

# CHOIX DES TECHNOLOGIES PRINCIPALES

## BOB PARTY

Dès le début du projet, nous nous sommes tout de suite mis d'accord pour faire une application mobile et cross-plateforme, qui fonctionnerait donc à la fois sur Android et sur IOs, tout en prenant soin d'éviter de faire une PWA (Progressive Web App), qui ne serait jamais acceptée sur IOs en situation réelle (sources 1&2). Nous avons donc fait quelques recherches et trois grandes possibilités de technologies sont sorties du lot, à savoir Xamarin, Flutter et React Native (sources 1&4). Pour nous aider à y voir un peu plus clair, nous avons dressé le tableau suivant, dressant les avantages et les inconvénients de chacune de ces technologies :

Comme on peut le constater, Flutter et Xamarin ont tout deux de très gros avantages de leurs côtés, notamment le

Xamarin test cloud (outil pour tester l'application sur de nombreux appareils, source 8) ou le système de widget pour Flutter (source 6). Seulement, ces deux technologies sont peu répandues voire même peu soutenues par la communauté (source 12). Notre réflexion a donc été la suivante : « Est-ce réellement utile d'apprendre une technologie peu répandue qui risque de ne pas nous resservir à l'avenir ? ». Ainsi nous avons fait un premier pas vers

Technologies	Langage(s)	Avantages	Inconvénients
Flutter (sources 6&7)	Dart, C, C++ (source 4)	- open source - excellentes performances - système de widgets	- performances toujours inférieures a celles des applications natives - framework jeune (2017, peu répandu et communauté réduite)
Xamarin (sources 8,9,11)	.NET , C# (source 4)	- excellentes performances - Xamarin test cloud - Support de Microsoft	- peu répandu - applications très volumineuses (2* la taille des fichiers natifs environ) - possibilité que l'application ne marche pas sur tout les appareils
React Native (sources 3,5,10)	Js (source 4)	- excellentes performances - open source - rechargements en direct - techno très populaire et très soutenue des développeurs (42% des app des applis cross plateforme)	- mauvaise gestion de la mémoire - mauvais niveau de sécurité du au Js, manque de quelques composants

React Native. En plus du fait que React Native est beaucoup utilisé dans le monde des applications mobiles cross-plateformes (source 12), la technologie de rechargement en direct nous a beaucoup attirée. De plus, nous trouverons sans doute beaucoup plus d'aide à la résolution de problème en ligne sur React Native que pour les autres technologies, due à sa notoriété. En ce qui concerne les inconvénients de cette dernière technologie, nous avons appris à gérer la mémoire en cours, et en ce qui concerne le niveau de sécurité, avouons que notre application n'a pas besoin d'être très optimale là dessus (pas de transactions bancaires par exemple)(source 5). Le seul véritable problème que nous pourrions rencontrer avec cette technologie est donc le manque de quelque composants, ce qui n'est pas si dramatique (source 5).

## Sources :

1. <https://talks.freelancerepublik.com/creer-une-app-mobile-quelle-techno-choisir/>
2. <https://www.inflexsys.com/application-native-ou-hybride/>
3. <https://blog.back4app.com/fr/quest-ce-que-le-framework-react-native/>
4. <https://blog.devgenius.io/flutter-vs-xamarin-vs-react-native-which-one-to-choose-in-2022-48368604be8>
5. <https://blog.back4app.com/fr/react-native-avantages-et-inconvenients/>
6. <https://www.appstud.com/fr/blog-flutter-avantages-inconvenients/>
7. <https://www.bocasay.com/fr/principaux-avantages-flutter/>
8. <https://www.software-developer-india.com/fr/avantages-et-inconvenients-de-xamarin/>
9. <https://blog.back4app.com/fr/xamarin-vs-react-native/>
10. [https://www.youtube.com/watch?v=gvkqT\\_Uoahw](https://www.youtube.com/watch?v=gvkqT_Uoahw)
11. <https://www.youtube.com/watch?v=fq4N0hgOWzU>
12. <https://blog.back4app.com/fr/les-10-principaux-avantages-de-react-native/>