J. Seth Daniel

job@sethdaniel.org

https://github.com/WhoIsSethDaniel

https://www.sethdaniel.dev/

214-886-3770

**SUMMARY**

Graduate of Texas A & M University, with over two decades of experience in

software engineering, seeking full-time software engineering position. Strong

background with many variants of UNIX, problem solving, software development,

and troubleshooting. Experience with web services and APIs, microservices,

networked applications, database driven applications, multi-process

applications, web development, and tool development.

**SKILLS**

*Operating Systems*: Linux (Centos, Redhat, Debian, Ubuntu)

*Programming*: Go, Perl, web-based APIs, web services, Python, Erlang, Bash

*Databases*: MySQL, PostgreSQL, Riak, ElasticSearch

*SCM*: Git (GitHub and GitLab)

*Observability*: Grafana, Graphite, Prometheus, Icinga, Icinga 2, Collectd

*Other*: Docker, GoCD, Jenkins

**EXPERIENCE**

*Silicon Shuttlecock*

Senior Software Developer

October 2019 - April 2020 (company folded)

Worked on services, written in both Go and Perl, that were meant to be used

for the tracking and management of silicon wafers during the assembly of

semiconductors.

Unfortunately the software did not sell and the company folded in April 2020.

*Xome*

Senior Software Developer

October 2018 - September 2019

Joined a small team of developers at Xome. Worked on breaking apart a small

monolithic application into smaller services. Each service had a Postgres backend

to store (mostly) JSON data using the JSONB column type in Postgres. Each

service exposed a REST interface and (initially) all intra-service communication

was done via REST. All code ran within Docker containers on locally managed

hardware.

Maintained an accounting service of primary interest to the accounting group

within Xome. Worked closely with the accountants when adding new features, and

made all technical decisions for the team. Was able to convert multiple by-hand

projects into automated projects to the great relief of the accountants.

Added the first asynchronous requests to the new service architecture by

introducing a job queue and having requests place jobs on the queue. The job

queueing software used is named Minion, a Postgres based job queueing system.

It fit well within the current architecture and expertise of the team. Introducting

a different system, such as RabbitMQ, would have been more difficult and Minion

worked extremely well given the architecture and performance needs of the project.

*OpenX*

Senior Software Developer

February 2010 - September 2018

Was the founding member and team lead of the Automation Engineering and

Observability team at OpenX. This team was directly responsible for a number

of things including a system that tracked all of the hardware in each

datacenter and what role each piece of hardware was meant to fulfill, along

with monitoring, alerting, graphing, imaging, and central tooling.

Wrote and maintained the system that helped OpenX organize and administer its

data centers (several data centers worldwide with nearly twenty thousand

servers). This system was used to track machines, network switches, IPs, PDUs,

and many other items in each data center. Among its many functions this system

wrote configuration files for various monitoring systems, controlled which

users could access which hosts, and re-provisioned hosts.

At the core of the data center management system is a web-based API that

allowed for things such as creating virtual machines, allocating physical

machines, rebooting hosts, and creating new images for our machines. This API

was backed by several different services each using its own backend (primarily

Postgres, but also Elasticsearch). Long running, asynchronous jobs were

delegated to a job queuing system.

Designed and performed initial development on a monitoring system that tied

together Grafana, Graphite, Icinga 2, the data center management system, and

an in house metric collection system. This monitoring system simplified and

improved the way monitoring and alerting of metrics occurred at OpenX.

All of the services my team was responsible for were tested and deployed via

our CI/CD system (GoCD at first, followed by Jenkins). We were the first team

at OpenX to use continuous deployment and we helped other teams that wished to

move toward CD (be it continuous delivery or continuous deployment).

Testing (unit, functional, operational) is very important. The codebase had

well over 13,000 tests. Most of the code had well over 90% test coverage.

*Oversee.net*

Senior Software Developer

June 2008 - February 2010

Primary developer on next generation domain ad serving platform. This included

a custom web server frontend and many backend servers. The entire system was

asynchronous. The backend servers were all built from a common server

framework. I wrote the web server frontend, the common server framework, and a

number of the backend servers.

Wrote a packaging and release system built on top of Jenkins. Using

Jenkins we could package up our code (using RPM) and release it to QA.

After validation by QA the code could be released to production using the

same mechanism.

*Yahoo! Search Marketing*

Senior Software Developer

December 2005 - May 2008

Worked on and maintained a web-based delivery platform for statistics and

reporting for Yahoo! Search Marketing. This system was used by thousands

of internal employees for gathering information about many diverse

advertising related topics. It was built using Perl, mod\_perl, and

Apache 1.3. It communicated with a number of database backends (Oracle,

Postgresql, and an internal data warehousing technology). A separate

team wrote the reports that this system would run and display.

Developed the communication layer for our reporting platform to query an

internal Yahoo! data warehousing technology. This layer was asynchronous

and exposed an API which was used by the developers of the reports.

Worked on and enhanced the SOAP API which our reporting system exposed.

This API allowed automated processes to authenticate, run a report, and

gather results.

Worked closely with the developers of the reports (a team with as many as

20 people). Taught new report developers how to write reports, led

instructional meetings regarding new features, and gave day to day

support.

*The Planet*

Senior Software Developer

March 2005 - November 2005

*Vindicia*

Senior Software Developer

June 2004 - February 2005

*Pulse Entertainment*

Software Developer

September 2002 - May 2004

*Zack Systems*

Software Developer

October 2000 - March 2002

*Texas Instruments*

Software Developer

January 1997 - September 2000

**EDUCATION**

Texas A&M University

Bachelor of Computer Science, 1996