

Amissan AMOUSSOU-G.

Ingénieur iOS senior

Alpes-Maritimes · Télétravail complet ou fréquent · Français · anglais professionnel (TOEIC 840) · amissan.ag@outlook.fr · GitHub · LinkedIn

PROFIL

Actuellement ingénieur de développement iOS chez **Instant System**, où la **billettique mobile** est mon domaine d'expertise.

J'ai eu la chance de travailler dans des entreprises de toutes tailles depuis le début de ma carrière : une startup de moins de dix salariés, une entreprise de taille intermédiaire avec Instant System, et un grand groupe du CAC 40 avec Orange, via Inetum. Je suis très versatile et je m'adapte facilement à mon environnement comme à mon cadre de travail. Ce qui ne change pas, quel que soit le contexte, c'est ma rigueur et mon engagement dans mon équipe et dans la vie de l'entreprise.

Pendant **une année complète**, j'ai tenu seul la **billettique mobile iOS** d'Instant System. J'y ai conçu des fonctionnalités qui sécurisent aujourd'hui nos applications de m-ticket.

~5 M

d'utilisateurs des applications auxquelles j'ai contribué

6 ans

d'ingénierie iOS, depuis octobre 2020

> 99,9 %

de sessions sans crash sur KCalories

EXPÉRIENCE

Instant System

Sophia Antipolis · mai 2023 – aujourd'hui

Ingénieur iOS

Référent technique billettique

Scrum Master · depuis février 2025

Ambassadeur IA

Tuteur

- **Seul développeur mobile** du périmètre billettique pendant un an, sur l'ensemble du portefeuille : stabilisation de la production et développement de fonctionnalités.
- Développement **quasiment seul du rechargement de carte de transport** sur l'application TCL, dont je suis aujourd'hui l'un des référents techniques.
- Deux **bibliothèques internes** conçues et livrées : la protection anti-fraude des titres à l'écran, et la génération des QR codes de validation.
- **Création du module d'un nouveau prestataire billettique**, et remontée au niveau du produit des implémentations indépendantes du prestataire — défendue, puis obtenue.
- **Refonte complète de la gestion de session** — jetons, renouvellement, retry, déconnexion — portée par des acteurs et `async/await`.
- **Onboarding de tous les développeurs mobiles** arrivés après moi, iOS et Android ; tutorat officiel d'un développeur iOS.
- **Ateliers d'architecture avec le CTO** et les architectes ; démos client ; conduite d'entretiens de recrutement.
- **Scrum Master** de mon équipe sans quitter le développement — et resté l'un de ses principaux contributeurs techniques.

STACK · Swift · SwiftUI · UIKit · Swift Concurrency, acteurs, `async/await` · Combine · Clean Architecture 3 puis 4 couches · injection de dépendances · SPM, XCFrameworks, CocoaPods · XcodeGen et specs YAML · Jenkins · fastlane · match · Bitbucket · SAlFe

Mail Orange — Orange, via Inetum

Mougins · janv. 2022 – avr. 2023

Ingénieur iOS

- Fonctionnalités et corrections sur une messagerie à **environ un million d'utilisateurs**, dans une équipe de trois développeurs iOS.
- Sujets menés **de bout en bout** dès les premières semaines : brouillons, gestion des pièces jointes.
- Maintien de la **compatibilité multi-versions d'iOS** et du **layout iPad**, sur une base très utilisée en tablette.
- Travail sur un existant hétérogène — VIPER sur le composeur, MVP ailleurs.
- Participation aux raffinements et aux décisions techniques ; échanges avec les **équipes tierces** sur les services intégrés.

STACK · Swift · Objective-C · UIKit programmation, xib, storyboards, Auto Layout · MVP · VIPER · Delegate, Observer · UIDocumentPicker · app extensions · CocoaPods · SAlFe

STILT

Nice · oct. 2020 – oct. 2021

Ingénieur iOS — alternance

- **Migration complète d'Objective-C vers Swift** de l'application d'autopartage de véhicules premium, déjà publiée sur l'App Store — la base passe de **30 % à plus de 92 % de Swift en un an**.

- Migration menée **en parallèle des livraisons de fonctionnalités** par un collègue, sur la même base, sans interrompre les mises en production. La stratégie était de mon ressort.
- Refonte de la couche réseau, profiling et optimisation.

STACK · Objective-C → Swift · UIKit · MVC · storyboards, xib · CocoaPods · migration incrémentale sur application vivante

PROJET PERSONNEL

KCalories, le compagnon nutritionnel

[App Store](#)

DÉCISIONS ET RÉALISATIONS

- **Construire les quatre stacks moi-même** plutôt que d'assembler des services : application iOS, backend, back-office d'administration, page de présentation. Tenir le produit de bout en bout était le but, pas un moyen.
- **Architecture stricte et modules SPM dès le premier jour**, quand rien ne l'imposait encore — les frontières se posent avant d'avoir de quoi les remplir, sinon elles ne se posent jamais.
- **Feature flags et intégration continue dès la première semaine**, pour livrer sans bloquer et sans rituel manuel.

STACK · SwiftUI · SwiftData · Clean Architecture · Modules SPM · Feature flags · Backend · Back-office · CI/CD · FR · EN

RÉSULTATS

- **Publiée sur l'App Store cinq mois** après la première ligne de code.
- **Plus de 99,9 % de sessions sans crash**, relevés dans App Store Connect — au-dessus de ce qu'on attend d'une application mobile qui tourne bien.
- **Version 2.0 vingt jours** après la 1.0 : la chaîne de livraison tenait.
- Disponible en **français et en anglais**, avec deux thèmes complets.

COMPÉTENCES CLÉS

Concurrence et état partagé

Les jetons de session, leur renouvellement et les retours d'erreur sont l'endroit où les accès concurrents produisent les bugs qu'on ne reproduit jamais. Acteurs, async/await, isolation : ce que le compilateur garantit à ma place vaut mieux que ce que je promets de vérifier.

Frontières et modularisation

Clean Architecture en modules séparés plutôt qu'en dossiers, et des modules qui dépendent d'un contrat, jamais d'une implémentation. Une frontière que le compilateur ne refuse pas n'est pas une frontière — c'est une convention qu'on finit par oublier.

Du code au produit livré

Génération de projet par spec plutôt que par fichier binaire, signature gérée par fastlane et match, pipelines par produit et par module, feature flags pour découpler la livraison de la mise en service. Ce qui n'est pas automatisé finit par ne pas être fait.

COMPÉTENCES

LANGAGES

Swift Objective-C

INTERFACE

SwiftUI UIKit Auto Layout Dynamic Type iPad App extensions

CONCURRENCE

async/await Acteurs Swift Concurrency Combine GCD

ARCHITECTURE

Clean Architecture MVVM MVP VIPER Injection de dépendances Modules SPM Feature flags

OUTILS ET CI/CD

Xcode XcodeGen SPM CocoaPods XCFrameworks fastlane match Jenkins Git Bitbucket

COLLABORATION ET PILOTAGE

Jira — administration, JQL, boards kanban Confluence Google Workspace — agendas partagés, salles, visios

MÉTHODE

SAFe Scrum Code review Tests unitaires Refinement Mentorat

FORMATION

2020 – 2021

Master 2 Ingénierie informatique
Université Côte d'Azur — Nice

2019 – 2021

Diplôme d'ingénieur en sciences informatiques
Polytech Nice Sophia Antipolis — spécialité Intelligence Ambiante et Mobile

2016 – 2019

Ingénieur des travaux informatiques
Institut Africain d'Informatique — Lomé

CERTIFICATIONS

novembre 2025

Professional Scrum Master I
Scrum.org — [Vérifier](#)

2025

Certified SAFe 6 Practitioner
Scaled Agile — [Vérifier](#)

APPLICATIONS EN PRODUCTION

33 — Ales'Y · at bus · Aux_R_M · Bibus · Choletbus · Citibus · Clermoov · ENVIA · Envibus · impulsyon · Irigo · Izilo · JV Malin · lIO · Mobilités TCM & TCAT · My Astuce · Métropole Mobilité · Ondea · Oûra · Palm Bus · QUB · RLV Mobilités · Ruban · Réseau Mistral · Réseau TAC · Setram · Sibra · TAC Mobilités · TCL · Tere Tahiti · TUB · TUL · Twisto

Applications éditées par les autorités de transport et leurs exploitants. Ma contribution porte sur la couche mobile de billettique — achat, usage, validation et contrôle — et son intégration dans chaque application.