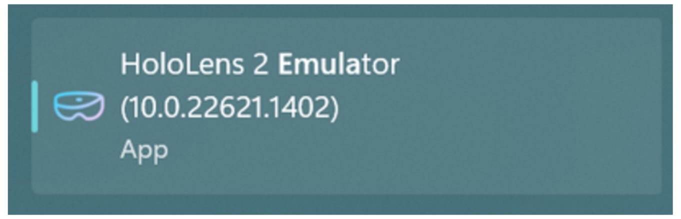
Dataflow Chart (From Left to Right)

HoleLens 2 (on device) / Emulator

1. Runs the Unity application with hologram and 3D objects
2. **Outputs raw gaze data (monocular signal)**



HoloLens 2 Emulator
(10.0.22621.1402)
App

1. Runs the Unity application with hologram and 3D objects
2. **Outputs raw gaze data (monocular signal) using mouse as input**

Unity Application

OpenXR

1. Abstracts the hardware specific implementations
2. **Takes raw data and does preliminary processing (integrates Calibration, Noise Reduction, IMU data). Then produces (standard transformed coordinates, gaze vector)**
3. Available through Mixed Reality OpenXR plugin on Unity
4. Microsoft provides vendor extensions for hand-tracking
5. Has cross-vendor extensions for gaze data

MRTK MIXED REALITY TOOLKIT

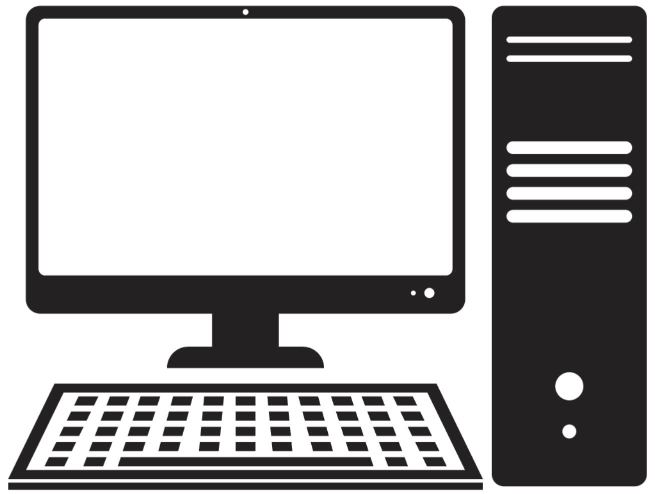
1. Abstracts the different input APIs into a unified interface
2. **Makes OpenXR gaze data available to Unity and for other uses (exporting, saving) to (json, csv, etc.)**
3. Data collection for further analysis and research done here.

unity

1. Has all holograms and 3D objects
2. Uses input data from MRTK to update the scene

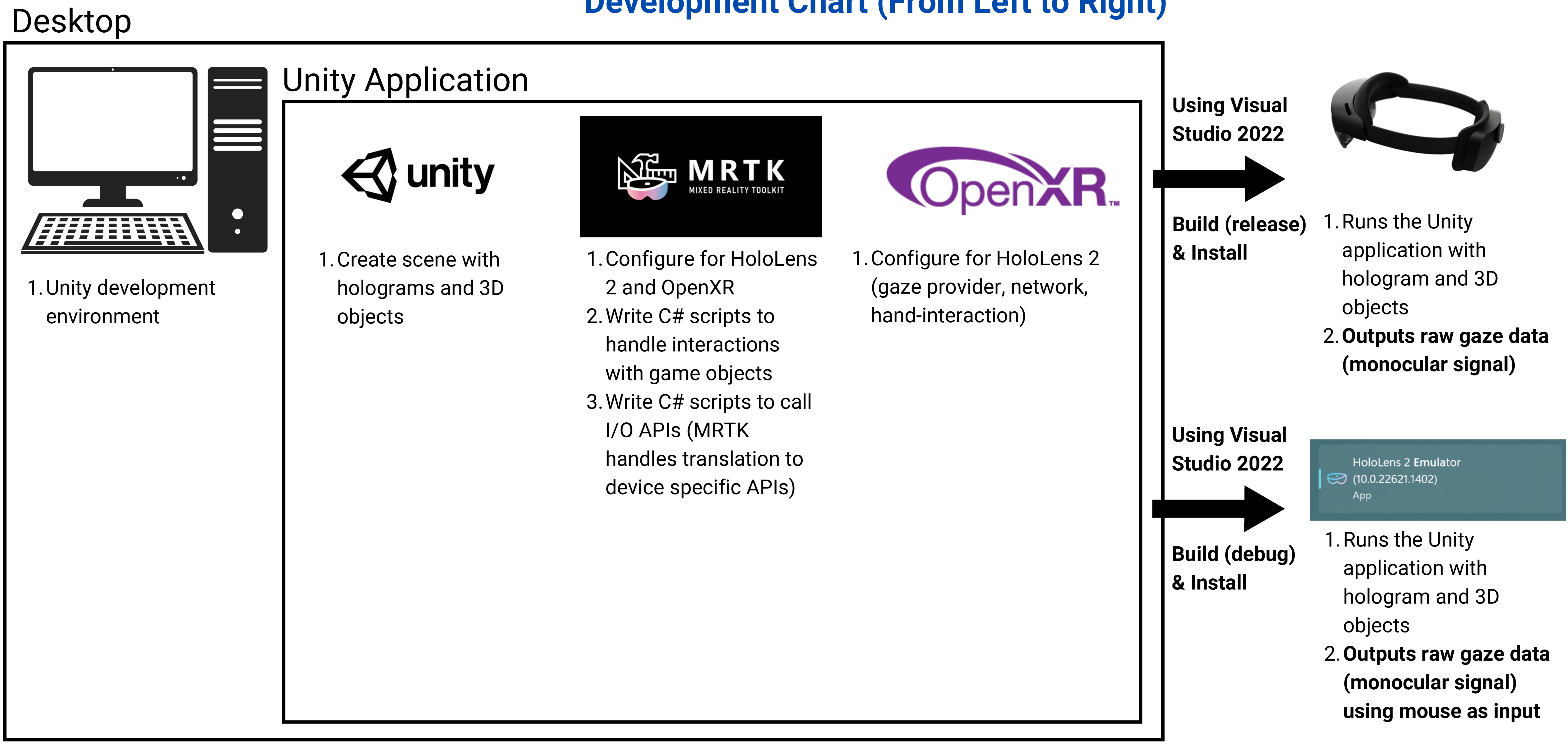
From HoloLens 2 device storage

Wi-Fi / Bluetooth / USB-C



1. **Download and store gaze data from HoloLens 2**
2. **Processing and analysis**

Development Chart (From Left to Right)



Desktop

1. Unity development environment

Unity Application

1. Create scene with holograms and 3D objects

1. Configure for HoloLens 2 and OpenXR
2. Write C# scripts to handle interactions with game objects
3. Write C# scripts to call I/O APIs (MRTK handles translation to device specific APIs)

1. Configure for HoloLens 2 (gaze provider, network, hand-interaction)

Using Visual Studio 2022

Build (release) & Install

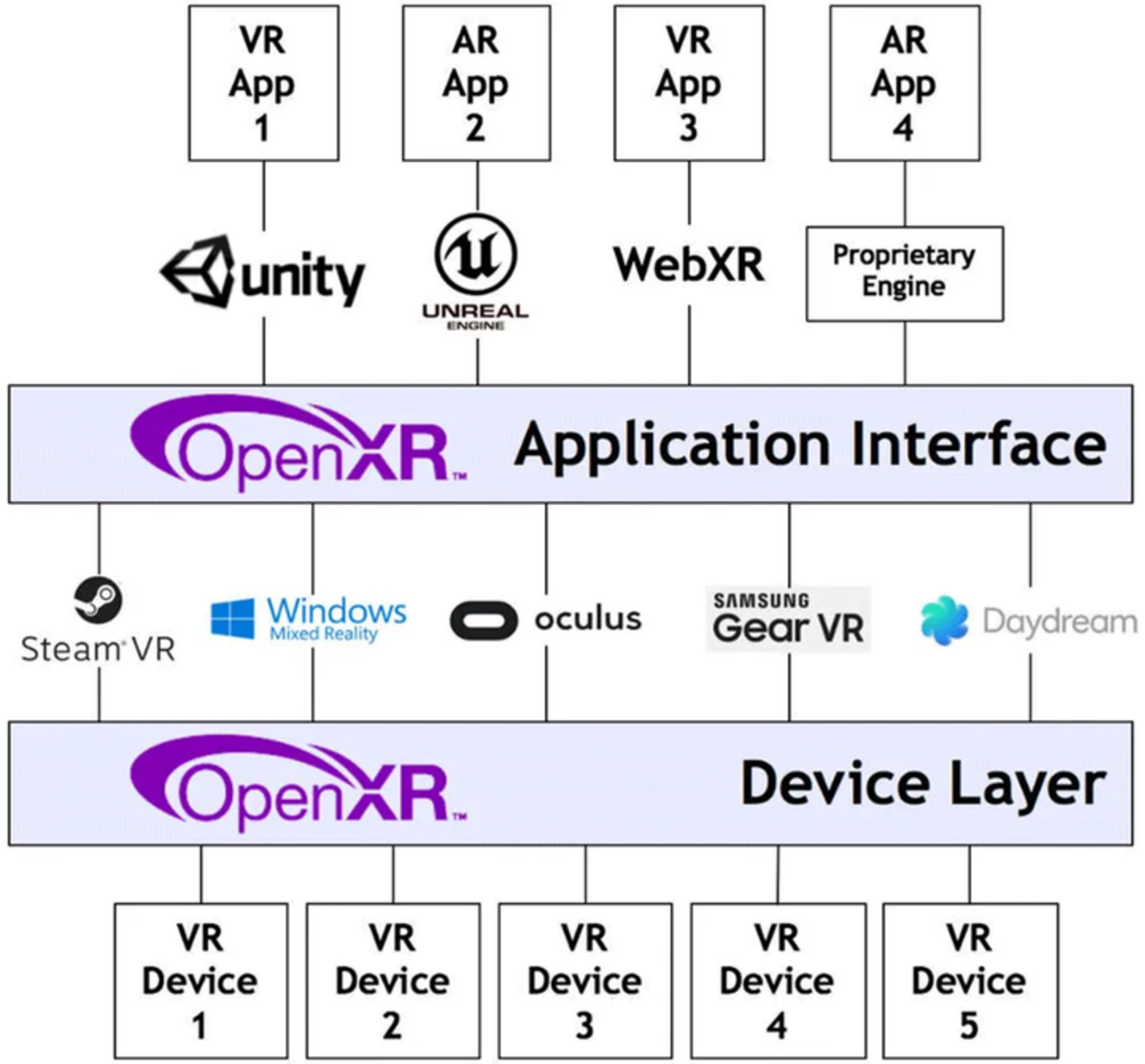
1. Runs the Unity application with hologram and 3D objects
2. **Outputs raw gaze data (monocular signal)**

Using Visual Studio 2022

Build (debug) & Install

1. Runs the Unity application with hologram and 3D objects
2. **Outputs raw gaze data (monocular signal) using mouse as input**

OpenXR abstracts the hardware I/O into APIs that software such as MRTK can access and used in game engine (Unity)



OpenXR Goal

Useful resources:

1. An Assessment of the Eye Tracking Signal Quality Captured in the HoloLens 2: https://dl.acm.org/doi/fullHtml/10.1145/3517031.3529626
2. Mixed Reality OpenXR: https://learn.microsoft.com/en-us/windows/mixed-reality/develop/native/openxr