

Vulkanised 2026

The 8th Vulkan Developer Conference
San Diego, USA | February 9-11, 2026

Vulkan on Android

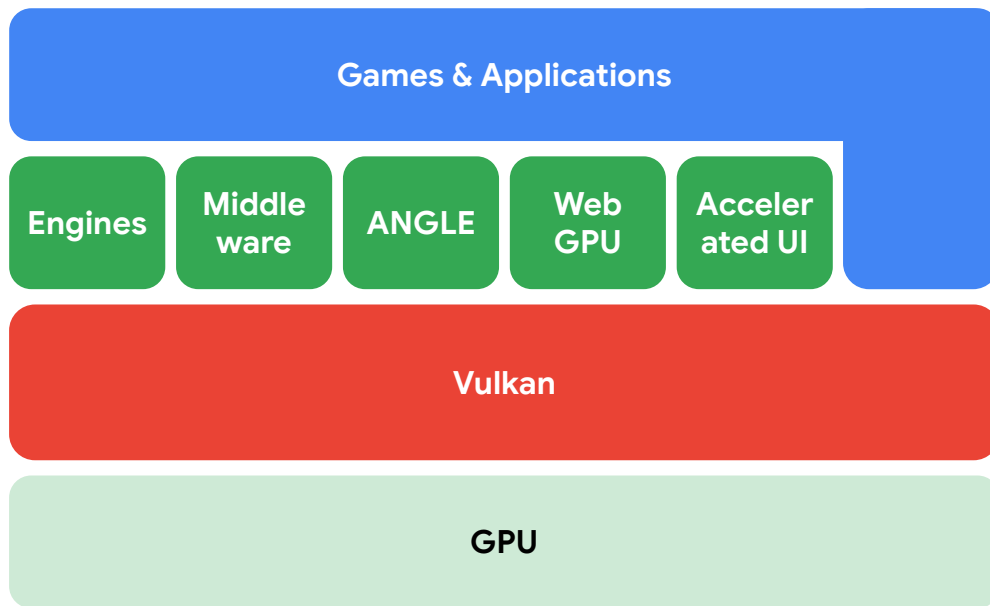
Ian Elliott, Google



Android's Vulkan Only Strategy

Provide Vulkan as the GPU HAL on Android

We're helping the industry focus on Vulkan by supporting all existing and future content on top of Vulkan



Vulkan Adoption is progressing well

We've had substantial recent success with Vulkan:

- NetEase used Vulkan to implement ray tracing in Diablo Immortal, bringing their world to life with advanced visual effects
- Game of Thrones: Kingsroad, Deep Rock Galactic, and Dredge are recent x-platform games made available on Android through Vulkan
- Genshin Impact is also using Vulkan on newer devices
- We've had a close collaboration with Game Engines such as Unity, and now over 55% of sessions from new Unity titles are using Vulkan, and we expect this number to grow rapidly*

* Games launched between August, 2025 and February, 2026



Why should you care about Android?

You can reach a large audience

- Billions of Android phones
 - Vulkan 1.1 is required since Android 15 (2024).
- A large range and variety of devices
 - Sub-\$100 phone to >\$1,000 Flagships
 - New Chrome devices will be on Android
 - Other “verticals” (watches, TV, automotive)
- Use Vulkan directly or indirectly via
 - Layered APIs (e.g. OpenGL ES, WebGPU, Skia, HWUI)
 - Middleware
 - Game engines



However, there are challenges

- Many of the low-cost phones:
 - Use older GPUs
- Most Android devices don't get driver updates
 - No new features
 - No bug fixes

Available Vulkan Functionality

How to know what Vulkan functionality to use?

What do you want to do?

- Do you want broad “reach” (old & new, low & high-end)?
- Do you want to run on the newer flagships with new functionality?

There’s always Sacha Willems’s <https://vulkan.gpuinfo.org/>

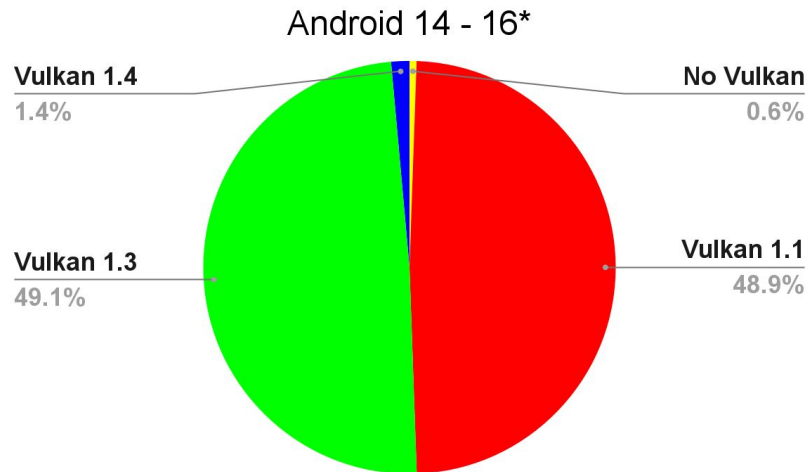
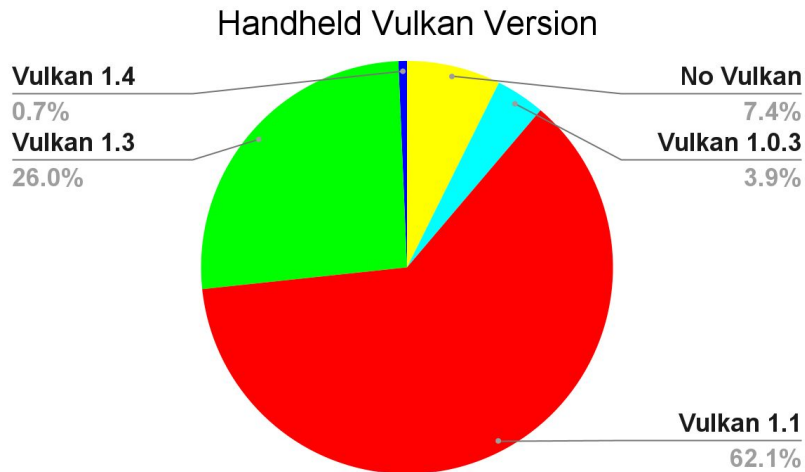
We have additional resources to help you:

- Dashboard
- Vulkan profiles
- Plug-ins to know when you’re on a high-quality device
- etc.

Distribution Dashboard

developer.android.com has the [Distribution dashboard](#)

- Data from active Android handheld devices



*Android 14 was released in October, 2023
Android 16 was released in June, 2025

Vulkan Profiles

Different types and uses:

- Allows your application to query Vulkan capabilities
 - Use for what your application wants/needs
 - `vpCreateDevice == vkCreateDevice + enable everything in your profile`
- Allows you to target specific functionality dictated by others
 - Android and Khronos collections of functionality
 - You can make informed choices by looking at what's in these:
 - Widely available functionality (today)
 - Android pushing software developer needs
 - Khronos pushing newer functionality

Our what's “widely available” profiles

Q4 - 2021

[Android Vulkan Profile 2021](#)

VP_ANDROID_vulkan_profile_2021

- **96.05%** of active Vulkan-capable devices support this profile
- Any individual part of the profile may be supported on more devices

What was widely available at the end of 2021

Q4 - 2022

[Android Vulkan Profile 2022](#)

VP_ANDROID_vulkan_profile_2022

- **86.61%** of active Vulkan-capable devices support this profile
- Any individual part of the profile may be supported on more devices

What was widely available at the end of 2022

Q3 - 2025

[Android Vulkan Profile 2025](#)

VP_ANDROID_vulkan_profile_2025

- **81.5%** of active Vulkan-capable devices support this profile
- Any individual part of the profile may be supported on more devices

What was widely available in Q3-2025

Vulkan Requirements for Android

Android 15

[Vulkan Requirements for 15](#)

VP_ANDROID_15_requirements

- **3.69%** of active Vulkan-capable devices support this profile
 - A few items* kept the percentage down
 - Most parts of the profile are supported on many devices

Requires and builds on top of:

- Vulkan 1.3
- Android Baseline Profile 2022

Android 16

[Vulkan Requirements for 16](#)

VP_ANDROID_16_requirements

- **0.85%** of active Vulkan-capable devices support this profile
- Individual parts of the profile are supported on more devices

Requires and builds on top of:

- VRA15

Android 17

[Vulkan Requirements for 17](#)

VP_ANDROID_17_requirements

- **0%** of active Vulkan-capable devices support this profile for the unreleased Android 17

Requires and builds on top of:

- Vulkan 1.4
- AVP 2025

* Those items are: VK_EXT_external_memory_acquire_unmodified, VK_KHR_maintenance5, VK_KHR_vertex_attribute_divisor, VK_ANDROID_external_format_resolve

What if you need more/finer-grained data?

- What would help you?
 - Some possibilities:
 - Profiles for specific renderers from popular game engines (Unity's HDRP)
 - Profiles emulating other APIs (DxVk, VkD3D)
 - Some questions:
 - Update Cadence: Would it help if AVP profiles were updated more than annually?
 - Hardware Telemetry: Would aggregated hardware specifications help you?
 - Something similar to the Steam Hardware Survey (e.g., CPU cores/frequencies, refresh rates, RAM, storage etc.)
 - Please come talk with us afterwards with your requests!
- We are considering a dashboard that will reflect the percentage of active devices support the specific:
 - Extensions
 - Feature bits
 - Properties
 - Limits
 - Formats
 - etc.

How you should use Vulkan profiles

- Glean information from the Android and Khronos profiles
 - Especially learn what's widely available
- Create your own
 - Use for what your application wants/needs
 - `vpCreateDevice == vkCreateDevice` + enable everything in your profile
- Perhaps have a set of profiles
 - For example:
 - Minimum Must have at least this much to run
 - Better Can turn on more of your features
 - Best Can turn on all of your features

Tools and Layers

Current Developer Tools

- Cross-Android tools
 - RenderDoc
 - Android Graphics Inspector (AGI)
 - Portions of the Vulkan SDK (e.g. Validation Layers, GFXReconstruct)
- Vendor-specific tools
 - Mobile vendors have their own tools
 - Some examples:

Arm	Streamline Performance Analyzer
Imagination Technologies	PVRTune
Qualcomm	Snapdragon Profiler
Samsung (Xclipse GPU)	Sokatoa (releasing soon)

New Android Profiler Coming

- Standalone, cross-Android, open-source, Vulkan GPU Profiler
 - Taking Android Graphics Inspector (AGI) to the next level
 - Works across all major Android GPUs
 - An improved system profiler launching soon
 - Faster, more reliable, with a smoother UX
 - Frame screenshots
 - Supports very large traces
 - View traces side-by-side
 - Will eventually be open sourced
 - An improved frame profiler with alpha coming in H2, 2026
 - Based on GFXReconstruct, with frame looping for accurate measurements
 - Render pass graph to help you optimize memory bandwidth
 - Frame and render pass timings and counters to find costly operations
 - Export to RenderDoc for deep inspection
 - Roadmap for 2027+
 - Deeper inspection
 - What-if experiments

Use of Vulkan layers on Android

Implicit layers were forbidden for security reasons, except for:

- Debuggable applications
- Users of devices with rooted/userdebug builds of Android

Your application can use layers:

- During development
 - E.g. Validation layers
- That you ship in your APK
 - E.g. generic implementations of extensions that not all drivers support

Future Layer Improvements

- We're adding a new/limited capability for implicitly-loaded layers:
 - OEM layers tested and shipped by phone manufacturers
 - Platform layers tested and shipped by Android vertical "platforms" (e.g. XR, TV)
- WSI extensions will migrate from the loader to a platform layer
 - We'll still provide common behavior across all phones and tablets
 - Some verticals may not use this layer

Engines and Layered APIs

For those using Vulkan indirectly

Improving Engine support



Game engine support

Game Engine Vulkan support is improving.

- We're working closely with the major commercial and in-house game engines to help improve their Vulkan renderers
- We're collaborating on new higher fidelity features, as well as just general stability
- On major game engines today, Vulkan is now on by default, as it is the more suitable backend.



De-risking using Vulkan

Active devices may not support Vulkan well

- Some don't have a high-quality Vulkan driver

We've been working with large engines

- Selection mechanism for which devices/releases to still use GLES on
 - Denylist on newer Android versions
 - Allowlist on older Android versions

Try out the allowlist today at:

<https://developer.android.com/games/engines/unity/unity-vkquality>



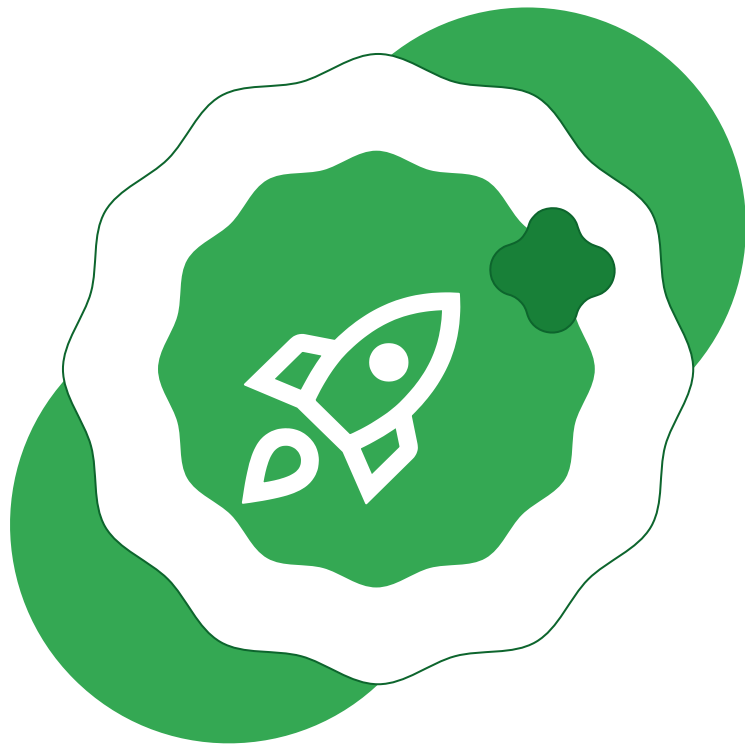
Accelerated UI

Improving App performance

Most apps have no need for direct rendering and so use HWUI and/or Skia to render.

HWUI layers on top of Skia

Skia uses Vulkan on new chipsets since 2024.



WebGPU

A higher level graphics language for app developers

App developers (especially those using Kotlin or Java) can't access Vulkan easily. We are encouraging them to use WebGPU to access the GPU on Android.

Google's implementation of WebGPU uses Vulkan and can be accessed directly from managed code. It launched as a jetpack library at the end of 2025.

Next, we are adding support for WebGPU to Android itself to make it easier for developers to access.

We're also adding WebGPU CTS into Android CTS to ensure high quality



ANGLE

GL ES 3.2 Conformant layered implementation of OpenGL ES (GLES) on top of Vulkan

- Is the only GLES driver on some phones today
- Will replace legacy GLES drivers over the next few releases of Android
- Focuses ecosystem on just one GPU HAL to
 - Accelerate our progress on Vulkan
 - Supports existing GLES applications
- **Coming soon:** Application manifest flag supported on new SoCs to explicitly request ANGLE as the app's GLES driver



Driver Quality and Driver Updates

A focus on driver quality **for Android 16** (and beyond)

- We're continuing to **invest further in driver quality**.
- **Additional test suites** being added to our graphics tests, such as the **ANGLE End-to-end tests** and **WebGPU tests**, to further exercise the Vulkan driver in different ways.
- In Android 17, new and updating devices will use the latest certification tests to ensure driver quality

More testing means better drivers and better experience for end users

ANGLE improves Vulkan driver quality

Driver quality improves with exposure to content

- Many Vulkan drivers haven't been exposed to enough content

ANGLE implements OpenGL ES (GLES) on top of other APIs

- The ANGLE backend is a sophisticated driver
- Varies its Vulkan API usage based on content and Vulkan driver
- Exposes Vulkan drivers to a lot of content

Observation: the Vulkan drivers for companies that have embraced ANGLE have better quality

- Fewer bugs
- Lower overhead
- Better performance characteristics for the supported HW

Thank you

