Daniel Van Der Maden

✉ contact@danielvandermaden.com ☎ (714) 251-4974
danielvandermaden.com github.com/Daniel-VDM in linkedin.com/in/Daniel-VDM

EXPERIENCE

Harmony Blockchain Protocol (Startup) **Mountain View, CA**
Software Engineer – Core Protocol Development *September 2019 – Present*

- Optimized transaction processing & spam protection for the node transaction pool to handle 100k+ concurrent transactions per sec.
- Integrated new staking transactions with the node transaction pool & improved debugging with new error handling/reporting.
- Overhauled transaction processing functions for block consensus resulting in better test coverage & memory usage.

Software Engineer – Node API Lead

- Implemented Coinbase's Rosetta Node API specification as lead developer. Propagated fixes upstream to the Rosetta SDK.
- Overhauled the node RPC layer making it externally consumable, robust, and efficient with use of single-flight requests & caching.

Software Engineer – Developer Tooling & Testing

- Lead & maintained the development of the CLI – providing an easy way to interact with any Harmony chain over the command line.
- Created an OS agnostic (via Docker) integration test framework to easily & reliably test complex scenarios against any core changes.
- Developed core logic (transaction construction, key management, and staking utilities) for the Harmony Python & Golang SDK.
- Helped develop smart contract execution tracing to reveal how a transaction's code was executed on the Harmony chain.

Software Engineer – Developer Operations

- Managed test network operations. Deployed & maintained test networks with 500+ machines with a 95%+ up-time.
- Developed advanced tools to automatically recover & report consensus halts/bugs on test networks via a Jenkins job.
- Primary on-call backup for main network operations. Helped develop operational processes resulting in 99.9% main-net uptime.
- Designed & implemented a tool (AutoNode) to run and manage a node/validator on the Harmony network with optimal rewards.

Microsemi/Microchip Corporation **Aliso Viejo, CA**
Software Engineering Intern – Frequency and Timing Division *May 2018 – August 2018*

- Develop embedded programs in C for a modern radio navigation and data system as part of a DARPA project.
- Worked closely with other software engineers to rapidly implement multiple signal schemes and meet fixed live test deadlines.

EECS Department, UC Berkeley **Berkeley, CA**
Tutor – Self-Paced Center *January 2019 – May 2019*

- Improved student understanding by holding office hours for students learning C, C++, Java, and Python.

Academic Intern – Structure & Interpretation of Programs Course (Python) *August 2018 – December 2018*

- Fostered interest in Computer Science fundamentals by providing guidance on homework and projects during office hours.

Computer Science Mentors, UC Berkeley **Berkeley, CA**
Peer Mentor – Designing Information Devices & Systems Course (Circuits / Linear Alg) *January 2019 – May 2019*

- Helped students solidify core EECS concepts by designing & leading a supplemental 2 hour discussion section each week.

EDUCATION

University of California, Berkeley **Berkeley, CA**
Bachelor's degree in Computer Science – GPA: 3.73/4.00 *2019*

NOTABLE PROJECTS

More projects (with repo links) can be found on my personal website

Rosetta Data & Construction API – Harmony Blockchain Protocol https://tinyurl.com/y4n5zoug

- Fully implemented Coinbase's Rosetta API specification for Harmony's core. Technical medium article (linked) endorsed by Coinbase.
- Implemented fixes to both the Rosetta SDK & Harmony core during implementation ensuring absolute accuracy of all data.

Port Curve.fi from Ethereum on to Harmony https://tinyurl.com/y6luyb6d

- Curve.fi is a distributed application that facilitates swaps between "stable coins". This was done as a 1 week hackathon project.

Approximate Solver for a NP-Hard Problem – Efficient Algorithms Course Project https://tinyurl.com/y2dttmrv

- A polynomial-time approximate solver (implemented in Python) that uses a greedy algorithm with various heuristics.
- Yielded solutions that were in the top 10% of all approximations in the course (which had ≈ 700 students).

SKILLS

Programming Languages: Golang, Python, JavaScript, Solidity, Vyper, Java, C++, C, Scheme, RISC-V/x86, SQL, HTML, CSS
Technologies: Blockchain, EVM, AWS, Hadoop, OpenMP, Intel AVX, Scipy/Numpy/Pandas, Scikit-learn, Bootstrap, MySQL, Firebase