

EFE ATASAYAR

Full-Stack Software Engineer & Game Developer · Founder, Lector Interactive

efe@efeatasayar.com | github.com/lectorsalamanca | lectorinteractive.com | moonlite.gg | efeatasayar.com |
İstanbul, Türkiye · open to remote / relocation

SUMMARY

Full-stack software engineer and game developer (4+ years hands-on) and founder of **Lector Interactive**, a one-person studio where I own products end-to-end: architecture, backend, real-time networking, 3D graphics and audio, web front-ends, infrastructure, security and operations. In production today: **MoonLite**, a ~53,000-line TypeScript multiplayer 3D web game with custom netcode and its own audio engine (live at moonlite.gg, Steam edition in progress), and **Horus**, a ~41,000-line real-time OSINT platform with a multi-agent LLM harness. Largely self-taught and entrepreneurial: a fast learner and hard worker whose core strength is **problem-solving**, bringing a creative, unconventional perspective to product and engineering. Strongest in TypeScript/Node.js, Java, C#, Python and React/Next.js; fluent English, native Turkish.

TECHNICAL SKILLS

Languages: TypeScript, JavaScript, Java, C#, Python, C, Dart, GLSL, SQL, Solidity, HTML/CSS
Frontend & Web: React, Next.js (App Router), Vite, Tailwind CSS, Three.js / WebGL, Web Audio API & AudioWorklets, Web Workers, WebCrypto, IndexedDB, mobile web / touch UX, i18n, SEO, Electron
Backend & Infrastructure: Node.js, WebSockets, Python (FastAPI), REST/SSE APIs, PostgreSQL, SQLite, DuckDB, Cloudflare (Workers, Durable Objects, Pages, R2, D1, KV, Access, Tunnel), Docker Compose, Linux VPS ops (systemd, Caddy, ufw/fail2ban, SSH hardening), rolling deploys, Firebase
Real-Time & Games: custom multiplayer netcode (binary protocols, tick-based relays, area-of-interest culling, delta snapshots, clock sync, deterministic sims), Unity (C#, Netcode for GameObjects), jMonkeyEngine (Java), Steamworks (auth, overlay, P2P voice, microtransactions), voxel engines, procedural generation, audio DSP (WSOLA, synthesis)
Security & AI: OAuth 2.0 / PKCE, RBAC, rate limiting, CSP/HSTS, threat modelling, E2E encryption (X25519 + ML-KEM-768 hybrid, AES-GCM); designing and building LLM agent harnesses from scratch (typed tool use, sub-agent orchestration, memory & context compaction, permission-gated actions), OpenAI / OpenRouter / Ollama APIs, custom agent-driven development workflows (own agents and sub-agents controlling the full system, incl. Claude Code)
Tooling & Practice: Git/GitHub (signed commits), Vitest, TypeScript strict mode, npm monorepos, Gradle, JVM profiling (JFR, ZGC tuning), load testing

PROFESSIONAL EXPERIENCE

Lector Interactive · Founder & Lead Software Engineer 2024 – Present
Independent software & game studio, İstanbul (incorporated as *Lector Interactive Yazılım Ltd. Şti.* in Jul 2026; sole owner).
Solely responsible for product, engineering, infrastructure, security, legal/compliance and client delivery.

MoonLite · multiplayer 3D “after-hours park”: music, cinema, mini-games Jun 2026 – Present · Live · Steam upcoming
TypeScript (strict) · Three.js · Web Audio / AudioWorklet · Node.js WebSocket relay · Cloudflare Workers / Durable Objects / Pages / R2 / D1 / KV · Linux VPS fleet · Electron + Steamworks · Next.js

- Solo-designed and built a ~53,000-line strict-TypeScript monorepo (Three.js client · shared binary protocol · Node.js relay · Electron shell) plus a 5,400-line Next.js landing site. First commit to public launch in ~3 weeks; live at **moonlite.gg** since Jul 2026, 418 automated tests, ~390 signed commits.
- Cross-platform from one codebase: runs in any modern browser on Windows, macOS, Linux, iOS and Android (dedicated touch layer: virtual joystick, gestures, phone GFX profile) and ships as an **Electron + Steamworks desktop build** (Steam overlay verified in-process, system-browser OAuth via loopback code exchange) for the upcoming Steam release.
- Custom netcode: three isolated planes (reliable control · lossy int16-quantized presence at a 15 Hz tick with area-of-interest spatial hashing and delta snapshots · batched reactions), 7-byte inputs, ~62 binary message types, per-plane token buckets and back-pressure. Load-tested on a €4/month 2-vCPU VPS: 480 concurrent players at 1.5–2.5 ms worst tick (~25× headroom), zero drops.
- “Bytes never touch the server”: content-addressed media (SHA-256 ids) resolves local IndexedDB cache → Cloudflare R2 at zero egress; a shared transport clock keeps every client’s playback aligned. Built the browser audio engine from scratch: multi-stem mixer with Ableton-style auto-warp, WSOLA pitch-preserving time-stretch in Web Workers, AudioWorklet synth with FX bus, and a beat-grid arrangement DAW (waveform clips, piano roll, collaborative editing).
- Rendering & simulation: voxel avatars, single-draw-call instanced crowd, static scenery batched from ~900 to ~12 draw calls; 12+ networked attractions (chess, 8-ball, bowling, mini-golf, pinball, go-kart races, rides, synced YouTube cinema) built as pure, deterministic, unit-tested sims synced via tick-stamped inputs; no frames on the wire (late-join snapshots as small as 54 bytes).
- End-to-end encrypted DMs with a hybrid post-quantum scheme (X25519 + ML-KEM-768 → HKDF → AES-256-GCM, per-device keys, fresh KEM per message) pushed through a Cloudflare Durable Object; the server stores only opaque envelopes.

- Owned production end-to-end: Cloudflare Pages + a ~68-route Worker API (D1/KV/R2), multi-VPS relay fleet with rolling drain-based deploys and client auto-update, Google OAuth, server-side alpha gate, origin lockdown, CSP/HSTS, hardened systemd/SSH, admin console behind Cloudflare Access with audit log, DMCA/moderation tooling; wrote the threat model, EN + TR GDPR/KVKK legal set and Steam monetization plan.

Horus · real-time OSINT & situational-awareness platform with an AI agent harness 2026 · Private beta
[Python](#) / [FastAPI](#) · [Next.js](#) · [React](#) · [TypeScript](#) · [PostgreSQL](#) · [DuckDB](#) · [Docker Compose](#) · [Caddy](#) · [Cloudflare Tunnel](#)

- Solo-built ~41,000-line full-stack platform: a live Cesium 3D globe fusing 8+ feeds (ADS-B aircraft, AIS ships, satellite TLEs, USGS/EMSC seismic, GDACS disasters, sanctions lists) with cross-source entity resolution, entity graph, alerting and time-scrubbable replay; 87 REST/SSE endpoints, 32-table multi-tenant schema.
- Streaming analytics (incremental spatial/temporal co-travel detection at ~O(k) per observation minting graph edges live, cross-feed spoofing detection, deterministic 0–100 reliability scoring) plus a multi-agent LLM harness: 28 typed, schema-validated tools (mutating actions gated by user confirmation and audit-logged), parallel tool execution, sub-agent orchestration, three-layer memory with context compaction, pluggable LLM providers (OpenAI, OpenRouter, local Ollama, bring-your-own-key endpoints).
- Hardening: Google/X OAuth with PKCE, four-role RBAC, per-feature token budgets with atomic settlement, SSRF guards, two red-team passes, a 328-case agent regression suite (56 adversarial); load-tested at 355 rps (p99 311 ms, zero errors) behind a zero-inbound-port Docker / Caddy / Cloudflare Tunnel deployment.

Project Persei · co-op space-survival voxel sandbox built on real exoplanet data 2026 – Present · In development
[Java 21](#) · [jMonkeyEngine 3](#) · [GLSL](#) · [Gradle](#)

- ~35,000-line engine-level project built solo on open-source jMonkeyEngine: streaming 32³-chunk voxel world (cross-chunk face culling, procedural terrain/biomes, fluids, power circuits, crafting/farming/weather), ships built block-by-block and carved into flyable meshes with seamless surface-to-space transitions, and all 6,278 confirmed NASA exoplanets rendered at true galactic coordinates with physics-based appearance rules; runtime-generated textures, fully procedural audio, custom GLSL water shaders.
- Measured performance work: removed per-frame allocation hot paths (worst case 750k objects/s), tuned generational ZGC from JFR profiles, batched chunk remeshing, delta-based atomic save system.

Debait · 4–22-player online debate party game with live voice chat Steam release planned Mar 2027
[Unity](#) · [C#](#) · [Netcode for GameObjects](#) · [Unity Relay/Lobby](#) · [Steamworks](#) · [Node.js](#) · [Firebase](#)

- Server-authoritative multiplayer with a custom real-time voice and music streaming pipeline over Steam P2P (Opus, ring-buffer playback), zone-based and hand-drawn secret-ballot voting (compressed stroke data), an AI fact-checker, and Steam microtransactions/inventory/stats on a self-built Node.js + Firebase backend.

Production websites · [lectorinteractive.com](#) · [moonlite.gg](#) · [haweddings.com.tr](#) (client) 2026 · Live
[Next.js](#) · [React](#) · [TypeScript](#) · [Tailwind CSS](#) · [Vercel](#) · [Cloudflare Pages](#)

- Designed, built and shipped three production sites end-to-end: a 5-language (EN/TR/DE/ES/ZH) studio site with Medium-RSS articles, dynamic OpenGraph images and a Resend contact pipeline; a static game landing page (Three.js night sky, in-browser synth, GDPR/KVKK pages); and a bilingual client site for an İstanbul wedding studio with theme switching, video galleries and full SEO.

Pera Bilişim Teknolojileri A.Ş. · Sales Associate (IT products) Jan 2023 – Oct 2023

- Managed B2B/retail customer relationships; advised clients on hardware/software configurations matched to their technical needs.

Armil Bilgisayar · Sales Associate & Developer Jun 2020 – Dec 2022

- Combined technology sales with in-house development, delivering internal tools that supported daily operations.

Freelance · Web3 developer · coach · tutor (Fiverr / private) 2021 – Present

- Custom tokens and smart-contract integrations for international clients (2023 – present); 1-on-1 esports coaching (Rocket League) with consistently positive reviews (2022 – present); STEM tutoring for high-school students (2021 – 2023).

ADDITIONAL PROJECTS

- Rhyme Finder**: cross-platform multi-language rhyme app (Flutter/Dart) with language-specific phonetic handling (2025 – 2026). **Minecraft Java server plugins**: numerous gameplay and server-management plugins shipped for live servers.

EDUCATION

Koç University, İstanbul · Computer Engineering (B.Sc. programme, not completed) 2018 – 2023

Completed core coursework (Data Structures, Algorithms, OOP, Software Engineering, Computer Graphics, Databases, Operating Systems, Discrete Math, Linear Algebra, Probability & Statistics), then left before graduation for financial reasons to pursue software/game entrepreneurship full-time.

VKV Koç Özel Lisesi, İstanbul · High School Diploma 2013 – 2018

LANGUAGES & INTERESTS

- Languages**: Turkish (native), English (fluent). **Interests**: competitive ballroom dancer (medalist: Tango, Quickstep, Waltz, Foxtrot); table-tennis medalist; multi-instrumentalist, music producer and vocalist.