Engineer Mjukvara

Byggde en native macOS-klient i Swift 6

Byggde en native meddelandeklient för macOS i Swift 6 och SwiftUI — 11 141 rader i 53 filer — ovanpå C-gränssnittet till en Rust-kärna som länkas som ett statiskt arkiv från en låst revision.

augusti 2026

11,141

rader Swift ovanpå en Rust-kärna

Situation. Meddelandeprotokollet hade en vältestad kärna skriven i Rust och klienter på flera plattformar, men skrivbordsupplevelsen på macOS var ett plattformsoberoende skal — den såg inte ut som en Mac-app, betedde sig inte som en och bar på en körmiljö som en nativ app inte behöver. Kärnan exponerar sin funktionalitet via ett C-gränssnitt, vilket innebär att varje språk som kan anropa C kan bygga på den, och det kan Swift.

Uppgift. En nativ klient måste byggas direkt på det C-gränssnittet, i plattformens nuvarande språk och nuvarande gränssnittsramverk, med kärnan inlänkad i stället för levererad vid sidan av.

Handling. Appen består av 11 141 rader Swift i femtiotre filer i appens target, byggd på SwiftUI med kärnan länkad som ett statiskt arkiv kompilerat från en fastlåst uppströmsrevision. Att låsa revisionen i stället för att följa en gren är det beslut som gör bygget reproducerbart: C-gränssnittet är kontraktet, och en kärna i rörelse ändrar kontraktet utan att en enda rad Swift ändras. Skiktningen håller den osäkra ytan liten — en tunn wrapper äger varje pekare och varje sträng som korsar gränsen, modellerna ovanför är vanliga Swift-värden och gränssnittsskiktet ser aldrig en rå pekare. Allt som C-gränssnittet returnerar har en ägarskapsregel, och ett misstag i någon av dem ger antingen en läcka eller en krasch utan en enda kompilatorvarning på vägen, så det är i wrappern som hela projektets minnesdisciplin finns. Bygget drivs av en task runner med femtiosex mål som täcker kompileringen av kärnan, Swift-bygget, testsviterna, lintverktygen och paketeringen.

Resultat. En nativ klient som startar som en Mac-app, använder plattformens egna material och kontroller och inte bär på någon inbäddad webbläsare. Det som inte är gjort bör sägas rakt ut: detta är ingen levererad release. Det finns ingen notariserad distribution, ingen uppdateringskanal och bara ett språk i gränssnittet, så det är en fungerande app snarare än en produkt som någon annan kör.

Använd erfarenheten i ert projekt

Använd ”Byggde en native macOS-klient i Swift 6” som utgångspunkt för en liknande utmaning i er organisation. Tjänsterna bakom arbetet finns tillgängliga för ert projekt. Systemets form utarbetad på papper först, så att delarna passar ihop och kostnaden är känd innan någon skriver kod.

Se våra tjänster Hör av er →

Se bakom resultatet

Läs om situationen, besluten och resultatet i exemplet. Vi kan diskutera vad som passar era förutsättningar och vad som kräver ett annat angreppssätt. Byggde en native macOS-klient i Swift 6 — 11,141 rader Swift ovanpå en Rust-kärna

Se våra tjänster Hör av er →

Behöver du en riktig native-app och inte en webbsida i ett skal?

Prata om ert projekt med ingenjören som ansvarar för leveransen. Vi avtalar omfattning, hur priset beräknas och överlämning skriftligt innan arbetet börjar.

Roll i uppdraget: Applikationsingenjör för macOS. Det bygger på Lösningsarkitektur, Frontendutveckling, Fullstackutveckling, API:er och integration, Designsystem och UI och UX/UI-design. Och nu när ni är här: vi är ganska bra på Plattforms- och lösningsarkitektur, Frontendutveckling, Fullstackutveckling av produkter och UI/UX-design och designsystem.

Hör av er Se våra tjänster

En del av **en längre historik** — och den finns i sin helhet här.

Varje post är skriven på samma sätt: situationen, uppgiften, vad vi gjorde och vad som förändrades.

Se hela historiken

Definiera förbättringen ni behöver

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Behöver du en riktig native-app och inte en webbsida i ett skal?

Fråga om besluten bakom ”Byggde en native macOS-klient i Swift 6”. Ni talar med personer som kan förklara arbetet och bedöma ert eget projekt.

Se våra tjänster

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://software.engineer.company/sv/portfolio/built-a-native-macos-client-in-swift-6/