

Justin McGirr

📞 (226) 476 4479
✉ hire@jmcgirr.com
🌐 jmcgirr.com

Education **Bachelor of Computer Science, Digital Hardware Option,**
University of Waterloo, Waterloo, 2011 - 2016

5 - 10 years **Linux, Python, C, C++, JavaScript, HTML & CSS**

1 - 5 years Go, SQL, L^AT_EX, PHP, (TI-)BASIC, mongodb, C#, VHDL, Verilog

Employment

October 2020 **Software Engineer, Google, Waterloo and Seattle**

- March 2023 Made GPUs more accessible, reliable, and secure. Wore many hats in a fast-growing team with the ultimate responsibility of accelerating AI and ML workloads for Google Cloud customers. Worked in Python, C++, SQL, Bash, Markdown, and Google-specific languages.

Created the first monitoring and alerting for A100 GPUs. This revealed bugs in production, and enabled us to launch A100s to all cloud customers - allowing those customers to run much bigger, multi-gpu AI models.

Built several dashboards, Jupyter notebooks, and Python tools to monitor the GPU fleet. This allowed us to understand & fix machines that were stuck, unable to receive updates or repairs.

Delivered fixes for a security vulnerability that had the potential for persistent root-in-prod. My quick delivery of the fixes allowed us to avoid drastic action - leadership was considering blocking all new customers, which would have cost millions in revenue. Coordinated across teams in the middle of a production freeze to deliver quickly, earning personal recognition (and a bonus!) from our director.

Designed, developed, and delivered a complete solution for the same vulnerability. This required significant input and collaboration with multiple teams inside and outside the company. One of the biggest challenges was engineering the production rollout schedule & sequence to always allow safe rollbacks and never break production, despite changes being spread across components, teams, and organizations.

Revamped the team knowledge base, and mentored new members as the team grew from 5 to 20 over 2.5 years.

January 2020 **Software Development Consultant, Freelance**

- June 2020 Created factory test fixtures for a local consumer electronics company, Monogram Creative Console. Created an end-to-end solution to validate the i2c and electrical connections between modules. Wrote the firmware, driver, and Python for direct deployment as a factory fixture.

November 2017 **Firmware Engineer, Fitbit, Waterloo**

- February 2019 Worked on the Fitbit Developer Platform, allowing developers to create JavaScript applications to run on Fitbits around the world. Wrote code primarily in C, Python, HTML, and TypeScript.

Took the lead on designing and developing our scientific and signal processing APIs, allowing developers to efficiently process sensor data by offloading numeric code to C. Think μ -numpy.

Was a regular contributor to discussions about our new developer APIs across firmware and mobile, with my contributions being valued enough to make me a gating reviewer, along with the tech leads.

Improved average JavaScript execution time by 3x in order to enable scientific computations to be written directly in JavaScript. This was primarily achieved by optimizing which functions were placed in which memory regions, reducing the time spent in trampolines jumping between regions. Also reduced the JSON.stringify runtime in the JerryScript JS engine from $O(n^2)$ to $O(n)$.

Played a significant role in the successful creation and maintenance of our team's integration tests. Added unit tests for the integration tests, used across Fitbit.

Internships

Fall 2015 **Software Engineering Intern**, *LinkedIn*, Mountain View

Worked on LinkedIn's next-generation distributed graph database, a ground-up rewrite with a much faster core, this time in C++, and much more generalizable datalog query language.

In addition to my regular tasks writing concurrent C++, I proactively improved our test suite runtime from minutes to seconds, and simplified incremental builds from 200 chars to `mint bi`

Spring 2015 **Software Dev. Intern**, *Microsoft Research spinoff - Intentional Software*, Seattle

Worked under Dr. Charles Simonyi, original architect of Microsoft Word and Excel, on an experimental meta-programming platform. Used primarily C++ and C#.

Added advanced text rendering features - ligatures, kerning, RTL layout, and more. Removed more lines than I added. Took the initiative to improve the UI for accepting changed integration tests results - adding keyboard shortcuts and the ability to accept multiple changed tests.

Fall 2013 **Web / Firmware Co-op**, *Pebble Technology Inc.*, Palo Alto

& Summer 2014 Helped bring the modern smartwatch to life.

Mentored a new hire in the rewrite of our filesystem, and designed & wrote the core storage layers on top. Made writes atomic and eliminated storage GC pauses.

Completely rewrote the on-device analytics subsystem - which told us what features users were using. Made tools in Python+Angular for battery lifetime analysis, using my prior analytics work to identify unexpected power hogs. Reduced users getting < 1 day of battery life by 90%.

Helped build PebbleGL, a fixed-point 3D graphics library.

Spring 2013 **Web Developer Co-op**, *Kik Interactive*, Waterloo

Built a fully GPU-accelerated photo viewer, complete with pinch to zoom, panning, and swiping between photos - in HTML, CSS, and JS. Successfully passed QA, after working around numerous bugs in old (and new!) Android WebKit implementations.

Created a new base theme for app.js ⚡, providing theming for apps used by tens of millions.

Summer 2012 **Software Engineering Legend (Co-op)**, *Sweet Tooth Rewards Inc*, Waterloo

Created an analytics system in PHP that allowed optimization of the pricing tiers and model. Built "releases pieces" to enable precise and simple deployment of many projects together.

August 2011 **Technology Consultant**, *Dexterity Consulting*, Calgary

Designed, tested, and deployed a data mining application, using C# and SQL.

Projects

Also on my website at jmcgirr.com ⚡, with pictures, live demos, and code!

BuildBlast, bb.jmcgirr.com ⚡

Fully in-browser multiplayer minecraft-inspired 3d voxel-based game with an editable world. It has pvp attacking, with full server verification and lag compensation. Written using THREE.js and carefully hand-optimized JavaScript, with a multithreaded go backend.

2015 **Trains**, CS452/Realtime Project Course - github.com/crazy2be/cs452 ⚡

Complete microkernel implemented from scratch in C + ARM assembly. Also made a track simulator in Python, & a simple computer vision pipeline using OpenCV and Scipy.

2015 **Hardware MJPEG Decoder**, ECE423 Project Course

The fastest mjpeg decoder our professor had ever seen. Used VHDL + a NIOS II.

2011 **WFDR Web Framework**, github.com/crazy2be/wfdr ⚡

Docker before docker, built in go. Over-engineered microservice manager and reverse proxy.

2017 **Scrawl**, Multilingual pictionary in ~1000 lines of ES6 - scrawl.jmcgirr.com ⚡

2013 **Gitdefence**, Tower defense game in HTML5 canvas.

2010 **Highrise Developer**, Project website in php, MYSQL, and JavaScript

2010 **OpenTower**, Open source simtower in Object-Oriented C++ & SFML

2009 **LiteStep Portable**, UI on USB. C, svn, HTML, JScript, VBScript, and WIN32API