KHRONOS GROUP

OpenXR™

OpenXR BOF

Empowering Cross-Platform Immersive Experiences

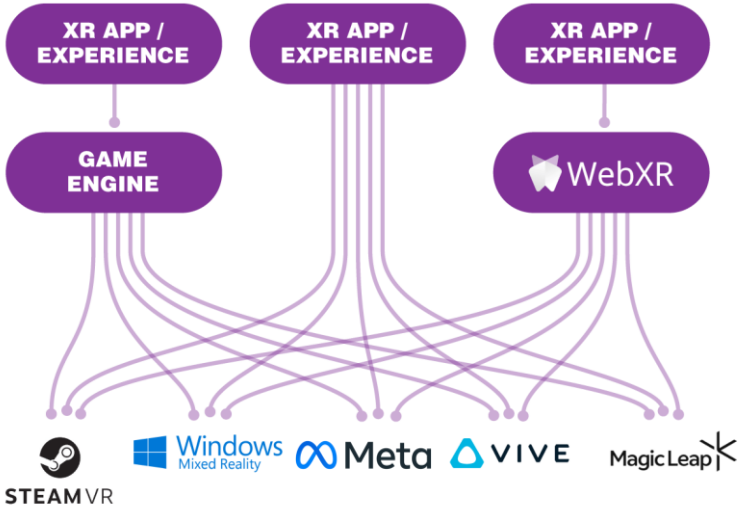
Neil Trevett and Arjun Dube
NVIDIA

What is OpenXR?

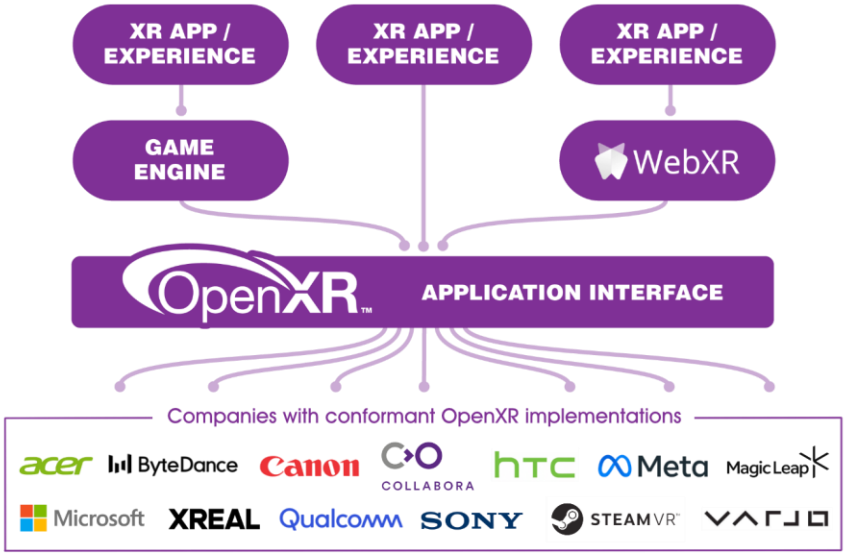
A royalty-free, cross-platform, cross-vendor open standard for high-performance access to Augmented Reality (AR) and Virtual Reality (VR) - collectively known as XR - devices and platforms

Empowering Cross-platform Immersive Experiences

OpenXR Cross-Platform Portability



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



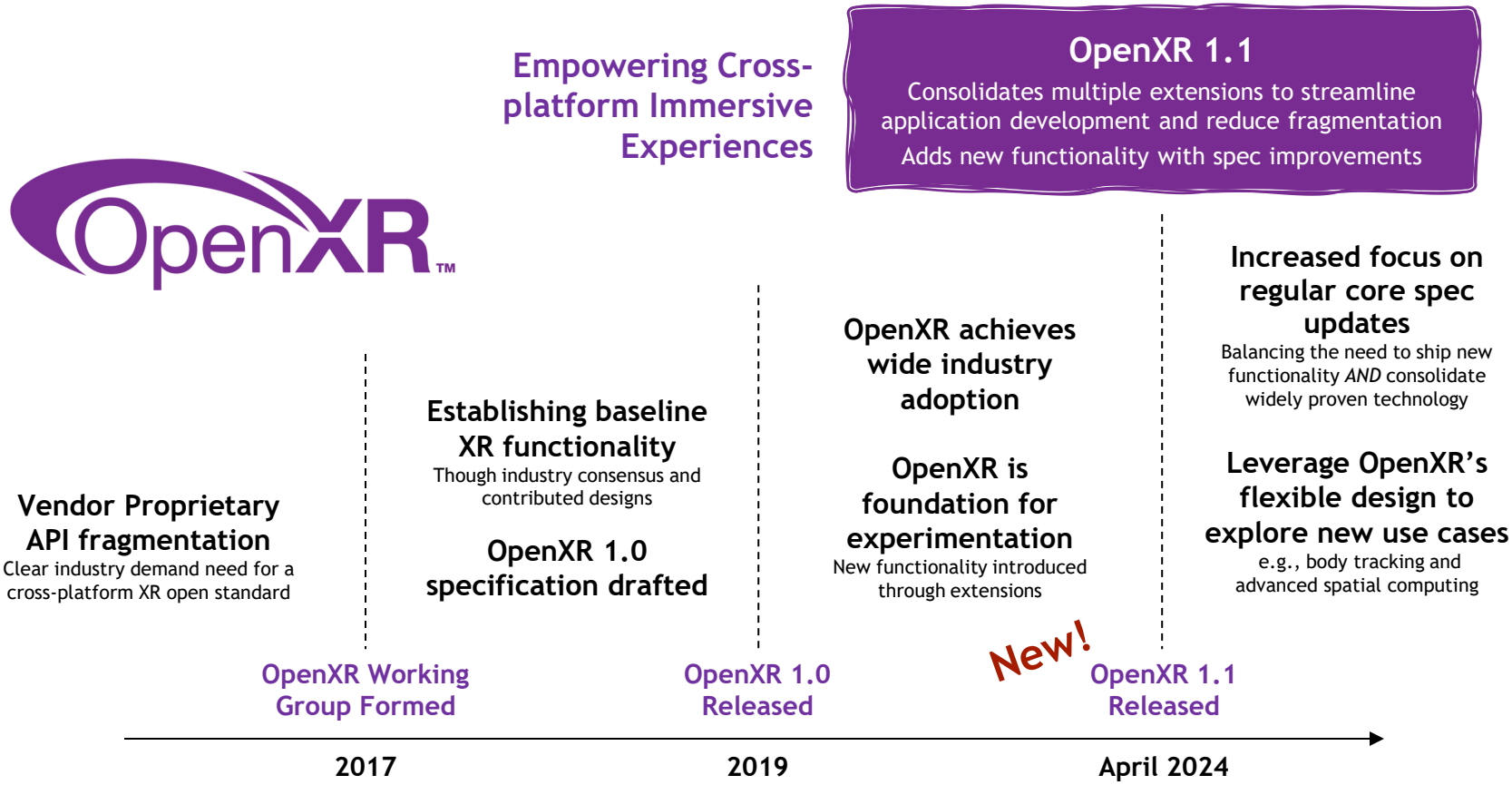
OpenXR provides a single cross-platform, high-performance API between applications and all conformant devices.

Applications and engines can portably access any OpenXR-conformant hardware

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 ByteDance	 SONY
Spatial Labs Display Series	Neo 3 and Pico 4	Spatial Reality Displays

The OpenXR Story So Far…



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

OpenXR Games and Applications

Blender uses OpenXR for native scene inspection in VR	Adobe Substance 3D Modeller uses OpenXR for VR support	Kitware's Paraview uses OpenXR for VR support	Meta Horizon Workrooms	OpenBrush uses OpenXR for Desktop and Quest support
War Thunder now uses OpenXR	Cubism uses OpenXR for VR support	Vermillion uses OpenXR for VR support	The Light Brigade uses OpenXR for VR support	XPlane12 uses OpenXR for VR support
Minecraft uses OpenXR for desktop VR support	Microsoft Flight Simulator uses OpenXR for VR support	Supports over 27 devices thanks to OpenXR	Phasmophobia switched from OpenVR to OpenXR	Beat Saber on PC uses OpenXR

Khronos and W3C: Bringing XR to the Web

XR Applications and Engines
use an API from both the 3D and XR Stacks

three.js babylon.JS unity UNREAL ENGINE GODOT **Engines**

WebGL WebGPU WebXR W3C®

OpenGL ES Vulkan OpenXR KHRONOS GROUP

3D Stack
Driving GPUs to render scenes and augmentations

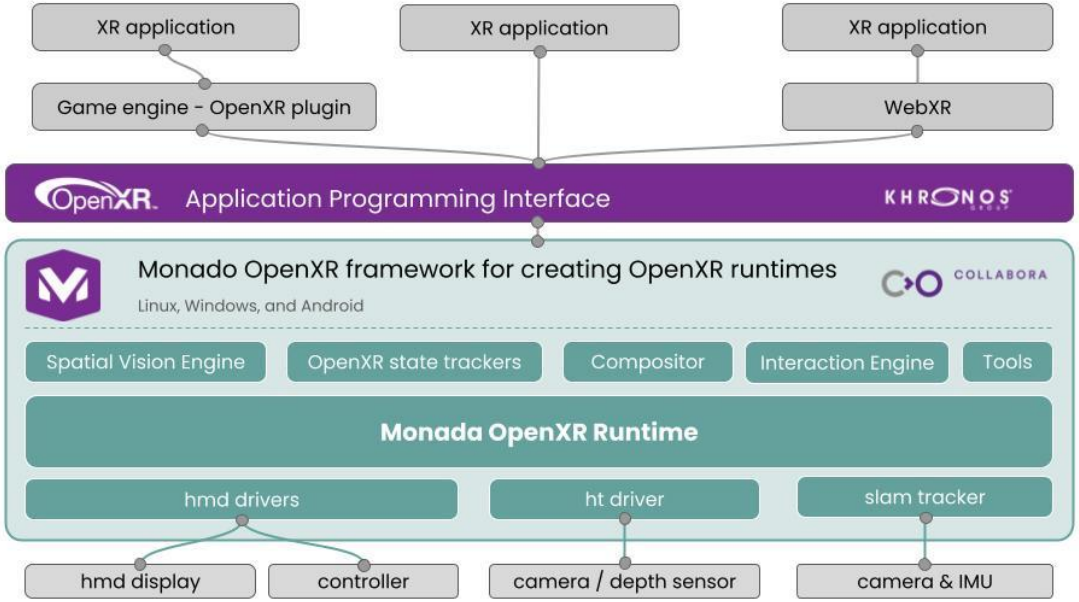
XR Stack
Handling XR Devices for creating UI

KHRONOS GROUP

Monado from Collabora

OPEN-SOURCE

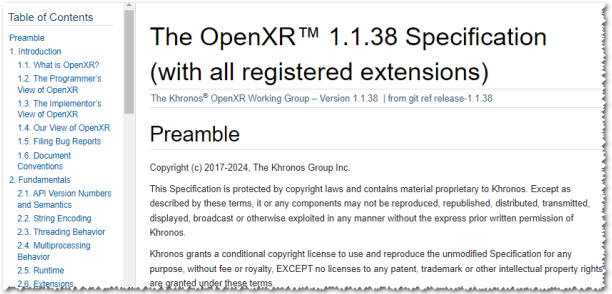
- Open source OpenXR Runtime and Framework
- Framework provides building blocks to simplify XR development



OpenXR Ecosystem Components

OpenXR Specification

Rigorous specification enables multiple implementations for pervasive industry adoption

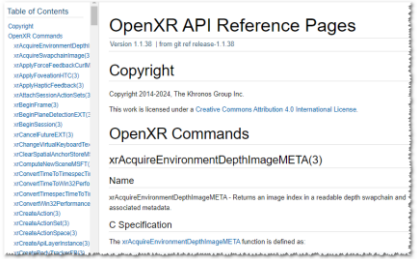


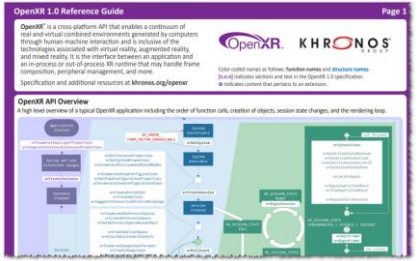
OpenXR Conformance Test Suite

Open source CTS promotes cross-vendor runtime implementation consistency

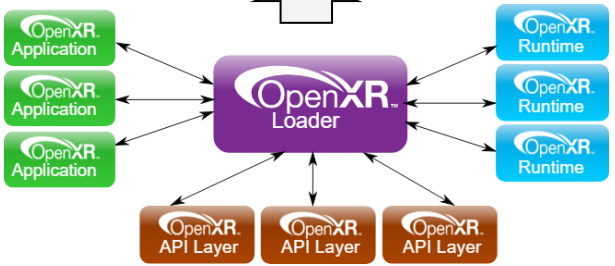
OpenXR Pages and Reference Guide

Developer Documentation





OpenXR Developers

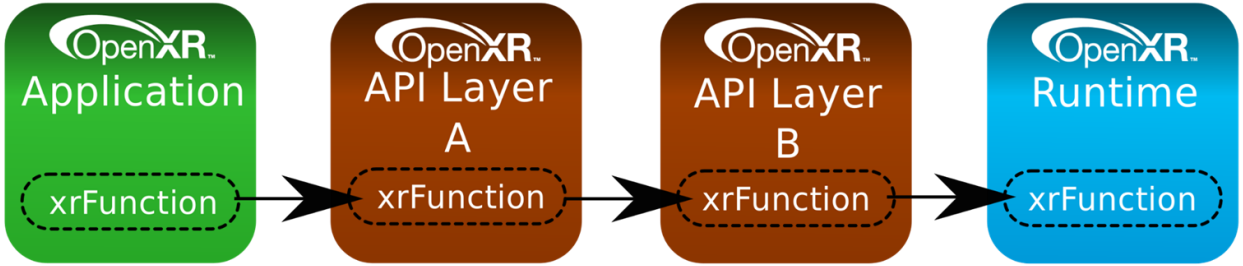


OpenXR SDK and Loader

Handles multiple OpenXR driver and API layers

OpenXR API Layers

- **Intercept and modify API calls between an application and the OpenXR runtime**
  - Modular components for validation, debug, or extended functionality
- **Isolate different functionalities into distinct layers without impacting runtime**
  - Flexible - add or remove features without altering the core application code
  - Debug or add logging capabilities by intercepting and analyzing API calls
  - Create custom behaviors or extensions layered onto existing APIs
- **Notable examples**
  - The OpenXR Core Validation API layer
  - UltraLeap OpenXR Hand Tracking Layer



OpenXR 1.1 Launch

Consolidates multiple extensions into OpenXR 1.1 core

Streamlined development and reduced fragmentation

Today

OpenXR 1.1 Feature Enhancements

Additional functionality
Spec clarifications and improvements

Continue leveraging OpenXR's Flexible Design

To foster innovation in developing extensions to explore new use cases

Ongoing

Drive for Immersive Experience Portability

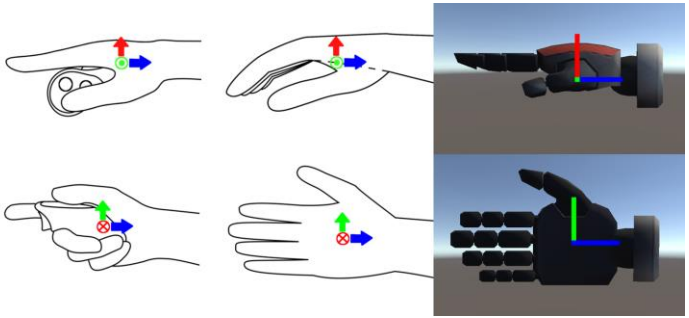
Increased focus on integrating widely adopted extensions into core for cross-platform portability

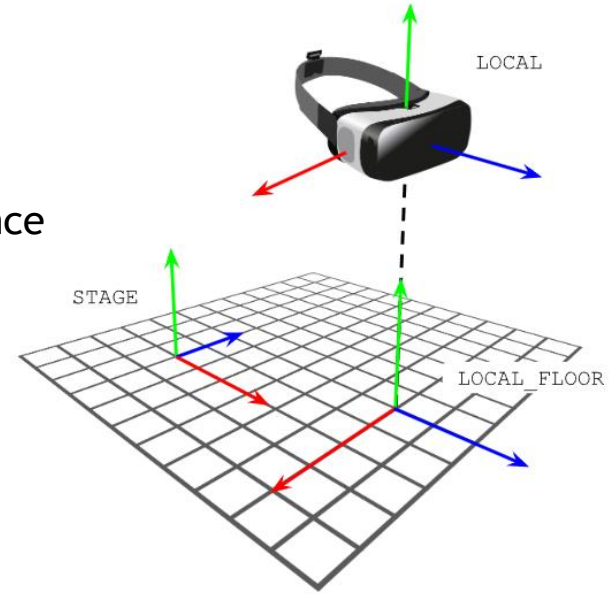
OpenXR 1.1 Key Extensions Promoted to Core

- **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**
  - 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

OpenXR 1.1 - April 2024

- **13 new interaction profiles added to the core spec**
- **5 extensions promoted to core**
- **Added notes for application developers:**
  - Benefits of foveated rendering
  - Which reference space to use
  - Which pose identifier to use
  - Unified extension error codes
- **Runtime clarifications**
  - Consistent cross-platform user and developer experience





OpenXR Spec + SDK 1.1.38 - June 2024

- **New XR_EXT_composition_layer_inverted_alpha vendor extension**
  - Allows runtimes to follow their conventions for transparency
- **Specification fixes and clarifications, including…**
  - Fix documentation for XrCompositionLayerPassthroughFB
  - Fix specification for XR_EXT_plane_detection

OpenXR CTS 1.1.37 - June 2024

- **Tests OpenXR 1.1 functionality by default**
  - Supports running in 1.0 mode for 1.0 submissions and to ensure backward compatibility
- **New interactive tests:**
  - `XR_KHR_composition_layer_equirect`
  - `XR_KHR_composition_layer_equirect2`
- **Improvements include:**
  - Interactive test improvements for haptics tests
  - Action test fixes and message clarifications
  - Grip surface parameter tuning

Coming Soon…

- **Extending hand tracking**
  - To include full body tracking
- **Enhanced handling of spatial entities**
  - Standardized methods to interact with the user's environment
  - Support for advanced spatial computing applications
- **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

OpenXR and Spatial Entities

- **Enhanced handling of spatial entities for advanced spatial computing applications**
  - Standardized methods to interact with the user's environment
- **Multiple spatial entity types**
  - Planes
  - Objects
  - World Meshes
  - Spatial Anchors
  - Marker Tracking (ArUco, AprilTag, QR code)
- **With BROAD development support from all the major players**
  - Expecting wide portability

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