

# Documentation Technique - Walk-Up

## Introduction

Walk-Up est une application conçue pour encourager la marche et la course à pied en groupe. Inspirée de l'application Karos mais adaptée aux piétons, elle permet aux utilisateurs de partager un itinéraire et d'inviter d'autres membres à les rejoindre.

Ce document décrit en détail l'architecture technique de l'application, les technologies choisies et les raisons de ces choix, en tenant compte des critères de performance, de facilité d'intégration, et d'éco-conception.

# 1. Architecture Générale

L'application Walk-Up repose sur une architecture **serverless**, intégrant Firebase pour la gestion des utilisateurs et des données, ainsi que Mapbox pour l'affichage des cartes. Elle est développée en **Next.js** pour le frontend, garantissant une **performance optimale** grâce au rendu côté serveur (SSR). L'hébergement est assuré par **Scaleway** ou **Infomaniak**, des solutions respectueuses de l'environnement.

## Schéma de l'architecture

1. **Utilisateur** → Accède à l'application (PWA) via un navigateur.
2. **Next.js (React)** → Interface utilisateur et gestion des vues.
3. **Firebase Authentication** → Gestion des comptes utilisateurs.
4. **Firebase Firestore** → Stockage des trajets et messages du chat.
5. **Mapbox API** → Affichage des cartes et itinéraires.

6. **Mapbox Static API** → Génération des cartes statiques pour l'application.

## 2. Technologies et Justifications

### 2.1 Développement Frontend

#### Next.js (React)

- **Performance** : Rendu côté serveur (SSR) et statique (SSG) pour une navigation fluide.
- **Facilité d'intégration** : Compatible avec Firebase et Mapbox.
- **Éco-conception** : Gestion optimisée des ressources et réduction des requêtes inutiles.

## 2.2 Système de Cartographie

### Geoapify API

- **Coût avantageux** : Un modèle de tarification plus abordable que Google Maps, tout en offrant un plan gratuit généreux.
- **Performance comparable** : Fournit des services de cartographie et de géocodage performants.
- **Facilité d'intégration** : API bien documentée et compatible avec React et Next.js.
- **Évolutivité** : Une alternative robuste qui s'adapte bien à la croissance du projet.
- **Green IT & Éco-conception** : Infrastructure plus optimisée et respectueuse de l'environnement, contribuant à un projet plus durable.

## 2.3 Authentification et Sécurité

### Firestore Authentication

- **Gestion complète des utilisateurs** (email, Google, Facebook...).
- **Règles Firestore** pour sécuriser les accès aux données.
- **Évolutivité** : S'intègre nativement avec Firestore.

### Pourquoi passer par un tiers de confiance ?

L'utilisation d'un **tiers de confiance** comme **Docaposte** est essentielle pour garantir :

1. **Fiabilité des vérifications** : Les documents d'identité sont analysés avec des algorithmes avancés et validés par des experts si nécessaire.
2. **Sécurité des données** : Protection contre la fraude et stockage sécurisé conforme aux normes légales.
3. **Simplicité pour l'utilisateur** : Expérience fluide où les utilisateurs peuvent vérifier

## 2.4 Chat en Temps Réel

### Firestore + Firestore Triggers

- Synchronisation instantanée pour le chat entre utilisateurs.
- WebSockets intégrés via Firestore Triggers pour des échanges rapides.
- Moins énergivore que d'autres solutions WebSockets dédiées.

## 2.5 Base de Données

### Firestore (NoSQL)

- Évite l'utilisation de plusieurs services en restant cohérent avec Firebase.
- Scalabilité automatique sans gestion de serveur.
- Optimisé pour les requêtes en temps réel et les notifications.

## 2.6 Statistiques et Suivi des Données

### Chart.js / Recharts

- Graphiques légers et dynamiques.
- Bonne compatibilité avec Next.js.
- Impact environnemental réduit comparé à D3.js.



## 2.7 Intégration de la Météo

### MeteoStat API

- Données météo précises et open-source.
- API légère, réduisant la consommation de ressources.
- Mise en cache possible via Firebase Functions.

## 2.8 Hébergement Green

### Scaleway

- Data centers alimentés en énergie renouvelable.
- Infrastructure optimisée pour réduire l'empreinte carbone.
- Compatibilité avec Firebase Hosting pour servir les fichiers statiques.

### 3. Conclusion

L'architecture technique de Walk-Up repose sur une stack moderne, performante et respectueuse de l'environnement. L'intégration de Firebase pour la gestion des utilisateurs et des données simplifie le développement et améliore la scalabilité. Mapbox garantit un affichage des trajets fluide et précis, tandis que l'hébergement green (Scaleway ou Infomaniak) assure un impact environnemental réduit.

En choisissant ces technologies, Walk-Up garantit une expérience utilisateur fluide, sécurisée et alignée avec les principes de l'éco-conception.

## 4. Matrices de Décision

Les choix technologiques ont été validés à l'aide des matrices de décisions suivantes :

### 4.1 Framework Frontend

Technologie	Coût	Performance	Facilité d'intégration	Évolutivité	Support communautaire	Green IT
Next.js (React)	3	5	5	5	5	3
Vue.js (Nuxt.js)	4	4	4	4	4	4
SvelteKit	3	5	3	4	3	5

## 4.2 Système de Cartographie

Technologie	Coût	Performance	Facilité d'intégration	Évolutivité	Support communautaire
Google Maps API	2	5	5	5	5
OpenStreetMap (Leaflet.js)	5	4	3	3	4
Mapbox API	3	5	4	5	4
Geoapify API	4	5	5	5	4

# 4.3 Matrice de Décision - Tier de Confiance pour la Vérification d'Identité

Technologie	Coût	Performance	Facilité d'intégration (Next.js)	Évolutivité	Support communautaire	Cor (l e
Docaposte	4	5	5	5	4	5
Onfido	3	5	4	5	4	4

## 4.4 Matrice de Décision - Hébergeur Green

Hébergeur	Coût	Performance	Facilité d'intégration (Next.js)	Évolutivité	Support communautaire	Green IT
Scaleway	4	5	5	5	4	5
Infomaniak	4	5	4	5	4	5
Vercel	3	5	5	5	5	3
Cloudflare Pages	3	5	5	5	5	3

## 4.5 Matrice de Décision - Base de Données

Base de Données	Coût	Performance	Facilité d'intégration (Next.js)	Évolutivité	Support communautaire	Gratuité
Firebase Firestore (NoSQL)	4	5	5	5	5	4
PostgreSQL (Google Cloud SQL)	3	5	4	5	5	3
MongoDB Atlas	3	4	4	5	4	3
Supabase						

## Annexe

# Sources et Références

## Framework Frontend (Next.js)

- [Next.js Documentation](#)
- [Next.js GitHub](#)
- [Why Next.js for Production](#)

## Système de Cartographie (Geoapify, Mapbox, Google Maps, OpenStreetMap)

- [Geoapify API](#)
- [Mapbox API Documentation](#)
- [Google Maps API Pricing](#)
- [OpenStreetMap Documentation](#)



## Authentification et Sécurité (Firebase Auth, Docaposte, Onfido)

- [Firebase Authentication](#)
- [Docaposte Identity Verification](#)
- [Onfido Identity Verification](#)

## Chat en Temps Réel (Firebase Firestore, WebSockets)

- [Firebase Firestore](#)
- [Firestore Real-time Features](#)
- [Socket.IO Documentation](#)

## Base de Données (Firestore, PostgreSQL, MongoDB, Supabase)

- [Firestore NoSQL Database](#)
- [PostgreSQL Official Site](#)
- [MongoDB Atlas Documentation](#)
- [Supabase PostgreSQL](#)

## Intégration des Statistiques et Graphiques (Chart.js, Recharts)

- [Chart.js Documentation](#)
- [Recharts Documentation](#)
- [D3.js Documentation](#)

## API Météo (MeteoStat, OpenWeather, WeatherAPI)

- [MeteoStat API](#)
- [OpenWeather API](#)
- [WeatherAPI Documentation](#)

## Hébergement Green (Scaleway, Infomaniak, Vercel, Cloudflare Pages)

- [Scaleway Green Hosting](#)
- [Infomaniak Green IT](#)
- [Vercel Serverless Hosting](#)
- [Cloudflare Pages](#)

## Éco-Conception et Green IT

- [Green IT Best Practices](#)
- [The Green Web Foundation](#)