Introduction

Lecture 1

Security of Mobile Devices

2023

Outline

SMD

Team and Schedule

- Team
  - Laura Ruse, Cosmin Chenaru, Adrian Florescu, invited speakers
- Schedule
  - Lecture: Monday 8-10
  - Labs: Thursday 8-10, 10-12, 18-20

Syllabus

- Android OS:
  - SDK
  - Internals
  - Security architecture
  - Network security
  - Vulnerabilities and malware
- Invited speakers from industry

Resources

- Wiki: `http://ocw.cs.pub.ro/courses/smd`
  - Lectures
  - Labs
  - Class registrer
  - Calendar
- Moodle: `http://curs.upb.ro/`
  - `https://curs.upb.ro/2022/course/view.php?id=4873`

Grading

- **0.5 points** Lecture tests and attendance
- **1 point** Lab activity
- **4 points** Project
- 2.5 points from lab, project & tests are required to enter the exam
- **2 points** Mid-term exam
- **2.5 points** Final exam
- A total of 5p are required to pass the class

Grading

- **0.5 points** Lecture tests
  - the test will be held at the end of the lecture, on Moodle
  - the test will consist of one simple question
  - the question will be related to what was presented at the lecture

Grading

- **1 point** Lab activity
  - Android Studio, Java, Kotlin (if you want to)
  - Github Classroom
  - The lab will be solved during the lab (it may be finished after the lab)
  - Submit until Sunday 23:55 (same week)

Grading

- **4 points** Project
  - `https://ocw.cs.pub.ro/courses/smd/res/assignment`
  - Project theme registration - 0.3p penalty
  - Intermediary project presentation - 0.5p penalty
  - Final project presentation

Grading

- **2 points** Mid-term exam
- **2.5 points** Final exam
- 20 multiple choice questions
- 20 minutes
- each question has 4 choices of which only one is correct

Recommended Bibliography

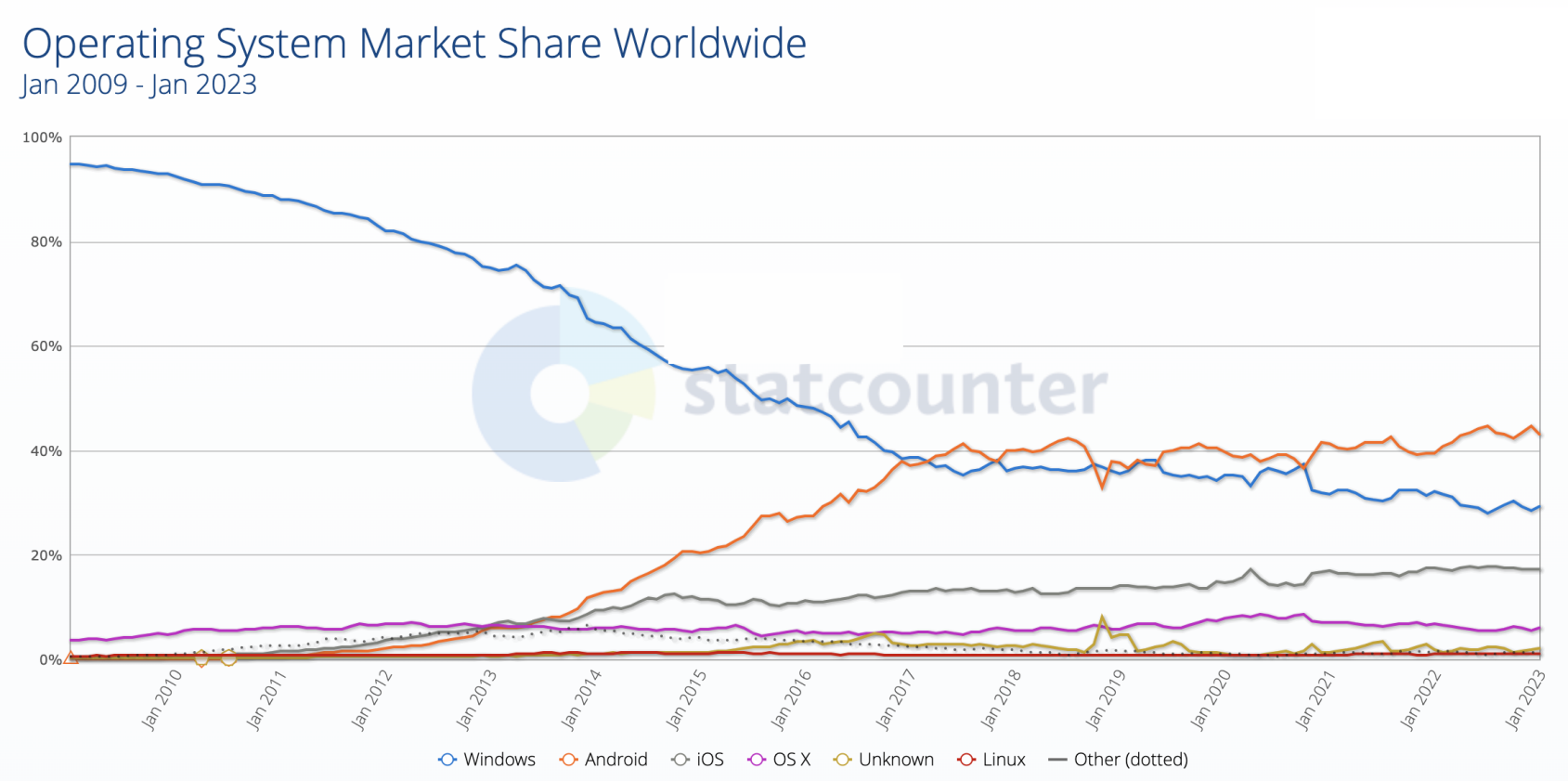
- `https://source.android.com/docs/`
- `http://developer.android.com`
- Introducere in sistemul de operare Android - Laura Ruse, Vlad Traistă-Popescu, 2021
- Securitatea sistemului de operare Android - Laura Ruse, Vlad Traistă-Popescu, 2021

Outline

Android popularity

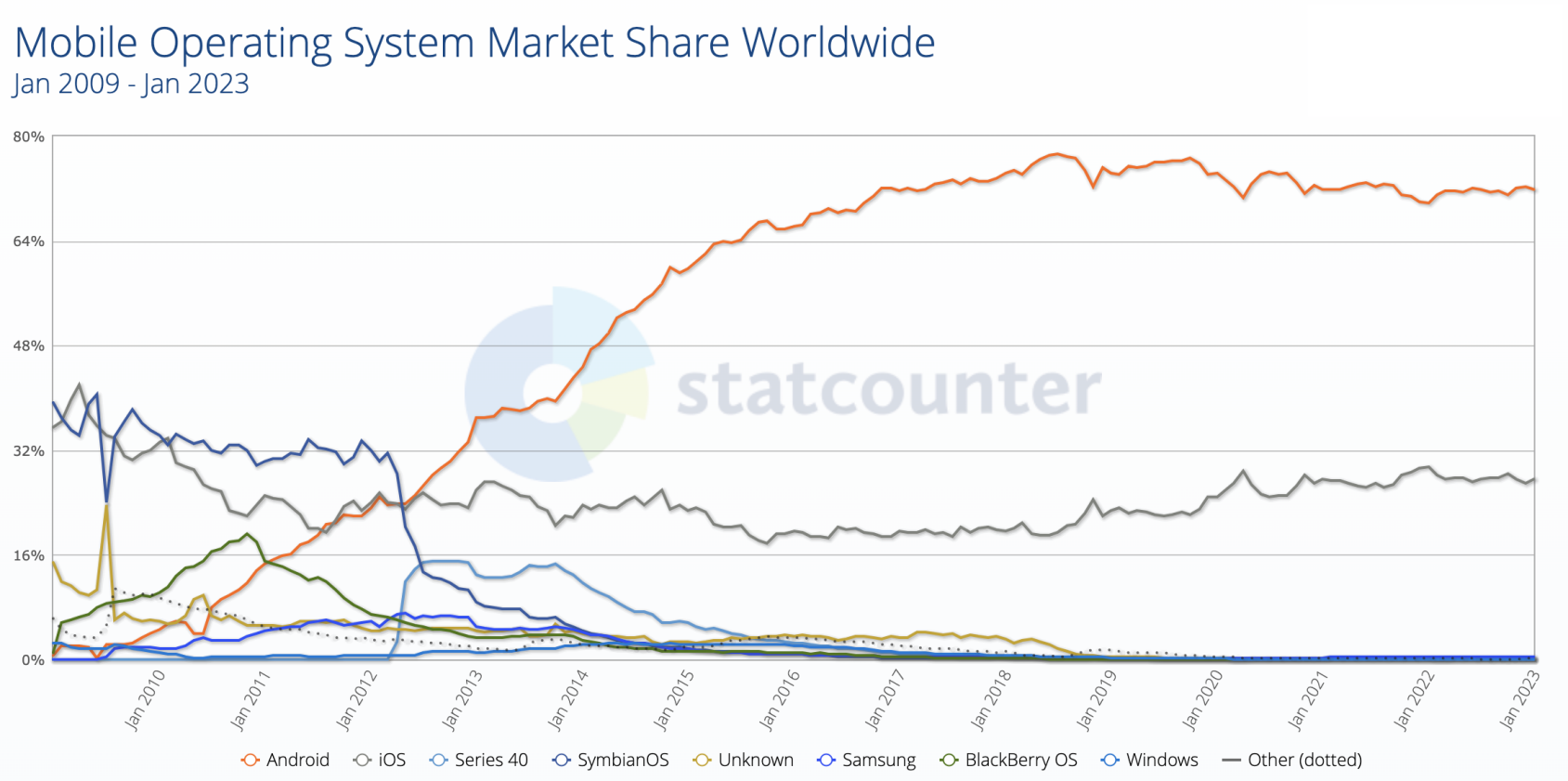
- Open-source OS for mobile devices
- 3.3 billion active users (2023)
- Mobile OS market share (Jan 2023)
  - Android 71.74%
  - iOS 27.63%
- OS market share (across all devices) (Jan 2023)
  - Android 43.01%
  - Windows 29.18%
  - iOS 17.24%
  - OS X 6.03%
  - Linux 1.15%
- Source: Statcounter
- Official application market: Google Play Store

OS Market Share Evolution



Source: Statcounter

Mobile OS Market Share Evolution



Source: Statcounter

Potentially Harmful Applications (PHA)

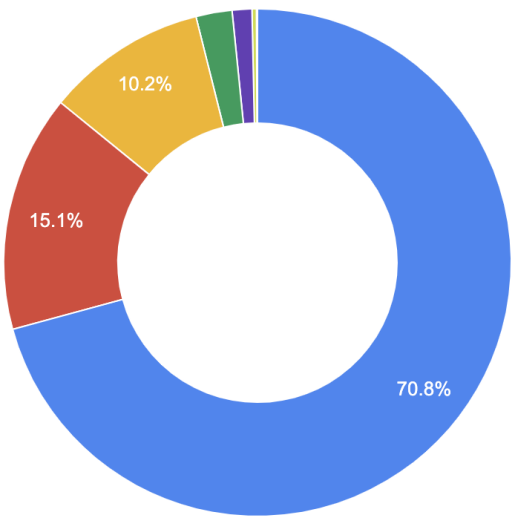
- Apps that could put users, user data and devices at risk
- Why the keyword potentially?

- Google Play Protect - detect and remove PHAs
- Statistics from Google:
- `https://transparencyreport.google.com/android-security/store-app-safety?hl=en`

PHAs detected by Google Play Protect

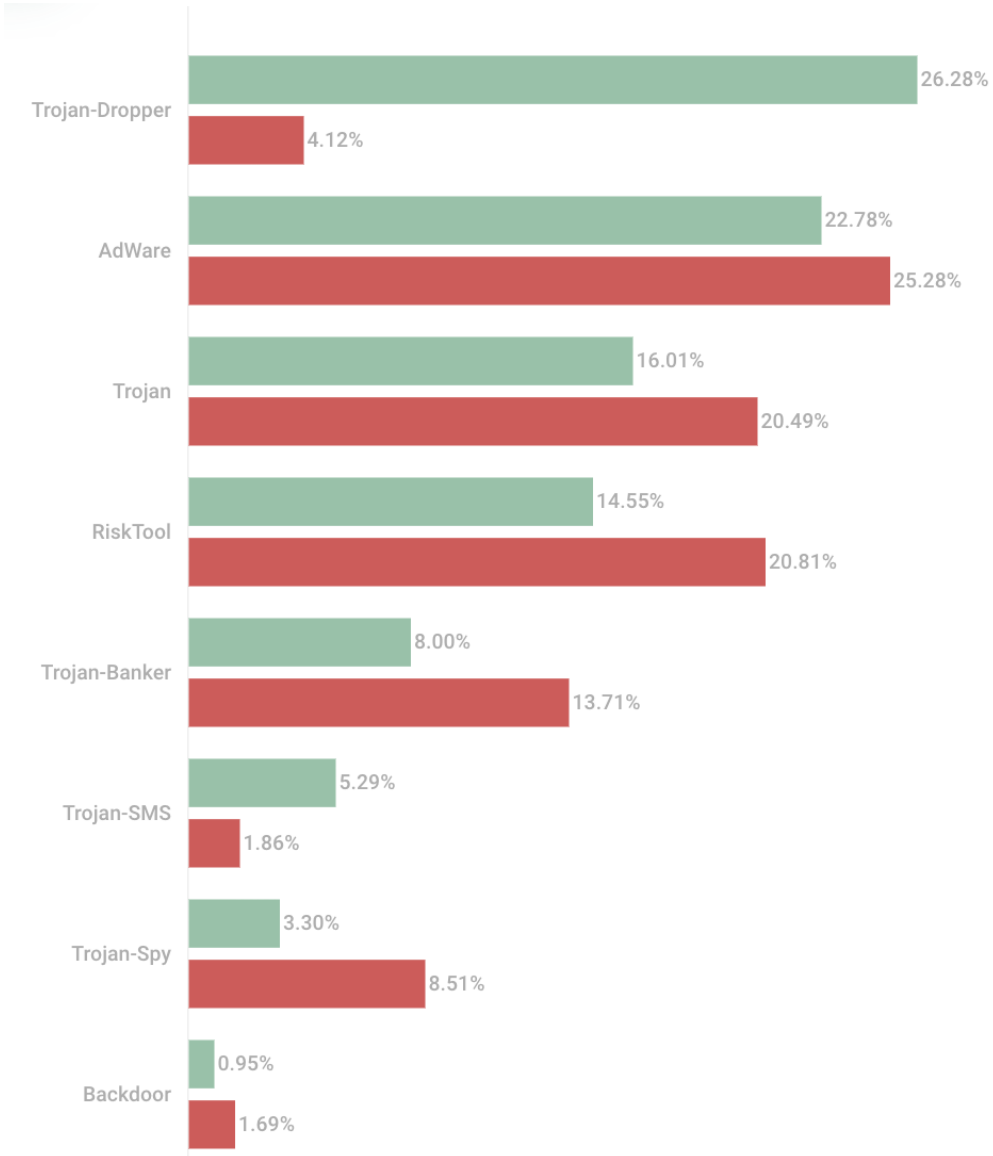
Jul 2022 – Sep 2022

Google Play



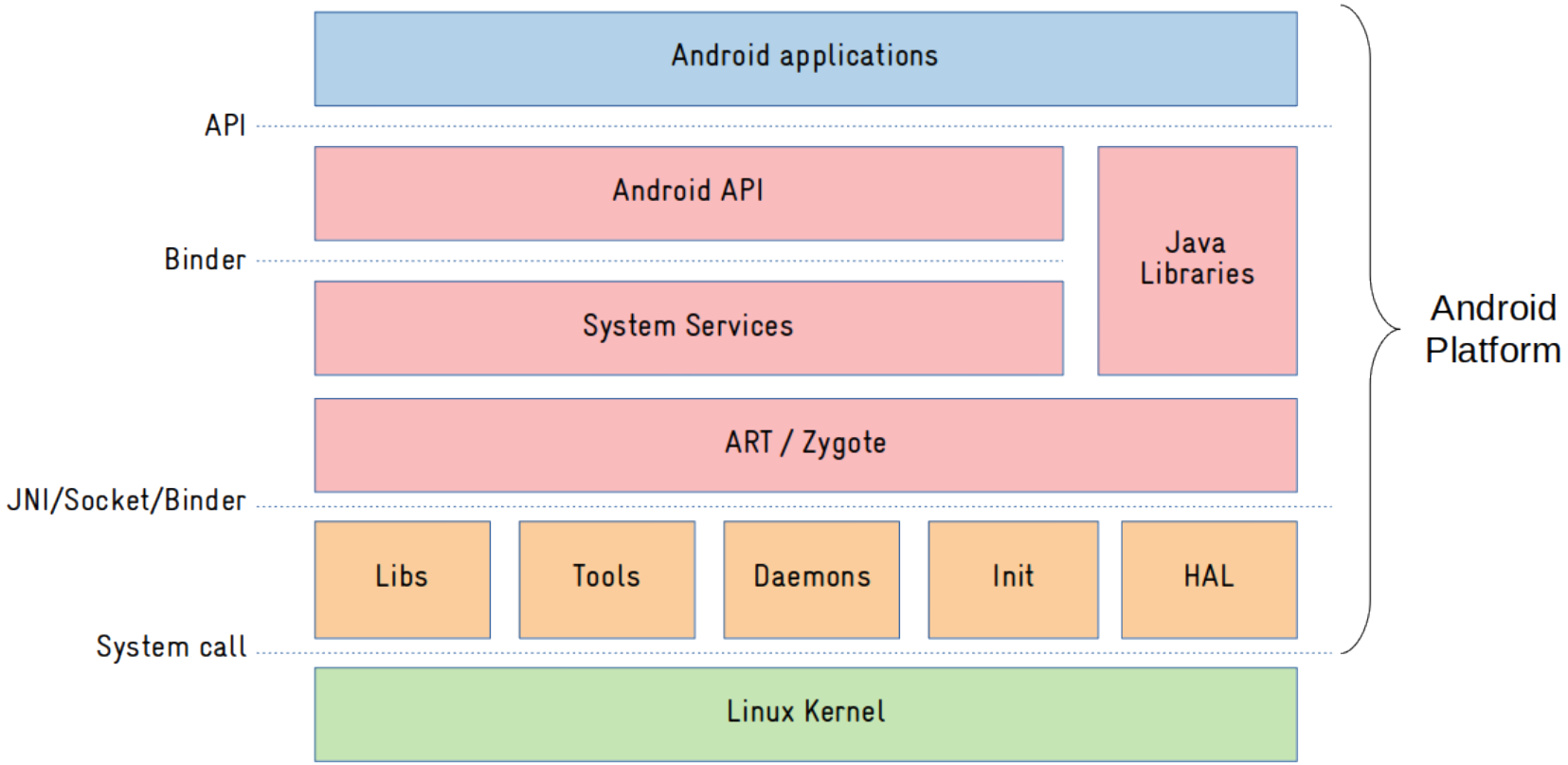
Category	PHA Install Rate
Privilege escalation	0.12902158%
Spyware	0.027533266%
Toll fraud	0.018666881%
Phishing	0.0041948882%
Backdoor	0.0022811875%
Trojan	0.0005138487%
Hostile downloader	0.000083418%
Commercial spyware	0.0000390038%
DOS	0.0000020137%
SMS fraud	0.0000016394%
Rooting	0.0000009107%
Spam	0.0000005276%
Windows malware	0.000000047%
Call fraud	0.000000007%
Ransomware	0.0000000002%

Mobile Threats in 2022



Source: Kaspersky

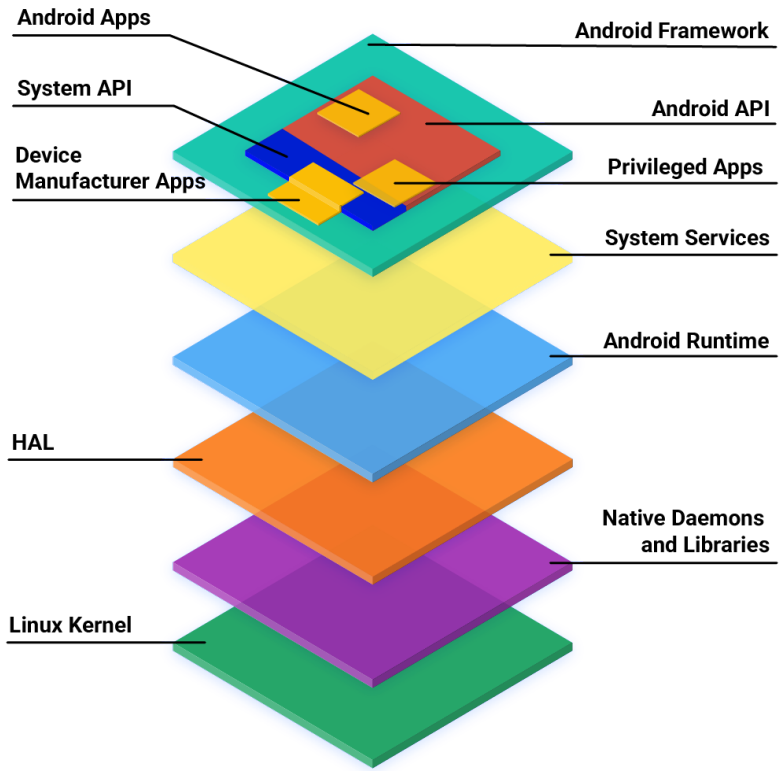
Android Architecture



Source: https://embeddedbits.org/what-differs-android-from-other-linux-based-systems/

Android Architecture

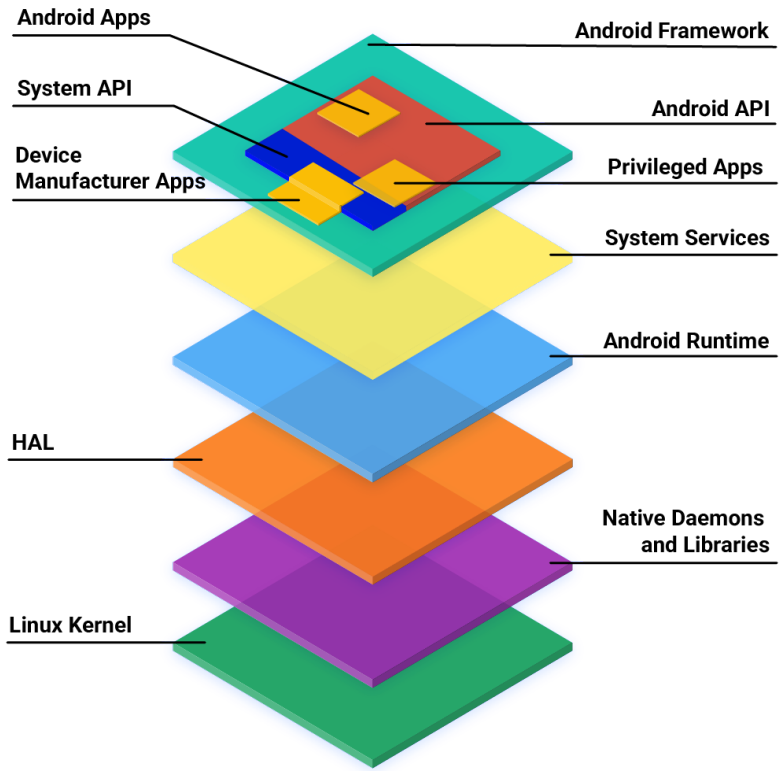
- Linux kernel
  - Androidisms
  - Security
  - Device drivers
- Hardware Abstraction Layer (HAL)
  - Standard interfaces
  - Multiple library modules



Source: `https://source.android.com/docs/core/architecture`

Android Architecture

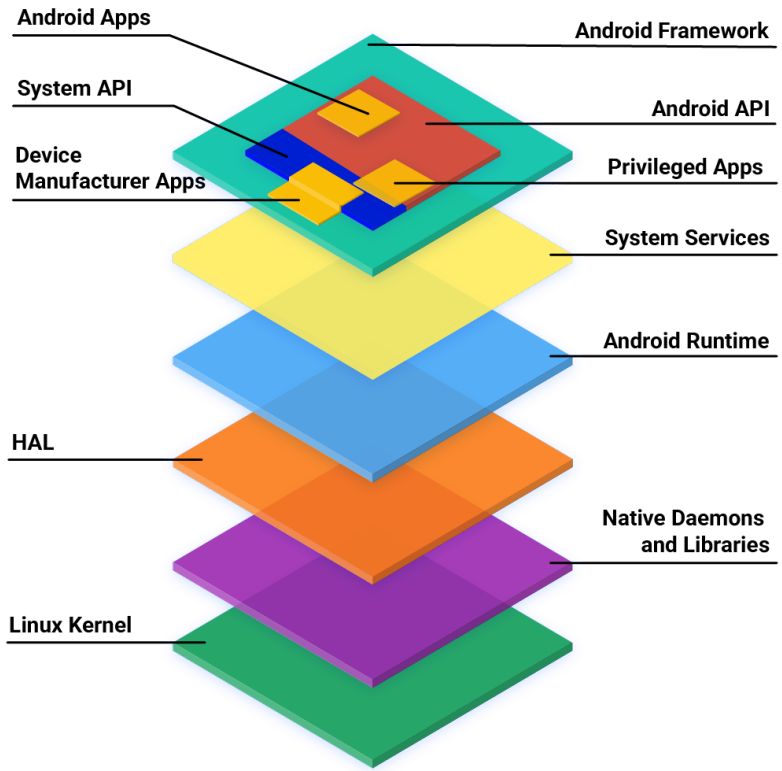
- Native userspace
  - init process
    - Starts `installd`, `adbd`, `servicemanager`, `Zygote`
  - Native daemons
  - Native libraries
    - Through Java framework APIs
    - Through Android NDK



Source: `https://source.android.com/docs/core/architecture`

Android Architecture

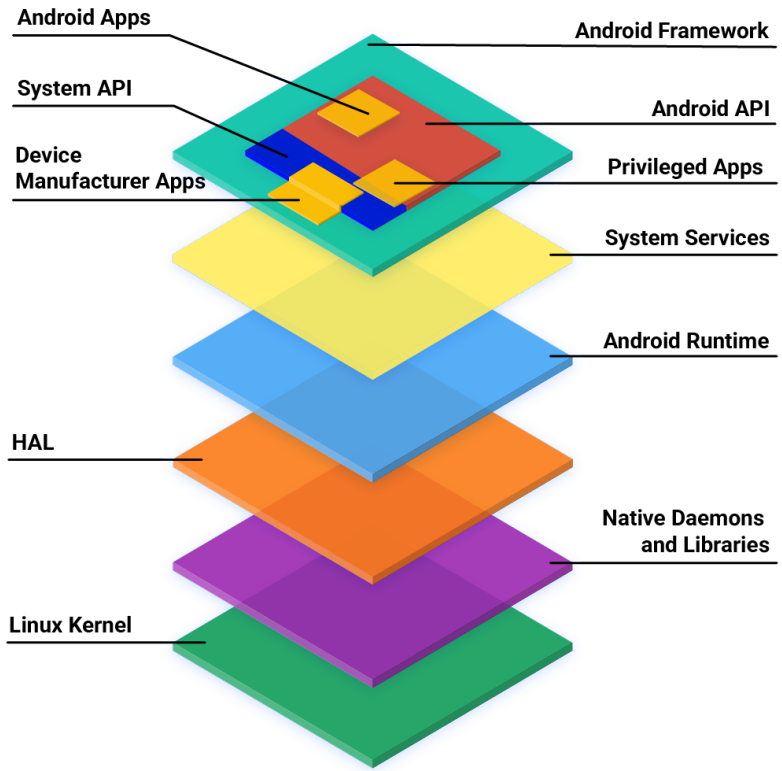
- Android Runtime
  - Dalvik
  - ART
  - Ahead-Of-Time (AOT) compilation
- Java Runtime libraries
  - `java.*` and `javax.*`
  - Apache Harmony Project
  - Java Native Interface (JNI)



Source: `https://source.android.com/docs/core/architecture`

Android Architecture

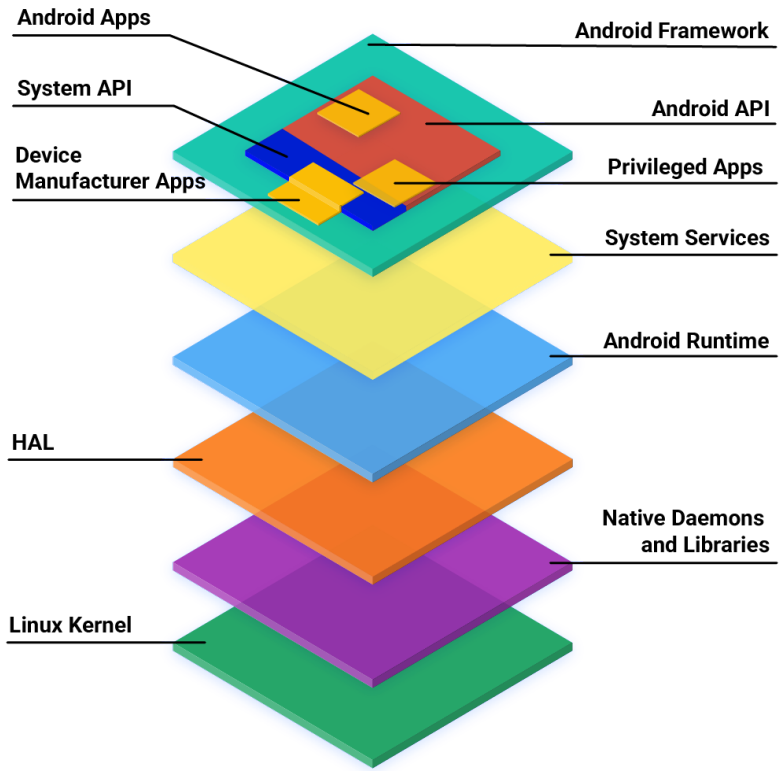
- System services
  - Fundamental features of Android
  - Location, touch screen, telephony, networking
  - Native and Java code
  - Service interface
  - Through the Binder



Source: `https://source.android.com/docs/core/architecture`

Android Architecture

- Android API
  - Base components for app development
  - Interaction with the hardware
  - Interaction with high level services

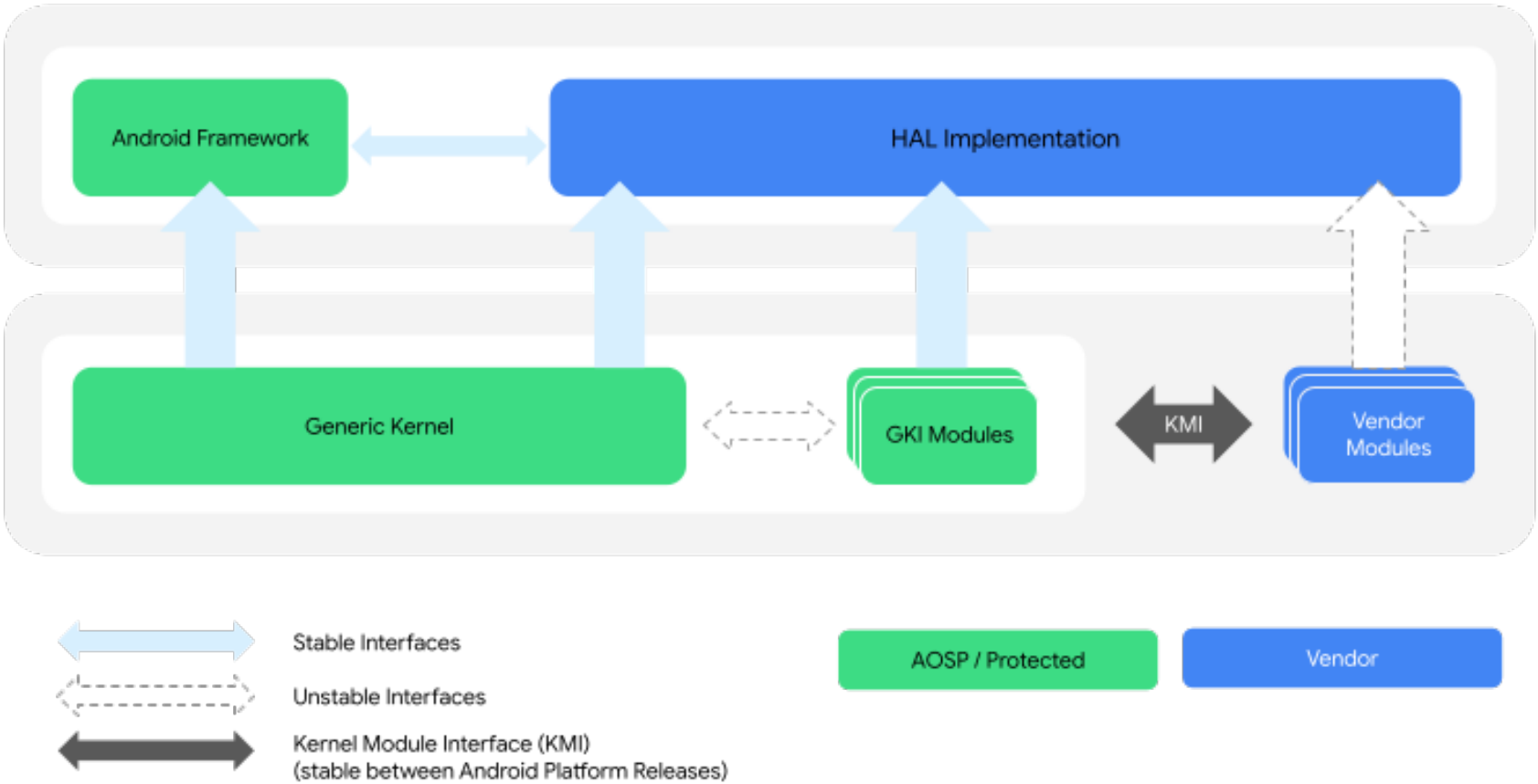


Source: `https://source.android.com/docs/core/architecture`

Linux Kernel

- Based on upstream Linux Long Term Supported (LTS) kernel
- LTS + Android-specific patches
- Generic Kernel Image (GKI)
  - separation of hardware-agnostic core kernel and hardware-specific GKI modules
  - interacts with hardware-specific vendor modules through Kernel Module Interface (KMI)

Linux Kernel



Source: `https://source.android.com/docs/core/architecture/kernel`

Android Runtime - Dalvik VM

- Default until Android 5.0
- Runs Dalvik-specific byte-code
- Dalvik Executable Format (DEX)
  - Runs .dex files instead of .jar files
  - .dex is 50% smaller than corresponding .jar
- Just-In-Time compilation
  - From Android 2.2
  - Short segments of bytecode translated into native machine code at runtime
  - Improves performance

Android Runtime - ART

- From Android 5.0
- More advanced runtime architecture
- Ahead-Of-Time compilation
  - Just once, at installation
  - Entire DEX file -> executable for target device
  - Instead of JIT compilation and Dalvik interpretation
  - More efficient, reduced power consumption
  - More space to store the executables
- Improved memory allocation, GC, debugging and profiling

Native Libraries

- bioniC (libc)
  - Much smaller and faster than glibc
- SQLite
  - Managing SQL databases
- OpenGL ES
  - Standard software interface for 3D processing hardware
- SSL
  - Securing the communication over Internet

Android Framework

- System Services and Managers
  - Telephony
  - Location
  - Activity
  - Package
  - Notification
- System Content Providers
  - Calendar
  - Dictionary
  - Contacts
  - Settings

Outline

Application components

- Activities
  - GUI
  - Back stack
- Services
  - Background operations
  - Runs in the same process by default
  - Tasks for current or other apps

Application components

- Broadcast Receivers
  - Receiving broadcast messages
  - Announcements, notifications
- Content Providers
  - Storing, sharing data
  - SQLite, files

Intents

- Similar to signals
- Delivered by the OS
- Start activities
- Start services
- Sending broadcast messages

Binder

- Lightweight RPC
- Remote object invocation
- Communication with system and app services
- Transmit parcels of data
- Synchronous calls (blocking)

Outline

Sandboxing

- Linux kernel security
  - Isolate user resources (file permissions)
  - Process runs with user's UID/GID
- In Android UIDs are used to identify applications
  - Isolate applications
  - Basis for sandboxing

Sandboxing

- Unique UID assigned to each application at installation time
- Dedicated process running as that UID
- Dedicated directory - only that UID has read/write/execute permissions
- Process-level and file-level sandbox
- Kernel level sandbox - all applications

UIDs

- Well-defined UIDs for system apps and daemons
- Very few daemons under root UID 0
- UIDs for system services start at 1000
- User *system* has UID 1000
  - Special privileges
- App UIDs start at 10000

Application Files Permissions

- Each app - dedicated data directory
- Database, images, other files
- rwx permissions only for that app UID/GID

Application Permissions

- Operations outside sandbox
- Declared statically in the Manifest file
- Before Android 6
  - Granted at installation time
  - Cannot be revoked
- From Android 6
  - Granted at runtime
  - Revoked and granted from settings

Enforce Permissions

- Permission enforcement
- Access to lower-level resources
  - Enforced by the Linux kernel
  - Check UID/GID vs resource's owner
- Access to high-level Android components
  - Enforced by Android OS or a certain component

Code Signing Key

- All apps signed by their developer
- Methods:
  - Personal app signing key
  - Upload key
- Same origin policy
  - App updates from the same developer

Outline

Bibliography

- https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-200901-202301
- https://gs.statcounter.com/os-market-share#monthly-200901-202301
- https://developers.google.com/android/play-protect/potentially-harmful-applications
- https://securelist.com/it-threat-evolution-in-q3-2022-mobile-statistics/107978/
- https://transparencyreport.google.com/android-security/store-app-safety?hl=en
- https://medium.com/@khetanrajesh/android-boot-up-process-init-c05371c4f976

Bibliography

- https://developer.android.com/guide/platform
- https://source.android.com/docs/core/architecture
- https://source.android.com/docs/core/architecture/kernel
- https://source.android.com/docs/core/runtime

Bibliography

- https://developer.android.com/guide/components/activities/intro-activities
- https://developer.android.com/guide/components/services
- https://developer.android.com/guide/components/broadcasts
- https://developer.android.com/guide/topics/providers/content-provider-basics
- https://developer.android.com/guide/components/intents-filters

Bibliography

- Karim Yaghmour, Embedded Android: Porting, Extending, and Customizing, Chapter 2
- Android Security Internals, Nicolay Elenkov, 2015
- Android Hacker's Handbook, Joshua J. Drake, 2014
- Introducere in sistemul de operare Android - Laura Ruse, Vlad Traistă-Popescu, 2021
- Securitatea sistemului de operare Android - Laura Ruse, Vlad Traistă-Popescu, 2021

Keywords

- PHA
- Mobile malware
- Linux kernel
- Android Runtime
- ART
- Native libraries
- Application framework
- Activities
- Services
- Broadcast receivers
- Content providers
- Intents
- Binder
- Sandboxing
- Permissions
- Code signing