

Ryan Rix: Infrastructure Wizard and Computer Plumber

I can be reached at hireme@whatthefuck.computer Or at +1 (602) 633-5950. You can download this resume [here](#) here.

After four years shipping privacy-enhancing technologies, I am looking to return to working in a Platform Engineering, Cloud Engineering, DevOps, or SRE role.

You can see my code at <http://code.rix.si> and the rest of my knowledge work [online here](#) , and [here](#) .

I am a self-taught software engineer and Linux hacker, and have more than ten years of experience in both small startups and a rapid-growth unicorn startup as backend engineer writing **Python**, **Golang**, **Node.js** and as an **SRE**, **Infrastructure Engineer** and **Privacy Engineer**.

I have been designing a B2B web standard called [Data Rights Protocol](#) which is nearing launch and want to transition back in to full-time backend and architecture development as the protocol shifts from R&D to implementation by the protocol partners.

I am looking for full-time remote work with a growing startup or healthy SME.

I am also open to a 3-6 month contract or contract-to-hire situation working on cloud engineering and platform maintenance.

I would prefer to work with the technologies I've outlined below, but I am a strong generalist willing to learn new tools and ideas.

My Work Timeline and Key Experiences

- July 2021 – September 2023: [Data Rights Protocol](#) with Consumer Reports Innovation Lab, **Django**, **Web Standards**, **Consortium Design**, **Coopetitive Consensus Building**, **Nonprofit Work**
- Sept 2019 – July 2021: Sabbatical and COVID lockdown, **NixOS**, **Rust**, **Work life balance**, **Mentorship**, **Personal Web Publishing**
- July 2014 – Sept 2019: Uber Technologies
 - 2017–2019: Privacy Engineering, Data Portability/DSAR, **Golang**, **GDPR**, **Hadoop Data Engineering**, **HIVE/Presto**, **Cadence Workflow Engine**, **Privacy by Design review**, **Informal SRE/DevOps Support**
 - 2016–2017: Payments SRE, **Python**, **Payments infrastructure**, **PCI Compliance**, **Scalability**, **Site Reliability Engineering Practices**
 - 2014–2016: Dispatch + Realtime Management and Operations, **Node.js**, **Observability**, **Release engineering**, **On-call management and training**, **Handling hypergrowth**
- Sept 2013 – June 2014: Storehouse, Inc. Backend Developer, **Ruby on Rails**, **Amazon Web Services**, **Ansible**, **Product Launch & Go To Market**
- Jan 2013 – August 2013: MadeSolid 3D Printing Company, Co-Founder + Hacker. **Ruby on Rails**, **Hardware Design**, **Startup Networking and Growth**, **Handling Failure**
- 2010–2012: Arizona State University Undergraduate + involved in HeatSync Labs

About me

I live in Eugene, Oregon and am not interested in roles which would require relocation. When I am not at a computer or on my bike I spend my time practicing Zen meditation and supporting my Zen center as board treasurer and bookkeeper.

In my personal work I have been an advocate for **user-focused tools** and building sustainable self-hosting environments. I am currently running a cluster of **NixOS machines** that serve most of my personal computing needs and all of my web traffic -- this site included -- from a real, live server racked in [my living room](#) over **Tailscale**.

I have shipped software written in **Ruby, Python, Elixir, Rust, and Golang**. I practically **live in Emacs** and have built a productive personal environment on top of it and Org Mode, but I won't ask you to care about that or argue that Vim or Emacs are better.

I stopped working in SRE/DevOps just before Kubernetes was "a thing" but can learn how to use it quickly when necessary because I understand the layers and technologies underneath like **containerization, Linux networking and CPU tuning, cgroups, cloud services**, etc.

Here are some more details about the work I've done over the years:

2011: Start up Startup Engineer

Out of University I spent time at a community workshop/coworking space where I built Ruby on Rails apps during the day and at night would build laser cutters and 3D printers and build things with them. This confluence of interests lead me to join a hardware startup in Oakland, CA and move out there where I would ultimately join a few startups before joining Uber in 2014.

2013: Launching an Award-Winning iOS Application

I built the back-end for an iOS application called **Storehouse**, which won an Apple Design Award in 2014 after our launch. I built the first iterations of our **Ruby on Rails API** and a **scalable cloud backend** from scratch on AWS using **Ansible**, successfully scaling from 100 to 100,000 users literally overnight.

The company did well but failed to find product market fit, and I left shortly before the team was acquired by Square.

2014: Hyper-scaling Uber's Dispatching stack

I worked on Uber's software systems which the mobile apps directly interfaced with and which managed the state of all trips in progress. My team was one of the earliest in the company to begin to **adopt Site Reliability Engineering methodologies** from companies like Google because if dispatch went down, all the mobile apps and every customer would immediately be effected. We maintained a suite of **Node.js** services and were responsible for **release management** and **oncall support engineering** for services which for years would see traffic growth at 10% week over week.

As the company grew, my team Realtime Management and Operations grew in to a Site Reliability Engineering team **embedded with teams across the company**. We bootstrapped an entire copy of Uber's infrastructure in China, scaled it faster than our US presence ever needed to and then decommissioned it two years later.

2016: Founding Uber SRE & Hyper-scaling Uber's Payments Stack

As an SRE embedded with Uber's payments team, I consulted with developers on building **resilient**,

highly available systems in the face of global payment providers whose SLAs were at best "wait up to three seconds" and at worst "send a message to a Telegram chat with the developers on the other side of the world and wait until they get back to you about an outage or error scenario". We improved SLAs and architected service changes to insulate important payment and onboarding flows to increase signup and massively reducing dunning costs.

I worked with my team, senior Payments engineers, and our compliance officers to **design and implement a PCI-compliant credit card storage and payments processing environment** from scratch on Amazon Web Services and managed with **Terraform Infrastructure as Code**.

Since much of the company's infrastructure was built using **in-house tooling deployed in physical datacenters**, this isolated environment let us iterate quickly and safely without burdening much of the rest of the company with compliance duties and change management controls. Bringing card processing and vaulting in-house in this manner saved the company tens of millions of dollars without sacrificing developer agility.

2017: Building Uber's GDPR Data Portability and DSAR System

Having shipped this payments environment and been too close to some terrible HR and Management situations (idk,,, just Google around for posts from 2017 about Uber SRE), I moved from the Site Reliability Engineering group in to Uber's **Privacy Engineering** organization to build systems which serviced GDPR data rights requests. After another one of those terrible HR and Management situations led to 125,000 users trying to delete their account at once, the team was in need of **scaling expertise and operational support** and someone who could build out new technologies to service the new rights granted by the European GDPR legislation. I acted as both of these for a small, growing Privacy Engineering group. **The [Download Your Data](#) tool which I designed and lead implementation of still [services these requests](#) for the global user base, delivering data bundles in under an hour on average.**

All throughout my time at Uber I was responsible for taking **engineering interviews** and hiring probably 100 people, I acted as **team lead** and **system architect**, participated in and lead internal career development efforts like **CS/Systems paper club**, architecture and **Privacy By Design reviews**, etc.

Throughout my time at Uber I carried a pager and engaged in developing on-call best practices for my teams and the larger SRE organization. I was part of a small team that did on-call training and worked with internal and external partner teams to triage outages and issues small and large. I am not afraid to carry a pager, as long as I can work to make that process better.

2020: Sabbatical, Lockdown, and Data Rights Protocol

After leaving Uber, I stepped in to a smaller and more tightly scoped role, contracting with [Consumer Reports Innovation Lab](#) designing and implementing **a machine protocol for expressing and responding to Data Rights Request** made under CCPA and GDPR.

I developed a novel web-protocol from scratch which will be used by a consortium of businesses competing in this space to make privacy rights more secure and trustworthy for both consumers and businesses while being cheap to implement. I [presented](#) this work at [USENIX PEPR 2023](#).