



Wilbert (Wil) Moore III

POLYGLOT SENIOR SOFTWARE ENGINEER, DEVOPS, SRE



github.com/wilmoore



linkedin.com/in/wilmoore



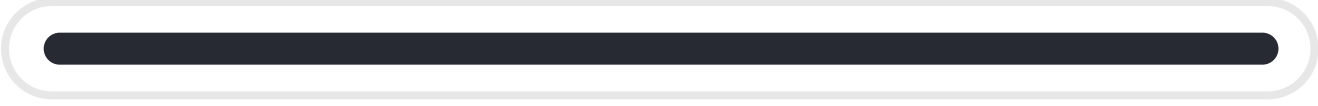
wil.moore@polyglotsoftware.io

“I have experience in a lot of technologies; however, that isn’t as interesting as the fact that I understand the following at a conceptual level: The HTTP protocol, regular expressions, pragmatic test driven development, when, why, and how to use design patterns, most design patterns exist to work around mostly OOP language gaps, shell exit codes and assert(), the unix/linux shell environment, database indexing vs inverted indexing and materialized views, hash tables, hash functions and their relationship to inverted indexes, and finally, the relationship between Linux containers (LXC) and Docker. Almost everything else today is an implementation detail and a convenience on top of these concepts.”

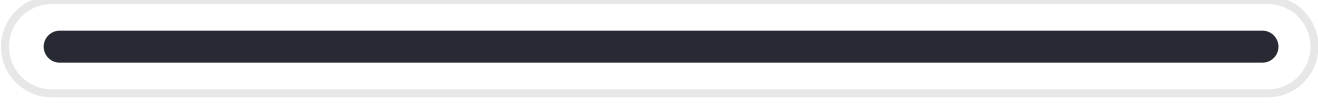
PROGRAMMING LANGUAGES

Languages I’ve shipped to production, refactored to another language, authored an open-source library or framework for.

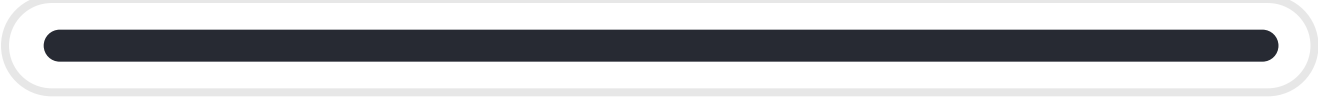
JAVASCRIPT (V8/NODE.JS, BROWSER/DOM)



JS COMPILED (TYPESCRIPT, BABEL, COFFEESCRIPT)



PYTHON



RUST *



ERLANG / ELIXIR



PHP



RUBY



KOTLIN *



GO *



JAVA (VERSION 8) *



REGEX, AWK, SED



SQL (DDL/DQL/DML, SPROCS, TRIGGERS, VIEWS)



SHELL SCRIPTING (SH, BASH, ZSH, FISH)



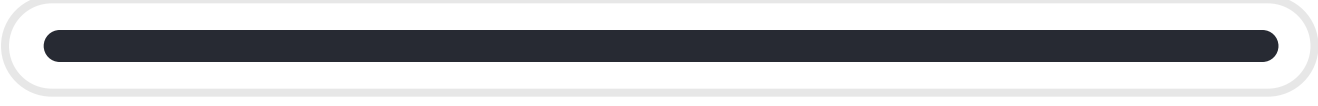
MARKUP / STYLE (HTML/5, CSS/3, SASS, LESS, JSX)



DATABASES

Databases I’ve deployed and/or managed in production scenarios, or I am actively learning.

RDBMS (POSTGRESQL, MYSQL, SQL SERVER, ORACLE)



NOSQL (DYNAMODB, REDIS, MONGODB)



GRAPH (NEO4J) *



SEARCH (ELASTICSEARCH)



WORK EXPERIENCE

My contract and direct hire experiences with detailed depictions of notable projects I’ve delivered and in some cases, the reasons behind the tech decisions.

Senior Full-Stack Software Engineer, devOps, SRE
Nordstrom - Denver, CO

(CONTRACT) MAR 2018 - FEB 2019

[API ARCHITECTURE + IMPLEMENTATION, BACKEND, DEVOPS] Designed and implemented a **near real-time messaging system** with message **fanout** arriving via SSE. **SSE (EventSource)** was choosen over **Websockets** as my experiments and research revealed that SSE is superior in simplicity of debugging, testing, and deployment over Websockets, especially when client-to-API two-way messaging is not needed. Deployment is **automated via Terraform and CloudFormation**. The **RESTful Swagger API (OpenAPI)** written in **YAML** format is fronted by **AWS API Gateway** and **proxies to AWS Lambda** which persists messages to **DynamoDB** for long-term storage, which subsequently emits an event to a **DynamoDB Stream** and is picked up by an AWS Lambda function which produces to a **Kafka topic**. The Kafka topic is watched by a **Node.js worker process** and eventually fanned out to clients via SSE.

[FULL-TEXT SEARCH SCHEMA DESIGN, IMPLEMENTATION, BACKEND] Designed and implemented an **Elasticsearch** schema and cluster for a search/filtering strategy which replaced an existing **Redis** cluster which was previously merging results in-flight which was less than efficient and not very flexible in terms of agility or surviving change. The Elasticsearch cluster was deployed in **AWS** on **EC2** instances. I crafted and **documented several Elasticsearch queries** per the team’s requirements as well as delivered performance metrics via **Artillery-based load testing**. Integrated a simple **Elasticsearch HTTP client** along with the queries into an existing **Node.js/TypeScript API** (not originally maintained by me or my main team – helped another team out with this implementation).

[DB SCHEMA DESIGN, DB INDICES, DEVOPS, BACKEND] Contributed various production features and enhancements to a **Node.js Apollo GraphQL API** deployed to **AWS ElasticBeanstalk** fronted by an **Nginx load balancer in a VPC**. I increased response throughput by +3x (at least) by way of **adding database indices** and **pagination** in hot portions of the source. The performance enhancement was verified with pre and post load testing with **Artillery**. Wrote custom express middleware as a provider for mocking the calls to the **Node.js GraphQL API** for **Cypress.io** testing a **React** front-end application.

[DOCUMENTATION, DEVOPS, SRE] Authored sections of a **Docker/Kubernetes deployment tutorial**. Authored an **incident response guide: a guide for on-call SREs** which included examples of previous incident responses along with **screenshots**. Built **dashboards, monitors and alerts** in **Datadog** and **Kibana**, with integration to **PagerDuty** and automated with **terraform**.

Senior Full-Stack Software Engineer
Charter Communications - Greenwood Village, CO

(CONTRACT) OCT 2017 - MAR 2018

[API ARCHITECTURE + IMPLEMENTATION, BACKEND, DEVOPS] Designed and implemented **micro-service APIs** to automate **test account creation** through **orchestration** of several internal **APIs with Node.js** and the **IBM/StrongLoop Loopback** API framework. Developed **custom Loopback middleware** and **mixins** for **uploading** and persisting **attachments** in **base64** format.

[DB SCHEMA DESIGN, DB INDICES] Using raw **SQL DDL**, designed and created **MySQL database schemas** to support new features and enhancments in the API.

[DOCUMENTATION] Authored a new **developer onboarding guide** which covered getting started with contributing to the project, getting around the company’s campus, and how to setup the VPN.

Senior Full-Stack Software Engineer
Shutterstock - Denver, CO

(CONTRACT) FEB 2016 - SEP 2017

[API ARCHITECTURE + IMPLEMENTATION, DB SCHEMA DESIGN, DB TUNING] Maintained and refactored an e-commerce **Node.js API** on top of the **express** framework backed by a **MySQL** database using **test-driven development (TDD)**. Implemented a **currency exchange API** with **Ruby** and the **Sinatra** framework backed by a MySQL database using test-driven development (**TDD**). Contributed to **Node.js orchestration APIs** and built JavaScript **API clients** using test-driven development (**TDD**), including **npm packaging** and release into a **private npm registry**. Evolved the **schema** of a MySQL database through features and enhancements through **DDL** and **DML SQL change scripts**. Introduced a **message queuing** strategy through use of the **RabbitMQ** message broker. Built **architecture diagrams** and facilitated architecture reviews, and authored **developer guides**.



Wilbert (Wil) Moore III

POLYGLOT SENIOR SOFTWARE ENGINEER, DEVOPS, SRE



github.com/wilmoore



linkedin.com/in/wilmoore



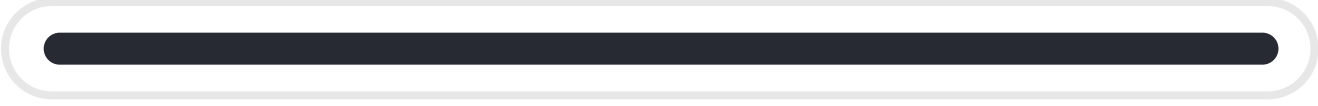
wil.moore@polyglotsoftware.io

*“Unfortunately, many software developers suffer the following (I must concede that I’ve at times struggled with this as well): we sometimes fall in love with technologies, languages, and frameworks which can cause tunnel vision. **Even more unfortunate is the fact that many hiring officials have and continue to blindly dismiss great candidates on the premise of one easy to train technical gap; these people may posses more passion for the product than even the hiring official, and these gems, these excellent people are dismissed before they can shine their light ... and this over some slight gap in experience with an esoteric tool which could be trained up on in a few hours for a few dollars...sorry, but you can’t train a lack of passion for the cause.**”*

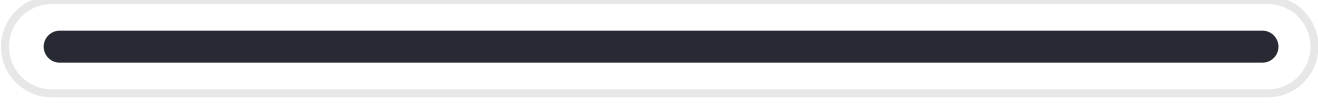
CLOUD TECHNOLOGIES

Technologies that I’ve worked with in the cloud from the perspective of architecture as well as implementation.

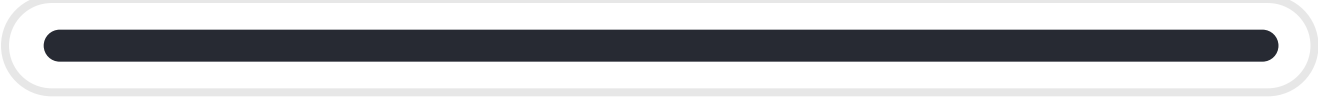
COMPUTE (AWS EC2/ELASTICBEANSTALK, HEROKU)



FUNCTIONS (AWS LAMBDA, SERVERLESS FRAMEWORK)



NOSQL (AWS DYNAMODB, ELASTICACHE)



RDBMS (AWS RDS POSTGRESQL & MYSQL)



STORAGE (AWS S3, RACKSPACE CLOUD FILES)



QUEUE (AWS SQS, RABBITMQ)



MESSAGING (AWS SNS)



DATA WAREHOUSE (AWS REDSHIFT)



CONFIGURATION (TERRAFORM, AWS CLOUDFORMATION)



LOAD BALANCING (AWS ELB)



DNS (AWS ROUTE 53)



CDN (AWS CLOUDFRONT)



MONITORING (AWS CLOUDWATCH, DATADOG, KIBANA)



STREAM (AWS KINESIS, KAFKA)



NETWORK (AWS VPC)



WEB SERVICES

Web service (or micro-service) technologies in which I’ve implemented the producing side (the service itself) and/or a consuming client.

REST & RESOURCE-BASED SERVICES (OPENAPI/SWAGGER)



GRAPHQL (APOLLO-BASED & VANILLA NO-FRAMEWORK)



SOAP (CLIENTS)



WORK EXPERIENCE

My contract and direct hire experiences with detailed depictions of notable projects I’ve delivered and in some cases, the reasons behind the tech decisions.

Senior Full-Stack Software Engineer
Aetna (formerly iTriage) - Denver, CO

(CONTRACT) MAY 2015 - FEB 2016

[API ARCHITECTURE + IMPLEMENTATION, BACKEND, DEVOPS] Implemented and deployed multiple **RESTful APIs with Swagger, Node.js, Hapi, and Docker**. Contributed to **analytics client libraries** writing in **JavaScript** and **Objective-C**. Contributed to a **Ruby/Sinatra** RESTful API. Contributed to an analytics platform built on top of **AWS Redshift**. Built several reusable project **templates** and **boilerplates** for quickly **scaffolding** backend **APIs** including documentation with careful consideration for developer experience (**DX**). Service instrumentation with StatsD, end-to-end testing with WebdriverIO/Selenium. Performed **R&D prototyping** with **Elasticsearch** for a project that unfortunately never materialized.

[FRONT-END] Implemented a browser-based **single-page-application (SPA)** secure messaging dashboard application utilizing the Angular framework.

Senior Full-Stack Software Engineer
Welltok - Denver, CO

(CONTRACT) AUG 2014 - MAY 2015

[API ARCHITECTURE + IMPLEMENTATION, BACKEND, DEVOPS] Designed and implemented Several micro-service APIs with Node.js, Express which powered the core platform for CafeWell Concierge. Concierge which was comprised of a native mobile application for health care benefits recipients and a web application for administration. APIs were deployed via **Docker** to **AWS ElasticBeanstalk** with some assets being deployed to **AWS S3** and ditributed through **AWS CloudFront**. The administration dashboard was written in **Ruby/Rails** and the **Rails Admin** package with deployment to **AWS ElasticBeanstalk** with **Docker**.

[SEARCH / AUTO-COMPLETE] Utilized **Elasticsearch** to build a search / auto-complete service which was exposed via a **Node.js RESTful API** using the **Koa** framework. The **auto-complete** service was consumed by the Concierge mobile application.

[EXTERNAL SERVICE API INTEGRATIONS] As Concierge was meant to be a “cognitive” application, there was some integration with IBM Watson (cognitive computing engine that that won Jeopardy).

Designed and implemented a **RESTful API** using the **Node.js Koa framework** with integration with the **Active.com API** (http://developer.active.com/docs/read/v2_Activity_API_Search). Packaged as a **Node.js module, Open-sourced the code**, and published to the **npm registry** (<https://www.npmjs.com/package/active-activity-search>).

[ETL] Migrated a legacy Python **ETL** project that was not written for scale. Re-implemented the ETL project as **data transformations** powered by **Node.js** streams. **Packaged as a Node.js module, Open-sourced the code, and published to the npm registry** (<https://www.npmjs.com/package/text-pipe>).

Senior Front-End Software Engineer
Time Warner Cable - Broomfield, CO

(CONTRACT) AUG 2012 - MAY 2015

Built an **HTML5 single-page-application (SPA)** and **HLS streaming video streaming video player** with custom scrubbing features. This system was targeting a **whole-home IP TV** (cable set-top box) system targeting millions of customer homes. Utilized **test-driven-development (TDD)** to implement and deliver features, bug-fixes, enhancements, and performance tuning.

CI & and releases using **Puppet, VirtualBox, and Jenkins**. Unit/Integration testing w/ **Mocha, Jasmine, Sinon, Chai, and PhantomJS**. Build automation with **GNUMake** and **continuous integration** with **Jenkins. Node.js** for utility scripting, testing, and backend stubbing.

Designed & Developed various HTTP API clients using **jQuery AJAX, Superagent, Promises**, higher-order functions, and “README”-driven development. API clients for retrieving and playing HLS streams as well as clients for retrieving emergency (**EAS**) notifications and application updates. Originally utilized **WebSockets** which were switched to **Server Sent Events, Mocha, Grunt, Gulp, and GNUMake**.



Wilbert (Wil) Moore III

POLYGLOT SENIOR SOFTWARE ENGINEER, DEVOPS, SRE



github.com/wilmoore



linkedin.com/in/wilmoore



wil.moore@polyglotsoftware.io

“After about your **4th** or **5th programming language, testing framework, cloud service, template engine, or relational database**, one starts to realize that **ramp up** on new technologies **no longer takes years or months** ... it takes **minutes** to get the **general concept, days** to **build something** useful, and **weeks** to start **working at expert level**.”

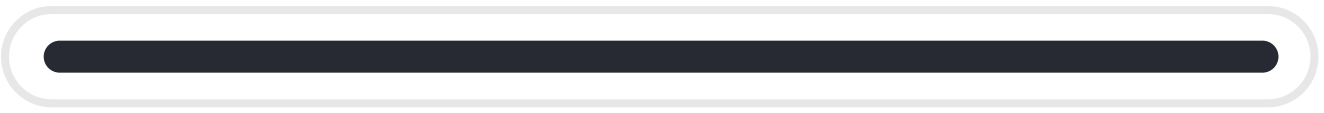
TEST FRAMEWORKS

Automated test frameworks that I have used and feel comfortable writing new or modifying existing test cases.

JS (TAPE, MOCHA, JASMINE, TESTEM, KARMA, JEST)



RUBY (MINITEST, RSPEC)



PHP (PHPUNIT)



API FRAMEWORKS

API libraries and frameworks that I've worked with.

NODE.JS (EXPRESS, KOA, FASTIFY, CONNECT, HAPI)



NODE.JS (APOLLO GRAPHQL, STRONGLOOP LOOPBACK)



PYTHON (FALCON) *



RUBY (SINATRA)



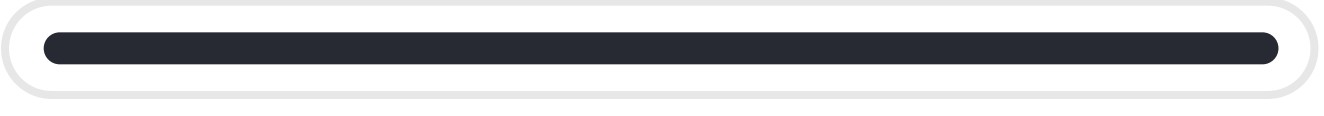
WEB & MOBILE FRAMEWORKS

API libraries and frameworks that I've worked with.

REACT, REACT NATIVE



VUEJS



ANGULAR



BACKBONE



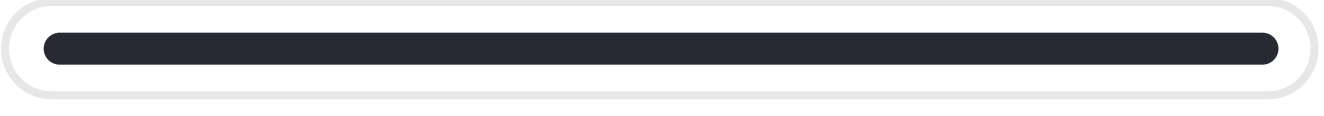
CI/CD

Continuous Integration and continuous deployment systems that I've used, setup, and/or maintained.

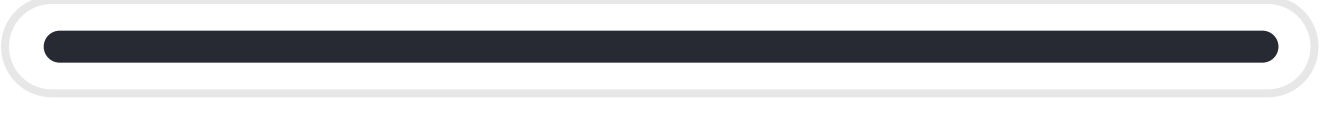
JENKINS



GITLAB CI



SEMAPHORE, CIRCLE, TRAVIS



WORK EXPERIENCE

My contract and direct hire experiences with detailed depictions of notable projects I've delivered and in some cases, the reasons behind the tech decisions.

Senior Full-Stack Software Engineer

DEC 2011 - AUG 2012

Net-Results - Golden, CO

[FRONT-END, BACK-END, UI/UIX, DEVOPS] UI/UX design and implementation for a multi-tenant browser-based **single-page-application (SPA)** for customer **cloud file management** interface on top of **AWS S3** with a **PHP 5.3** backend.

[FRONT-END, BACK-END] Contributed to a **REST API** for **account provisioning** and client maintenance with **jQuery** and backed by **MongoDB**. Contributed to a **multi-tenant** browser-based **single-page-application (SPA)** for building marketing landing pages using **drag-and-drop** using HTML5, CSS3, and **YUI3**.

(Team Lead) Senior Full-Stack Software Engineer

JUN 2010 - AUG 2011

Navigant (formerly Ethos Partners) - Denver, CO

[API ARCHITECTURE + IMPLEMENTATION, BACKEND, DEVOPS] Lead a **team of 3** developers in successfully re-architecting a legacy monolithic web application that **allowed health care providers create health risk assessments (HRA)** with custom business rules, **branching**, and demograph intake.

The legacy version included no architecture diagrams, no documentation, 1% test coverage, and an abyss of poorly designed MySQL database tables which were missing key indices. The original software was written with PHP 4.x which my team replaced with **PHP 5.x** using the **Zend Framework**, **95% test coverage**, an increase of **94%**.

Mid-Level Software Engineer

(CONTRACT) OCT 2009 - JUN 2010

Kaango - Greenwood Village, CO

Designed, implemented, and documented a Full-Stack application utilizing **PHP, HTML, JavaScript**, and **CSS** for **managing classifieds**. Implemented a job queue on top of **AWS SQS (Simple Queue Service)** for handling batches of classified ads which were **transformed** before being persisted into a **MySQL database** as well indexed into **Solr** for **full-text search**.

Designed, implemented, and documented a Full-Stack application utilizing **PHP, HTML, JavaScript**, and **CSS** for a **question and answer website** (similar to StackOverflow). Implemented a authentication and **RBAC** (Role Based Access Control) system in **PHP**. Refactored poorly performing **SQL** queries by **tuning indexes**. **Increased unit test coverage** from 3% to approximately 80%.



Wilbert (Wil) Moore III

POLYGLOT SENIOR SOFTWARE ENGINEER, DEVOPS, SRE



github.com/wilmoore



linkedin.com/in/wilmoore



wil.moore@polyglotsoftware.io

“I enjoy a team culture that fosters **role acceptance, trust, and communication**. This is critical for team cohesion. What I won’t stand for is criticizing and talking behind people’s back. This is not the way to accomplish a **strong team dynamic**. It doesn’t matter how smart you are, how many dollars you’ve saved the company, nor how many favors you’ve done for various people throughout the company. Playing politics has no place on any team. This is why I’ve let very few teams hire me directly; but you’re not supposed to tell the truth in interviews 😏. Some developers are **naturally open and vulnerable problem solvers**; others are information gatherers. Know your personnel. Do not demean people for being who they are. Role acceptance is not a mediocracy; We accomplish more being kind and working within the strengths of our people.”

WEB MAPPING

Web mapping libraries, frameworks, and Web APIs that I have worked with.

MAPBOX



LEAFLET



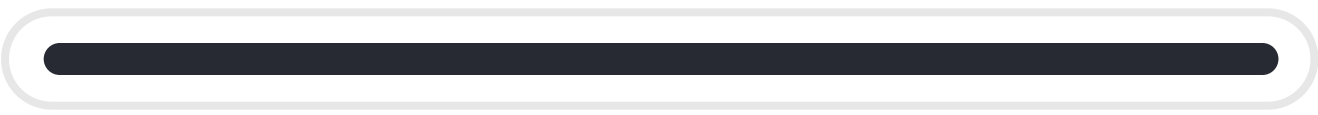
WEB APIS (GOOGLE MAPS, MAPQUEST)



CONTAINER PLATFORMS

Containerization platforms, including container orchestration platforms that I’ve worked with.

DOCKER



KUBERNETES (DEV DEPLOYMENTS)



MESOSPHERE (DEV DEPLOYMENTS)



GRAPHICS, UI, DIAGRAMS

Vector graphics, photo editing, illustration, desktop publishing, UI prototyping, and diagramming programs I’ve worked with.

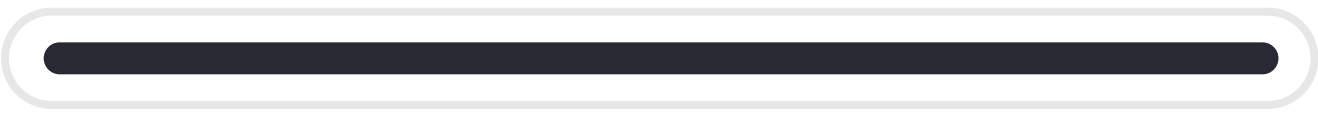
VECTOR (ILLUSTRATOR, AFFINITY DESIGNER, FREEHAND)



PHOTO (PHOTOSHOP, PIXELMATOR)



DIAGRAMMING (OMNIGRAFFLE, BALSAMIQ, LUCIDCHART)



PUBLISHING (PAGEMAKER, INDESIGN, QUARKXPRESS)



NOW LEARNING *

Technologies that I am currently actively learning are listed here. Because I’ll run out of space here, look for items with an asterisk * next to them. Those are items that should also be here.

SOFTWARE ENGINEERING MANAGEMENT *



DEEP LEARNING *



DATA SCIENCE *



KUBERNETES OPS WITHOUT HIGH-LEVEL TOOLING *



PRODUCTS

Entrepreneurial product ventures as well as notable things I’ve published...

CTO, Senior Full-Stack Software Engineer TipOff, LLC.

JAN 2019 - PRESENT
tipoffgame.com

I stepped in as **CTO** of a **mobile game** called **TipOff**. The previous developer stopped working on the app, then subsequently dissappeared and left the two co-founders **without source code** and no access to the **CMS** they were using to manage the game’s content.

As the source code was not available, I quickly learned the **React Native platform** with **expo** and **fastlane**. I was able to re-build the game from scratch in about 2-weeks. Though this was a minimal viable version of the game with some non-pertinent features missing; I had a working game ready to be published for **iPhone**; though, I conceed that my builds were rejected several times for over one week before being accepted. I ran into various issues, mostly to do with screenshots; however, I also once attempted to publish an erroneous build; however, that experience taught me how to correctly version my builds. Obviously **previous exposure to React** and the **npm ecosystem** helped keep the time to build relatively short.

Book (Author) Inside GraphQL

DEC 2018 - PRESENT
STATUS: Chapter 1 (started)

“**Inside GraphQL**” is a book that seeks to detail the **internals** of some of the most important **projects** and **products** within the **GraphQL ecosystem**. This book is **not an introductory book**. There are plenty of those. This book is for power-users and developers that want to contribute to the ecosystem or simply understand how to tear down and debug a running **GraphQL service**.

Lead Instructor, Full-Stack Software Engineering Polyglot Software (Software Engineering Education)

DEC 2018 - PRESENT
wil.moore@polyglotsoftware.io

The current state of software engineering education does not scale well. I’ve mentored enough Junior software engineers to realize that most of their education stops as soon as they get hired. Some of the more dilligent that seek mentors do better; however, they are essentially cut off from their original source of education which is the bootcamp.

Full-Stack Software Engineering Polyglot Software (GeoSpatial SaaS Application)

DEC 2018 - PRESENT
STATUS: UI (50%)

Pointmap is a GeoSpatial-based SaaS application that allows physical store owners with many locations (i.e. Apple) to provide a beautiful (and working) store locator which integrates into their existing website either by opening in a new window or via pop-over. We will in the future offer web component integrations with libraries such as React and Vue.



Wilbert (Wil) Moore III

POLYGLOT SENIOR SOFTWARE ENGINEER, DEVOPS, SRE



github.com/wilmoore



linkedin.com/in/wilmoore



wil.moore@polyglotsoftware.io

“I believe that every professional software engineer should have at least a basic understanding of the fundamentals of systems administration; starting with their own computer. I believe that every professional software engineer should understand at least the basics of functional programming as well as how to use classes properly and when to use them. I believe that every professional software engineer that will be handling user data must have at least a basic understanding of multi-byte characters, unicode, and the concept of timezones. I believe that every web professional should have at least a basic understanding of how DNS, HTTP, HTTPS (SSL/TLS), and SMTP works.”

CERTIFICATIONS

Various technical certifications that I've successfully tested for over the years.

ZEND CERTIFIED ENGINEER - 2010 (ID: ZEND015278)

<https://goo.gl/T6TePT>

MICROSOFT CERTIFIED PROFESSIONAL (ID: 2592587)

FORMAL EDUCATION

Formal education for which I did not receive a degree as I started a family and my career and I decided that those things were more important to me.

CSULB - INFORMATION SYSTEMS (1996 - 1999)

UNIVERSITY OF PHOENIX - INFORMATION SYSTEMS ('02 - '03)

OPEN SOURCE, ARTICLES, CONFERENCE TALKS

Open-Source projects I've authored, articles I've written on software development, and conference talks I've given that may be moderately interesting to mention.

selectn.js

npmjs.com/package/selectn

RELEASED: MAR 2013

~368K downloads per month

selectn.js is a curried **property accessor** function written in **JavaScript** for **Node.js** and the **browser** that resolves **deeply-nested object properties** via dot/bracket-notation string path while mitigating **TypeErrors** via friendly and composable API.

I wrote **selectn** because I noticed that **most of my colleagues** on the **HNAV** project during my time at **Time Warner Cable** were **repeating the following pattern** throughout the codebase:

person** && **person.info** && **person.info.name** && **person.info.name.full

selectn.js allows you to refactor the above **to the following**:

selectn('info.name.full', person)

php-version

github.com/wilmoore/php-version

RELEASED: JAN 2012

@ GitHub 582 stars, 43 forks

php-version is a command-line **PHP version manager**. I wrote **php-version** so as to easily switch between different versions of PHP since at the time I was frequently switching between different client PHP projects. Though I've personally moved past using version managers as part of my everyday development workflow, I still maintain it as it's useful to many developers. I plan to re-factor it away from being a bash shell program to a compiled binary written in **rust**.

uuid-regex

npmjs.com/package/uuid-regex

RELEASED: JAN 2016

~5K downloads per month

uuid-regex is a **JavaScript** regular expression (**RegExp**) for finding an **RFC4122** compliant **UUID** in a string. While it may seem unintuitive that something as tiny as a single regular expression would need to be open-sourced, consider the fact that this problem no longer needs to be solved in the JavaScript ecosystem and if I missed some edge-case, it will be resolved in a single place.

curry2

npmjs.com/package/curry2

RELEASED: MAY 2015

~100K downloads per month

curry2 curries a binary function, also said to be a function that expects 2 parameters or arguments. **curry2** is useful in **JavaScript** because unlike **true functional languages** where functions are **automatically curried** (i.e. **Scala**, **Haskell**), in order to approximate the same, a lot of boilerplate code must be written to wrap the function you want curried. There are other libraries that will wrap a function so it acts like a curried function; however, most of them trade-off clarity and performance for convenience of currying any number of parameters. **curry2** makes the trade-off of only supporting functions with an arity of 2 (thus a **binary function**) as I realized that I never had the urge to curry anything other than a **binary function**. You might also like to check out the following article that I wrote back in 2016 about some of the benefits of currying in JavaScript.

<https://medium.com/@wilmoore/un-bind-your-js-with-curry-a8657a4138cb>

Confoo Montreal Conference

<https://confoo.ca/en/speaker/wil-moore>

2012 & 2013 Conferences

<https://joind.in/user/wilmoore>

Conference Talks

10 advanced, yet digestible SSH techniques
speakerdeck.com/wilmoore/10-advanced-yet-digestible-ssh-techniques

Painless, version-controlled database refactoring
speakerdeck.com/wilmoore/painless-version-controlled-database-refactoring

DIY Private Cloud w/ VirtualBox and Chef
confoo.ca/en/2012/session/diy-private-cloud-w-virtualbox-and-chef