

Sinan KARAKAYA

Software Engineer | Systems, Graphics & Product Engineering

Rennes, France | EU work authorization | +33 7 81 98 47 83 | karakaya.sinan@proton.me
sinankarakaya.fr | github.com/Sinan-Karakaya | linkedin.com/in/sinan-karakaya-a9355b228

Product-focused software engineer building performance-sensitive software across operating systems, hardware integration, graphics, mobile and backend systems. Production experience at Canonical and OVHcloud, with hands-on work in Swift/Metal, C++, Rust, ARM64, FPGA/NPU enablement and low-level tooling. Ranked #1 of approximately 800 students nationally during the first two years at EPITECH.

EXPERIENCE

Partner Engineer | Canonical Ltd.

Remote | Mar 2026 - Present

- Lead development of FPGAd, a Rust system daemon that enables containerized applications on Ubuntu Core to securely configure FPGA bitstreams without bypassing system isolation. Drove the project from alpha toward release while expanding device support, reliability, security and day-to-day usability.
- Integrated support for all Qualcomm NPU platforms into Canonical's inference stack, enabling accelerated inference on Qualcomm hardware and designing the platform abstraction so additional silicon families can be integrated without architectural limits.
- Maintain, audit and ship more than 10 FPGA-related Ubuntu packages. Work directly with upstream software providers and hardware partners to understand software behavior, resolve packaging or compatibility issues, and deliver production-ready updates through Ubuntu.
- Built internal tooling to streamline recurring packaging and upstream-integration work, reducing manual friction when onboarding or updating heterogeneous third-party software.
- Collaborate across partner, industrial and silicon-focused engineering teams, working at the boundary between operating-system services, hardware enablement, containerization, security and application-level product requirements.

AI/ML & Backend Developer Intern | OVHcloud

Rennes | Feb 2024 - Jun 2024

- Built an AI-assisted support product in Python and Go for the domain-name support organization, integrating more than 10 internal data sources and APIs with classifiers, LLMs, MCP services and GraphRAG-based knowledge retrieval.
- Integrated the system into the production ticket workflow so every newly created ticket - approximately 20 per day - was automatically analyzed for a 5-6 person support team, providing structured diagnostic context as incidents arrived.
- Implemented ingestion and sanitization across heterogeneous and legacy data sources, including services that required complex proxying and normalization before their data could be safely consumed by downstream models.
- Worked end-to-end across dataset preparation, classifier training and deployment, backend orchestration, LLM-assisted synthesis, frontend/chatbot integration and production workflow adoption.
- Automated continuous knowledge refresh through scheduled data workflows so retrieval-backed answers remained aligned with changing internal documentation and operational data.

Fullstack & Mobile Developer Intern | Ecosystem Restoration Standard

Paris | Jul 2022 - Dec 2022

- Built web and mobile product features used to monitor ecosystem-restoration projects in the field, where operators frequently worked with poor or unavailable network connectivity.
- Implemented offline-first local storage and synchronization behavior designed to recover cleanly from conflicts when devices reconnected to the backend.
- Worked on map-heavy GeoJSON creation, editing and review workflows, with a strong focus on performance and battery usage on mobile devices.
- Contributed across frontend, mobile and backend concerns in a fast-moving product team, coordinating technical decisions with other developers and operational stakeholders.

SELECTED PROJECTS

Swift / Metal Game Engine - Apple Platforms | Swift, Metal, macOS/iOS, real-time graphics

- Designed and implemented a native real-time game engine in Swift and Metal targeting Apple platforms, using the project to learn Apple's graphics stack deeply rather than only at the application-framework level.
- Built a PS1-inspired renderer while relying on modern Metal 3-era GPU techniques underneath, separating the visual style from the actual rendering architecture.
- Worked with performance-sensitive Swift code, memory-management behavior, frame-time constraints and GPU resource management required by real-time applications.
- Explored Apple platform integration beyond rendering, including application packaging, platform-specific behavior and game-oriented system integrations.
- The long-term target is a polished playable demo inspired by the technical presentation of Metal Gear Solid, turning the engine into a complete product rather than only a rendering prototype.

SELECTED PROJECTS - CONTINUED

PS2 Static Recompiler | C++, MIPS, x86-64, ARM64, SDL-GPU

- Built a static recompilation pipeline capable of translating PlayStation 2 game code from MIPS into native x86-64 and ARM64 executables.
- Reached full-game recompilation rather than isolated function translation, requiring handling of program structure, platform behavior and the runtime assumptions made by original PS2 software.
- Designed a hook-based architecture that allows recompiled games to be modified, enabling mod capabilities and custom behavior without changing the original game binaries directly.
- Used SDL-GPU for portable graphics/runtime integration so recompiled games can target modern desktop architectures while remaining independent from the original console environment.
- The project combines low-level binary understanding, instruction architecture knowledge, code generation, debugging and cross-platform systems design.

Cross-platform Multiplayer Game Engine | C++, ECS, Lua, UDP, editor tooling

- Led a team of five developers building a reusable 2D game engine and a multiplayer R-Type-style game on top of it.
- Implemented a custom entity-component system, Lua scripting, editor tooling, packaging and a custom UDP networking layer designed for fast but reliable game sessions.
- Balanced engine architecture with the requirement to ship an actual playable networked game, keeping tooling and runtime behavior aligned with product needs.
- Owned technical direction and team coordination because of prior experience with the stack and project-management responsibilities.
- Received a perfect score and finished as the highest-graded team nationally for the project.

CUDA Distributed Raytracer | C++, CUDA, networking, performance optimization

- Implemented GPU raytracing with CUDA for a networked offline renderer, while also contributing to distributed job dispatch across worker machines.
- Profiled execution and optimized hot paths, with particular attention to memory layout and avoiding cache-unfriendly access patterns.
- Achieved the fastest result in the cohort: under 10 seconds for a complex 1080p scene with multiple shapes, materials and a high sample count.
- The project combined GPU programming, multithreading/distribution, numerical rendering work and low-level performance analysis.

ADDITIONAL ENGINEERING PROJECTS

Autonomous Vehicle | ROS2, OpenCV, LiDAR, SLAM, robotics

- Programmed a LIMO mobile robot during Korea University's Autonomous Vehicle Technology course to follow lanes, react to traffic lights and navigate from a starting point to a dynamically selected destination.
- Integrated camera and LiDAR perception with ROS2 nodes and SLAM-based localization/navigation in an environment designed to mimic changing urban or warehouse conditions.
- Implemented adaptable computer-vision logic that had to operate in real time alongside the rest of the ROS2 stack while supporting different driving priorities such as fast, secure and navigation modes.

ADDITIONAL ENGINEERING PROJECTS - CONTINUED

Large-scale Mobile Marketplace | Mobile product architecture, maps, performance

- Built an end-of-studies marketplace concept for iOS and Android connecting users with nearby local producers and a community-based delivery model.
- Designed the interaction model around three user roles - customer, co-rider and producer - while keeping the user experience consistent through a shared adaptive interaction layer.
- Focused on performance for map- and list-heavy mobile workflows, and on keeping UI architecture testable despite role-dependent actions and screens.

tcsh-compatible Shell Recreation | C, Unix process model, shell behavior

- Recreated the tcsh shell in C, implementing the core shell behaviors and command-processing features expected from a Unix command interpreter.
- Used the project to deepen understanding of processes, system interfaces, parsing and low-level Unix application behavior.

EDUCATION

Software Engineer | EPITECH - IONIS Group

Paris | Sep 2021 - 2026

- Software Engineer title. Ranked #1 nationally out of approximately 800 students during the first two years; GPA 3.30/4.00.

Exchange Student - Software Convergence | Korea University Sejong

South Korea | Sep 2024 - Jun 2025

- Coursework focused on robotics, AI and algorithms, including hands-on autonomous-vehicle development with ROS2, OpenCV, LiDAR and SLAM.

TECHNICAL SKILLS

Programming: Swift, C++, Rust, C, Go, Python, TypeScript

Apple / Graphics: Metal, Xcode, macOS/iOS development, CUDA, SDL-GPU

Systems / Hardware: Linux, Ubuntu, ARM64, MIPS, FPGA, Qualcomm NPU, Docker, Kubernetes, Git, Jenkins

AI / Backend: LLMs, MCP, GraphRAG, API integration, data pipelines, model training/deployment, n8n

LANGUAGES

- French - Native
- English - TOEIC-equivalent 925/990
- Japanese - JLPT N4-level knowledge (not officially certified)

INTERESTS

Photography and video, retro video games, nature, movies and TV series, science and technology.