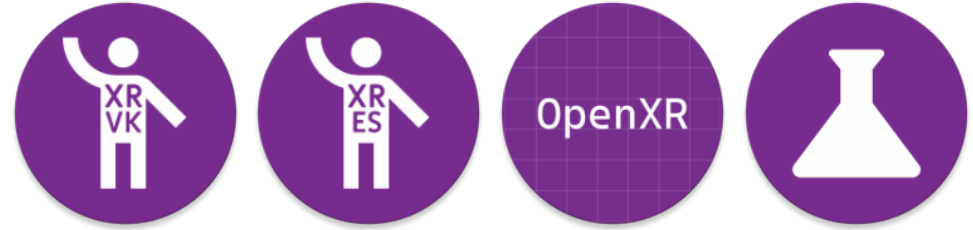


OpenXR on Android – Source Included

Ryan A. Pavlik
FOSS XR 2022



OpenXR and the OpenXR logo are trademarks owned by The Khronos Group Inc. and are registered as a trademark in China, the European Union, Japan and the United Kingdom



COLLABORA

Ryan Pavlik

**Principal Engineer at
Collabora
OpenXR Specification
Editor**

Open First



Outline

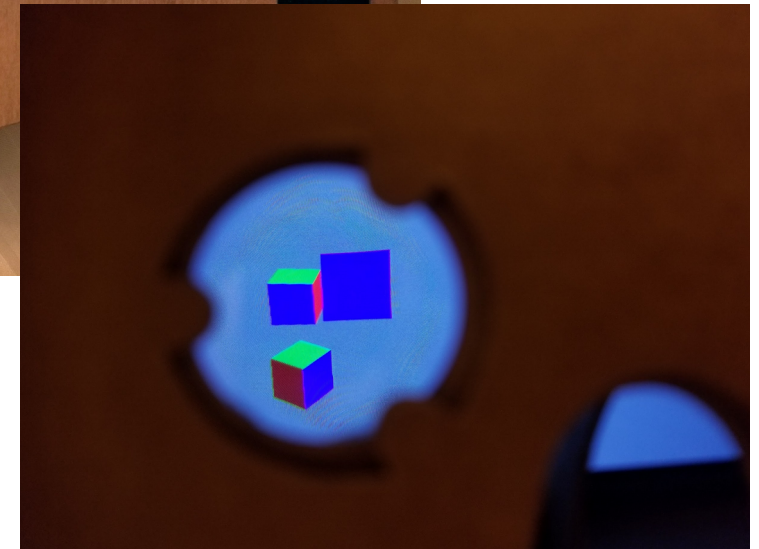
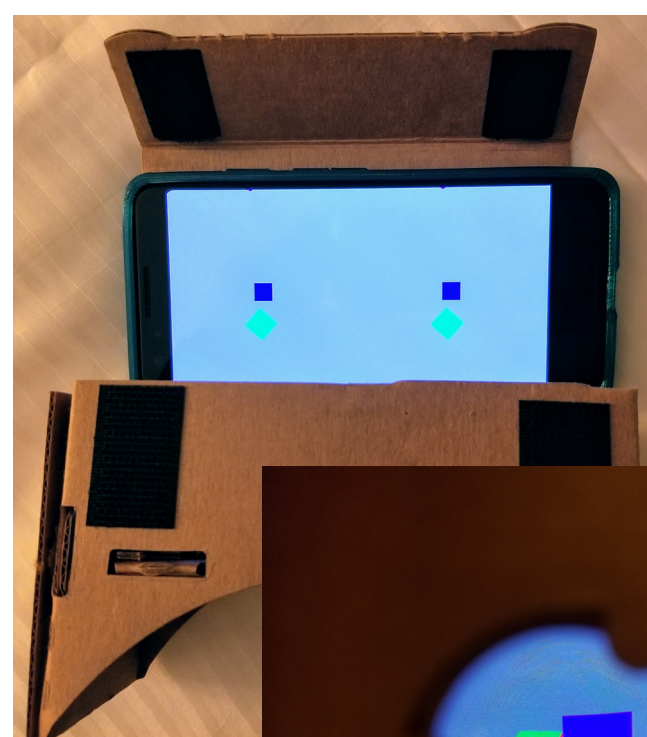
- Requirements, use cases, and design decisions
- What we built and how to use it
- Question and answer time

- Note: I cannot and do not speak for the WG, but I can discuss the considerations I discussed and kept in mind as we built the stack to where it is today.
- “We” might mean “working group” or might mean “the Monado community”



Sneak Peek

- You can run the same Hello XR binary on the Vive Focus 3 and on a stock Pixel 3 running Monado in a Cardboard-style “phone holder” headset
- The stack is designed to be open source and support user choice.



COLLABORA

Open First



COLLABORA

Requirements and use cases

Background

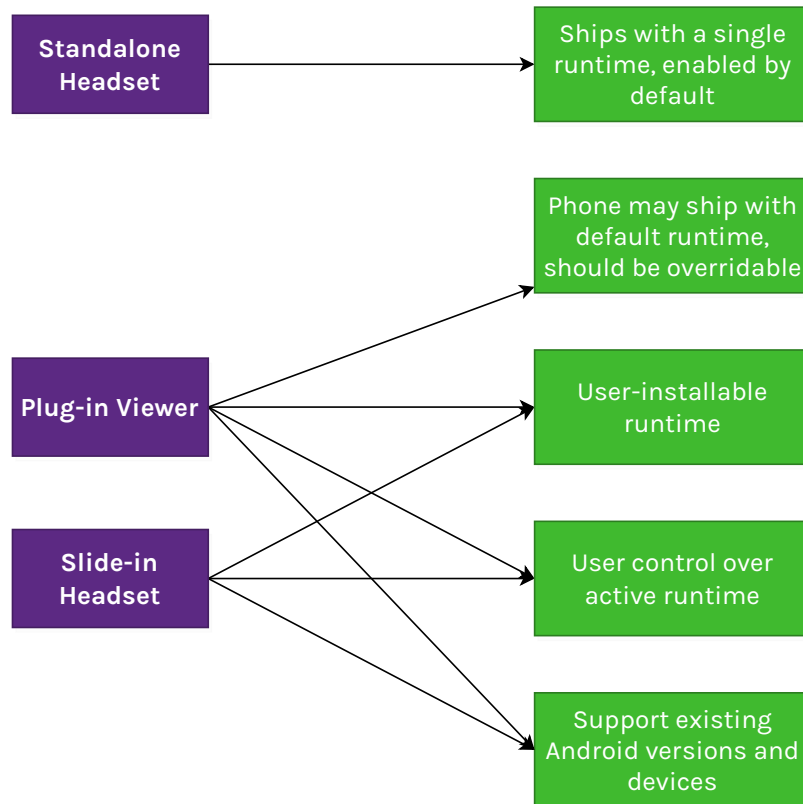
- OpenXR provides a standard, uniform interface for XR devices and runtimes
- Applications build against an “OpenXR Loader”, not a given headset/runtime
- That loader finds the active runtime and dynamically loads it, among other features
- Runtimes may implement additional features as optional “extensions”, which is how new APIs are prototyped



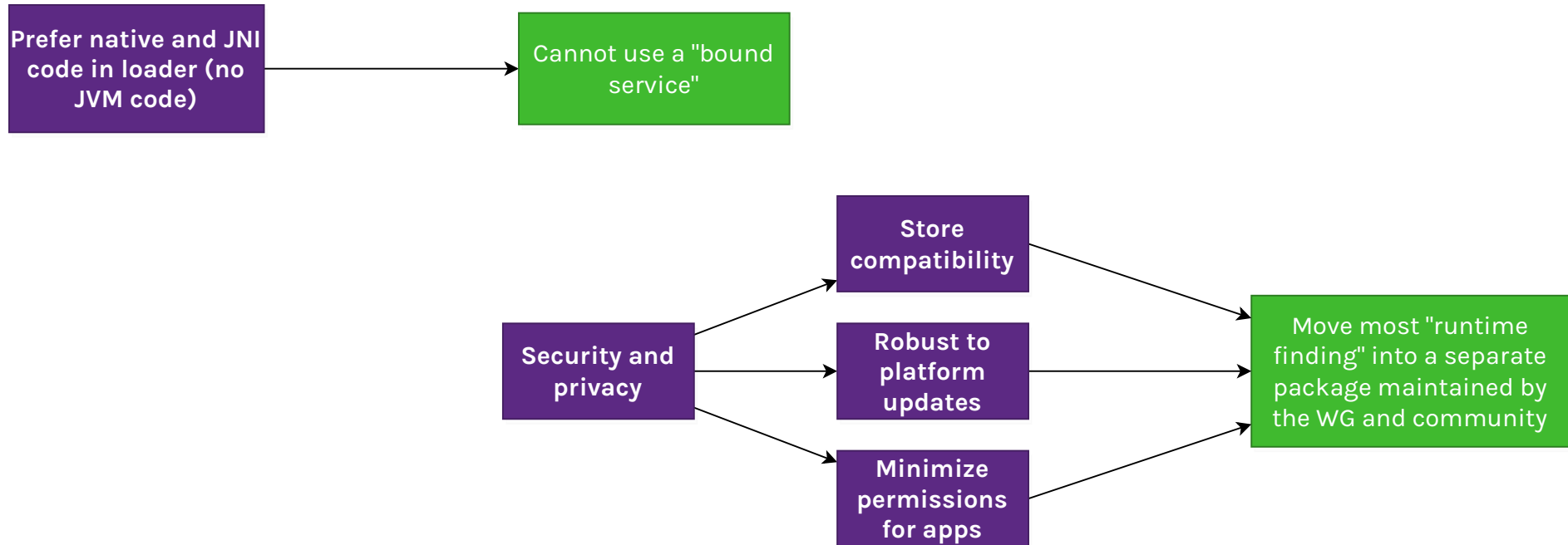
The Basics

- Primary general requirement: a single loader (library) to use on all OpenXR Android devices
 - Should be possible to build a single APK to run on any Android-based OpenXR device
- Main use cases
 - All-in-one headset
 - Plug-in viewers
 - Slide-in “Phone holder”

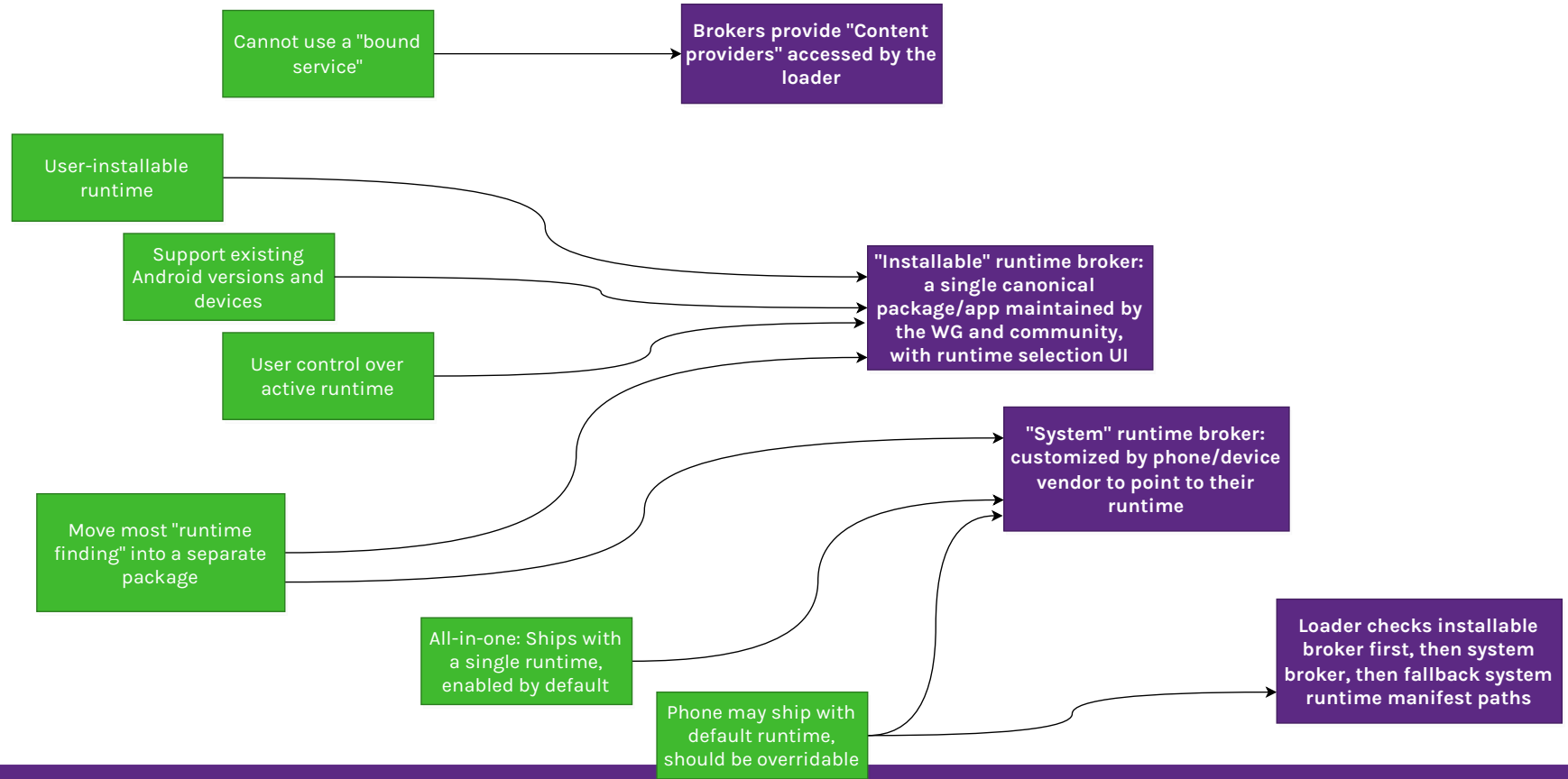
Use Cases



Additional requirements



Leading to our solution



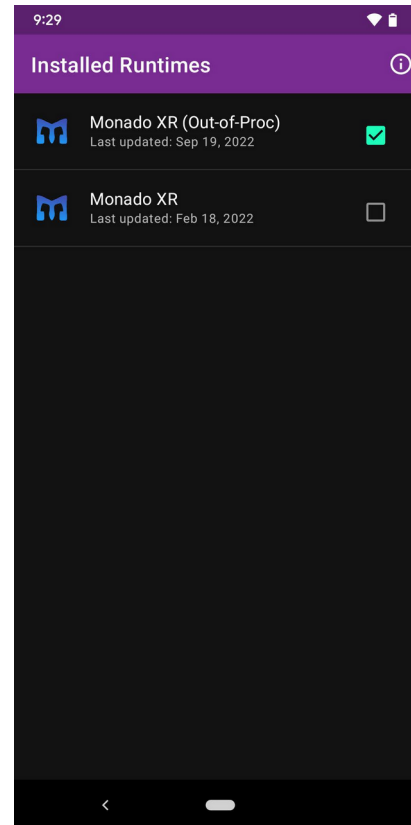


COLLABORA

What we built and how to use it

What: Installable broker + UI

- Lists runtimes found by **manifest metadata**
- If none are selected, loader falls back to system runtime, then system manifests
- Plan to publish to F-Droid and Google Play
- Open source maintained by the Monaco project and the OpenXR WG
- Vendors **must not customize** this for their runtime: use the published discovery interface, or customize the system runtime broker



COLLABORA

<https://gitlab.freedesktop.org/monado/utilities/openxr-android-broker>

Open First

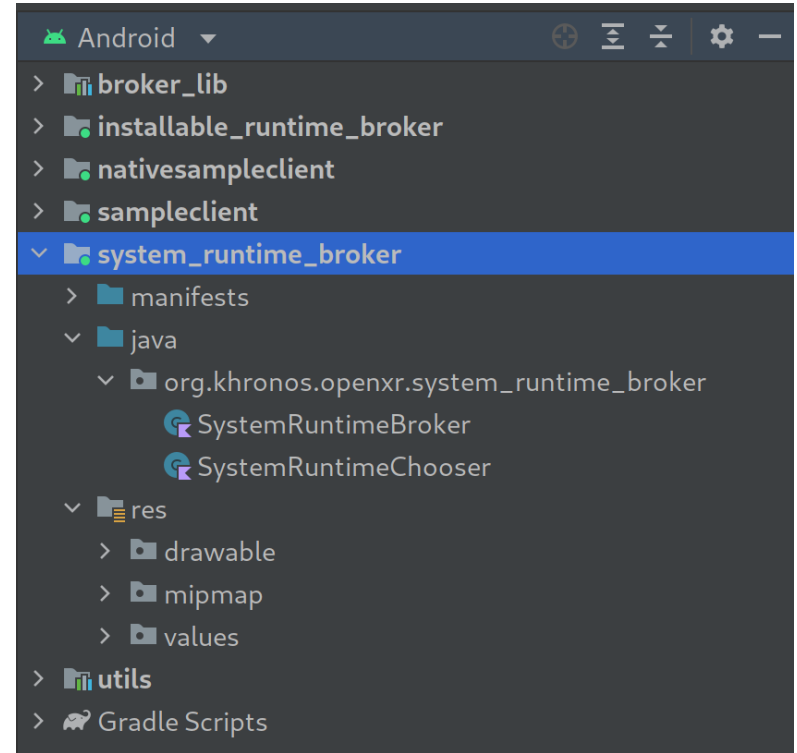
Using the Installable Broker

- Download and install a release from the GitLab project
- Install a runtime (just wait)
- Open the broker and enable your installed runtime



What: Stubs for system runtime broker

- In the same repo/project as the installable broker, but meant for vendors to customize and ship in their device image
- No UI, meant to hard-code runtime data



How to use the system broker

- If you're using an HTC Vive Focus 3: You already are using it :)
- If you're developing your own standalone device:
 - clone/fork the runtime broker repo
 - customize SystemRuntimeChooser with your logic
 - include the resulting APK in your system image.



What: Cross-Vendor Android Loader

- Connects to the OpenXR runtime brokers to find the runtime to load
- One APK *will be* able to work on all Android-based OpenXR devices
 - Some vendors are still migrating to this from their legacy loader
- Maintained in the same source tree as the desktop loaders: open source (Apache-2.0 OR MIT)
- AAR artifacts published to Maven Central for Gradle usage, and published on GitHub Releases

Use it in Android Gradle Plugin

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 29
    ndkVersion '21.3.6528147'

    defaultConfig {
        applicationId = 'com.sample.teapotxr'
        minSdkVersion 24
        targetSdkVersion 29 // 30 breaks loader rn
    }

    externalNativeBuild {
        cmake {
            version '3.10.2'
            path 'src/main/cpp/CMakeLists.txt'
        }
    }
    ...

    // enable prefab support for the OpenXR AAR
    buildFeatures {
        prefab true
    }
}

dependencies {
    implementation 'org.khronos.openxr:openxr_loader_for_android:1.0.23'
}
```

Note: 1.0.24 and 1.0.25 have some issues in the published AAR when used with Gradle. Fix merged, will be in the 1.0.26 AAR



What: Android release binaries



- Hello XR, in versions that default to Vulkan and OpenGL-ES
- OpenXR-CTS – Release APKs will be available for next CTS release
 - Now includes the conformance layer in the APK so conformance can pass on non-rooted devices
 - They still need `adb shell setprop` to configure and run the tests, however: no immersive or pancake UI for test setup.



Get release binaries

- OpenXR-SDK-Source GitHub releases:

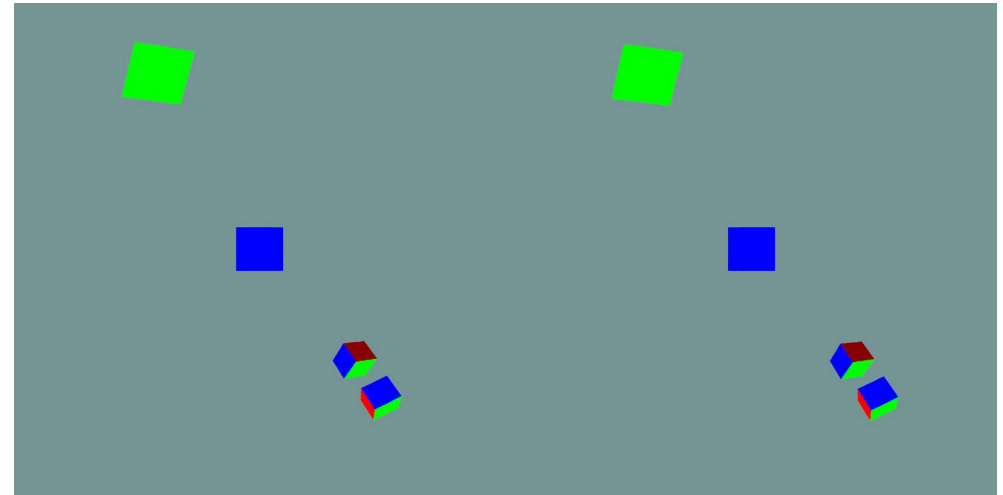
<https://github.com/KhronosGroup/OpenXR-SDK-Source/releases/>

- OpenXR-CTS GitHub releases (APKs coming soon):

<https://github.com/KhronosGroup/OpenXR-CTS/releases/>

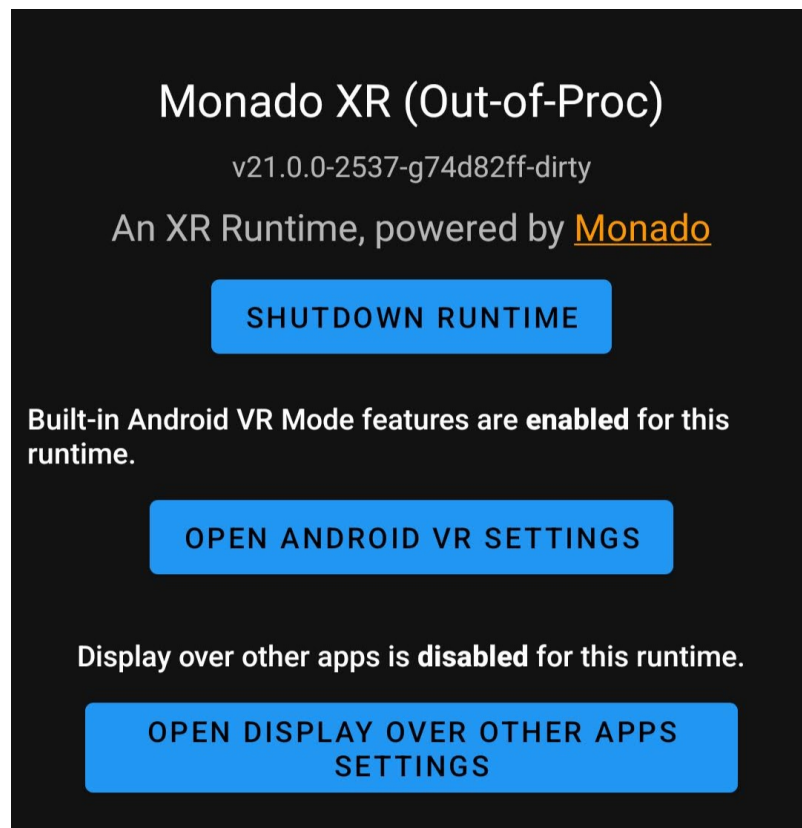
What: Android support in Monado

- Runs on Android, though not officially conformant.
- By default runs in “phoneholder mode” a la Google Cardboard, using onboard IMU on a wide range of devices



How to use Monado on Android

- “Debug”-signed APKs are built by Monado CI, in both “in-process” and “out-of-process” modes – download and install
- Or, build locally with Android Studio or Gradle
- Enable the runtime as “active” in the “installable” runtime broker
- Grab hello_xr binaries from the OpenXR-SDK-Source project releases or use some other OpenXR app APK



Thank you to the community!

- Thanks to contributors to OpenXR-SDK-Source, the OpenXR Android runtime broker repo, and Monado for helping to build this software
- Thanks to the working group members who contributed requirements, constraints, ideas, and code to the process
- Thanks to Collabora for funding my work on OpenXR and Monado

How to participate

- Khronos public GitHub repos – open to contributions:
 - Loader, Hello XR: <https://github.com/KhronosGroup/OpenXR-SDK-Source>
 - CTS: <https://github.com/KhronosGroup/OpenXR-CTS>
- Runtime broker repo: <https://gitlab.freedesktop.org/monado/utilities/openxr-android-broker>
- Monado: <https://monado.freedesktop.org>
- Android NDK sample “classic-teapot” ported to be an OpenXR app:
<https://gitlab.freedesktop.org/monado/demos/androidteapots>
- Khronos OpenXR WG – if employer is a Khronos member



Thank you!



COLLABORA

Open First