

Senior Android Engineer with 12+ years building Android apps — the last 7 on Just Eat Takeaway.com's consumer app (10M+ MAU), where I co-author the company's cross-app Compose design system and own much of its platform tooling. Specialising in platform engineering, developer experience, and early adoption of new Android capabilities; technical writing featured in *Android Weekly* and platform work featured at Google I/O.

Work Experience

Senior Android Engineer — Just Eat [Takeaway.com](https://www.just-eat-takeaway.com) Jan 2022 – Present

Android Platform & Experience team — consumer app, 10M+ MAU Bristol, UK

Kotlin monorepo of millions of lines and hundreds of modules, shipping 16 branded apps to Google Play.

- Co-author and maintainer of **PIE, JET's Compose-first design system** (its internal Material equivalent) — a full component suite driving hundreds of screens across the consumer, courier, and restaurant order-pad apps, giving ~100 Android engineers ready-made UI to ship faster.
- Led JET's early adoption of **Android 16 Live Updates** for live order tracking — securing pre-release platform access from Google, reporting platform bugs directly to their engineers, and shipping at GA across an 88-ticket workstream. Drove a **22% increase in post-order screen views**, a 2% reduction in support-contact propensity, and thousands of additional order reviews (42% of users keep the notification active through the full order).
- **Single-handedly migrated the monorepo's dependency injection to Dagger Hilt** (30+ tickets, 40+ PRs across every feature domain) — deleting thousands of lines of boilerplate, letting any screen be built as an independent component, and unblocking the app's navigation rework.
- Delivered the **single-activity architecture and adaptive navigation** (bottom bar, rail, and drawer) across the app's top-level screens — unifying deep linking, enabling Android 15 predictive back, and modernising transitions.
- Shipped a **complete rebrand of the consumer app in Jetpack Compose** (2022) within a release cycle or two — my second full rebrand of the app (see previous role).
- Drove the **modularisation of the monorepo** into hundreds of Gradle modules with affected-module CI — later restructuring it to cut full builds from 90+ minutes to under 45 (sub-30 for most changes) and hold cold start under 1s, as the Android team scaled from 12 to ~50 engineers.
- Own CI through **two platform migrations** — **TeamCity** → **Jenkins** → **GitHub Actions** — running thousands of unit and UI tests (Firebase Test Lab) plus thousands of Roborazzi screenshot tests on every PR; fixed systemic test flakiness to **cut Firebase Test Lab spend 45%** (≈\$10.2K → \$5.6K/month).
- Own the monorepo's **code-quality gates** — 15+ custom detekt rules, Android Lint on CI, SonarQube coverage enforcement (60% minimum), and CI autoformatting that raises fix-up PRs automatically — deterministic checks that eliminate nitpick review comments across ~50 engineers.
- Serve on the **release-manager rota for the bi-weekly release train** (~20–30% of my time) — averted a Google Play takedown by driving an urgent PayPal/Braintree SDK upgrade with the vendor, and halted staged rollouts early (wakelock and ANR regressions) before they reached users.
- Keep **production healthy** through extensive logging, automated alerts, and PagerDuty on-call — e.g. diagnosing a Firebase Analytics regression that was silently degrading targeting for €70–100K/day of UK ad spend.
- Core interviewer for the **Android hiring pipeline** — ~20 technical interviews a year since 2023, trained 5 engineers to run the loop, and provided structured feedback that helped hire 15 Android engineers across consumer and ventures teams.

Android Engineer — Just Eat Jan 2019 – Dec 2021

Same team — promoted to Senior Android Engineer, Jan 2022 Bristol, UK

- Delivered the **2020 rebrand of the consumer app end-to-end** on the legacy View-based UI, plus Guest Checkout — both shipped on tight deadlines with no release slips.
- **Halved app cold-start time** (to ~465ms) and eliminated the app's major ANRs; also fixed a years-old top crash in the internal logging pipeline.
- Built the **country-switching foundations** and globalised core modules (address book, settings) that underpinned JET's later consolidation into a single global app.
- First engineer to integrate the PIE design system across the app, mentoring the chapter through the migration.
- Overhauled **observability** — Firebase custom keys, screen-tagged Kibana logs — and personally surfaced 8+ production incidents, including a silent 20% failure rate on the reviews endpoint. Rated “Exceptional” in 2020.

Android Developer — i-neda Ltd Sep 2017 – Jan 2019

Pari-mutuel (tote) wagering systems for horse racing Farnborough, UK

- Built an Android kiosk application used by staff at UK horse-racing tracks to accept tote bets, integrating with i-neda's OpenTote/MicroTote wagering platforms.
- Developed reactive real-time betting flows with RxJava, optimised for dedicated low-spec Android terminal hardware.

Independent Developer — Google Play & Microsoft Store 2013 – Present

- Began as a self-taught indie developer, self-publishing my own apps on Google Play (Anime Soundboard, Plenty of Cats, and others) — still maintained as a testbed for new tech.
- Currently self-publish two open-source C#/.NET WinUI apps on the Microsoft Store — [ADB Extension](#) (Android device management) and [Markets Extension](#) (market tracking) for PowerToys Command Palette — owning the full lifecycle solo, from design and development to packaging and store distribution.

Education

- **Physics (undergraduate studies)**, University of Athens, Greece.

Technologies and Languages

- **Languages:** Kotlin, Java
- **Android:** Jetpack Compose, Coroutines & Flow, Dagger Hilt, RxJava, WorkManager, MVVM/ViewModel, SavedState & process-death handling
- **Quality & CI:** GitHub Actions (previously Jenkins, TeamCity), Firebase Test Lab, Roborazzi, detekt & Android Lint (custom rules), SonarQube, Gradle modularisation, feature flags & staged rollouts, Datadog/Kibana observability
- **Beyond Android:** Myriads of web frameworks, Kotlin Multiplatform, C#/.NET WinUI 3, PowerToys Command Palette extensions, Chrome extensions.

Writing and Recognition

- Semi regularly featured in Android Weekly e.g “Injecting Composables with Dagger without losing it” featured in **Android Weekly #624** (2024).
- Android 16 Live Updates work at JET **featured at Google I/O** and shared by Google's official Android Developers channels (2025).
- 25+ technical articles since 2020 on *costafotiadis.com* and the JET engineering blog — Jetpack Compose internals, Dagger/Hilt, static analysis, Kotlin Multiplatform.

Other

- Earlier career in retail (UK & Greece, 2006–2011); national service, Greek Army (2011–2012).