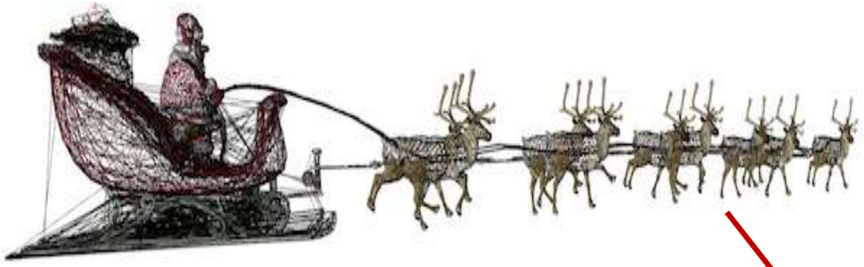
glTF Introduction

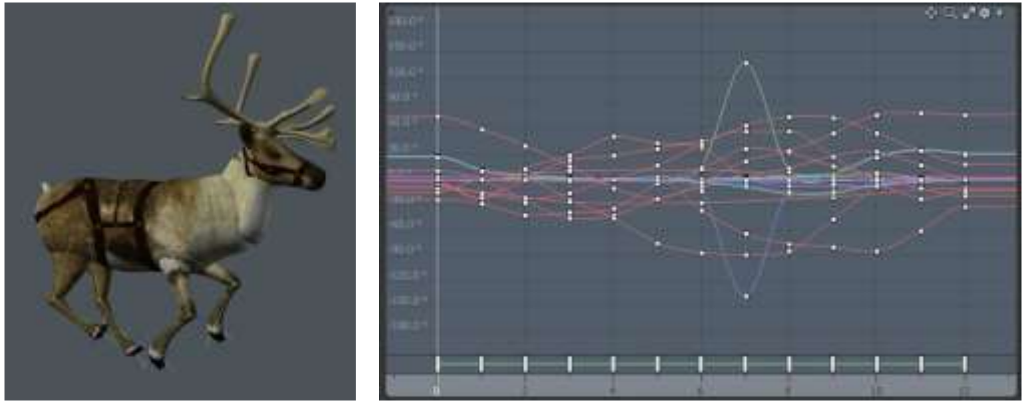
‘OpenGL Transmission Format’

October 2015

What’s in a 3D Asset or Model?



Scene hierarchy and geometry

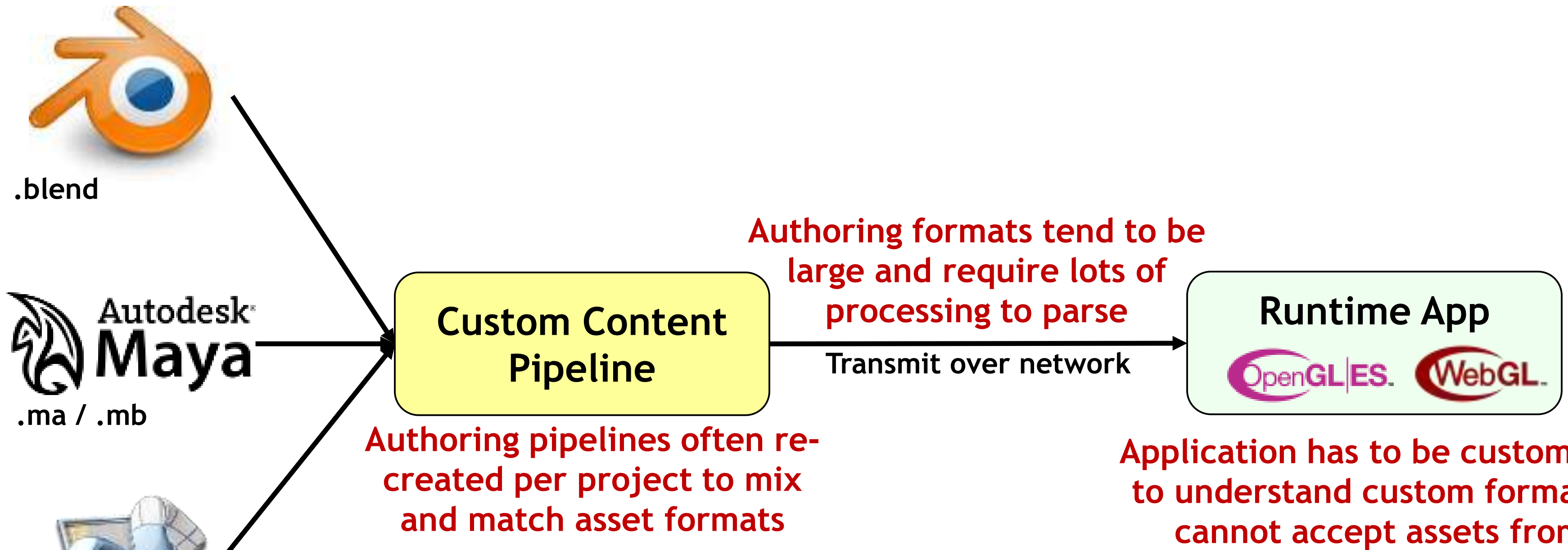


Animations and skins

Materials and textures

Final Asset in Scene

3D Model Creation and Deployment - Today



.blend

.ma / .mb

.lxo

Custom Content Pipeline

Transmit over network

Runtime App

Authoring formats tend to be large and require lots of processing to parse

Authoring pipelines often re-created per project to mix and match asset formats

Application has to be customized to understand custom formats - cannot accept assets from diverse servers
-> Silo'd content

>30 3D formats in use
OBJ/STL contain single-models NOT scenes
Need lights, cameras, animations, scene hierarchy etc.

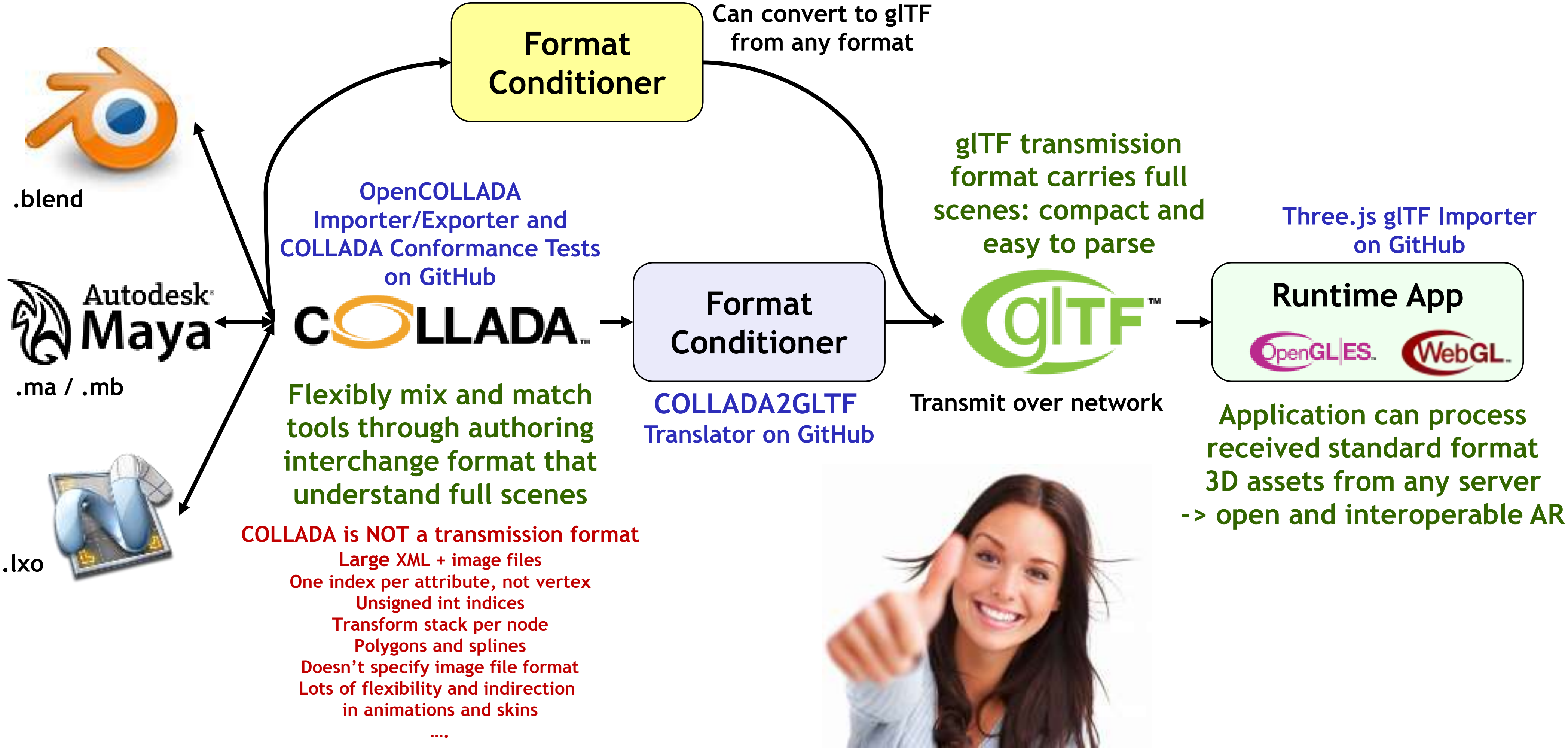
3D Needs a Transmission Format!

- **Efficient run-time transmission of 3D assets becoming essential**
  - Connected applications need access to increasingly large asset databases
- **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	?
napster	YouTube	facebook	!

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

3D Model Creation and Deployment Standards!



OpenCOLLADA Importer/Exporter and COLLADA Conformance Tests on GitHub

Flexibly mix and match tools through authoring interchange format that understand full scenes

COLLADA is NOT a transmission format

Large 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
….

COLLADA2GLTF
Translator on GitHub

glTF transmission format carries full scenes: compact and easy to parse

Three.js glTF Importer on GitHub

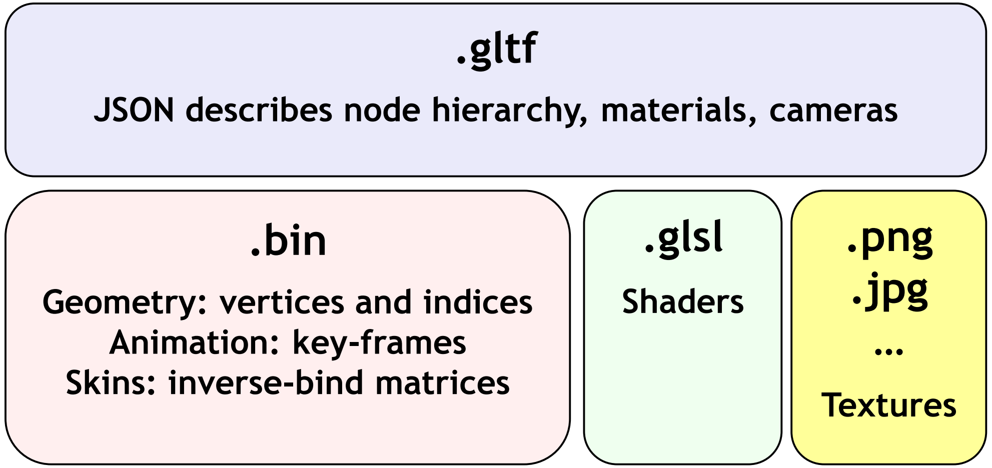
Application can process received standard format 3D assets from any server
-> open and interoperable AR

glTF = "JPEG for 3D"

- 'GL Transmission Format'
  - 3D asset runtime format for any application
  - Optimized for WebGL, OpenGL ES, and OpenGL apps
- **Compact representation for download efficiency**
  - Binary mesh and animation data
- **Loads quickly into memory**
  - GL native data types require no additional parsing
- **Full-featured scenes**
  - 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

JSON describes node hierarchy, materials, cameras

.bin

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

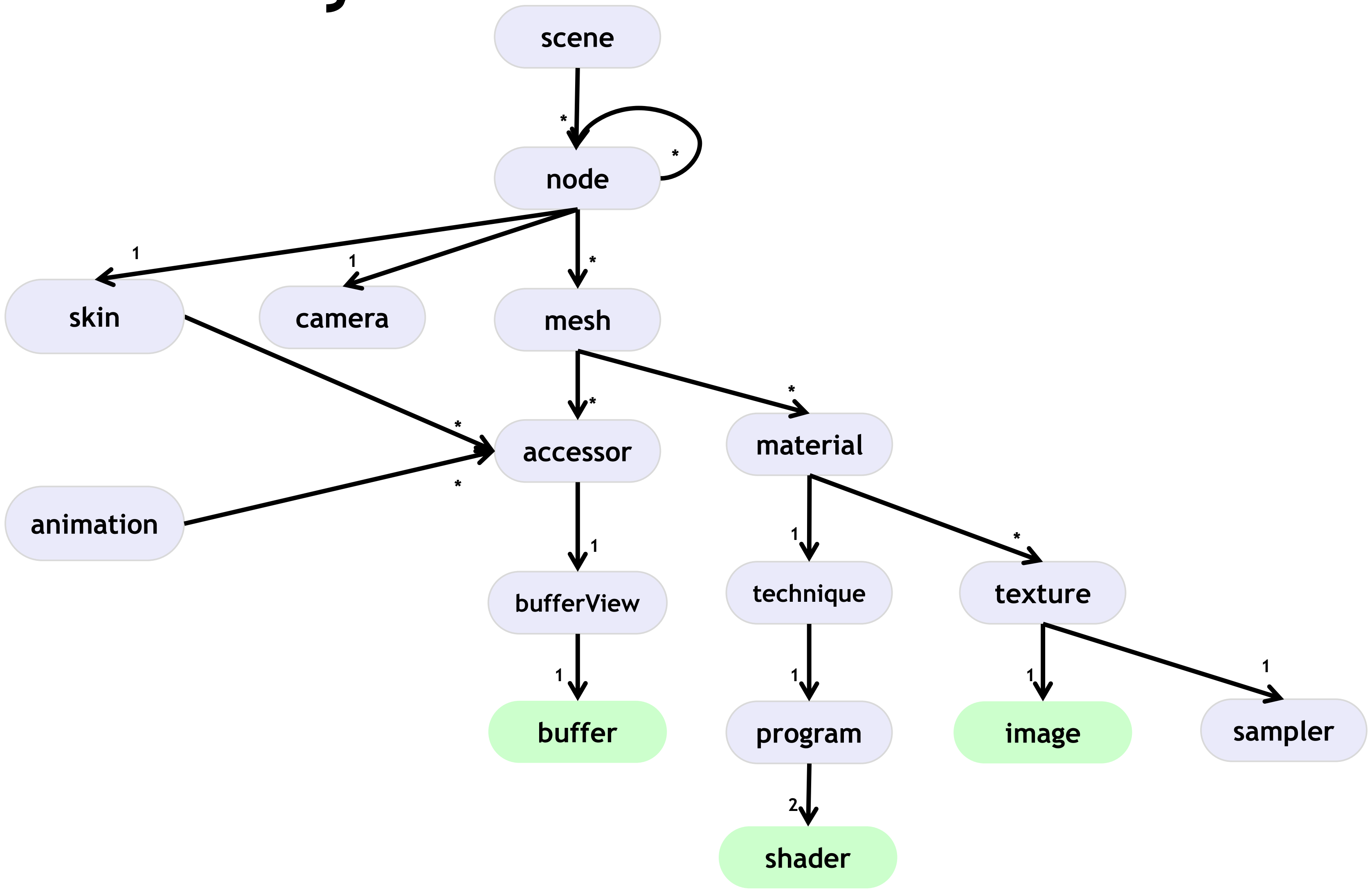
.glsl

Shaders

.png
.jpg
…

Textures

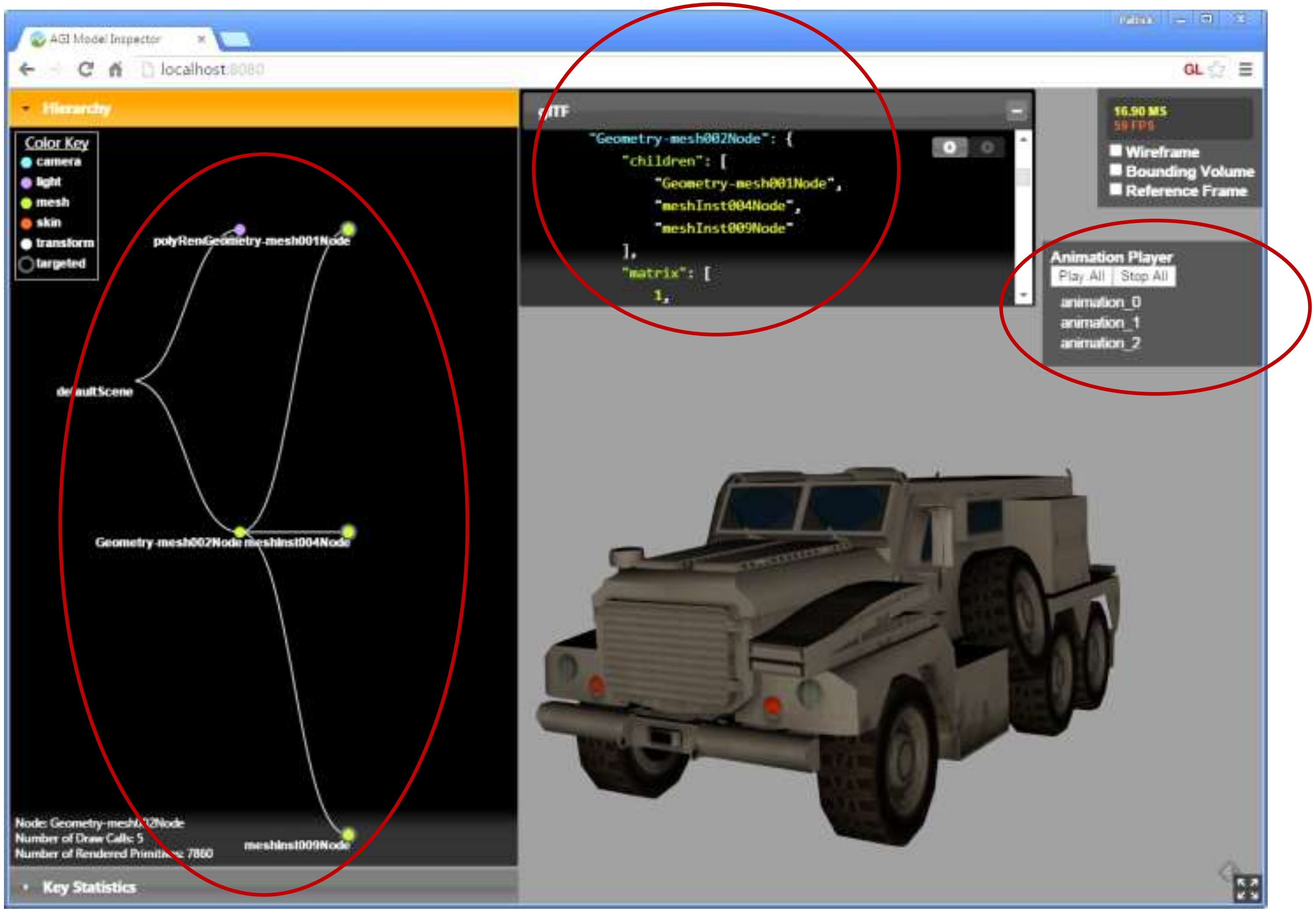
glTF Hierarchy



glTF Example

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

Visualization of Node Hierarchy



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

Node: Geometry-mesh002Node
Number of Draw Calls: 5
Number of Rendered Primitives: 7860

Three animations - one for each set of wheels

glTF Project Status

- **Open specification; Open process**
  - Specification and multiple loaders and translators in open source
  - https://github.com/KhronosGroup/glTF
- **glTF 1.0 spec finalized**
  - Launched in October 2015
- **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/

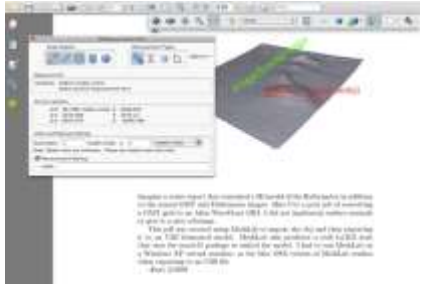
CESIUM

It's the native format!

http://cesiumjs.org/

LibreOffice

Native import and display of glTF models



Microsoft

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 Convertor

(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

- **Any company can define glTF vendor extensions**
  - Khronos manages extension name space
  - Popular extensions can be proposed to be adopted into standard extensions and then possibly into core
- **KHR_binary_glTF (Khronos extension)**
  - Enables a glTF file to use binary asset packages
- **EXT_quantized_attributes (vendor extension)**
  - Quantization-based attribute compression
  - Decompression in vertex shader
- **MPEG 3D mesh compression (in progress)**
  - MPEG-SC3DMC codec (Scalable Complexity 3D Mesh Compression)
  - Uses Open3DGC open source - C++ encoder/decoder + JavaScript decoder
  - 40–80% compression for many 3D assets
  - Extensions inserts decompression between file buffer and vertex data
  - Building support into the COLLADA2GLTF converter and Cesium loader

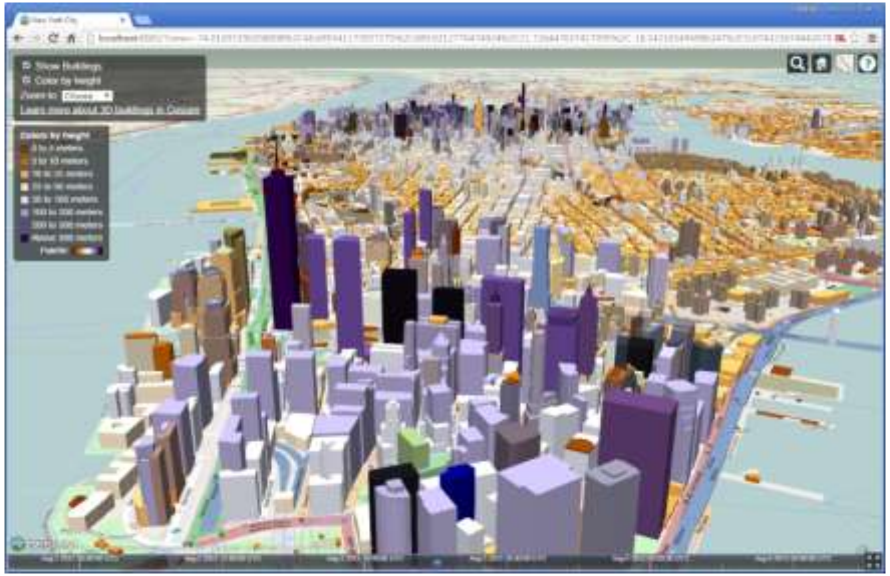
Open3DGC glTF Extension Initial Results

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)

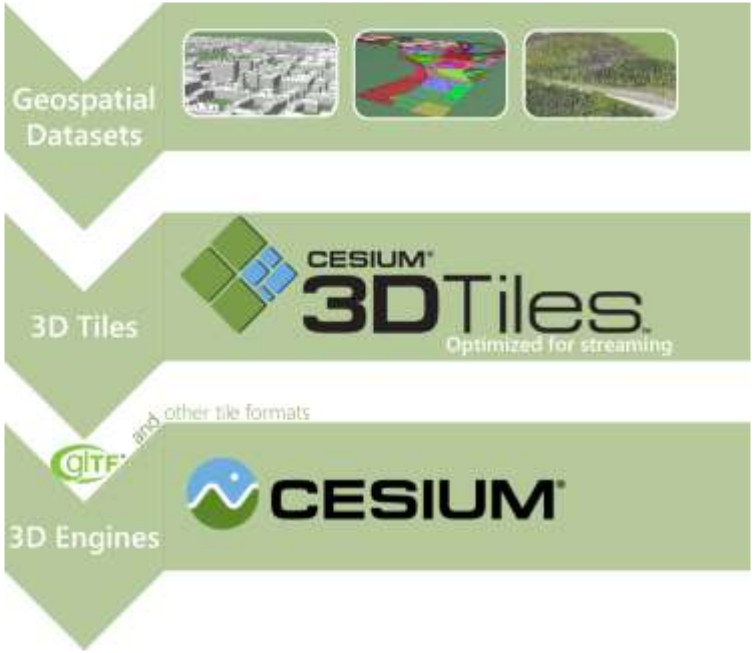
Google Chrome 44.0, Windows 8.1, Intel i7-4980HQ @ 2.80GHz

Cesium 3D Tiles Using glTF (Spring 2016)

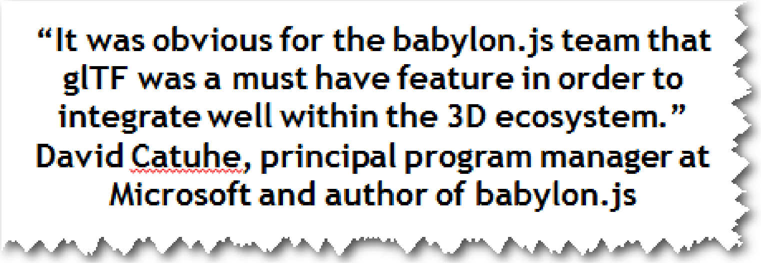
- An open specification for streaming massive 3D geospatial datasets
  - Streams 3D content including buildings, trees, point clouds, and vector data
- Hierarchical Level of Detail (HLOD)
  - Only visible and prioritized tiles are streamed
  - glTF payloads can be compressed, e.g., using 3DGC extension



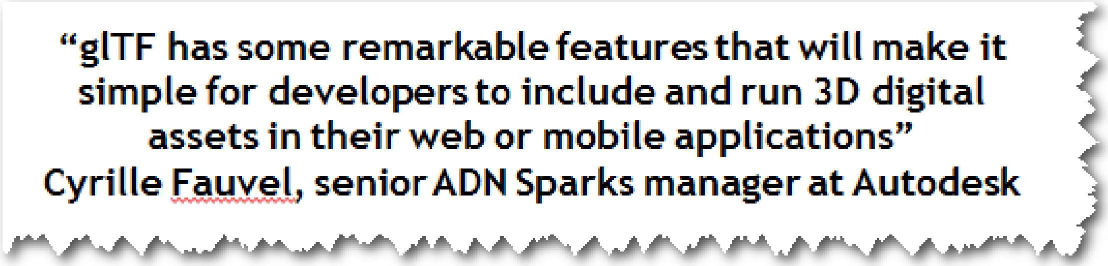
Over 1.1 million OpenStreetMap buildings in New York City



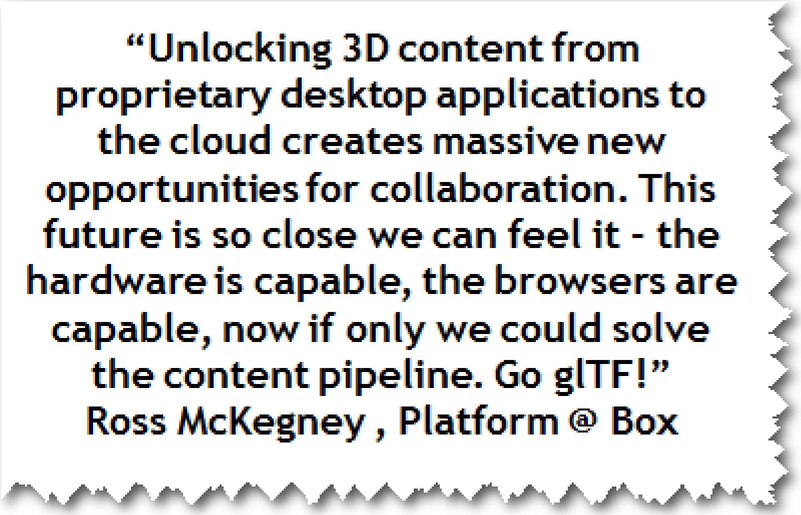
Launch Industry Support



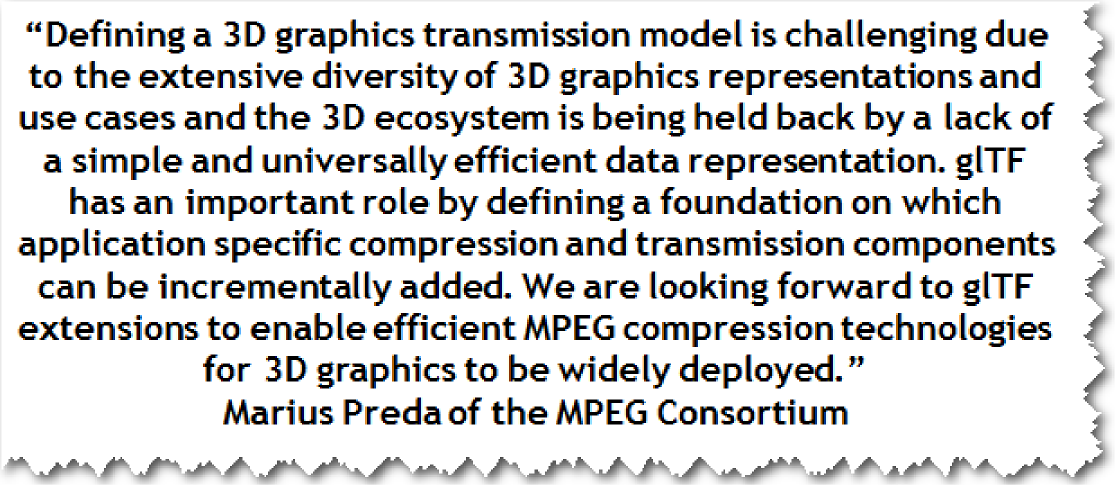
"It was obvious for the babylon.js team that glTF was a must have feature in order to integrate well within the 3D ecosystem."
David Catuhe, principal program manager at Microsoft and author of babylon.js



"glTF has some remarkable features that will make it simple for developers to include and run 3D digital assets in their web or mobile applications"
Cyrille Fauvel, senior ADN Sparks manager at Autodesk



"Unlocking 3D content from proprietary desktop applications to the cloud creates massive new opportunities for collaboration. This future is so close we can feel it - the hardware is capable, the browsers are capable, now if only we could solve the content pipeline. Go glTF!"
Ross McKegney , Platform @ Box



"Defining a 3D graphics transmission model is challenging due to the extensive diversity of 3D graphics representations and use cases and the 3D ecosystem is being held back by a lack of a simple and universally efficient data representation. glTF has an important role by defining a foundation on which application specific compression and transmission components can be incrementally added. We are looking forward to glTF extensions to enable efficient MPEG compression technologies for 3D graphics to be widely deployed."
Marius Preda of the MPEG Consortium

Get Involved with glTF!

- **glTF specification**
  - Review and use the specification: https://github.com/KhronosGroup/glTF/blob/spec-1.0/specification/README.md
- **More details**
  - https://www.khronos.org/gltf/
- **Questions and supportive quotes**
  - ntrevett@nvidia.com
  - @neiltd3d
  - #gltf