

HUU TAI LE

Corvallis, OR • taihl.work@gmail.com • huu67.com • github.com/SGDlab67

PROFESSIONAL SUMMARY

Software engineer focused on Rust and data infrastructure. Built and operated a continuous Rust pipeline streaming Solana mainnet lending activity into ClickHouse with freshness monitoring, gap detection, and incident write-ups, and ships Anchor and Token-2022 programs with their TypeScript frontends. Independently building a Rust instrument for Solana's validator gossip network with research direction from Dr. Zane Ma (Oregon State University).

EDUCATION

Oregon State University, M.S. in Computer Science | Expected Summer 2028

Relevant Coursework: Data Structures, Web Development

University of South Florida - B.S. in Hospitality Management, Minor in Finance | Aug 2022 – May 2026

TECHNICAL SKILLS

Languages: Rust, Python, TypeScript, JavaScript, Solidity, SQL, Bash

Testing and Quality: unit, integration, regression, fuzz, and invariant testing, boundary and negative-case design, data-quality audits, root-cause analysis, monitoring and alerting

Data and Infrastructure: ClickHouse, Apache Arrow, Parquet, Apache DataFusion, Yellowstone gRPC, Docker, systemd timers, Google Cloud, AWS, Git/GitHub

Blockchain / Web3: Solana (Anchor, Web3.js, SPL, PDAs, CPI), Solidity (Foundry, Ethers.js), Codama IDL Generation, ZK Private Transfer Systems, x402 Payment Protocol

Frameworks & Libraries: Anchor, Next.js 16, React 19, FastAPI, Tailwind CSS, @solana/react-hooks, Hugging Face

Tools & Platforms: Git/GitHub, Docker, AWS Lightsail, Claude Code, Hermes Agent, Linux (Ubuntu)

Certifications: Solidity Smart Contract Development, Chainlink Fundamentals, Foundry Fundamentals, Advanced Foundry, Blockchain Basics

PROJECTS

karat67: Integrity Checks for Solana Indexers | Sep 2026 – Present

Rust • CLI crate • Solana RPC • Carbon • github.com/SGDlab67/karat67 • [Carbon proposal #591](#)

- Shipped the first check in an open-source Rust crate and CLI (karat): an IDL-informed shape check that validates the discriminator and exact account size for Kamino Lend accounts, the failure class that once emptied this pipeline's Obligation rows while every liveness signal read green.
- Designing the next two checks: reconciliation that samples indexed accounts through getMultipleAccounts and diffs them against indexed values, and slot coverage compared against getBlocks rather than a contiguous range.
- Integrating with Carbon, an open-source Solana indexing framework, and opened an upstream proposal to count decode failures separately from foreign accounts. Entry for Colosseum's Crypto World's Fair 2026.

klend-indexer: DeFi Lending Data Pipeline | Jul 2026 – Aug 2026

Rust • Yellowstone gRPC • ClickHouse • Parquet • Google Cloud • github.com/SGDlab67/klend-indexer • [Failure modes](#) • [Frozen dashboard](#)

- Compared three Solana data providers (Alchemy, Helius, Triton) on monthly cost and fit, and chose Alchemy pay-as-you-go at \$1–\$2/month over \$499–\$2,900/month plans.
- Built and operated a continuous Rust streaming indexer ingesting Kamino Lend account state from Solana mainnet over Yellowstone gRPC into ClickHouse: 1.04M account updates and 140K decoded lending positions across 11 days unattended on GCP, at a 4-second median ingest lag on roughly 7.45 KB/s of bandwidth.
- Designed a data-integrity layer beyond liveness monitoring: Anchor discriminators derived at runtime rather than transcribed, and slot-gap detection keyed to a provider-independent clock.
- Reduced analytical storage cost from \$548/month to under \$1/month by exporting 4.25 GB of warehouse history into 91 MB of Hive-partitioned Parquet on Google Cloud Storage in 65 seconds, served through a DataFusion query path.
- Traced an 8.4-hour silent stall to a database write with no timeout on a half-open connection; shipped an insert timeout and an external freshness watchdog; wrote 50 unit tests including regression tests that replay both production incidents and boundary cases such as off-by-one gap tolerance, 128-bit overflow, and malformed account data.
- Caught silent data corruption that every liveness check missed: an accounts_data_slice request was returning empty payloads, and a distribution audit traced 64,650 lost rows over 158.9 hours to the stream filter misconfiguration behind it.

Solana Network Measurement (Research) | Aug 2026 – Present

Rust • Solana gossip protocol • scheduled data collection

- Independently building an observatory that monitors Solana's validator gossip network on a schedule, with research direction from Dr. [Zane Ma](#) (Oregon State University).
- Checking Anza's protocol documentation against Agave validator source code and recording where the docs have drifted from current behavior.

Decentralized Stablecoin: Advanced Foundry Course Project | Feb 2026

Solidity • Foundry • Chainlink • github.com/SGDlab67/foundry-defi-stablecoin-sgd • [Etherscan: DSCEngine](#) • [Etherscan: token](#)

- Implemented an overcollateralized stablecoin during Cyfrin Updraft's Advanced Foundry course: an ERC-20 whose only owner is the engine, so no wallet controls the supply, plus a DSCEngine holding wETH and wBTC collateral that enforces 200% overcollateralization and reads Chainlink price feeds through an OracleLib that reverts on stale data.
- Tested with unit, fuzz, and invariant tests in Foundry using mocks for failed transfers and stale price feeds, then deployed and source-verified both contracts on Sepolia.

Mentik World - Solana Summer School | June 2026 – August 2026

Solana Devnet • Rust • Anchor • Token-2022 • Next.js • React 19 • Zustand • [Live Demo](#)

- Designed and deployed two Anchor programs on Solana devnet: a non-custodial SOL staking pool with a u128 fixed-point reward index, Flexible / 7-day / 30-day lock tiers, and an in-place account-layout migration for legacy stake accounts.
- Launched the MENTIK token on Token-2022 with a Metadata Pointer, a Permanent Delegate, a PDA mint authority, and an admin-set SOL rate stored on-chain, with buy, force-transfer, and force-burn instructions.
- Built the browser roguelite in React 19 / Next.js / Zustand: state-machine exploration, turn-based combat, itemization, NPC shops, level-gated zones, and Phantom and Solflare wallet integration, keeping a strict client/server boundary so client state can never mint value on-chain.

Hermes Agent: Self-Hosted AI Operations | May 2026 – Present

Python • Hermes Agent (Nous Research) • open-weight models (Hugging Face) • OKX Agentic Wallet SDK • AWS Lightsail

- Run a self-hosted AI agent on scheduled jobs for repository maintenance, system updates, and this research project's data collection.
- Monitor agent runs, API spend, and runtime stability through network timeouts and volatile conditions.

EXPERIENCE

Tan Le Quang Jewelry Store | Sales and Digital Marketing Apprentice | May 2024 – Aug 2024

Vietnam • On-site • Web Development • Social Media Marketing • tanlequangjewelry.com

- Designed, built, and deployed **tanlequangjewelry.com** as a discovery-to-visit funnel, pairing a curated collection showcase with clear location, hours, and directions to convert online interest into in-store visits.
- Grew **Facebook and TikTok traffic by 30%** through localized, seasonal content engineered to bring customers into the physical store rather than sell online.

TRAINING & CONTINUING EDUCATION

Solana Developer Bootcamp 2026 - Solana Foundation | Spring 2026

- Completed the official 2026 cohort of an intensive engineering program focused on **low-level Solana architecture, Rust compute-unit optimization, and advanced state management** via the Anchor framework. Built an on-chain real-world asset raffle as the cohort project: a PDA-isolated vault program in Rust and Anchor with a Codama-generated TypeScript client.

Solana Summer School - led by [Andres Correia](#) | June 2026 – August 2026

- Studied **Token-2022 extensions** such as the metadata pointer, permanent delegate, and transfer hooks, and their role in designing safe, programmable on-chain token economies.
- Explored the internal **structure of escrow and vault programs** across both the **Anchor and Pinocchio** frameworks, comparing high-level account validation against low-level, compute-unit-optimized state handling.