

Zo Mamitiana Rasolomampionona

Ingénieur Full-Stack Senior — Architecture Front-End & Performance

rasolomampionona.zm@gmail.com · +261 34 93 978 09 · zmras.dev · linkedin.com/in/zmras
Antananarivo, Madagascar · Ouvert à la relocalisation · Télétravail

PROFIL

Ingénieur full-stack senior, 8 ans dans la tech dont 7+ à concevoir des applications web à fort trafic en production. Je prends en charge tout le cycle de livraison — architecture front et back, CI/CD, tests, infrastructure — et je vais plus loin que la plupart sur la performance et le rendu front-end : je profile avant d'optimiser, et je justifie chaque décision avec des données. Chez FoodChéri, je maintiens trois produits distincts (application client, POS en boutique et bornes self-service) qui expédient plus de 3 000 commandes par jour — plus d'un million par an, en full remote au sein d'une équipe française. J'ai co-construit le MVP de Seazon en 2018 et j'en ai toujours la charge aujourd'hui, à l'échelle — ce qui veut dire que j'ai vécu assez longtemps avec mes propres décisions d'architecture pour savoir lesquelles étaient les bonnes.

Stack principale : React · TypeScript · Node.js · Performance web · CI/CD · Architecture logicielle

EXPÉRIENCE

Ingénieur Full-Stack Senior — Architecture Front & Back

août 2021 – présent

FoodChéri S.A. · France (Télétravail)

Périmètre. Architecture front et back de trois produits distincts pour FoodChéri et sa marque sœur Seazon, expédiant plus de 3 000 commandes par jour à eux trois, en full remote au sein d'une équipe française :

- **L'application client** — sélection des repas, paiement et livraison.
- Le **système POS** utilisé par les équipes dans les boutiques FoodChéri en entreprise (clients corporate accueillant une boutique dans leurs locaux).
- Les **bornes self-service**, où les salariés badgent avec leur carte d'entreprise, commandent, paient et récupèrent leurs repas dans un frigo sur place — sans personnel, donc sans personne pour rattraper une panne.

Trois surfaces, trois types d'utilisateurs, trois modèles d'interaction et trois modes de défaillance : un client chez lui, un caissier sous pression, et quelqu'un seul devant un terminal. Cette diversité a façonné ma façon de penser la fiabilité — un crash sur la borne, c'est quelqu'un qui n'a pas de déjeuner, sans personne à qui demander de l'aide.

Performance & rendu. La grille de sélection de repas — le cœur du parcours d'abonnement sur les deux marques — était lente malgré des données rapides. Le profiling flamegraph de React DevTools a tracé le coût jusqu'à react-jss, qui recalculait les styles depuis des props changeantes à chaque render, multiplié sur 24 à 50 cartes. La solution évidente était la virtualisation ; je l'ai écartée parce qu'elle masque le coût au lieu de le supprimer, et ajoute une complexité permanente de scroll et d'accessibilité. À la place, j'ai remplacé le CSS-in-JS runtime par Vanilla Extract (CSS généré au build, zéro coût à l'exécution) et mémorisé les composants de carte — l'un rend chaque render peu coûteux, l'autre rend les renders rares. Résultat : ~60 % de re-renders en moins et, combiné au code-splitting par route et à une migration Webpack→Vite, ~40 % de chargement en moins sur les deux marques (mesuré dans Lighthouse).

Architecture de build & livraison. Consolidation de six dépôts (trois produits × front/back) en un monorepo NX unique. L'arbitrage pesé : complexité de l'outillage monorepo et rayon d'impact plus large, contre configuration dupliquée, dépendances qui divergent et du code partagé qui était en réalité trois copies — un risque qui s'est matérialisé quand un changement Stripe a atterri dans une copie et pas les autres, faisant échouer des paiements et annulant une journée de commandes. J'ai envisagé des packages npm privés ; ce modèle se justifie quand les applications appartiennent à des équipes indépendantes avec des cadences de release différentes, ce qui n'était pas notre cas — même équipe, fortement couplées, évoluant ensemble. Le monorepo nous a permis de modifier le code partagé et tous ses consommateurs en un seul commit. Associé à un pipeline CI/CD (GitHub → CircleCI → Docker → GCP), la fréquence de déploiement a augmenté d'environ 75 % ; le graphe des affected de NX a gardé la CI rapide en ne rebuildant que ce qui change.

Fiabilité & qualité. Mise en place de Sentry et des React Error Boundaries dans les applications, réduisant le taux de crash de l'interface d'environ 80 %. Remplacement des tests de régression manuels avant release par une couverture end-to-end Playwright des parcours critiques — un arbitrage entre effort d'écriture des tests en amont et confiance à la release, avec des heures-ingénieur récupérées à chaque cycle.

Paiements. Unification de cinq prestataires (Stripe, PayPal, Braintree, Paygreen, Swile — dont les titres-restaurant exigés par les clients corporate) derrière un tunnel de commande unique, en abstrayant les différences entre prestataires pour que le checkout reste agnostique et que de nouveaux moyens de paiement s'ajoutent sans toucher au cœur du flux.

Leadership. Définition des standards front-end et accompagnement d'une équipe d'environ 8 développeurs — revue de code, pratiques de performance, patterns React modernes et les décisions d'architecture ci-dessus.

Stack : React, TypeScript, Node.js, Redux, Vanilla Extract, Material UI, Vite, NX, Docker, CircleCI, GCP, Playwright, Sentry, WebSockets, MongoDB

Ingénieur Full-Stack

sept. 2018 - juil. 2021

FoodChéri S.A. · France (Télétravail)

Périmètre. Développement front et back sur FoodChéri et Seazon, avec un focus sur le greenfield, les fondations UI partagées et l'infrastructure de contenu.

Construire Seazon depuis zéro. J'étais l'un des deux ingénieurs qui ont construit le MVP d'origine de Seazon. Le produit est passé de rien à la production et tourne encore aujourd'hui — et j'en ai toujours la charge, désormais pour des centaines d'utilisateurs quotidiens. Construire un produit de zéro puis vivre avec ces décisions pendant huit ans est une discipline différente de celle qui consiste à rejoindre une base de code mature : chaque raccourci pris au début est un raccourci qu'il a fallu ensuite soit justifier, soit rembourser.

Fondations front partagées. Développement d'une bibliothèque de composants React réutilisables, documentée dans Storybook, partagée entre les deux marques — socle d'un design system toujours utilisé. Privilégier une source de vérité unique et documentée plutôt que des composants par projet : un arbitrage entre flexibilité initiale et cohérence et vélocité à long terme sur deux produits. Comme les fondations sont partagées, le travail ultérieur sur la performance et les tests a profité aux deux marques et non à une seule.

Infrastructure de contenu. Conception et développement d'un système de landing pages piloté par CMS permettant à l'équipe marketing de publier des modifications en moins d'une minute — contre une journée entière nécessitant un développeur et un déploiement. L'intention d'architecture : retirer entièrement l'ingénierie du chemin de mise à jour du contenu — le marketing publie seul, les développeurs cessent d'être un goulot d'étranglement.

Backend & APIs. Développement d'APIs Node.js pour le suivi de livraison, la gestion des commandes et les notifications SMS/email, ainsi que des tableaux de bord d'administration et d'exploitation (React, Material UI, Recharts) pour les équipes internes.

Stack : React, Node.js, Storybook, Material UI, Recharts, API REST, MongoDB, Parse Platform

Développeur Logiciel

janv. 2018 – août 2018

Connecteo · Madagascar

- Développement d'outils internes pour les équipes support, traduisant les workflows métier en solutions concrètes, en collaboration étroite avec le produit.

Administrateur de Base de Connaissances

nov. 2016 – déc. 2017

Telma Madagascar (anciennement Yas) · Madagascar

- Définition de la stratégie de gestion des connaissances et maintien d'une base de contenu structurée et consultable pour les équipes support.

Stagiaire Systèmes Embarqués

oct. 2015 – avr. 2016

Orange Madagascar · Madagascar

- Conception d'un système de supervision de la QoS réseau 2G/3G (Raspberry Pi, Python, commandes AT) avec un tableau de bord HTML/CSS/PHP pour les métriques réseau en temps réel.

COMPÉTENCES TECHNIQUES

Frontend :	React, TypeScript, JavaScript (ES6+), Redux, Vanilla Extract, Material UI, Tailwind CSS, Framer Motion, Recharts, React-JSS, HTML5, CSS3
Backend :	Node.js, Express, Parse Platform, MongoDB, API REST, WebSockets, Webhooks
Performance & Tests :	optimisation des performances web, profiling React DevTools, Playwright, Jest, React Testing Library, Storybook, Sentry
Build & Infrastructure :	Vite, Webpack, Docker, CI/CD, CircleCI, GitHub Actions, Google Cloud Platform (GCP), monorepo NX
Architecture & Pratiques :	architecture logicielle, design systems, revue de code, mentorat, leadership technique

FORMATION

Master (M2) — Génie Logiciel & Intelligence Artificielle

Institut Supérieur Polytechnique de Madagascar (ISPM) · 2014 – 2016

Licence — Génie Logiciel

Institut Supérieur Polytechnique de Madagascar (ISPM) · 2010 – 2013

LANGUES

Français (courant) · Anglais (professionnel – IELTS 6.5) · Malgache (langue maternelle)