Engineer

Built a native macOS client in Swift 6

Built a native macOS messaging client in Swift 6 and SwiftUI — 11,141 lines across 53 files — over the C interface of a Rust core linked as a static archive from a pinned revision.

August 2026

11,141

lines of Swift over a Rust core

Situation. The messaging protocol had a well-tested core written in Rust and clients on several platforms, but the desktop experience on macOS was a cross-platform shell — it did not look like a Mac application, did not behave like one, and carried a runtime that a native application does not need. The core exposes its capability through a C interface, which means any language that can call C can build on it, and Swift can.

Task. A native client had to be built directly on that C interface, in the platform's current language and current interface framework, with the core linked in rather than shipped alongside.

Action. The application is 11,141 lines of Swift across fifty-three files in the app target, built on SwiftUI with the core linked as a static archive compiled from a pinned upstream revision. Pinning the revision rather than tracking a branch is the decision that makes the build reproducible: the C interface is the contract, and a moving core changes the contract without changing a line of Swift. The layering keeps the unsafe surface small — a thin wrapper owns every pointer and every string that crosses the boundary, models above it are ordinary Swift values, and the interface layer never sees a raw pointer. Everything the C interface returns has an ownership rule, and getting one wrong produces either a leak or a crash with no compiler warning in between, so the wrapper is where the entire project's memory discipline lives. The build is driven by a task runner with fifty-six targets covering the core compile, the Swift build, the test suites, the linters and the packaging.

Result. A native client that starts as a Mac application, uses the platform's own materials and controls, and carries no embedded browser. What has not been done should be stated: this is not a shipped release. There is no notarised distribution, no update channel, and one language in the interface, so it is a working application rather than a product anyone else is running.

Bring this experience to your project

Use "Built a native macOS client in Swift 6" as a starting point for a similar challenge in your organisation. The services behind this work are available for your project. The shape of a system worked out on paper first, so the parts fit and the cost is known before anyone writes code.

Explore our services Get in touch →

Look behind the result

Read the situation, decisions and result in this example. We can discuss which parts transfer to your constraints and which need another approach. Built a native macOS client in Swift 6 — 11,141 lines of Swift over a Rust core

Explore our services Get in touch →

Need a real native app, not a wrapped web page?

Talk through your project with the engineer responsible for delivery. We agree the scope, price basis and handover in writing before work begins.

Role context: macOS Application Engineer. It draws on Solution Architecture, Frontend Engineering, Full-Stack Development, APIs & Integration, Design Systems & UI and UX / UI Design. And while you are here, we are rather good at Platform & Solution Architecture, Frontend Development, Full-Stack Product Development and UI/UX Design & Design Systems.

Get in touch Explore our services

One piece of **a longer record** — and it is all on this site.

Every entry is written the same way: the situation, the task, what we did and what changed.

See the whole record

Define the improvement you need

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Need a real native app, not a wrapped web page?

Ask about the decisions behind “Built a native macOS client in Swift 6”. You will speak with the people who can explain the work and assess your own project.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

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

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/portfolio/built-a-native-macos-client-in-swift-6/