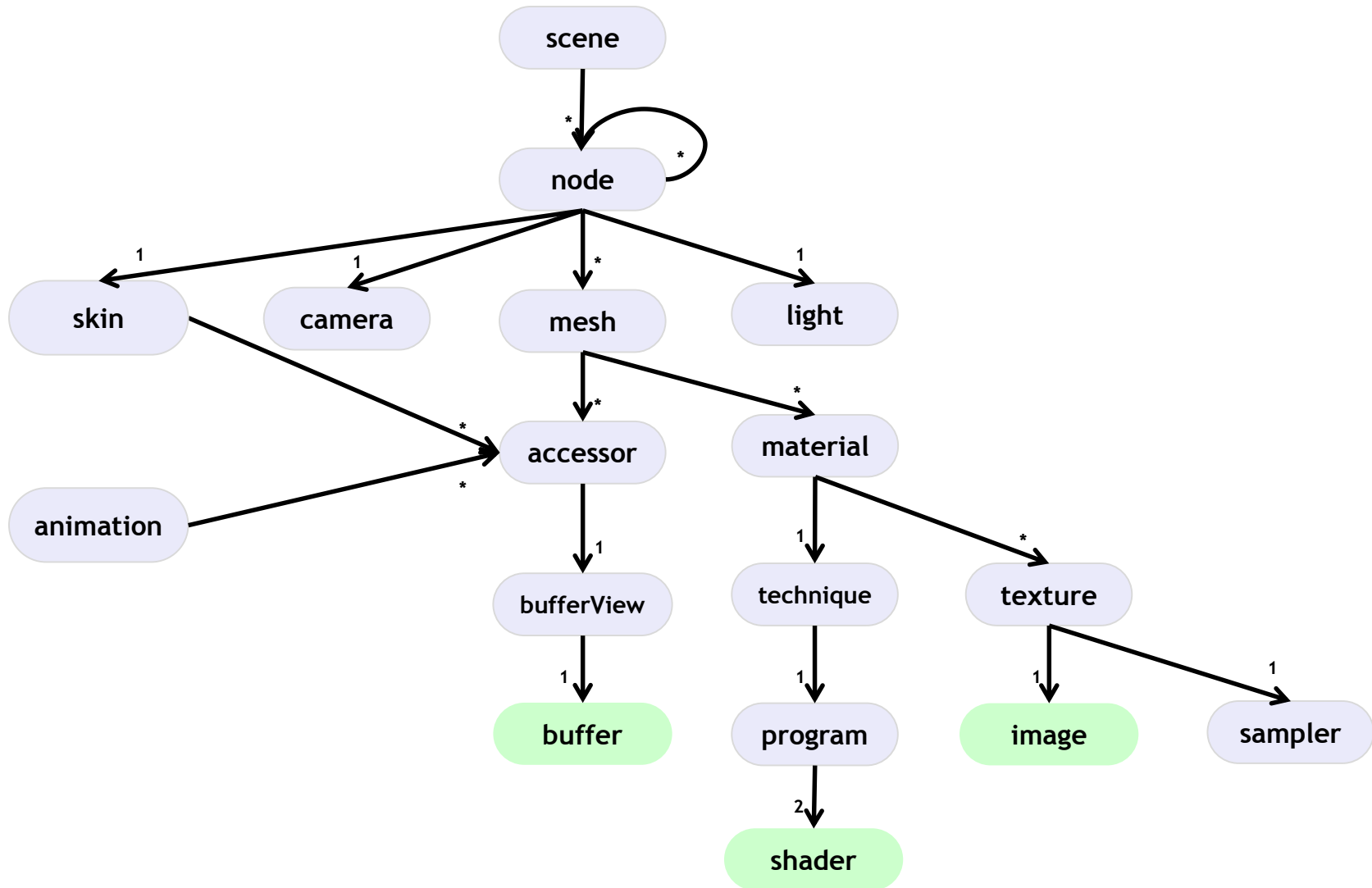
- ‘GL Transmission Format’
  - Runtime asset format for WebGL, OpenGL ES, and OpenGL applications
- Compact representation for download efficiency
  - Binary mesh and animation data
- Loads quickly into memory
  - GL native data types require no additional parsing
- Full-featured
  - 3D constructs (node 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



# glTF Internals



# glTF Internals

The screenshot displays the AGI Model Inspector application. The main window shows a 3D model of a truck. On the left, a Hierarchy panel shows a tree structure with nodes: defaultScene, polyRenGeometry\_mesh001Node, Geometry\_mesh002Node, meshInst004Node, and meshInst009Node. A Color Key panel is also visible. The top right shows performance metrics: 16.90 MS and 58 FPS. Below this are checkboxes for Wireframe, Bounding Volume, and Reference Frame. An Animation Player panel is also present. The bottom left shows Key Statistics for the selected node: Geometry\_mesh002Node, with 5 draw calls and 7860 rendered primitives. The bottom right shows a glTF JSON viewer with the following content:

```
glTF
{
  "Geometry_mesh002Node": {
    "children": [
      "Geometry_mesh001Node",
      "meshInst004Node",
      "meshInst009Node"
    ],
    "matrix": [
      1,

```

# glTF Project Status

- **Open specification; Open process**
  - Specification and multiple loaders and translators in open source
  - <https://github.com/KhronosGroup/glTF>
- **Draft glTF 1.0 spec is imminent**
  - spec-1.0 branch
- **Extension mechanisms fully defined**
  - Vendor, multi-vendor and official Khronos extensions (mirrors OpenGL)
  - Anyone can ship vendor extensions at any time - no permissions needed
  - First extensions will be included in launch



# glTF Adoption

three.js Loader

<https://github.com/mrdoob/three.js/>

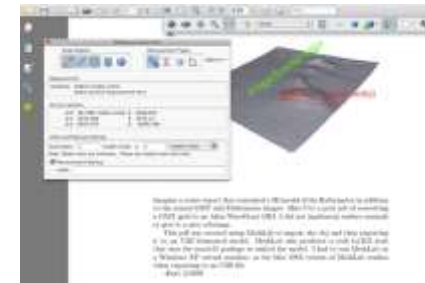


It's the native format!

<http://cesiumjs.org/>



Native import and display of glTF models



Babylon.js Loader (in development)

<http://www.babylonjs.com/>



## PIPELINE TOOLS

collada2glTF converter

<https://github.com/KhronosGroup/glTF>

Online drag and drop COLLADA to glTF converter

<http://cesiumjs.org/convertmodel.html>

FBX to glTF Converter

(in development)

Drag and drop convertor coming

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



AUTODESK

[a.mo.bee]

3D Advertising Solutions with native glTF import



# Initial glTF Extensions

- **KHR\_binary\_glTF (Khronos extension)**
  - Enables a glTF file to refer to external binary asset packages
- **EXT\_quantized\_attributes (vendor extension)**
  - Quantization-based attribute compression
  - Decompression in vertex shader
- **Open3DGC - MPEG 3D mesh compression (in progress)**
  - C++ encoder/decoder + JavaScript decoder
  - 40-80% compression
  - Extensions inserts decompression between file buffer and vertex data
  - Building support into COLLADA2GLTF converter and Cesium loader





# Open3DGC glTF Extension Initial Results

Model	Vertices	Tris	Flat + Gzip	Open3DGC + Gzip	Compression Amount	JavaScript Execution Time
COLLADA Duck	2.1k	4.2k	54 KiB	14 KiB	-74%	24 ms
Stanford Bunny	2.5k	5.0k	105 KiB	56 KiB	-47%	30 ms
Stanford Dragon	435k	871k	7792 KiB	2141 KiB	-73%	630 ms
3D Tile	12.8k	6.5k	102 KiB	59 KiB	-42%	—
OpenStreetMap NYC	—	—	337 MiB	207 MiB	-39%	(Streamed)

Google Chrome 44.0, Windows 8.1, Intel i7-4980HQ @ 2.80GHz

# Cesium 3D Tiles Using glTF

- An [open specification](#) for streaming massive 3D geospatial datasets
  - Streams 3D content including buildings, trees, point clouds, and vector data
- **Hierarchical Level of Detail (HLOD)**
  - Only visible and prioritized tiles are streamed
  - glTF payloads can be compressed, e.g., using [3DGC](#) extension



Over 1.1 million OpenStreetMap buildings in New York City



# Get Involved!

- **WebGL**

- Experiment with WebGL 2 implementations:

[https://www.khronos.org/webgl/wiki/Getting\\_a\\_WebGL\\_Implementation](https://www.khronos.org/webgl/wiki/Getting_a_WebGL_Implementation)

- **glTF**

- Review the spec:

<https://github.com/KhronosGroup/glTF/blob/spec-1.0/specification/README.md>

- **Contact**

- Patrick Cozzi
- @pjcozzi
- pjcozzi@siggraph.org

