Cameron Duley

cameron.duley99@gmail.com | github | linkedin | 301-526-8577

Thank you for considering my application! Writing software's more than just my career, it's my life's passion. I enjoy working on software from frontend to backend, design tokens to databases, and client consultation to developer operations! My favorite languages are Elixir and Go for their concurrent runtimes that enable unparalleled problem solving.

Skills

Elixir	1 year. Contributor to the language itself and open source libraries.
Go	4 years. Log processing, backend services, and web3 integrations.
SQL	4 years. Well versed in schema design w/ Postgres and MariaDB.
JS/TS/Node	6 years. Experience with React, Vue, Svelte, jQuery, and vanilla JS.
PHP	4 years. Maintaining legacy enterprise apps and content-driven sites.
Python	4 years. Ecommerce backends and 3D model pipelines.
macOS/Linux	5 years. Preferred development and hosting respectively.
Windows	6 years. Comfortable with using and supporting.
AWS	4 years. Used for hosting a majority of clients served.
GCP	1 year. Favored for personal projects.

Work Experience

Senior Software Developer

Oct 2019 - Present

Coretechs Consulting — Remote, Kensington MD

- Developed a Kevo client library in Elixir, for use in a broader Phoenix app that required smart lock integration. Client library features websocket support and leverages HTTP/2 multiplexing for optimal throughput. Implementation is open sourced at github.com/Moosieus/kevo_ex.

- Assumed leadership on a legislative tracking system which had been stuck in development for years. Accelerated development by aggressively addressing technical debt in the existing PHP codebase and overhauling the MySQL database schema. Implemented a new frontend using jQuery and DataTables, backed by a JSON/REST API. Leveraged Apache Solr search engine to provide fast and accurate search results. Notable users include Astellas, ZoomInfo, and TikTok.

- Designed and implemented a blockchain transaction handler to process financialized assets using Go, go-ethereum, and Postgres. Leveraged Go's concurrency model to multi-thread processing and minimize blocking for scalable performance. System architected to accommodate changing business requirements, and process events idempotently so corrections could be issued w/o rollbacks.

- Developed a facilities management system for universities utilizing Go for the backend REST API and Svelte for the frontend. Designed stack to integrate with Central Authentication Service SSO

and run on limited colocated hardware. Emphasis placed on ease of use for students and administrators to minimize misuse of facilities.

- Resolved performance problems for NGINX proxies operating behind the Great Firewall. Found solutions through gathering metrics and performing data analysis on HTTPS traffic. Client implemented the advised solution which allowed business operations to resume.

- Led development on an Electronic Health Record (EHR) system written in Laravel. Translated HIPAA requirements into technical improvements that improved productivity by ensuring secure and compliant defaults. Simplified deployments into a single command using deployer.org.

Freelance Web Developer

Feb 2019 – Oct 2019

Self Employed

Rockville MD

- Designed websites and applications for individuals with small businesses and personal ventures. Services were budget-friendly with a strong emphasis on user experience and presentation.

- Used Vue with Nuxt and Vuetify for frontend and design. Websites were primarily content-driven and took full advantage of static site generation for low cost edge hosting. Applications were created using Electron and provided operator-friendly front ends for CLI-native apps and web APIs.

3D Printing Workflow Automation Engineer

Jan 2016 – Oct 2019

BigSpool

D.C. Metro Area

- Hired as an engineering intern to oversee operations of a 3D printer lab, manufacturing products for e-commerce. Problems were present in production due to suboptimal model quality and poor G-code generation. These issues affected over 17,000 SKUs all which required manual slicing.

- Used Process Hacker on Windows to identify an undocumented command line API in the proprietary slicer software our 3D printers required, implementing an automated frontend in Python. This reduced our processing time from ~2 weeks of manual labor to an automated process that ran overnight.

- Revamped the design-to-model pipeline with the gained iteration speed and time, moving from Node/Three.js to Python and OpenSCAD. New pipeline produced models to precise tolerances, eliminated water-tightness issues, allowed for additional product sizing options, and rapid integration of products derived from our sales analysis and web scraping efforts.

Open Source Contributions

kevo_ex - A client library for Kwikset's Kevo smart locks

Kevo smart locks don't have an official API but can be interfaced with by reverse engineering their web client at mykevo.com. kevo_ex leverages HTTP/2 multiplexing to improve throughput, and Erlang's gen_statem to provide strong fault tolerance.

Elixir (language) - Implemented Unicode U+FFFD Substitution

Coming in 1.16, `String.replace_invalid/2` allows UTF-8 encoded network data (such as JSON or XML) with encoding errors present to be transformed into a next-best usable form. The implemented solution follows the Unicode Standard's recommendations, so systems running different languages all interpret malformed data the same way. The implementation is fast, concise, well tested, and only allocates 128 bytes for any size input.

elixir-a2s - Implementation of Valve's A2S Protocol for Elixir.

I chose to implement A2S as Elixir had no existing implementation, and being able to interface with game servers provides novice developers a great platform to learn.

A2S is particularly challenging to implement as it transmits values as they are in-memory; strings are null terminated, numbers are little endian, and significant bits are used as flags. The protocol uses UDP instead of TCP which requires packet management at the application level.

elixir-a2s leverages BEAM's Actor-Model runtime to enable concurrent requests where implementations in other languages sequentially block. Elixir's binary pattern matching is used to fluently parse the packets, whereas most other languages depend on several procedural steps. Erlang's :gen_statem behavior is used to handle packet management in lieu of TCP.