TJ Horner | Software Engineer | New York City

(760) 677-1329 | tj@tjtjtj.tj | horner.tj/github | horner.tj/linkedin | tjhorner.dev

TECHNICAL SKILLS

Strong: JavaScript (ES6+), Node.js, Golang, Webpack, HTML5/CSS3/SCSS, git, Shell/Bash, *NIX System Administration, React, Redux, Relational and Non-relational Databases, Network Design, Apache, Electron, Project Management, Team Leadership, Google Cloud, GitHub Actions Automation, Protocol Buffers, Kotlin, Swift, Android, iOS, Ruby on Rails, Microsoft Azure, Docker, C#/.NET/.net core

Experienced: AWS, Travis CI, Jest, TDD, C, Agile/Scrum, Java, PHP, jQuery, Structured Text, TypeScript, NGINX, Caddy, Traefik, Unity 3D, Unreal Engine, Source Engine, Scala

PROFESSIONAL EXPERIENCE

MakerBot | Full Stack Software Engineer **2019 - Present**

- Managed software to connect and manage an average of **50,000 3D printers** concurrently to the internet by utilizing highly scalable RESTful Node.js microservices on GCP with Kubernetes
- Minimized size of printer toolpath file **40-fold** by developing a Protobuf-based file format, increasing user workflow speed by several hours
- **Technical lead** for project involving refactoring legacy server-side-rendering PHP 5.0 back-end with TypeScript React-based front-end, improving speed and performance by **10x**
- Introduced build pipeline into development to push static assets into Fastly/S3 content delivery network, reducing load times on critical MakerBot properties by up to **10x**
- Developed internal tools to assist with rapid testing, including a fully-featured 3D printer emulator written in Golang with a React front-end, dramatically reducing development time for new features across all software teams
- Built an internal dashboard to monitor **all MakerBot properties** and services in real-time by utilizing Grafana, Prometheus, and custom Golang-based monitoring tools
- Dramatically improved user experience by implementing RESTful Node.js microservice to allow code-based authentication to several MakerBot 3D printers
- Implemented new software development practices across web development team to streamline feature development and release process

SRND | Full Stack Software Engineer **2014 – 2019**

- Decreased wait times at physical events with an average of **4,000 attendees** by developing an app using Android (Kotlin) and iOS (Swift) to allow attendees to check-in automatically
- Increased organizer happiness by creating tools to automate several recurring tasks such as emailing teachers, generating an event summary, etc.
- Connected thousands of attendees to mentors automatically by developing a Rails-based web service
- Allowed teams at physical events to showcase their projects online with a Rails-based web service
- Managed Azure virtual machines to host web services critical to business and event management

Academus | Full Stack Software Engineer **2018 – 2019**

- Designed and implemented system to handle up to **100,000 concurrent synchronization jobs** with third-party APIs by utilizing smart-syncing, vertical and horizontal scaling
- Developed a distributed, easily scalable back-end system to handle **thousands of concurrent users** at once by utilizing heavy Redis-based caching, load balancing, and more
- Developed Android client (Kotlin) for main product, implementing all core features from scratch

RECENT TALKS

Intro to Go - CodeDay Seattle Winter 2019

Docker: What is it and why do I keep hearing about it? - CodeLabs Virtual Coding Camp, July 2018

SELECTED PROJECTS

Neodeck - Rails + Socket.io application to create Cards Against Humanity-style card decks

schema.tl - AngularJS single page application to interactively view Telegram's Type-Language schema

Esyx - Imageboard viewer written entirely in SwiftUI, using APIs like Core Data and iCloud Sync

CompileBot - Telegram Messenger bot for compiling code inline in chats, built using Node.js and the Docker API

thermostatd - Golang daemon and HTTP server for controlling my HVAC unit with a Raspberry Pi

makerbot-rpc - Golang library to communicate with MakerBot printers with low-level API