

A software engineer based in Seoul, Republic of Korea, with diverse experience in systems programming, programming languages, game servers, ML accelerator software and, most recently, agentic coding. Values correctness and user experience above all, backed by a specification-driven, verification-heavy workflow.

Email: kang.seonghoon@mearie.org (preferred)

Further contact points available at the end of this résumé.

Work Experiences	FuriosaAI Inc. https://furiosa.ai/ , Senior SW Engineer	Dec 2022 — Jan 2026
	<ul style="list-style-type: none"> Design and development of “sprinter”, a tokio-based runtime executor for compiled ML workloads. <ul style="list-style-type: none"> Built it to faithfully execute compiler-scheduled graphs mixing NPU computation, data copies and Python pre/post-processing, realizing the full potential of the schedule through concurrent ops, pipelining and latency hiding. Implemented peer-to-peer pipelines spanning multiple NPUs, and a zero-copy host I/O path (huge pages and io_uring registered buffers), among others. Optimized it to the level required for running MLPerf benchmarks. Maintenance of various Python bindings and SDK packages, heavily utilizing PyO3. 	
	DevCAT Corporation https://devcat.com/ , Senior Game Programmer	Oct 2020 — Feb 2022
	Nexon Korea Corporation https://nexon.com/ , Senior Game Programmer	Aug 2014 — Sep 2020
	<ul style="list-style-type: none"> Maintenance of an internal communication tool, including the complete frontend rework. (2020–2022) Design and development of internal Silvervine Server Engine 2 (SSE2) server framework. (2018–2019) Development of an unreleased multiplayer online jigsaw game demo. (2018) Design and development of Kailua type checker and VS Code extension for Lua. (2015–2017) Server development and live operation for <i>Mabinogi Duel</i>. (2015) Client and server development of an unnamed internal prototype. (2014) 	
	iPlateia Inc. https://www.iplateia.com/ , Software Engineer	Feb 2013 — Jun 2014
	<ul style="list-style-type: none"> Architecture, development and optimization of a backend for a patented audio recognition solution. Development of an Android SDK for the aforementioned solution. 	
	Move Inc. https://move.is/ , Software Engineer	Feb 2012 — Dec 2012
	<ul style="list-style-type: none"> Development of a backend, API server and Android application for the photo sharing service Roomie. Development of a backend, web service and trending algorithm for GAE9 https://gae9.com/. 	
	NexG Co., Ltd. https://www.nexg.net/ , Software Engineer	Jan 2008 — Dec 2008
	<ul style="list-style-type: none"> Maintenance of a command line framework for routing daemons. Development of various user-land daemons mostly from scratch: SNMP v1/v2c/v3 client and server, TACACS+ server, SSH server and TLS server. 	
Education	KAIST, Master of Science	Mar 2010 — Feb 2012
	<ul style="list-style-type: none"> An alumnus of the Programming Language Research Group https://plrg.kaist.ac.kr/. 	
	KAIST, Bachelor of Science	Mar 2005 — Feb 2010
	Gyeonggi Science High School	Mar 2003 — Feb 2005

The full list is available at <https://github.com/lifthrasiir> and <https://mearie.org/projects/>.

Aqueous Archive Format <https://github.com/lifthrasiir/aqueous> **Rust (Agentic) | 2026**

- A next-generation, self-describing compressed archive format built around lossless recompression of common file formats; built on open standards such as **WebAssembly** and **CBOR**.
- Compresses Matt Mahoney's 10 GB benchmark to 2.58 GB, 38% smaller than `zstd --long` and 30% smaller than `zpaq`, decompressing at over 400 MB/s; every file is round-trip verified on compression.
- Each file is a stack-machine program over WebAssembly modules embedded in the archive, so a decoder extracts formats it has never seen; unknown modules run in an OS-level sandbox.
- ~140K lines of Rust in four weeks with Claude Code, kept correct by a ~5K-line **spec** and 700+ tests.

WAH <https://github.com/lifthrasiir/wah> **C99 (Agentic) | 2025 — 2026**

- A WebAssembly 3.0 interpreter (and later, JIT compiler) in a single C header file (~19K lines), implementing the full deterministic profile including GC, exceptions, relaxed SIMD and memory64.
- Passes the entire official spec test suite; found and reported an upstream test generation bug (**spec#2150**).
- Carefully designed to run untrusted modules with predictable resource bounds (fuel, deadline and memory limits); fuzzed with libFuzzer and AFL and checked with sanitizers on 8 CI configurations.
- Started with Gemini CLI and moved to Claude Code, shifting from frequent hands-on fixes to fully delegated implementation. Personally audited and documented the entire public API.

J40 <https://github.com/lifthrasiir/j40> **C99 | 2022**

- A self-contained decoder implementation of the ISO/IEC 18181 **JPEG XL** image file format.
- The first ever reimplementation besides **libjxl**; uncovered hundreds of specification bugs.

Roadroller <https://github.com/lifthrasiir/roadroller> **JavaScript | 2021**

- A size-optimizing heavyweight packer for large JavaScript demos (e.g. **js13kGames**).
- Based on rANS coding, logistic context mixing and a large number of sparse context models.
- Used by many js13kGames submissions in 2021, saving an additional 1–2 KB over then-best packers.

HTML5 Jigsaw Puzzle <https://s.mearie.org/jigsaw> **JavaScript, C# | 2017 — 2018**

- A no-frills jigsaw puzzle playable on desktop and mobile.
- Originally intended as a proof-of-concept and an SVG experiment at devCAT, later fully developed as a multiplayer online game demo for SSE2 (not public).

Kailua <https://github.com/devcat-studio/kailua> **Rust | 2015 — 2017**

- A type checker for Lua 5.1, implemented in Rust. Also included a VS Code IDE extension.
- The annotation language is unmodified Lua code with special comments for easy migration.
- The type system is based on Hindley–Milner and significantly extended with subtyping, gradual typing and row polymorphism, comparable to that of TypeScript 1.x.
- Used by several internal Lua projects; ultimately shelved because of the decision to switch to C#.

Rust programming language, Library Maintainer and Contributor **2012 — 2017**

- The architect and former maintainer of popular libraries including **Chrono**.
- Contributed several features to the standard library and tools, most significantly the correctly rounded float-to-decimal conversion using the then state-of-the-art algorithm (Grisu3).
- Recognized as a **Friend of the Tree** by the Rust Team.

Selected Presentations and Articles	Nexon Developer Conference (NDC) 2019: 실버바인 대기열 서버 설계 리뷰 <i>Design Review of Silvervine Waiting Line server</i>	
	NDC 2017: 내가 만든 언어의 개발환경을 Visual Studio Code로 빠르고 쉽게 구축하기 <i>Fast and easy development environment for my own languages in Visual Studio Code</i> (the second Kailua presentation)	
	NDC 2016: <마비노기 듀얼> 서버, 동적 타입 언어에서 반 정적 타입 언어로의 변신 <i>Mabinogi Duel server, the transformation from dynamic types to semi-static types</i> (the first Kailua presentation)	
	2016: <i>Why is a Rust executable large?</i>	
Selected Academic Publications	Seonghoon Kang and Sukyoung Ryu, Formal specification of a JavaScript module system , In <i>Proceedings of the ACM international conference on Object oriented programming systems languages and applications</i> (OOPSLA '12), October 2012.	
	Seonghoon Kang and Sukyoung Ryu, FortressCheck: Automatic Testing for Generic Properties , In <i>Proceedings of the 26th ACM Symposium on Applied Computing</i> (SAC 2011), March 2011.	
Selected Awards	Nexon Developer Conference, Best Presentation	Apr 2019
	International Obfuscated C Code Contest (IOCCC), Best short program	Sep 2012
	<pre>long long n,u,m,b;main(e,r)char **r;{f\ or(;n++ (e=getchar()) 32)>=0;b="ynwtsflrabg"[n%=11]-e?b:b*8+ n)for(r=b%64-25;e<47&&b;/=8)for(n=19;n[n["1+DIY/.K430x9\ G(kC["]-42&255~b (m+=n>15?n:n>9?m%u*~-u:~(int)r?n+ !(int)r*16:n*16,b=0))u=111<6177%n--*4;printf("%llx\n",m);}</pre>	
	KAIST Computer Science Department Best Master's Thesis	Feb 2012
	ACM ICPC — Seoul Regional, 8th place (shared) and NHN award	Nov 2007
Expertise	<ul style="list-style-type: none"> • Agentic software development: specification-driven design, tests and fuzzers as the source of truth, and auditing agent output (Claude Code, Gemini CLI). • Design and implementation of programming languages, type systems and interpreters. • Performance (architectural and/or low-level), memory and code size optimization. • Distributed and cryptographic protocol design, e.g. all protocols of SSE2 including their encryption layer. • Reverse engineering and static software analysis (x86, Android). • Design, implementation and specification of compression algorithms and archival file formats. <p>Primary languages: Rust, C, C++, Python, C#, JavaScript/TypeScript. Korean (native), English (spoken English slightly limited) and a bit of Japanese.</p>	
Other Activities	Probably the earliest discoverer of a large class of <code>preg_replace</code> vulnerabilities (CVE-2005-1820). Served as an administrator of Korean Wikipedia https://ko.wikipedia.org/ from 2006 to 2008.	
Contact	kang.seonghoon@mearie.org (preferred) • mearie.org • @senokay at X • @mearie.org at Bluesky • lifthrasiir at GitHub • linkedin.com/in/seonghoon-kang-169627229 • mobile available upon request (emergency use only) • not willing to relocate	