Android Virtualization Framework (AVF) overview

Archived: 2026-04-06 02:04:50 UTC

Android Virtualization Framework (AVF) overview Stay organized with collections Save and categorize content based on your preferences.

Android Virtualization Framework (AVF) provides secure and private execution environments for executing code. AVF is ideal for security-oriented use cases that require stronger, even formally verified, isolation assurances over those offered by Android's app sandbox. Android provides a reference implementation of all the components needed to implement AVF. AVF is supported only on ARM64 devices. Figure 1 shows the architecture of AVF:

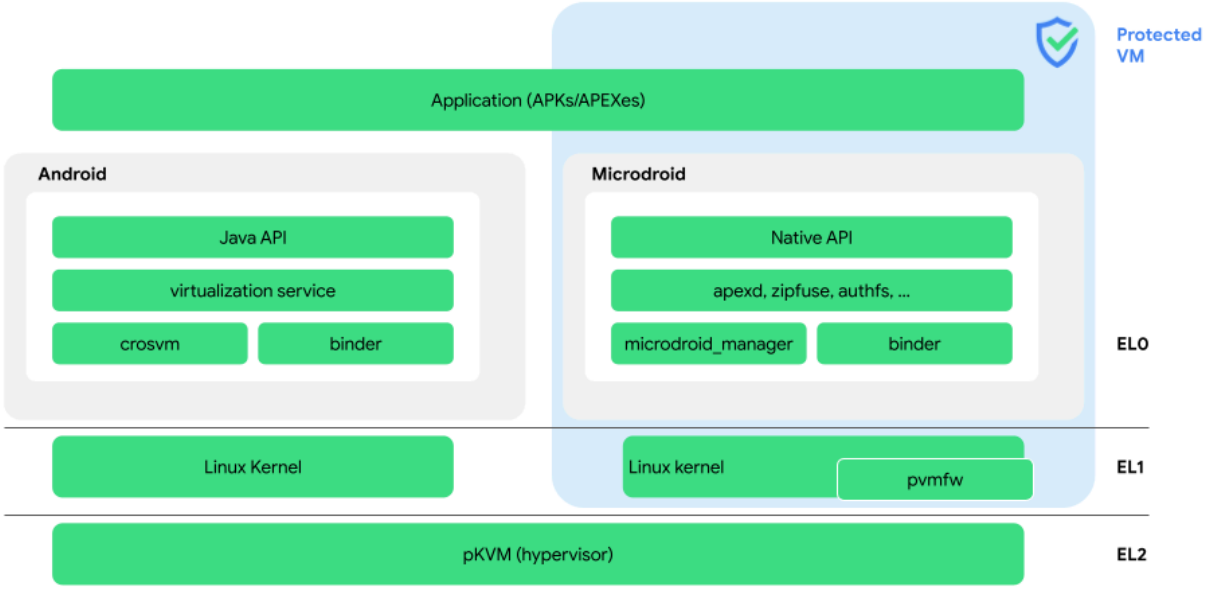

Figure 1. AVF architecture.

Here are the definitions for the most important terms from figure 1:

apexd and zipfuse
: Securely mounts APEXes and APKs imported from host.

authfs
: A fuse file system for securely sharing multiple files between Android and pVM (host and guest).

binder
: Primary means of inter-VM communication.

crosvm

A virtual machine monitor written in rust. crosvm allocates VM memory, creates virtual CPU threads, and implements the virtual device's back-ends.

Generic Kernel Image (GKI)

A boot image certified by Google that contains a GKI kernel built from an Android Common Kernel (ACK) source tree and is suitable to be flashed to the boot partition of an Android device. For further information, see the Kernel overview.

hypervisor

The virtualization technology used by AVF, also known as *pKVM*. The hypervisor maintains the integrity of the executed code and confidentiality of the pVM's assets, even if host Android or any of the other pVMs are compromised.

Java API

The VirtualizationService Java APIs, which are present only on devices with AVF support. These APIs are optional and not part of `thebootclasspath`.

Microdroid

A Google-provided mini-Android OS that runs in a pVM.

Microdroid Manager

Manages the pVM lifecycle, inside the pVM, and instance disk.

Native API

A subset of the Android Native Developers Kit (NDK).

protected kernel-based virtual machine (pKVM)

See Hypervisor.

pVM firmware (`pvmfw`)

The first code that runs on a pVM, `pvmfw` verifies the payload and derives the per-VM secret.

protected virtual machine (pVM)

A mutually distrusted isolated execution environment (*guest*) that runs alongside the main Android operating system (*host*). One important aspect of pVM security is even if the host is compromised, the host doesn't have access to a pVM's memory. pKVM is the standard hypervisor for running pVMs.

Compared to existing trusted execution environments (TEEs), pVMs provide a richer environment, including the ability to run a mini-Android distribution called Microdroid (though Microdroid can also run on an unprotected VM). pVMs can be used dynamically and provide a standard set of APIs in a trusted environment available across all devices that support them.

VirtualizationService

The Android service that manages the lifecycle of pVMs.

What's next?

- If you want to better understand the need for AVF, refer to Why AVF?.
- To read about how AVF can be used for isolated compilation, refer to Use cases.
- If you want a more in-depth explanation of the AVF reference implementation's architecture, refer to AVF architecture.

- If you want to learn about Microdroid, refer to Microdroid.
- If you are interested in how AVF handles security, refer to Security.
- To understand the role of the virtualization service, refer to VirtualizationService.
- For source code of AVF or in-depth explanation about individual components, refer to AOSP repository

Last updated 2026-03-11 UTC.
