Mobile Security

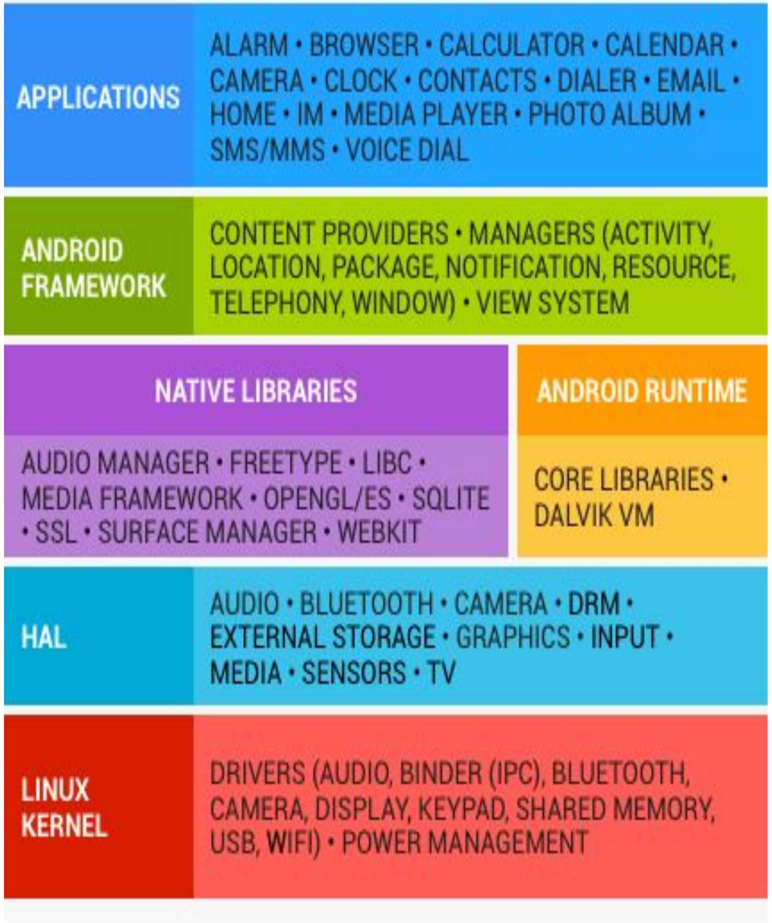
Dr. Eleonora Losiouk
Department of Mathematics
University of Padua
elosiouk@math.unipd.it
https://www.math.unipd.it/~elosiouk/

UNIVERSITÀ DEGLI STUDI DI PADOVA

SPRITZ SECURITY & PRIVACY RESEARCH GROUP

DIPARTIMENTO MATEMATICA

Android Framework Architecture (reloaded)



APPLICATIONS

ALARM • BROWSER • CALCULATOR • CALENDAR • CAMERA • CLOCK • CONTACTS • DIALER • EMAIL • HOME • IM • MEDIA PLAYER • PHOTO ALBUM • SMS/MMS • VOICE DIAL

ANDROID FRAMEWORK

CONTENT PROVIDERS • MANAGERS (ACTIVITY, LOCATION, PACKAGE, NOTIFICATION, RESOURCE, TELEPHONY, WINDOW) • VIEW SYSTEM

NATIVE LIBRARIES

AUDIO MANAGER • FREETYPE • LIBC • MEDIA FRAMEWORK • OPENGL/ES • SQLITE • SSL • SURFACE MANAGER • WEBKIT

ANDROID RUNTIME

CORE LIBRARIES • DALVIK VM

HAL

AUDIO • BLUETOOTH • CAMERA • DRM • EXTERNAL STORAGE • GRAPHICS • INPUT • MEDIA • SENSORS • TV

LINUX KERNEL

DRIVERS (AUDIO, BINDER (IPC), BLUETOOTH, CAMERA, DISPLAY, KEYPAD, SHARED MEMORY, USB, WIFI) • POWER MANAGEMENT

Image from https://source.android.com/security

Android Framework Architecture (reloaded)

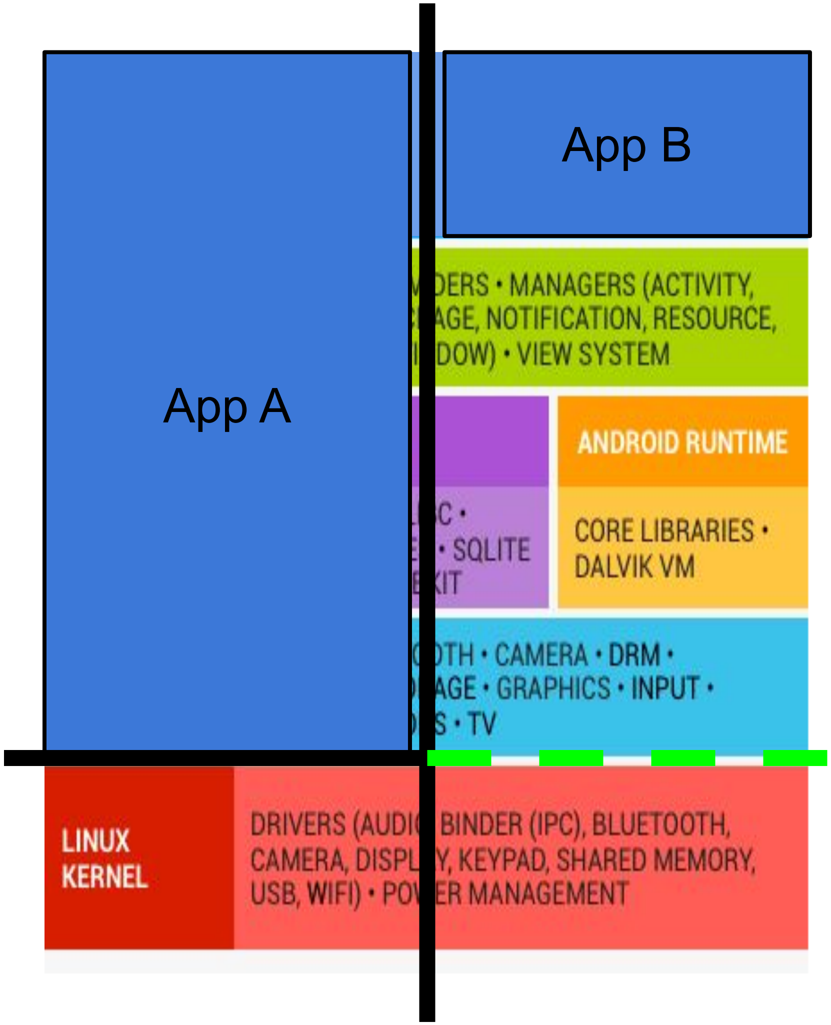

App B
App A
DERS • MANAGERS (ACTIVITY, AGE, NOTIFICATION, RESOURCE, DOW) • VIEW SYSTEM
ANDROID RUNTIME
C • • SQLITE KIT
CORE LIBRARIES • DALVIK VM
TH • CAMERA • DRM • AGE • GRAPHICS • INPUT • S • TV
LINUX KERNEL
DRIVERS (AUDI BINDER (IPC), BLUETOOTH, CAMERA, DISPL Y, KEYPAD, SHARED MEMORY, USB, WIFI) • PO ER MANAGEMENT

Image from https://source.android.com/security

Building blocks for security

- Google security services
- Android OS / Linux kernel
- Device hardware

Google Security Services

Google Security Services

- Google Play
  - A central place from where to install apps
  - Developers and "apps on your phone" are linked via app signatures
  - Community reviews, app security scanning, etc.
- Android Updates
  - Updates via the web or Over The Air (OTA updates)
- Monthly Security Updates
  - A new security update every month (available via OTA)
  - https://source.android.com/security/bulletin/index.html

Google Security Services

- Bug report / Triaging
- Process type
  - Constrained process, Unprivileged process (third-party app), Privileged process (app with more privileged than unprivileged ones), Trusted Computing Base (TCB, part of the kernel, baseband processor, etc.), Bootloader, Trusted Execution Environment
- Local vs. Remote
- Severity: Critical, High, Moderate, Low

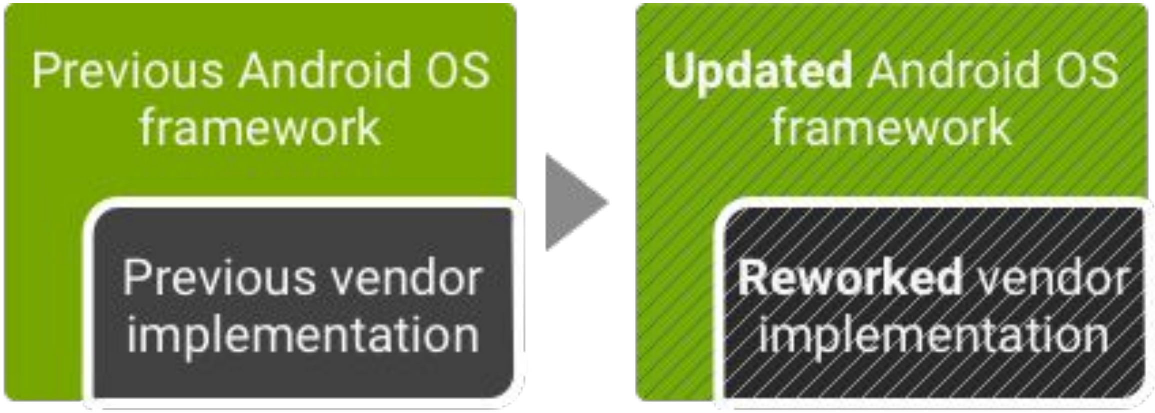
Project Treble

- Android's core problem
  - Fragmentation
  - Many devices don't get updates
  - The release update cycle is slow, especially for non-Google devices

- One core problem
  - Previously: no clear interface between Android OS framework and vendor implementation
  - When the Android OS is updated, vendors need to rework their parts
    - This introduces delays

Project Treble

- Up to Android 7.x



Previous Android OS framework

Previous vendor implementation

Updated Android OS framework

Reworked vendor implementation

Image from https://source.android.com/devices/architecture

Project Treble

- In Android 8.0+ (with Project Treble)

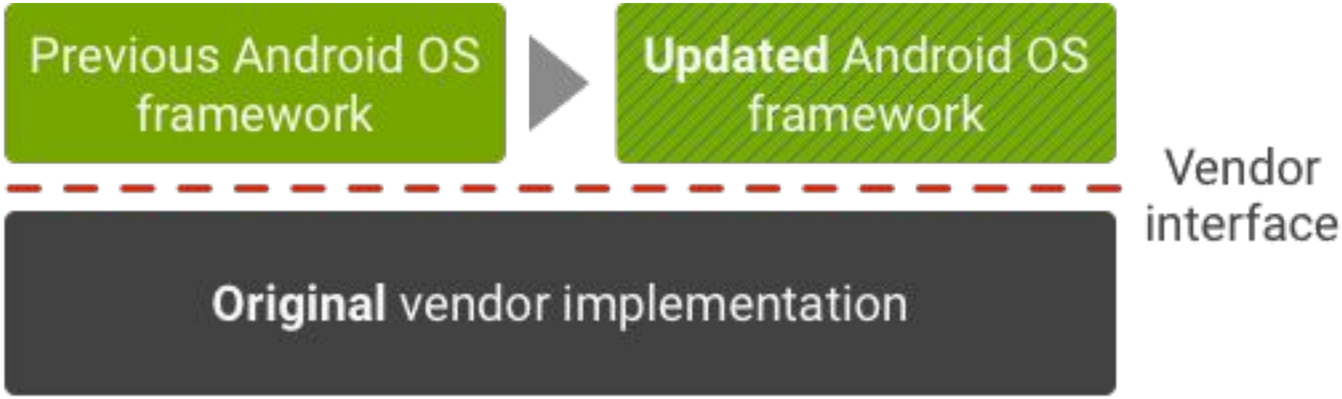

Previous Android OS framework
Updated Android OS framework
Vendor interface
Original vendor implementation

Image from https://source.android.com/devices/architecture

Google Security Services

- Application Services
- Frameworks that allow Android apps to use cloud capabilities
- Examples
  - Backup service (doc)
  - Firebase Cloud Messaging (FCM) (doc)
    - It allows "push" notifications from cloud to device

Google Security Services

- SafetyNet
  - "A privacy preserving intrusion detection system to assist Google tracking and mitigating known security threats"
  - It runs on all Google phones
- SafetyNet Attestation
  - API to determine whether the device is "CTS compatible"
    - CTS ~ "Android Compatibility Tests"
    - It collects a software + hardware profile in a "trusted" way
  - Check whether the app has been modified by an unknown source

Google Security Services

- Verify Apps
  - System run by Google that continuously scan apps on the device
  - Actually: continuously check the cached result of an app's analysis
- Android Device Manager
  - Web app + Android app to locate lost / stolen devices
  - https://www.google.com/android/find

Kernel Security & SELinux

Linux Security

- A user-based permission model
- Process isolation
- Extensible mechanism for secure IPC
- The ability to remove unnecessary components

Linux's guarantees

- Prevents user A from reading user B's files
- ~~Ensures that user A does not exhaust user B's memory~~
- ~~Ensures that user A does not exhaust user B's CPU resources~~
- ~~Ensures that user A does not exhaust user B's devices (e.g. telephony, GPS, Bluetooth)~~

The Application Sandbox

- Each app is isolated from each other
    - How? Each app is assigned to a different user
- The sandbox is in the kernel
    - Native code can't bypass it
- To bypass it, an attacker would need to compromise Linux

Defense in Depth

- Multiple layers of security controls are placed throughout a system
- It prevents single vulnerabilities from leading to compromise of the OS or other apps
- Rephrasing: redundancy

DAC vs MAC

- DAC: Discretionary Access Control
  - Each resource has a list of users who can access it
  - Permission is set by the data owner
  - Example: Linux file permission

- MAC: Mandatory Access Control
  - There are a number of levels, each user has a specific level
  - A user can access all resources with non-greater level than hers
  - Example: SELinux policies

SELinux

- Security-Enhanced Linux: a Linux kernel security module
  - Useful to define access control security policies
- "Deny by default" policy
- Two modes
  - Permissive mode: permissions denials are logged but **NOT** enforced
  - Enforcing mode: permission denials are logged **AND** enforced

SELinux

- "SELinux domain"
  - A label identifying a process (one or more)
  - All processes labeled with the same domain are treated equal
- Early versions in Android: installd, netd, vold, and zygote
- Current version: 60+ domains

SELinux in Android 5.0

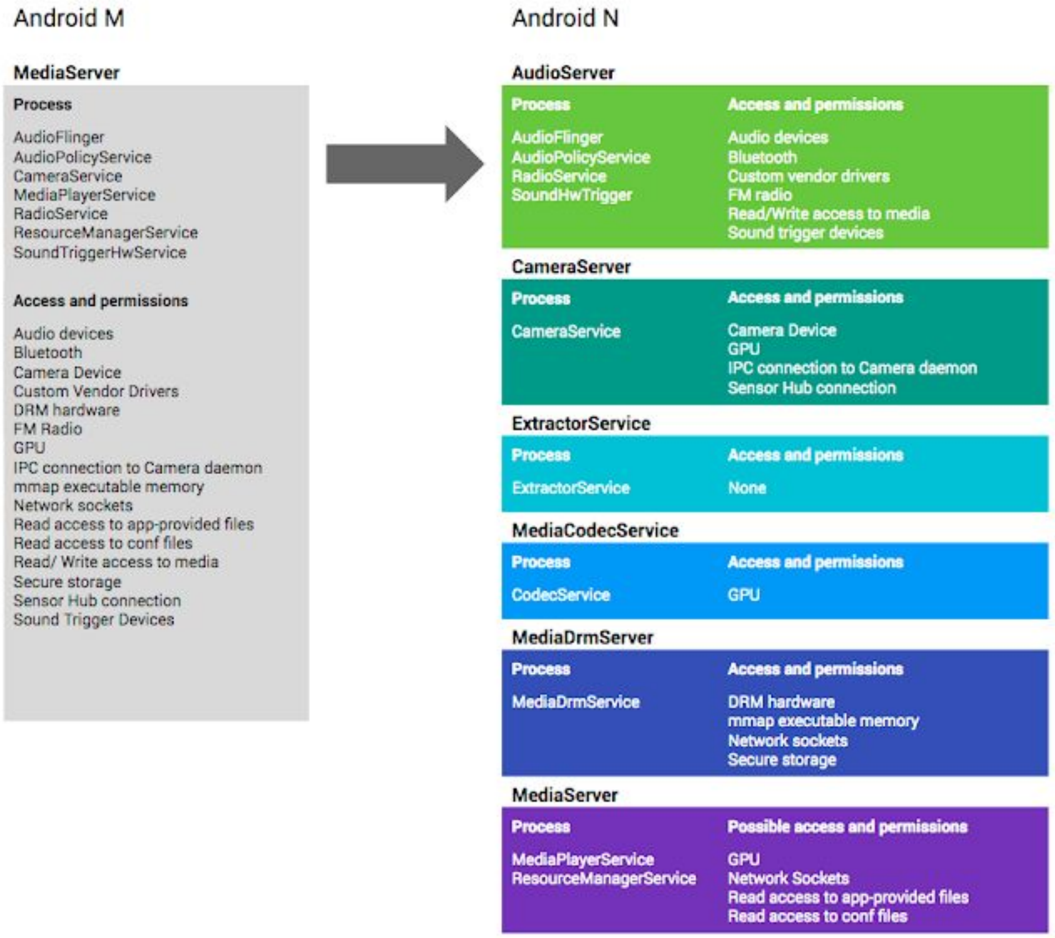
- MAC separation between system & apps
- Run in "enforcing" mode for the first time
- Redundancy?

SELinux in Android 6.0

- Makes the policy more restrictive / tighter domains
- IOCTL filtering
  - Minimize exposed services
- Extremely limited /proc access
- SELinux sandbox to isolate across per-physical-user boundaries
  - App's home dir default permissions changed from 751 to 700

SELinux in Android 7.0

- It broke up the monolithic mediaserver stack into smaller processes



Android M

MediaServer

Process

AudioFlinger
AudioPolicyService
CameraService
MediaPlayerService
RadioService
ResourceManagerService
SoundTriggerHwService

Access and permissions

Audio devices
Bluetooth
Camera Device
Custom Vendor Drivers
DRM hardware
FM Radio
GPU
IPC connection to Camera daemon
mmap executable memory
Network sockets
Read access to app-provided files
Read access to conf files
Read/ Write access to media
Secure storage
Sensor Hub connection
Sound Trigger Devices

Android N

AudioServer

Process	Access and permissions
AudioFlinger AudioPolicyService RadioService SoundHwTrigger	Audio devices Bluetooth Custom vendor drivers FM radio Read/Write access to media Sound trigger devices

CameraServer

Process	Access and permissions
CameraService	Camera Device GPU IPC connection to Camera daemon Sensor Hub connection

ExtractorService

Process	Access and permissions
ExtractorService	None

MediaCodecService

Process	Access and permissions
CodecService	GPU

MediaDrmServer

Process	Access and permissions
MediaDrmService	DRM hardware mmap executable memory Network sockets Secure storage

MediaServer

Process	Possible access and permissions
MediaPlayerService ResourceManagerService	GPU Network Sockets Read access to app-provided files Read access to conf files

SELinux in Android 8.0

- All apps run with a seccomp-bpf filter that limits syscalls that apps can use (attack surface reduction)

SELinux in Android 9.0

- Each app has an individual SELinux context
- It prevents apps from making their data world-writable

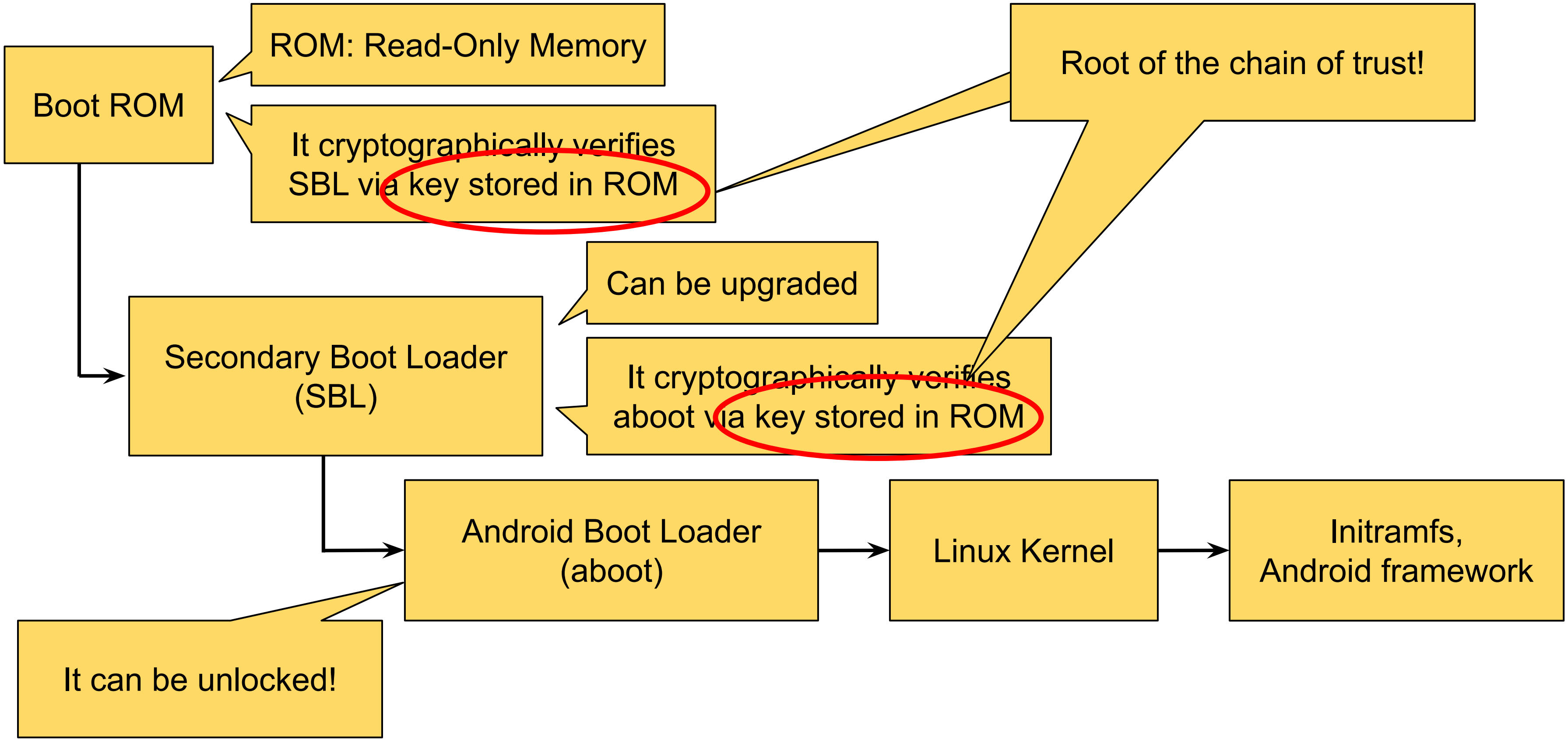
SELinux in Android X.Y?

- SELinux policies (and Android itself) are in constant evolution
- Things that work now may not work in the future

Boot and Verified Boot

The boot process and Verified Boot

- The system boots in "stages"
  - Each stage loads and verifies the next one


Boot ROM
ROM: Read-Only Memory
It cryptographically verifies SBL via key stored in ROM
Root of the chain of trust!
Can be upgraded
Secondary Boot Loader (SBL)
It cryptographically verifies aboot via key stored in ROM
Android Boot Loader (aboot)
Linux Kernel
Initramfs, Android framework
It can be unlocked!

The boot process and Verified Boot

- "Unlock bootloader"
  - aboot will NOT enforce the chain of trust over the subsequent stages
  - That's how you can install custom mods, etc.
- aboot itself CANNOT be changed
  - Well, you can. But SBL will not load it. High risk of bricking the device.
- Same for the other partitions (boot, system)
  - You can change them, but if aboot is locked, it refuses to load them
- Not all devices allow bootloaders to be unlocked

fastboot

- fastboot is a tool / protocol for writing data directly on the device's memory

  ```
  $ fastboot devices
  $ fastboot flash system system.img
  ```

  - This flashes system.img to the system partition

- It's "implemented" in aboot... pay attention

Example: How to unlock Pixel 3's bootloader

- Boot device in "bootloader mode" (or "fastboot" mode)
  - Technique #1
    - press power + volume down button while booting
    - a special menu will appear
  - Technique #2
    - $ adb reboot bootloader # This tells the device 'go in bootloader mode'

- $ fastboot flashing unlock
- Check the device's screen: confirm to unlock

Additional security mechanisms

- Device's data is wiped upon bootloader unlock
- If unlocked, the bootloader shows a warning to the user every time the device boots

Additional security mechanisms

- By default, you cannot unlock the bootloader
- "Allow OEM unlocking" settings
  - It's in the "developer options" 'hidden' menu
    - Settings -> System -> About phone -> Tap on "build number" 7 times
    - Developer options -> Allow OEM unlocking (this may ask for PIN)
    - Developer options -> Allow USB debugging
- If a thief steals my phone, she can't do anything with it

Device State

- The "device state" indicates how freely software can be flashed to a device and whether verification is enforced

- Possible states: LOCKED and UNLOCKED

- LOCKED devices boot only if the OS is properly signed by the root of trust

Flash new image

- You can flash each partitions with new data

```
$ fastboot flash system system.img
```

- Factory images from Google (link) come with a flash-all.sh script

flash-all.sh

```
$ cat flash-all.sh

fastboot flash bootloader bootloader-bullhead-bhz10m.img
fastboot reboot-bootloader
sleep 5
fastboot flash radio radio-bullhead-m8994f-2.6.31.1.09.img
fastboot reboot-bootloader
sleep 5
fastboot -w update image-bullhead-mhc19q.zip
```

image-bullhead-mhc19q.zip

```
$ unzip -l image-bullhead-mhc19q.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
      101  2009-01-01 00:00   android-info.txt
2005102896 2009-01-01 00:00   system.img
 11793638  2009-01-01 00:00   boot.img
195274360  2009-01-01 00:00   vendor.img
 12870890  2009-01-01 00:00   recovery.img
  5824660  2009-01-01 00:00   cache.img
139966976  2009-01-01 00:00   userdata.img
---------                     -------
2370833521                     7 files
```

Many partitions

- boot
  - It contains a kernel image
  - ramdisk: small partition, /init & config files, mount other partitions
- system
  - It contains everything that is mounted at /system
  - Android framework, system apps
- vendor: Binary that is not part of AOSP
- userdata: It contains everything that is mounted at /data, third-party apps
- radio: the 'radio' image, super sensitive, run on its own processor
- recovery: like boot, but for 'recovery mode'

Recovery mode

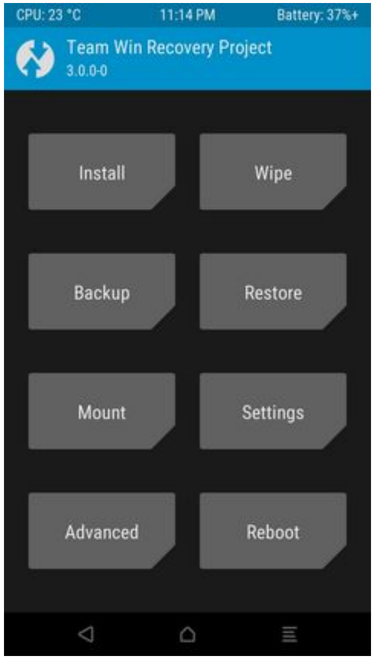
- Two modes of booting your device
  - Go in bootloader mode -> then choose "start" vs. "recovery mode"
- "Normal mode"
  - The system you are used to know, Android OS, etc.
  - boot partition -> system -> vendor / userdata
- Recovery mode
  - By default, empty
  - Very basic system to perform "admin" operations
  - Once the bootloader is unlocked, you can flash what you want here

TWRP

- Team Win Recovery Project (TWRP) is a custom recovery image for Android-based devices.

- https://dl.twrp.me/

```
$ fastboot flash recovery <twrp.img>
```



Protection of System Partition

- The "system" partition contains Android's kernel, OS libraries, application runtime, "the framework", and pre-installed apps
- The "system" partition is read-only
- It helps preventing attacker's persistence on the device

Safe Mode

- You can boot the device in "safe mode"
  - Usually: press device's power button + volume down button when the animation starts
- In safe mode, third-party apps are not started automatically
  - But they can be launched "manually" by the owner
- It prevents user-space attacker's persistence

Data Encryption

Data encryption (doc)

- User-created data are encrypted before writing to disk

- Full-disk encryption (Android 5.0+)
  - One single key protected by the user's device password
  - The user must provide her credentials upon boot
  - UX problems: nothing works without password, not even alarm clocks
  - What if the user changes her password? Not a problem...

Data encryption (doc)

- File-based encryption (FBE) (Android 7.0+)
  - Files are encrypted with different keys, which can be unlocked independently
  - Apps can work in a limited context before full unlock
  - It makes work profiles more secure: not only one "boot-time password"

Metadata Encryption

- FBE encrypts file contents and names

- FBE does NOT encrypt other informations
  - Directory layout, file sizes, permissions, timestamps

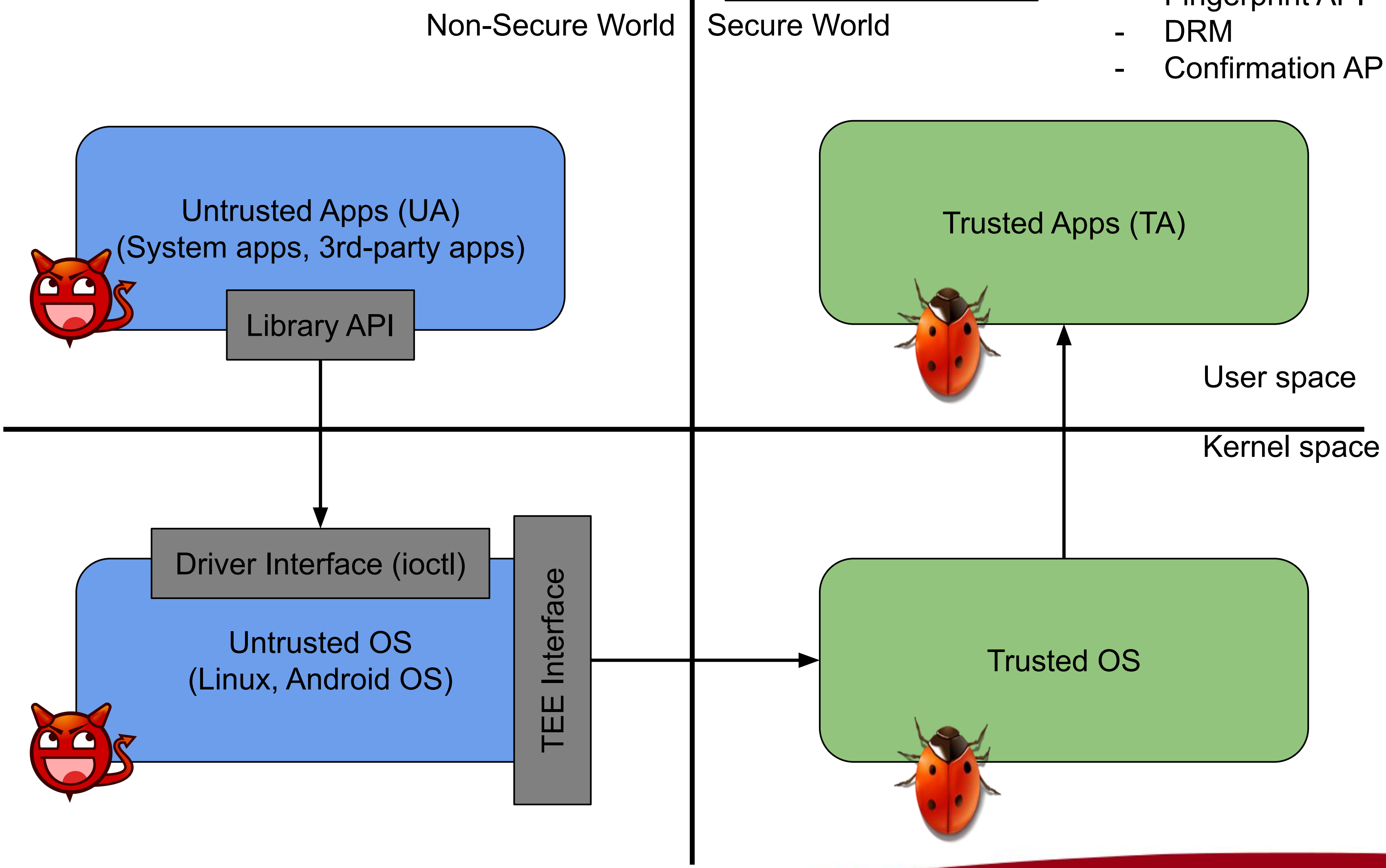
- Android 9 has support for metadata encryption
  - It encrypts whatever is not encrypted by FSE
  - It needs hardware support

TrustZone

ARM TrustZone

Secure even if the Linux/Android OS is compromised!!!

- KeyMaster
- Fingerprint API
- DRM
- Confirmation API



Non-Secure World

Secure World

Untrusted Apps (UA)
(System apps, 3rd-party apps)

Library API

Trusted Apps (TA)

User space

Kernel space

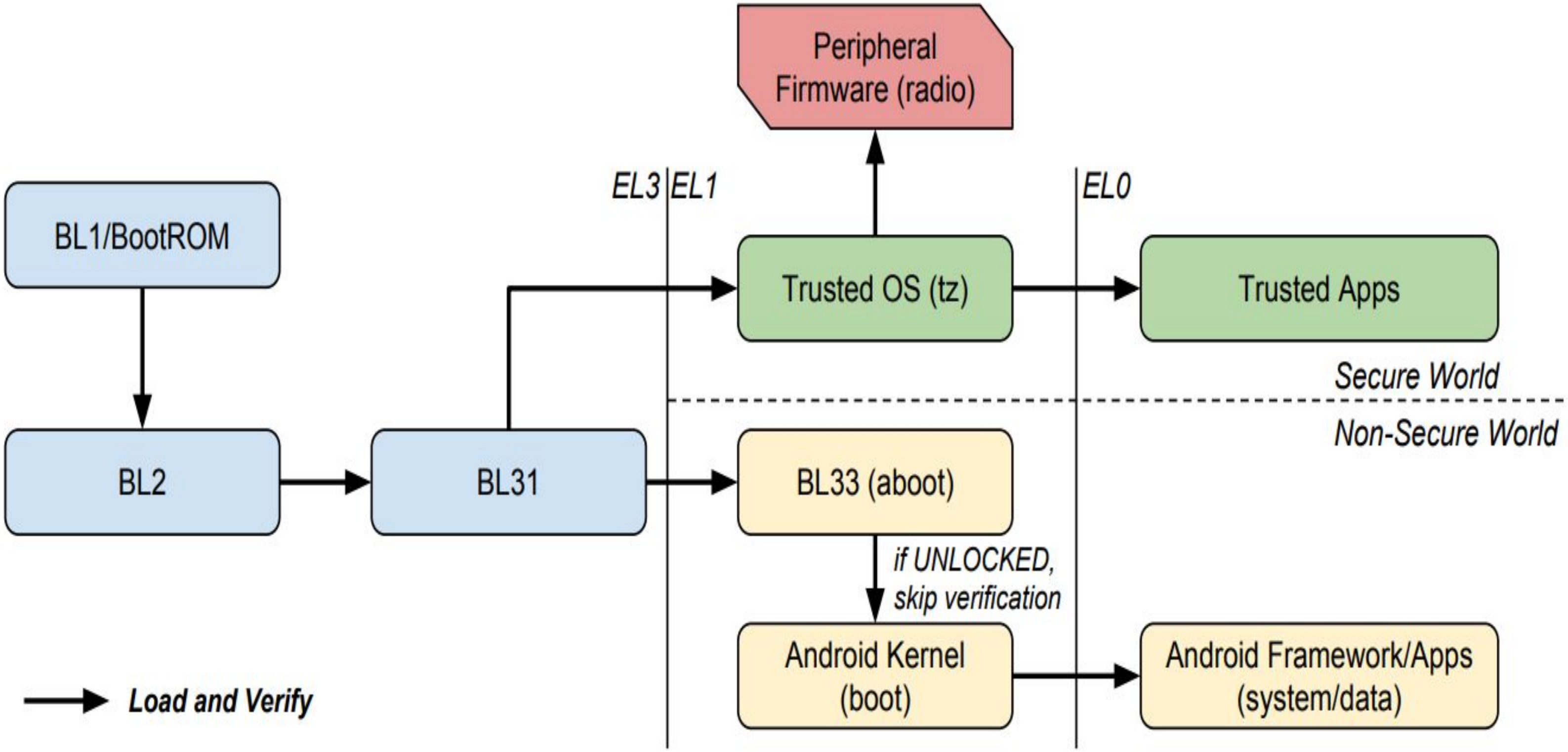
Driver Interface (ioctl)

Untrusted OS
(Linux, Android OS)

TEE Interface

Trusted OS

Verified Boot (reloaded)



Questions? Feedback? Suggestions?