

Robert Fielding

Software Developer

7445 Larne Ct
Lorton, VA 22079
(703)944-7006
rob.fielding@gmail.com

EXPERIENCE

Leidos, Reston, VA — Software Developer
September 2023 - Present

Worked on the Prosight project at Leidos, which is a very successful and high-profile airport security project. I mostly worked on the body scanner, to get it integrated into the Prosight network. I also worked on the Explosives and Drugs detection machine to get that also integrated into the network. To a lesser degree, I also worked on the carry-on bag scanner machines as well. I got the first integration with Keycloak working on these devices; which involved C# and Golang. Much of the work involved traveling on-site to do the reverse-engineering work; capturing network HTTPS traffic with a proxy.

greymatter.io, Arlington, VA — Software Developer
October 2015 - July 2023

Worked for a hybrid commercial/government startup from its inception. I introduced Go into the company. The first major product that I worked on (commercial side, delivering to government), was a very successful service for handling file uploads (hundreds of terabytes of files in production, and in production for many years with few major bugs, as it was extremely well unit tested and integration tested in docker before any commits were made). It allowed for high performance audio, video, document attachments with enforced security labels. That was made of Go, S3, MySQL; using nginx/gatekeeper/mTLS for the mesh..

I also worked on iterations of its successor that also supported non-government uses, similarly made of Go, S3, MongoDB. One deployment of this, into government deployments, was all about uploading such files, and efficiently Elasticsearch indexing all of the content, and enforcing the complex security labels on the uploaded content. This service evolved along with our service mesh product; made of Go, Envoy, Kubernetes, etc.

I had checked into the integration tests some demo code to use AWSRekognition to recognize faces, label content, detect inappropriate content, and navigate between familiar faces; in addition to Elasticsearch indexing all results (including the ML labeling).

I had done some work with machine learning, setting up a laptop to accelerate some Reinforcement Learning demos for an NVIDIA GPU; to speed up a team that was doing unaccelerated ML tasks all day.

SKILLS

Go, Java, Python, C.
Performance testing, build automation, security analysis.

LANGUAGES

Go, Java, Python, C, bash, etc.
Some C#, Javascript, etc.
OpenSCAD (3D printing),
Arduino (embedded hardware C++)

Check Point / Network Flight Recorder (NFR Security), Rockville, MD; Tel Aviv Israel — Software Developer

2004 - October 2015

NFR was one of the early IPS/IDS pioneers, with its ncode language for inspecting network traffic. Was on the team that built the Java desktop based user interface, The Alert Browser, and the JBoss based application server (highly modified J2EE), with a MySQL backend. Sentivist Alert Browser had excellent performance, handling a few million events per day per customer; with intuitive real-time visualizations of the alert stream.

Daily work included network-analysis, reading packet captures (ie: decoding HTTP, packet reassembly), automating network setup and firewall rules. Responsible for automating the build that checked out source code and burned an ISO containing a windows client installer and linux installers after the build was gathered from the machines that did the compilation.

Built a WFC (a Windows C# UI framework) based debugger for the ncode language (rebranded as cpcode); which involved the C# code invoking a cygwin built runtime for the C-based cpcode engine; that we had heavily modified to support step debugging. Also worked on the cpcode engine (C based) as it was included in a Linux kernel module for the IPS functionality.

Worked on the Check Point ExtJS (Javascript front-end) web console to the server for about a half year, adding features and fixing bugs.

SensorNet - A project that was experimenting with Hadoop analysis - created a Java map reduce implementation of nested-PageRank over the graph of nodes connecting to each other. The end result was the ability to query for a source subnet and a destination subnet, and get pre-computed graphviz graphs summarizing how the addresses interacted (who attacked who, what was being done, and when).

Project MSSP - A web based firewall log collector. It collected logs from customers's CheckPoint firewalls that were subscribed to this service, for high-level visualization of trends and managing the workflow of analysts that are reading this information in real-time. Was brought on to solve scalability issues; getting queries down from being 24-hour batch jobs to running in real-time. This involved putting a Cassandra-like counters system over top of the Postgres database. The stack included: Django code with Celery/Rabbit for queued messaging, a Java based application server

(Maven, OSGI, etc), ExtJS/JavaScript, and C code.

Wizdom Music, Moonlighting while at Check Point (2004-2015) — *App Developer for iPhone and iPad apps, working with professional musicians*

2010 - 2012

A month after the iPad was released, I produced an app called Mugician which saw significant real-world use from well-known musicians (Underoath, Dream Theater, Gorillaz, The Flying Eyes, etc). Its popularity was mostly explained by its unusually low latency characteristics for an iPad app at the time.)

I worked with other iOS developers on fixing (and documenting) MIDI implementations to handle special problems that a continuous multi-touch surface allows. Various vendors such as Roli Labs and Roger Linn Labs have turned this into a formal specification. I created source code and build instructions for Arduino devices that do MIDI filtering to alter the behavior of standard keyboards, including some battery-less builds that draw their power from the input signal. (My github repo “rfielding/octaveRounder”).

ComTek (under Northrop Grumman and DISA) – Software Developer

1999-2004

Worked on multi-machine application suites primarily providing APIs wrapping up LDAP for DoD PKI applications. This involved client-side SSL authentication (software and smartcard tokens).

Authored a Java Servlet based UI toolkit (similar to Echo2 framework before it became JavaScript based) to deal with scenarios where JavaScript was not yet allowed for security reasons.

Federal Reserve – Programmer

1996-1999

Worked on banking surveillance applications. Generally, work was done in Power-builder (and occasional bits of Win32 C API). While there, I introduced the use of Java for these kinds of applications, and the first use of Java in production was by way of Silverstream.

EDUCATION

George Mason University, Fairfax, Virginia — Computer Science BS

1993 - 1996

Computer Science degree. Focus on compilers, operating systems, discrete mathematics..

NOVA Community College, Woodbridge, VA

1991-1993

Getting credits out of the way to go on to George Mason for university

PROJECTS

Gosqlite - Sketch out a CMS foundation

When content is uploaded into a site, all of the content should be full-text indexed (with sqlite). But most file formats need a text extract (pdf, doc), or a Machine Learning service for images (Google Vision, AWS Rekognition, etc) to extract text to describe them (ie: alt-text). So, a pipeline is required to do text extracts, thumbnails, permissions, reverse-proxy urls to assist uploaded static apps (ie: React apps mounted in). It is an all in one simple container to focus on sketching out the idea.

<https://github.com/rfielding/microcms>

OctaveRounder — *Arduino Hardware device for MIDI protocol proxying*

This was my first real Arduino project. It takes advantage of Arduino hardware to make professional quality (ie: low latency, high reliability) MIDI filter devices. The compiled binary fits onto a device with only a half-kilobyte of memory, and is powered off of the input signal (which is unusual for a MIDI device):

<https://github.com/rfielding/octaveRounder>

BrailleTools - Random Braille tools that includes a 3D printable USBC iPhone case with a mechanical keyboard in Computer Braille layout. This lets you code fully on an iPhone, and type without looking at the iPhone while it's still in its case and in your hands. It also includes code to perform ASCII to computer braille translation; like the liblouis command for 8-dot computer braille.

I use this braille keyboard every day, because it is part of my phone. When I had vision issues and had trouble reading, it turned out that this iPhone case (3D printed and Python coded) was the best preparation I made.

<https://github.com/rfielding/brailleTools>

There are other projects that you might find interesting as samples. Mugician was the most starred app, because

many people were making clones of it to put into the app store.

<https://github.com/rfielding>