Engineer

Client natif macOS en Swift 6

Développé un client de messagerie natif macOS en Swift 6 et SwiftUI — 11 141 lignes dans 53 fichiers — sur l'interface C d'un cœur Rust lié comme archive statique depuis une révision épinglée.

août 2026

11,141

lignes de Swift sur un cœur Rust

Situation. Le protocole de messagerie disposait d'un cœur bien testé écrit en Rust et de clients sur plusieurs plateformes, mais l'expérience de bureau sur macOS était une enveloppe multiplateforme — elle ne ressemblait pas à une application Mac, ne se comportait pas comme telle et embarquait un environnement d'exécution dont une application native n'a pas besoin. Le cœur expose ses capacités via une interface C, ce qui signifie que tout langage capable d'appeler C peut s'appuyer dessus, et Swift en est capable.

Tâche. Un client natif devait être construit directement sur cette interface C, dans le langage et le framework d'interface actuels de la plateforme, avec le cœur lié à l'application plutôt que livré à côté.

Action. L'application représente 11 141 lignes de Swift réparties sur cinquante-trois fichiers dans la cible applicative, construite sur SwiftUI avec le cœur lié comme archive statique compilée depuis une révision amont fixée. Fixer la révision plutôt que suivre une branche est la décision qui rend la construction reproductible : l'interface C est le contrat, et un cœur mouvant change ce contrat sans modifier une ligne de Swift. La structuration en couches limite la surface non sûre — une fine enveloppe possède chaque pointeur et chaque chaîne traversant la frontière, les modèles au-dessus sont des valeurs Swift ordinaires et la couche d'interface ne voit jamais de pointeur brut. Tout ce que retourne l'interface C a une règle de propriété, et se tromper sur l'une produit soit une fuite, soit un plantage, sans avertissement intermédiaire du compilateur ; l'enveloppe concentre donc toute la discipline mémoire du projet.

La construction est pilotée par un exécuteur de tâches avec cinquante-six cibles couvrant la compilation du cœur, celle de Swift, les suites de tests, les linters et le conditionnement.

Résultat. Un client natif qui démarre comme une application Mac, utilise les matériaux et les contrôles de la plateforme et n'embarque aucun navigateur. Ce qui n'a pas été fait doit être dit : ce n'est pas une version distribuée. Il n'existe ni distribution notariée, ni canal de mise à jour, et l'interface ne propose qu'une langue ; c'est donc une application fonctionnelle plutôt qu'un produit utilisé par d'autres.

Mettez cette expérience au service de votre projet

Prenez « Client natif macOS en Swift 6 » comme point de départ pour un défi similaire dans votre organisation. Les prestations mobilisées pour cette réalisation sont disponibles pour votre projet. Définir d'abord le système sur le papier pour que ses éléments s'articulent et que son coût soit connu avant d'écrire du code.

Découvrir nos prestations Contactez-nous →

Examinez ce qui a conduit au résultat

Découvrez la situation, les décisions et le résultat de cet exemple. Nous pouvons discuter des éléments transposables à vos contraintes et de ceux qui nécessitent une autre démarche. Client natif macOS en Swift 6 — 11,141 lignes de Swift sur un cœur Rust

Découvrir nos prestations Contactez-nous →

Besoin d'une vraie application native plutôt que d'une page web emballée ?

Discutez de votre projet avec l'ingénieure responsable de sa réalisation. Nous convenons par écrit du périmètre, de la base tarifaire et de la transmission avant le début des travaux.

Rôle exercé dans ce contexte : Ingénieure applications macOS. Cette réalisation fait appel à Architecture de solutions, Ingénierie frontend, Développement full-stack, API et intégration, Systèmes de design et interfaces et Conception UX/UI. Et puisque vous êtes ici, nous sommes aussi plutôt doués en Architecture de plateformes et de solutions, Développement frontend, Développement de produits full-stack et Conception UI/UX et systèmes de design.

Contactez-nous Découvrir nos prestations

Un élément **d'un parcours plus long** — entièrement présenté sur ce site.

Chaque entrée suit la même structure : la situation, la tâche, ce que nous avons fait et ce qui a changé.

Voir l'ensemble du parcours

Définissez l'amélioration nécessaire

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Besoin d'une vraie application native plutôt que d'une page web emballée ?

Interrogez-nous sur les décisions qui sous-tendent « Client natif macOS en Swift 6 ». Vous parlerez aux personnes capables d'expliquer le travail et d'évaluer votre propre projet.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/portfolio/built-a-native-macos-client-in-swift-6/