# Gregory Shuflin

✉ *greg.shuflin@protonmail.com*  
🏠 greg.everydayimshuflin.com  
git gitea.everydayimshuflin.com  (personal)

📍 Oakland, California  
📞 510-332-6344  
git github.com/gshuflin  (professional)

## Qualifications and interests

Motivated, professional computer scientist with broad expertise across several disciplines of industry and open-source software development:

- Programming language theory, type theory, strongly-typed functional programming languages. Haskell, Elm, Scala. Started writing Rust before 1.0 release in 2015.

- Network programming and troubleshooting, hardware and software packet processing, embedded Linux hardware bringup, writing clear and maintainable C and C++. Can solder if necessary.

- Full-stack web development in multiple web frameworks in multiple languages. Modern frontend web development (Javascript/Typescript/React).

- Linux system administration in production cloud environments, Docker containerization, AWS/Terraform. I have a running 4U VM server on a shelf in my apartment.

## Professional Experience

**Toolchain Labs**                                                          San Francisco, CA  
*Software Engineer*                                                                *2019 - 2021*

- Primarily responsible for contributing business-critical features, bugfixes, and documentation to the Pants open-source build system.

- Updated old Python codebase to modern typed Python + Rust execution core for improved performance and maintainability. Frequent Rust/Python FFI and Python metaprogramming.

- Routinely collaborated with external contributors in a spirit of good open-source citizenship.

- Contributed to internal React/Typescript frontend work, Django/AWS backend work at early-stage startup (<10 employees) when all hands needed on deck.

**3D Robotics**                                                                     Berkeley, CA  
*Senior Software Engineer*                                                          *2017 - 2019*

- One of two members of the backend/devops team. Primarily responsible for feature development for API microservices in Scala (Akka/Couchbase NoSQL), and AWS provision of the same (Docker, Terraform).

- Built systems for processing GDAL geospatial data, efficiently running TensorFlow/OpenCV ML workflows on quadrotor drone photos.

- One time I grabbed a Windows laptop and set up a Visual C++ environment so I could apply a 10-line bugfix patch to some Autodesk software we depended on. Only time so far I've needed to do that.

**Cisco Meraki**                                                            San Francisco, CA  
*Software Engineer*                                                                *2013 - 2017*

- Primarily responsible for feature development and support of several different models of cloud-managed Ethernet switches and WiFi access points.

- Healthy mix of embedded Linux firmware development on networking hardware (C++) and full stack web development (Ruby on Rails/PostgreSQL + a little Scala).

- Designed and built WiFi statistics monitoring widget with d3 and React, looks cool in demos and saved at least one deal.

- Regular troubleshooting and debugging of hardware and software in production network environments. The 802.1Q protocol identifier value 0x8100 is burned into my memory from seeing it in packet logs so often.

## Education

**University of California, Berkeley**                               *graduated Winter 2012*
*Bachelor of Arts, Computer Science, Linguistics, Japanese Language*
Relevant Coursework: Artificial Intelligence, Compilers, Operating Systems, Algorithms, Data Structures, Computer Graphics

## Personal Open-source Projects

**Untrusted (`github.com/AlexNisnevich/untrusted`)** An open-source browser-based roguelike game. Every level is unbeatable and requires the player to modify the Javascript code defining the level in order to progress.

This started out as a college hackathon project by me and a friend, that we decided to keep working on and release publicly. A surprisingly large number of people around the world have contacted us telling us they've enjoyed playing it, or found it to be a fun and useful pedagogical tool for teaching programming.

**Hilite (Rust Port)** Rust port (with a few new features) of Hilite, a simple command-line utility to highlight stderr output, useful for build systems.