Owen Hope

609-618-7065 • owen.hope96@gmail.com • Belmar, NJ • GitHub: owen-hope11

Senior Software Engineer specializing in distributed systems architecture and high-performance infrastructure. Expert in event-driven design, real-time processing, and system optimization, with experience ranging from government network research to healthcare technology at scale.

Education

New Jersey Institute of Technology

Masters of Science: *Computer Science* **Specialization:** *Networking & Cyber Security* **May 2021**

Bachelor of Science: *Computer Science* **Minor:** *Applied Mathematics* **May 2019**

Skills Proficient with Kotlin, Rust, C/C++, Python, Temporal, Postgres, Event Driven Architecture, AWS Bedrock, Kafka, JOOQ, GraphQL, Micronaut, Snowflake, DataDog, AuthZed, Git, AWS, Docker, Bash, Low Level Networking, TCP/IP

Working knowledge in Terraform, RabbitMQ, Kubernetes, Java, TypeScript, PHP, MySQL, Azure, Wireshark

Work Experience

Curative Health Inc.

Senior Software Engineer Feb 2023 - Present

Software Engineer Oct 2021 - Feb 2023

- Reduced API latency by 20% by migrating synchronous appointment scheduling system to distributed asynchronous architecture processing 10K+ events/day, implementing event-driven flows for booking, cancellation, and rescheduling workflows with Lambda-based processors for third-party API integration.
- Designed and implemented organization-wide Event-Driven Architecture using AWS EventBridge, SQS, and Lambda, establishing cross-team standards (CloudEvents, AsyncAPI) and developing reusable Terraform modules for consistent event-driven patterns across engineering teams.
- As Tech Lead, led migration to Service-based architecture by extracting a cross-team domain into a standalone microservice with internal RPC APIs handling 50K+ requests/day across claim processing and member access workflows. Serves 3 engineering teams with dedicated matching logic and data access patterns, reducing monolith database overhead by 35% and enabling independent deployment cycles.
- Mentor 2 engineers through weekly 1-on-1s and pair programming sessions, providing guidance on technical challenges, code review best practices, and system design patterns.
- Lead the design of a tool to be used by the Financial team to track the movement of millions of dollars across multiple accounts. Implement a high level of reliability, transparency, and redundancy with Temporal by tracking each transaction and action performed on the data as a dedicated activity.
- Implemented a Retrieval-Augmented Generation (RAG) system powered by AWS Bedrock LLMs and a vector database to optimize knowledge retrieval for Member Services, reducing average benefit-related query resolution time by 5 minutes per appointment
- Lead the design and development of a highly visible scheduling system that every user must go through with zero downtime. Build a scheduling algorithm that takes Provider availability into account and builds out all available times. This led to a reduction of the Round Trip Time from p99 of 25+ seconds to < 1s
- Architected and implemented a comprehensive database audit logging system with user attribution for Kotlin GraphQL API. Provisioned AWS MSK infrastructure via Terraform with Debezium CDC connectors, engineered custom Kotlin annotations for user context propagation to Redis cache, and developed a Kafka Streams microservice to correlate database change events with authenticated user sessions.
- Lead the migration and architecture of a core billing system running in jupyter notebooks to a Tonic Rust based microservice with Foreign Function Interface (FFI) into the legacy Python application. Architect a clear logging pattern and database storage to allow easy data engineering to be complete. Decreased the runtime of billing service by 3x and simplifying the nested logic

Raytheon BBN Technologies

Software Developer/Staff Scientist 1 May 2021 - Oct 2021
Associate Network Scientist July 2019 - May 2021

- Key member of the team, leading development of government sponsored research through advanced knowledge in Rust, C and network protocols.
- Architect, design, and implement shared memory Inter-Process Communication (IPC) channel, C foreign function interfaces, and in line packet digestion and analysis on a multi-threaded Rust system.
- Identification of Encrypted flows over a network that can be correlated to Applications and Entities using clustering based algorithms.
- Lead API architect in implementing gRPC API, message caching, and orchestrate the data pipeline system in Rust, following Test Driven Development.
- Working with key partners to integrate RabbitMQ wrapper libraries into an overlay routing system to deliver reliable updates and forwarding tables.
- Develop Multiplexing and Load Balancing functionality on Heterogeneous Overlay Mesh Routing System in C/C++ across multiple environmental domains (sea, air, land, and space).
- Programmed a Python API used to orchestrate docker containerization and networking as well as networking simulators such as NS3 for easy to create scenarios.

Projects:

Branchspawn:

- Implement a Rust base CLI tool using the Clap library for creating dedicated database containers for your git branches. To be used for workflows where each feature branch needs its own isolated database environment.