

Firefox sur Apple Silicon : dans les coulisses du portage

Les futurs Mac auront-ils des écrans tactiles ? Mozilla se l'est demandé en juin dernier, à l'heure où Apple officialisait la [transition vers Arm](#).

À la mi-décembre, la fondation livrait un premier portage stable de Firefox. Elle fait désormais [le point](#) sur quelques-uns des défis qu'elle a dû relever pour en arriver là.

Le code du navigateur était déjà adapté à l'architecture Arm64. C'est à plus bas niveau qu'il a fallu travailler. Notamment sur les **interfaces entre JavaScript et C++**. Le **compilateur Rust**, en particulier, a posé des problèmes. Il a fallu attendre la mi-août pour disposer d'une version fonctionnelle.

Autre point délicat : la **gestion des DRM** pour les plates-formes de streaming vidéo. Mozilla a opté pour la même implémentation qu'avec Windows sur Arm. Elle a utilisé les versions x64 des modules de décodage, isolés et exécutés sur la couche d'émulation Rosetta 2.

Les processus qui chargent les modules en question dépendent de bibliothèques dont il a fallu récupérer les versions x64. La fourniture de Firefox sous forme de binaire universel – alliant versions Arm et Intel – a facilité les choses... au prix d'un installateur plus lourd.

Quelle *roadmap* pour Firefox ?

La bêta de macOS Big Sur a rassuré Mozilla : Apple n'abandonnait aucune des API majeures dont dépend le *browser*.

La numérotation des versions de l'OS (passage de 10.x à 11.0) a en revanche engendré quelques soucis. La **mise à jour des installations existantes** de Firefox x64 aussi. Et pour cause : les applications émulées sur Rosetta lancent obligatoirement leurs processus sur cette même couche.

La version native de Firefox se montre deux fois plus performante « sur certains *benchmarks* », affirme Mozilla. Le démarrage, en particulier, est beaucoup plus rapide. Cela tient probablement au compilateur JavaScript JIT. Sur Rosetta, le code qu'il produit doit obligatoirement être traduit pour l'architecture Arm64.

Mozilla a, sur sa feuille de route à court terme, un autre compilateur : celui qui gère WebAssembly. Une version plus rapide (Craneflight) est en cours de test sur le canal Nightly. La fondation en développe également un en interne (Ion) et entend en finaliser le portage Arm64 « sous quelques semaines ».

Sur la roadmap, il y a également WebRender, pour le moment inactif à cause de bugs graphiques. Son retour devrait intervenir avec Firefox 85. Mozilla entend aussi moderniser son usage des API de rendu héritées de macOS (problèmes, entre autres, avec le mode sombre). Et exploiter l'architecture de la puce Apple M1 (4 cœurs haute performance et 4 cœurs moins énergivores).