Multi-Criteria Evaluation Model to Generate Tentative Energy Ratings for Google Play Store Apps

Abdullah M Almasri

Under the guidance of
Prof. Doctor Luis Borges Gouveia

University Fernando Pessoa, Portugal

Disclaimer

This research did not cover Apple smartphones for reasons related to lack of open source materials

Motivation

Nokia 3310

Popular Uses

- SMS
- Voice Dialing
- Calculator
- Snake
- Customize Ringtones
- To break a glass

Battery Standby

For up to a month

https://www.nokia.com/phones/en_int/nokia-3310

Motivation

Smartphones

Popular Uses

- Internet connectivity.
- A mobile browser.
- The ability to sync more than one email account to a device.
- Embedded memory.
- A hardware or software-based QWERTY keyboard.
- Wireless synchronization with other devices, such as laptop or desktop computers.
- The ability to download applications and run them independently.
- Support for third-party applications.
- The ability to run multiple applications simultaneously.
- Touchscreen.
- Wi-Fi.
- A digital camera, typically with video capability.
- Gaming.
- Unified messaging.
- GPS.

Average Battery Standby

6 hours

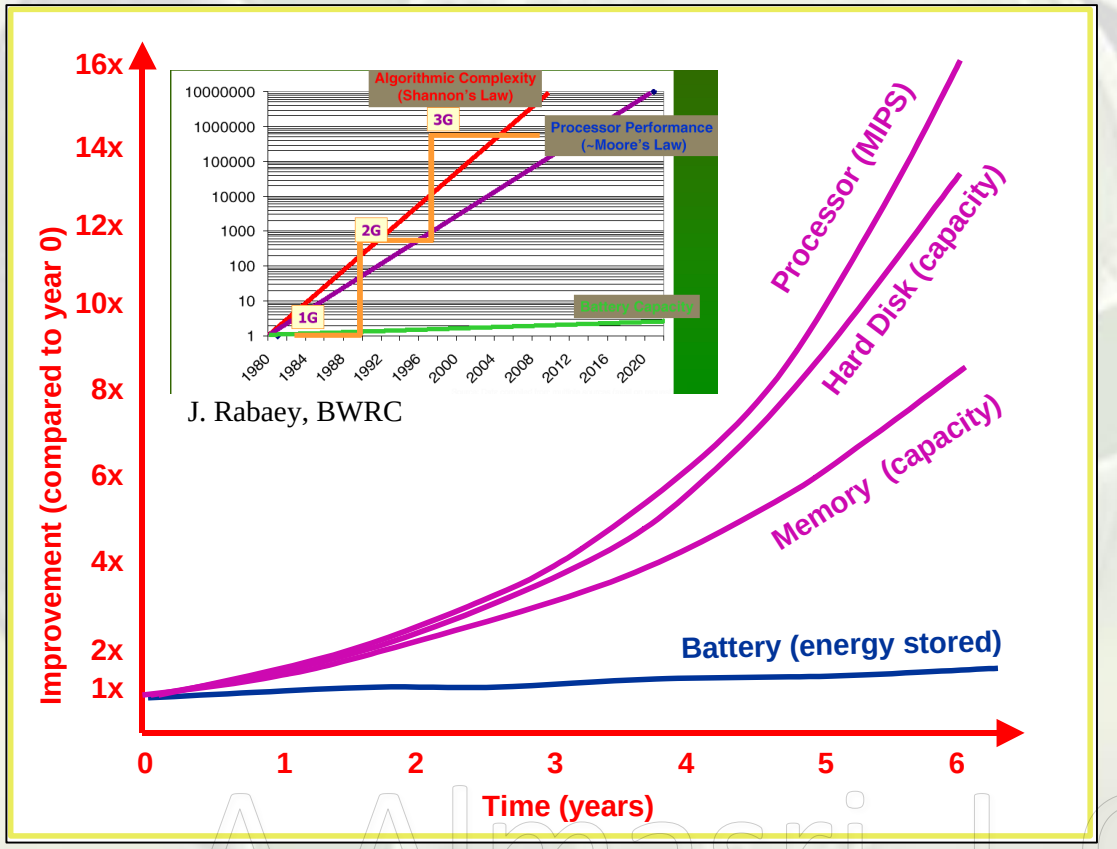
Motivation

Reverse Relationship between phone functionality and battery life

Less Technology More Battery Life

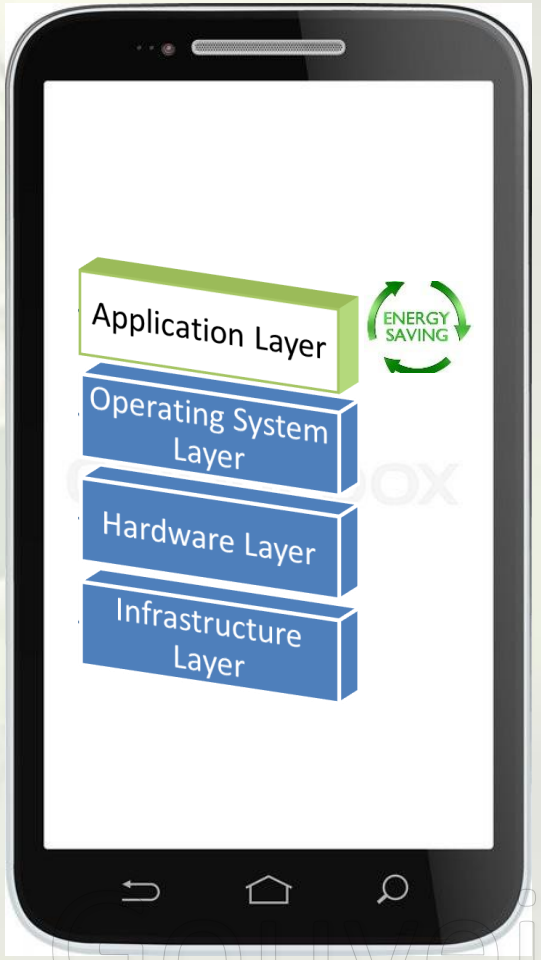
More Technology Less Battery Life

Motivation

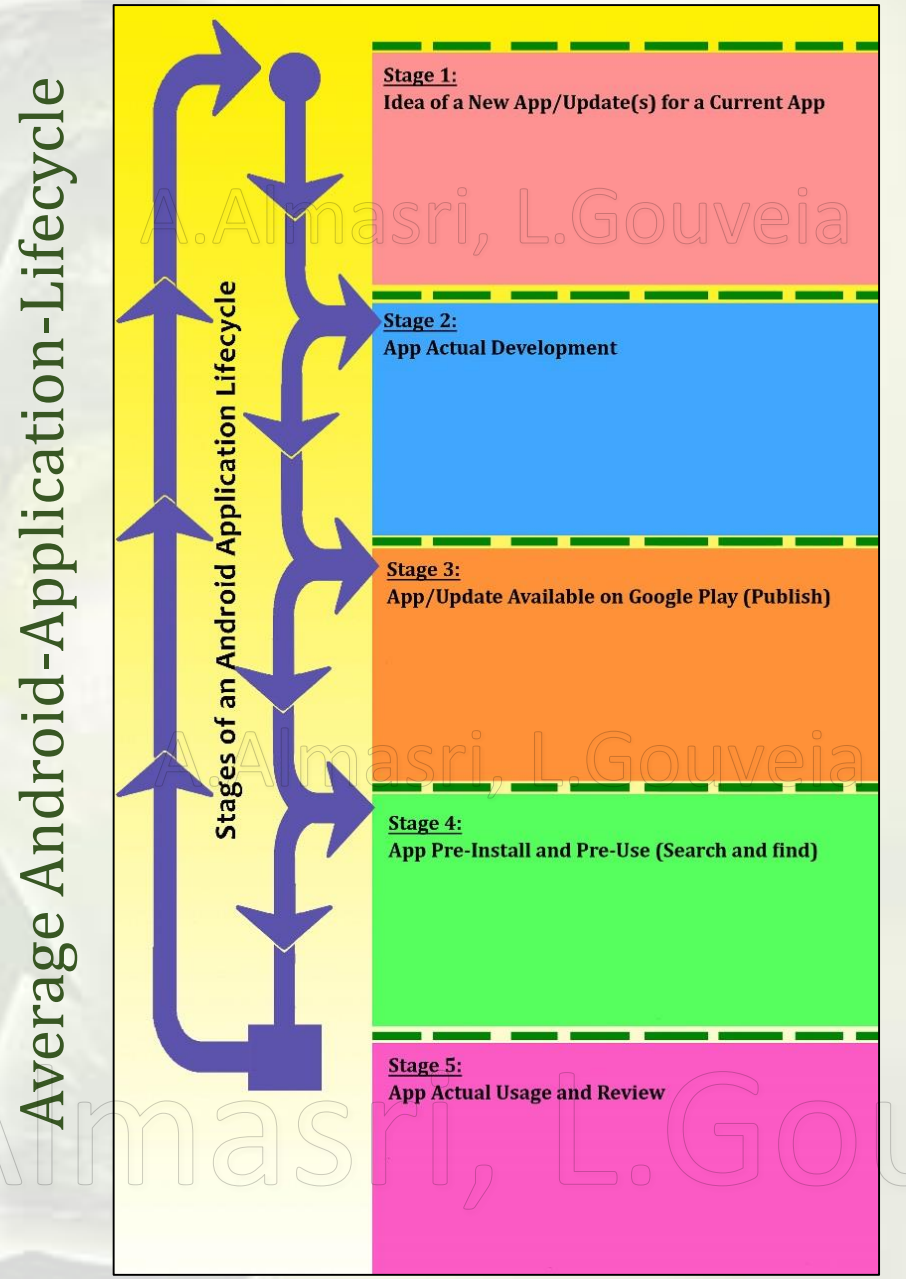


Introduction:

- A common issue among android smartphones is the battery life.
- The Need to Save Smartphones Battery-Life During COVID-19 Pandemic is more important to consider as many governments implemented location-tracking applications.



Understanding Power-saving Basic Approaches



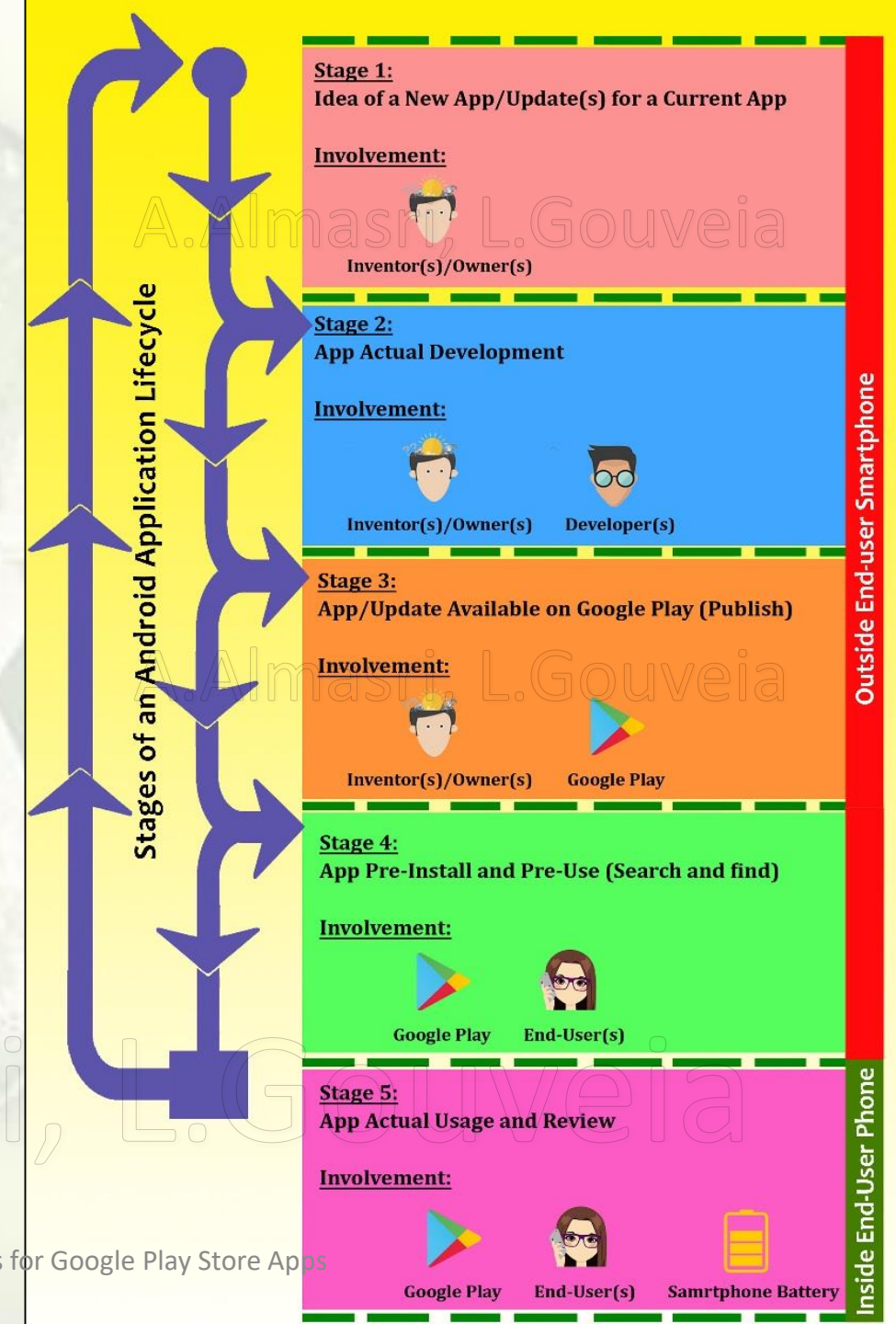
List the Concerned Parties and Identify their Involvement

- Following the previous step, we list the parties which are involved in an android application lifecycle

Status of App Among Different Stages of an Application Lifecycle

1. Outside the End-User's Phone
   (Under development or Available on Google Play)

2. Inside the End-User's Phone
   (Installed & Running)



Current power-saving approaches and its usage among the lifecycle

Approach 1:

To Simulate & Estimate

or

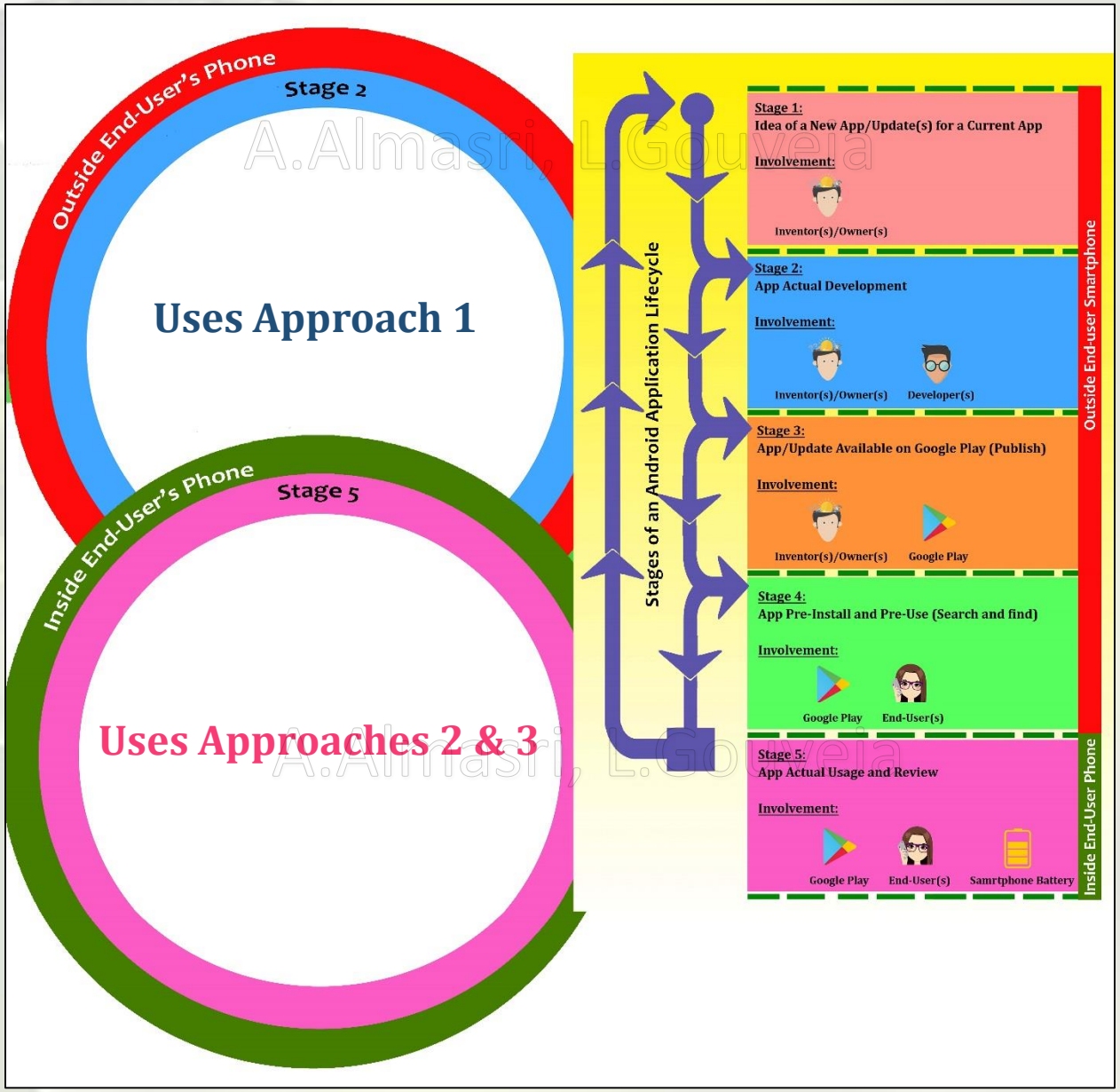
To Apply Thin Client

Approach 2:

To Monitor, Detect and Control

Approach 3:

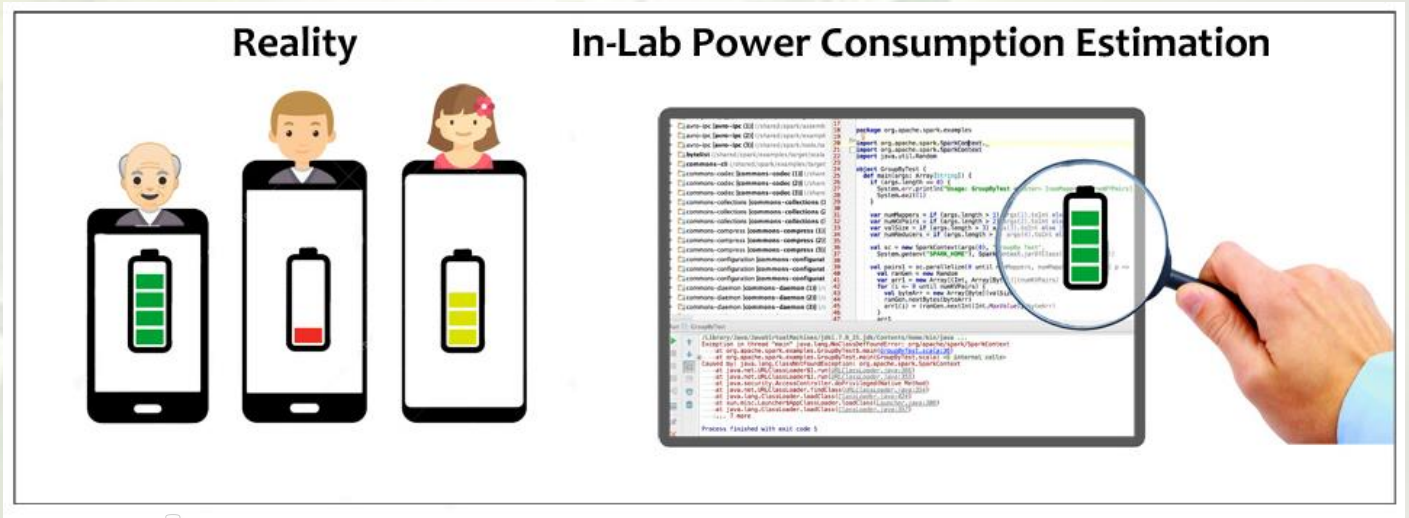
To Sacrifice Technology



Key Issues of Current Power-Saving Approaches

Key Issue of the “Simulate & Estimate” Approach:

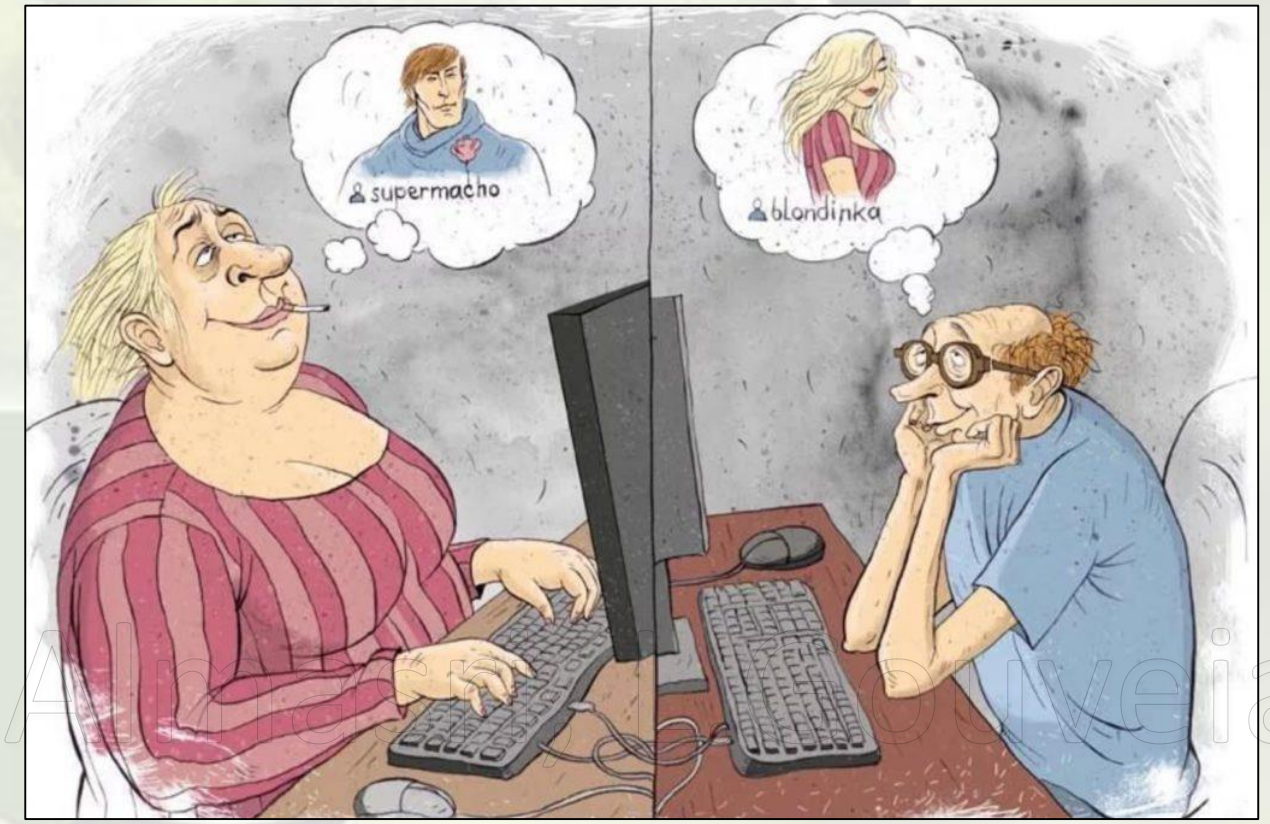
Questioned Accuracy



Key Issues of Current Power-Saving Approaches

Key Issue of the “Simulate & Estimate” Approach:

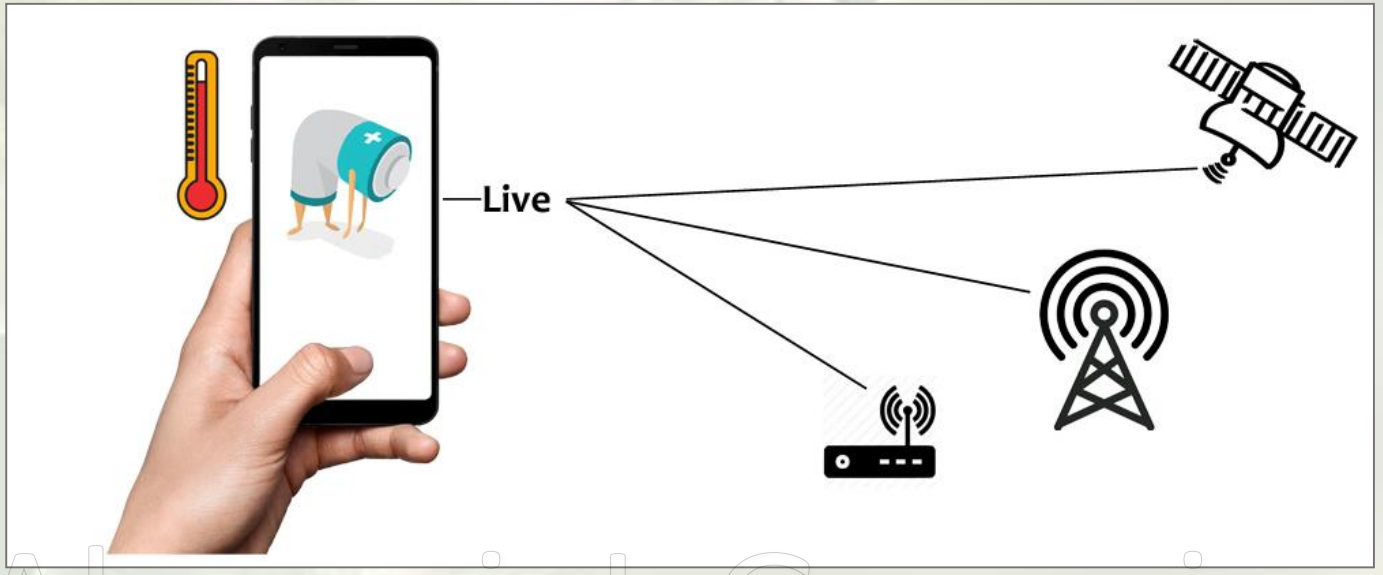
Imagination-Based !!



Key Issues of Current Power-Saving Approaches

Key Issue of the “Thin Client” Approach :

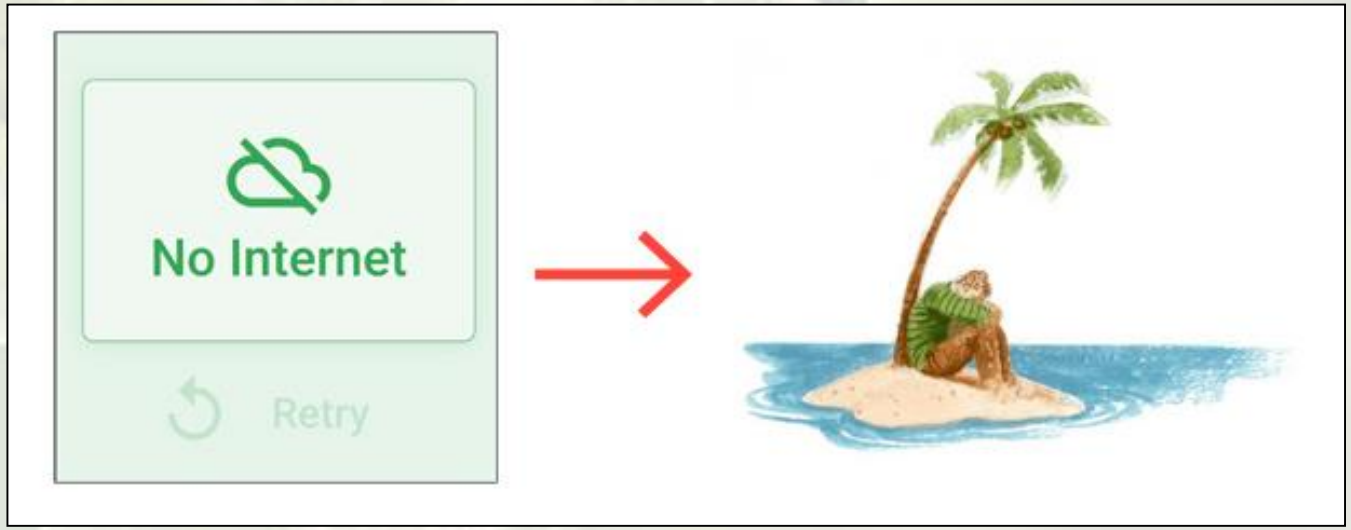
Continuous Connectivity !!



Key Issues of Current Power-Saving Approaches

Key Issue of the “Thin Client” Approach :

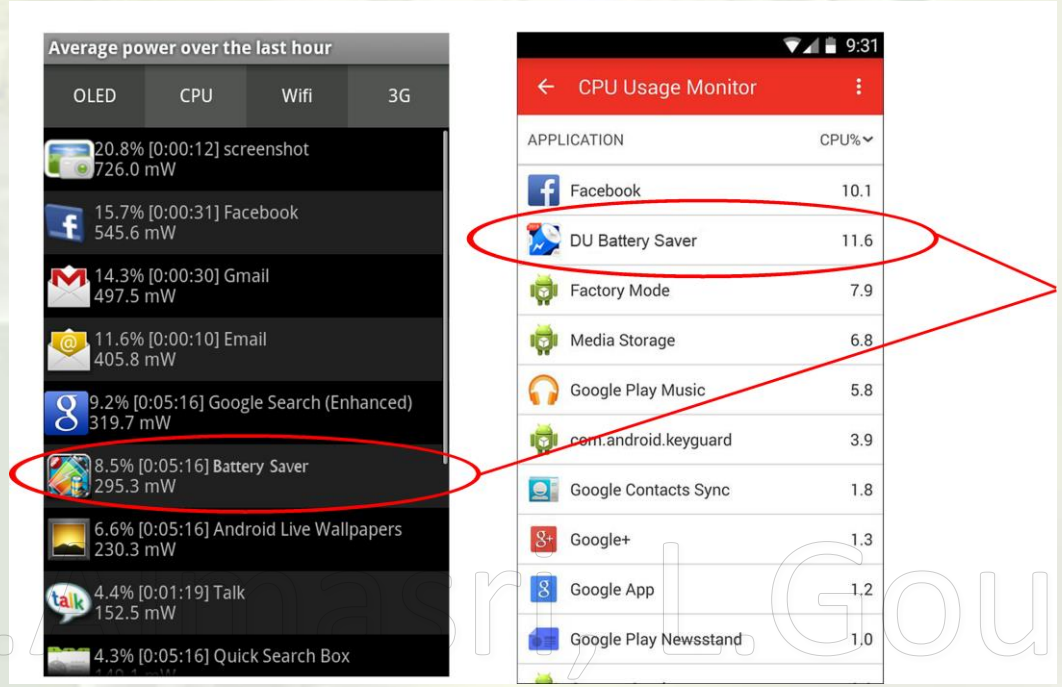
Service Outages !!



Key Issues of Current Power-Saving Approaches

Key Issue of “Monitor and control” Approach :

Requires Power !!



Using PowerTutor® and Trepn Profiler® to read the amount of energy consumed by two popular power-optimizing applications

Key Issues of Current Power-Saving Approaches

Key Issue of “Sacrifice” Approach :

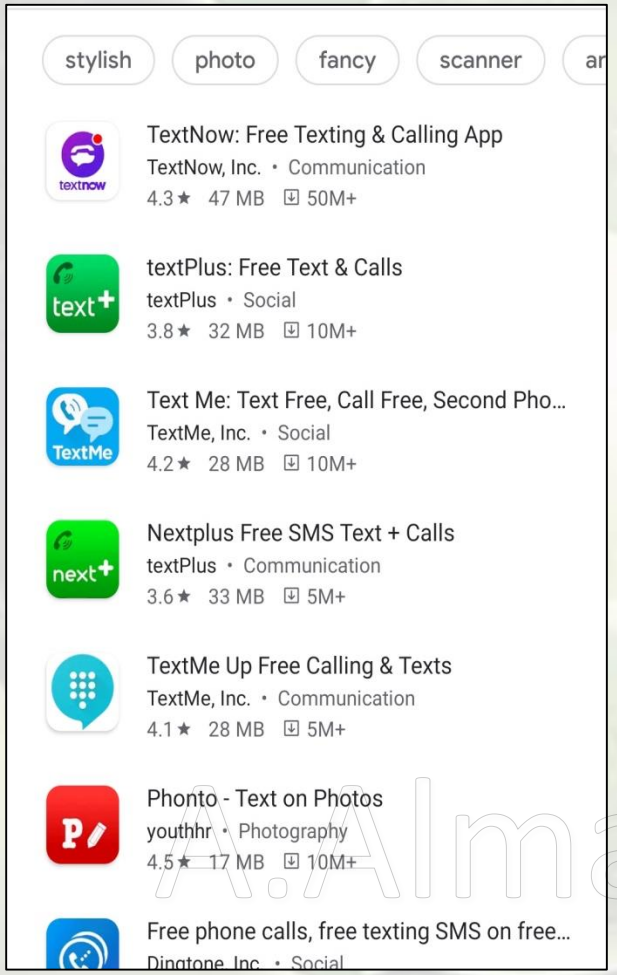
One-Size-Fits-All !!

Key Issues of Current Power-Saving Approaches

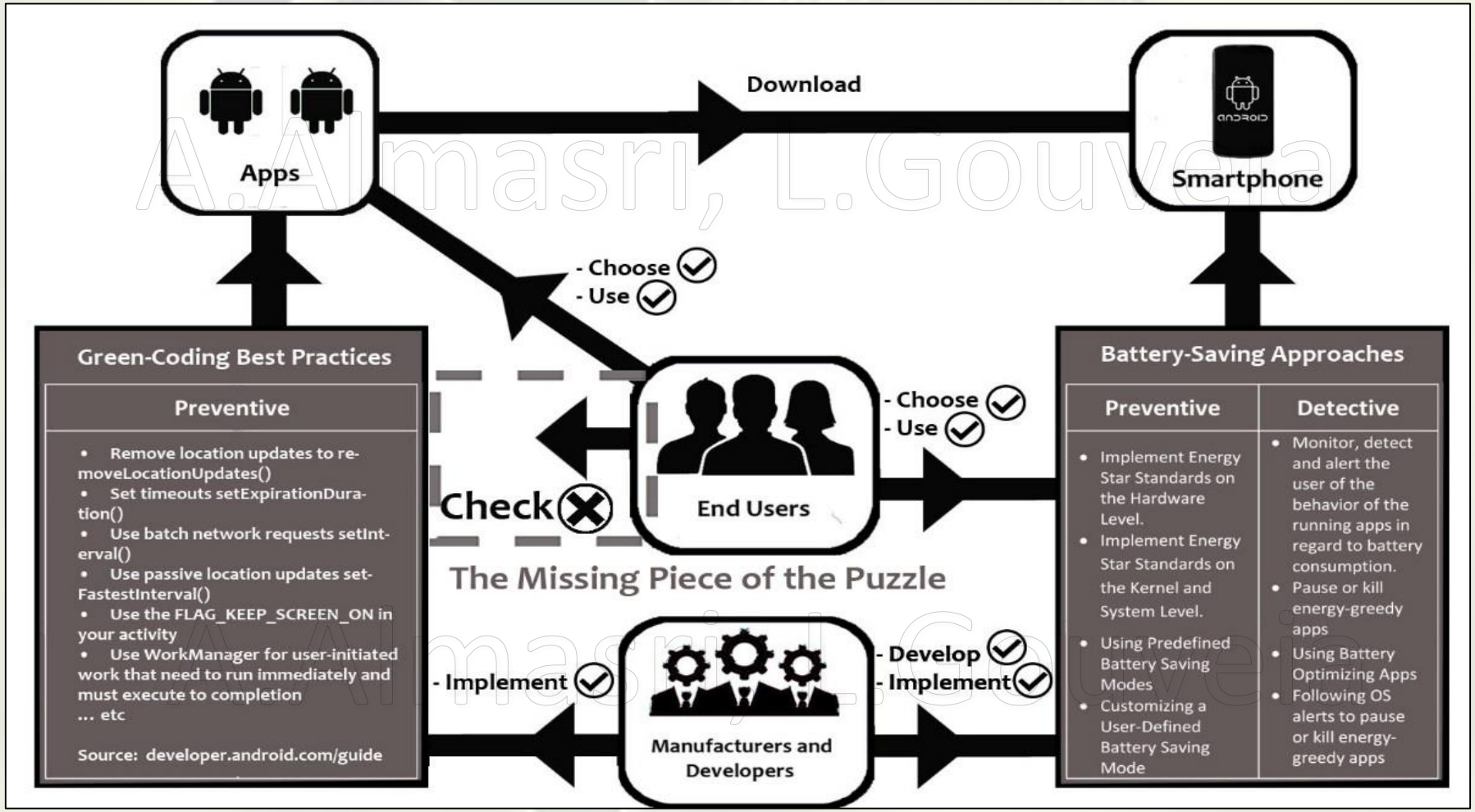
Key Issue of “Sacrifice” Approach :

Depriving Users from Technology !!

The Key Difference



The Missing Piece of the Puzzle



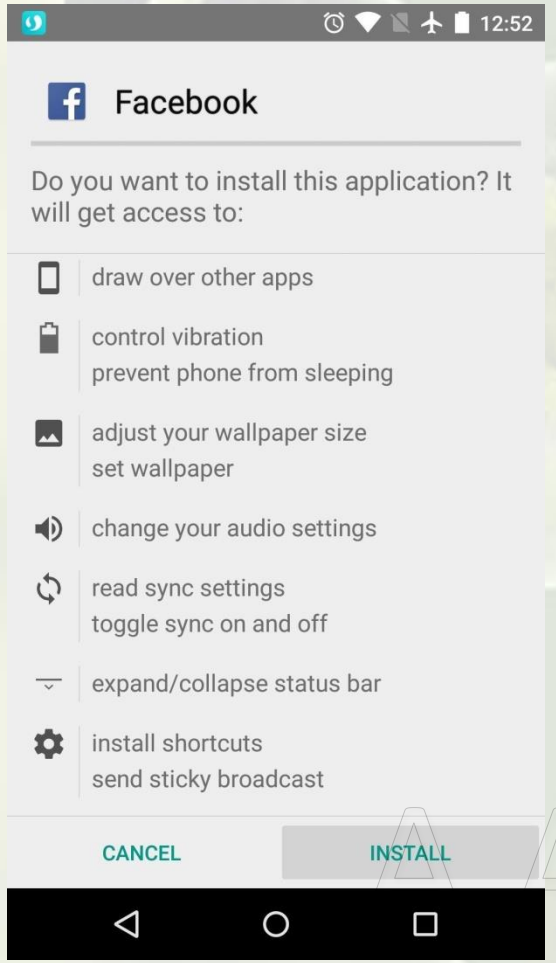
Proposed Solution

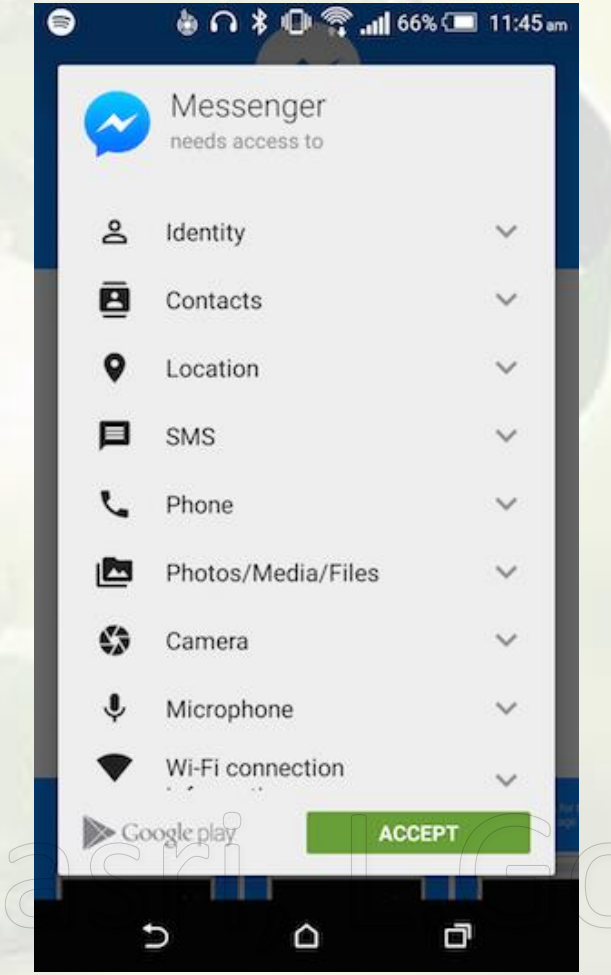
Rating Google-Play Apps' Energy Consumption on Android Smartphones

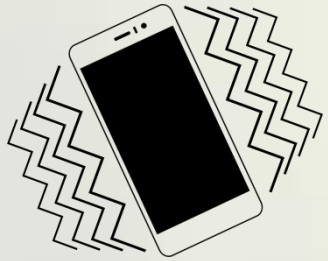
Permissions-Based Rating (Basic)

Technical Comparison Rating (Advanced)

"Permissions" in the area of power consumption (Basic)







Vibration Motor

Screen Light

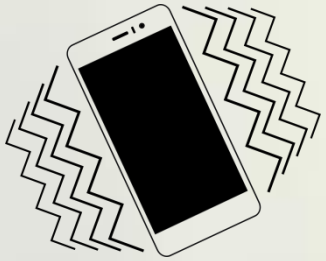
Wi-Fi Radio

Flash Light

Speaker

GPS Radio

Cell Data Radio



Vibration Motor

Average Amount of Energy Consumption (mAh/m) = ~15

Screen Light

Average Amount of Energy Consumption (mAh/m) = ~ 16

Wi-Fi Radio

Average Amount of Energy Consumption (mAh/m) = ~ 12

Flash Light

Average Amount of Energy Consumption (mAh/m) = ~ 19

Speaker

Average Amount of Energy Consumption (mAh/m) = ~ 9

GPS Radio

Average Amount of Energy Consumption (mAh/m) = ~ 25

Cell Data Radio

Average Amount of Energy Consumption (mAh/m) = ~ 17

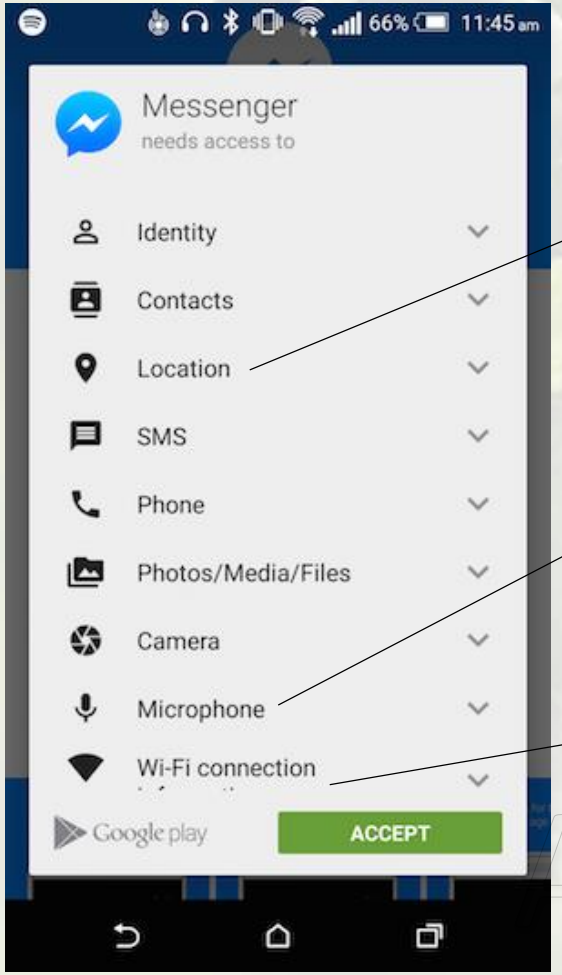
Proposed Solution (Cont.)

Rating Google-Play Apps' Energy Consumption on Android Smartphones

Smartphone Component	Capacity of the phone Battery before fully activating the component (mAh)	Capacity of the phone Battery after fully activating the component alone for a period of 60 seconds (mAh)	Average Amount of Energy Consumption measured (mAh/m)
GPS	2600	~ 2575	~ 25
Flash Light	2600	~ 2581	~ 19
Cellular Radio	2600 (Fully Charged)	~ 2583	~ 17
Cameras	2600	~ 2583	~ 17
Screen	2600	~ 2584	~ 16
Wi-Fi Radio	2600	~ 2588	~ 12
Bluetooth Radio	2600	~ 2590	~ 10

Google Play Power Consuming Applications Permissions	
• access Bluetooth settings • allow Wi-Fi Multicast reception • Broadcast data messages to apps • change network connectivity • change system display settings • change your audio settings • change/intercept network settings and traffic • connect and disconnect from Wi-Fi • control flashlight • control vibration • directly call phone numbers • download files without notification • full network access • make app always run	• modify phone state • modify secure system settings • modify system settings • pair with Bluetooth devices • precise (GPS) location • prevent phone from sleeping • read your social stream • record audio • run at startup • send sticky broadcast • take pictures and videos • toggle sync on and off • view Wi-Fi connections • write to your social stream

"Permissions" in the area of power consumption (Basic)



Average Amount of Energy Consumption (mAh/m) = **~ 25**

+

Average Amount of Energy Consumption (mAh/m) = **~ 9**

+

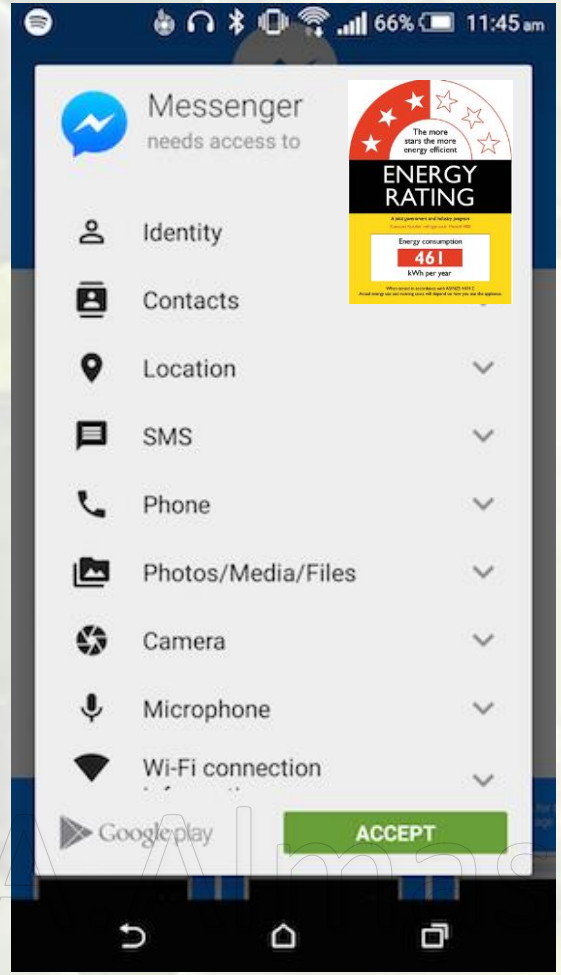
Average Amount of Energy Consumption (mAh/m) = **~ 12**

Rating Google-Play Apps' Energy Consumption on Android Smartphones

Power Consuming Applications Permissions	Amount of Energy Consumption of each Used Component	Permission Average Energy Consumption Amount per minute	Permission Star Rating out of Six Stars (~ 1 to ~ 30 mAh)
access Bluetooth settings	Bluetooth Radio (~ 10 mAh)	~ 10 mAh	★★
Broadcast data messages to apps	Wi-Fi Radio (~ 12 mAh) Cellular Radio (~ 17 mAh)	~ 15 mAh	★★★
connect and disconnect from Wi-Fi	Wi-Fi Radio (~ 12 mAh)	~ 12 mAh	★★
precise (GPS) location	GPS (~ 25 mAh)	~ 25 mAh	★★★★★

Category	App	Needed Power Consuming Permissions for the App	Permissions consumption rate	Average Energy Consumption Amount per minute	Application Star Rating out of Six Stars (~ 1 to ~ 30 mAh)
Entertainment	4shared	full network access	~ 16 mAh	~ 18 mAh	★★★★
		send sticky broadcast	~ 16 mAh		
		prevent phone from sleeping	~ 18 mAh		
		run at startup	~ 20 mAh		

"Permissions" in the area of power consumption (Basic)



green computing

Technical Comparison Rating (Advanced)

Technical Comparison Rating (Advanced)


```
package imrankst1221.website.`in`.webview

import ...

class MainActivity : Activity() {
    private lateinit var mContext: Context
    internal var mLoaded = false

    // set your custom url here
    internal var URL = "https://www.infixsoft.com/"

    //for attach files
    private var mCameraPhotoPath: String? = null
    private var mFilePathCallback: ValueCallback<Array<Uri>>? = null
    internal var doubleBackToExitPressedOnce = false


    //AdView adView;
    private lateinit var btnTryAgain: Button
    private lateinit var mWebView: WebView
    private lateinit var prgs: ProgressBar
    private var viewSplash: View? = null
    private lateinit var layoutSplash: RelativeLayout
    private latein
    private latein  var layoutNoInternet: RelativeLayou


    @SuppressLint(
    override fun o
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        AppCompatDelegate.setCompatVectorFromResourcesEnabled(true)

        mContext = this
        mWebView = findViewById<View>(R.id.webview) as WebView
```


```
package imrankst1221.website.`in`.webview

import ...

class MainActivity : Activity() {
    private lateinit var mContext: Context
    internal var mLoaded = false

    // set your custom url here
    internal var URL = "https://www.infixsoft.com/"

    //for attach files
    private var mCameraPhotoPath: String? = null
    private var mFilePathCallback: ValueCallback<Array<Uri>>? = null
    internal var doubleBackToExitPressedOnce = false

    AppCompatDelegate.setCompatVectorFromResourcesEnabled(true)

    mContext = this
    mWebView = findViewById<View>(R.id.webview) as WebView
```

Energy-Aware Refactoring For Apps

Average Amount of Energy Consumption for running the code is = ~895 mW

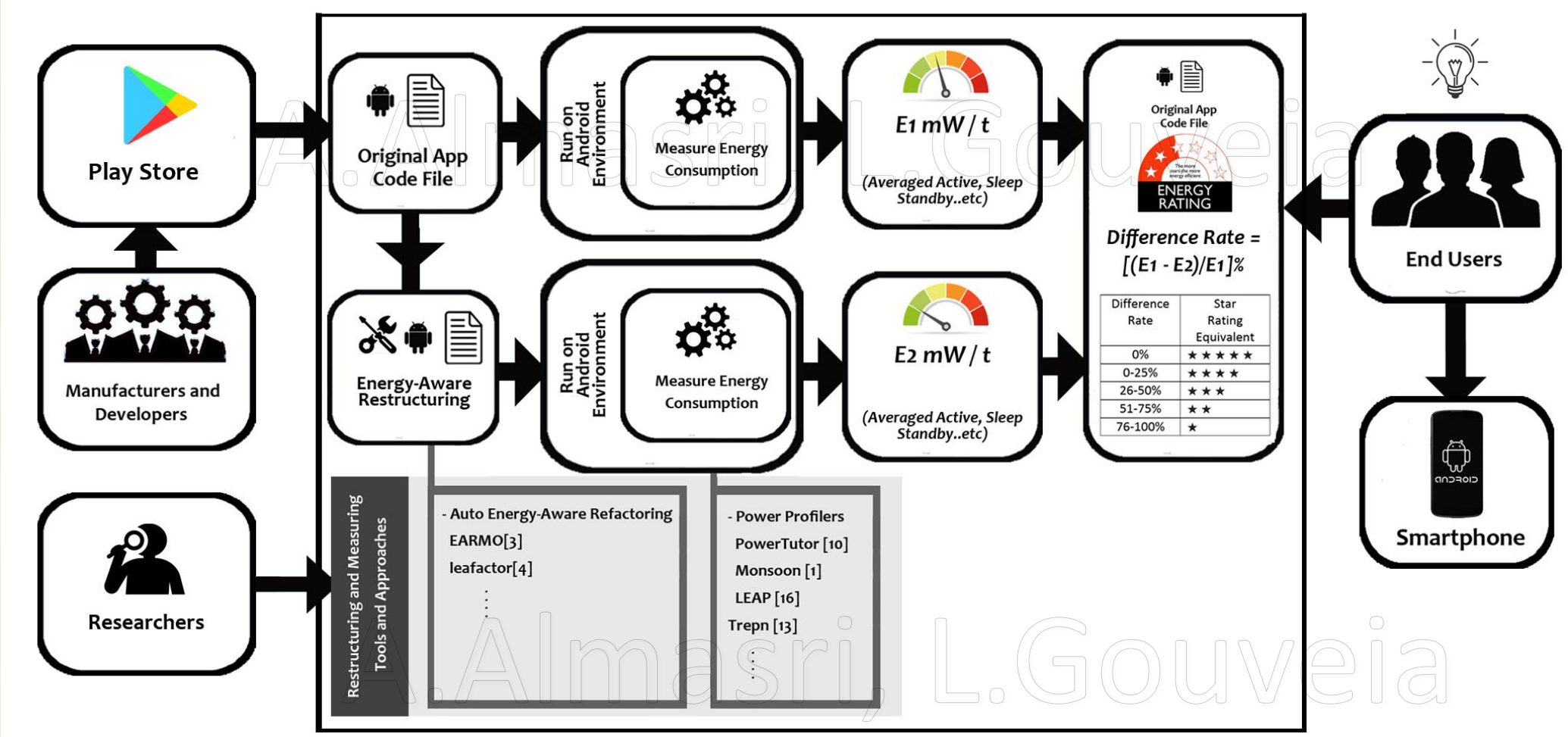
```
package imrankst1221.website.`in`.webview

import ...

class MainActivity : Activity() {
    private lateinit var mContext: Context
    internal var mLoaded = false

    // set your custom url here
    internal var URL = "https://www.infixsoft.com/"

    //for attach files
    private var mCameraPhotoPath: String? = null
    private var mFilePathCallback: ValueCallback<Array<Uri>>? = null
    internal var doubleBackToExitPressedOnce = false


    //AdView adView;
    private lateinit var btnTryAgain: Button
    private lateinit var mWebView: WebView
    private lateinit var prgs: ProgressBar
    private var viewSplash: View? = null
    private lateinit var layoutSplash: RelativeLayout
    private lateinit var layoutWebview: RelativeLayout
    private lateinit var layoutNoInternet: RelativeLayout


    @SuppressLint( ...value: "SetJavaScriptEnabled")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        AppCompatDelegate.setCompatVectorFromResourcesEnabled(true)

        mContext = this
        mWebView = findViewById<View>(R.id.webview) as WebView
```

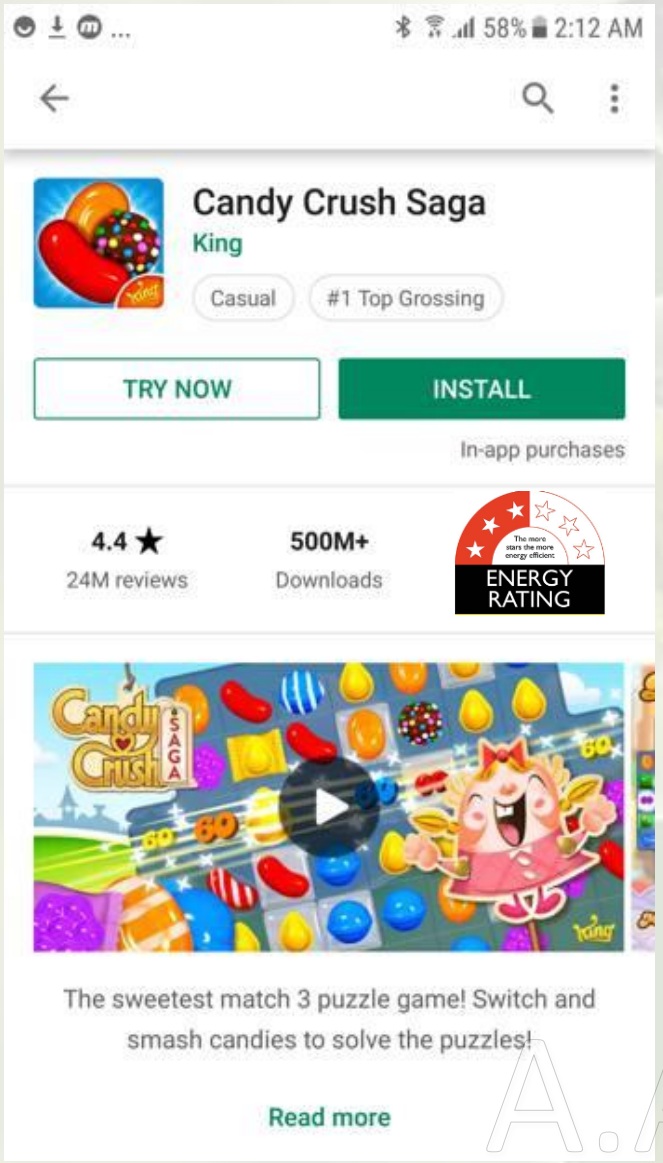
```
package imrankst1221.website.`in`.webview

import ...

class MainActivity : Activity() {
    private lateinit var mContext: Context
    internal var mLoaded = false

    // set your custom url here
    internal var URL = "https://www.infixsoft.com/"

    //for attach files
    private var mCameraPhotoPath: String? = null
    private var mFilePathCallback: ValueCallback<Array<Uri>>? = null
    internal var doubleBackToExitPressedOnce = false

    AppCompatDelegate.setCompatVectorFromResourcesEnabled(true)

    mContext = this
    mWebView = findViewById<View>(R.id.webview) as WebView
```

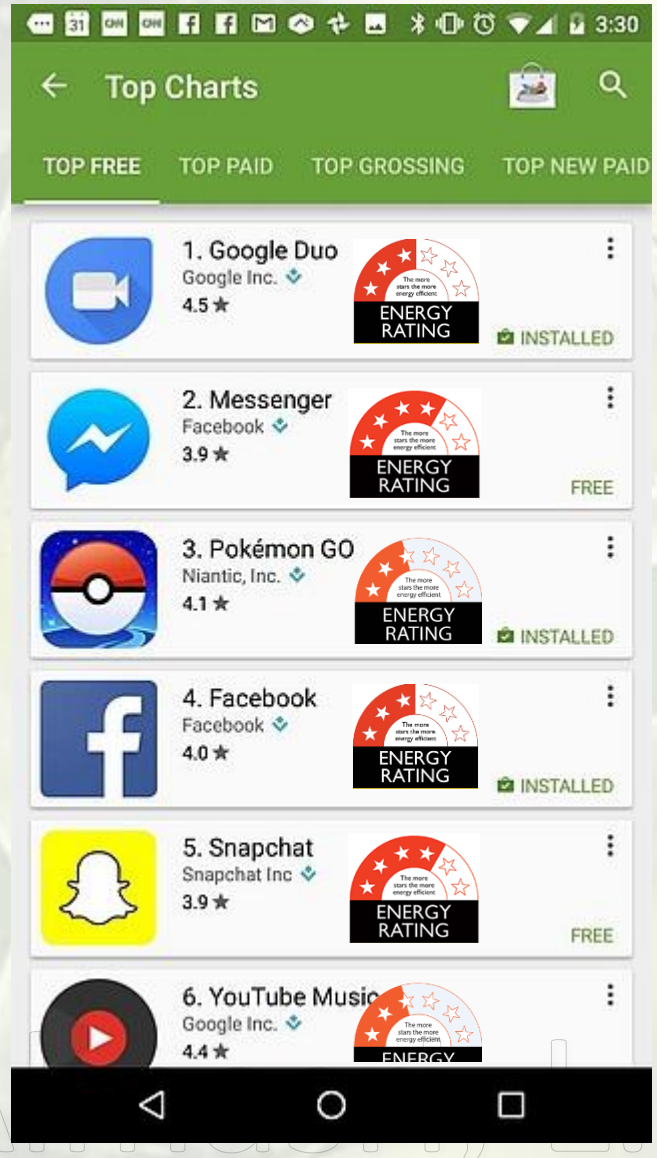
Average Amount of Energy Consumption for running the code is = ~795 mW

green computing

Technical Comparison Rating (Advanced)







Thank You

Obrigado

شكراً لكم