



Crédit Mutuel
ARKEA

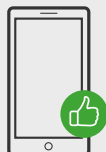


GREENSPECTOR

User Case Study



Energy savings of 50% (3 hours of autonomy)



10 MB reduction in final APK volume



Optimization of requests and sequencing of network usage

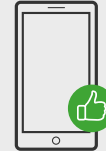
All the results are available on www.greenspector.com



| RESULTS



Energy savings of 50% (3 hours of autonomy)



10 MB reduction in final APK volume



Optimization of requests and sequencing of network usage

Crédit Mutuel ARKEA

| NEED

Arkea wanted to detect the reasons of malfunctions on their remote bank application :

- Reduce energy footprint
- Improve user experience (reduce the size of the application and increase the fluidity of the application)
- Customer loyalty
- Optimize the application to make it more functional



Activity area :
Bank



Number of employees :
78 000



Date : **2016**



| METHODOLOGY

Measurements :

- Remote bank application for individuals and professionals.

From the first energy measurements, a high consumption was detected on the connection screen of remote banking applications. Analysis showed that the animation for the login button was the cause of this overconsumption. The deletion of this animation made it possible to obtain a gain of 50% on the autonomy.

We note a long loading (up to 10 seconds) and energy consumer (continuous sollicitation of the radio cell) in use on degraded connexions. Study of network flux allows us to identify a certain numbers of unnecessary or optimizable requests. For example, ressources (images, analytics, fonts) systematically downloaded from servers, can be loaded inside the application for limiting consumption's impact and user experience. Optimisation of requests should allowed to release mobiles ressources and servers for other treatments.

Similarly, a significant memory leak was detected during the disconnection process. Indeed, linking multiple connections / disconnections had the effect of creating and stacking Android activities that were not released, thus saturating the memory.

The initial size of the APK was 40MB. The static analysis of the source code of the application made it possible to identify a potential gain of more than 10Mo on the volumetry of the latter by removing unused resources (images, dead code ...)



| DETAILED RESULTS

- Energy gains (3 hours of additional autonomy, which represents a gain of 50 %)
- 10 MB reduction in final APK volume (from 40Mo to 30Mo)
- Optimisation of requests and sequencing of network usage