

Tobias S. Davis

(402) 468-8404
tobias@davistobias.com
6723 S 94th Circle
Omaha, NE 68127

Guiding Principles:

- Always be learning new things.
- Find the right solution that fits the need.
- Make quality things that users care about.
- Work with other people who care about those same principles.

Professional Experience:

Besides working with other companies, I also build and maintain many open-source libraries which interact with AWS services (DynamoDB, SQS, S3, etc.), MongoDB (Atlas and community), and IMAP, as well as many libraries for content management and components for web app UI.

Ford Electriphi: March 2023-Present

Working as lead engineer with several large teams, including internationally across multiple languages and time zones, to build and maintain an Electric Vehicle (EV) charging management platform. This involves coordinating development, QA, and deployment of many systems ranging from system management, internal use, customer facing across many companies/branches, hardware and firmware, and hardware and software vendors. The services are primarily written in JavaScript, Go, and Python, and run either in an AWS environment as part of several serverless stacks, or as part of a larger Kubernetes cluster.

As a lead engineer, I regularly am involved with research and system design discussions and am regularly asked for assistance with deeper understanding of various parts of the architecture. I am also frequently asked to accomplish specific high-touch and critical-path tasks as I am known to be a dependable and knowledgeable member of the team.

Currently am the lead engineer responsible for design and implementation at the software/service level with a vendor providing credit card terminals installed on EV chargers. Responsible for coordinating testing and rollout with a dedicated hardware team and with several other inter-Ford teams (e.g. payment management at different scopes, security, and so on). As a full stack engineer, I was responsible for many different details, from making sure customer-facing UI represented the Ford brand well, to making sure generated reports would serve several branches and external parties.

While building these applications and services, some of the things I utilized regularly were:

- TypeScript, Python, Go, Node.js (18 and 20), Amazon Web Services (including SQS, Lambda, API Gateway), MongoDB, Kubernetes, Argo CD, Vault, Gitlab (including Pipelines), Docker, and many more.

Solo Software Architect: April 2021-March 2023

Built a custom web application for a client managing car shipments within the US. They were previously paying for a proprietary application which was unmaintained and difficult to work with, supplementing their workflow with many spreadsheets and personal notes, and no way to grow the company or optimize repetitive and costly workflows. Using my skills as a full-stack developer and my skills in business management, I designed an architecture that is affordable and resilient to this day, and a web UI that is simple and easy to work with.

Built a CMS for a non-profit client with a very large collection of writings: more than a dozen full-length books, and over 1,000 essay. Helped the client walk through a few architectural decisions, summarizing the technical aspects into business-actionable decisions. Used an iteratively designed architecture to end up with a very low latency, low cost, highly configurable content publication pipeline.

While building these applications and services, some of the things I utilized were:

- Browser Related: SvelteJS, Apex Charts, Bootstrap, JavaScript, Moment.js, AJV.
- Server Related: Modern NodeJS (20+), Amazon Web Services (including S3, SQS, SNS, SES, Lambda, API Gateway), Cloudflare (Workers, Pages, KV), MongoDB (Atlas and Data API), Algolia (Search), TypeSense, Digital Ocean, and many more.

From Now On (Omaha, NE): September 2016-April 2021

Worked with a small customer-focused team creating custom mobile apps for sport fans, providing live event information such as gameplay information, audio streams, and more.

As a full-stack developer, system architect, and dev-ops lead, I built and maintained the web UI of the content management system that the clients interfaced with, designed and maintained the backend architecture of the many services used in their system, and managed deployment and uptime for their services. Those services were primarily written in JavaScript and run in AWS environments as part of a “serverless” stack.

When I arrived at From Now One, I inherited a large and incomplete code refactoring and re-architecting project, which I took from an unusable state to a fully functioning, well tested, battle-hardened webapp. The web application and services were outdated and generally untested, but I introduced modern best-practices such as automated tests on all pull requests, code linting as part of those tests, and continuous deployment for the webapp. This reduced bugs and regressions significantly, and greatly reduced the time it took to develop features and fix bugs.

My most significant project was a large re-architecture of the backend services, taking the existing monolithic services and migrating them from Azure to AWS, while breaking the different services apart into modular microservices. The migration took over a year to complete, but it cut the backend costs in half and allowed a ~10x growth in paying clients.

While working at From Now On, some of the things I became very familiar with:

- Client Related: SvelteJS, Browserify, Bootstrap, Babel (ES6 to ES5 mostly), LESS, JS, HTML, CSS, XML, Moment.js (with timezone offsets), jQuery.
- Server Related: Node.js 4-10, Amazon Web Services (AWS, including EC2, S3, SQS, SNS, Kinesis, Lambda, API Gateway), JSON, MySQL, Redis, MongoDB, YAML.
- Development Tools: Eclipse, WebStorm (IntelliJ), bash, Node.js, npm, browserify, ES6 and ES5, minify/uglify for JS, Mustache (template language), Tape (test framework), Markdown to HTML transforms, Git, Github, Bash, Ruby, Homebrew.

eDataSource (New York, NY 10001): May 2012-September 2016

Designed and maintained data analysis tools and several public facing and business-to-business webapps. Oversaw several large software upgrades.

2012-2013

I helped develop software for analysis of email data, using aggregated data gathered from a panel of over 1 million active users.

I spent the first couple years primarily designing tools which analyzed emails, parsing financial data from third-party receipts that were frequently broken. The accuracy of the tools I designed was high enough that the tools and aggregated data were still used in predictive tools for financial investors several years later.

The generated aggregate data was stored in MySQL, in tables with many millions of rows. As part of the QA process, and as part of client driven requests, I was responsible for writing queries which were performant, even with very large tables and inadequate indexing.

2013-2016

Switched focus to full-stack work on a new webapp which presented email information to our clients, helping them determine effectiveness of email campaigns.

I was the lead for two big and painful software upgrades (Bootstrap 2.2 to 3.3, AngularJS 1.1 to 1.3). The upgrade process spanned more than a month and involved refactoring and changing hundreds of HTML and JS files, and tracking down dozens of difficult to find bugs.

Front-end lead of a two-person team designing a new feature which helped clients test email campaigns prior to sending them. The design had ~8 new screens, several new complex UI objects, and required the design of many new API endpoints.

2016

Worked with a team of 2-3 other developers on a redesign of one of our webapps, accessed by thousands of users per day. Due to our work we saw signup numbers increase and churn rate drop drastically. As part of the redesign, I designed and built the client and server side of the OAuth flow for Google, Yahoo, and Microsoft/Outlook, along with a complete test suite for it.

Toward the end I worked with a team of 2-4 in designing a new screen for our analytics webapp, which took aggregated data about a company and presented a view comparing that company's metrics with the metrics of other companies, to let our clients know how their email campaigns were performing relative to their competitors. The feedback from our clients was very positive.

Although I was not the sysadmin, since we were a small team I learned some sysadmin and devops skills. In particular, I became familiar with many AWS offerings—SQS queues handling many millions of messages per month, Aurora for terabyte-sized databases, auto-scaling EC2s for our webapps and backend services, and S3 and Glacier to store hundreds of millions of files.

While working at eDataSource, some things I became very familiar with:

- Client Related: AngularJS, Riot.js, Highcharts, Browserify, Bootstrap, Babel, PreCSS, PostCSS, LESS, JS, HTML, CSS, XML, Socket.io, Moment.js, jQuery, ACE.
- Server Related: Java 1.6-1.8, Amazon Web Services (AWS, including EC2, S3, SQS, SNS), Google Guava, GSON/JSON, MySQL, HyperSQL, Redis, JDBC, JAX-RS, Jetty, Jersey, Maven, Joda (date library), Log4J/SLF4J, OAuth, YAML, JMock, Spring Framework, Atmosphere (Java websocket implementation), Swagger.io, PhantomJS.
- Development Tools: IntelliJ, Eclipse, JRebel, bash, Node.js, npm, browserify, ES6 and ES5, minify/uglify for JS, Mustache (template language), Tape (test framework), BrowserStack, Markdown to HTML transforms, Git, Github, Bash, Ruby, Homebrew.

Prior to eDataSource:

My professional career began at eDataSource, but prior to that for ~2 years I was doing private contract work with WordPress and PHP.

For about a year, during college, I built custom WordPress templates and plugins for a paying client, and for some friends, *pro bono*.

Personal Projects:

My personal philosophy is to always be learning new things. Outside of work hours I build modules and websites, exploring new technology primarily in the JavaScript/npm world.

I've made and maintain a number of npm modules. A few that I am pretty happy with are:

- **imap-tools:** Small collection of tools making it easier to work with IMAP in JavaScript.
- **hex2words:** Given a hexadecimal value, it returns a list of PGP Words, used to easily verify data signatures are identical.
- **json-schema-to-markdown:** Turn a JSON schema (a JSON object used to validate other JSON objects) into a human-readable markdown file. It makes writing documentation a lot easier.
- **pipe-transform-cli:** A command line utility that lets you stream in one encoding and stream out another encoding. For example, hex in and base64 out.
- **ftp-core:** Although the project has been set aside, this was an attempt to create a pure JavaScript implementation of the FTP protocol. It was really fun to take the RFC specs and turn them into code. Sadly, I ran out of time to finish it.

Tech Stack for Personal Projects:

In addition to the buzzword listed earlier, I've created tools and products using these:

- Client-Side: SvelteJS, AngularJS, Ractive.js, Highcharts, RollupJS, Browserify, Bootstrap, JS (modern and historic), LESS, HTML, CSS, XML, Socket.io, Moment.js.
- Serve-Side: Express, Ecstatic, MySQL, OAuth, YAML.
- Development Tools: Sublime Text, bash, Node.js, npm, browserify, JS cli tools (minimist is nice), minify/uglify for JS, Mustache (template language), Tape / Ava / zora (test frameworks), Markdown to HTML transforms, Git, Github, Bash, Ruby, Homebrew.
- Operating Systems: Windows, Mac, Ubuntu, Debian.
- Other Things: SQLite, PHP, basic cryptography techniques (AES, RSA, SHA, SHA-2, PGP/GPG), Swagger.io, JSON Schema, I've contributed to ngbp (AngularJS boilerplate), Noddity, and have contributed to PhantomJS.