

Eralp Karaduman

Senior Full-Stack & Mobile Engineer · Web · Native Mobile · Cloud Infrastructure

E-Mail	eralp@eralp.dev
Website	eralp.dev
Projects	eralp.dev/projects
LinkedIn	linkedin.com/in/eralpkaraduman
GitHub	github.com/eralpkaraduman

Engineer with 18 years shipping production software across the entire stack, now increasingly AI-augmented. Genuine depth in three areas that rarely overlap in one person:

- **Frontend & Backend Full-Stack:** React, TypeScript, 9+ years building and running production web apps end to end.
- **Native & Cross-Platform Mobile Apps:** 6+ years of native iOS (Swift, Obj-C) plus 7+ years of React Native and Flutter; apps shipped to the App Store.
- **Cloud & infrastructure:** designing and operating server and streaming systems on AWS and GCP, with hands-on CI/CD, Docker, Kubernetes and build tooling.

I ship independently, without hand-holding, and I build for the people who use what I make: I measure how each tool lands and improve it from there.

I build and customize agentic workflows and AI coding agents to ship faster across all of the above.

I also write at eralp.dev, post [screenshots and demos of my side projects](#), produce a [comedy podcast](#), and host live Twitch shows about software development.

Core Skills

Area	Technologies
Mobile	Swift, Objective-C, React Native, Flutter / Dart, native iOS & Android
Frontend	React, Next.js, Redux, TypeScript, JavaScript, SPA architecture
3D / Graphics	Three.js, React Three Fiber, Blender
Backend	Node.js, GraphQL, FastAPI, Pydantic, Socket.IO, PostgreSQL, MySQL, MongoDB, SQLite, Redis, Kafka
Cloud & Infra	GCP, AWS, Firebase, Cloudflare Workers, Cloudflare Pages, Coolify, Docker, Kubernetes, Microservices, Distributed Architectures, CI/CD, Observability (Prometheus, Grafana, OpenTelemetry), Mobile Ops, Fastlane, Jenkins
AI-Augmented Dev	AI coding agents, agentic workflows, prompt engineering, generative AI
Languages	TypeScript / JavaScript, Swift, Objective-C, Dart, Lua, Python

Work Experience

June 2022 - now **Starship Technologies**, Helsinki

Senior Software Engineer

- Full-stack engineer building and owning internal web, mobile, and backend tools (React/React Native, Node/TypeScript/GraphQL, Kafka) that run the autonomous delivery fleet across 8 countries; defined a success metric for an internal ops tool and grew its weekly active usage beyond that target.

June 2021 - June 2022 **Qvik**, Helsinki

Principal Software Engineer

- Led React Native TV companion games for YLE (Finnish public broadcaster) and, on my initiative, re-architected them to React Native Web for on-demand loading; built a Flutter investment app (Ovoro) and contributed to technical sales.

March 2019 - June 2021 **Qvik**, Helsinki

Senior Software Engineer

- Consultant engineer on client product teams, building React Native mobile apps and TypeScript/React web frontends; led the React Native TV companion games for YLE across client engagements before being promoted to Principal Software Engineer.

Jan 2017 - February 2019 **Seriously Digital Entertainment** (later acquired by Playtika), Helsinki

Technology Programmer

- Built a live-content/A-B-testing CMS (Java + React + Unity editor tooling), native Unity3D plugins, ELK ops dashboards on GCP/Kubernetes, and an on-prem Jenkins CI/CD pipeline with on-device performance testing.

Sep 2015 - Jan 2017 **DealDash**, Helsinki

iOS Developer

- Built features for the company's main iOS app (Swift/PHP) and Dropio, a real-time auction app (Swift + Node.js/Socket.IO + PHP/Symfony) with anti-cheat bid integrity and automated bidder bots.

Sep 2014 - Oct 2016 **CupsOn.Me**, Istanbul

Co-Founder & Mobile Developer

- Co-founded a mobile-first third-wave-coffee discovery & purchasing platform; built the iOS & Android apps with in-app payments (3D Secure) and in-venue code redemption; launched in Dublin, London, and Istanbul.

Apr 2015 - Aug 2015 **Hipo**, Istanbul

iOS Developer

- Built Chroma (Fieldguide), an Obj-C iOS social network for collectors with high-performance, memory-efficient image galleries. ([App Store](#))

November 2012 - Apr 2015 **Adphorus / Publik** (later acquired by Sojern), Istanbul

Mobile (iOS & Android) Software Developer

- Built native iOS (Obj-C) & Android (Java) clients for Adphorus (Facebook ads SaaS) and the Cevap.Tv video platform (Wowza HLS/FLV streaming on AWS, Flash/AS3 player with Google IMA ads, FFmpeg/Node.js transcoding).

Sep 2008 - Nov 2012 **MagiClick**, Istanbul

Senior Flash Developer & Team Leader

- Built web experiences with Flash, multiplayer advergames (Red5, Box2D, Papervision3D), interactive in-store installations with Arduino and sensors, and early FLARToolKit augmented reality; led a team of 6.

Education

2005-2009 **Kadir Has University, Computer Science & Statistics Ba.**

Projects

A deeper look at selected projects behind the roles listed above.

Full-time roles

Starship Technologies

Starship Technologies operates a large fleet of autonomous delivery robots across multiple countries. As a full-stack engineer I built the web, mobile, and backend systems its field and operations teams use to run the fleet.

Greenfield operations-management web platform · 2026

IC, OKR owner and main responsible · TypeScript, Hono, GraphQL, React, Redis, Terraform, Prometheus, Grafana, OpenTelemetry, Amplitude

- Built a read-only, mobile-first ops web app from scratch (backend, React UI, infra) giving site leads and customer-success managers a quick at-a-glance view of fleet status and targets across sites.

- Hardest parts: aggregating live data from several upstream services into one fast, real-time view, and standing up the service's observability (structured logging, Prometheus/Grafana dashboards, OpenTelemetry tracing).
- Defined the success metric and passed the adoption target I set: at least half of the roughly 20 key users active weekly within the quarter. It cleared that bar in the first month, with non-key users picking it up too, across 8 countries.

Mission Control: cross-site operational warnings dashboard · 2023–2024

IC, OKR owner and main responsible · React, TypeScript, GraphQL, built into the "Hype" hypervisor panel

- Before this, site leads and operations staff had no single view of operational warnings: understaffed sites, offline robots, fleet issues, road closures, and weather alarms were scattered and easy to miss.
- Built a real-time, per-service-area overview with at-a-glance red/green indicators across deliveries, tasks, online robots, staffing, merchants, fleet, closures, and weather alarms, letting ops spot problems across every site at once and drill into any area.
- Adopted by the control-center team; later succeeded by a separate read-only, mobile-first quick-glance tool (operations-overview, above), making me owner of the product line across two generations.

Mobile app re-platforming & delivery pipeline · 2022–2026

Led the re-architecture and built the CI/CD · React Native (Android), TypeScript, GraphQL, GitHub Actions, Docker, Detox

- Migrated a field app (~200 MAU, 8 countries) from a managed Expo setup to bare React Native + typed GraphQL when its native needs outgrew the managed workflow.
- Built CI/CD from scratch (builds, signing, e2e, OTA), replacing a fully manual process where local builds were hand-uploaded to the device-management portal, so the team shipped updates and bug fixes far more frequently.

Visual equipment inspection & damage-reporting system · 2023–2026

Full-stack engineer + interaction design · React Native (SVG-in-WebView diagram, video, barcode), Node/TypeScript, GraphQL

- Built a guided inspection flow around an interactive 2D diagram of the delivery robot: the inspector taps a part to select it, then enters damage details for that exact component, attaching video evidence, tracking spare parts, and auto-linking and de-duplicating related tickets.
- Hardest part: making the diagram interactive by embedding an SVG in a WebView inside the React Native app and building a custom bidirectional event bridge between WebView and React Native to keep part selection and damage state in sync across the two.
- Replaced ambiguous, duplicated free-form tickets with clear, structured repair reports, and rolled it out to inspectors in stages with feature flags and in-app training.

Hardware integration: native mobile printer plugin & returns (RMA) workflow · 2022–2023

Designed and built it · React Native native module (Java/Kotlin), USB/Bluetooth I/O, Node/TypeScript, GraphQL

- Built a custom native module bridging the app to a USB/Bluetooth label printer, handling device I/O at the native layer and Android-version print-compatibility issues, plus the returns

workflow and automatic return-order creation against the inventory system.

Charging-infrastructure management tooling · 2022–2026

Contributor (mobile/web/backend) · React Native, React, Node/TypeScript, GraphQL

- Full-stack tooling to provision, monitor, repair, and track the fleet's custom charging stations across the field app, web panels (alerts, energy-management), and backend (scheduling, charger service, inventory).

Certificate-based device sign-in for shared, externally-operated devices · 2025–2026

IC, main responsible · certificate/PKI auth, MDM-managed devices, React Native, Node/TypeScript

- Built a certificate-based alternative to SSO for shared devices without per-user identities: the in-app certificate sign-in flow plus server-side certificate validation and session issuance.
- Live in production, rolled out in stages across selected countries, sites, and key operators (currently enabled in a subset of countries).

Backend service powering a global field-operations app · 2022–2026

Contributor → owner · Node.js, TypeScript, Kafka, GraphQL, MongoDB, push notifications, Cloudflare

- Owned a backend service in an event-driven microservices architecture (Kafka for inter-service communication) powering a ~200-user, 8-country mobile app with real-time data, access control, and push notifications.
- Owned the service's authorization and infra/security.

Qvik

Consultancy work across several client engagements and also internal technical sales.

YLE (Finnish public broadcaster), creative content

Contributor · React Native, React Native Web, TypeScript, React, Redux, Clojure

- **Hyvät katsojat**: TV show companion game built in React Native, embedded inside the yle.fi React Native TypeScript app. Contributed to the CMS backend in Clojure.
- **Elämäni biisi**: TV show companion game built in React Native, embedded inside the yle.fi React Native TypeScript app. Contributed to the CMS backend in Clojure.
- **React Native → React Native Web (my initiative)**: when the client wanted the games removable between show seasons to cut the app's asset footprint, I proposed and built the move from React Native into a React Native Web shell, turning each game into a standalone web app embedded back into the yle.fi app as a WebView so its content loaded only on demand without bloating the host app. I built the WebView-to-native bridge that let the embedded web game communicate with the host app, and the same approach was reused to build later apps.
- **"Tehtävä" quiz / voting / form tool**: Clojure backend with a TypeScript web frontend (React + Redux) driving a complex state system. The frontend is embedded into multiple web apps and also used as an internal tool, and the backend doubles as the backend for the TV game companion apps.

Ovoro, investment app

Contributor · Flutter, Provider, Firebase

- Built a investment app from scratch in Flutter using Provider and InheritedWidgets, helping

the client bootstrap their app business.

- Set up Firebase auth and notifications, the first-time user experience, and screen flows for critical user actions.

Technical sales

Contributor (acting tech lead on the engagements) · Internal Qvik work

- Worked on technical sales: project scope, time estimates, and offer pricing for a major global retail business and a major bank in Finland.

Seriously Digital Entertainment

Live-content & A/B testing platform (CMS) for the game

Full-stack + Unity tooling · Java, React, Unity3D (AssetBundles, ScriptableObjects, editor tooling), Docker, Kubernetes, GCP

- Built a CMS in two parts: a full-stack Java server with a React web frontend, and Unity-side tooling with its own custom editor tool windows inside the game project.
- Authored seasonal content and A/B test variants (both UI and code behaviour) as Unity ScriptableObjects packaged into AssetBundles; the tooling serialized them and pushed them to the web app, which served them on request so the game could sideload and deserialize them at runtime, shipping new content and experiments live without an app-store release.
- Built the scheduler that decided which variant to serve, with every available variant and experiment registered and managed through the web UI.
- Hardest parts:
 - Reliably serializing and deserializing the AssetBundle and ScriptableObject variants across the Unity-to-web boundary.
 - Designing the scheduler and the variant/experiment management UI, and building the whole pipeline end to end.
 - Running the mobile game headless in the cloud to test variants against it, which meant containerizing a headless Unity build.

Native Unity3D plugins

Contributor · Unity3D, Objective-C (iOS), Java (Android)

- Developed native plugins bridging Unity3D to iOS (Obj-C) and Android (Java) platform features.

Operations dashboards

ELK stack, GCP, Kubernetes

- Built operations dashboards on the ELK stack running on GCP and Kubernetes.

On-prem CI/CD & on-device performance testing

Jenkins (Groovy), GameBench, on-prem build servers

- Built and ran the on-prem CI/CD system on Jenkins, writing extensive custom pipelines in Groovy.
- Ran GameBench performance tests on real phones physically connected to the on-prem build servers.
- Managed scheduled nightly, ad-hoc, and release builds.

DealDash

DealDash mobile app

Contributor · Swift, PHP

- Worked on the company's main mobile app in Swift, contributing to multiple new features and its PHP backend.

Dropio: real-time auction app

iOS + realtime backend · Swift, Node.js, Socket.IO, PHP / Symfony

- A new-business experiment: a standalone app with live, real-time auction bidding, built in Swift with a PHP/Symfony backend and a Node.js + Socket.IO realtime layer.
- Hardest parts:
 - Building and scaling the Node.js/Socket.IO server to broadcast live auction state (price and countdown) to all connected clients in real time.
 - Enforcing bid integrity to prevent cheating: locking a bid once submitted and rejecting any bid placed after the auction had ended.
 - Running many parallel auctions concurrently.
 - Building automated bidder bots that participated in auctions alongside real users.

CupsOn.Me

Mobile-first coffee discovery & purchasing platform

Co-founder (one of two technical founders) · iOS (Objective-C), Android (Java)

- Co-founded the startup and built both native clients from scratch: the iOS app in Objective-C and the Android app in Java, a mobile-first platform for discovering and buying from third-wave coffee shops.
- Built in-app credit purchasing through a local payment provider: users entered card details and completed a 3D Secure verification flow (ahead of its time), then spent the credits at the venue.
- Built the in-venue redemption flow: the user presented a keypad confirmation screen and the venue confirmed the spend by entering their own code.
- Maps-based shop discovery was the most advanced piece of the UI.
- Shipped to the App Store and Google Play and rolled out to real coffee shops across Dublin, London, and Istanbul.

Hipo

Chroma (Fieldguide), iOS app for collectors

iOS developer · Objective-C

- Built Chroma (tagline "Fieldguide"), an Objective-C iOS social network for collectors, with image-heavy galleries and polished UI ahead of its time.
- Hardest parts:
 - Keeping long, image-heavy lists fast and memory-efficient by rendering content just ahead of where the user was about to scroll and aggressively recycling view objects to stay within memory limits.
 - Managing complex UI state across many screens, including deep drill-down navigation from top-level categories into sub-categories and individual species.

Adphorus / Publik

Cevap.Tv, video how-to platform

Mobile + video streaming infrastructure · Objective-C (iOS 6), Flash/AS3, Wowza, HLS/FLV, SMIL, Node.js, FFmpeg, AWS (EC2, S3)

- Built the iPad app for Cevap.Tv, a YouTube-like video platform, with a two-column layout in Objective-C for iOS 6, which was technically challenging for its time.
- Built the video streaming backend on Wowza: a multi-bitrate adaptive HLS and FLV streaming server driven by SMIL configurations, deployed on AWS.
- Built a custom Flash/AS3 video player and integrated Google's video ads (IMA) SDK into it.
- Built a companion video-processing service in Node.js that used FFmpeg to stitch intros onto source videos, transcode them into multiple bitrates, store the outputs in S3, and generate the SMIL configurations; deployed with pm2 on AWS EC2.

Adphorus, Facebook ads management SaaS

Mobile developer · iOS (Objective-C), Android (Java)

- Built the native iOS and Android clients for Adphorus, a SaaS platform for managing Facebook ad campaigns, focused on campaign monitoring with charts and adjustable bid-rate controls.

MagiClick

Senior Flash developer, later team lead of 6 (2012). Built websites, advergames, and rich-media campaigns.

Multiplayer Flash advergames

Senior Flash developer · ActionScript 3, Red5 (multiplayer server), Box2D, Papervision3D

- Built multiplayer advergames on a Red5 game server with matchmaking, lobby, chat, and leaderboards, where players competed for prizes over a campaign period.
- Built the games in AS3, some using Box2D physics and some using Papervision3D for 3D graphics.

Interactive in-store electronics experiences

Hardware + software · Arduino, sensors, Flash

- Built rich-media experiences combining Arduino and sensors with software, deployed into grocery stores as interactive installations.

Early augmented reality / marker-detection experiences (FLARToolKit)

FLARToolKit, Papervision3D, ActionScript 3

- Built early augmented-reality experiences with FLARToolKit (the first Flash-based Augmented Reality library), doing webcam marker detection well ahead of its time.
- One registered marker patterns for 29 letter stickers so players could complete a collection by showing each sticker to the webcam.
- Another recognised a marker and mounted an animated 3D object (Papervision3D) onto the live camera view where the sticker appeared, like a toy.

Freelance

Arca

Freelance Software Consultant · Helsinki (hybrid) · Nov 2024 – July 2025

Arca (Lifeline Ventures-backed) is a content-curation app where people collect content into personalized portals. As the mobile engineer on the team I bootstrapped the React Native app and its CI/CD in about six weeks, built its share-sheet extension, and helped with the GCP backend's scheduled tasks.

Cross-platform mobile app & share-sheet extension

Freelance mobile engineer · React Native, TypeScript, iOS & Android

- Built the React Native app shipped to iOS & Android, including a share-sheet extension that lets users save content into Arca from any other app.
- Built the share-sheet UI in React Native too, running it from a second, slightly modified entrypoint in the same codebase rather than as a separate native UI.
- Hardest parts:
 - Holding the user's session inside the extension and refreshing it there, then syncing the fresh credentials back so the main app stayed signed in, sharing credentials between the extension and app via iOS app groups and a shared keychain.
 - Staying within the iOS share extension's tight memory budget so the OS did not kill the React Native runtime mid-share.
 - Normalizing the different content types the OS passes in (URLs, text, images) into one save flow.

Mobile release infrastructure & backend support

React Native, Fastlane, CodePush (OTA), GCP

- Set up mobile CI/CD with Fastlane, automating iOS & Android release pipelines, including building and signing the extra share-extension target and its separate entrypoint.
- Configured push notifications and the Apple Developer account from scratch.
- Set up over-the-air (OTA) updates with CodePush.
- Helped build background task queues in the GCP backend.

Hype Hype / Frogmind

Freelance engagement (similar in shape to Arca), building app UI on top of a custom game engine.

Mobile game-creation & creator-economy app · Jul 2020 – Sep 2020

Contributor · Flutter (forked, embedded in a custom C/C++ game engine)

- Worked on Hype Hype, a mobile game-creation and creator-economy app, building UI in Flutter on top of the client's custom game engine, with Flutter embedded into the engine.
- Built critical UX flows such as content moderation and content-reporting UI.
- Hardest parts:
 - Working against a non-conventional Flutter setup: a fork of Flutter that was hard to keep in sync with mainline, requiring work down in the C/C++ toolchain.
 - Cross-communication between the Flutter UI layer and the C engine.
 - Managing UI state in Flutter without a persistent Flutter instance (unlike a regular Flutter app, the engine did not keep one alive), which made holding state across the UI complex.

Speechgrinder

The client, Speechgrinder, later became Speechly and was acquired by Roblox.

React Native speech-recognition native module (iOS) · Jan 2017 – Feb 2017

Freelance · React Native native module, Objective-C, iOS SFSpeechRecognizer

- Built a React Native native iOS module that let the client's React Native app use iOS's built-in speech recognizer (SFSpeechRecognizer) directly from JavaScript.
- Streamed recognition results from the native layer to the React Native side in real time over an event-based bridge.

Side Projects

Screenshots and short demos of these are at eralp.dev/projects.

LinkedFin: multilingual fish-name etymology database · Jan 2025 – present

React 19, TanStack (Start/Router/Query), TypeScript, SQLite-WASM, Kysely, Fuse.js, Cloudflare Pages + Pages Functions + D1, Vite, Tailwind ([source](#))

- A backendless single-page app: the entire fish-name etymology database ships as a SQLite file that the browser loads and queries fully client-side via the official SQLite WASM build (Kysely for queries, Fuse.js for fuzzy search), so all search and browsing happen in the frontend with no server or API.
- Solved per-page link previews, which a JS-only SPA normally cannot have because crawlers do not run the client code: a Cloudflare Pages Function injects per-page Open Graph / Twitter meta tags via HTMLRewriter, and a separate function renders the preview image dynamically at the edge (workers-og).
- Backed the OG data at the edge with a Cloudflare D1 database, kept in sync with the source SQLite database by a CI step that replicated it into D1; later simplified the pipeline to a build-time JSON snapshot bundled into the functions.

Text Wallpaper: text-to-image wallpaper generator · Jan 2015 – present

JavaScript SPA, iOS (Cordova/WebView wrapper), AdMob + Facebook Ads ([source](#))

- A personal side-project single-page app for generating text wallpapers, with a sizeable audience (~3,000 monthly active users).
- Later shipped an iOS counterpart that wrapped the web app in a WebView and integrated AdMob and Facebook Ads SDKs for monetization.

Card Quest: a card game shared across mobile & web · Jan 2021 – Mar 2021

React Native, react-native-web, TypeScript, Metro + Webpack, styled-components, react-navigation + react-router-dom ([source](#), [write-up](#))

- An open-source single-player card game (Donsol) running on iOS, Android, and web from one React Native codebase, built to explore practical code-sharing between React Native and React web.
- Ran both bundlers in a single project (Metro for native, Webpack for web with `react-native` aliased to `react-native-web`), splitting platform-specific code by `.native.ts` / `.web.ts` postfixes and keeping everything else shared.
- Decoupled the game logic from React entirely as plain TypeScript (GameController, EventDispatcher, Observable), bound to the UI through a Context provider and hook, and kept platform-native navigation (react-navigation on mobile, react-router-dom on web) behind shared link/screen abstractions.

Super Damage Games: indie game-jam handle · Sep 2013 – present

Game jams, Ludum Dare ([Rockmorse](#))

- My indie game-development handle for game jams and experimental side projects.
- Featured project, Rockmorse: an astronaut stranded on Mars, with only one-way communication to Earth, arranges Martian rocks into Morse code for passing satellites to read his replies; built for Ludum Dare and open-sourced, inspired by Andy Weir's The Martian.

Updated August 11 2026. [Get latest](#)