

# Android Internals - A Confectioner's CookBook

Status (05/25/2022)

- Vol I: Done - v2.1
- Vol II: Done
- Vol III: Still pretty far
- Vol IV: A bit closer

## Volume I : The Power User's View

Major rewrite for Android 11/12 - 440 pages, complete (now with Config chapter, too)

Changes/additions with respect to 1<sup>st</sup> edition shown in yellow

The original (ISBN: 978-0-9910555-2-4) remains free on this website (Thank you, CIA).

The book underwent a complete rewrite (effectively, a 2<sup>nd</sup> edition). Updates to align with Android 11, and latest SD865, Exynos, MediaTek and Kirin devices

### 0. About this book

- Overview and Reading Suggestion
- The AOSP
- Experiments
- Tools
- Conventions Used in this Book
- The Companion WebSite

### 1. Introduction and evolution of the Android Architecture

A technical overview of the Android architecture, evolution of its features, and forked derivatives

- Android Versions - From Cupcake to Marshmallow R (11.0)
- The Android Architecture
- Android vs. Linux: Notable differences
  - Not just another Linux distribution
  - And then came Android
  - Commonalities and Divergences from Linux
  - Obtaining and compiling Android
  - The sources of Android Code
    - AOSP
    - AOSP external
    - The Linux kernel
    - Platform/BSP (vendor)
    - ODM
    - Carrier
- A high level view of the Android Architecture
  - Applications
  - The Android Frameworks
  - Dalvik/Android Runtime
  - JNI
  - Native Binaries
  - Native Libraries
  - Bionic
    - Omissions
    - Additions
    - Porting Challenges
  - The Hardware Abstraction Layer (HAL)
    - The Traditional HAL (2.2-8.0)

- The Linux Kernel
      - Linux kernel features
      - Androidisms
      - Drivers and Modules
      - Generic Kernel Image (GKI)
  - Android Derivatives
    - FireOS, FireTV
    - Android Wear
    - Android Auto & Automotive
    - Google Glass
    - Headless Android
  - Pondering the way ahead

## 2. Hardware

A new chapter providing a detailed introduction to the hardware of Android devices, with an emphasis on hardware abstraction and Project Treble compliance

- The ARM architecture
  - Aarch32 and Aarch64
  - ARM architecture revisions
- Devices
- System on Chip (SoC) overview
- SoC vendors
  - Qualcomm (Snapdragon)
  - Samsung (Exynos)
  - Huawei (Kirin)
  - MediaTek (MTK)
- The Device Tree
- Firmware images

## 3. Partitions & Filesystems

Examining Android storage types and partitions, as well as a detailed breakdown of directories and files in /system and /data.

- Partitioning scheme
  - The Need for Separate Partitions
  - Flash Storage Systems
  - GUID Partitioning (GPT)
  - A/B/[C] Slotted devices
  - Dynamic partitioning (super.img)
  - Lpddumpd (Android 10+)
- Android Device Partitions
  - Mountable Partitions
    - System-as-root
    - Supported filesystem types
    - /vendor, /odm, /product - Division of responsibility
  - Non-Mountable Partitions
    - boot, vendor\_boot and recovery
    - dtbo
    - frp
    - misc
    - vbmeta
- Chipset-specific Partitions
  - Qualcomm
    - cdt
    - devinfo
    - splash
  - Samsung
  - Huawei
  - MediaTek
- Linux Pseudo-Filesystems
  - bpf (/sys/fs/bpf)

- [conmgrs](#)
- [debugfs](#)
- [functionfs \(/dev/usb-ffs/adb\)](#)
- [FUSE](#)
- [incrementalfs \(11.0\)](#)
- [overlayfs](#)
- [procfs \(/proc\)](#)
- [pstore \(/sys/fs/pstore\)](#)
- [sdcardfs/esdfs](#)
- [securityfs \(/sys/fs/selinux\)](#)
- [sysfs \(/sys\)](#)
- [tmpfs](#)
- [tracefs \(/sys/kernel/debug/tracing\)\)](#)

#### 4. Files & Directories

[A detailed breakdown of directories and files Android, up to and including Android 11](#) (Spun off from first edition's discussion in [Partitions & Filesystems](#) chapter, and greatly expanded).

- [The Root Filesystem](#)
- [/system](#)
  - [/system/bin](#)
  - [/system/lib\[64\]](#)
    - [Core Libraries](#)
    - [Other system libraries](#)
    - [Framework support libraries](#)
    - [External native libraries](#)
  - [/system/etc](#)
- [/vendor](#)
  - [/vendor/bin](#)
    - [Qualcomm specific binaries](#)
    - [Huawei specific binaries](#)
    - [Samsung specific binaries](#)
    - [MediaTek specific binaries](#)
- [/data \(Excerpt\)](#)
  - [/data/data](#)
  - [/data/misc](#)
    - [/data/misc\\_ce and /data/misc\\_de](#)
  - [/data/system](#)
    - [/data/system\\_ce and /data/system\\_de](#)
  - [/data/vendor](#)
- [/cache](#)

#### 5. Storage Management

[Refactoring vold and OBB/ASEC, as well as adding new content on Storage\\* and APEX](#)

- [Mounting](#)
  - [Mount options](#)
  - [Loop mounting](#)
  - [Bind mounting](#)
  - [Mount namespaces](#)
  - [fs\\_mgr](#)
  - [The fstab files](#)
  - [External Storage](#)
    - [Portable Storage](#)
    - [Adoptable Storage](#)
  - [Scoped Storage \(Android 10\)](#)
  - [Incremental FS \(Android 11\)](#)
- [Daemons](#)
  - [vold](#)
  - [StorageManager](#)
  - [storaged](#)
  - [storagestats](#)

- Protected filesystems
    - Obb - Opaque Binary Blobs
    - ASec - Android Secure Storage
  - APEX - Android Pony EXpress (Android 10)
    - apexd
    - Execution Flow
    - Additional Command Line Arguments
    - The AIDL interface
    - APEX and the linker configuration
    - Android 11.0 modifications
- 6. Android System Images & Updates
 

Split from the older Chapter III, and greatly expanded

  - Factory Images and OTA updates
    - Factory Images
    - OTA packages
    - Samsung OTA
  - 
  - Standardized Payload Formats
    - Android Sparse Images
    - super[\_empty].img
    - Block Based Updates (transfer.[dat|list])
    - Mounting Filesystem Images
  - Android boot.img
  - Vendor boot.img (11.0, GKI)
  - Updates
    - Fastboot
    - Samsung: ODIN
    - Updates via recovery
    - The update\_binary
    - Updates on slotted (A/B) devices
    - update\_engine
  - Generic System Images (Android 9+)
    - gsid (Android 10+)
    - Dynamic System Update (DSU)
    - The dynamic\_system service
- 7. The Android boot process
 

Generalizing the Android Boot process amongst vendors, and then focusing on vendor specifics

  - The Boot ROM/PBL
  - Second Stage/eXtensible Boot Loader
    - Qualcomm (SD835+) UEFI Loader
    - Samsung S-BOOT
    - MediaTek Preloader
  - The Android Boot Loader
    - Little Kernel (32-bit, ARMv7 and ARMv8 non Qualcomm UEFI)
    - (Generalized) LK execution flow
    - LinuxLoader (Qualcomm UEFI)
  - Boot loader locking
  - The Linux Kernel
    - Kernel Boot
  - The RAM Disk (initramfs)
  - The Boot Control HAL
- 8. User mode startup - init and Zygote
  - Init
    - as watchdogd

- System Properties
  - Accessing properties
  - Special namespace prefixes
  - Property files
  - PropertyInit()
  - The property store
  - The property\_service
- The rc files
  - Triggers, actions, and services
  - init.rc syntax and command set (updated for 11.0)
    - Command syntax
    - Service option syntax
    - Keychords
  - Putting it all together
- Zygote
  - Design Rationale
  - Zygote32, Zygote64 and webview\_zygote
  - UnSpecialized Application Processes (USAPS, Android 10)
- Android Daemons, at a glance

The Android Runtime services: Native Services chapter (formerly Chapter 5), has been removed, as now *all* daemons are covered, but each within its context

## 9. The Android Service Architecture

- The Service call pattern
- Binder (an overview)
  - A little history
  - So what, exactly, is Binder?
  - Using Binder
  - 8.0+: The vndbinder and hwbinder
  - Tracing Binder (bindump, etc)
- Service Manager, revisited
- The system\_server architecture
- Handling services
- Startup and Flow
- A bird's eye view of Android's services

## 10. Configuration & Management

New chapter dealing with users, settings and more

- User Management
  - The user service
- Account Management
  - The accounts database
  - The account service
- Configuration Settings
  - config.xml and other files
  - Overlays
  - The device\_config service
  - Server Configurable Flags (10.0)
  - The settings service
  - The etc/sysconfig directories
  - The system\_config service (11.0)
- Mobile Device Management - Moved to Volume III
  - Work profiles
  - The device\_policy service
  - The restrictions service

## 11. Android Applications through Linux Lens

Monitoring and viewing Applications through the Linux command line

- Application during runtime (with /proc/task/..)

- User mode memory management
- USS, PSS, RSS, VSS, etc
- procrank, librank, and /proc/./smaps
- Native binaries, libraries and ELF Tools
- Optimizations in Android native and Dalvik apps

## 12. Logging, Statistics & Monitoring

- Android Logging
  - logd
- Statistics
  - statsd
  - statscompanion
  - The IStats HIDL
  - Lesser Statistics Services
- Incident Reporting
- Vendor Diagnostics
  - Qualcomm's Diag (/dev/diag)
- Debugging
- Monitoring
  - inotify
  - pt race(2)-based tools
    - strace
    - jtrace
  - Using eBPF for tracing

## 13. Power Management

- Native APIs
- The PowerManagerService and Friends
- Battery Monitoring
- Low-level CPU Control
  - MultiCore
  - Interrupt Affinity
  - Governors
  - Heterogeneous Multi-Processing (HMP) Scheduling
  - Energy Aware Scheduling (EAS)
- Thermal Monitoring
  - Linux kernel support
  - Android support
    - hardware\_properties service
    - thermalservice
    - The thermal HAL
  - Vendor thermal support
    - Qualcomm
    - Samsung
    - Huawei
    - MediaTek
    - Case study: Google Pixel
- The Power HAL interface
- Power Management Statistics

---

**Volume II: The Developer View - Available ! 360 pages**

## 1. Building Android from the source

- The AOSP

- [A whirlwind tour of Android projects](#)
- The NDK
  - [Android.mk and Android.bp \(soong\)](#)
  - [Cross compiling with custom Makefiles](#)
- 2. Android at a Native Level
  - Bionic, in depth
  - Native Level debugging, core dumps and tombstones
- 3. Package Maintenance
  - APK Components
    - AndroidManifest.xml
    - classes.dex
    - resources.arsc
    - Digital signatures on apps
  - Runtime Resource Overlay (RRO)
  - Package Installation
    - Behind the scenes
    - installD
    - The package database
    - Monitoring Packages
    - Package statistics
    - The PackageManagerService
    - APK snapshots & rollback
- 4. Anatomy of an an Android Application
  - [Break down and detail of APK and application components](#)
  - Application Components
    - Activities
    - Services
    - Broadcast Receivers
    - Content Providers
  - JNI
    - The need for native code
    - Compiling JNI code
    - Houdini - Briding the ARM/Intel divide
  - Decompiling applications
  - Application Internals
    - [Break down and detail the nooks of crannies of starting an app, and maintaining its lifecycle](#)
    - Runtime Primitives
      - Looper
      - Handler
      - MessageQueue
    - Zygote explained
    - Accessing services from native code
    - Activity Manager in depth
    - Behind the scenes of the application lifecycle
- 5. Application Services
- 6. Dalvik Internals
  - [The inner workings of Android's Virtual Machine and bytecode format](#)
  - Dalvik vs. Java
  - DEX, demystified
  - Running DEX apps
  - Dalvik's JNI implementation

## 7. Android RunTime Internals

- ART - An alternative to Dalvik
- ART Components (The com.android.art APEX)
- The files: OAT, ART, CDEX, VDEX, etc
- The runtime
  - Setup and initialization
  - Support threads
- Compilation
- JIT
- Profiling/Tracing
- Memory allocators and Garbage Collection

## 8. Binder, in depth

- A Brief Overview of Binder
  - The App Developer's Perspective - AIDL
    - AIDL Syntax
    - AIDL code generation
    - The Parcel object
    - The Parcel wire format
  - The Framework Perspective - `android.os.Binder`
    - `Binder.java`
    - References
    - Default Transactions
    - 11.0: Extensions
    - Death Notifications
    - The JNI Layer
  - The native code Perspective - `libbinder.so`
    - `RefBase`: Strong and Weak Pointers
    - The `BpBinder` and `BBinder`
    - The native proxy and stub interfaces
    - `ProcessState` and `IPCThreadState`
    - Detailed case study: A Binder service in native code
  - The Binder kernel interface
    - The Binder character devices
      - 11.0: `BinderFS`
      - The `ioctl(2)` command set
    - The `BINDER_WRITE_READ ioctl(2)` code
    - Transactions
    - Flattened Objects
  - Binder Driver Internals
    - Module initialization
    - Device open
    - Transactions in-kernel processing
    - Kernel Driver State
    - Thread Management
    - Death Notifications
  - Tracing & Debugging
- 

## Volume III: The [Hacker/Security Analyst]'s View

This was chapter 8 in the 1<sup>st</sup> edition - but that was a long time ago, before TrustZone, Titan, AVB ... and a host of Android exploits & APTs..



## 1. Software-implemented security

- Linux Native Permissions
  - AID ranges
  - Treble and the return of passwd/group files
- SELinux
- SECCOMP-BPF
- Android Runtime permissions
- Appops

## 2. Hardware-backed security

- TrustZone
  - Theory & Design
  - Vendor Implementations:
    - Qualcomm: QSEE/QHEE
    - MTK/Older Samsung: Mobicore
    - Samsung: TEEGRIS
    - Google: Trusty
- Beyond Trustzone: Hardware Security Modules
  - Titan M/M2
  - Qualcomm SPU

## 3. Authentication subsystems

- The Lock Screen (lock\_settings service)
- The auth service
- The biometric service
- Face authentication (The face service)

## 4. Encryption facilities

- DM-Crypt
- Ext4Crypt
- Keystore
- Linux keyrings
- Gatekeeper

## 5. Integrity & Attestation

- Android Verified Boot
    - AVB 1.0
    - AVB 2.0
    - AVBMeta tool
  - DM-verity
  - 11: App Integrity, File Integrity (fs-verity)
  - Samsung TIMA & Knox
  - Google SafetyNet
- 

## 6. Introduction/Threat Modeling Android

[Lorem ipsum](#)

- Threat Modeling
- Attack classes
  - ..
  - ...
- Android Security Model

## 7. Rooting

[Rooting Android using boot-to-root methods](#)

- 
- Android OEMUnlock interface
  - ...
  - Case Study: Magisk
  - Malware Case Study: Intellexa's "Alien"
8. Vulnerability/Exploit case studies:  
(Jury's still out on which of those I'll use - comments/suggestions welcome)
- Linux Kernel: CVE-2021-1048 (epoll) or CVE-2022-0847 (Dirty Pipe)
  - AOSP Linux Kernel: Bad Binder (CVE-2019-2215) and/or num\_not\_so\_valid CVE-2020-0041
  - Vendor: Pixel 6 - Samsung's MFC
  - TrustZone: likely Trusty
  - AOSP: (still looking for something nice here)
  - Vendor: MTK-su and/or Boot chain vulnerability?
  - Baseband: Samsung Exynos (Shannon) VoLTE/SIP vulns
9. Appendices:
- Android App Hardening Guide
  - Android System Hardening Guide
- 

## Volume IV: The Implementer's View

1. HAL & Treble
  - The Hardware Abstraction Layer (pre-Treble)
  - hwservicemanager
  - HIDL and Binderized HAL
  - sensors/CHRE
    - Oslo/Soli as a case study
  - GPS
  - Implementing a custom HAL module
2. The Android Input Architecture
  - The Linux Kernel Layer
  - The Native Layer
    - InputManager
    - EventHub
    - InputReader
    - InputDispatcher
  - The Dalvik Layer
    - The Input Pipeline
    - Getting to the user callback
3. Android Media
  - The Audio Architecture
    - Audio at the Linux Kernel level
    - The Audio HAL
      - Audio modules
      - Audio policy modules
      - Audio Effects
    - AudioFlinger
    - Media Player
    - The Dalvik APIs
  - Video
  - Digital Rights Management

- Android Graphics Architecture
- Graphics at the Linux Kernel Level
- Graphics at the Native Level
  - SKIA
  - OpenGL ES
  - RenderScript
  - MinUI
- SurfaceFlinger
- StageFright

## 5. Connectivity

- Bluetooth
- Android Beam & NFC
- Wi-Fi & Wi-Fi Direct
  - wpa\_supplicant
  - WifiMonitor
  - WifiNative
  - WifiConfigStore
  - WifiStateMachine
- VPN (Racoon and MTPd)
- Tethering and Mobile Access Point
  - Kernel Layer: NetFilter
  - User Mode support
    - hostapd
    - dnsmasq
- Detecting Network State
- Monitoring Data Usage

## 6. Telephony

- Radio interface layer (phone)

## 7. Location

## 8. Android and USB

- Android as a USB Target
  - Framework USB Target Support
  - The Linux Gadget Driver
  - ADB
    - Authenticated ADB
  - MTP/PTP
  - Mass Storage Device
  - RNDIS (USB Tethering)
- Android as a USB host
  - Framework USB Host Support