



ecoCode

MOBILE

*How **green** is your app?*

Olivier Le Goaër



Olivier Le Goaër

# Qui suis-je ?

## Enseignant-Chercheur en Informatique

➡ Collège STEE (Sciences & Technologies pour l'Energie et l'Environnement)

## Recherche en Software Engineering

➡ Comment construire des logiciels + verts ?

## Responsable du module Android en Master 2

➡ Cours en ligne sur SlideShare (130k vues)





# Climat : trajectoire +2°C

**l'industrie du numérique mobile doit faire sa part**

# Ébriété numérique mobile

## Mobile app developers

5,9

millions  
Android devs



2,8

millions  
iOS devs



## Available apps

2,56

millions  
Google Play Store



1,85

millions  
iOS Apple Store



## Annual app downloads

109

billiards  
Google Play Store



38

billiards  
iOS Apple Store



## SmartPhone users

3,8

billiards  
72% Android  
27% iOS  
1% autres

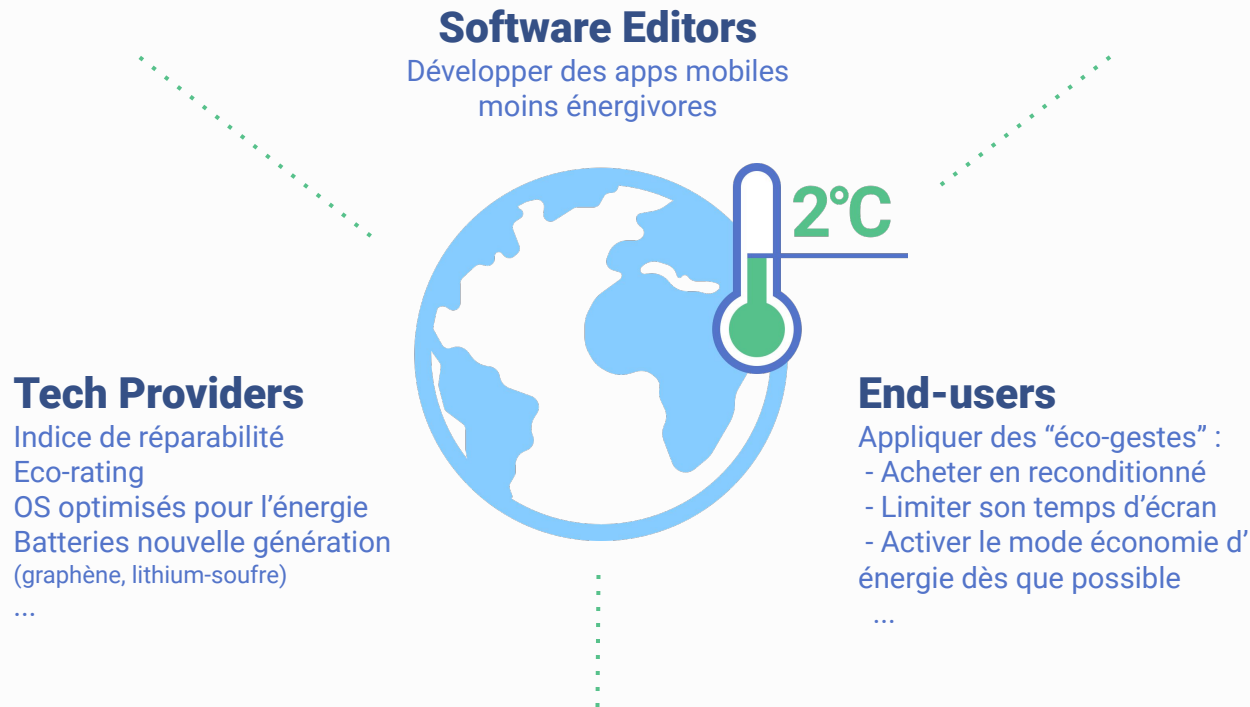
## Behavior

4,2

heures / jour



# Décarboner le numérique mobile



# Ordres de grandeur

## éco-conception MATÉRIELLE

considère le cycle de vie des smartphones/tablettes, de la fabrication à la gestion des déchets et au recyclage



**PRODUCTION**

**90%**

de la consommation d'énergie  
(et son corollaire en GES)

**UTILISATION**

**10%**

de la consommation d'énergie  
(et son corollaire en GES)

## éco-conception LOGICIELLE

vises essentiellement à réduire la consommation d'énergie des apps mobiles pendant leur phase d'usage



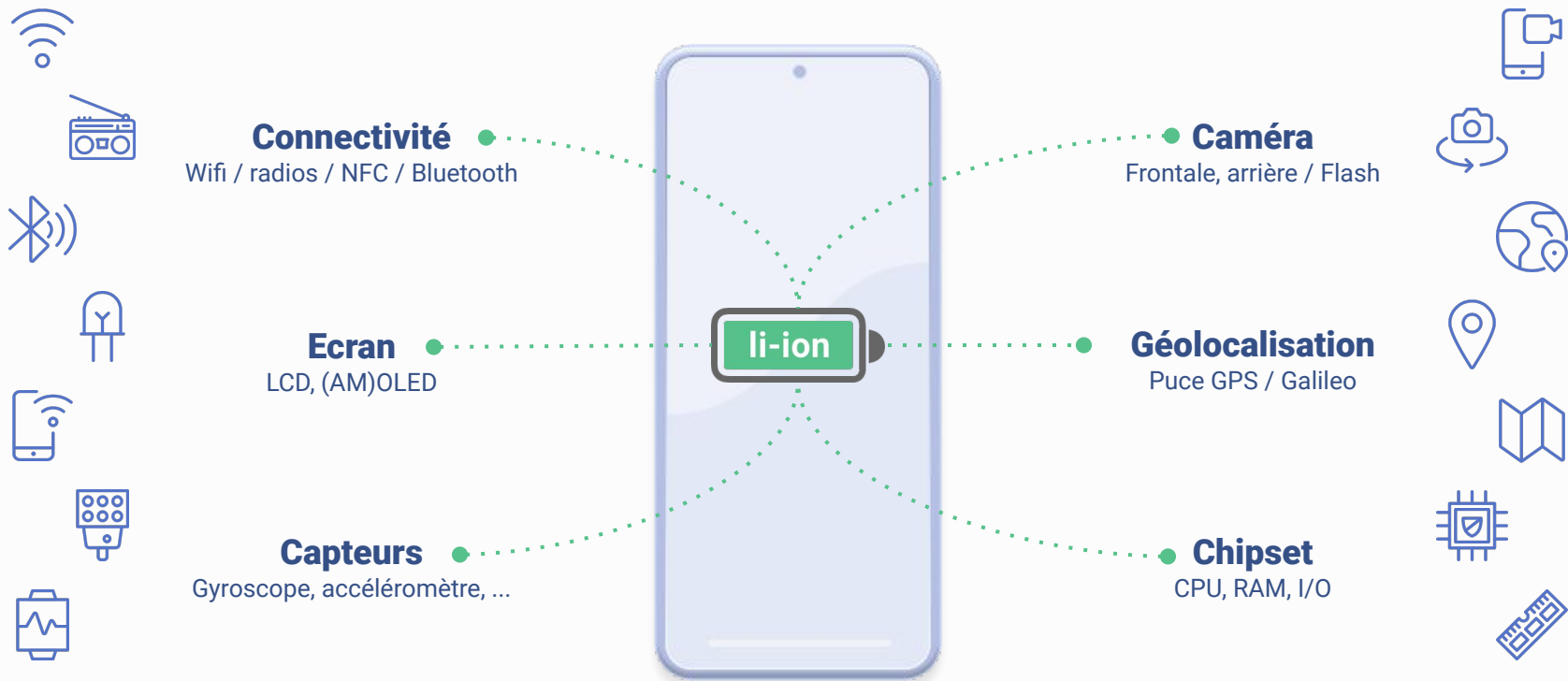
## Energie en phase d'usage

Considérons un smartPhone moyenne gamme doté d'une batterie de **3,6V** et d'une capacité de **3000mAh**, qui consomme toute sa batterie chaque jour. Sa consommation d'énergie journalière est donc de  $3\text{Ah} \times 3,6\text{V} = \mathbf{10\text{Wh}}$ , et au bout d'un an, de **3,65kWh**.

Avec **3,8 milliards** de smartPhones actifs dans le monde, la consommation annuelle atteint **15 Térawatt-heure**.

- A peine moins que la consommation annuelle en électricité d'un pays comme l'Islande (18 TWh en 2020)
- La facture énergétique va fatalement s'accroître avec 5 milliards de smartphones prévus pour 2025...

# Mais qu'est ce qui vide la batterie ?





# De l'intérêt des apps “eco-friendly”

Une app mobile qui gaspille de précieux micro-watts :

- 1 réduit la **durée de vie** de l'appareil car la batterie a un nombre de cycles charge/décharge limité
- 2 a un **effet cumulatif** important, à chaque exécution et sur chaque appareil où l'app est installée
- 3 peut créer des utilisateurs **mécontents** qui notent négativement l'app sur les magasins

# Eco-conception logicielle

Concevoir des apps en gardant à l'esprit qu'elles ont une empreinte écologique.  
**Former la nouvelle génération** de développeurs à l'éco-conception (Cf. loi REEN).

*“ Les mesures relatives à l'écoconception des services numériques sont très faibles aujourd'hui. Quand on est formé au développement, on est peu sensibilisé à « l'écologie du code » ; il est nécessaire d'avancer sur ce sujet-là. ”*

Cédric O, secrétaire d'État chargé de la transition numérique et des communications électroniques (décembre 2020)

# L'éco-conception est un tout...

Une app mobile c'est un programme **côté client** uniquement

Cela ne dit rien sur l'efficiencia du programme **côté serveur**, ni de la consommation électrique des **infrastructures réseaux** traversées



*Même si l'app Netflix (Android/iOS) est éco-conçue,  
cela ne signifie pas que le service numérique Netflix l'est.*



# Comment agir ?

sans travailler chez Google ou Apple

# Les leviers d'action

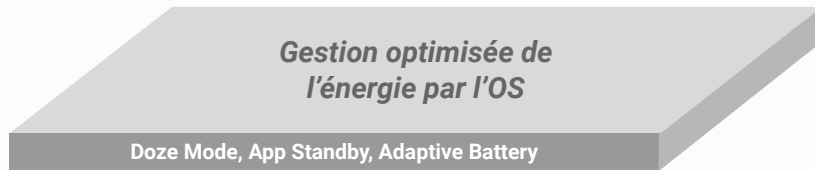
- **Dégraisser les fonctionnalités** des apps est sans conteste le gisement d'économie le plus fort, mais exige une étude **ad-hoc**

➡ *Mesurer, comprendre, arbitrer*

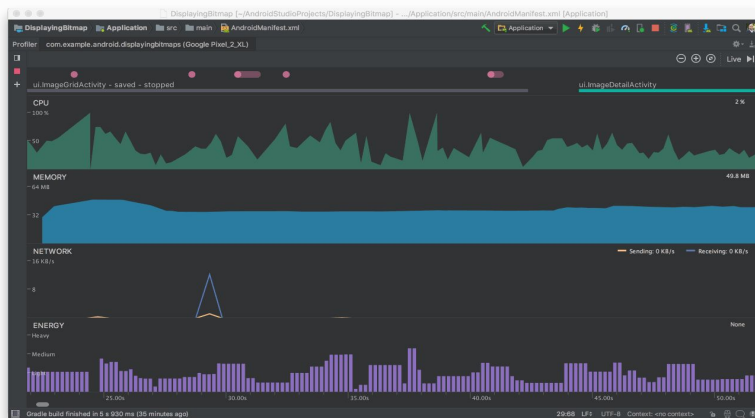
- **Améliorer la qualité du code** des apps a un impact plus modeste, mais est **sysématisable** à l'ensemble de la production logicielle mondiale

➡ *Analyse et correction automatique*

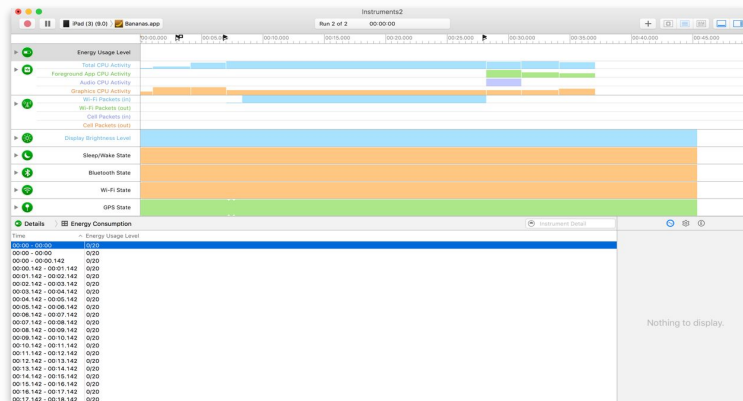
# Agissez là où vous le pouvez



# Trouver les racines du mal n'est pas simple



Android Energy Profiler (Android Studio)



iOS Energy Profiler (Xcode)

# Code smells ?

*“ Code Smells are patterns of code that suggest there might be a problem, that there might be a better way of writing the code or that more design perhaps should go into it. ”*

- ➡ Concept popularisé via le **Clean Code**
- ➡ **N'est pas un bug** car n'empêche pas le programme de fonctionner
- ➡ Les bad smells contribuent à la **dette technique**



# Energy code smells pour Android

Pourquoi ne pas appliquer ce concept à l'efficacité énergétique ? (*a fortiori* sur des terminaux limités par leur batterie)

- ➡ Détecter les structures de code “**pas terribles**” pour la batterie
- ➡ Les smells se **nichent potentiellement partout** dans un projet Android : code source, manifest, gradle, ressources organisées (layouts, images, ...)
- ➡ Évaluer le temps nécessaire pour y remédier (**rembourser la dette technique**)
- ➡ Les **corriger** automatiquement si possible

## 8 Bonnes pratiques

### # OPTIMIZED API

Check battery-efficient APIs that have been specifically designed to substitute regular APIs

### # LEAKAGE

Make sure that an acquired resource is always released, to avoid unnecessary battery drain

### # BOTTLENECK

Avoid accumulation of data or operations that will require an energy peak to be processed

### # SOBRIETY

Make reasonable accommodations between user experience and more energy efficient variants

### # IDLENESS

When the app enters in a idle state, reduce the workload accordingly

### # POWER

Adapt the app's behavior to the battery status helps prolong the battery life

### # BATCH

Grouping individually costly operations allows to save energy globally

### # RELEASE

Favour the compile-time tasks that decrease the energy footprint of the deployment of the app

# 40 Energy Code Smells

<http://olegoaer.perso.univ-pau.fr/android-energy-smells/>

## # OPTIMIZED API (2)

*Fused Location, Bluetooth Low-Energy*

## # LEAKAGE (3)

*Media Leak, Sensor Leak, Everlasting Service*

## # BOTTLENECK (4)

*Internet In The Loop, Wifi Multicast Lock, Uncompressed Data Transmission, Uncached Data Reception*

## # SOBRIETY (10)

*Dark UI, Day Night Mode, Brightness Override, Thrifty Geoloc, Thrifty BLE, Thrifty Motion Sensor, Thrifty Notification, Vibration-free, Torch-free, High Frame Rate*

## # IDLENESS (6)

*Keep Screen On, Keep CPU On, Durable Wake Lock, Rigid Alarm, Continuous Rendering, Keep Voice Awake*

## # POWER (4)

*Ignore Battery Optimizations, Companion in background, Charge Awareness, Save Mode Awareness*

## # BATCH (3)

*Service@Boot-time, Sensor Coalesce, Job Coalesce*

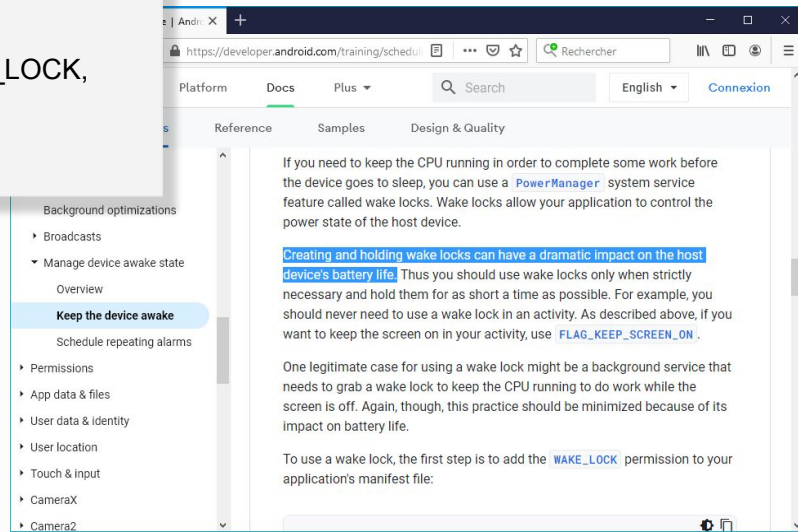
## # RELEASE (8)

*Supported Version Range, Same dependencies, Duplicate dependencies, Fat app, Clear Cache, Disable Obfuscation, Shrink Resources, Convert to WebP*

## Idleness

# Keep CPU On

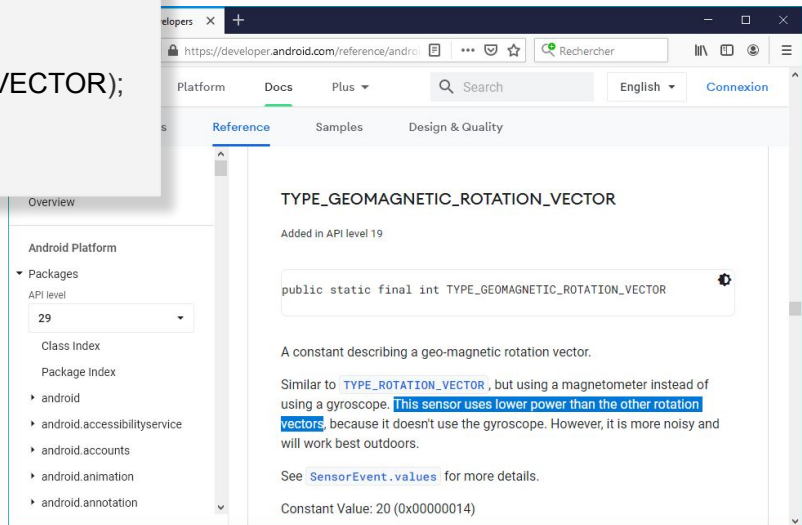
```
PowerManager powerManager = (PowerManager)
getSystemService(PowerManager.SERVICE);
WakeLock wakeLock =
powerManager.newWakeLock(PowerManager.PARTIAL_WAKE_LOCK,
    "MyApp::MyWakelockTag");
wakeLock.acquire();
```



## Sobriety

# Thrifty Motion Sensor

```
SensorManager mSensorManager = (SensorManager)
getSystemService(SENSOR_SERVICE);
// Accelerometer, Magnetometer, AND (when present) Gyroscope
mRotationSensor =
mSensorManager.getDefaultSensor(Sensor.TYPE_ROTATION_VECTOR);
mSensorManager.registerListener(this, mRotationSensor,
SENSOR_DELAY);
```







# Outils pour le développeur

Computer-Aided Software Engineering (CASE)

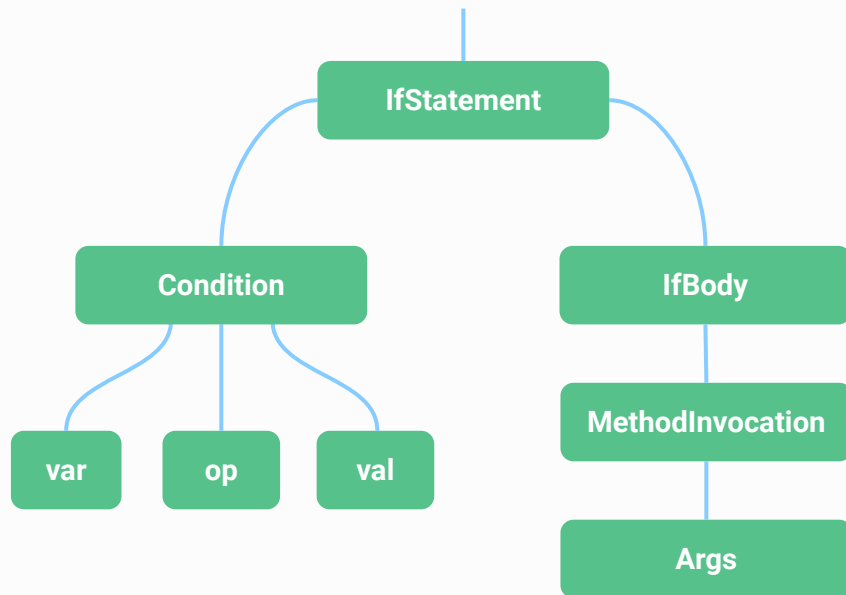
# Analyseurs de code pour Android

Embold		
Android Lint		
Detekt		
<b>SonarQube</b>		
SonarLint (version IntelliJ)		
Semgrep		
Codacy		
...		

# Abstract Syntax Tree (AST)

```
WakeLock wakeLock =  
powerManager.newWakeLock(PowerManager.  
PARTIAL_WAKE_LOCK,  
"MyApp::MyWakelockTag");
```

```
if (wakeLock != null) {  
    wakeLock.acquire();  
}
```



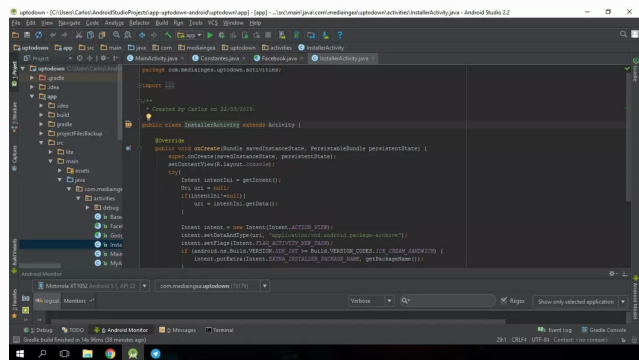


ecoCode

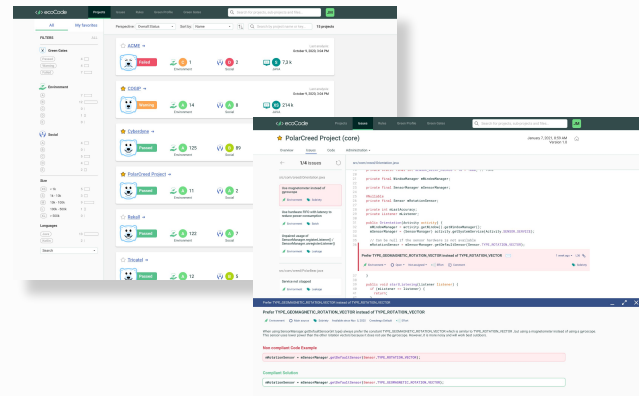
# Un plugin SonarQube

- SonarQube est l'outil de qualité de code le plus **populaire** du marché
- **Analyse statique** de code : le programme n'est jamais exécuté !
- Totalement **indépendant** de la taille, de la catégorie et des fonctionnalités
- Un **eco-score** (de A à E) peut être attribué
- Verdissement automatisé via un **pipeline CI/CD**
- S'adresse en priorité aux **chefs de projets et lead devs**

# ecoCode Software as a Service



Projet local avec Android Studio



<https://sonarqube.ecocode.io>

# ecoCode

## Base de règles

The screenshot shows the ecoCode web application interface. The top navigation bar includes the ecoCode logo, tabs for Projects, Issues, Rules (active), Quality Profiles, and Quality Gates, a search bar, and a user profile icon (SS). The left sidebar contains filters for Language, Type (CODE SMELL), Bug, Vulnerability, Code Smell (19), Security Hotspot, Tag, Repository, Default Severity, Status, Security Category, Available Since, Template, and Quality Profile (ECOCOD...). The main content area displays a list of 19 rules, each with a status icon, name, language, and tags.

Rule Name	Language	Tags
Bottleneck: Internet In The Loop	Java	Code Smell, ecocode, environment, optimized-api
Idleness: Keep Screen On (addFlags)	Java	Code Smell, ecocode, environment, idleness
Idleness: Keep Screen On (setFlags)	Java	Code Smell, ecocode, environment, idleness
Leakage: Camera Leak	Java	Code Smell, ecocode, environment, leakage
Optimized API: Fused Location	Java	Code Smell, ecocode, environment, optimized-api
Idleness: Keep Cpu On	Java	Code Smell, ecocode, environment, idleness
Idleness: Durable Wake Lock	Java	Code Smell, ecocode, environment, idleness
Optimized API: Bluetooth Low-Energy	Java	Code Smell, ecocode, environment, optimized-api
Sobriety: Thrifty Motion Sensor	Java	Code Smell, ecocode, environment, sobriety
Leakage: Location Leak	Java	Code Smell, ecocode, environment, leakage
Sobriety: Brightness Override	Java	Code Smell, ecocode, environment, optimized-api
Leakage: SensorManager Leak	Java	Code Smell, ecocode, environment, leakage
Bottleneck: Wifi Multicast Lock	Java	Code Smell, bottleneck, ecocode, environment
Leakage: Media Leak (MediaRecorder)	Java	Code Smell, ecocode, environment, leakage

# ecoCode

## Projets analysés

ecoCode

Projects

Issues

Rules

Quality Profiles

Quality Gates

?

Search for projects...

+

SS

My Favorites

All

Filters

Quality Gate

Passed 0 |

Failed 2

Reliability ( 🐛 Bugs )

A 2 |

B 0 |

C 0 |

D 0 |

E 0 |

Security ( 🔒 Vulnerabilities )

A 2 |

B 0 |

C 0 |

D 0 |

E 0 |

Maintainability ( 🐛 Code Smells )

A 1

Perspective: Overall Status

Sort by: Name

Search by project name or key

2 projects

🏠

☆ Spotify

Last analysis: yesterday

🐼

0 A

0 A

79 C

0.0%

0.0%

9k S

FAILED

🐛 Bugs

🔒 Vulnerabilities

🐛 Code Smells

Coverage

Duplications

Java, Xml

☆ QuelPrixImmo

Last analysis: 5 hours ago

🐼

0 A

0 A

8 A

0.0%

2.7%

2.2k S

FAILED

🐛 Bugs

🔒 Vulnerabilities

🐛 Code Smells

Coverage

Duplications

Java, Xml

2 of 2 shown

This application is based on SonarQube™ but is not an official version provided by SonarSource SA.

Community Edition - Version 8.6.1 (build 3632) - LGPL v3 - Community - Documentation - Plugins - Web API - About

ecoCode

# Qualité verte d'un projet

ecoCode

Projects Issues Rules Quality Profiles Quality Gates

Search for projects...

SS

QuelPrixImmo


master

June 23, 2021, 1:09 PM Version unspecified

Project Information

Overview Issues Security Hotspots Measures Code Activity

QUALITY GATE STATUS



FAILED

On Overall Code

8 Issues is greater than 4

MEASURES

New Code

Since June 23, 2021  
Started 6 hours ago

Overall Code

0 Bugs

Reliability A

0 Vulnerabilities

Security A

0 Security Hotspots

Reviewed

Security Review A

2h 40min Debt

8 Code Smells

Maintainability A

# ecoCode

## Problèmes détectés

The screenshot displays the ecoCode web application interface. The top navigation bar includes 'ecoCode', 'Projects', 'Issues', 'Rules', 'Quality Profiles', and 'Quality Gates'. A search bar is present on the right. The main content area shows the 'Issues' tab for the 'QuelPrixImmo' project. On the left, a sidebar lists various issue types and rules. The main panel displays a list of detected issues, each with a description, severity, status, and effort. The issues are categorized by type (Bug, Vulnerability, Code Smell) and severity (Major, Open, Not assigned). The issues are also grouped by rule, with each rule having a unique identifier (e.g., L15, L528, L529, L528, L529, L528). The issues are sorted by effort, with the most critical issues at the top.

**ecoCode** Projects Issues Rules Quality Profiles Quality Gates

QuelPrixImmo master

June 23, 2021, 1:09 PM Version unspecified

Overview Issues Security Hotspots Measures Code Activity

Project Information

**Type**

- Bug 0
- Vulnerability 0
- Code Smell 8

**Severity**

**Scope**

**Resolution**

**Status**

**Security Category**

**Creation Date**

**Language**

**Rule**

Search for rules...

(Java) Sobriety: Thrifty Geolocation 4

(Java) Leakage: Location Leak 2

(XML) Idleness: Keep Screen On 1

(XML) Sobriety: Dark UI (Theme) 1

4 shown

Bulk Change

to select issues to navigate 1 / 8 issues 2h 40min effort

src/main/AndroidManifest.xml

Using a light theme may have a significant impact on energy consumption on (AM)OLED screens. See Rule 2 days ago L15 ecocode, environment, sobriety

Code Smell Major Open Not assigned 1h effort Comment

src/main/java/fr/univpau/quelpriximmo/ResultListFragment.java

Failing to call android.location.LocationManager#removeUpdates() can drain the battery in just a few hours. See Rule 9 days ago L528 ecocode, environment, leakage

Code Smell Major Open Not assigned 20min effort Comment

Location updates should be done with a minimum time and distance interval. See Rule 9 days ago L528 ecocode, environment, optimized-api

Code Smell Major Open Not assigned 10min effort Comment

Location updates should be done with a minimum time and distance interval. See Rule 9 days ago L528 ecocode, environment, optimized-api

Code Smell Major Open Not assigned 10min effort Comment

Failing to call android.location.LocationManager#removeUpdates() can drain the battery in just a few hours. See Rule 9 days ago L529 ecocode, environment, leakage

Code Smell Major Open Not assigned 20min effort Comment

Location updates should be done with a minimum time and distance interval. See Rule 9 days ago L529 ecocode, environment, optimized-api

Code Smell Major Open Not assigned 10min effort Comment

ecoCode

# Détail d'un problème

The screenshot displays the ecoCode web application interface. At the top, there is a navigation bar with tabs for Projects, Issues, Rules, Quality Profiles, and Quality Gates. A search bar is located on the right. The main content area shows a project named 'QuelPrixImmo' with a 'master' branch. The 'Issues' tab is selected, showing a list of 3 issues. The first issue, 'Falling to call android.location.LocationManager#r...', is highlighted. The details of this issue are shown on the right, including the code snippet and the rule description. The issue is categorized as a 'Code Smell' and is marked as 'Major'. The rule is 'Sobriety: Thrifty Geolocation'.

ecoCode

Projects Issues Rules Quality Profiles Quality Gates

Search for projects...

QuelPrixImmo master

June 23, 2021, 1:09 PM Version unspecified

Overview Issues Security Hotspots Measures Code Activity

Project Information

3 / 8 issues

...pau/quelpriximmo/ResultListFragment.java

Falling to call android.location.LocationManager#r... can drain the battery in just a few hours. Code Smell

Location updates should be done with a minimum time and distance interval. Code Smell

Location updates should be done with a minimum time and distance interval.

```
@SuppressWarnings("MissingPermission")
@Override
public void onResume() {
    super.onResume();
    mLocationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, this);
}
```

Falling to call android.location.LocationManager#removeUpdates() can drain the battery in just a few hours. See Rule 9 days ago L528

Code Smell Major Open Not assigned 20min effort Comment ecocode, environment, leakage

Location updates should be done with a minimum time and distance interval. See Rule 9 days ago L528

Code Smell Major Open Not assigned 10min effort Comment ecocode, environment, optimized-api

Location updates should be done with a minimum time and distance interval. See Rule 9 days ago L528

Code Smell Major Open Not assigned 10min effort Comment ecocode, environment, optimized-api

Sobriety: Thrifty Geolocation

ecoCode-java:ESOB005

Code Smell Major ecocode, environment, optimized-api Available Since Jun 22, 2021 ecoCode (Java) Constant/Issue: 10min

With a call to `LocationManager#requestLocationUpdates(String provider, long minTime, float minDistance, LocationListener listener)`, the location provider will only send your application an update when the location has changed by at least `minDistance` meters AND at least `minTime` milliseconds have passed. So `minTime` should be the primary tool to conserve battery life, and, to a lesser extent, `minDistance`. These two must imperatively be greater than 0.

ecoCode

# Analyse SWOT

## FORCES

*Minimum Viable Product*  
Partenariat académique/industrie

## FAIBLESSES

Base de règles insuffisante  
Réglages empiriques

## OPPORTUNITÉS

Intérêt publique/politique croissant  
La réglementation évolue

## MENACES

Les *big players* arrivent  
Risque de *greenwashing*



ecoCode

# Open Source Software

En Janvier 2022, le plugin a été libéré en **open source** (excepté la GUI)

ecoCode a rejoint le dépôt du collectif Conception NUMérique Responsable (CNUMR) et **donne son nom au projet global** multi-technologies (php, python, js, android, iOS, ...)



<https://github.com/cnumr/ecoCode/>

ecoCode  
**Hackathon**



**CHALLENGE**  
<> ecoCode



2 juin 2022



3 juin 2022

# Des questions ?



(ekko, notre mascotte)