Brett Halvorsen Senior Mobile Engineer

Austin, TX · brett.halvorsen@gmail.com · +1 512-555-0189 · linkedin.com/in/bretthalvorsen

PROFILE SUMMARY

- **Senior Mobile Engineer** with **7 years** on **dating and job search apps**, now owning **chat and video calls** across the iOS and Android apps of a dating app with tens of millions of users.
- Builds with **Kotlin** and Swift, **Kotlin Multiplatform** with SQLDelight, Jetpack Compose and SwiftUI, **GraphQL**, and LaunchDarkly and Sentry for releases and monitoring.
- Strongest in **shared chat logic**, **offline message sync** and **staged feature rollouts**, using an outbox pattern and per-feature flags so new chat features reach every market safely.
- Works with **product, design, the backend chat team, trust and safety, and QA**, from the API schema review to the go/no-go call on each weekly release.
- Mentors **3 mobile engineers** on the chat squad, runs the monthly shared code review with the iOS and Android leads, and wrote the team's KMP debugging guide.

TECHNICAL SKILLS

Languages:	Kotlin, Swift, Java, Objective-C (reading)
iOS & Android UI:	Jetpack Compose, SwiftUI, UIKit, VoiceOver, TalkBack
Cross-Platform & Shared Code:	Kotlin Multiplatform, SKIE, Ktor
Architecture & Concurrency:	MVI, MVVM, Kotlin Coroutines, Kotlin Flow, Swift Concurrency, Dagger
Data & APIs:	GraphQL, SQLDelight, Room, Apollo Kotlin, WebSockets
Testing & CI/CD:	JUnit, XCTest, Espresso, XCUITest, BrowserStack, GitHub Actions, Fastlane
Release & Monitoring:	LaunchDarkly, Sentry, Android vitals, Google Play Console, App Store Connect

EDUCATION

University of Texas at Austin — **B.S. in Computer Science** — Austin, TX · Aug 2015 to May 2019

WORK EXPERIENCE

Bumble — **Senior Mobile Engineer** — Austin, TX · Mar 2022 to Present

- Moved chat message handling and sync rules into **Kotlin Multiplatform** for the Bumble iOS and Android apps, sharing **about 40%** of the chat code and ending the logic bugs that showed up on one app only.
- Rebuilt chat sync on a **SQLDelight** store with an outbox for unsent messages and a cursor-based **GraphQL** history query, cutting missing or duplicate message reports from 1.4% to **0.3%** of chats.
- Moved the chat squad onto the weekly release train with **LaunchDarkly** flags and kill switches per feature, rolling voice notes out to **12 markets** in stages with no hotfix.
- Added per-release **Sentry** and Android vitals alerts for chat, then fixed an ANR in the photo picker, cutting the Android chat ANR rate from 0.52% to **0.19%**.
- Shipped video call screens in **SwiftUI** and **Jetpack Compose** with matching flows and full VoiceOver and TalkBack labels, used in **9% of matches** within 2 months.

Indeed — **Mobile Engineer** — Austin, TX · Aug 2019 to Feb 2022

- Split the Android job search app into **14 Gradle modules** with MVVM and Dagger, so 3 squads could build only their own code, cutting clean builds from 11 minutes to **4**.
- Wrote Espresso and XCUITest suites for the apply flow and ran them on a BrowserStack device matrix, raising apply flow test coverage from 35% to **80%**.
- Built a **GitHub Actions** pipeline with Fastlane lanes that sent both apps to TestFlight and Play internal testing on every merge, cutting release prep from 2 days to **3 hours**.