KHRONOS GROUP

OpenXR™

OpenXR Ecosystem Update

Bringing to Life the Dream of Portable Native XR

July 2020

Khronos Standards for XR

3DCommerce

Create and deploy 3D assets and scenes

glTF

Vision and sensor processing, inferencing acceleration

SYCL

OpenVX

NNEF

OpenCL

OpenGL|ES

OpenGL

Vulkan

WebGL

High-performance, low-latency 3D Graphics

OpenXR

Portable access to native XR runtimes

OpenXR Ecosystem Updates

Khronos open sources conformance tests and launches Adopters Program

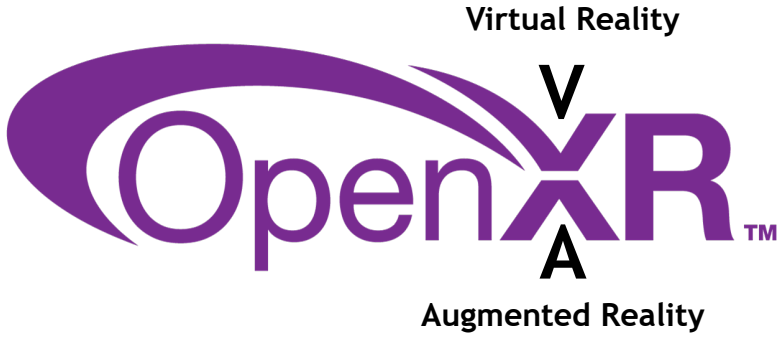
First officially conformant runtimes shipping from Microsoft and Oculus

Preview implementations from Valve, Varjo and Collabora

Hand and eye tracking cross-vendor extensions for advanced UI

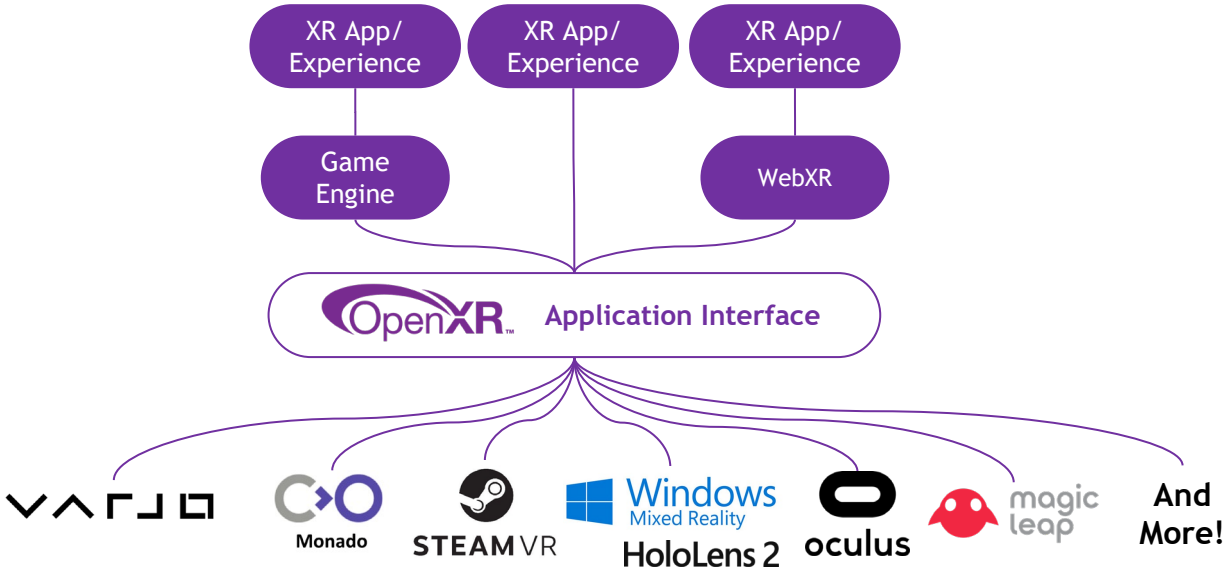
OpenXR being used by key games and open source software
Minecraft! Blender! WebXR!

Now is the time for application developers to leverage OpenXR for widespread deployment!



KHRONOS GROUP

XR Portability



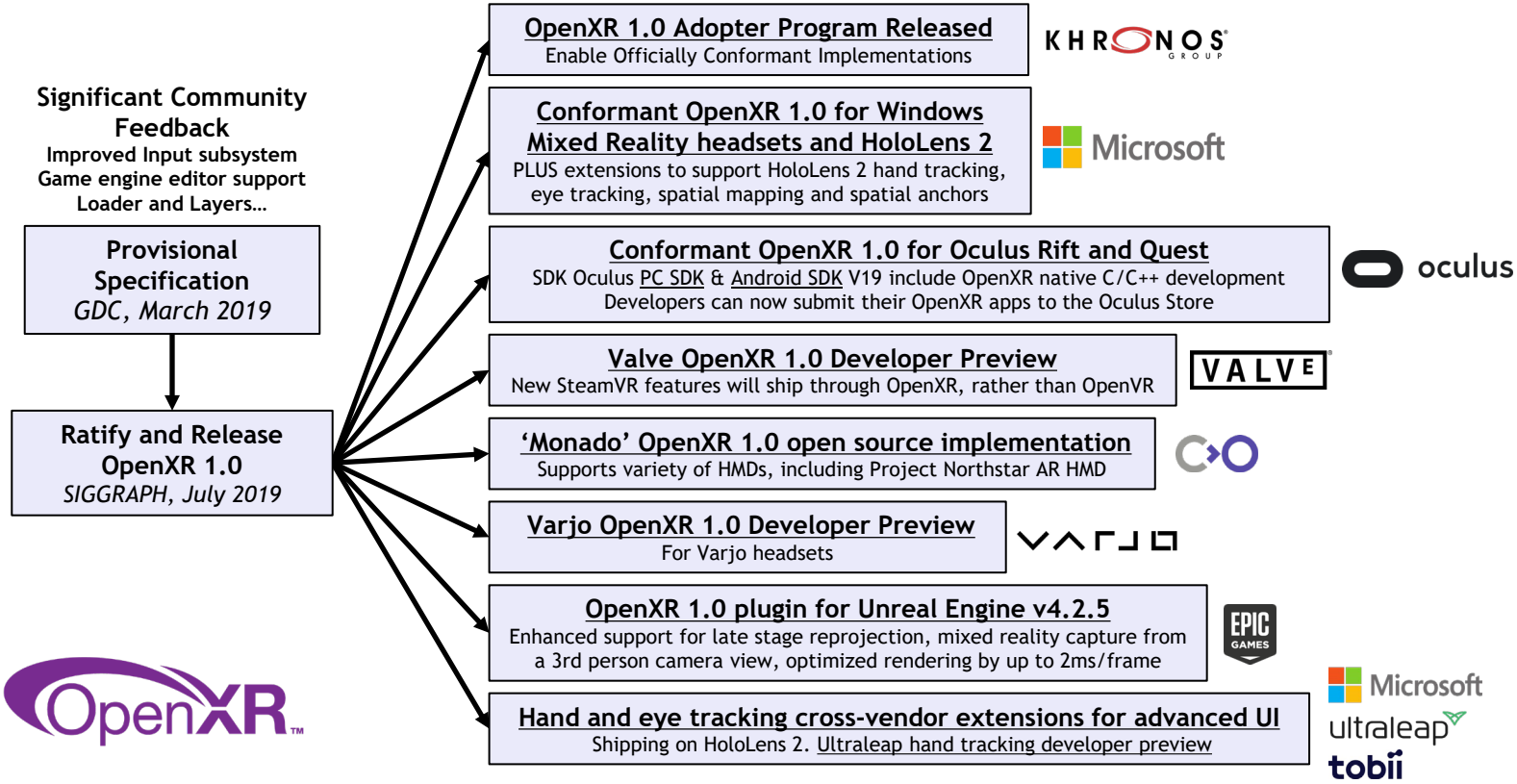
OpenXR provides cross-platform, high-performance access directly into XR device runtimes across multiple platforms

Widespread Industry Support

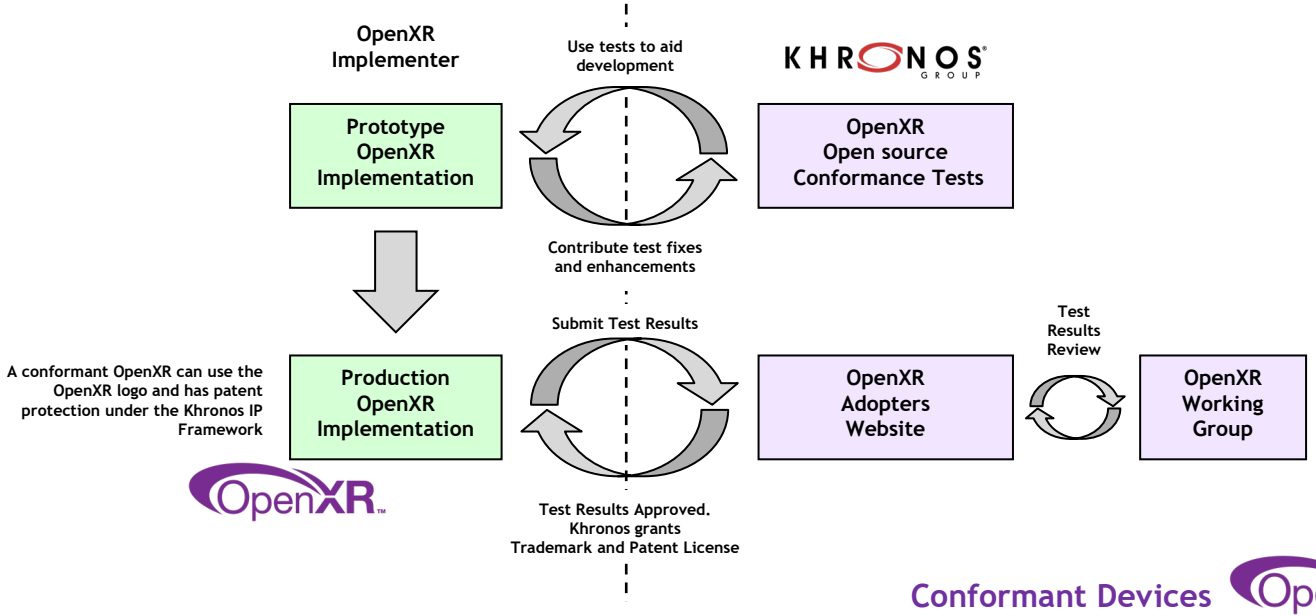
Companies publicly supporting OpenXR

OpenXR is a collaborative design that integrates many lessons from proprietary 'first-generation' XR APIs to create a new generation API with cutting-edge capabilities and a flexible, extensible, future-proof architecture

Broadening OpenXR 1.0 Availability



First Conformant OpenXR Devices



Conformant Devices OpenXR


Oculus Rift S


Oculus Quest


HoloLens 2


Windows Mixed Reality Headsets

OpenXR Architecture Flexibility

VR

OpenXR™

AR

VR

Windows 10

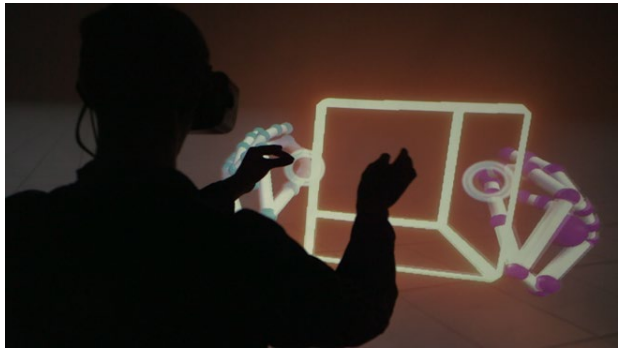
All In One

Tethered

Advanced UI Cross-Vendor OpenXR Extensions

Developers can build cross-platform applications that use advanced UI solutions from different technology vendors
OpenXR API layers can be used implement extensions

Hand Tracking



26 unique joints per hand for fully articulated hands visible to the user

Shipping on HoloLens 2
and Microsoft Hand Mesh Extension
for HoloLens 2 layers over it

Microsoft

Ultraleap developer preview available

ultraleap

Eye Tracking

Eye gaze interaction for intuitive interfaces
2-Step Interaction
Hand-eye coordination
Natural aiming

tobii

Shipping on HoloLens 2

Microsoft

OpenXR and Minecraft

Microsoft is excited to announce that Minecraft's new RenderDragon rendering engine is building its desktop VR support using OpenXR!

MINECRAFT

Microsoft

OpenXR and Open Source

Microsoft open source [OpenXR Samples] for Mixed Reality Developers, shows how to use OpenXR to access the full capabilities of HoloLens 2

README.md

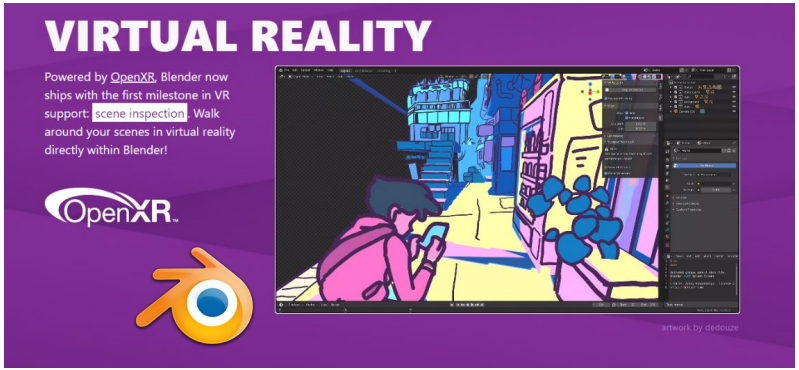
OpenXR Samples for Mixed Reality Developers

This repository contains OpenXR code samples tailored for developers who are familiar with and using the Visual Studio toolchain, e.g. HoloLens 2 developers.

These OpenXR samples use C++17 and D3D11. The same source code works across UWP applications running on HoloLens 2 and Win32 applications running on Windows Desktop with Windows Mixed Reality immersive headsets.

Prepare, build and run the samples

- Understand what is OpenXR and why OpenXR. Read the latest OpenXR 1.0 spec (HTML) and the latest openxr.h header file (Khronos GitHub).
- Update to Windows 10 May 2019 Update (1903) or later and Visual Studio 2017 or 2019. If you will be deploying to a HoloLens 2, you should install Visual Studio 2019 16.2 or later.
- Prepare a HoloLens 2 device or Windows Mixed Reality device.



[Blender 2.83] integrates OpenXR to deliver native VR scene inspection capabilities

WebXR

Chrome Hardware Support

Chrome supports WebXR on a variety of platforms and hardware. This page details the supported devices and required configuration (if any) to use them.

Chrome Compatibility Table

Device	OS	Runtime	Supported Session Modes	WebXR Support
Windows Mixed Reality PC headsets	Windows	Windows Mixed Reality	immersive-vr	Enabled in Chrome 79+
Daydream View Lenovo Mirage Solo	Android	Google VR	immersive-vr	Enabled in Chrome 79+
ARCore-compabile mobile devices	Android	ARCore	immersive-ar	Enabled in Chrome 81+
Oculus Rift Oculus Rift S	Windows	Oculus	immersive-vr	Experimental VR support
HTC Vive Valve Index Other OpenVR-compatible headsets	Windows	SteamVR	immersive-vr	Experimental VR support
OpenXR-compatible headsets	Windows	OpenXR	immersive-vr	Enabled in Chrome 81+, Requires the XR_EXT_win32_appcontainer_compatible extension

[Google Chromium 81 uses OpenXR] as its default backend for WebXR, enabling Google Chrome and Microsoft Edge browsers to use any OpenXR-compatible hardware

OpenXR is used with a 3D API

Application or Engine

High-performance, low-latency 3D rendering and composition*
- Multiview
- Context priority
- Front buffer rendering
- Tiled rendering (beam racing)
- Variable rate rendering

Display, composition and optical correction parameters

Cross-platform access to XR HMDs and sensors
- XR application lifecycle
- Input device discovery and events
- Sensor tracking and pose calculation
- Frame timing and display composition
- Haptics Control

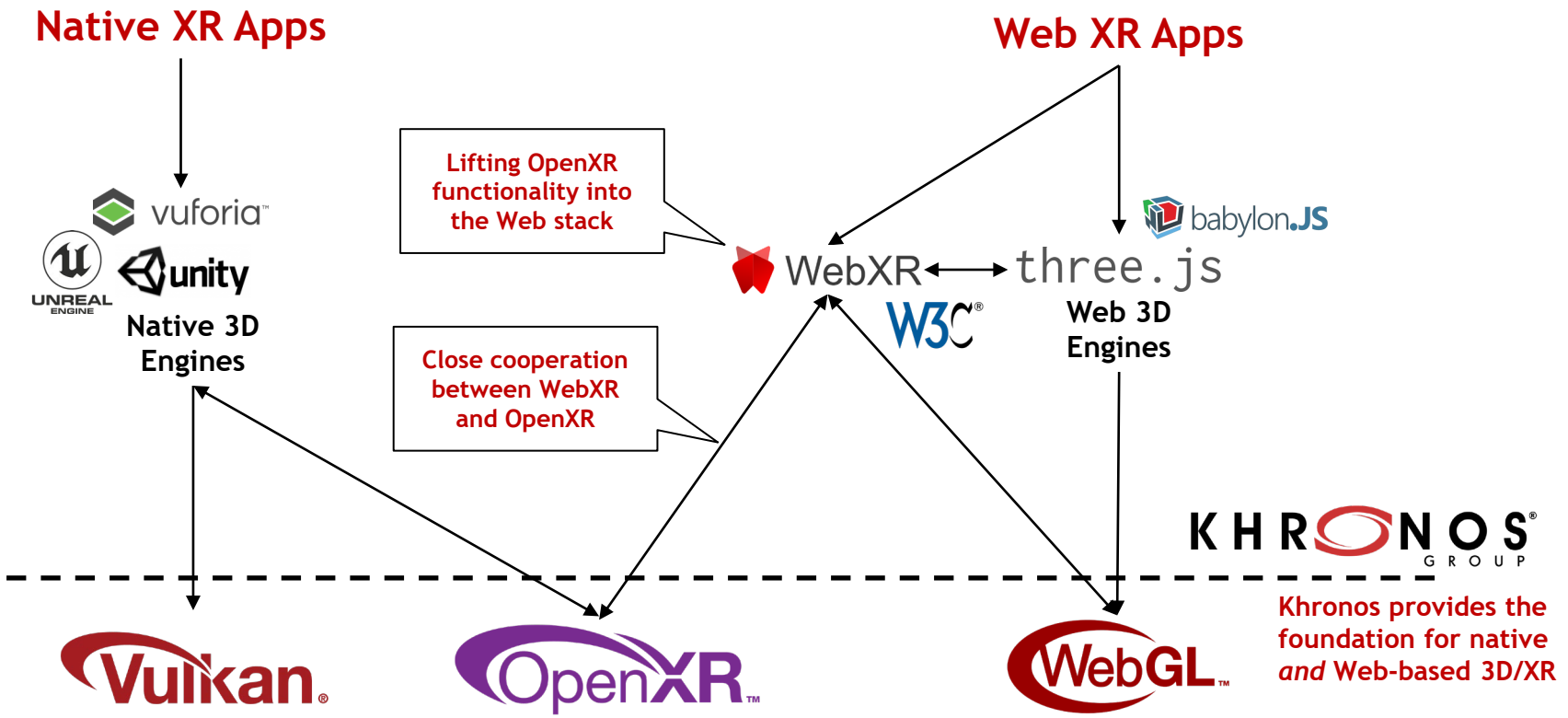
Vulkan®

OpenXR™

* OpenXR can be used with other 3D APIs such as Direct3D, OpenGL and OpenGL ES

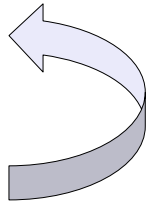
OpenXR is strongly influenced by Vulkan with a shared spec toolchain and support for API layers. OpenXR is a "lower-frequency" API than Vulkan and is a much smaller spec

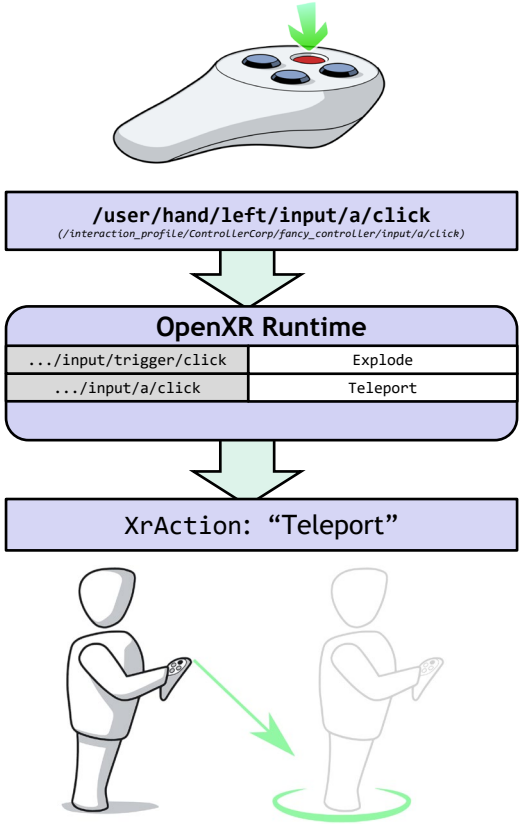
Bringing XR to the Web



Structure of an OpenXR App

- **Get started**
  - Create an Instance
  - Choose extensions, layers, bind to graphics API
- **Find out where/how to run**
  - Get HMD characteristics - mono/stereo, form factor etc..
- **Set up interaction/input handles**
  - Bind physical inputs to actions grab_object, teleport etc.
- **Prepare your immersive experience**
  - Create Session
  - Create Swapchain to drive the display
- **Participate in the frame loop**
  - Handle input and haptics
  - Manage swapchain to drive imagery to the display
  - Poll for events

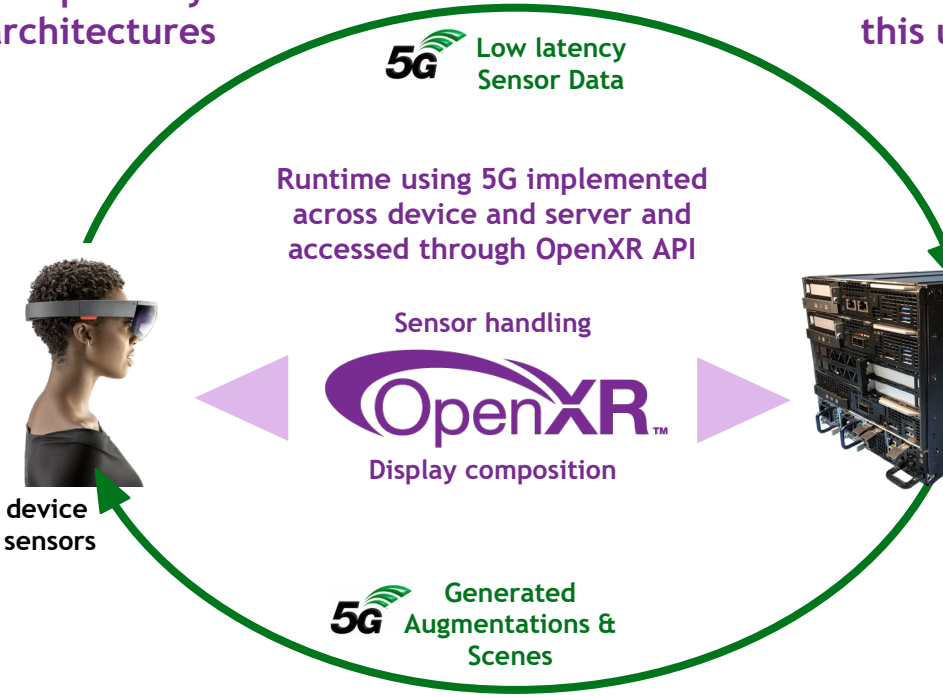




Cloud XR with 5G and OpenXR

OpenXR will enable AR applications to run portably on edge server architectures

Significant industry effort in developing this use case



Wireless mobile device with display and sensors

Vulkan

MEC (Multi-access Edge Computing) Server

1. Processes sensor data, can include machine learning for environmental lighting, occlusion, scene semantics, object reconstruction and UI
2. Generates imagery from 3D models, can include stereo, foveal rendering, ray-tracing, optics pre-distortion, varifocal processing

OpenXR Win-Win-Win