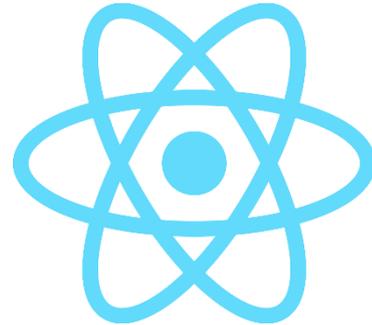


# **Veille Technologique : Les Évolutions de React Native en 2025, Tobias Leser**



## **Qu'est-ce qu'une veille technologique ?**

La veille technologique est un processus qui consiste à rester vif sur l'actualité des technologies pour pouvoir exploiter de nouvelles compétences le plus tôt possible. Pour un développeur par exemple, en s'adaptant le plus vite aux technologies il sera plus compétitif dans le milieu de l'informatique.

## **Comment j'effectue ma veille technologique ?**

J'alimente ma veille technologique principalement grâce au réseau social X (anciennement Twitter), qui personnellement m'informe très vite des avancées technologiques.

Je suis plusieurs comptes d'actualités IT comme :

**@TechCrunch**, **@ThePracticalDev**, **@verge** pour être toujours à jour sur les nouveautés, les tendances et les discussions dans l'univers tech.

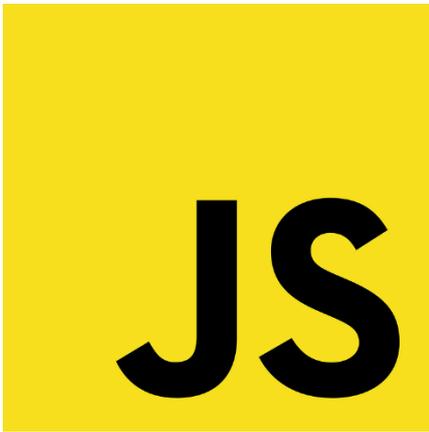
J'utilise aussi Google comme navigateur pour rechercher les dernières tendances et approfondir des sujets récents tendances.

## **Veille Technologique : L'évolution de React Native en 2025**

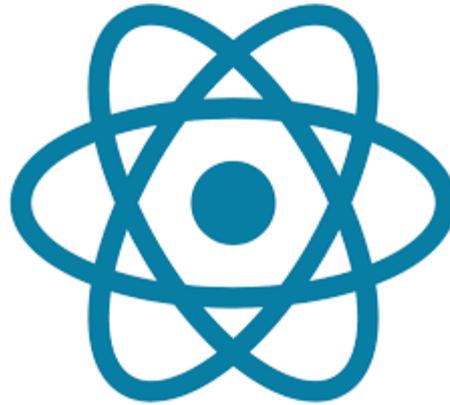
Dans le cadre de ma veille technologique, j'ai récemment consulté plusieurs articles qui m'ont interpellé sur l'avenir de React Native et sa concurrence avec Flutter.

## **Qu'est-ce que React Native ?**

React Native est un framework open-source créé par Meta (anciennement Facebook) qui permet de développer des applications mobiles multiplateformes (iOS et Android) en utilisant JavaScript et React.

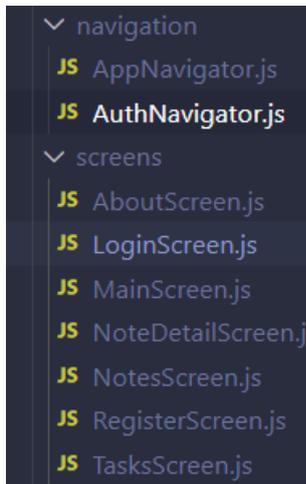


+



Exemple de code React Native : Thinkty, mon projet mobile :

Structure et AppNavigator.JS :



```
1 import React from 'react';
2 import { createStackNavigator } from '@react-navigation/stack';
3 import MainScreen from '../screens/MainScreen';
4 import AboutScreen from '../screens/AboutScreen';
5 import TasksScreen from '../screens/TasksScreen';
6 import NotesScreen from '../screens/NotesScreen';
7 import NoteDetailScreen from '../screens/NoteDetailScreen';
8
9 const Stack = createStackNavigator();
10
11 const AppNavigator = () => {
12   return (
13     <Stack.Navigator>
14       <Stack.Screen name="Accueil" component={MainScreen} />
15       <Stack.Screen name="À propos" component={AboutScreen} />
16       <Stack.Screen name="Mes Tâches" component={TasksScreen} />
17       <Stack.Screen name="Mes Notes" component={NotesScreen} />
18       <Stack.Screen name="Détail de la note" component={NoteDetailScreen} />
19     </Stack.Navigator>
20   );
21 };
22
23 export default AppNavigator;
```

Exemple d'un screen et son CSS directement intégré :

```
import React from 'react';
import { View, Text, TouchableOpacity, StyleSheet } from 'react-native';

const AboutScreen = ({ navigation }) => {
  return (
    <View style={styles.container}>
      <Text style={styles.title}>À propos de Thinkty</Text>
      <Text style={styles.description}>
        Thinkty est une application mobile conçue pour offrir un espace sécurisé aux utilisateurs.
        Grâce à Thinkty, vous pouvez gérer vos tâches, prendre des notes et stocker vos mots de passe
        en toute sécurité. Ce projet a été développé avec passion par Tobias Leser dans le cadre de ses études en BTS SIO.
      </Text>
      <TouchableOpacity style={styles.button} onPress={() => navigation.goBack()}>
        <Text style={styles.buttonText}>Retour</Text>
      </TouchableOpacity>
      <Text style={styles.copyright}>© 2025 Tobias Leser</Text>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    padding: 20,
    backgroundColor: 'white',
  },
});
```

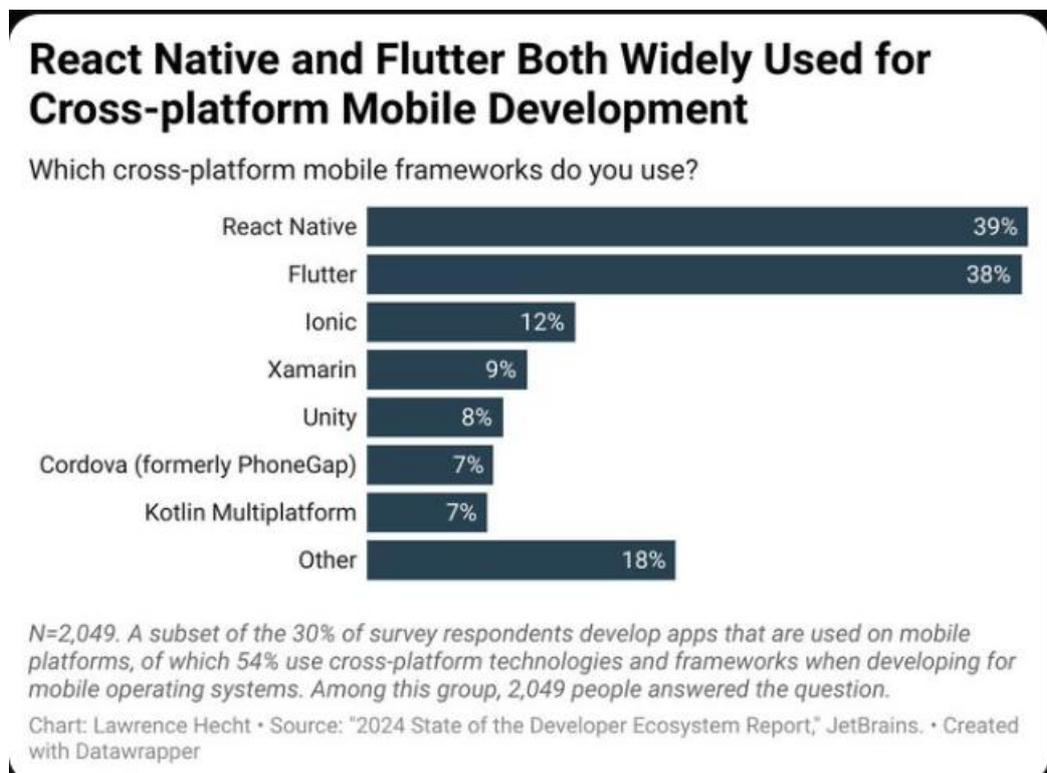
## Pourquoi choisir React Native ?

Code unique pour iOS et Android : Évite d'avoir deux bases de code distinctes.

Large communauté et support : Un écosystème avec de nombreuses bibliothèques.

Performance optimisée : Utilisation du bridge natif pour exécuter du code plus efficacement.

Facilité d'intégration : Compatible avec des applications existantes et des technologies web.



Source de l'image : <https://x.com/LawrenceHecht/status/1880641241924431954>

## React Native vs Flutter : une concurrence toujours d'actualité

En approfondissant ma recherche, j'ai découvert une analyse comparative sur Browser Stack qui met en évidence les différences entre React Native et Flutter. React Native se distingue par son écosystème mature rempli de bibliothèque et son utilisation du

JavaScript, tandis que Flutter mise sur Dart et une approche plus performante pour l'interface utilisateur.

Les tendances et innovations à surveiller

Enfin, le site Dernier Cri recense les tendances émergentes et les évolutions de React Native. Il est intéressant de noter que l'amélioration des performances et la compatibilité avec des plateformes comme VisionOS et d'autres technologies Apple sont des axes de développement importants.

	Language	Facilité	Cout	Performance	Système
Flutter	Dart	Apprendre le langage Dart	Open-source	Code Natif= ultra rapide	Supporté par Google
React Native	JS	Intuitif pour les dev JS	Open-source	Performance moyenne pour appli lourde	Supporté par Meta

## Les avancées de React Native en 2025

En 2025, React Native a fait de grands progrès qui renforcent son efficacité pour le développement d'applications mobiles :

**Nouvelle architecture :** L'adoption de la nouvelle architecture a permis d'améliorer les performances, notamment avec des applications qui affichent des rendus beaucoup plus rapides. Cela a été mesuré par des améliorations allant jusqu'à 2,5 fois la vitesse d'affichage des écrans d'accueil.

**Outils de développement améliorés :** Les nouveaux outils de développement offrent une meilleure expérience de débogage, incluant des fonctionnalités comme l'inspection de l'arbre des composants et l'analyse des requêtes réseau. Cela facilite grandement le processus de développement et de maintenance des applications.

**Adoption par des grandes entreprises :** Des sociétés comme Shopify, Microsoft et Meta continuent d'utiliser React Native pour leurs applications mobiles, ce qui témoigne de la fiabilité et de la performance du framework.

**Documentation enrichie :** La documentation de React Native a été améliorée, offrant aux développeurs des ressources plus détaillées pour optimiser le développement d'applications.

## Conclusion

Cette veille m'a permis de mieux comprendre l'évolution de React Native face à Flutter et d'identifier les améliorations à venir. Bien que la concurrence entre les deux technologies soit forte, React Native reste un choix pertinent pour le développement d'applications mobiles, notamment grâce à son intégration avec l'écosystème JavaScript et ses améliorations continues.

## **Sources et liens :**

<https://x.com/LawrenceHecht/status/1880641241924431954>

<https://www.epsi.fr/actualites/flutter-react-native>

<https://www.designrush.com/agency/mobile-app-design-development/trends/flutter-vs-react-native>

<https://www.browserstack.com/guide/flutter-vs-react-native>

<https://derniercri.io/radar-items/react-native>

<https://webevis.com/blog/is-react-native-still-worth-in-2025-and-beyond>

[https://www.reddit.com/r/reactnative/comments/13vm59d/why\\_react\\_native\\_and\\_not\\_flutter/?rdt=41974](https://www.reddit.com/r/reactnative/comments/13vm59d/why_react_native_and_not_flutter/?rdt=41974)

Lien officiel de React Native : <https://reactnative.dev/>

Copyright © 2025 Tobias Leser. Tous droits réservés. Document rédigé par Tobias Leser |

Pour tout contact ou informations : [lesertobias3@gmail.com](mailto:lesertobias3@gmail.com)