

Chaim Halbert

ד"ר

Phoenix, AZ 85016

github.com/chaimleib — linkedin.com/in/chaim-leib-halbert-78686089 — chaim.leib.halbert@gmail.com

GOALS

- Backend, full-stack, or cloud engineering
- Local in Phoenix area or remote
- Protect privacy, empower the customer

SKILLS

Web technologies: Full stack web development using Google Cloud Platform and AWS; React, Angular, jQuery, Webpack, Docker.

Coding: >50 programming languages, constantly learning more and inventing new ones (ask me about both!). Most recent work experience in Go, C#, JS/Node.js, MySQL and bash shell; less recent work was in Typescript, Java, and PHP. Personal project experience in C, Python, and Ruby. Can write parallel and concurrent code.

Development process engineering: Designs customized systems and operating procedures for the problem and the team, some of which have been battle-tested for six years and are still going strong.

Security and privacy: Catches potential attack vectors at code review. Protects coworkers' and customers' privacy as a matter of deep personal conviction to "do no harm," while allowing the power and flexibility to get the job done.

Other: Has studied and implemented algorithms for music synthesis and analysis, robot control, machine learning, OCR, pathfinding, 3D graphics and raytracing, parsers, compilers, interpreters, and operating systems. Uses Arch Linux as daily driver. This resume was written in vim using Markdown and rendered with pango.

WORK EXPERIENCE

Cloud Software Engineer, Evernote (June 2016 - February 2023; Redwood City, CA until October 2020, then remote in Phoenix, AZ)

Re-architected and migrated evernote.com from PHP with AWS to Go with Google App Engine My system design has been in production for six years so far.

- Recursive HTML/JSON template system, and compatibility checkers to ensure that all JSON data remains valid even as templates change.
- CMS integration allows non-programmers to assemble template components into pages via Crownpeak (C#, command-line SSH/SFTP, Google Cloud Compute+Storage).
- New tooling for deploy, preview, unit test, and code quality (Go, yarn, webpack, nvm/fnm, bash/zsh, rsync, SASS, prettier, eslint, goimports, golint-ci, shellcheck).
- Created search tools to answer frequent questions about the project, like, "What URLs do we serve?" (Go, MySQL, bash, awk, grep/ripgrep, sed, jq/yq, xargs, etc.)

Content Delivery Network (CDN): Google Cloud Storage. For images in Iterable emails.

Browser-based note editor and checkout flow: Typescript, React, Java.

Jenkins CI testing (BitBucket-Jenkins integration): Test-gated changes to git repo.

Google Cloud Platform: scaling, hosting, Pub/Sub, BigQuery, storage, secrets management.

Communication: Slack (+ API), Atlassian JIRA and Confluence, BitBucket, DataDog, PagerDuty.

Software Engineer, Coupa Software (May 2015 – January 2016; San Mateo, CA)

Ruby on Rails, MySQL, Bootstrap, Turnip unit tests, Jenkins, and bash. Optimized RSpec test harness to work 30% faster; developed in-browser integration tests and unit tests; cooperated via AGILE with hundreds of developers using GitHub and Atlassian JIRA.

Software Engineer Intern, Rustici Software (June 2014 - August 2014; Nashville, TN)

Created Python client library for TinCan API (xAPI 2.0) and created a JSON object verifier and recognizer using JSON Schema and Node.js/JavaScript. Worked on five-person team using GitHub.

Industrial Controls Engineer, Bridgestone (May 2012 - December 2012; Nashville, TN)

Designed safety systems and machine control systems with AutoCAD Electrical, CoDeSys, and RSLogix. Coded PLCs in structured text (IEC 61131-3) and ladder logic, and created AutoCAD plugins with C#, Visual Basic, and LISP.

Computer Programmer Intern, Oak Ridge National Laboratory Physics Division (June 2010 - August 2010; Oak Ridge, TN)

Modernized a browser-based experiment log system. Used PHP, JavaScript, MySQL, TinyMCE, ssh, vim, Red Hat Linux.

NOTABLE PERSONAL AND SCHOOL PROJECTS

github.com/chaimleib/intervaltree, 2014 onward. Modify and query intervals in logarithmic time in pure Python 2/3. This is currently the industry reference implementation of interval trees in Python, cited in numerous academic research papers for gene research and computer vision applications, etc.

github.com/chaimleib/dotfiles, 2014 onward. Bash and ZSH compatible environment setup, automatically install desired tools on apt/apk/pacman/brew, lssys detects OS/CPU/RAM, vim and neovim configuration for Language Server Protocol, etc.

github.com/chaimleib/rpn, 2023 onward. C project using flex and bison to implement a calculator. CI testing with CircleCI and SonarCloud.

Pyenv v2.3.14, github.com/pyenv/pyenv February 2023. Added patches in C for Python's source code, versions 3.6 and 3.5, fixing for the world compilation for modern OSes on x86_64 and Apple M1 processors.

Completed Linux From Scratch in 2006 and 2021 (<https://www.linuxfromscratch.org/lfs/>). Compiled and configured a complete Linux system from C/C++ source code and the command line.

chaimleib.com personal website and jQuery audio player (zichronos), 2014 onward. TLS certificate/HTTPS. Static pages and media generated using Ruby, Jekyll and GitHub Pages, Bash and SCSS, audio player using jQuery.

USB Nintendo 64 Cartridge reader, 2014. Allows backups of N64 games to the computer. Written in C for Atmega32u4 microcontroller (Teensy 1.0).

2-axis gimballed spotlight with IR camera target tracker, UC Berkeley EECS 149 Intro to Embedded Systems, 2014. Python, finite state machine program architecture, Raspberry Pi, Bluetooth, Nintendo Wii controller, RS485 serial servos.

Handwritten digit recognizer, Coursera.com Machine Learning, August 2012 with Prof. Andrew Ng. Written in Octave, an open-source clone of Matlab.

Lua-like interpreter, parser generator and bytecode compiler, UC Berkeley CS 164 Compilers, 2010. Written in Python 2. Also created a Vim syntax highlighting file for the new language.

EDUCATION

University of California, Berkeley – B.S. program in Electrical Engineering and Computer Science (August 2008 - March 2014)

Projects were completed in Python, C, C++, Java, JavaScript, Matlab, Scheme, Verilog, and Latex. Courses covered recursive data structures and algorithms, AI, computer graphics, compilers, embedded systems, DSP, FPGAs, analog and digital circuits. Nearly all technical courses required ssh to bash shell on Solaris or CentOS, GNU Makefile, and some sort of VCS, like SVN or mercurial, but usually git.