NATHANAEL J. BRACY
nate@bracy.dev — (413) 801-7771 — https://bracy.dev

EXPERIENCE

Innovation Engineer
APM Help (https://apmhelp.com/) *Feb 2025 - Present*

- Rapidly prototyped full-stack services to support new product initiatives, utilizing **Anthropic** and **OpenAI** models to drive AI-enabled business logic, while authoring **technical documentation** to facilitate knowledge transfer to the wider engineering team.
- Architected autonomous **computer-use browser agents** to navigate complex UIs, automating repetitive high-volume tasks across property management ecosystems including **Appfolio**, **Buildium**, and **RentVine**.
- Designed **secure authentication** mechanisms for agent deployments, aligning with **ISO 27001** and NIST standards by implementing **encrypted credential storage** and **least-privilege access controls** to protect sensitive property management data.
- Implemented distributed tracing via **Grafana** to debug failures in **agentic workflows** and deploy **self-healing fixes**.
- Led the **full project lifecycle** for an AI-powered invoice pipeline, identifying integration risks and coordinating with product teams to deliver a solution handling **30k+ invoices**.
- Engineered a high-throughput **1099 reconciliation pipeline**, cross-referencing 1099-MISC/NEC totals against raw cashflow ledgers, identifying variance in over **10k+ records** audited.

Student Ambassador
Intel Corporation *Feb 2025 - Aug 2025*

- Promoted awareness and adoption of Intel's **oneAPI** and associated toolkits among students and faculty.
- Organized and led workshops on high-performance computing (HPC), **parallel programming**, and heterogeneous computing.
- Provided **technical guidance and mentorship** to students exploring AI, machine learning, and accelerated computing using Intel platforms.

Software Engineering Intern
Rappo (https://buildrappo.com/) *May 2024 - Feb 2025*

- Developed a responsive landing page and blog using **Astro.js** and **Tailwind CSS** ensuring a modern and user-friendly frontend.
- Built reusable web components with **TypeScript** enhancing code maintainability and portability.
- Configured and optimized backend infrastructure on **Google Cloud Platform (GCP)**, to improve server performance and deployment efficiency.
- Coordinated development workflows using **Git** and **Docker**, performing **code reviews**, resolving merge conflicts, and enforcing branching strategies to maintain repository integrity.
- Ensured code quality and reliability through comprehensive unit tests and end-to-end testing with **Jest** and **Cypress**.

EDUCATION

Clarkson University, Potsdam, New York — Bachelor of Science
Computer Engineering, Minor in Mathematics

MEMBERSHIPS

- Institute of Electrical and Electronics Engineers
- Clarkson Open-Source Institute

TECHNICAL SKILLS

- Programming Languages: C++, Go, Python, Rust, Java, TypeScript, VHDL
- Frameworks: Astro, Cypress, Express, Gin, Jest, Puppeteer, Vue
- DevOps: CI/CD, Docker, Git, Kubernetes

CERTIFICATIONS

- AWS Certified Cloud Practitioner

OPEN-SOURCE

shards (https://github.com/servusdei2018/shards) — *Jan 2021 - Present*

- Developed a scalable sharding library in **Go** for the **discordgo** framework, featuring automatic scaling and zero-downtime restarts.
- Built a robust **Manager** API to streamline session routing, state management, and bulk Discord slash command deployment.
- Optimized performance and thread safety using **sync.RWMutex** to manage concurrent shard operations across large-scale bot deployments.

undo (https://github.com/servusdei2018/undo) — *Nov 2024 - Present*

- Developed a CLI tool in **Rust** to track and roll back file modifications made by external processes in real-time.
- Leveraged low-level **ptrace** syscalls to intercept process activity and monitor file writes at the system-level.

syllogix (https://github.com/servusdei2018/syllogix) — *Nov 2025 - Present*

- Architected an agentic reasoning framework in **Python** that enables LLMs to solve complex problems through verifiable, step-by-step syllogistic logic.
- Developed an auditable "train of thought" engine that validates model conclusions against formal deductive and inductive logical structures.
- Built a modular system using **uv** for dependency management and **pytest** for verification of logical proof generation.