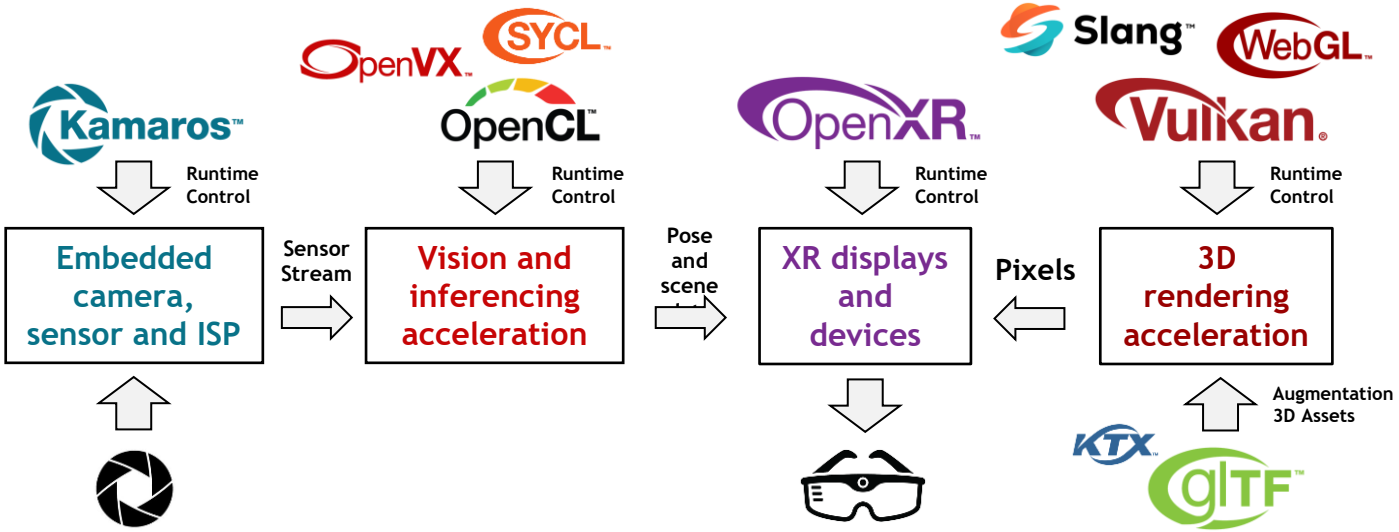
OpenXR and glTF Update

Portable Spatial Computing

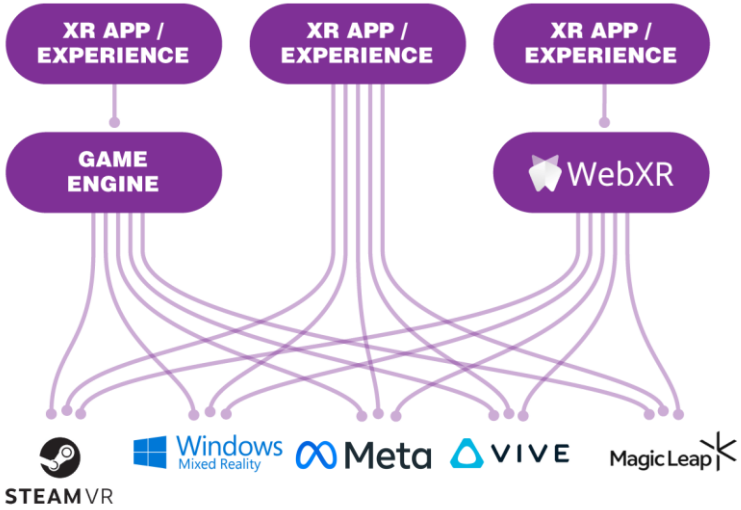
June 2025

Khronos Standards for Spatial Computing

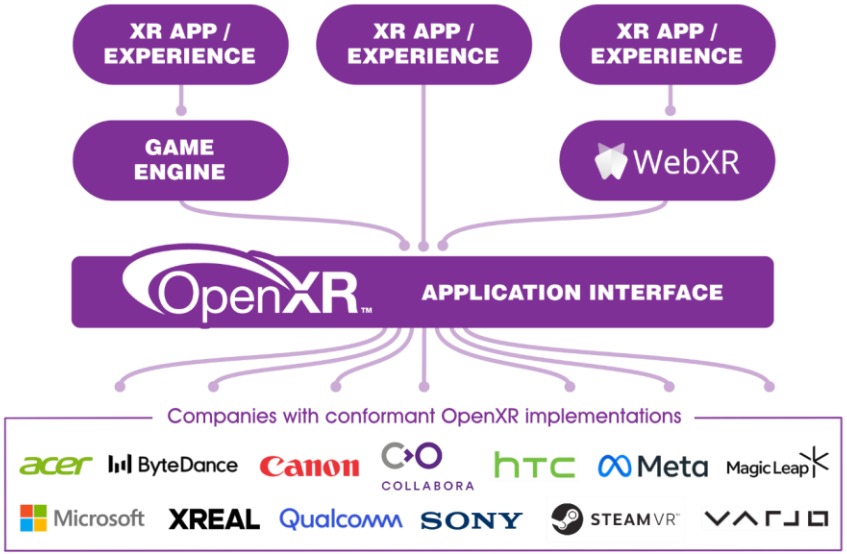
- Ongoing discussions and proposals on how to evolve OpenXR to meet developer needs
  - Carefully considered additions can be widely adopted by the OpenXR ecosystem
- Opportunity to leverage other Khronos standards for camera control and inferencing
- Join Khronos to help inform and steer the evolution of open standards for XR!



OpenXR Cross-Platform Portability



Before OpenXR: Applications and engines needed separate proprietary code for each device on the market.

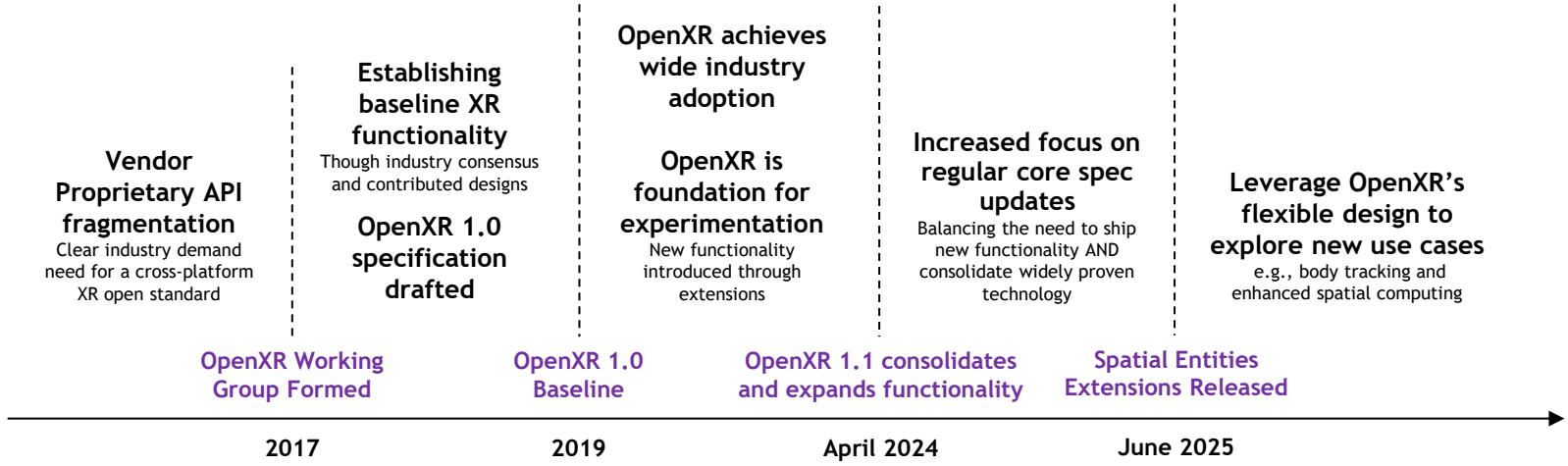


OpenXR provides a single cross-platform, high-performance API between applications and all conformant devices.

Applications and engines can portably access any OpenXR-conformant hardware

The OpenXR Story So Far…

Evolving the industry standard for portable spatial computing



OpenXR Spatial Entities

- **Plane Tracking**
  - Detect and track real-world surfaces
- **Marker Tracking**
  - Track visual markers like QR codes
- **Spatial Anchors**
  - Pin virtual content in the real-world
- **Cross-Session Persistence**
  - Manage spatial context across sessions

A structured set of OpenXR extensions creates the industry's first open standard for spatial computing

OpenXR™

Build Once. Persist Everywhere

- **Discoverable & Extensible cross-vendor API**
  - OpenXR Specification
  - Unified access across XR runtimes and devices
  - Accelerate innovation with a unified spatial model

- **Wide industry support**
  - No more platform-specific spatial code
  - Future-proof applications
- **Baseline XR_EXT_spatial_entities extension**
  - Foundation for representing and interacting with spatial elements in the user's environment
  - Enables ongoing innovation while maintaining consistency and compatibility
- **Developer feedback is shaping spatial computing API evolution**
  - Image and object tracking?
  - Generation and processing of mesh-based models of the user's environment?
- **Join the Discussion!**
  - OpenXR Discord channel | OpenXR Forums | GitHub Issue Tracker

Conformant OpenXR Devices

 Microsoft	 Meta	 htc
HoloLens and Mixed Reality Headsets. Hand and eye tracking extensions	Rift S, Quest 3, Quest 2 and Quest Pro Meta Deprecated own API for OpenXR	Vive Focus 3, Vive Cosmos, Vive XR Elite, Vive Wave Runtime
 VALVE	 VARJO	 Canon
Valve Index Valve Deprecated OpenVR APIs for OpenXR	All Varjo Headsets are fully compliant XR-3, XR-4	MREAL X1
 Magic Leap	 XREAL	 Snapdragon spaces
Magic Leap 2	XREAL Air 2, Air 2 Pro, Air 2 Ultra	Qualcomm Snapdragon Spaces XR Development Platform
 acer	 PICO	 SONY
Spatial Labs Display Series	Neo 3, Pico 4, Pico 4 Ultra	Spatial Reality Displays

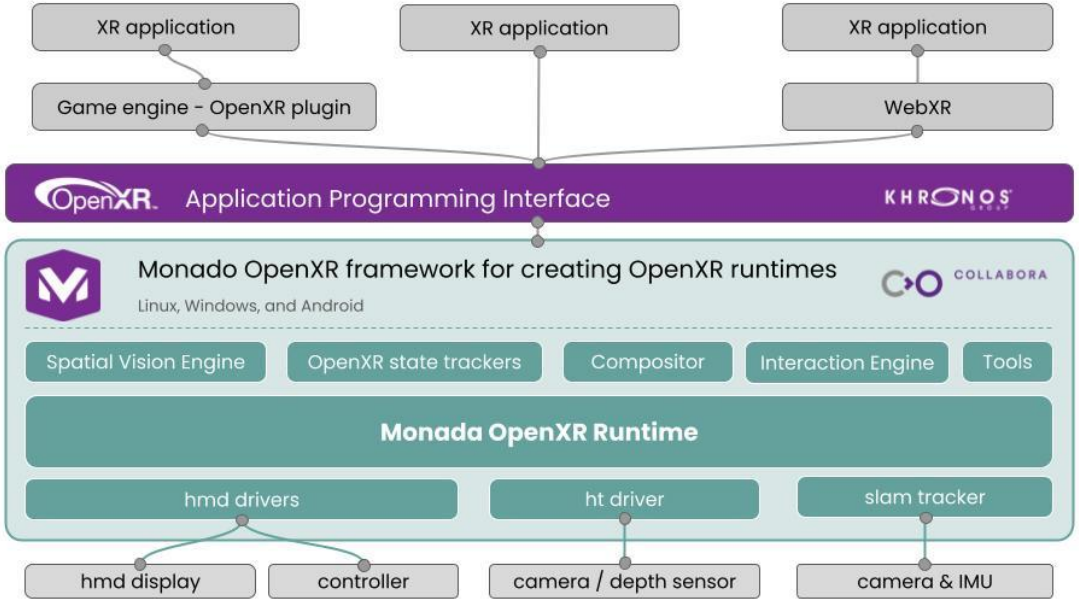
Engines, Browsers, and Libraries with OpenXR

Unreal has been providing support since 4.24. UE 5.0 supports OpenXR	Unity's OpenXR plugin available since 2020 LTS	Godot provides OpenXR support since March 2023 (Core 4.0 Alpha 4)
OpenXR supported since VRED 2023.4	NVIDIA Omniverse and CloudXR Platforms	WebXR in Chrome, Edge, and Firefox uses OpenXR as the default backend
Open-source OpenXR Implementation	A lightweight XR Meta XR Simulator to Speed Unity OpenXR Development	Open-source mixed reality library for building HoloLens and VR applications

Monado from Collabora

OPEN-SOURCE

- Open source OpenXR Runtime and Framework
- Modular Framework to simplify XR runtime development



OpenXR 1.1 Key Extensions Promoted to Core

- **Local Floor Reference Space**
  - Gravity-aligned world-locked origin for standing-scale content
  - Estimated floor height built in
  - Recenter to current user position at the press of a button without a calibration procedure
- **Grip Surface**
  - Reliable anchors visual content relative to the user's physical hand
  - Can be tracked directly or inferred from a physical controller’s position and orientation
- **Stereo with Foveated Rendering for XR headsets**
  - Runtimes MAY optionally expose eye-tracked or fixed foveated rendering
  - Portable across multiple graphics rendering APIs
  - Applications renders quad views (two high-res insets)
- **Additional enhancements**
  - Interaction Profile improvements
  - Spec language cleanup and clarifications
  - 13 new interaction profiles added to the core spec

OpenXR Releases in 2024

1.1.38 (Jun.2024)	1.1.40 (Aug.2024)	1.1.41 (Sep 2024)	1.1.43 (Nov 2024)
XR_EXT_composition_layer_inverted_alpha Maintenance updates	XR_KHR_metal_enable Maintenance updates	XR_HTC_body_tracking: XR_ML_spatial_anchors XR_ML_spatial_anchors_storage XR_ML_system_notifications XR_ML_world_mesh_detection XR_ML_view_configuration_depth_range_change Maintenance updates	XR_ML_facial_expression XR_META_passthrough_layer_resumed_event XR_META_colocation_discovery XR_META_spatial_entity_sharing XR_META_spatial_entity_group_sharing Maintenance updates

Coming Soon…

- **Extending hand tracking**
  - To include full body tracking
- **Expanded haptics support**
  - Support immersive experiences through PCM, vibrotractiles, and transients
- **Controller render models (glTF)**
  - Showing and animating a model of the user's actual controller in a future-proof way

OpenXR Development Resources & Tools

- **OpenXR SDK**
  - Headers, source code, and build scripts
  - https://github.com/KhronosGroup/OpenXR-SDK
- **Reference Pages and Reference Guide**
  - Developer documentation
- **OpenXR Tutorial**
  - For creating applications using Android, Linux, Windows
- **Conformance Test Suite**
  - For runtime developers to test, developed as open source
  - Part of the API Adopter Process to be an official OpenXR runtime requires passing these conformance tests
- **Support & Community Forums**
  - OpenXR on Discord
  - OpenXR Forum at Khronos
  - OpenXR Issue Tracker on GitHub
  - Developing OpenXR Resources? Let us know!

Beat Saber's PC implementation using OpenXR is portable to multiple devices

Get Involved!

Provide feedback and requirements on GitHub, Discord, or OpenXR Forums

Get questions answered and make suggestions for improvement!

Join Khronos and the OpenXR Working Group

https://www.khronos.org/openxr/

https://github.com/KhronosGroup/OpenXR-Docs

OpenXR Specification

OpenXR SDK GitHub

glTF: Widely Adopted, Openly Developed

- glTF is widely adopted by tools, engines and applications
- New glTF specifications are publicly developed on the Khronos GitHub
- This presentation provides roadmap highlights and encourages your input and feedback!

Join the ongoing roadmap discussions!

https://github.com/KhronosGroup/glTF/tree/main/extensions#in-progress-khronos-and-multi-vendor-extensions-and-projects

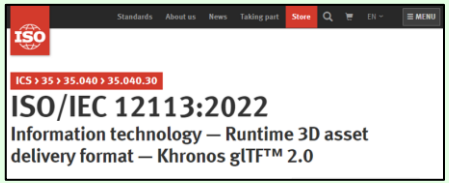
glTF as Foundational Standard

Khronos welcomes working collaboratively to leverage glTF extensibility

Market-specific extensions and use of glTF defined by partner standards organization
Accelerates development of market segment functionality
Avoid needless duplication and fragmentation

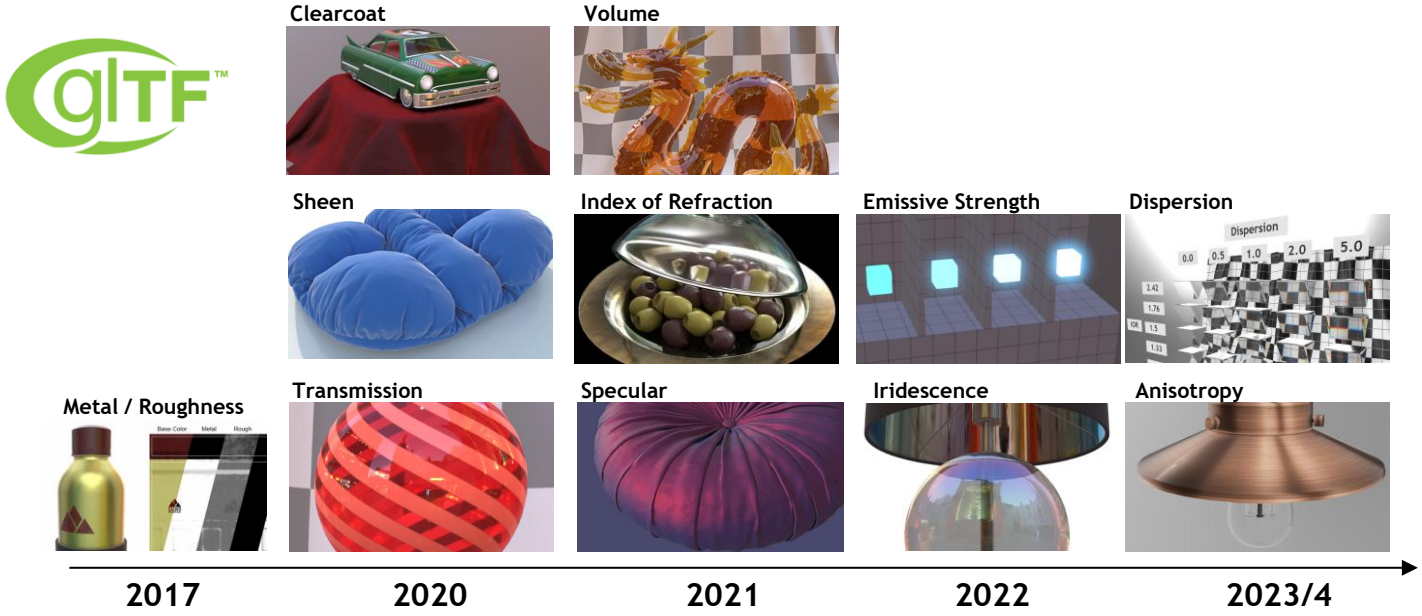
VRM	3D Tiles	MPEG	X3D	PDF	Additional Market Segments
Avatar Format *.vrm extension*	*.b3dm and .i3dm extensions*	ISO 23090-14:2023 MPEG-I *.mp4, miv, ivr*	ISO 19775-1:2023 X3D4 *.x3d extension*	ISO/TS 32007 glTF in PDF *.PDF extension*	

Foundation of
Core specification and
glTF working group extensions



glTF PBR Materials Roadmap

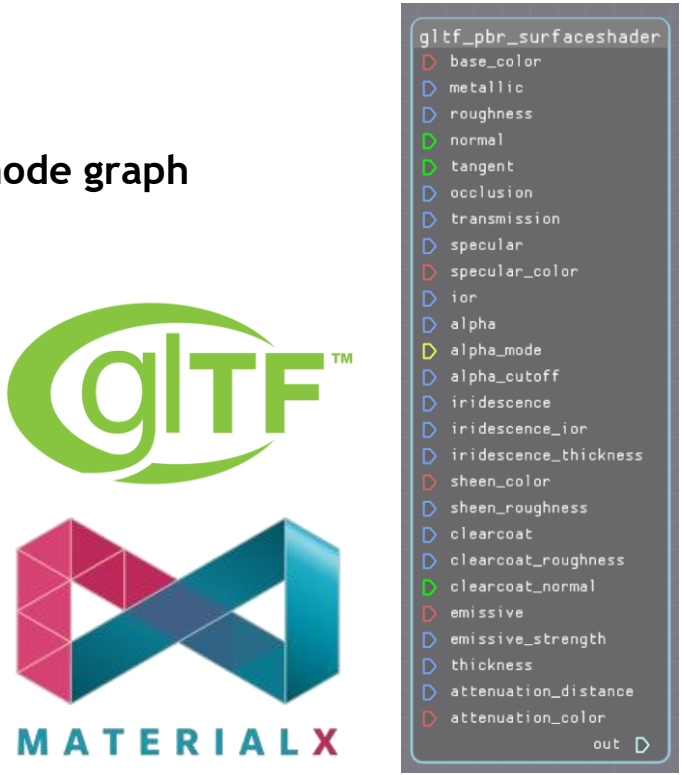
Incremental consolidation and meticulous specification of proven and accepted industry practice as it becomes pervasively deployable



In development
- Subsurface
- Diffuse Transmission

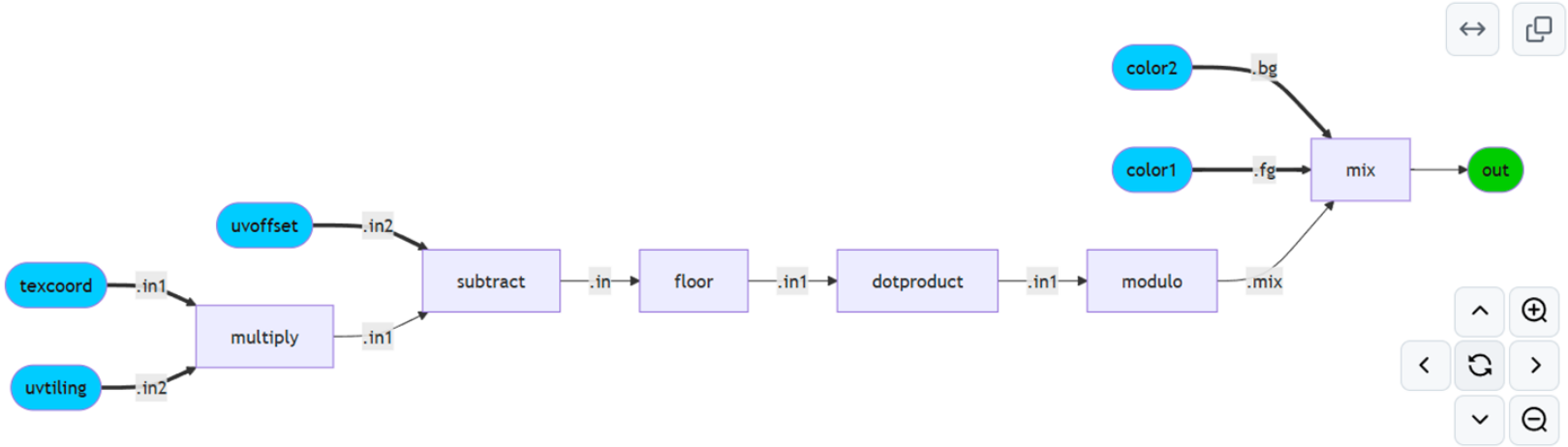
Khronos glTF PBR available in MaterialX

- **Feed MaterialX as a set of procedural texture inputs into glTF PBR**
  - Enable much higher detail in smaller assets
  - Remain compatible with existing PBR shaders
  - Optional texture atlas fallbacks for compatibility
  - Extension in development
- **Next step: Update glTF's PBR model as MaterialX node graph**
  - First released in 2022
  - Planned for 1H25



glTF Procedural Textures

- Procedural graphs which can be mapped to glTF PBR material channels
- Provides customizations and flexibility for the standard glTF PBR model

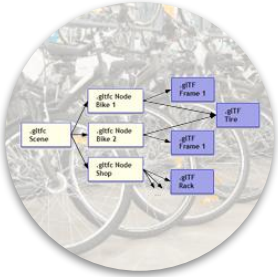


glTF Spatial Computing Roadmap

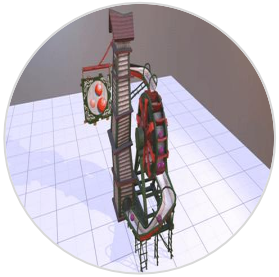
Interactivity

Node-based graph handling of user actions or events

Compose complex scenes from referenced glTF assets
Efficiency and flexibility in transmission/delivery use cases



Complex Scenes



Physics

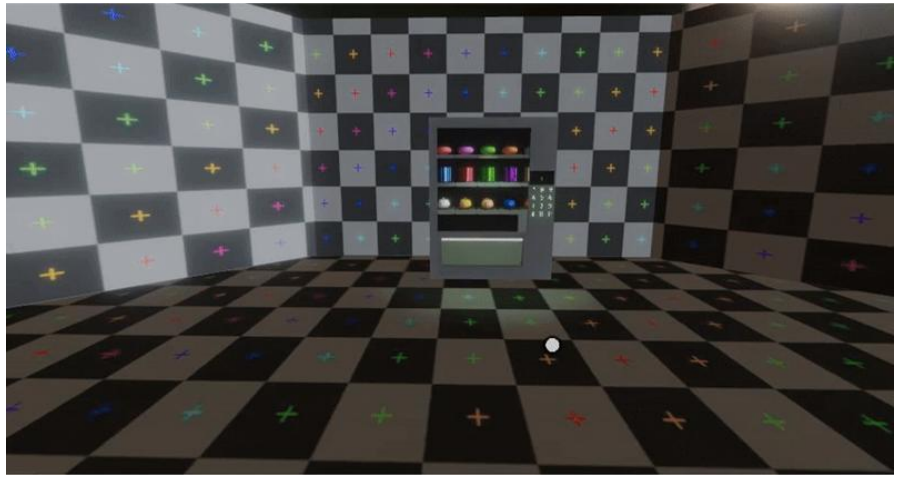
Describes physical properties of assets

Triggered and controlled from interactivity node graph
3D spatialized audio with 6DoF source/listener capabilities,
Play, stop, pause, loop, and speed controls
Splitting, merging, up/down-mixing, reverb, filtering

Audio

glTF Interactivity

- **Portable description of how content should respond to user actions or events**
  - Defined by a node-based graph – sweet spot between flexibility and security
- **Distillation of engine accepted practice**
  - Unity (Visual Scripting), Unreal (Blueprints), Nvidia Omniverse (Action Graph)
- **Node graph provides flexible scene updates**
  - Any scene state can be used in the calculation/animation of any mutable scene property
- **Enables simple interactive applications**
  - Games, Education, Design Review, e-commerce
- **Extensions**
  - KHR_animation_pointer
  - KHR_node_hoverability
  - KHR_node_selectability
  - KHR_node_visibility
  - KHR_accessor_float64



Try Interactive glTF today!

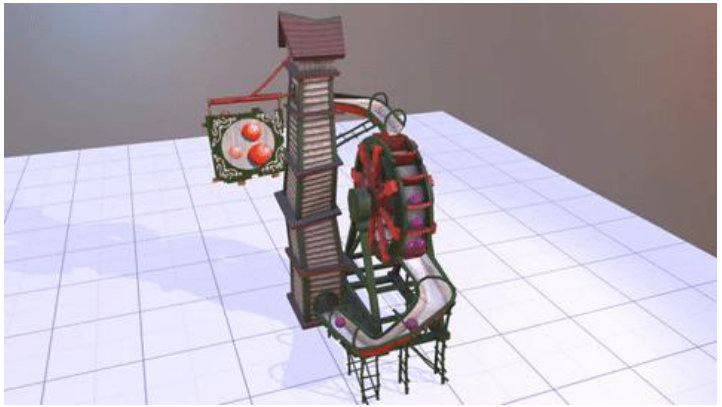
https://github.com/KhronosGroup/glTF/tree/interactivity/extensions/2.0/Khronos/KHR_interactivity

glTF Physics

- **Distillation of widely adopted physics engines practices**
  - KHR_implicit_shapes
  - KHR_physics_rigid_bodies
- **Express the physics properties of assets in a platform independent way**
  - Enables procedural animation
  - Makes scenes more interesting, believable, and dynamic
- **Enables scene understanding**
  - Possible with render geometry, but much more efficient with physics
- **Rigid Bodies**
  - Collision geometry, Rigid bodies, Motions
  - Materials, Joints, Filters

Try and join discussion glTF Physics Today!

https://github.com/eoineoineoin/glTF_Physics

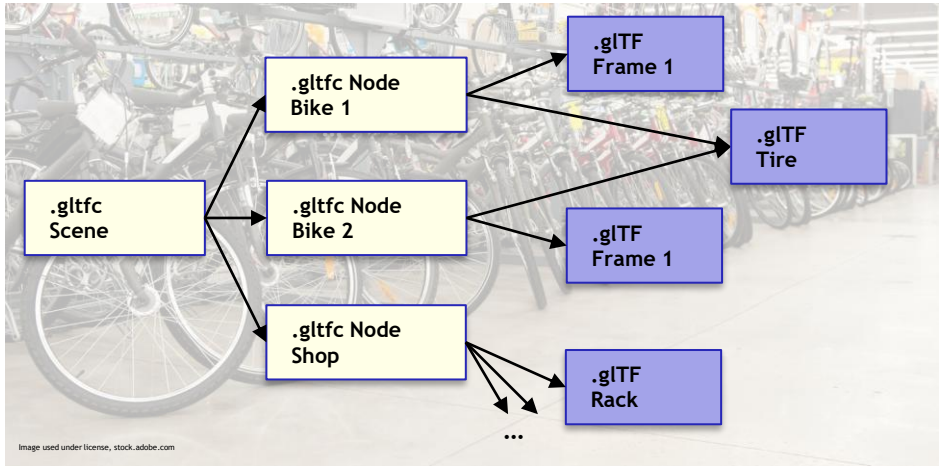


glTF External References

- **Compose scenes from multiple glTF assets**
  - External glTF files references
- **Designed for efficiency in transmission/delivery use cases**
  - Placement, Configuration, Cache Reuse, Personalization, Deferred Loading, LODs, Mesh Variants

Join in the ongoing discussions

https://github.com/KhronosGroup/glTF-External-Reference/tree/main/explainers



glTF Audio: KHR_audio_graph

- **Create a graph of interconnected audio node objects to the final audio output**
  - Manage audio routing, mixing, and processing which can be mapped to glTF asset
- **Superset of previous proposed audio extensions**
  - E.g., KHR_audio_emitter and MPEG_audio_spatial (including MPEG_media)
  - “Normalized” with Web Audio and popular game engines
- **Concepts**
  - Audio nodes
    - Source: audio data from internal or external sources or oscillator
    - Emitter: spatialized or non-spatialized
    - Listener
  - Audio processors
    - Gain, delay, pitch, channel splitter, channel merger, channel mixer, audio mixer
    - Filters: lowpass, highpass, bandpass, lowshelf, highshelf, peaking, notch, allpass
    - Reverb
  - Audio graph
    - DAG (directed acyclic graph)

Join in the ongoing discussions

https://github.com/facebook/glTF/tree/main/extensions/2.0/Khronos/KHR_audio_graph

Forum Mission – *Pragmatic* Metaverse Interoperability

***Assist* standards organizations in their mission to create interoperability for an open metaverse**

Gather broad input to inform standards development and drive visibility, adoption, testing and usage

Enable standards organizations to leverage each other's work and minimize overlaps/fragmentation

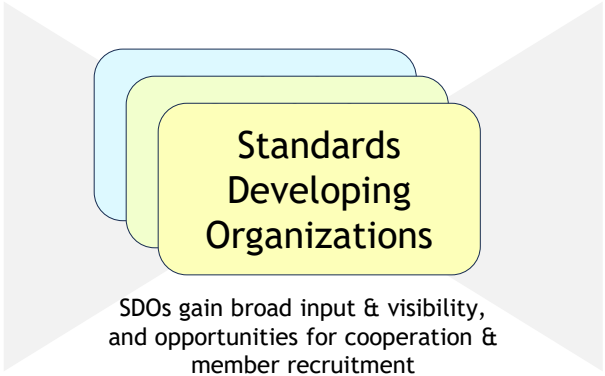
Pre-standardization

Identification of standardization gaps

Discovery of opportunities for standards organizations to leverage or align each others work

Create broad consensus on use cases and requirements

Generating reviews and feedback for draft specifications



SDOs gain broad input & visibility, and opportunities for cooperation & member recruitment

Post-standardization

Interoperability testing and testbeds

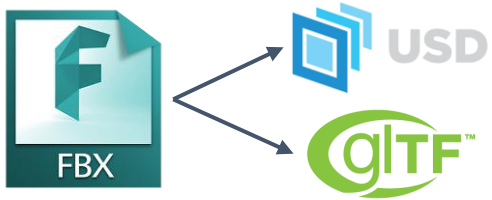
Development of conversion and layering tools

Publication of educational materials, reports, best practices & guidelines etc.

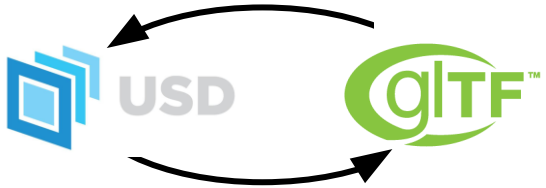
Bottom-up, pre- and post-standardization activities help address urgent 'pain points' creating a wavefront of short-term business opportunities on the road to an open, standards-based metaverse

USD and glTF Interoperability

- **Pragmatic projects to address immediate, real-world tooling pain points**
  - Discovery and exploration of interoperability issues
  - Testbeds to exercise existing or prototype interoperability solutions
  - Guidelines, documentation, or open-source tooling
  - USD/glTF specification recommendations to AOUSD/Khronos
- **Two initial projects**



FBX Migration



glTF ⇄ USD Interoperability

Any organization is encouraged to join the Forum to participate!
https://metaverse-standards.org/members/

One more thing: Gaussian Splats

- **Multiple Metaverse Standards Forum Townhalls on Gaussian Splat Standardization**
  - Radiance Fields, New York Times, Niantic, Meta, NVIDIA, PlayCanvas, Microsoft, Arrival.space, Waldek Technologies, Bentley Systems, Cesium, Hexagon, Esri, EARTHBRAIN, and JDULTRA
  - Blog post and recordings
- **Growing consensus that now IS the time to build extensible Splat standards**
  - glTF seeking to collaborate with other formats such as Niantic's .spz

Join in the ongoing discussions

https://github.com/KhronosGroup/glTF/issues/2454

glTF Roadmap Overview

	Baseline Today	Short Term **Roadmap (24-25)**	Roadmap **Discussions**
Geometry	Draco Mesh Compression meshopt Compression		Quads, SubDiv Vector Displacement Implicit spheres and strands? Nerfs, Gaussian Splats
External References		Placement, Configuration, Cache Reuse, Personalization, Deferred Loading, LODs, Mesh Variants	
Textures and Materials	KTX 2.0 textures w Basis Universal Material Variants PBR Core + PBR Extension Wave 1-4	PBR Wave 5: Subsurface Scattering MaterialX node graph update Video Textures, Procedural Textures, PBR: Diffuse Transmission,	HDR Universal Textures Material X Procedural Textures
Animations	Keyframe/Skinned	Blender-compatible animation	Multi-track animation/blending Skeletons, Rigs and Anchors Animation Compression
Lights	Punctual Point, spot, and directional		Rectangular Area Dome/Image
Interactivity		Node-based Behavior Graph	Node-based Behavior Graph with External References
Physics		Collisions, Rigid Bodies, Joints	Deformable Bodies
Audio		Playback (e.g., play, stop, loop), Spatial audio, Signal processing (gain, delay, pitch, reverb, filtering), multiple channels with splitting	

Join the ongoing roadmap discussions!

https://github.com/KhronosGroup/glTF/tree/main/extensions#in-progress-khronos-and-multi-vendor-extensions-and-projects