Grzegorz Otto



SENIOR SOFTWARE ENGINEER · MOBILE TECH LEAD · TYPESCRIPT / REACT NATIVE, GO, C++

Warszawa, Poland (CET) · +48 501 401 706 · grzott@gmail.com
grzegorzotto.dev · linkedin.com/in/grzegorz-otto

Polish (native) · English (B2/C1 written & technical; conversational spoken) · B2B or employment from Jan 2027 · Warszawa / EU-remote / global-remote

SUMMARY

Senior React Native engineer and mobile tech lead who ships production apps end-to-end - system design, custom native modules in Swift + Kotlin, CI/CD, and store delivery. Lead developer for Poland's public broadcaster, **Polskie Radio**: I architected its flagship app (**~60,000 daily active users, 150,000+ downloads, 99.9% crash-free**) and authored **pr.playback.service**, the cross-platform (Web + native iOS/Android) playback engine behind its app family, now re-architected into a shared Turborepo monorepo. I also architect **Flare Engine**, an open-source real-time 2D engine for React Native + Web, a **C++ gameplay core on Unreal Engine 5.7**, and a **Go** agent-harness CLI. I sustain high velocity as a single owner across large, strictly-bounded codebases, working with an AI-directed agent workflow in which agents implement and I own the architecture, the gates and the review.

Earlier, I spent seven years (2012–2019) leading regional sales at Deutsche Bank Polska - coordinating seven teams and running nationwide training through the Santander acquisition - before retraining into software. That background is why I default to ownership, mentoring, and shipping under deadline rather than just writing code.

TECHNICAL PROFILE

Languages TypeScript (strict mode), Swift, Kotlin, C++, Go, JavaScript

React Native RN 0.66 → 0.81 (New Architecture / Fabric), Expo SDK 55, Expo Router 6, Expo Modules API, TanStack Query 5, Zustand, MMKV, Reanimated, REST APIs, WebSockets

Native Expo Modules (Swift + Kotlin): OS media session (CarPlay / Android Auto / lock-screen), Google Cast, AVAudioSession / Android MediaSession, Foreground Service, PiP

Architecture Clean / Hexagonal, event-driven, state machines, command bus, dependency injection, Turborepo monorepo, ECS

Graphics & performance React Native Skia (GPU atlas rendering, RuntimeShader / SkSL), zero-allocation hot loops, on-device frame profiling; Unreal Engine 5 (C++)

DevOps & CI/CD Bun, Turborepo, Biome, Changesets, GitLab CI + Fastlane (iOS + Android), Expo config plugins, Sentry

Developer tooling Go CLI (Cobra) for a provider-agnostic agent harness: declarative gate definitions, a gate-proving runner that verifies each gate can actually fail, evidence ledgers

Practice tech leadership & mentoring, test-first, AI-directed development: agents implement against written contracts and scoped directories; I own the architecture, the gates and the review

EXPERIENCE

Polskie Radio — Mobile Apps · Senior React Native Engineer

Apr 2020 – Sep 2026 · Warszawa

App Store · Google Play

- Sole architect and lead developer of the flagship **Polskie Radio** app (Poland's national public broadcaster): **~60,000 daily active users** (40k Android / 20k iOS), **150,000+ downloads**, **99.9% crash-free**, and ~99% backend uptime (Sentry). Hardened the live-data layer with exponential-backoff retry over REST across 10 stations, replacing WebSockets for reliability.
- Authored **pr.playback.service** - a cross-platform (Web + native iOS/Android) audio & video playback engine that is the architectural core of the flagship and the 2.0 app family: event-driven playback, an 8-state state machine, a serialized command bus, and swappable audio / video / Cast adapters.
- Developed **2 custom Expo native modules** in Swift + Kotlin: OS media-session controls (CarPlay, Android Auto, lock-screen) and Google Cast with position-synced handoff.
- **Re-architected the app family around a shared Turborepo monorepo**: Jedynka 2.0 and Trójka 2.0 ship from one codebase (~95% shared packages - API client, presentation layer - differing only by branding and feed); the flagship app was next.
- **Tech lead and mentor** for a 2-3 person mobile team before taking sole ownership of the multi-app portfolio in 2024; set the React Native architecture, code standards, and CI/CD across the apps (RN 0.66 → 0.81).
- Automated CI/CD end-to-end: GitLab CI + Fastlane with 7 custom Expo config plugins, tag-triggered builds, staged Google Play rollout, and App Store submission. Stack: Expo Router 6, TanStack Query 5, Zustand, MMKV, Unistyles 3, Sentry, GA4 + Gemius.

Earlier career - pre-engineering.

Regional Sales Coordinator · Deutsche Bank Polska S.A.

Apr 2012 – Apr 2019 · Warszawa

- Owned the sales volume of a region spanning **7 teams** - full accountability for regional results, plus the onboarding, training, and ongoing coaching that delivered them.
- Designed and ran **nationwide training during the Santander acquisition of Deutsche Bank Polska**, coordinating a consistent rollout across sites through a large-scale banking M&A integration under a hard cutover deadline.

PERSONAL / R&D PROJECTS

Flare Engine — Real-time 2D rendering & animation engine for React Native + Web

2026 – present · open source

Open source (Apache-2.0) · public on npm and GitHub since Oct 2026

- Architected the engine as **42 packages** (`@flare-engine/*`) in a Bun + Turborepo monorepo (29,065 LOC, Nov 2026): strictly downward-only dependencies, enforced in CI, with **zero circular dependencies**. Bootstrapped the architecture in 12 days (303 commits).
- Built a **bitmask-query ECS** (~2K entities, escape hatch to 5K; **9.46 µs / 1,000-entity step**). **Zero-allocation hot paths** (Vec2 plain objects, static output-param math) and generator-based coroutines keep the loop GC-free.
- Designed a Skia GPU renderer with **two render modes** - retained-mode atlas batching via imperative `canvas.drawAtlas` and immediate-mode `PictureRecorder`; post-FX via a JSX `RuntimeShader` with **zero offscreen surfaces / readbacks**. The same GPU pipeline drives any animation-heavy RN UI, not only games.
- **Benchmarked on real devices** (Galaxy A55, Pixel 6, Pixel 10): frame p95 0.9–9.5 ms across boot, boss-fight, particle-storm, and zone-transition scenarios (worst cell Pixel 10 particle-storm 9.5 ms vs the 16.67 ms / 60 fps budget).
- **Open-sourced in Oct 2026**: all 42 packages public on npm, published from GitHub Actions with provenance, plus a scaffolder (`npm create @flare-engine`) with three starter templates; engine suite past **3,400 passing tests**. Consumed by *Pan Tvardowski* (below) and by grzegorzotto.dev. Drove the build via an **AI-directed workflow** (custom agents + skills, plan → build → verify).

The Dark Orb — C++ core-systems owner · Unreal Engine 5.7

2026 – present · alpha

Personal / team project · in development (not yet public)

- Own the **deterministic C++ gameplay core** (`DarkOrbCore`): an ECS + command-bus + reducer architecture with deterministic RNG, save/load, and damage / skill / status pipelines - UE-integrated but engine-portable and unit-tested with **property-based tests (rapidcheck)**.
- The same ECS + command/reducer discipline as Flare Engine, now in **C++ on Unreal Engine 5.7**; co-built with a small team (presentation + content by collaborators).

Pan Tvardowski — Showcase game built on Flare Engine

2026 – present · own repo (1,333 commits)

- Vertical shoot-'em-up themed on Polish folklore - the engine's proof of concept and hardest stress-test (games push 2D rendering to its limits).
- **8 zones**, **16 enemies**, **8 multi-phase bosses**, **4,224 passing game tests**, fully localized EN/PL; targets **iOS, Android, and Web** from one codebase.

WRITING

Technical blog on game-engine architecture, ECS, and rendering performance - **grzegorzotto.dev/blog** (English and Polish; shared on LinkedIn)

EDUCATION & TRAINING

Master's degree in Economics — Vistula University, Warszawa · 2011 – 2016

JavaScript & React Bootcamp — Coders Lab International · 2019 – 2020

Test, commit and line counts measured 2026-11-05 (Flare Engine cb7e077, Pan Tvardowski 3053bd9); other project figures 2026-08-30, except the October 2026 launch figures.