

Thread Twitter du 27/04/2020 10h43

[Elliot Alderson @fs0c131y](#)

1/ Dans le cadre du débat sur l'application de contact tracing Française [#StopCovid](#), on parle beaucoup de la limitation d'utilisation du Bluetooth sur iOS.

Il y a les mêmes limitations sur Android. Détaillons-les

2/ Il y a 2 semaines, j'avais déjà détaillé la limitation d'utilisation du Bluetooth sur iOS

<https://twitter.com/fs0c131y/status/1249633962865164289?s=20>

Elliot Alderson @ fs0c131y

1 / Techniquement, une application iOS ne peut pas utiliser Bluetooth tout le temps.

Par conséquent, une application de suivi des contacts qui utilise Bluetooth ne peut pas fonctionner sur iOS.

2 / Il y a 3 semaines, Singapour a publié une application de recherche de contacts pour combattre # Covid19. Les notes de l'application iOS sont assez mauvaises
<https://apps.apple.com/sg/app/tracetogether/id1498276074> h

3 / La première note disponible est assez claire: "Pourquoi l'utilisateur voudrait-il garder cette application ouverte et au premier plan?"

4 / Les développeurs de l'application ont répondu au commentaire avec la documentation technique et se sont référés au mode d'économie d'énergie qu'ils ont implémenté.

5 / La documentation Apple "Core Bluetooth Background Processing for IOS Apps" est assez claire: "Par défaut, la plupart des tâches courantes de Core Bluetooth [...] sont désactivées lorsque votre application est en arrière-plan ou suspendue"

https://developer.apple.com/library/archive/documentation/NetworkingInternetWeb/Conceptual/CoreBluetooth_concepts/CoreBluetoothBackgroundProcessingForIOSApps/PerformingTasksWhileYourAppIsInTheBackground.html

6 / Cela ne peut pas être plus clair: "Même si votre application prend en charge un ou les deux modes d'exécution d'arrière-plan Core Bluetooth, elle ne peut pas fonctionner indéfiniment"

7 / Pour "contourner" cette limitation, les gens de TraceTogether ont créé un "mode d'économie d'énergie"

8 / Ce n'est vraiment pas une bonne solution mais ils n'avaient pas le choix. Ils ont demandé à leurs utilisateurs de garder l'application au premier plan et de réduire la luminosité de l'écran

9 / Garder l'application au premier plan tout le temps videra votre batterie très rapidement

10 / Alors oui, une application de suivi des contacts sur iOS ne peut pas fonctionner à cause de cette limitation.

"Je voudrais faire d'autres choses sur mon téléphone pendant mes déplacements, au travail, en mangeant, etc."

11 / Oui, Apple peut probablement le corriger et supprimer cette limitation pour sa prochaine "API de suivi des contacts préservant la confidentialité". Mais l'homme, ce sera un énorme précédent.

12 / Ces dernières années, Apple et Google ont limité de plus en plus ce que les développeurs peuvent faire en arrière-plan.

"Pour améliorer l'expérience utilisateur, Android 8.0 (API niveau 26) impose des limitations sur ce que les applications peuvent faire lorsqu'elles s'exécutent en arrière-plan"

<https://developer.android.com/about/versions/oreo/background>

13 / "Pour améliorer l'expérience utilisateur": LOL

Les services d'arrière-plan sont une grande source d'abus sur Android. Les auteurs de logiciels malveillants l'utilisent tout le temps. C'est pourquoi ils "imposent des limitations sur ce que les applications peuvent faire lors de leur exécution en arrière-plan"

14 / L'API Apple / Google présente un autre inconvénient. Si un gouvernement souhaite publier une application de suivi des contacts fonctionnelle, il est obligé d'utiliser cette API. En tant que politicien, lorsque vous passez votre temps à parler de la souveraineté numérique, de la gravité des GAFAs, c'est un problème.

15 / Alors oui, ce n'est pas magique.

16 / Les applications de suivi des contacts avec Bluetooth **sont un non-sens technique**. J'ai écrit un article de 5 pages sur les raisons pour lesquelles cela ne peut pas fonctionner techniquement. Il sera publié prochainement.

3/ Lors du lancement d'Android 8, Google a introduit une nouvelle limitation sur l'exécution de services en arrière plan <https://developer.android.com/about/versions/oreo/android-8.0-changes>

Changes for all apps

These behavior changes apply to all apps when they run on the Android 8.0 (API level 26) platform, regardless of the API level that they target. All developers should review these changes and modify their apps to support them properly, where applicable to the app.

Background execution limits

As one of the changes that Android 8.0 (API level 26) introduces to improve battery life, when your app enters the [cached](#) state, with no active [components](#), the system releases any wakelocks that the app holds.

4/ Cette limitation étant un énorme changement, Google a même créé une page dédiée: "Background Execution Limits" <https://developer.android.com/about/versions/oreo/background#services>

watching video. To improve the user experience, Android 8.0 (API level 26) imposes limitations on what apps can do while running in the background. This document describes the changes to the operating system, and how you can update your app to work well under the new limitations.

5/ Il y a 2 type de services sur Android:

- Foreground service
- Background service

<https://developer.android.com/guide/components/services>

Foreground

A foreground service performs some operation that is noticeable to the user. For example, an audio app would use a foreground service to play an audio track. Foreground services must display a [Notification](#). Foreground services continue running even when the user isn't interacting with the app.

★ **Note:** The [WorkManager](#) API offers a flexible way of scheduling tasks, and is able to [run these jobs as foreground services](#) if needed. In many cases, using WorkManager is preferable to using foreground services directly.

Background

A background service performs an operation that isn't directly noticed by the user. For example, if an app used a service to compact its storage, that would usually be a background service.

★ **Note:** If your app targets API level 26 or higher, the system imposes [restrictions on running background services](#) when the app itself isn't in the foreground. In most situations, for example, you shouldn't [access location information from the background](#). Instead, [schedule tasks using WorkManager](#).

6/ Afin de gérer au mieux la mémoire de votre téléphone Android distingue les apps en premier plan et en arrière plan. La définition est très claire

<https://developer.android.com/about/versions/oreo/background#services>

pertains to memory management, but in the foreground as pertains to its ability to launch services.) An app is considered to be in the foreground if any of the following is true:

- It has a visible activity, whether the activity is started or paused.
- It has a foreground service.
- Another foreground app is connected to the app, either by binding to one of its services or by making use of one of its content providers. For example, the app is in the foreground if another app binds to its:
 - [IME](#)
 - Wallpaper service
 - Notification listener
 - Voice or text service

If none of those conditions is true, the app is considered to be in the background.

7/ Alors que se passera t'il si [#StopCovid](#) utilise un service en arrière plan?

Au moment où l'app passera en arrière plan, l'app aura une fenêtre de quelques minutes pour créer et utiliser ses services. Ensuite, le système tuera TOUS ses services. h

While an app is in the foreground, it can create and run both foreground and background services freely. When an app goes into the background, it has a window of several minutes in which it is still allowed to create and use services. At the end of that window, the app is considered to be *idle*. At this time, the system stops the app's background services, just as if the app had called the services' `Service.stopSelf()` methods.

8/ Ces limitations sont présentes sur les versions d'Android 8 et +. Combien de personnes possèdent cette version?

Une part importante: 60.8%

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99.8%
4.2 Jelly Bean	17	99.2%
4.3 Jelly Bean	18	98.4%
4.4 KitKat	19	98.1%
5.0 Lollipop	21	94.1%
5.1 Lollipop	22	92.3%
6.0 Marshmallow	23	84.9%
7.0 Nougat	24	73.7%
7.1 Nougat	25	66.2%
8.0 Oreo	26	60.8%
8.1 Oreo	27	53.5%
9.0 Pie	28	39.5%
10. Android 10	29	8.2%

9/ Une application de contact tracing type [#StopCovid](#) ne peut donc pas utiliser un service en arrière plan. Regardons la 2ième option: les foreground services

10/ Pour commencer un service en 1er plan à l'obligation de montrer une notification à l'utilisateur.

Combien de personnes agacées par la notification vont tuer l'app juste pour enlever la notification?

11/ Vous le voyez le gros pavé rouge avec marqué "Caution" dans la section "Running a service in the foreground"?

Et oui Google a vu venir les gros malins qui voudraient tourner leurs services en 1er plan en permanence.

Running a service in the foreground

A foreground service is a service that the user is actively aware of and isn't a candidate for the system to kill when low on memory. A foreground service must provide a notification for the status bar, which is placed under the *Ongoing* heading. This means that the notification cannot be dismissed unless the service is either stopped or removed from the foreground.

! Caution: Limit your app's use of foreground services.

You should only use a foreground service when your app needs to perform a task that is noticeable by the user even when they're not directly interacting with the app. For this reason, foreground services must show a [status bar notification](#) with a priority of [PRIORITY_LOW](#) or higher, which helps ensure that the user is aware of what your app is doing. If the action is of low enough importance that you want to use a minimum-priority notification, you probably shouldn't be using a service; instead, consider using a [scheduled job](#).

Every app that runs a service places an additional load on the system, consuming system resources. If an app tries to hide its services by using a low-priority notification, this can impair the performance of the app the user is actively interacting with. For this reason, if an app tries to run a service with a minimum-priority notification, the system calls out the app's behavior in the notification drawer's bottom section.

12/ C'est la dernière phrase ici qui nous intéresse ici. Si une app utilise abusivement d'un foreground service, Android affichera une notification à l'utilisateur en l'alertant qu'une application abuse des ressources de son téléphone. C'est bien fait, non?

Every app that runs a service places an additional load on the system, consuming system resources. If an app tries to hide its services by using a low-priority notification, this can impair the performance of the app the user is actively interacting with. For this reason, if an app tries to run a service with a minimum-priority notification, the system calls out the app's behavior in the notification drawer's bottom section.

13/ L'utilisation abusive d'un foreground service peut également avoir des impacts sur l'utilisation des autres applications

Every app that runs a service places an additional load on the system, consuming system resources. If an app tries to hide its services by using a low-priority notification, this can impair the performance of the app the user is actively interacting with. For this reason, if an app tries to run a service with a minimum-priority notification, the system calls out the app's behavior in the notification drawer's bottom section.

14/ En résumé, pour une app de contact tracing sur Android:

- impossible d'utiliser un service en arrière plan
- obligation d'utiliser un service en 1er plan
- mais notification en permanence
- et notification d'alerte par Android au bout d'un certain temps pour indiquer un abus

15/ Cela montre encore une fois l'amateurisme actuel. On gonfle les muscles dans la presse en parlant de bras de fer avec Apple alors qu'iOS ne représente que 21,1% du marché Français et que le pb est également présent sur Android.

16/ Alors pourquoi ne parle t'on que d'un bras de fer avec Apple et pas avec Google?

Même si Google était d'accord pour faire tomber ses propres limitations, ce qui ne sera jamais le cas, ils ne pourraient pas les faire tomber comme cela.

17/ Il faudrait pour cela, faire accepter cette régression à tous les OEMs, leur demander de faire une mise à jour sur tous les terminaux déployés (ce qui coute énormément cher). C'est littéralement impossible