



OpenXR: On an HMD near you now!

*Brent E. Insko, PhD
Lead XR Architect at Intel &
OpenXR Working Group Chair*



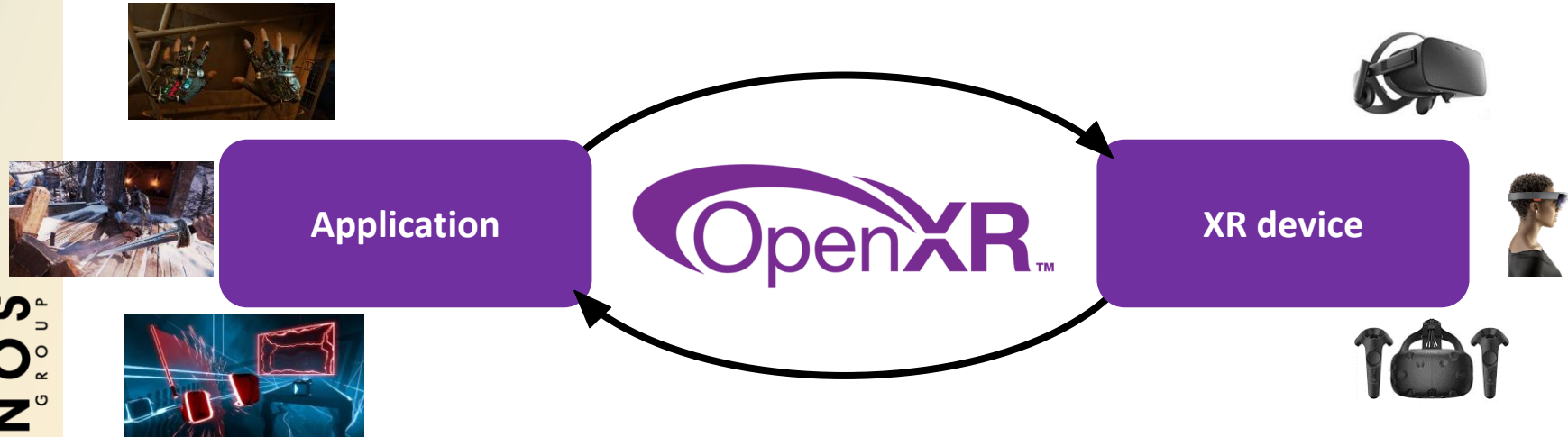
Agenda

- What is OpenXR?
- Why was it needed?
- What has been going on with OpenXR?
- Where are we going next?

What is OpenXR?

OpenXR is a royalty-free, open standard that provides high-performance access to Augmented Reality (AR) and Virtual Reality (VR)—collectively known as XR—platforms and devices.

OpenXR handles communication to and from an application and an XR device

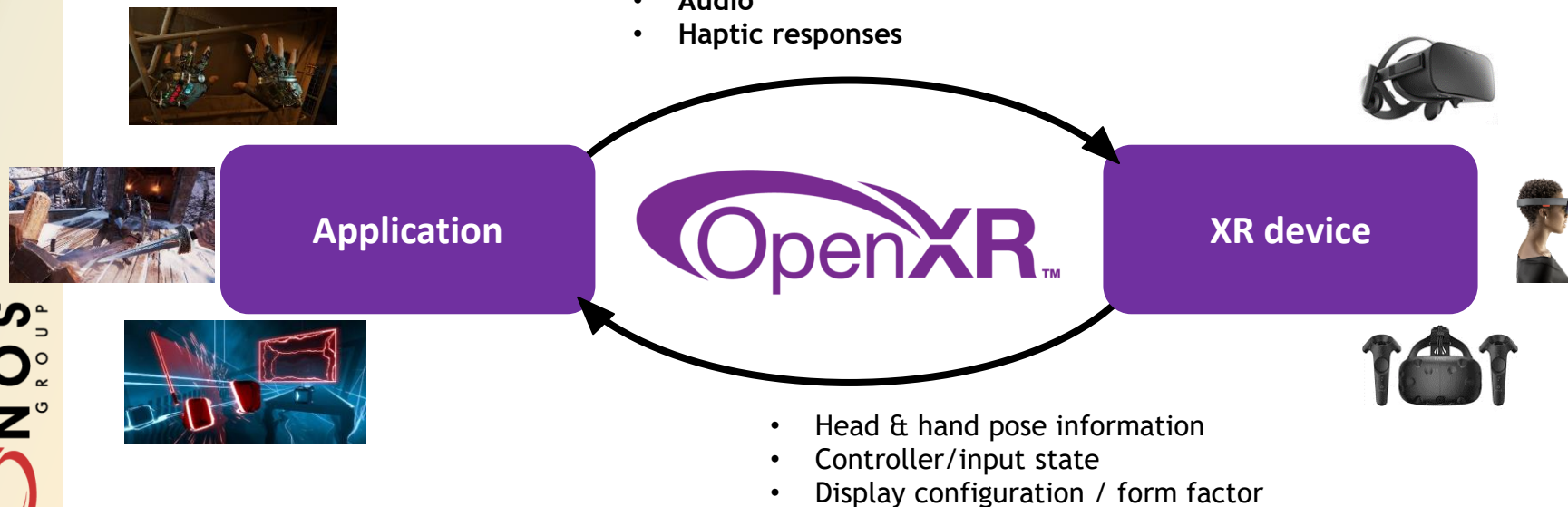


KRONOS[®] GROUP



- 

OpenXR handles communication to and from an application and an XR device



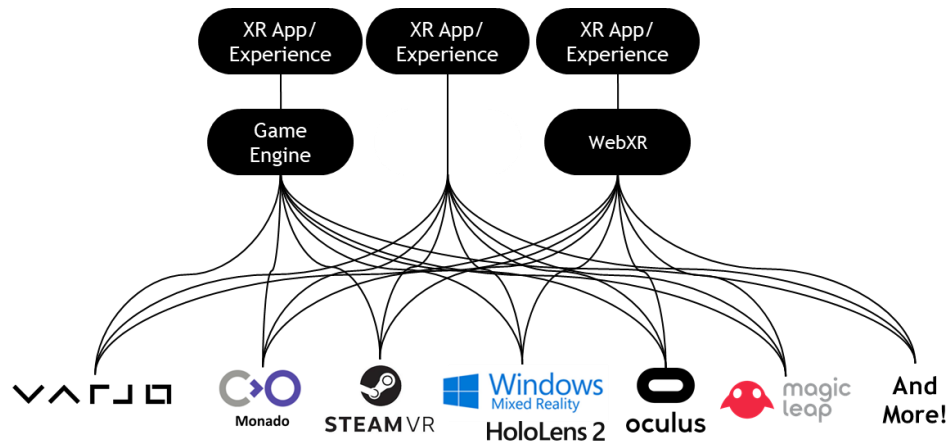
Why was OpenXR needed?



KHRONOS[®] GROUP

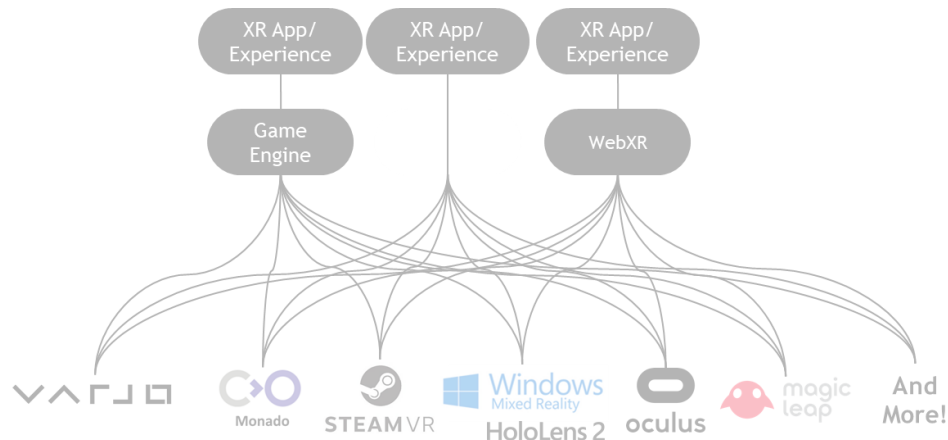


Cross Platform XR Application Development

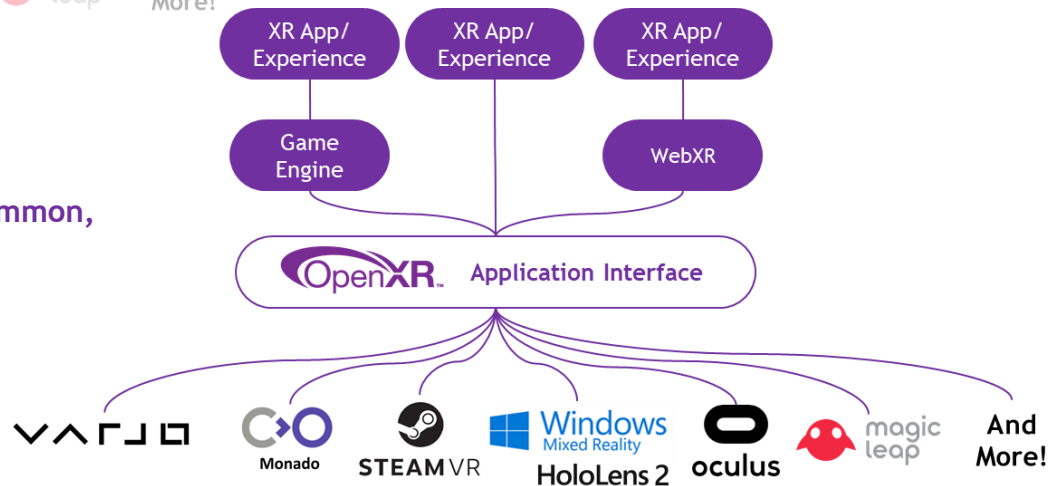


Before OpenXR, program to each platform's proprietary API

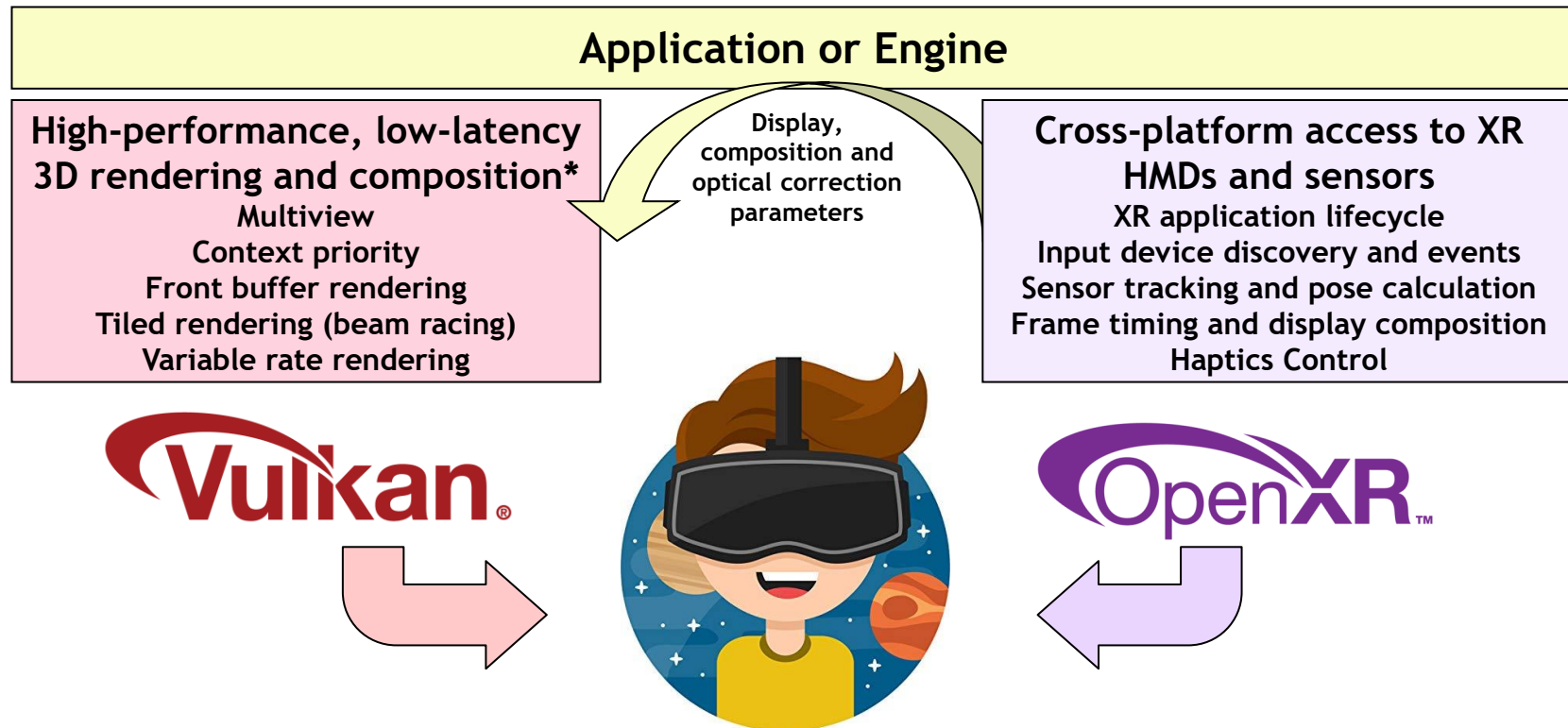
Cross Platform XR Application Development



With OpenXR, program to a single, common, high-performance API



OpenXR is used with a 3D API



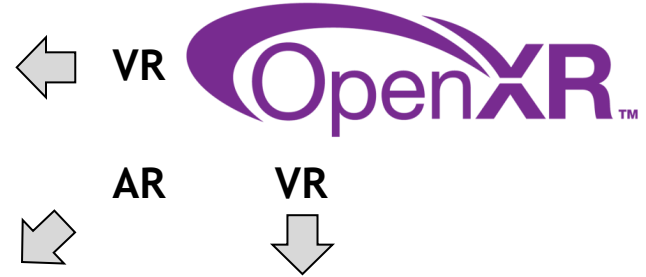
* OpenXR can be used with other 3D APIs such as Direct3D, OpenGL and OpenGL ES

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

OpenXR Flexibility



Oculus Quest / Quest 2



Microsoft
HoloLens 2

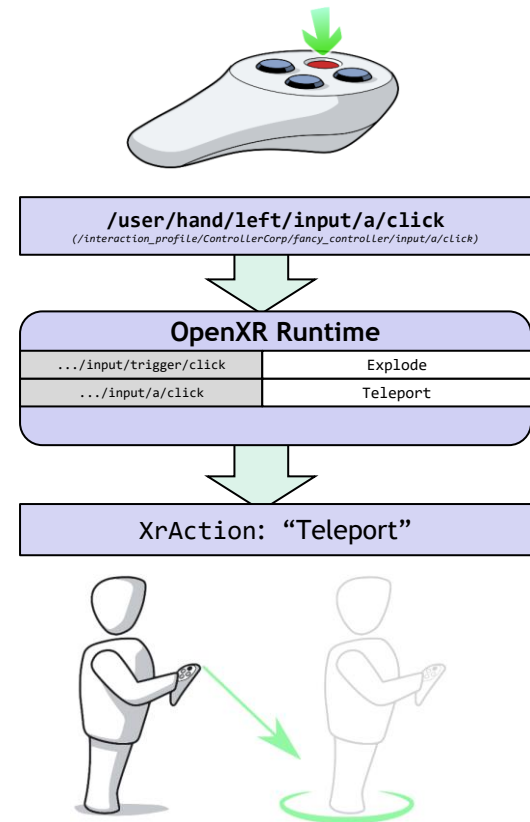
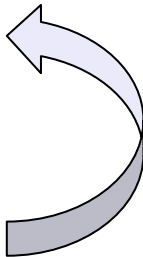
All In One



Tethered

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**
 - Update positions
 - Handle input and haptics
 - Manage swapchain to drive imagery to the display
 - Poll for events



OpenXR™ Development

Applications, Game Engines, Rendering Engines,
Browsers using OpenXR! - *Fall 2020 - 2021*

Conformant AR and VR platforms released - *Summer & Fall 2020*

Conformance suite developed and Adopters program released - *July 2020*

OpenXR 1.0 Specification Released - *SIGGRAPH, July 2019*

0.90 Provisional Specification Released - *GDC, March 2019*

OpenXR Working Group Formed - *2017*

Latest Developments with OpenXR

OpenXR Widespread Industry Adoption



Conformant Implementations



Microsoft HoloLens 2
Windows Mixed Reality Headsets



Oculus Rift S
Quest and Quest 2



HTC Vive Cosmos



STEAMVR™

Now Conformant!
Version 1.16

Valve SteamVR
Transitioning from OpenVR to OpenXR



Varjo
Preliminary support shipping now

ALL Varjo
Headsets
Conformant!
April 13th, 2021



Collabora
Open-source Implementation

Now Conformant!
Version 21.0.0

Engine Support



Unreal Engine
Support in 4.24. Optimizations in 4.25



Unity Engine
OpenXR Plugin Preview in 2020.2+

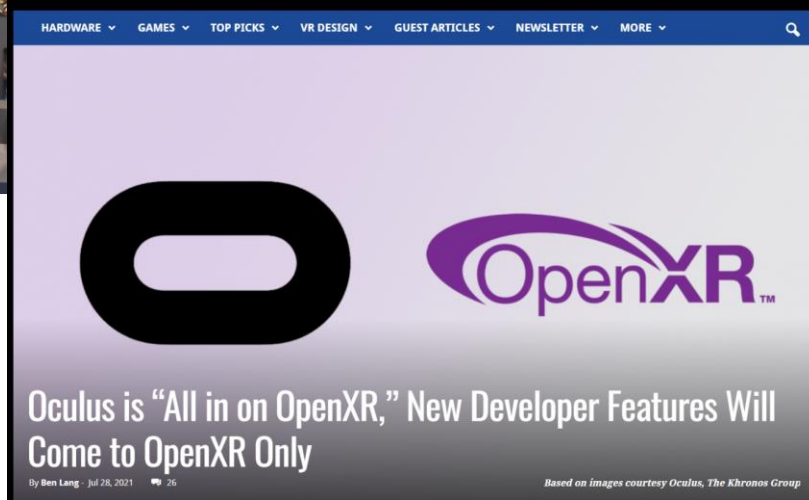
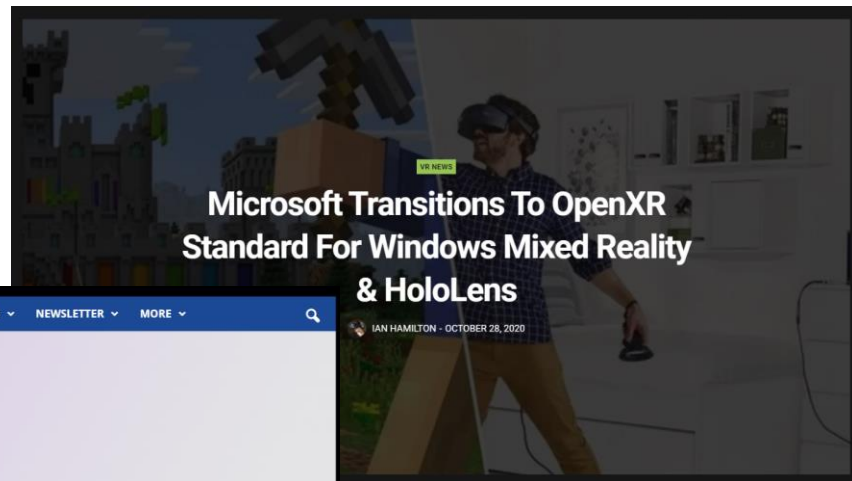
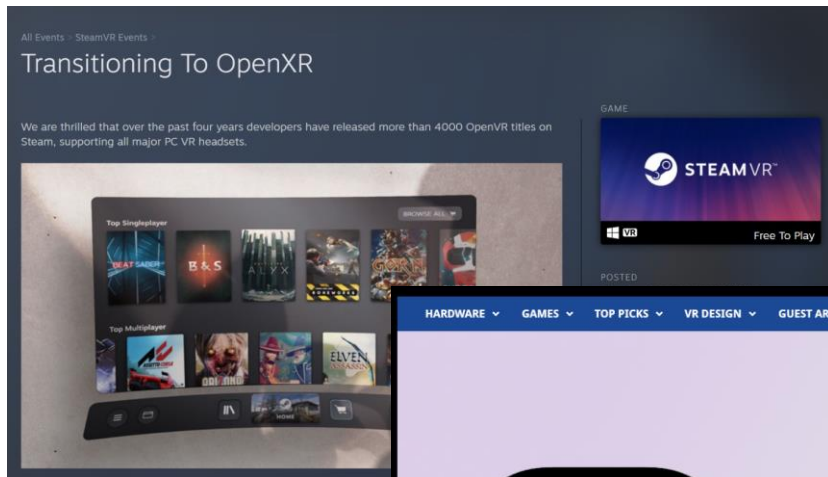
Just
Announced!



WebXR

Chromium 81 for Chrome and Edge Browsers
OpenXR default backend for WebXR

All the Major Platforms Transitioning to OpenXR for Future Development



OpenXR Applications



Minecraft's RenderDragon rendering engine uses OpenXR for desktop VR support



Microsoft Flight Simulator Sim Update 2 uses OpenXR for VR functionality



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 HTC	Windows	Windows Mixed Reality	immersive vr	Enabled in Chrome 79+
Daydream View	Android	Google VR	immersive vr	Enabled in Chrome 79+
ARCore compatible mobile devices	Android	ARCore	immersive vr	Enabled in Chrome 81+
Oculus Rift	Windows	Oculus	immersive vr	Experimental VR support
HTC Vive	Windows	SteamVR	immersive vr	Experimental VR support
Other OpenXR-compatible headsets	Windows	OpenXR	immersive vr	Enabled in Chrome 81+ Requires the XR_EXT_webxr_compatibility extension

Google Chromium

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

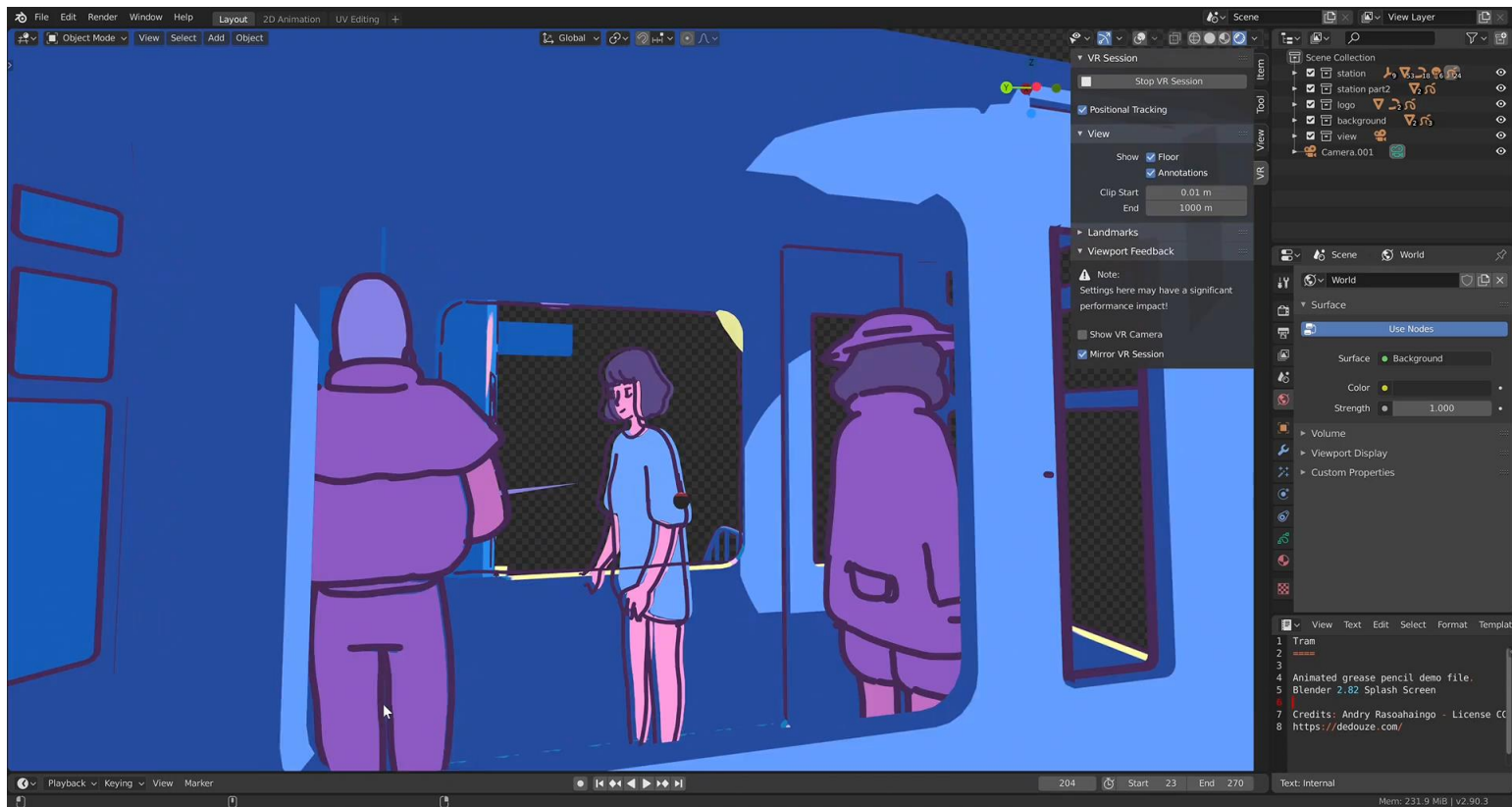


Google Chrome



Microsoft Edge

Blender VR Scene Inspection



Courtesy Blender 2.8.3 LTS release

Latest OpenXR Game

- Single Executable
- Multiple Platforms
- Built in Unity



Nominated for 'VR Game of the Year' at the 2021 VR Awards, 'Zombieland VR: Headshot Fever'

[XR Games](#), [Sony Pictures Virtual Reality](#)

What's Next for OpenXR?

Extensions

Core Standard

Core concepts that are fundamental to the specification for all use cases

Examples: Instance management, tracking, frame timing

KHR Extensions

Functionality that a large class of runtimes will likely implement

Examples: Platform support , Graphic API Extensions

EXT Extensions

Functionality that a few runtimes might implement

Examples: Performance Settings, Thermals, Debug Utils

Vendor Extensions

Functionality that is limited to a specific vendor

Examples: Device specific functionality

16 KHR Extensions at Launch

- **Platform-specific support:**
 - KHR_android_create_instance
 - KHR_android_surface_swapchain
 - KHR_android_thread_settings
- **Support for specific XR layer types:**
 - KHR_composition_layer_cube
 - KHR_composition_layer_cylinder
 - KHR_composition_layer_depth
 - KHR_composition_layer_equirect
- **Performance improvement by masking non-visible portions of the display:**
 - KHR_visibility_mask
- **Graphics API support:**
 - KHR_D3D11_enable
 - KHR_D3D12_enable
 - KHR_opengl_enable
 - KHR_opengl_es_enable
 - KHR_vulkan_enable
 - KHR_vulkan_swapchain_format_list
- **Time Conversion functions:**
 - KHR_convert_timespec_time
 - KHR_win32_convert_performance_counter_time

72 Current Extensions

XR_KHR_android_create_instance
XR_KHR_android_surface_swapchain
XR_KHR_android_thread_settings
XR_KHR_binding_modification
XR_KHR_composition_layer_color_scale_bias
XR_KHR_composition_layer_cube
XR_KHR_composition_layer_cylinder
XR_KHR_composition_layer_depth
XR_KHR_composition_layer_equirect
XR_KHR_composition_layer_equirect2
XR_KHR_convert_timespec_time
XR_KHR_D3D11_enable
XR_KHR_D3D12_enable
XR_KHR_loader_init
XR_KHR_loader_init_android
XR_KHR_opengl_enable
XR_KHR_opengl_es_enable
XR_KHR_swapchain_usage_input_attachment_bit
XR_KHR_visibility_mask
XR_KHR_vulkan_enable
XR_KHR_vulkan_enable2
XR_KHR_vulkan_swapchain_format_list
XR_KHR_win32_convert_performance_counter_time
XR_EXT_conformance_automation
XR_EXT_debug_utils
XR_EXT_eye_gaze_interaction
XR_EXT_hand_joints_motion_range
XR_EXT_hand_tracking
XR_EXT_hp_mixed_reality_controller
XR_EXT_performance_settings
XR_EXT_samsung_odyssey_controller
XR_EXT_thermal_query
XR_EXT_view_configuration_depth_range
XR_EXT_win32_appcontainer_compatible
XR_EPIC_view_configuration_fov

XR_FB_android_surface_swapchain_create
XR_FB_color_space
XR_FB_composition_layer_alpha_blend
XR_FB_composition_layer_image_layout
XR_FB_composition_layer_secure_content
XR_FB_display_refresh_rate
XR_FB_foveation
XR_FB_foveation_configuration
XR_FB_foveation_vulkan
XR_FB_swapchain_update_state
XR_FB_swapchain_update_state_android_surface
XR_FB_swapchain_update_state_opengl_es
XR_FB_swapchain_update_state_vulkan
XR_HTC_vive_cosmos_controller_interaction
XR_HUAWEI_controller_interaction
XR_MND_headless
XR_MSFT_composition_layer_reprojection
XR_MSFT_controller_model
XR_MSFT_first_person_observer
XR_MSFT_hand_interaction
XR_MSFT_hand_tracking_mesh
XR_MSFT_holographic_window_attachment
XR_MSFT_perception_anchor_interop
XR_MSFT_scene_understanding
XR_MSFT_scene_understanding_serialization
XR_MSFT_secondary_view_configuration
XR_MSFT_spatial_anchor
XR_MSFT_spatial_anchor_persistence
XR_MSFT_spatial_graph_bridge
XR_MSFT_unbounded_reference_space
XR_OCULUS_android_session_state_enable
XR_OCULUS_audio_device_guid
XR_VALVE_analog_threshold
XR_VARJO_composition_layer_depth_test
XR_VARJO_environment_depth_estimation
XR_VARJO_foveated_rendering
XR_VARJO_quad_views

© The Khronos® Group Inc. 2021 - Page 24

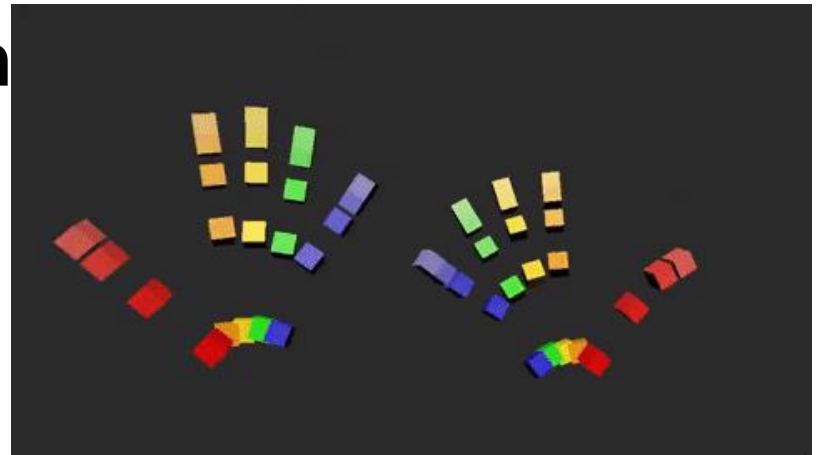
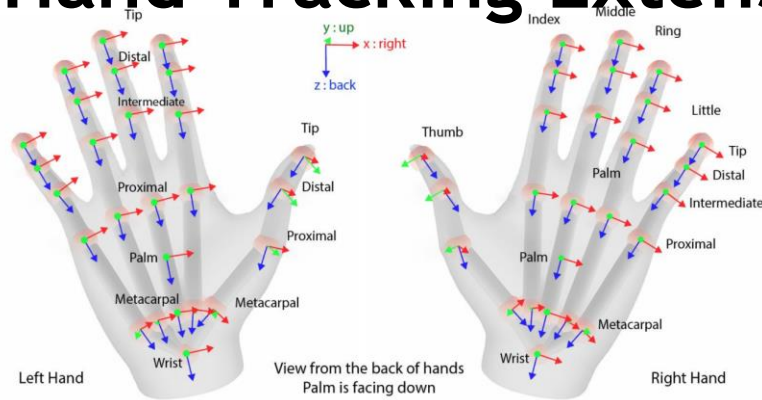
Let's talk about a few of the extensions...

OpenXR Overlays Extension

- *Simultaneous in-world* applications in OpenXR
- Can overlay one app's 2D content "on top" of an existing application's content
- Overlay applications add a rich content into other XR Applications: e.g. Desktop OS windows in-world, In-game HUD, Virtual keyboard, Chat (e.g. [Pluto VR's](#) Pluto)

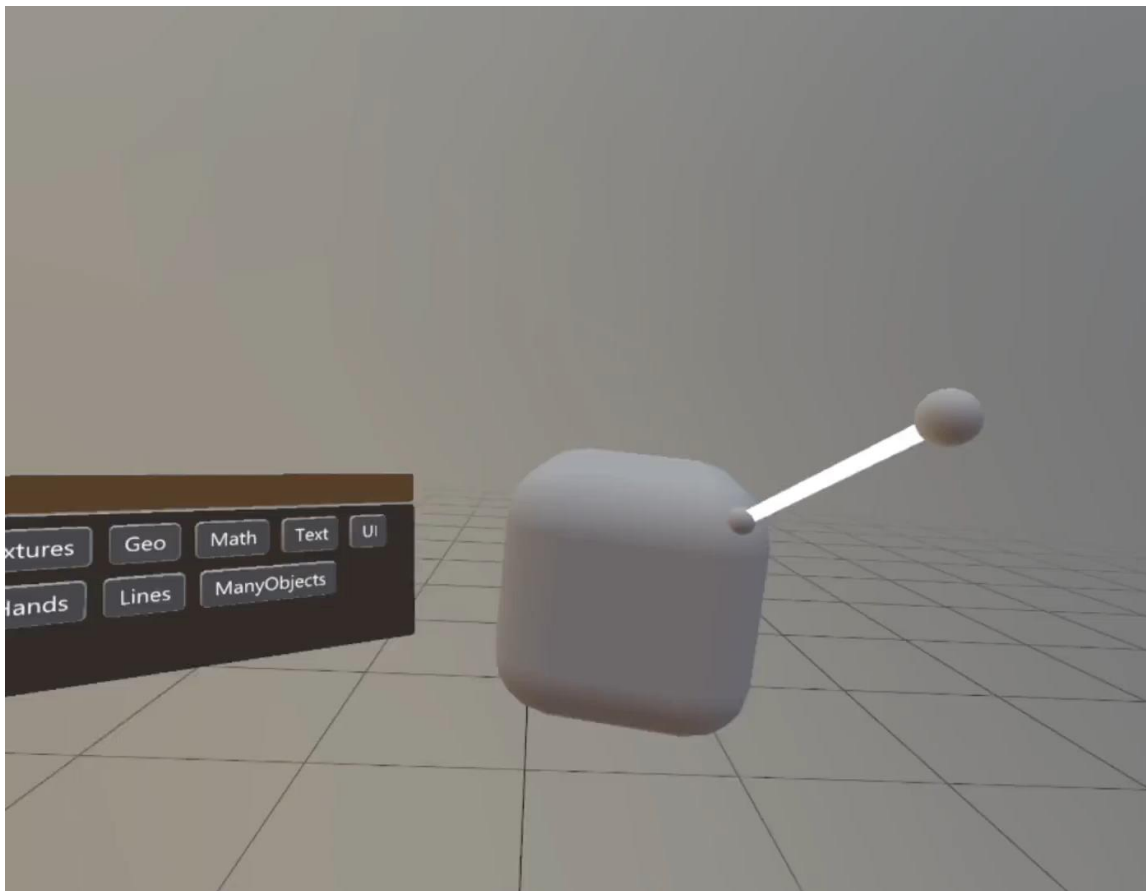


Hand Tracking Extension



- Standardize the **definition of a common set of hand joints**
- Five fingers are named as: thumb, index, middle, ring and little.
- Each finger has 5 joints: tip, distal, intermediate, proximal, metacarpal, except thumb lacks intermediate joint.
- Wrist joint - the pivot point of hand motion & Palm joint - the center of hand and easy access to palm orientation.
- This extension fits both VR and AR experiences.

Demo Video



stereokit

Library

+

OpenXR™

API

+

ultraleap

OpenXR API Layer

+



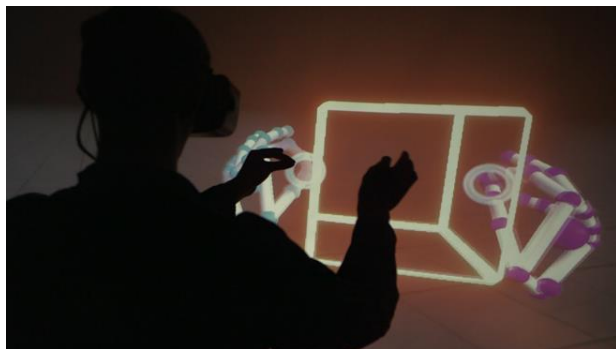
Windows Mixed
Reality

OpenXR Runtime

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



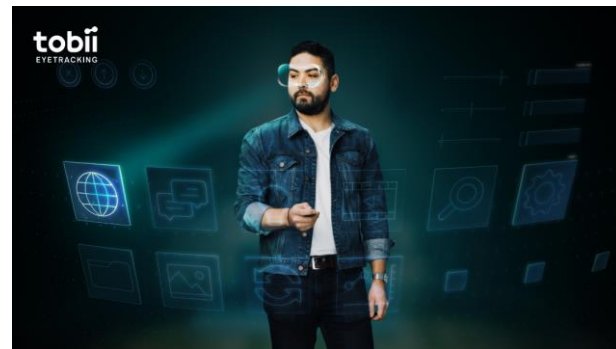
26 unique joints per hand for fully articulated hands visible to the user

Shipping on HoloLens 2
and [Microsoft Hand Mesh Extension](#)
for HoloLens 2 layers over it

[Ultraleap developer preview](#) available



Eye Tracking



Eye gaze interaction for intuitive interfaces

2-Step Interaction

Hand-eye coordination
Natural aiming



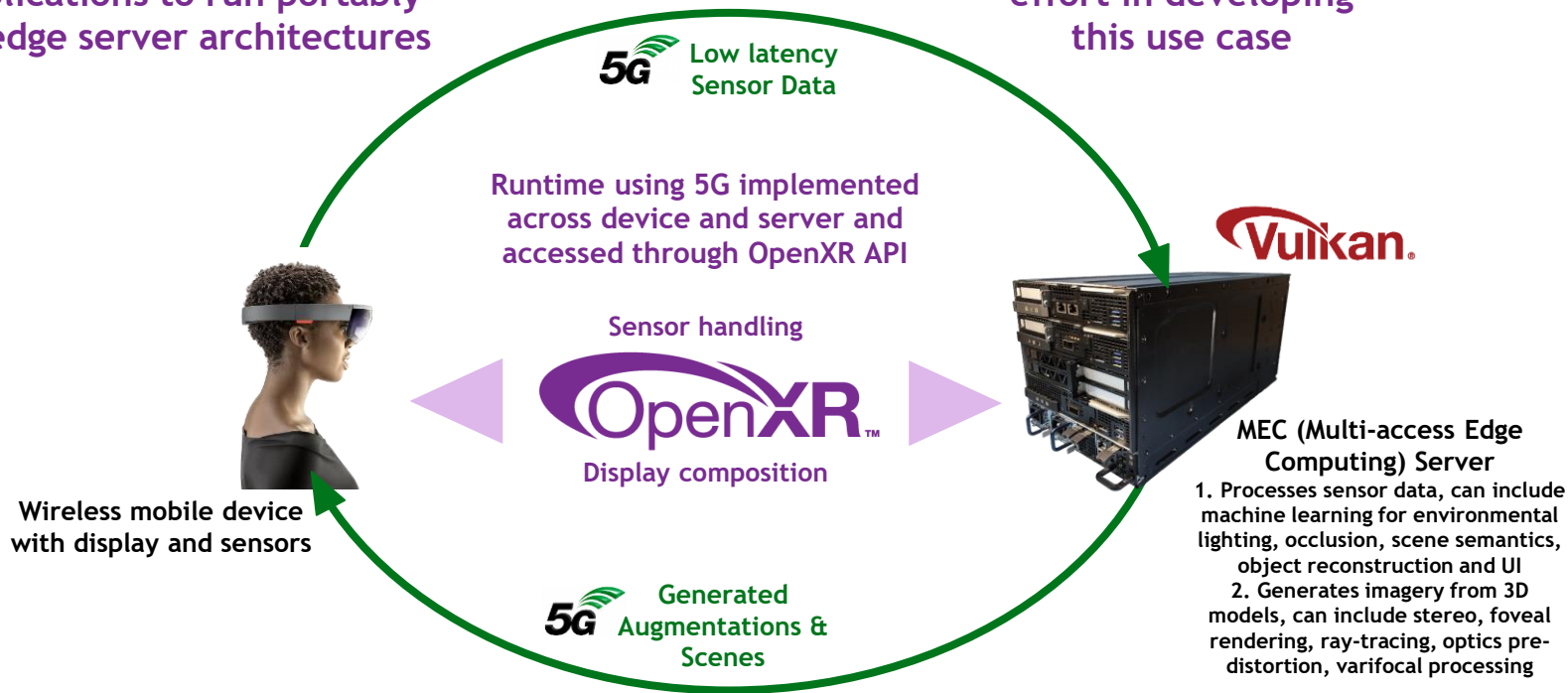
Shipping on HoloLens 2



OpenXR and Edge Server Applications with 5G

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

Significant industry effort in developing this use case



- 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

What is in the future for of OpenXR?

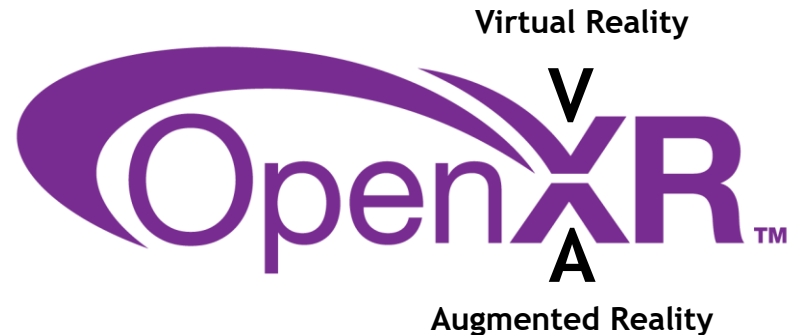
- Clarifying and fixing a few things from OpenXR 1.0
- Expanding support for AR and MR
 - Scene understanding
- Additional User input/output capabilities
 - Expanded haptics support
 - Expanded body tracking support
 - Expanded controller and device support

With OpenXR on an HMD near you, now is the time to get involved!

Officially conformant runtimes from:
Microsoft, Oculus, HTC, Valve, Varjo and Collabora

OpenXR being used by real software:
Unreal! Blender! Chrome! Edge! Unity! Flight Simulator! Minecraft! Games!

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

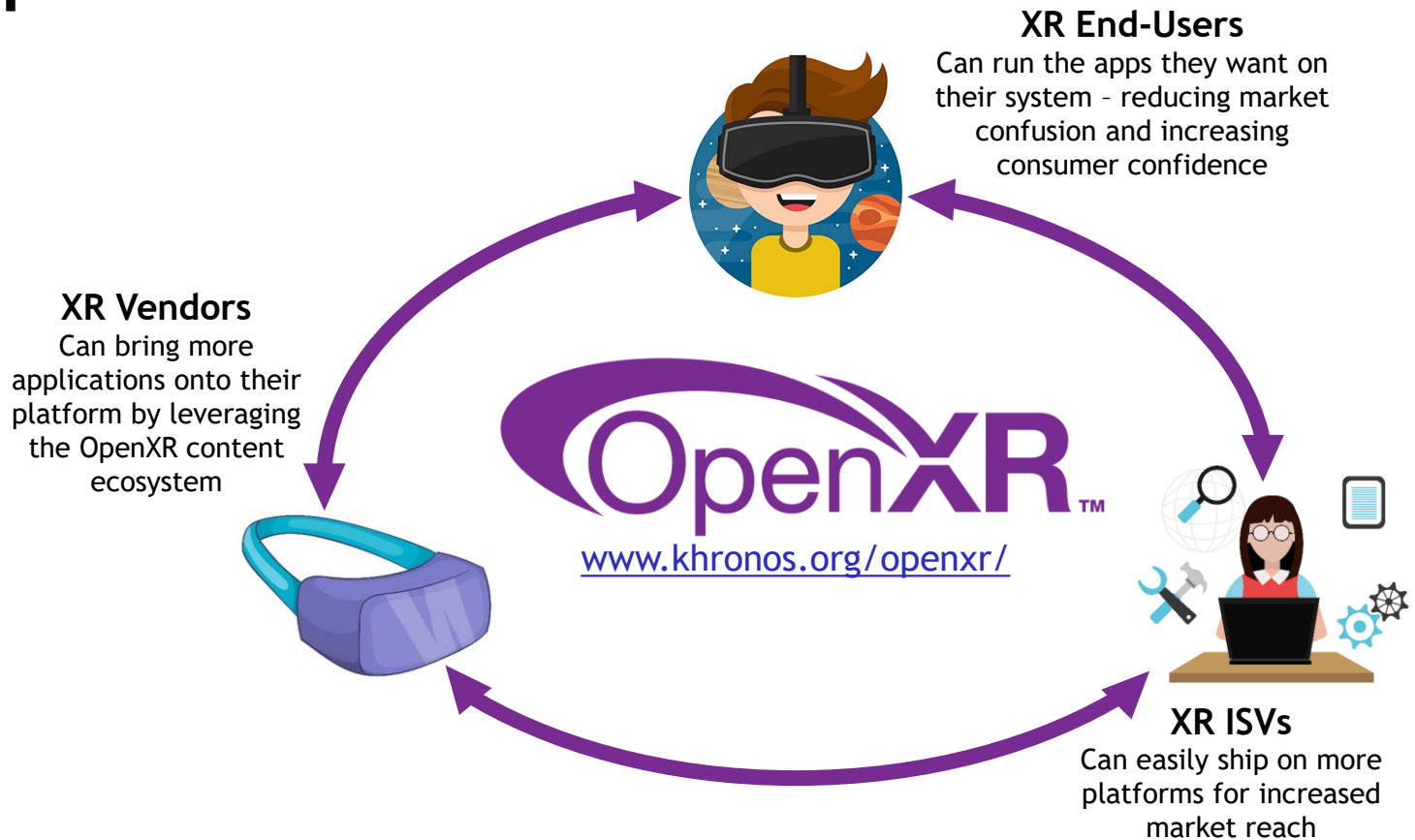


Widespread Industry Support



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

OpenXR Win-Win-Win





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





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

