POLITÉCNICA UNIVERSIDAD POLITÉCNICA DE MADRID

Grupo de Investigación SeCLab

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

Universidad Politécnica de Madrid ETSI SISTEMAS INFORMÁTICOS

Challenges and Opportunities in Mobile Security

Jorge Blasco Alis – jorge.blasco.alis@upm.es

24.06.2024 - 05.07.2024
CAEN - FRANCE

About me

- Phd in Computer Science - 2012
    - Universidad Carlos III de Madrid
- Moved to the UK
    - City, University of London
    - Royal Holloway, University of London
- Came back to Spain in 2022
    - Universidad Politécnica de Madrid
    - Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE
SISTEMAS INFORMÁTICOS
UNIVERSIDAD POLITÉCNICA DE MADRID

Outline

- **Mobile Security**
- **Challenges and opportunities**
  - **How to analyse apps**
    - **Static vs Dynamic analysis**
    - **Identification of dangerous behaviours**
  - **Analysing apps at scale**
    - **Privacy Leaks**
    - **Flaws in BLE**
    - **App Collusion**

Outline

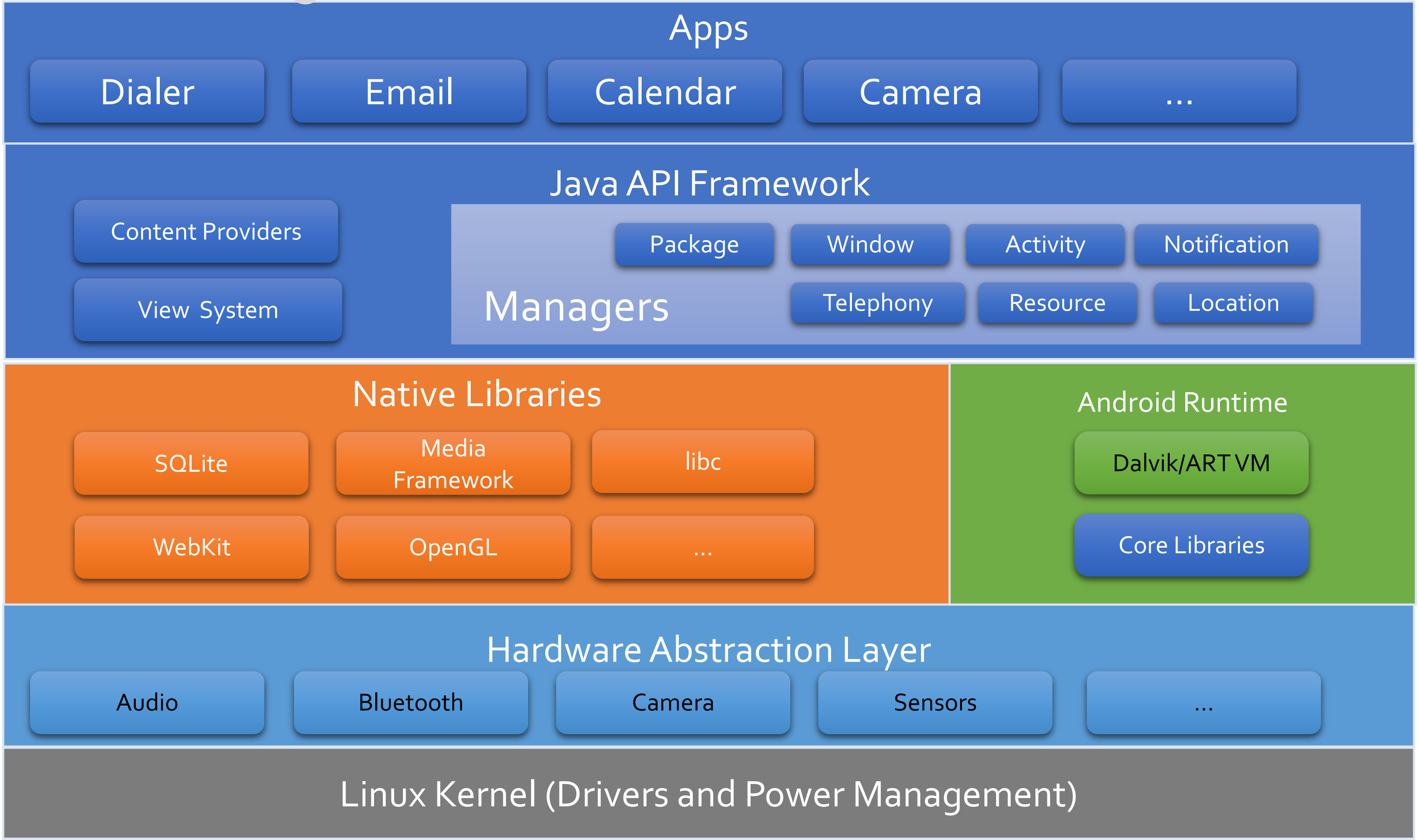
- **Mobile Security**
- Challenges and opportunities
  - How to analyse apps
    - Static vs Dynamic analysis
    - Identification of dangerous behaviours
  - Analysing apps at scale
    - Privacy Leaks
    - Flaws in BLE
    - App Collusion

Smartphone OS basics

Smartphone OS basics

- **Both are UNIX based**
  - **Android Linux**
  - **iOS is based in Darwin**
- **App based**
- **Power constrained devices**

Android System Architecture



Apps

Dialer | Email | Calendar | Camera | ...

Java API Framework

Content Providers

View System

Managers

Package | Window | Activity | Notification

Telephony | Resource | Location

Native Libraries

SQLite | Media Framework | libc

WebKit | OpenGL | ...

Android Runtime

Dalvik/ART VM

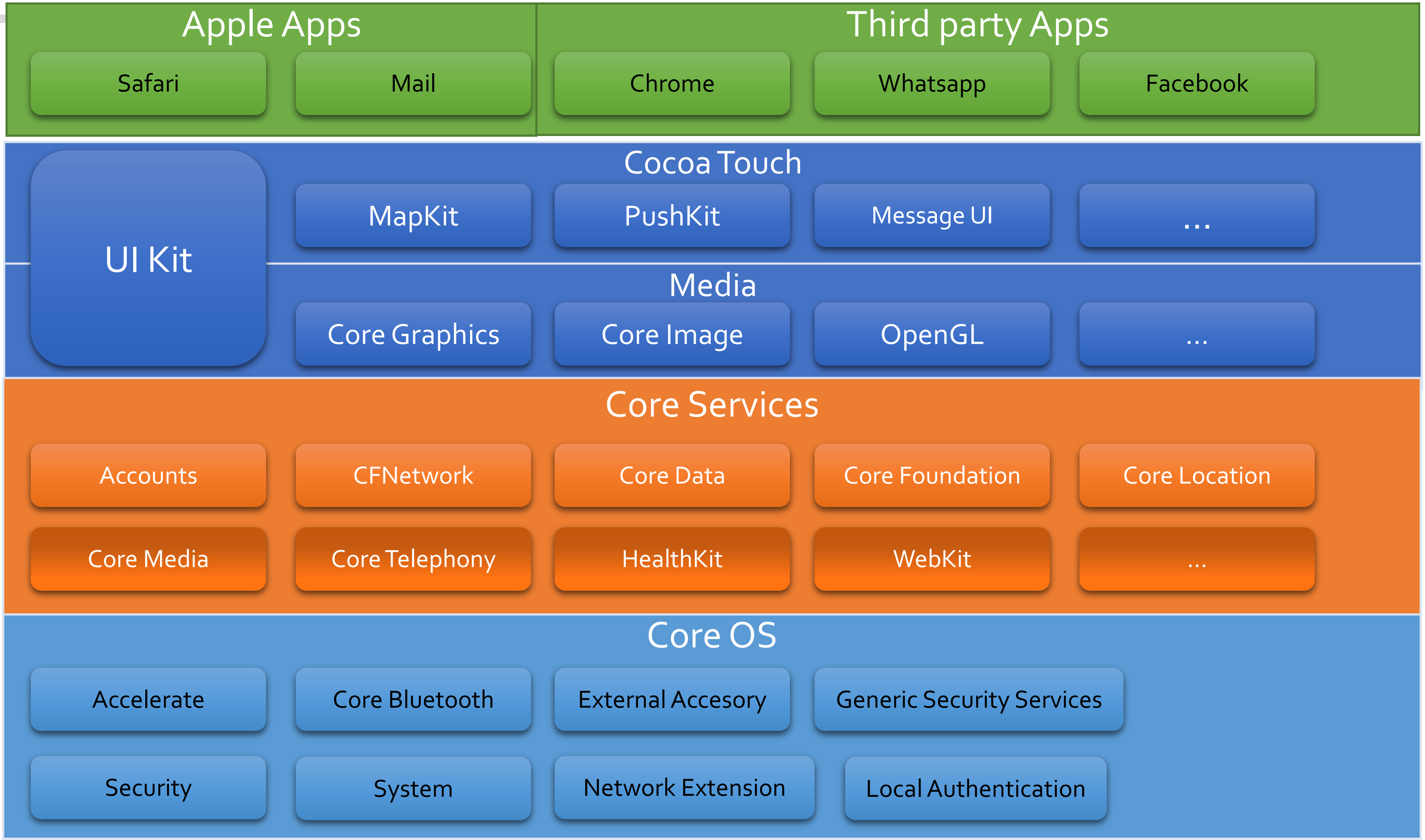
Core Libraries

Hardware Abstraction Layer

Audio | Bluetooth | Camera | Sensors | ...

Linux Kernel (Drivers and Power Management)

iOS System Architecture



Apple Apps
Safari
Mail

Third party Apps
Chrome
Whatsapp
Facebook

Cocoa Touch
UI Kit
MapKit
PushKit
Message UI
...

Media
Core Graphics
Core Image
OpenGL
...

Core Services
Accounts
CFNetwork
Core Data
Core Foundation
Core Location
Core Media
Core Telephony
HealthKit
WebKit
...

Core OS
Accelerate
Core Bluetooth
External Accesory
Generic Security Services
Security
System
Network Extension
Local Authentication

Mobile Security

Threat Model

- Physical Threats
  - Thieves
  - Modifications
- Software Threats
  - Apps
  - External Exploits
- Network Threats
  - Eavesdropping
  - Integrity

Security Requirements

- **Traditional Workstations**
  - **User Authentication**
  - **Most actions allowed**
  - **Network restrictions**
- **Mobile**
  - **User Authentication**
  - **Trusted OS**
  - **App isolation**
  - **Network restrictions**
  - ....

Security Solutions

- **Market-Level**
  - **App review**
  - **App signing**

- **System-Level**
  - **Access Control**
  - **Sandboxing**
  - **Permissions**
  - **Full-disk encryption**
  - **....**

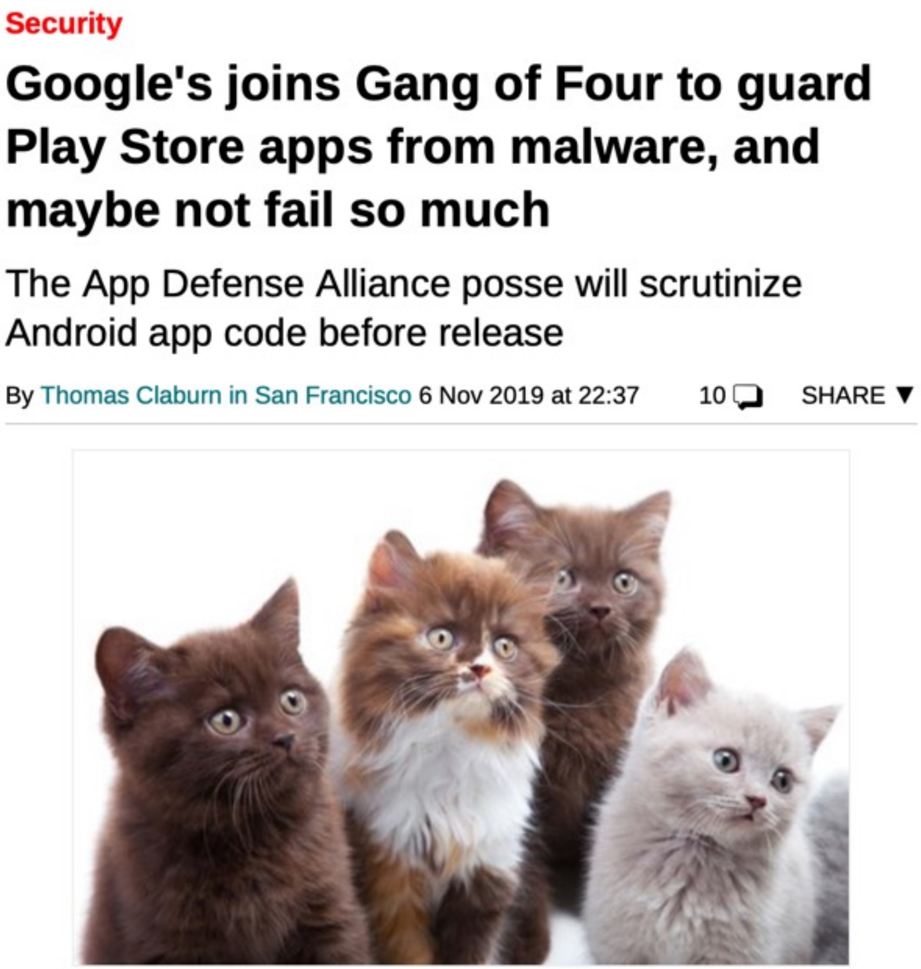
Market-Level

App Review

- **Apps distributed via Official markets are reviewed**
  - **Android**
    - **Security issues (Automatic analysis - Very fast)**
  - **iOS**
    - **Security issues (automatic and manual review – slow)**
    - **Apple design guidelines (manual review – slow)**

Google Security Review Today

- Zimperium
- ESET
- Lookout
- Google



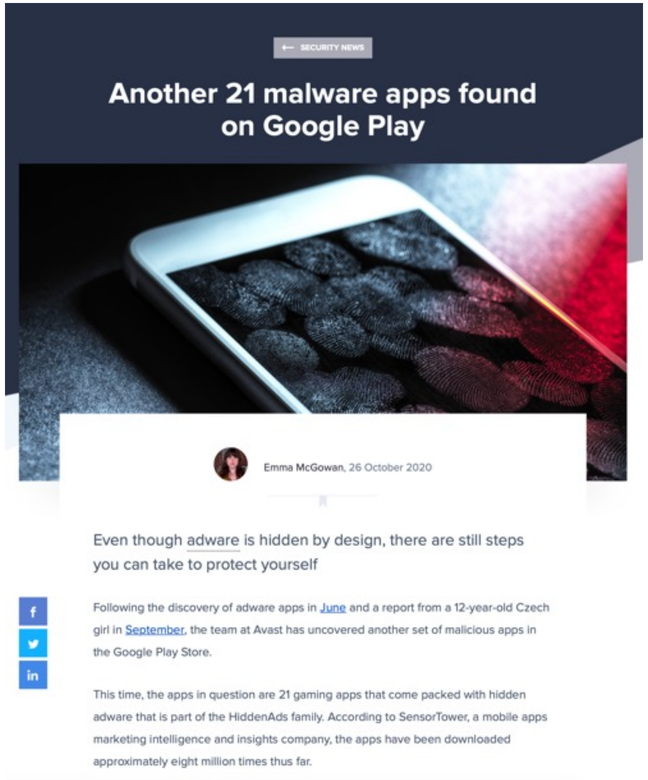
Security

Google's joins Gang of Four to guard Play Store apps from malware, and maybe not fail so much

The App Defense Alliance posse will scrutinize Android app code before release

By Thomas Claburn in San Francisco 6 Nov 2019 at 22:37 10 SHARE ▼

Problem solved, right?



Another 21 malware apps found on Google Play

Emma McGowan, 26 October 2020

Even though adware is hidden by design, there are still steps you can take to protect yourself

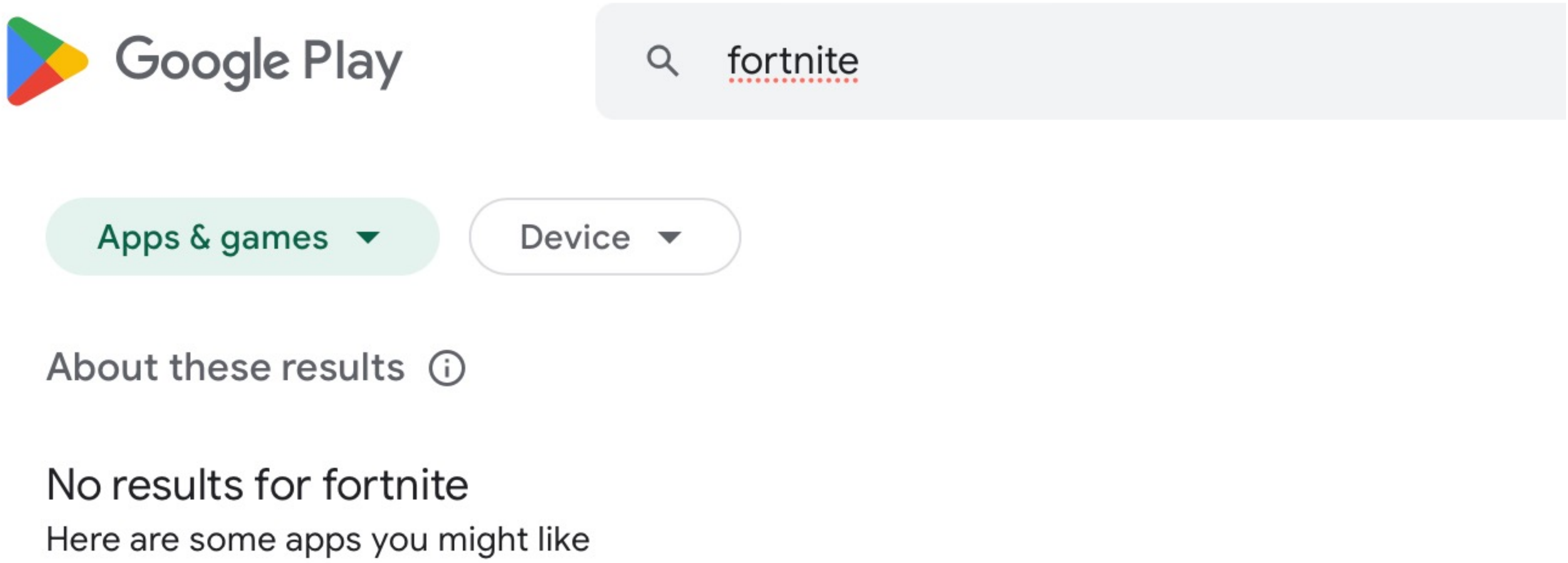
Following the discovery of adware apps in June and a report from a 12-year-old Czech girl in September, the team at Avast has uncovered another set of malicious apps in the Google Play Store.

This time, the apps in question are 21 gaming apps that come packed with hidden adware that is part of the HiddenAds family. According to SensorTower, a mobile apps marketing intelligence and insights company, the apps have been downloaded approximately eight million times thus far.

Source: Avast 2020

Does this always happen?

Fortnite in Google Play

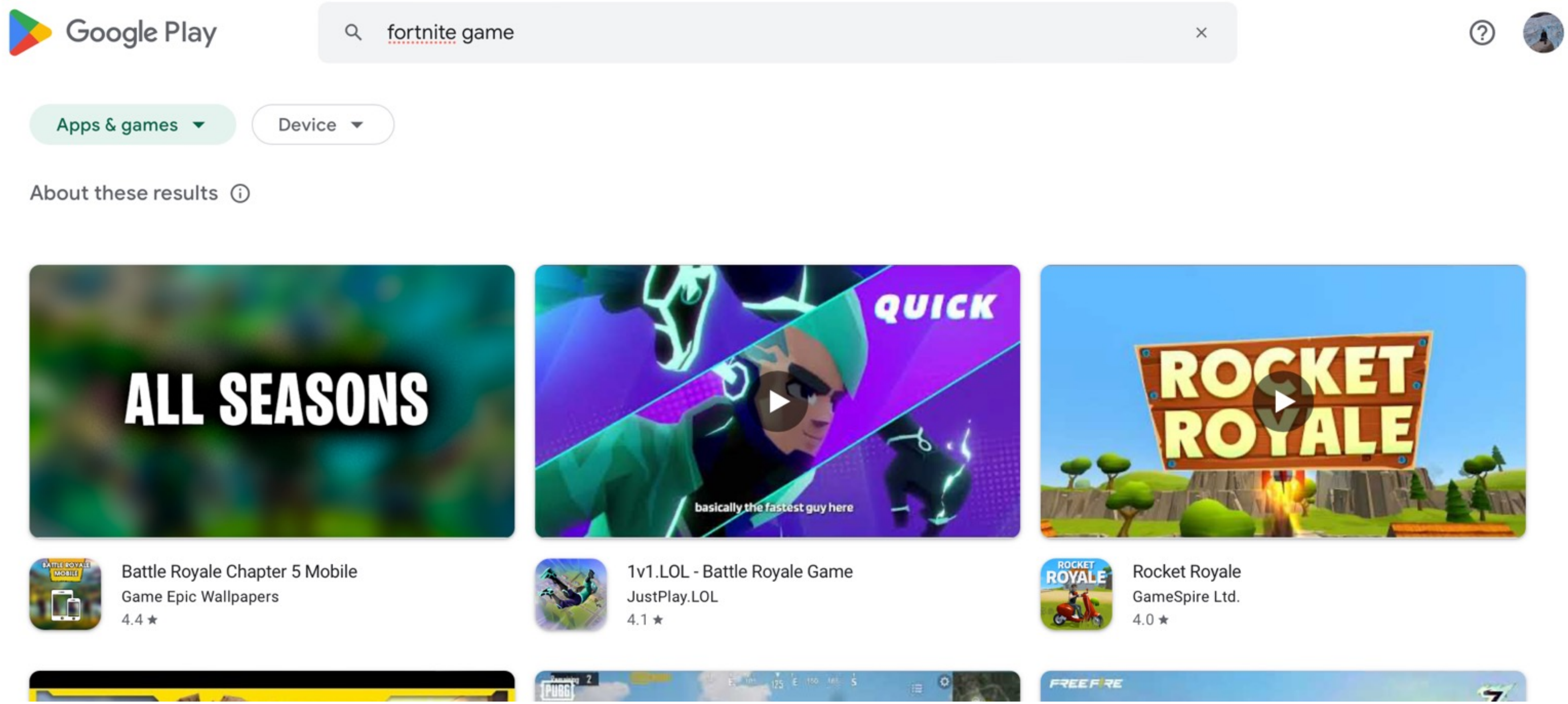


About these results

No results for fortnite

Here are some apps you might like

Fortnite in Google Play



Google Play

fortnite game

Apps & games ▾ Device ▾

About these results ⓘ

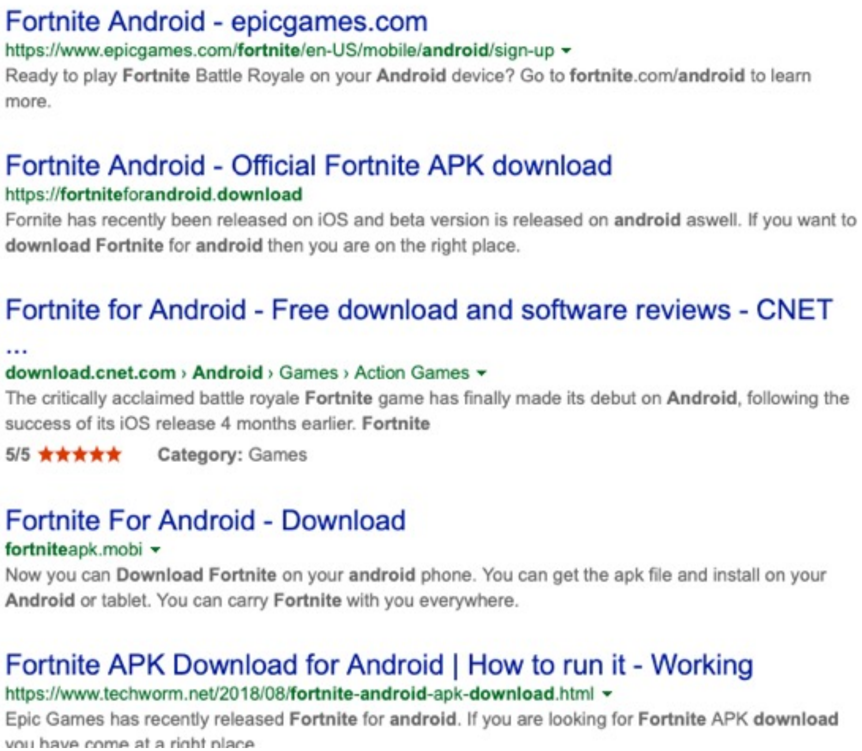
Battle Royale Chapter 5 Mobile
Game Epic Wallpapers
4.4 ★

1v1.LOL - Battle Royale Game
JustPlay.LOL
4.1 ★

Rocket Royale
GameSpire Ltd.
4.0 ★

Trying to download Fortnite for Android

- **2019**



Fortnite Android - epicgames.com
https://www.epicgames.com/**fortnite**/en-US/mobile/**android**/sign-up ▾
Ready to play **Fortnite** Battle Royale on your **Android** device? Go to **fortnite**.com/**android** to learn more.

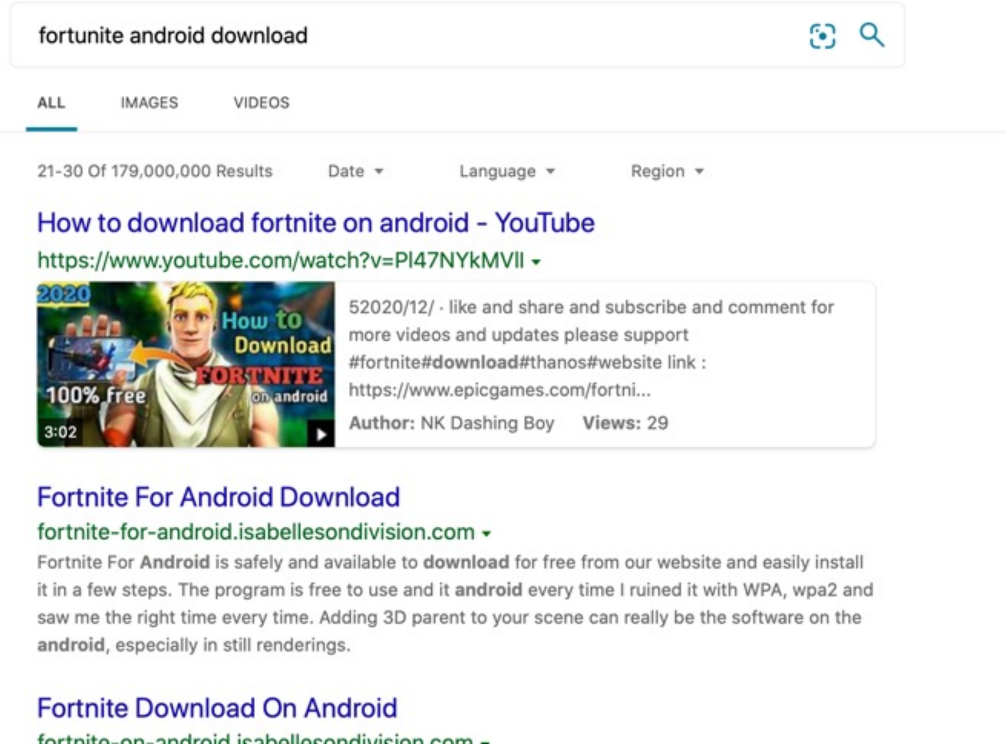
Fortnite Android - Official Fortnite APK download
https://**fortniteforandroid.download**
Fornite has recently been released on iOS and beta version is released on **android** aswell. If you want to **download Fortnite** for **android** then you are on the right place.

Fortnite for Android - Free download and software reviews - CNET ...
download.cnet.com › **Android** › Games › Action Games ▾
The critically acclaimed battle royale **Fortnite** game has finally made its debut on **Android**, following the success of its iOS release 4 months earlier. **Fortnite**
5/5 ★★★★★ **Category:** Games

Fortnite For Android - Download
fortniteapk.mobi ▾
Now you can **Download Fortnite** on your **android** phone. You can get the apk file and install on your **Android** or tablet. You can carry **Fortnite** with you everywhere.

Fortnite APK Download for Android | How to run it - Working
https://www.techworm.net/2018/08/**fortnite-android**-apk-**download**.html ▾
Epic Games has recently released **Fortnite** for **android**. If you are looking for **Fortnite** APK **download** you have come at a right place.

- **2020**



fortunite android download

ALL IMAGES VIDEOS

21-30 Of 179,000,000 Results Date ▾ Language ▾ Region ▾

How to download fortnite on android - YouTube
https://www.youtube.com/watch?v=Pl47NYkMVIl ▾

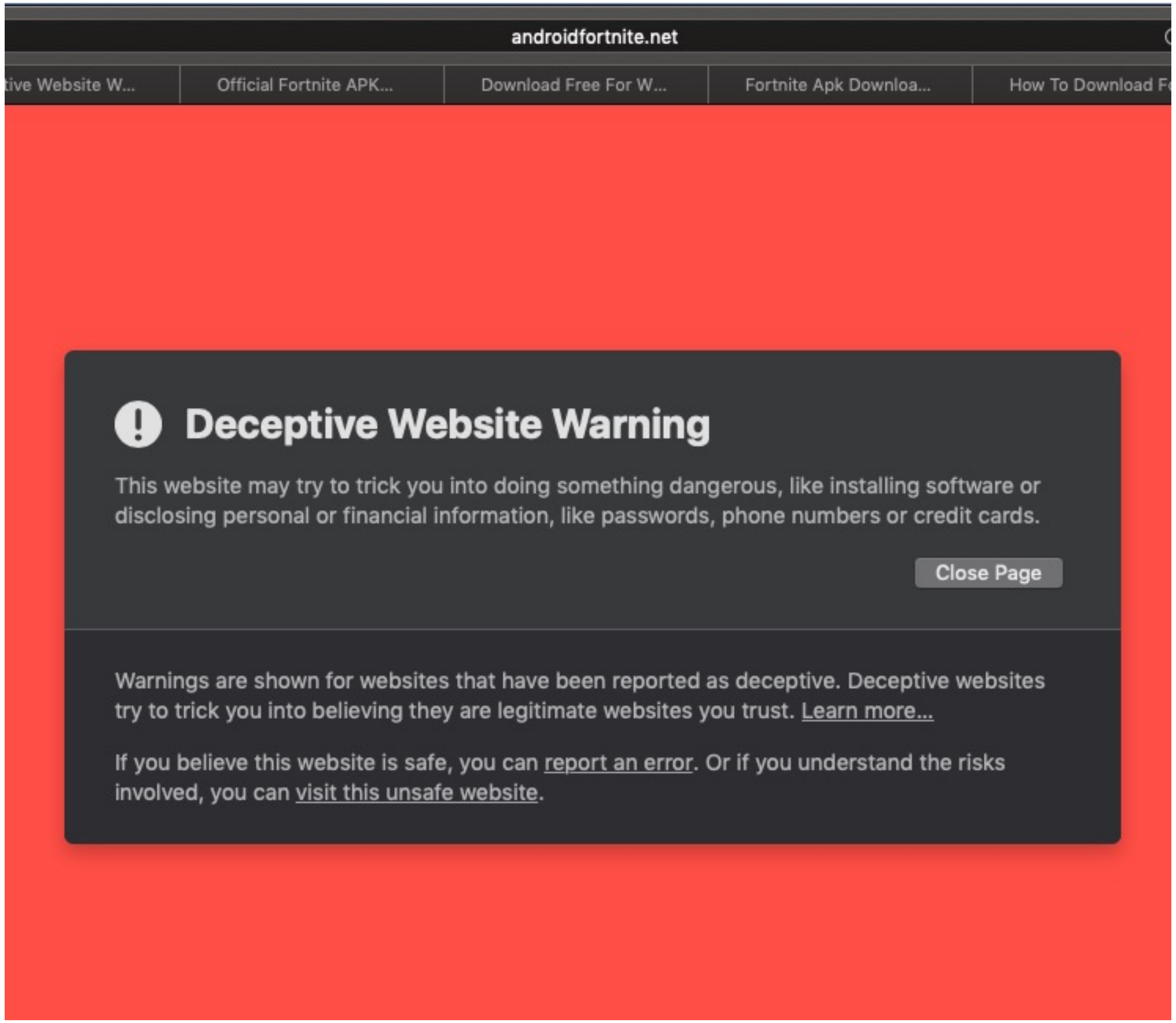
52020/12/ · like and share and subscribe and comment for more videos and updates please support #fortnite**#download**#thanos#website link : https://www.epicgames.com/fortni...
Author: NK Dashing Boy **Views:** 29

Fortnite For Android Download
fortnite-for-android.isabellesondivision.com ▾
Fortnite For **Android** is safely and available to **download** for free from our website and easily install it in a few steps. The program is free to use and it **android** every time I ruined it with WPA, wpa2 and saw me the right time every time. Adding 3D parent to your scene can really be the software on the **android**, especially in still renderings.

Fortnite Download On Android
fortnite-on-android.isabellesondivision.com ▾

Which one is the right one?

- https://fortniteforandroid.download
- https://extensinet.com/download-fortnite-android/
- https://fortniteforandroid.download
- https://www.epicgames.com/fortnite/android
- http://www.fortniteformobile.com



Deceptive Website Warning

This website may try to trick you into doing something dangerous, like installing software or disclosing personal or financial information, like passwords, phone numbers or credit cards.

Warnings are shown for websites that have been reported as deceptive. Deceptive websites try to trick you into believing they are legitimate websites you trust. Learn more...

If you believe this website is safe, you can report an error. Or if you understand the risks involved, you can visit this unsafe website.

App Signing

- **Both OS require apps to be signed to execute**
  - Android
    - Self-signed certificate
    - Identify developer and app updates
  - iOS
    - Certificate provided by Apple
    - Only apps signed with valid certificate go into the App Store
    - Organisations can bypass this (with restrictions)

Why?

System-Level

System-level

- Secure Boot
- Access Control
- Sandboxing and Permissions
- Network Security
- File-Based and Full-Disk Encryption
- Other features

Secure Boot

Smartphone Processors

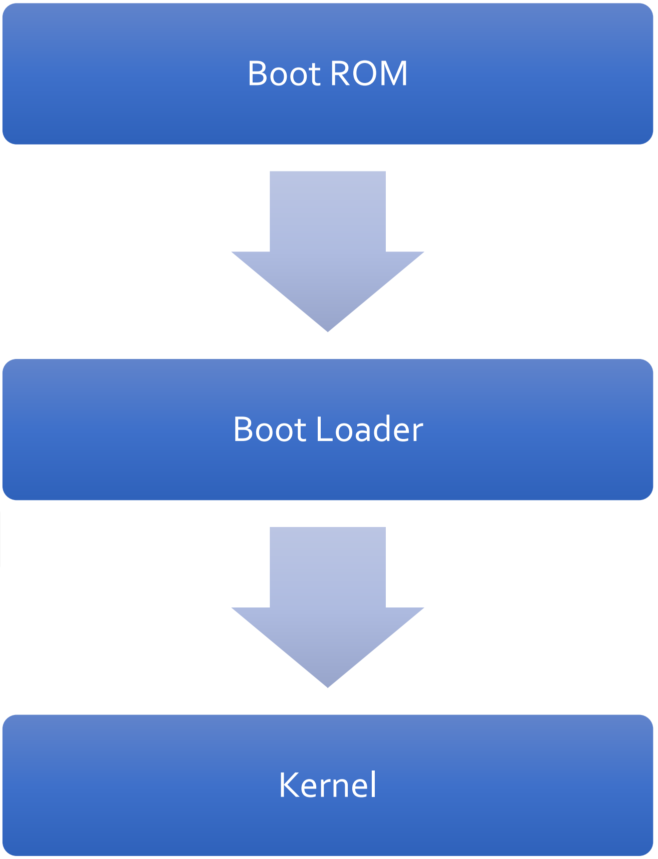
- **General/Application Processor**
  - **Executes apps and most of the OS**
- **Baseband**
  - **Manages wireless functionality (cellular)**
- **Secure Enclave or Trusted Execution Environment**
  - **Executes highly sensitive cryptographic operations**

Secure Boot

- **Ensures integrity and authenticity of OS (trusted source)**
  - **Also used for Baseband and Secure Enclaves (other processors)**
- **Root of trust comes from a hardware-protected source**
- **Starts the moment the device is turned on**
- **Most manufacturers implement similar approaches**
- **Each steps checks the integrity of the next phase**
- **If check fails device enters recovery mode**

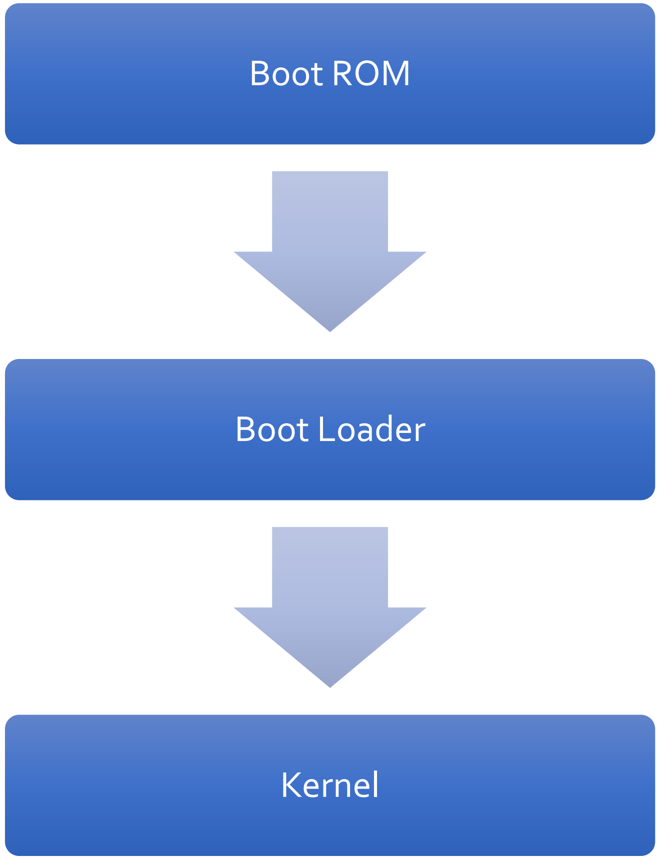
Boot ROM

- First code to be executed
- Read-only Tamper-Proof
- Implicitly trusted
- Includes root CA
- Checks next code has been signed



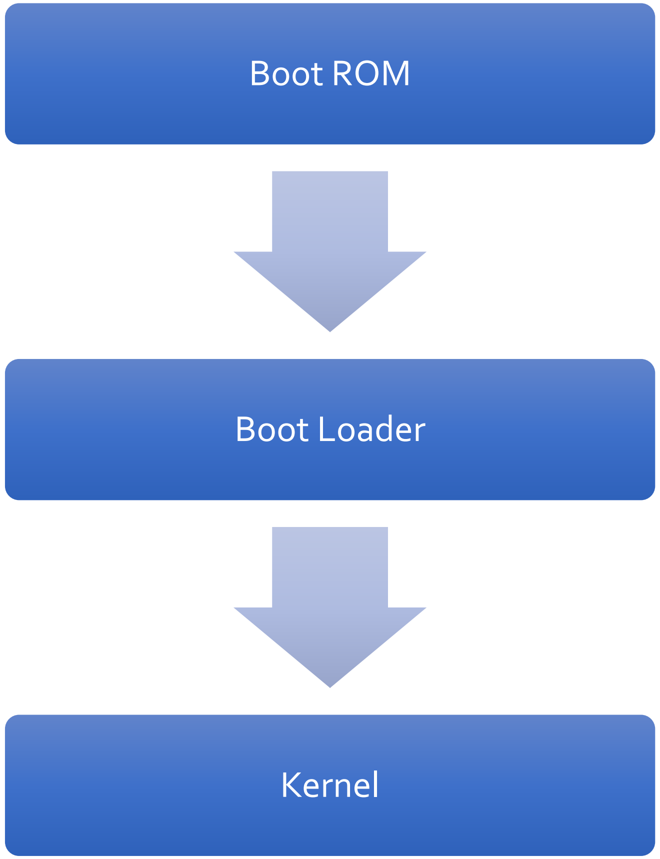
Boot Loader

- **Verifies integrity of Kernel**
- **Low-level initialization**
- **Loads firmware**
  - **Processors**
- **Loads Kernel**
- **In Android can be unlocked**
  - **Requires wipe**



Kernel

- Heart of the OS
- Enforces most of the rest of security features
  - Code signing
  - Sandboxing
  - Address Space Layout Randomization (ASLR)



Access Control

Physical Access Control

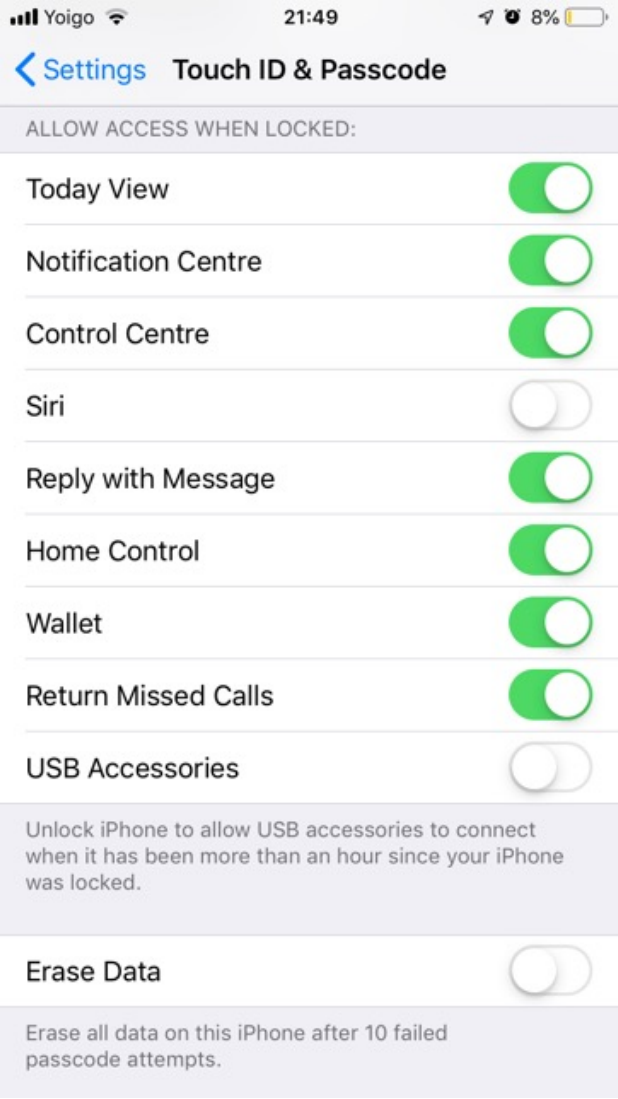
- Screen Lock avoids devices being used by unauthorised parties
  - PIN/Pass code
  - Biometrics
    - Also requires PIN/Pass Code
- Can be configured to wipe device
- Also used for file encryption

Can this be bypassed?

- **Aviv, Adam J., et al. "Practicality of accelerometer side channels on smartphones." *Proceedings of the 28th Annual Computer Security Applications Conference*. 2012.**
- **Zarandy, Almos, Ilia Shumailov, and Ross Anderson. "Hey Alexa what did I just type? Decoding smartphone sounds with a voice assistant." *arXiv preprint arXiv:2012.00687* (2020).**

iOS Specifics

- **Siri**
  - **Can be exploited by attackers**
    - **Phishing to remove Activation lock**
- **USB Accessories**
  - **Used by forensic tools to enable forensic image acquisition**



Touch ID & Passcode

ALLOW ACCESS WHEN LOCKED:

Today View	●
Notification Centre	●
Control Centre	●
Siri	○
Reply with Message	●
Home Control	●
Wallet	●
Return Missed Calls	●
USB Accessories	○

Unlock iPhone to allow USB accessories to connect when it has been more than an hour since your iPhone was locked.

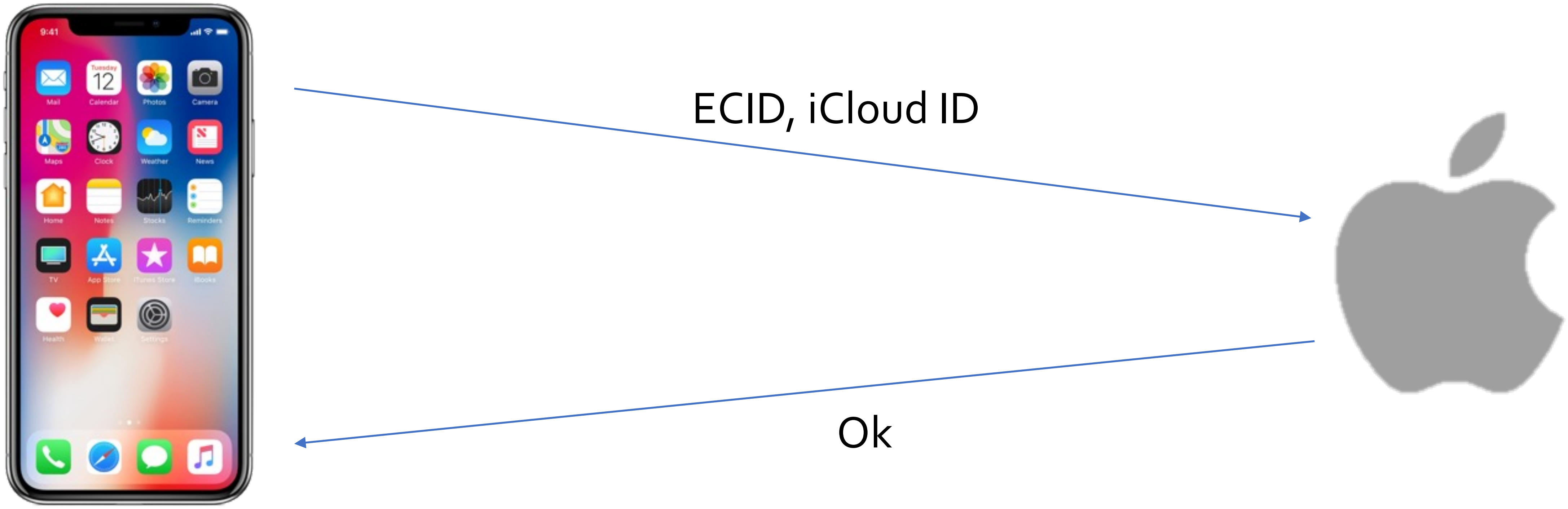
Erase Data ○

Erase all data on this iPhone after 10 failed passcode attempts.

Activation Lock

- **All Apple devices need to be activated by Apple**
  - **On first boot**
  - **Or after a reset if the phone was wiped to avoid pass code**
- **If a device has been registered with Find My, Apple will require the account credentials to activate it**

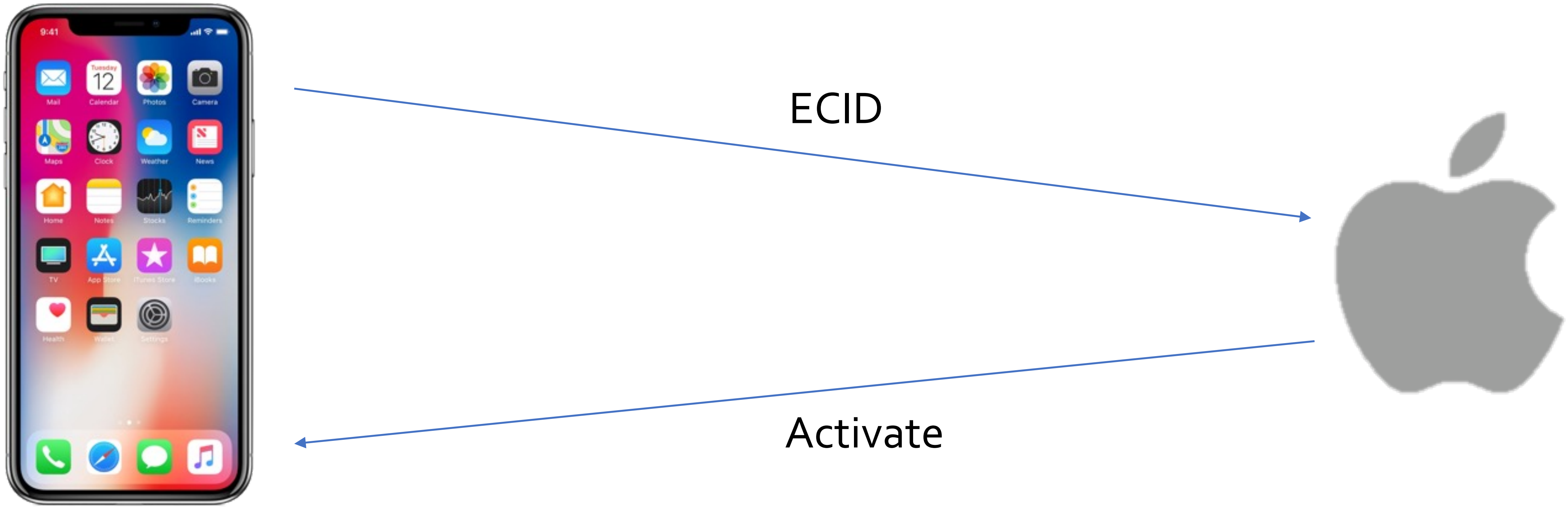
Activation Lock I – Sign up



ECID, iCloud ID

Ok

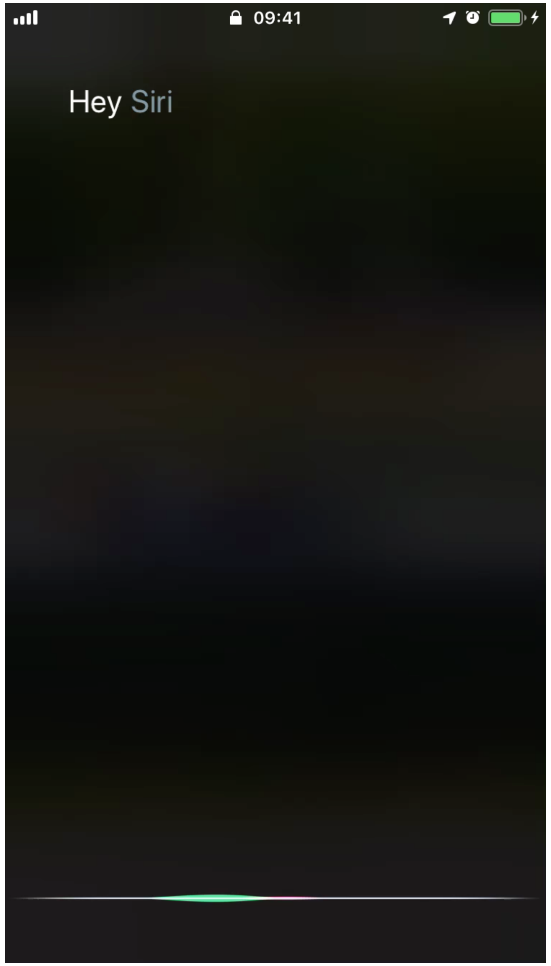
Activation Lock II - Check



ECID

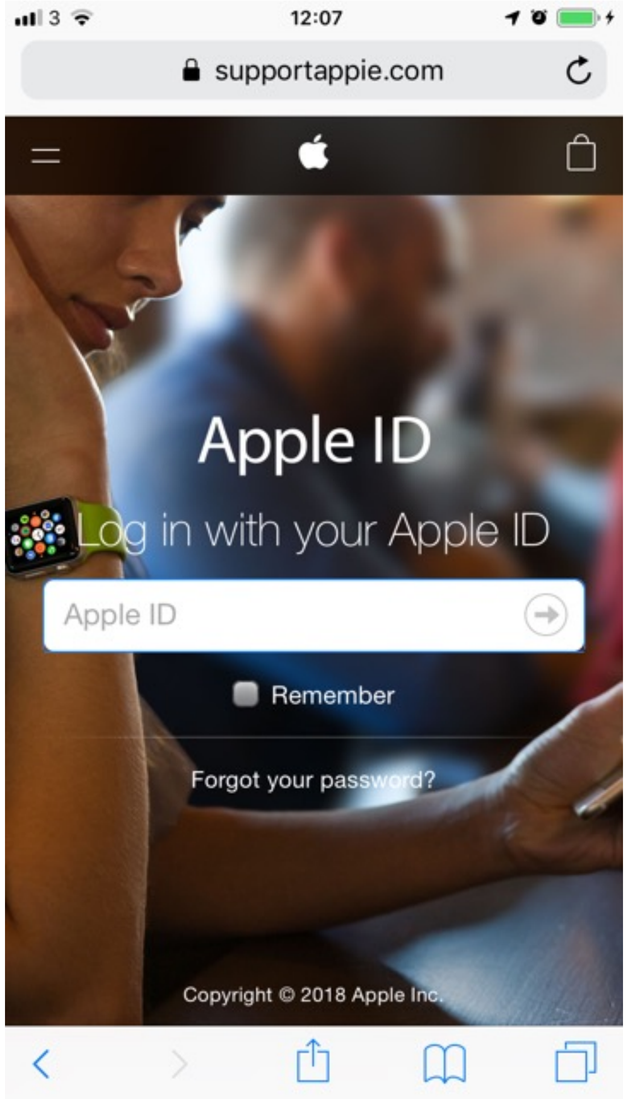
Activate

Unlocking an iPhone



Hey Siri

Your iPhone 7 Plus has been located at 9:58 AM. Check it's location : https://login-icloud.com/?e=eVK6 iSupport.



supportappie.com

Apple ID

Log in with your Apple ID

Apple ID

□ Remember

Forgot your password?

Copyright © 2018 Apple Inc.

Android Specifics

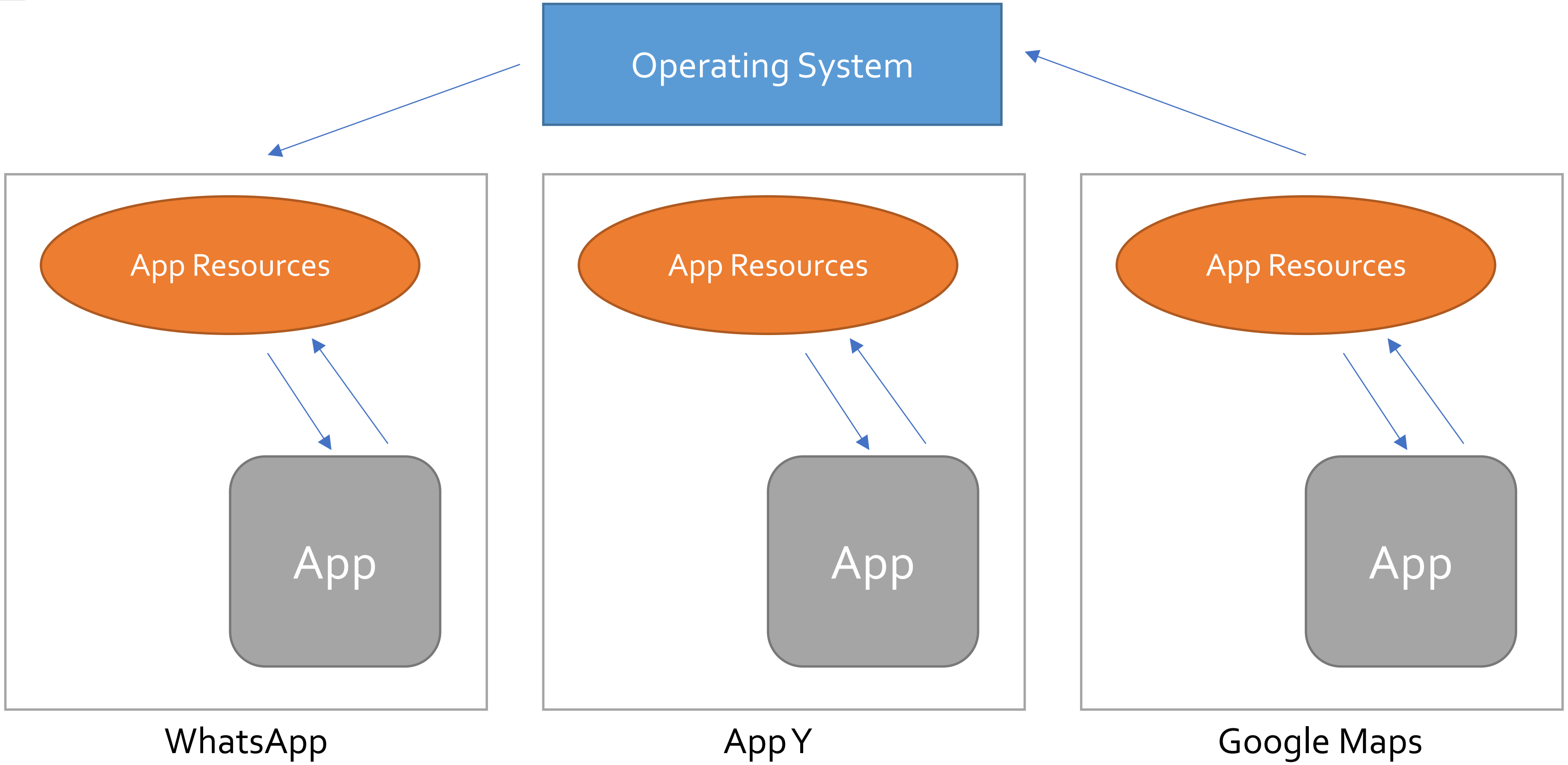
- Adds
  - Lock pattern
  - Voice recognition and others
    - Not very secure – Similar to Siri
- SD card not encrypted by default
- Boot Loader unlock allows bypass
  - Requires wipe

Sandboxing and Permissions

Sandbox

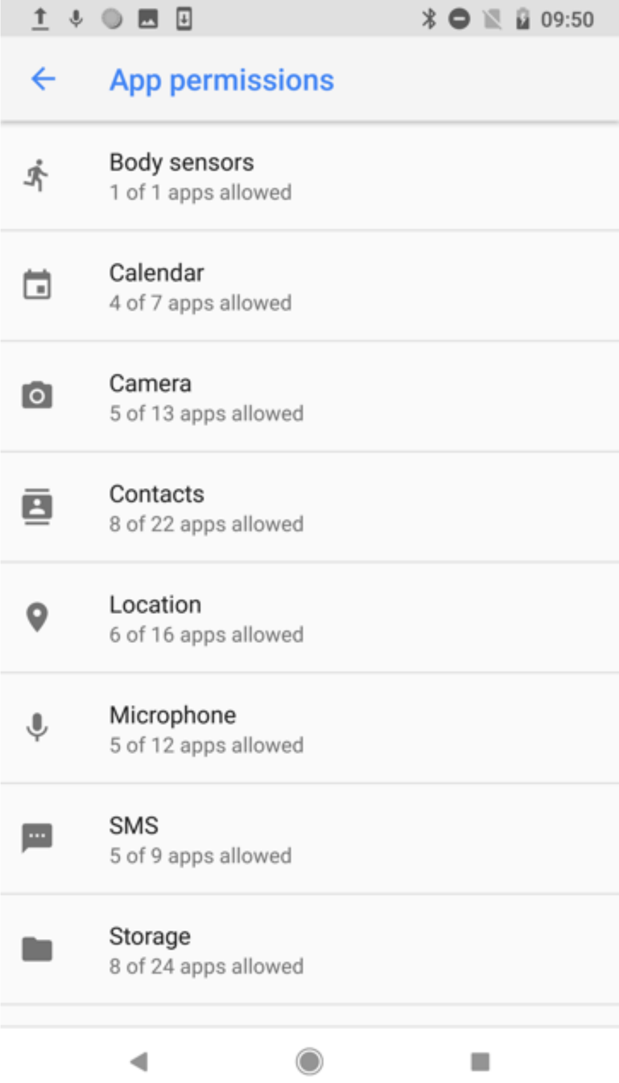
- All apps execute under the minimum privilege policy
- This means
  - Only access to their own directory
  - OS mediates access to all other resources
    - Other app resources (e. g. share via Whatsapp)
    - System resources (e.g. contacts or camera)

Sandboxing – Inter Process Communication



Android Specifics

- Each app executed as different user
- Sandbox implemented via SELinux
  - Adds domains
- Permissions declared at install-time
- Dangerous ones requested on run-time
- Include usage of SMS and Phone



09:50

App permissions

Body sensors
1 of 1 apps allowed

Calendar
4 of 7 apps allowed

Camera
5 of 13 apps allowed

Contacts
8 of 22 apps allowed

Location
6 of 16 apps allowed

Microphone
5 of 12 apps allowed

SMS
5 of 9 apps allowed

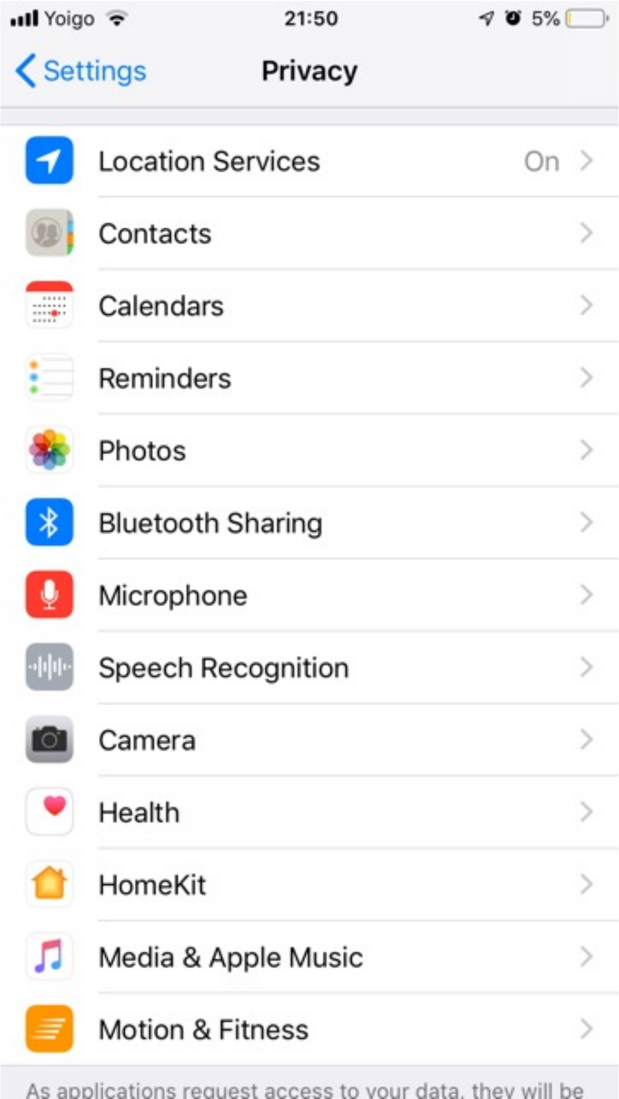
Storage
8 of 24 apps allowed

iOS Specifics

- Each app
  - Runs as user “mobile”
  - Random folder
- Apps cannot access other app data
  - Policies enforced via kernel extensions
- Two ways of enabling sensitive API calls
  - User-granted permissions
  - Entitlements

iOS Specifics - Permissions

- **More restricted than Android**
  - **No SMS and Phone**
- **Granted on run-time**
- **Apps have to be prepared to be denied a permission**
- **They can be modified by the user at any point in time**



Yoigo 21:50 5%

< Settings Privacy

Location Services On >

Contacts >

Calendars >

Reminders >

Photos >

Bluetooth Sharing >

Microphone >

Speech Recognition >

Camera >

Health >

HomeKit >

Media & Apple Music >

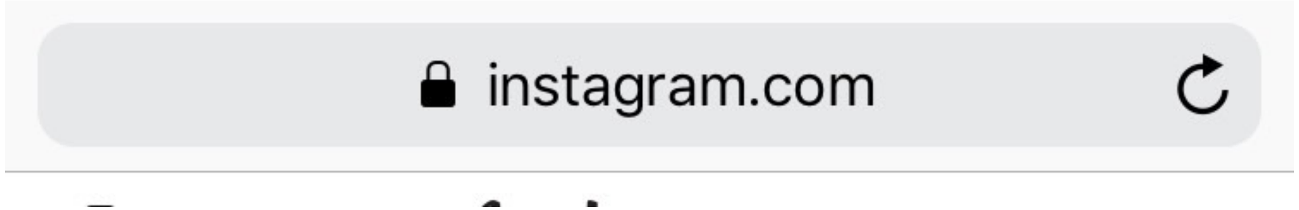
Motion & Fitness >

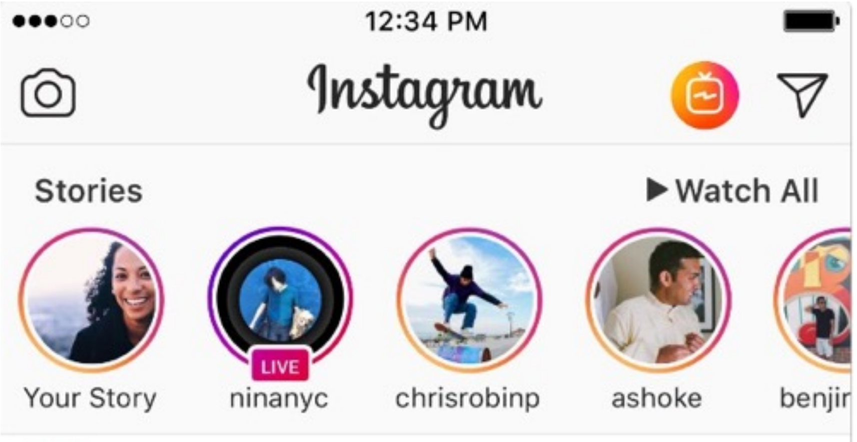
As applications request access to your data, they will be

Network Security

Network Security

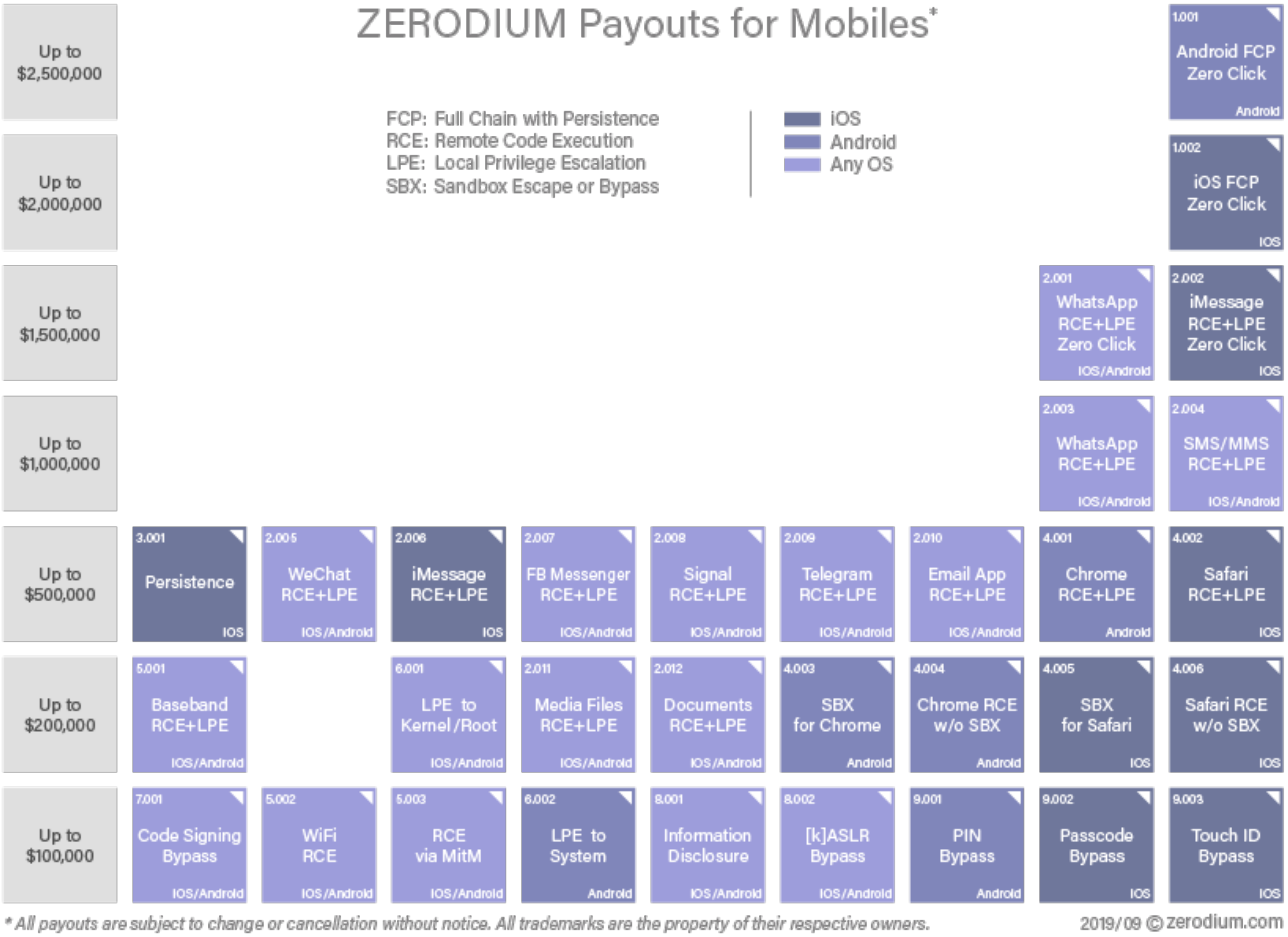
- **Apps can't show https!**





How we can make sure HTTP connection is secure?

File-Based and Full-Disk Encryption



ZERODIUM Payouts for Mobiles*

FCP: Full Chain with Persistence
RCE: Remote Code Execution
LPE: Local Privilege Escalation
SBX: Sandbox Escape or Bypass

■ iOS
■ Android
■ Any OS

Up to $2,500,000									1.001 Android FCP Zero Click Android
Up to $2,000,000									1.002 iOS FCP Zero Click iOS
Up to $1,500,000								2.001 WhatsApp RCE+LPE Zero Click iOS/Android	2.002 iMessage RCE+LPE Zero Click iOS
Up to $1,000,000								2.003 WhatsApp RCE+LPE iOS/Android	2.004 SMS/MMS RCE+LPE iOS/Android
Up to $500,000	3.001 Persistence iOS	2.005 WeChat RCE+LPE iOS/Android	2.006 iMessage RCE+LPE iOS	2.007 FB Messenger RCE+LPE iOS/Android	2.008 Signal RCE+LPE iOS/Android	2.009 Telegram RCE+LPE iOS/Android	2.010 Email App RCE+LPE iOS/Android	4.001 Chrome RCE+LPE Android	4.002 Safari RCE+LPE iOS
Up to $200,000	5.001 Baseband RCE+LPE iOS/Android		6.001 LPE to Kernel/Root iOS/Android	2.011 Media Files RCE+LPE iOS/Android	2.012 Documents RCE+LPE iOS/Android	4.003 SBX for Chrome Android	4.004 Chrome RCE w/o SBX Android	4.005 SBX for Safari iOS	4.006 Safari RCE w/o SBX iOS
Up to $100,000	7.001 Code Signing Bypass iOS/Android	5.002 WiFi RCE iOS/Android	5.003 RCE via MitM iOS/Android	6.002 LPE to System Android	8.001 Information Disclosure iOS/Android	8.002 [k]ASLR Bypass iOS/Android	9.001 PIN Bypass Android	9.002 Passcode Bypass iOS	9.003 Touch ID Bypass iOS

*All payouts are subject to change or cancellation without notice. All trademarks are the property of their respective owners.

2019/09 © zerodium.com

Outline

- Mobile Security
- Challenges and opportunities
  - How to analyse apps
    - Static vs Dynamic analysis
    - Identification of dangerous behaviours
  - Analysing apps at scale
    - Privacy Leaks
    - Flaws in BLE
    - App Collusion

Challenges and Opportunities

Threat Model

- Physical Threats
  - Thieves
  - Modifications
- Software Threats
  - Apps
  - External Exploits
- Network Threats
  - Eavesdropping
  - Integrity

App Analysis

Privacy

Security

App Analysis Techniques

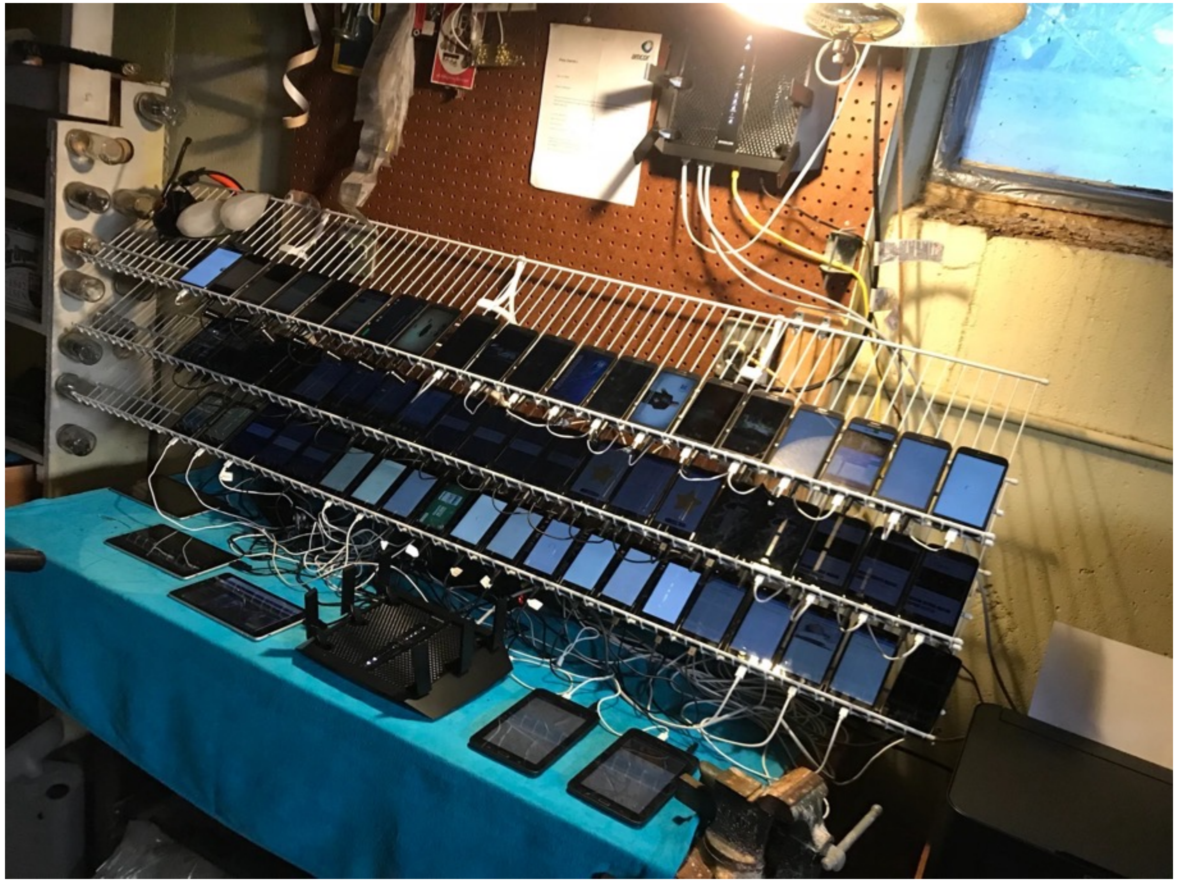
- **Static Analysis**
  - **We read and interpret the code and resources of a program**
  - **Identify parts that may lead to harmful behaviours**

- **Dynamic Analysis**
  - **We execute the program and measure what happens**
  - **Read logs, network packets, files, etc. to identify harmful behaviours**

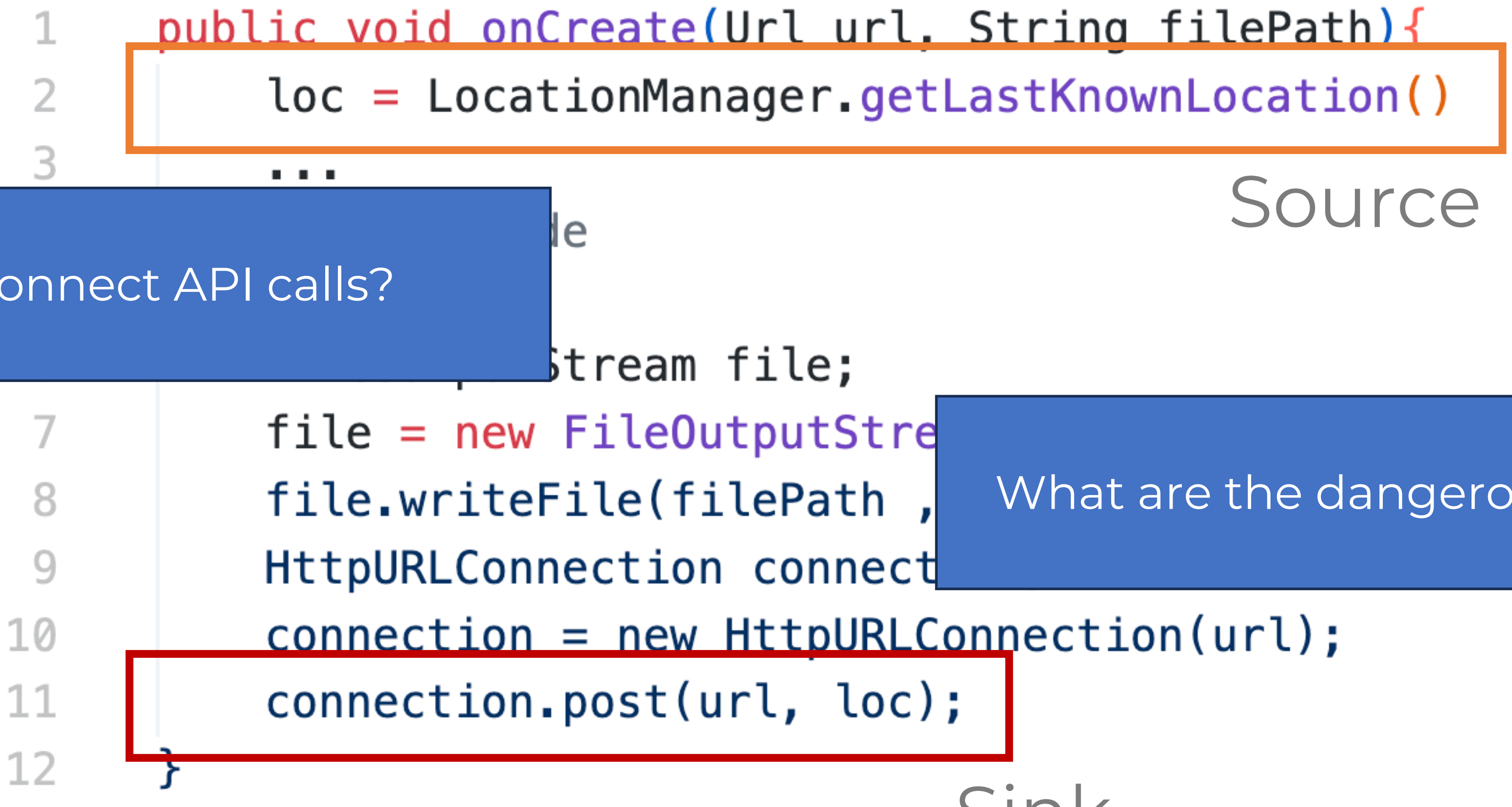
App Analysis Techniques

- **Static Analysis**
  - 👍 Fast
  - 👍 Very easy to automate
  - 👎 App may be obfuscated
  - 👎 App may download payloads
- **Dynamic Analysis**
  - 👍 As close to the real world as it gets
  - 👍 Can identify changes in behaviours and additional payloads
  - 👎 Very expensive computationally
  - 👎 How do we simulate real input?

Phone Farms



An Android Example



```
public void onCreate(Url url, String filePath){
    loc = LocationManager.getLastKnownLocation()
    ...
                   de

                   Stream file;
    file = new FileOutputStre
    file.writeFile(filePath ,
    HttpURLConnection connect
    connection = new HttpURLConnection(url);
    connection.post(url, loc);
}
```

Source

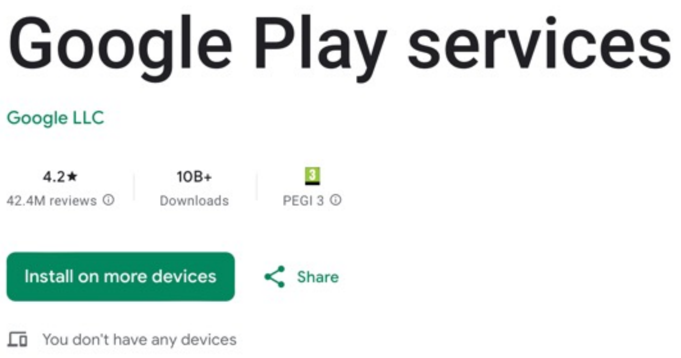
How do we connect API calls?

What are the dangerous system calls?

Sink

Where do we get information about sources and sinks?

- **Code**
  - **Android is huge!**
  - **Not all Android is Open Source!**



Google Play services

Google LLC

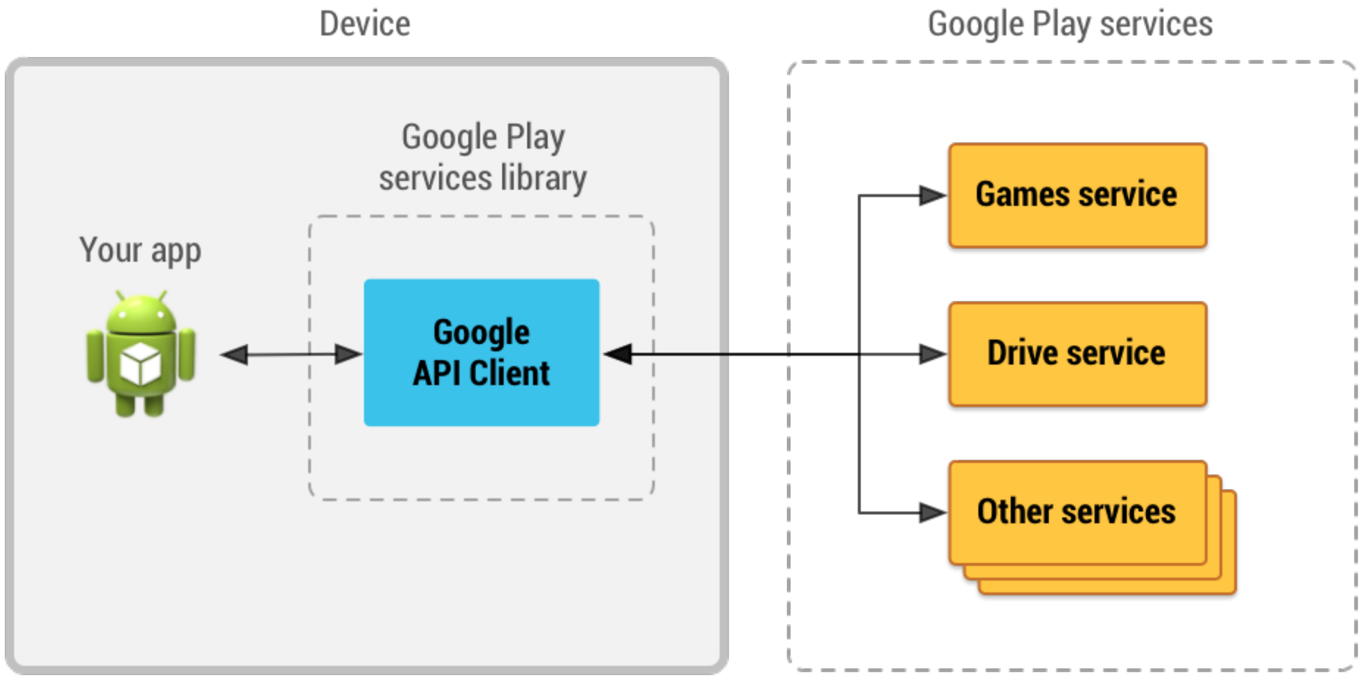
4.2★
42.4M reviews ⓘ

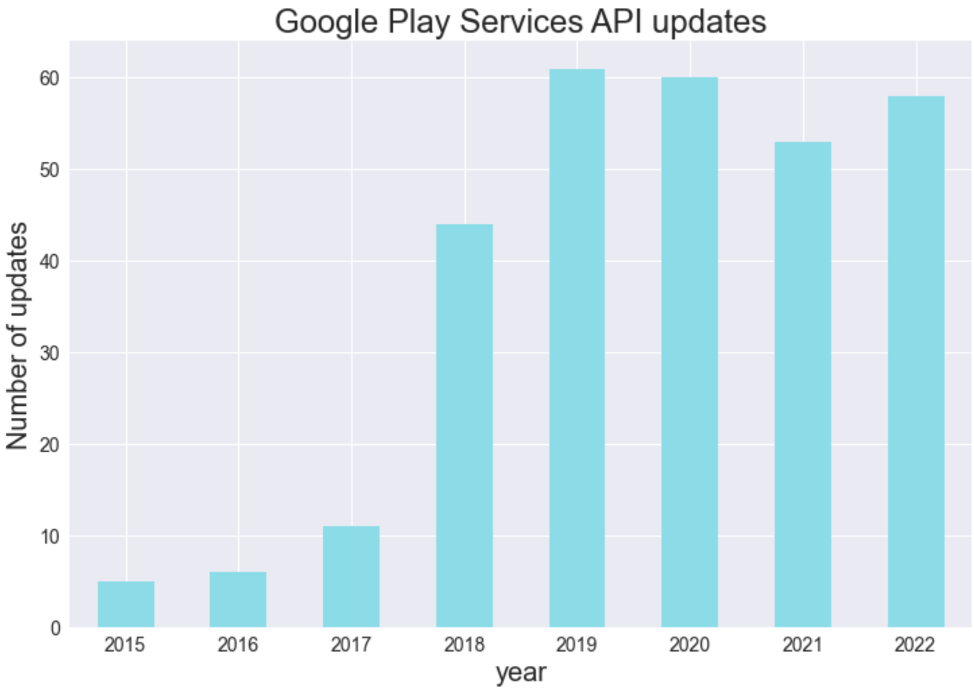
10B+
Downloads

3
PEGI 3 ⓘ

You don't have any devices

Advent of Closed-Source Google Play Services

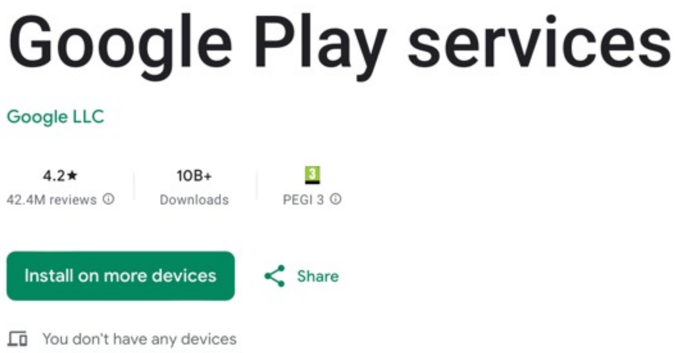




Google Play Services connects apps to other Google services, such as Google Sign-in and Google Maps.

Where do we get information about sources and sinks?

- Code
  - Android is huge!
  - Not all Android is Open Source!



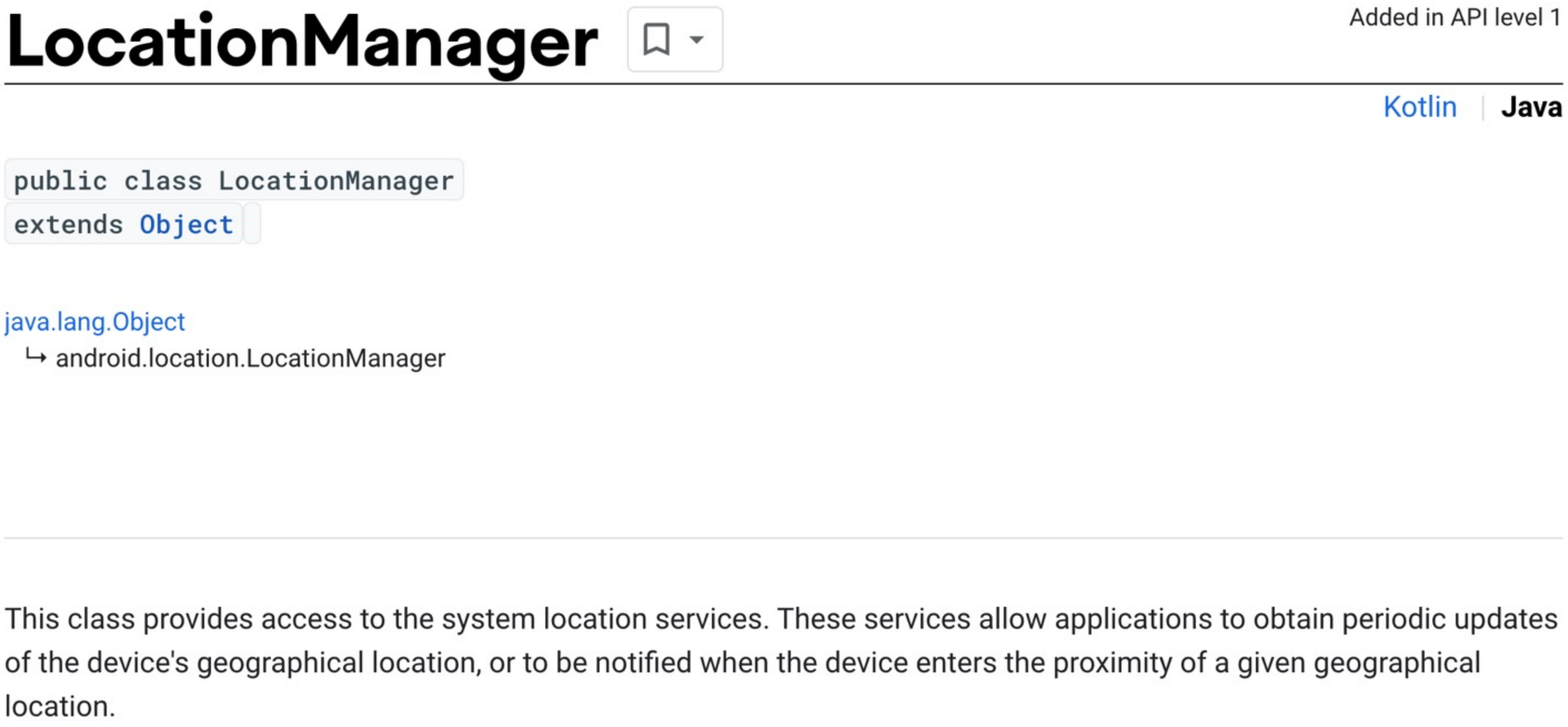
Google Play services

Google LLC

4.2★ 42.4M reviews | 10B+ Downloads | PEGI 3

- Documentation
  - Official libraries are normally well documented
  - Google has good practices that are followed by the AOSP developers

Example I



LocationManager

Added in API level 1

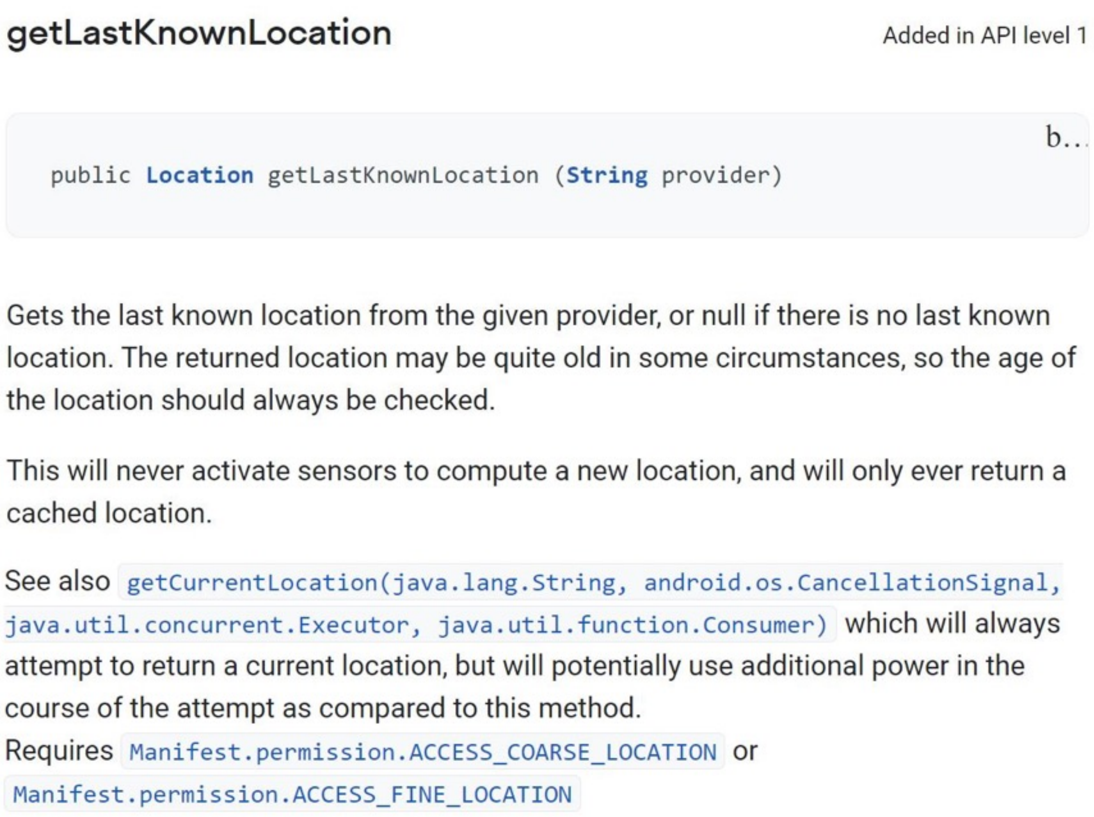
Kotlin | **Java**

```
public class LocationManager
extends Object
```

java.lang.Object
↳ android.location.LocationManager

This class provides access to the system location services. These services allow applications to obtain periodic updates of the device's geographical location, or to be notified when the device enters the proximity of a given geographical location.

Example II



getLastKnownLocation

Added in API level 1

```
b...
public Location getLastKnownLocation (String provider)
```

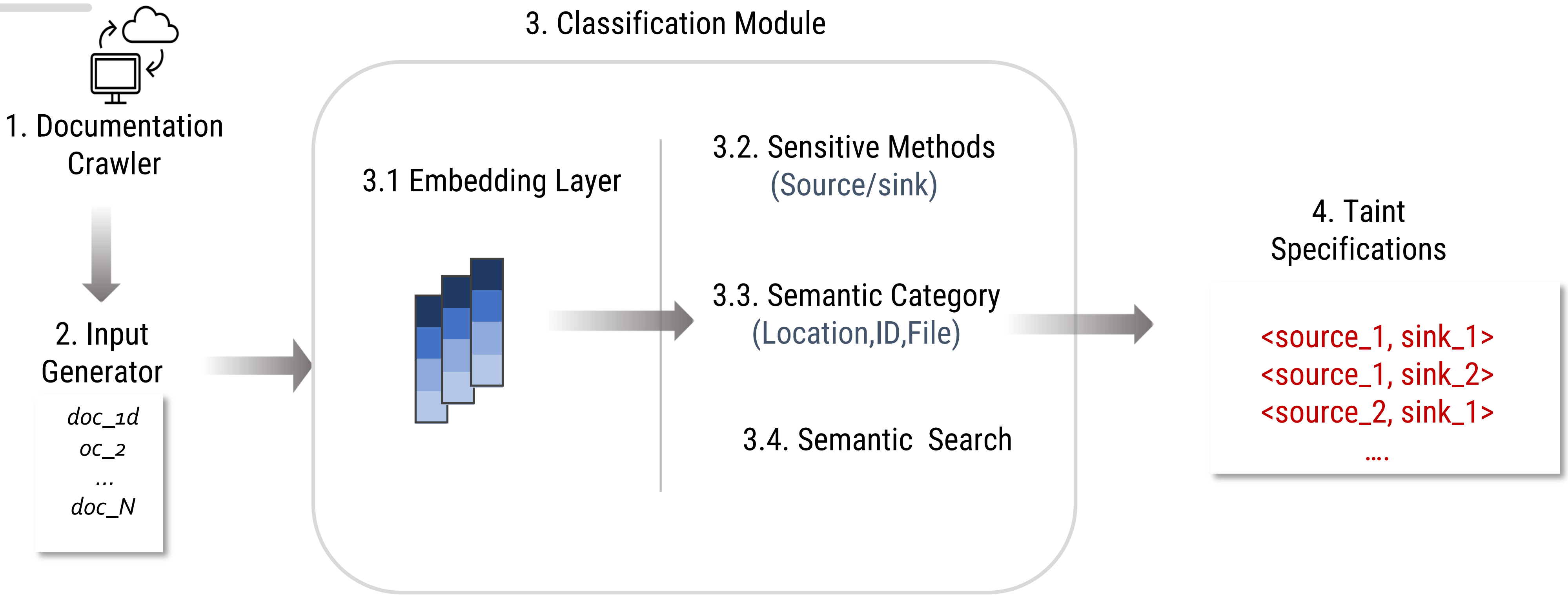
Gets the last known location from the given provider, or null if there is no last known location. The returned location may be quite old in some circumstances, so the age of the location should always be checked.

This will never activate sensors to compute a new location, and will only ever return a cached location.

See also `getCurrentLocation(java.lang.String, android.os.CancellationSignal, java.util.concurrent.Executor, java.util.function.Consumer)` which will always attempt to return a current location, but will potentially use additional power in the course of the attempt as compared to this method.
Requires `Manifest.permission.ACCESS_COARSE_LOCATION` or `Manifest.permission.ACCESS_FINE_LOCATION`

DocFlow

DocFlow Overview



1. Documentation Crawler

2. Input Generator

doc_1d
oc_2
...
doc_N

3. Classification Module

3.1 Embedding Layer

3.2. Sensitive Methods
(Source/sink)

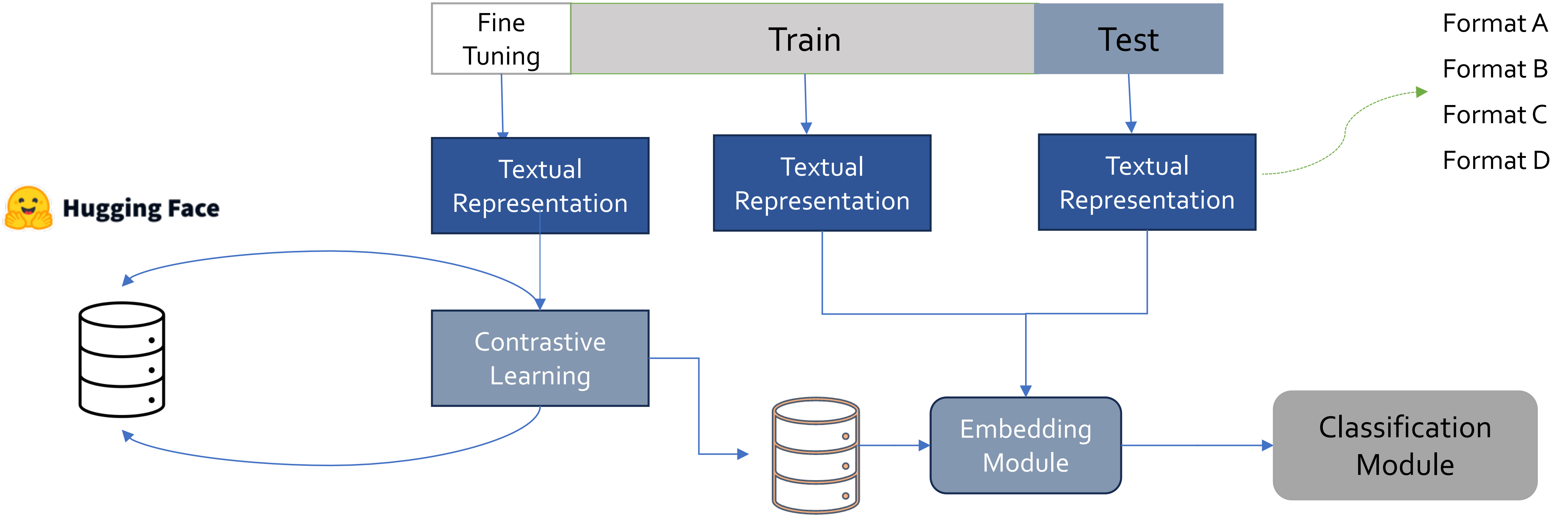
3.3. Semantic Category
(Location,ID,File)

3.4. Semantic Search

4. Taint Specifications

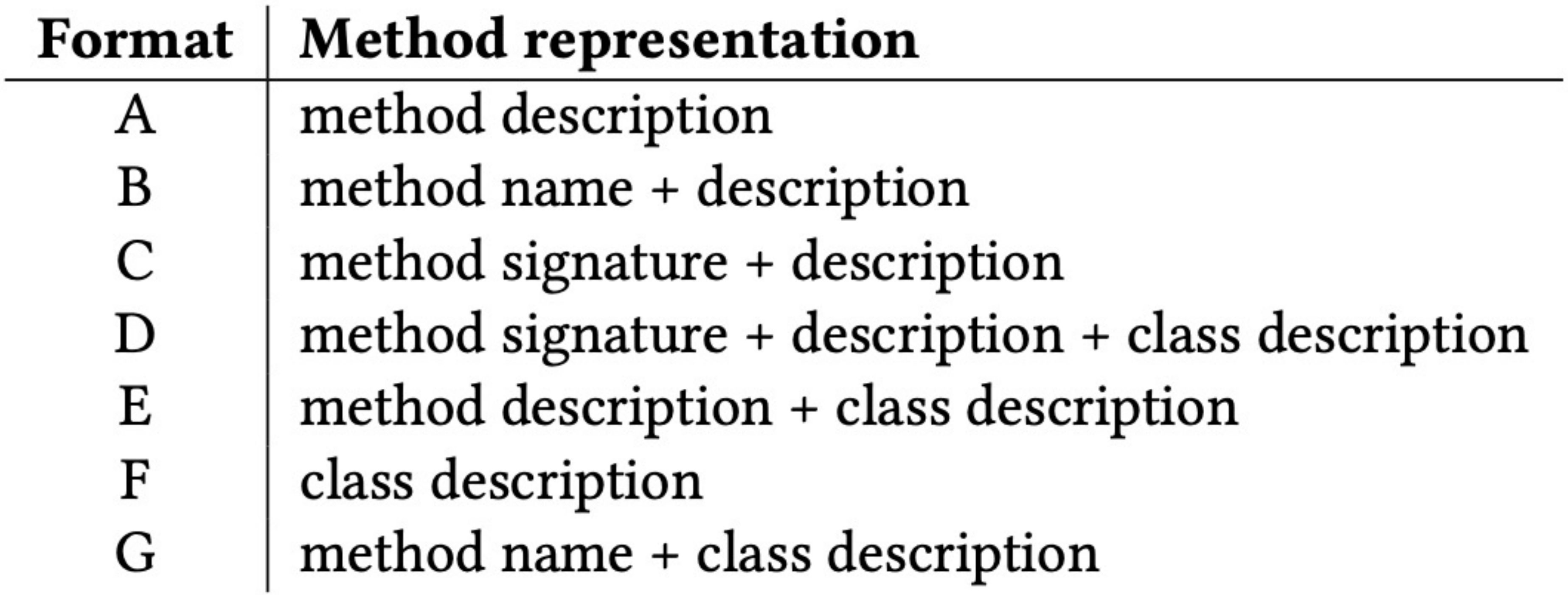
<source_1, sink_1>
<source_1, sink_2>
<source_2, sink_1>
....

DocFlow Method Classifier



all-mpnet-base-v2: maps sentences & paragraphs to a 768-dimensional dense vector space and can be used for tasks like clustering or semantic search.

Formats



Format	Method representation
A	method description
B	method name + description
C	method signature + description
D	method signature + description + class description
E	method description + class description
F	class description
G	method name + class description

Method Representation

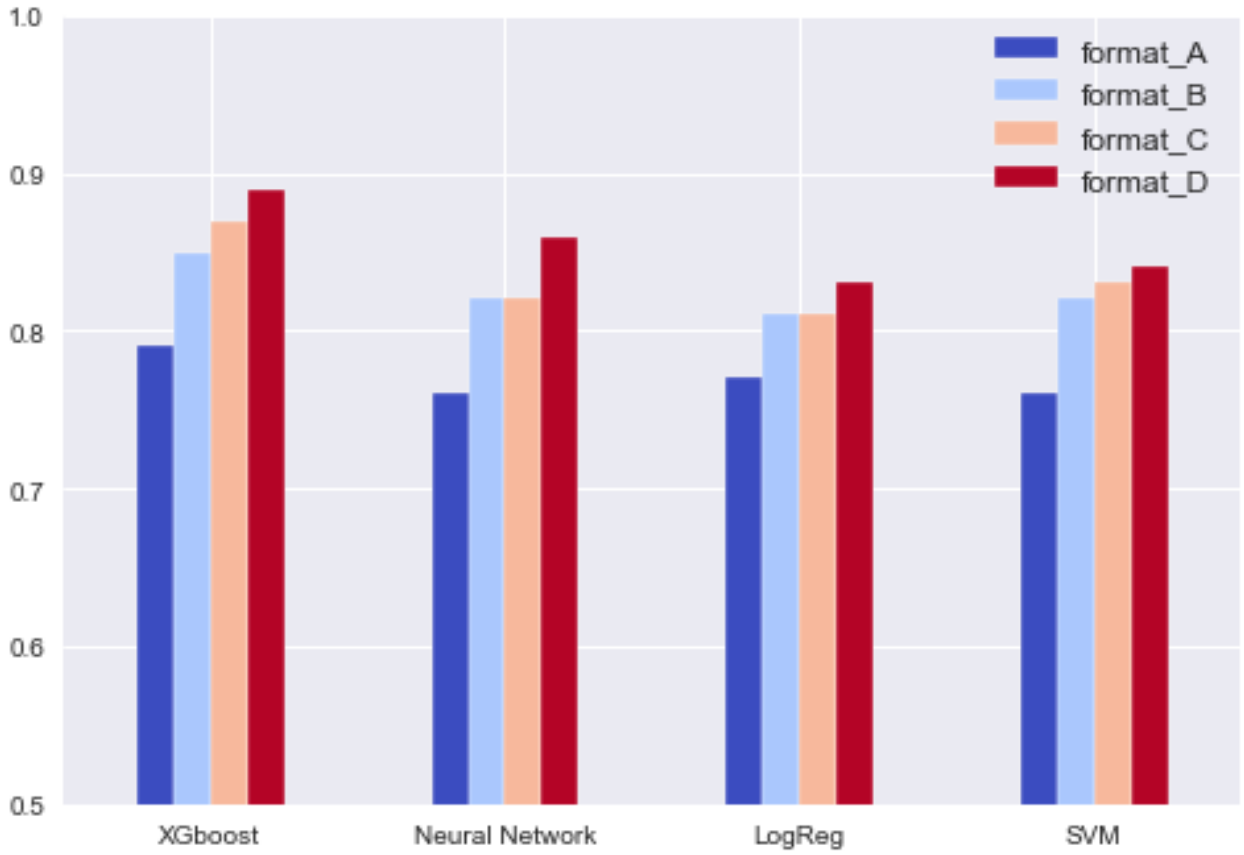
Format	features
A	Method description
B	Format A + class description

getLastKnowLocation()

- **Format A**: Gets the last known location from the given provider, or null if there is no last known location. The returned location may be quite old in some circumstances, so the age of the location should always be checked

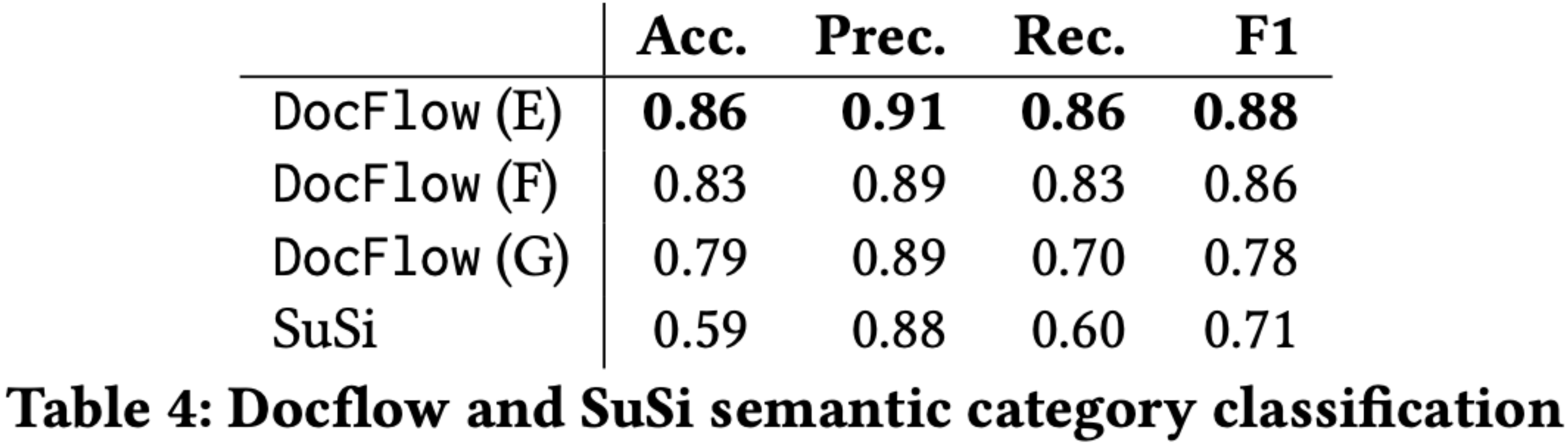
- **Format B**: Gets the last known location from the given provider, or null if there is no last known location. The returned location may be quite old in some circumstances, so the age of the location should always be checked. This will never activate sensors to compute a new location, and will only ever return a cached location.
  Requires Manifest.permission.ACCESS_COARSE_LOCATION or Manifest.permission.ACCESS_FINE_LOCATION. This class provides access to the system location services. These services allow applications to obtain periodic updates of the device's geographical location, or to be notified when the device enters the proximity of a given geographical location.

Accuracy of method classification by document representation



Format D = Method Signature + Method Description + Class Description

Semantic Category Classification



	Acc.	Prec.	Rec.	F1
DocFlow (E)	**0.86**	**0.91**	**0.86**	**0.88**
DocFlow (F)	0.83	0.89	0.83	0.86
DocFlow (G)	0.79	0.89	0.70	0.78
SuSi	0.59	0.88	0.60	0.71

Table 4: Docflow and SuSi semantic category classification

Conclusions

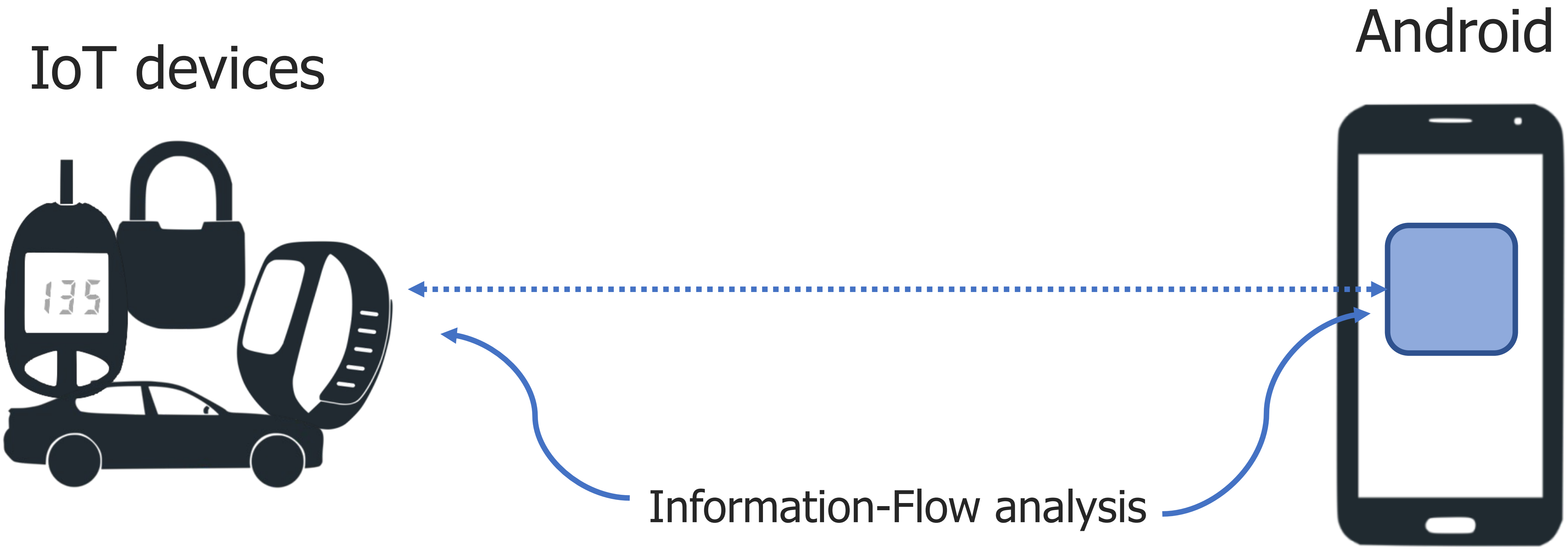
- Software documentation contains rich semantic information about security (and probably other properties)

- New Large Language Models can be leveraged to automatically extract that information

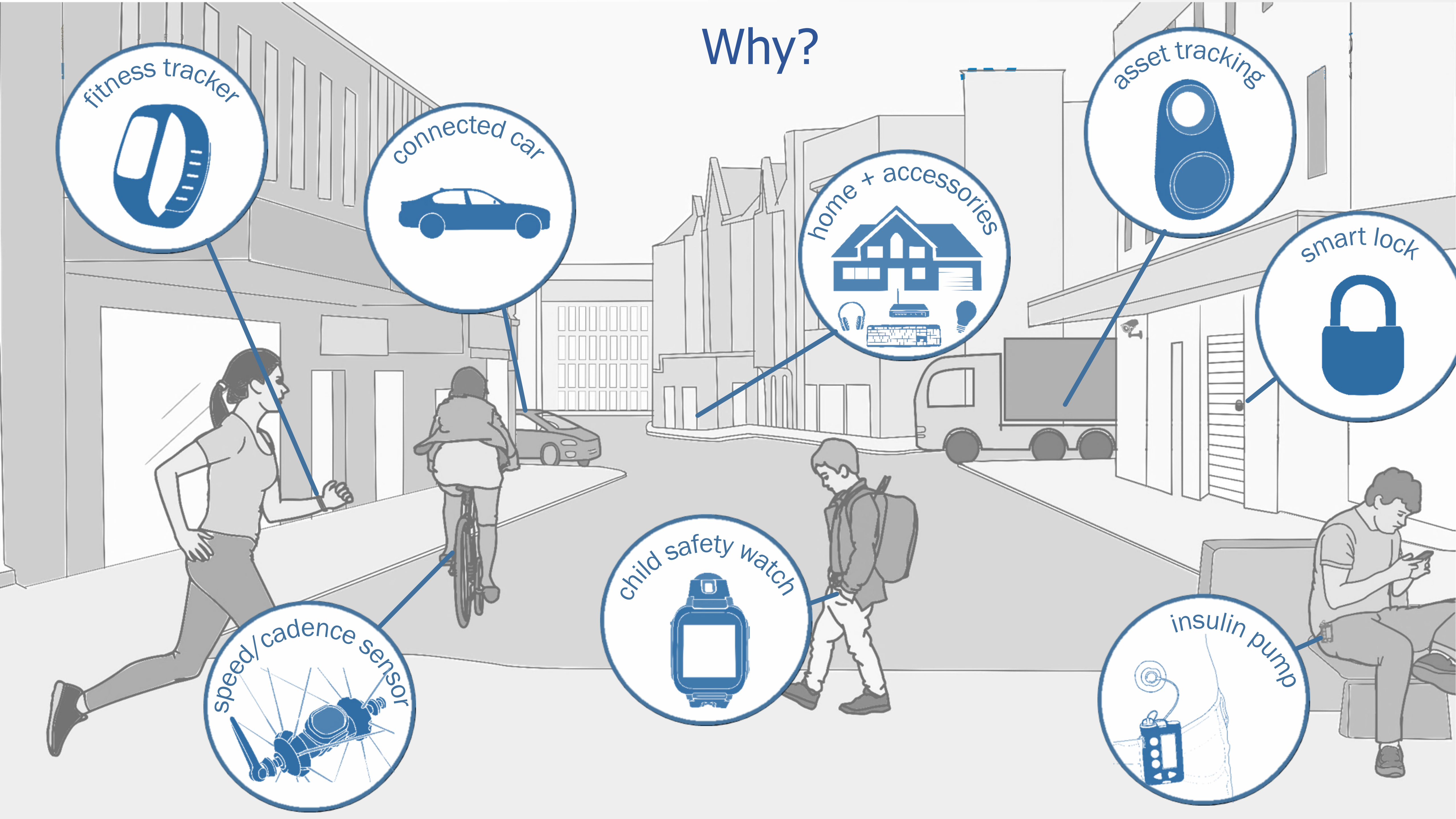
- This can be very useful to quickly incorporate new libraries or new operating system versions into security analysis tools

Identification of Security vulnerabilities

Context



Why?


fitness tracker
connected car
home + accessories
asset tracking
smart lock
speed/cadence sensor
child safety watch
insulin pump

Goals

- Analyse Mobile Apps
  - Datasets widely available[1]

- To identify
  - Privacy Leaks in Wear OS
  - Security vulnerabiliites in how BLE apps handle data

- Use Information-Flow Analysis
  - Analyse specific cases

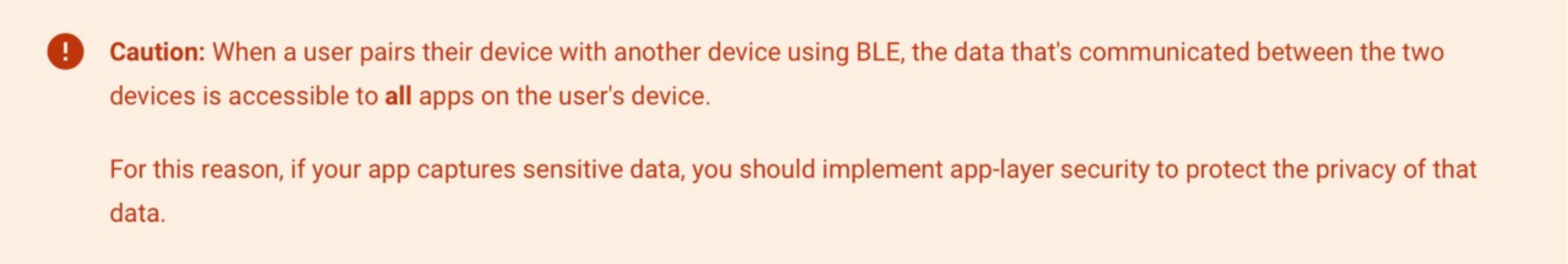
[1] Allix, K., Bissyandé, T. F., Klein, J., & Le Traon, Y. (2016, May). Androzoo: Collecting millions of android apps for the research community. In *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)* (pp. 468-471). IEEE.

Privacy Leaks in WearOS

- **Permission Delegation**
  - **Mobile app requests permissions and sends data to Wear App**
- **Data Leak**
  - **Data is transmitted to another device and is leaked to the internet from there**
- **Obfuscation**
  - **Split code between main app and companion app to make análisis difficult**

BLE vulnerabilities in Android

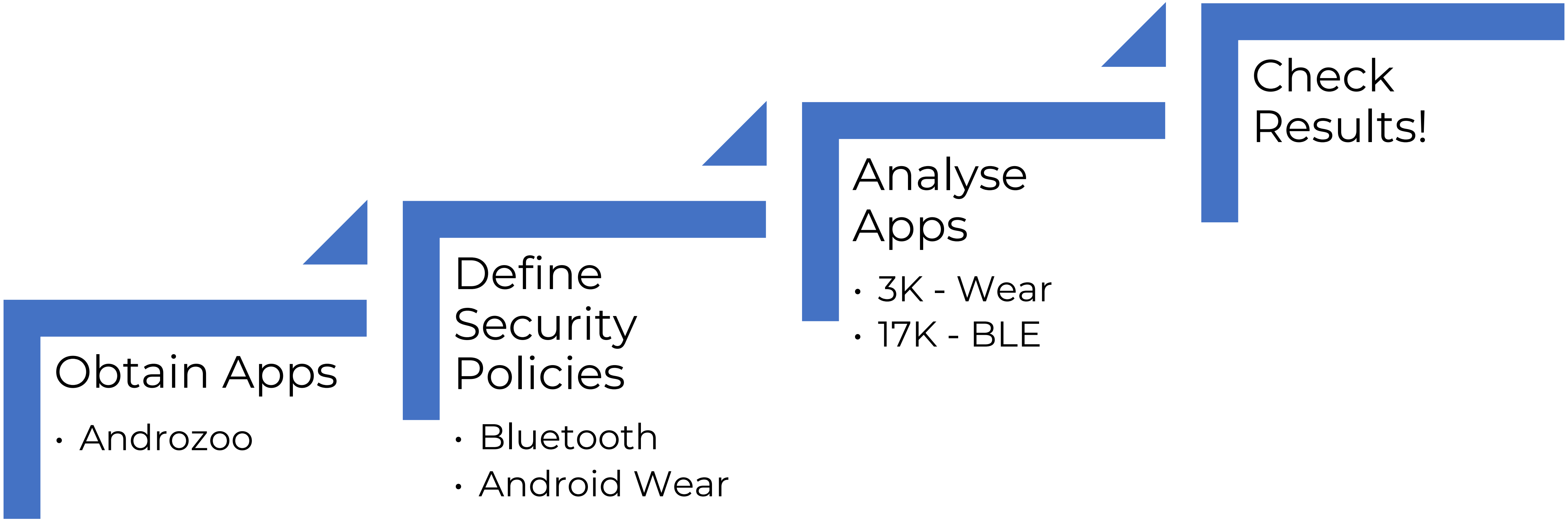
- **Bluetooth is a normal permission in Android**
  - **Only checked during installation**
- **Any app that requests this permission can access any BLE device connected to the phone**



❗ **Caution:** When a user pairs their device with another device using BLE, the data that's communicated between the two devices is accessible to **all** apps on the user's device.

For this reason, if your app captures sensitive data, you should implement app-layer security to protect the privacy of that data.

Process



Obtain Apps

- Androzoo

Define Security Policies

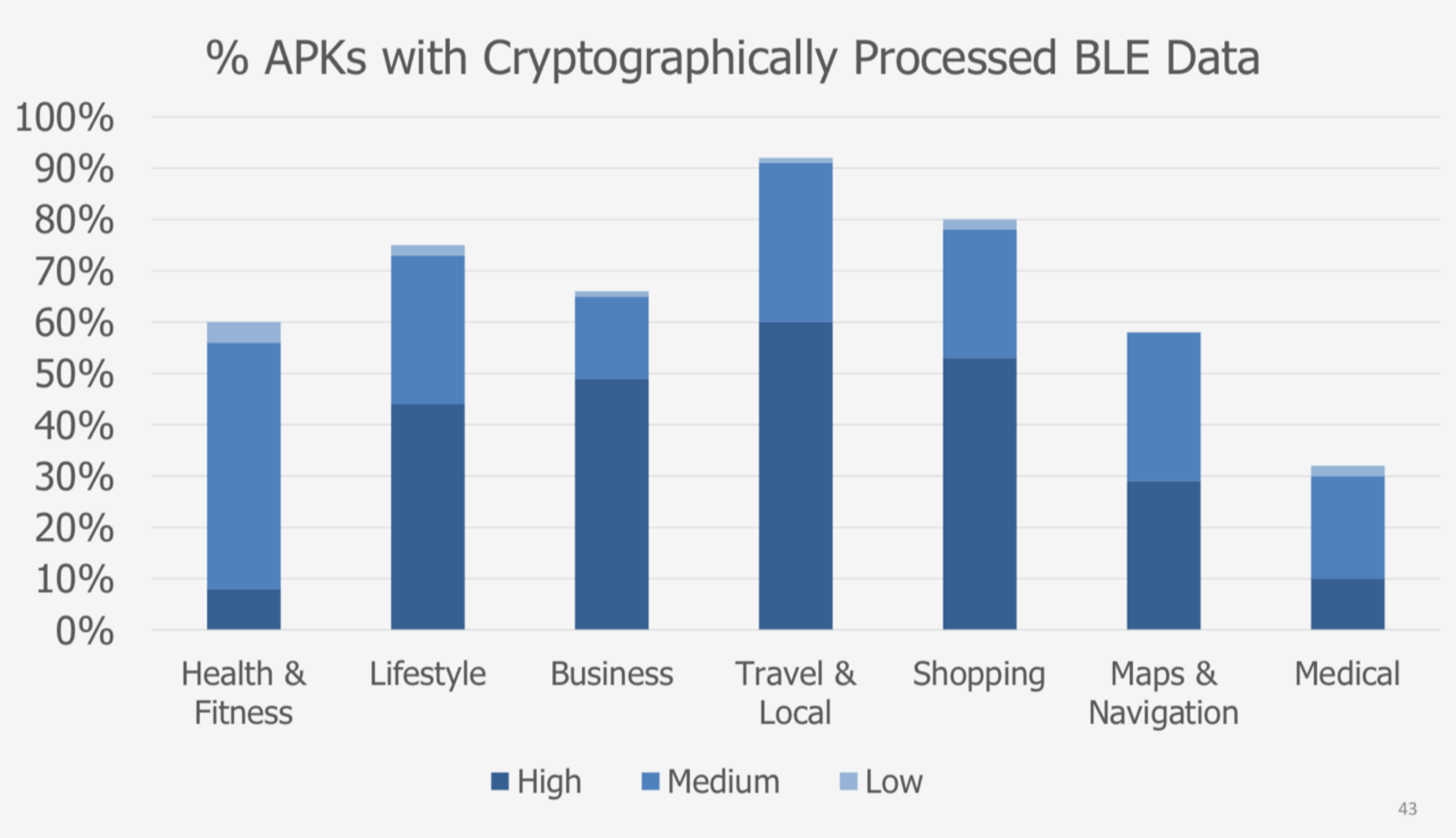
- Bluetooth
- Android Wear

Analyse Apps

- 3K - Wear
- 17K - BLE

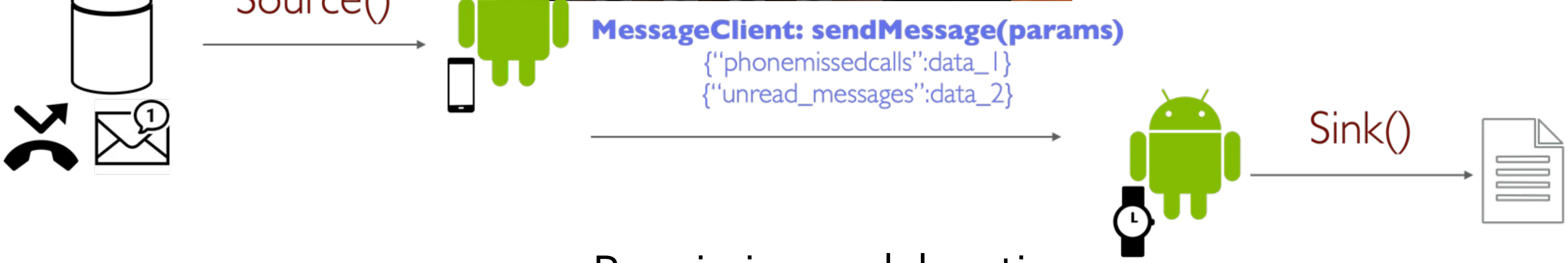
Check Results!

Vulnerabilities in BLE Processed data



Android Wear Example

fr.thema.wear.watch.venom


Source()
MessageClient: sendMessage(params)
{"phonemissedcalls":data_1}
{"unread_messages":data_2}
Sink()
1

Permission re-delegation

Conclusions

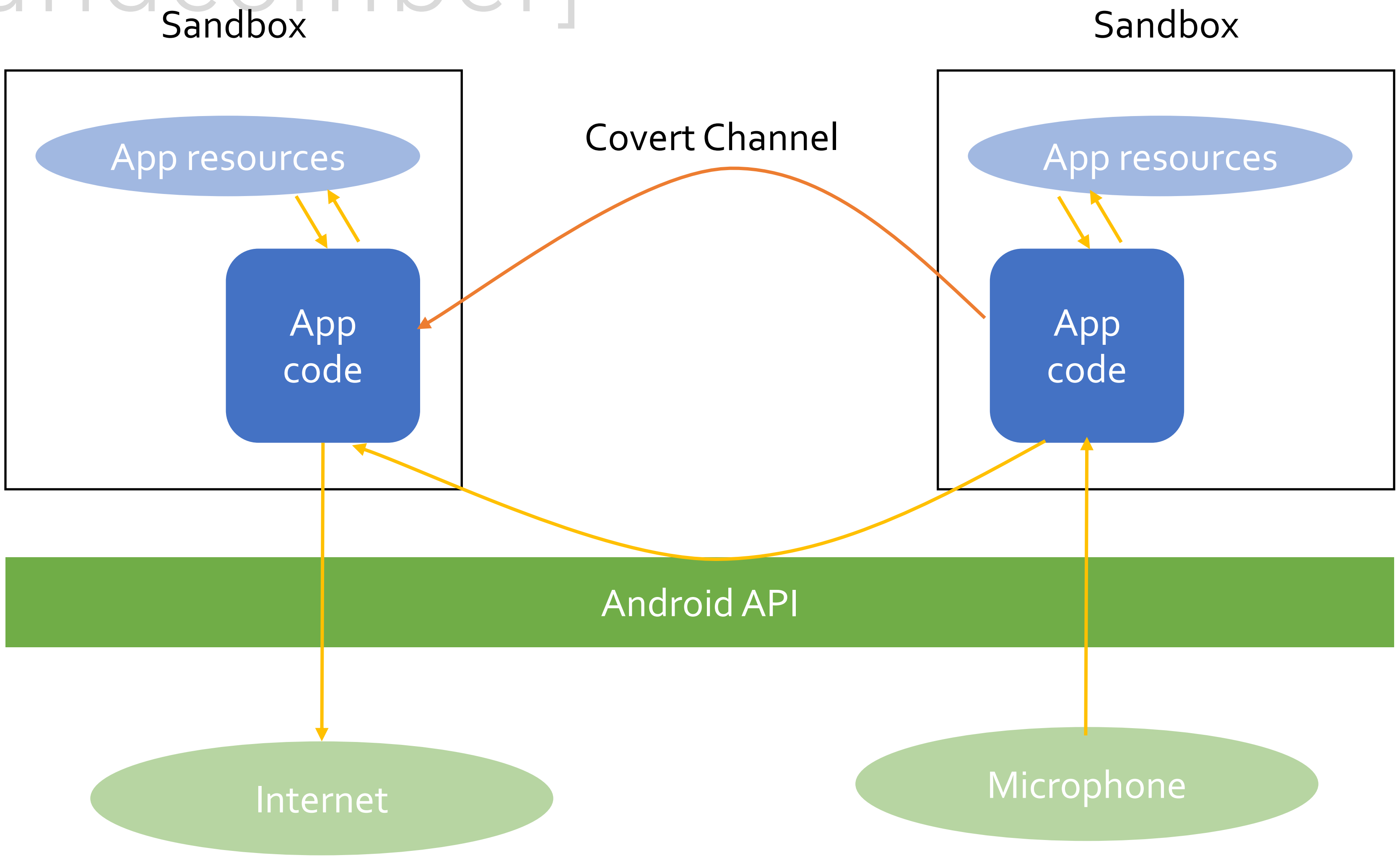
- **Software developers assume the platform they use will provide them with all security they need**

- **Apps, even when related to medical domains tend to have poor privacy and security practices**

- **Cheap devices usually result in bad implemented security**

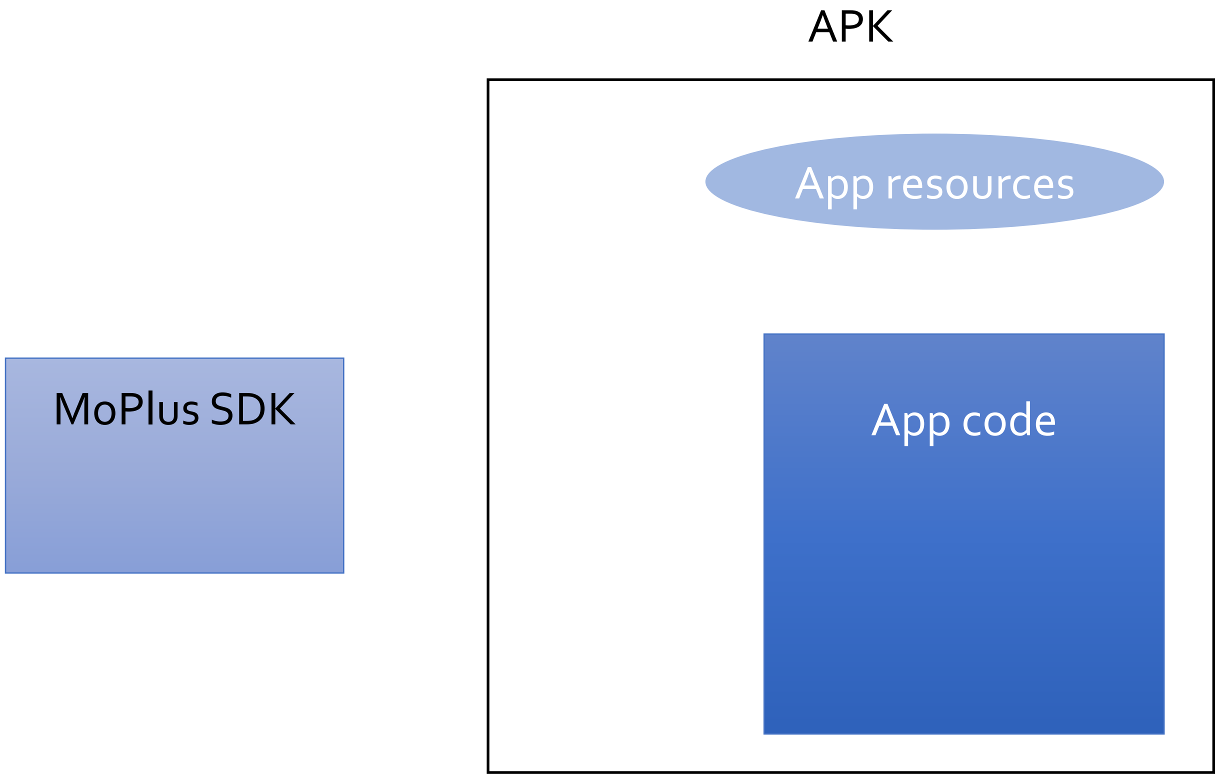
App Collusion

Application Collusion [Soundcomber]



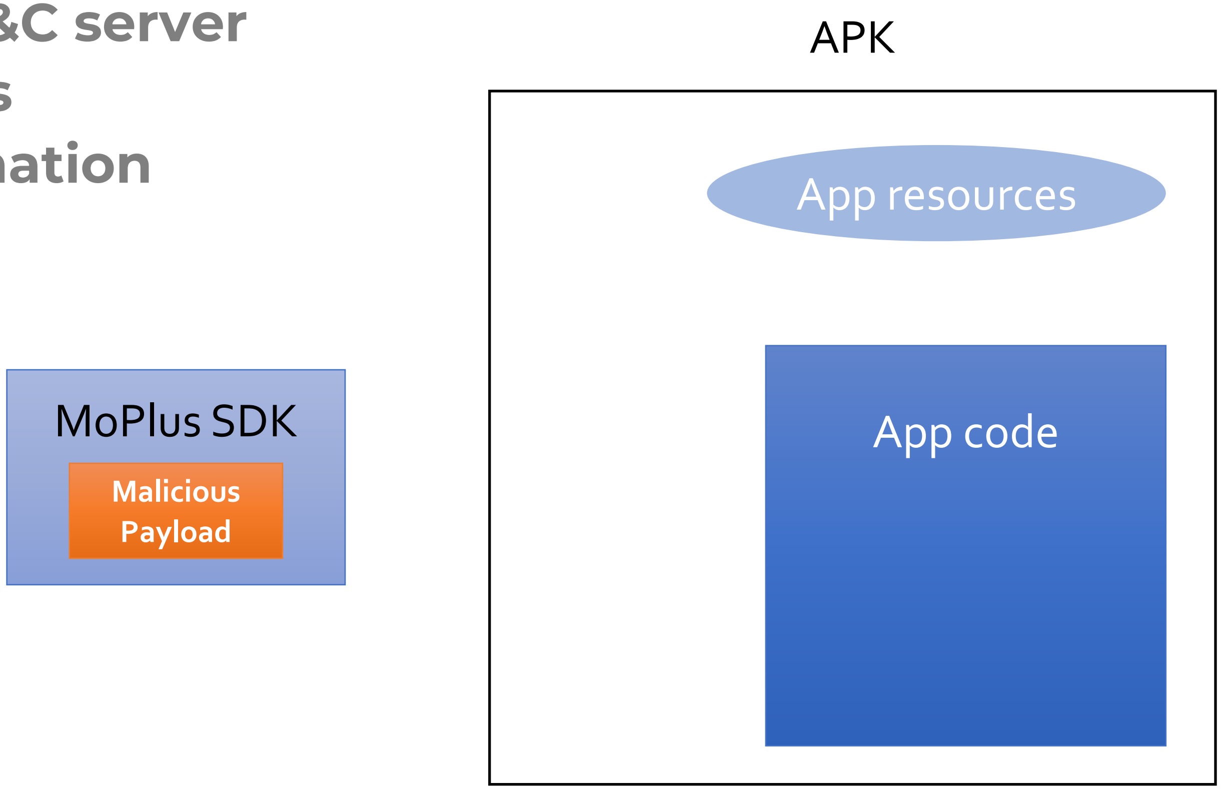
MoPlus SDK

Embedding a Library into your app



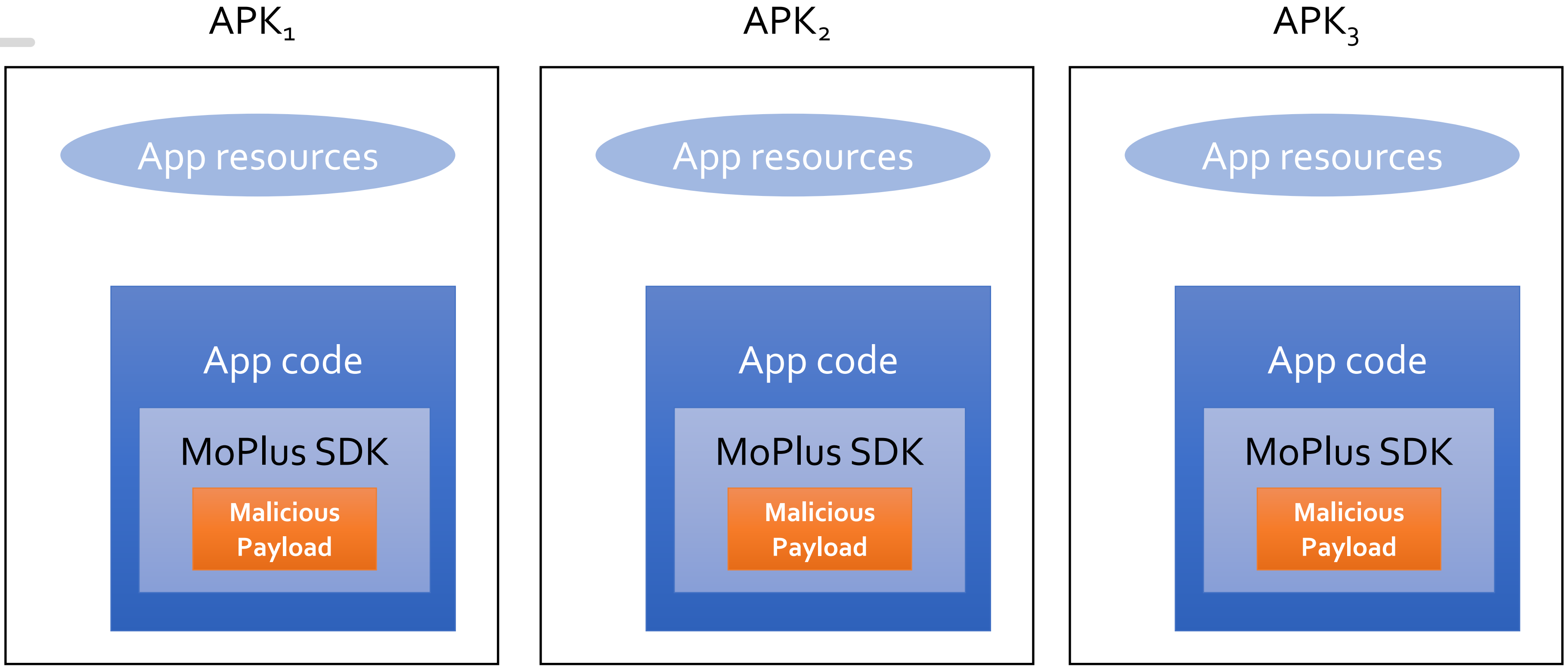
Malicious Behaviour

- Open port to listen C&C server
- Send arbitrary intents
- Read sensitive information
- Install apps (rooted)
- Add contacts



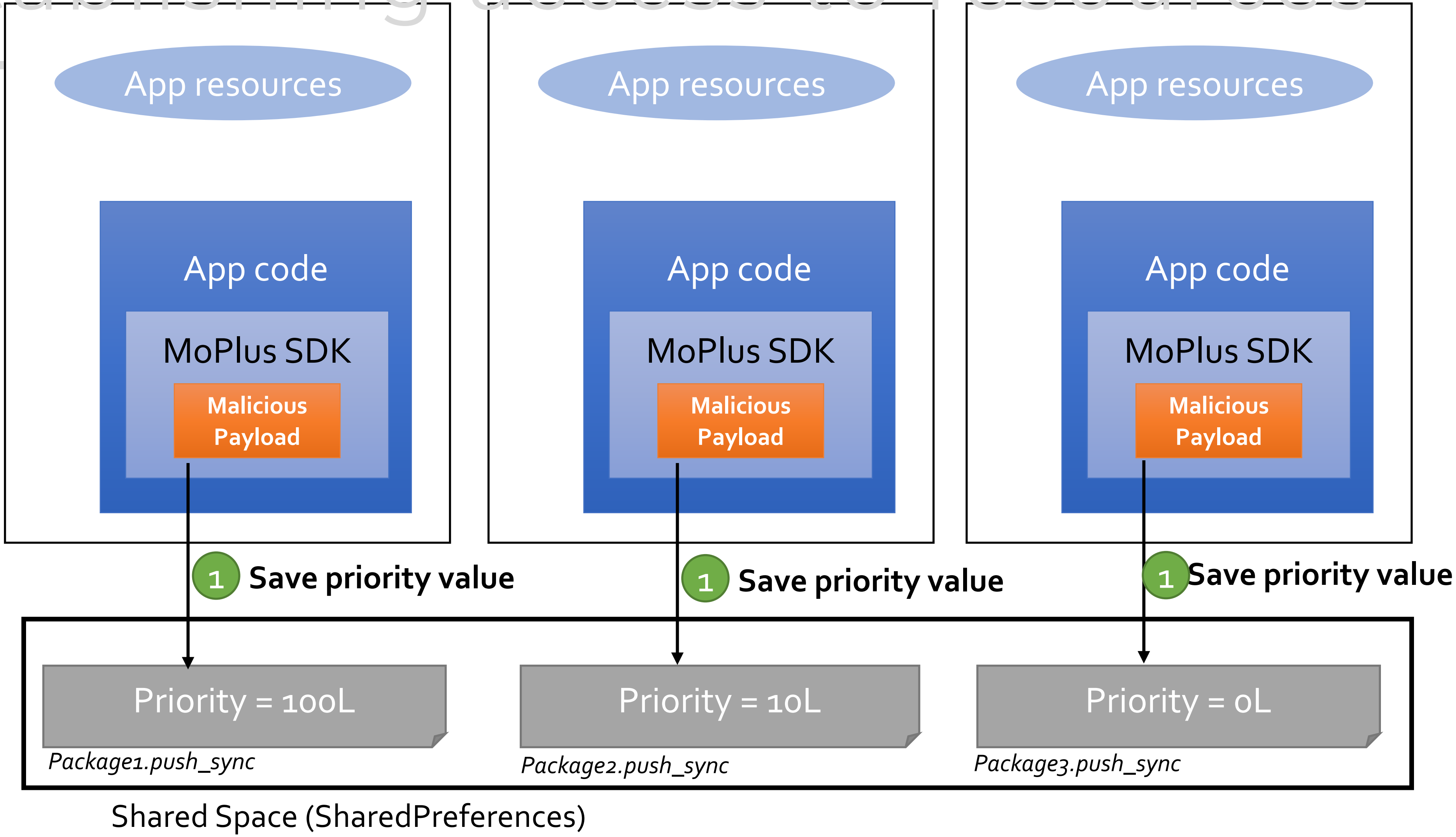
What happens if the app lacks enough permissions?

Colluding behaviour

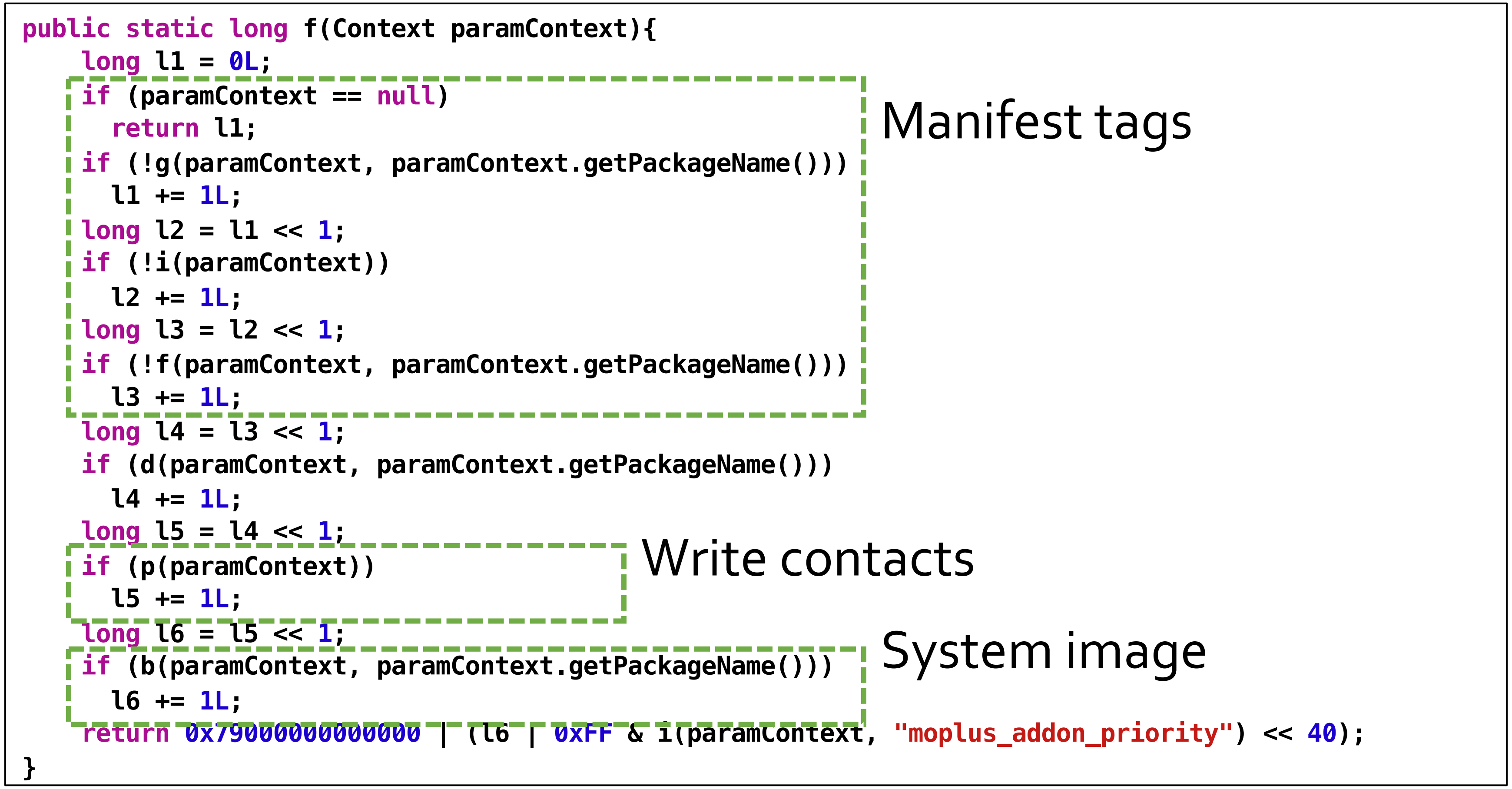


$$P_1 \neq P_2 \neq P_3$$

Establishing access to resources



Establishing the priority value



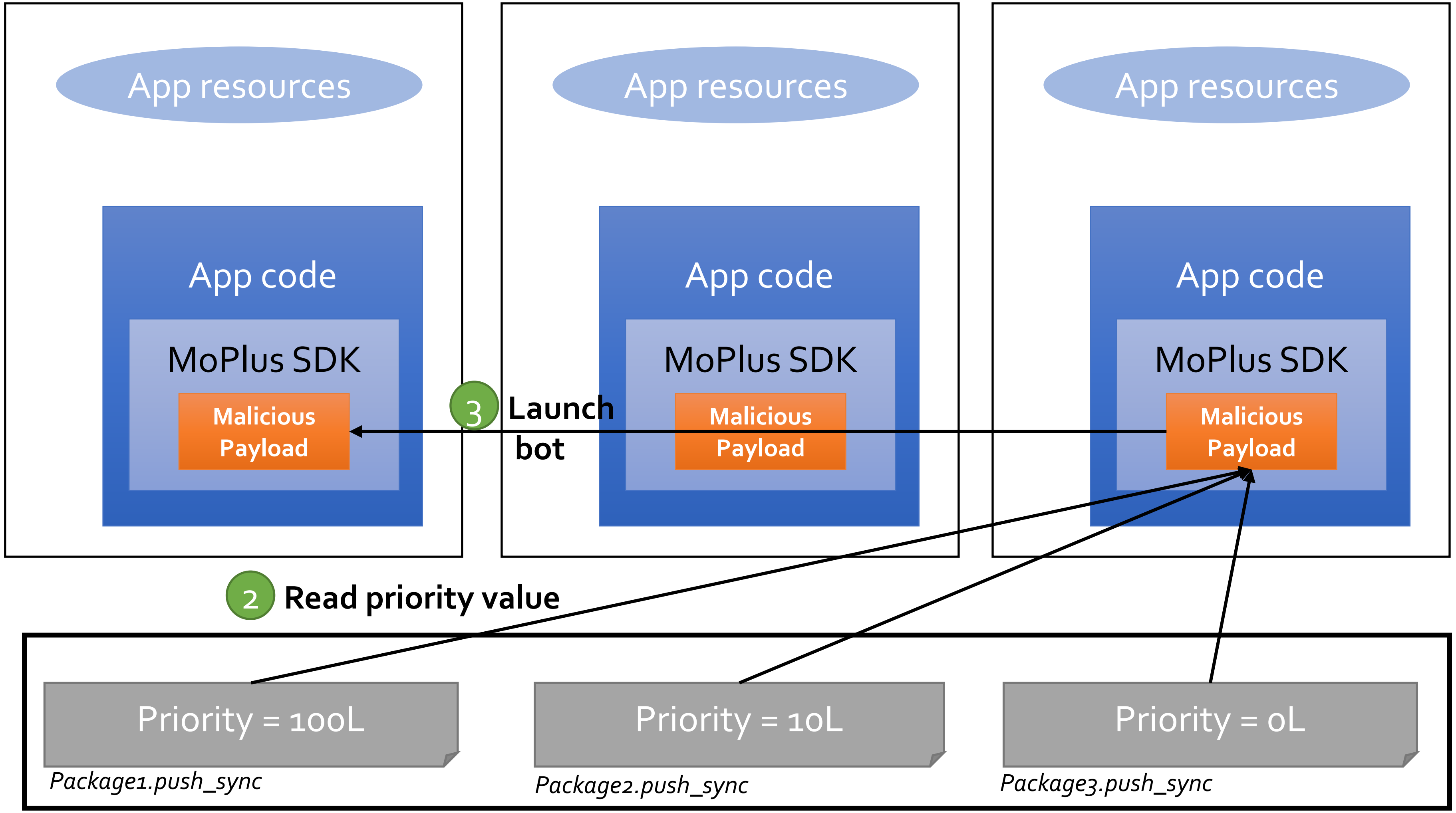
```
public static long f(Context paramContext){
    long l1 = 0L;
    if (paramContext == null)
      return l1;
    if (!g(paramContext, paramContext.getPackageName()))
      l1 += 1L;
    long l2 = l1 << 1;
    if (!i(paramContext))
      l2 += 1L;
    long l3 = l2 << 1;
    if (!f(paramContext, paramContext.getPackageName()))
      l3 += 1L;
    long l4 = l3 << 1;
    if (d(paramContext, paramContext.getPackageName()))
      l4 += 1L;
    long l5 = l4 << 1;
    if (p(paramContext))
      l5 += 1L;
    long l6 = l5 << 1;
    if (b(paramContext, paramContext.getPackageName()))
      l6 += 1L;
    return 0x79000000000000 | (l6 | 0xFF & i(paramContext, "moplus_addon_priority") << 40);
}
```

Manifest tags

Write contacts

System image

Launching the malicious payload



Conclusions

Conclusions

- Mobile phones are ubiquitous and part of our everyday lives
- Because of that they are appealing to
  - Criminals via malware
  - Data greedy companies
- This presents a series of challenges on how we can do security analysis at scale
- New advancements in analysis techniques and machine learning are great opportunities for defenders to reduce the gap and make applications more secure