

David Zemon

(636) 395-0126
david@zemon.name

<https://david.zemon.name/>
<https://github.com/DavidZemon/>

801 Green Ridge Ln
St. Peters, MO 63376

Skills

Languages: C++ (98/11/14/17), C, Java (6/7/8/11), Python (2.6/2.7/3+), TypeScript, JavaScript 5/6, Bash
Frameworks: Spring (Boot, MVC, ORM, JDBC), Angular (v1.x, v2+), React v4, Bootstrap (3/4), Express
Software: Docker, CMake, Conan, GNU Make, JetBrains IDEs, Eclipse (Java, CDT), Git, SVN, Debian & RedHat derivatives, Ansible, JIRA, TeamCity, Jenkins, SonarQube, Artifactory
Databases: Oracle, MySQL, MongoDB, DynamoDB

Professional Experience

Sr. Consultant: Mutual of Omaha Insurance (Object Partners, Inc) **Aug 2021 – Present**

- Co-lead effort to spin up a developer portal based on Spotify's Backstage framework
Tools: Spotify Backstage, NodeJS, Express, React, Material UI, Lerna, Kubernetes, Docker, HashiCorp Vault, Nunjucks
- Implemented authorization logic for resources based on access to original resource in BitBucket
- Developed custom action for accepting and storing secure data in HashiCorp Vault as part of the software templating pipeline
- Tracked and kept pressure on key road blocks to maintain target delivery dates

Sr. Consultant: First Horizon Bank (Object Partners, Inc) **Mar 2021 – Jul 2021**

- Applied feature enhancements to D3 banking software
Tools: Spring Boot (MVC, Web, AOP, JPA), Java 8, Docker, React, TypeScript
- Developed web application for discovery and resolution of bank merger data
Tools: Spring Boot (MVC, Web, Data, JPA, Security), Java 11, React, TypeScript, LDAP, JWT

Sr. Consultant / Technical Lead: Chewy (Object Partners, Inc) **Jun 2020 – Feb 2021**

The Autoship team is responsible for accepting, maintaining, and processing recurring orders from the Chewy.com retail website (70% of Chewy's total revenue). A migration effort broke the Autoship code of Chewy's monolith and into microservices hosted in AWS.

Tools: NodeJS, NestJS, Lerna, Jest, DynamoDB, Dynamoose, Swagger

- Translated architect's vision into concrete and digestible work items
- Served as general problem solver for any developer encountering unforeseen issues
- Actively encouraged developers to seek opportunities for personal growth and leadership as team grew and time for problem solving became scarce
- Organized team of nine developers to write the REST service and underlying shared libraries

Sr. Software Engineer: AuthenticID **Jan 2020 – May 2020**

- Developed dynamic React-based client and Spring Boot-based application server with page content and UI theme changing based on logged-in user
Tech stack: Java 11, Spring Boot 2, NPM, React router 4, TypeScript 3, InversifyJS, Bootstrap 4, i18next
 - Authenticate browser clients via JSON Web Tokens and use mTLS for internal services
 - Deployed & supported the application in a fast-paced environment, moving from concept to production to maintenance & support requests in less than three months
- Developed and supported a JavaScript 5.1 application using RequireJS to showcase our WebSDK product and act as a framework-independent example project and debugging tool for customers
 - Leveraged JSDoc to provide TypeScript levels of editor support in modern IDEs

Sr. Application Developer: Red Lion Controls, Inc.**Oct 2017 – Dec 2019**

- Lead designer and developer of high security module (HSM) to ensure Red Lion software only runs on genuine Red Lion hardware
Tech stack: GCC 7 / C++17, Google GTest & GMock, Valgrind, Conan, CMake 3, OpenSSL
 - First foray into C++17; Successful and extensive use of OOP/TDD practices, such as DI and mocking
 - Designed secure process for signing the HSM chips, despite an unavoidable physical attack vector
 - Deep dive into security/encryption theory in order to make use of the OpenSSL C library and ECDSA
- Responsible for SixView Manager, a NodeJS-based web application for administering a collection of routers. SVM managed firmware updates, custom package installation, health checks, and IAM
Tech stack: Docker, NodeJS 0.10 & 8, NPM, Express, ECMAScript 6, JavaScript 5.1, RequireJS, MongoDB 2.6 & 3.4, KnockoutJS, Bootstrap 3
 - Completed feature requests, bug fixes, and security enhancements
 - Upgraded NodeJS runtime environment from 0.10 to 8 and MongoDB from 2.6 to 3.4, running thorough automated and manual regression tests as part of each upgrade.
 - Introduced original SVM tech lead to newer JavaScript (ES6) features and successfully refactored legacy code during feature enhancements and bug fixes
 - Developed Python 3 / Selenium-based automated regression test suite
 - Produced and supported multiple SVM releases, including end-user support requests
 - Replaced RPM as customer-deliverable package with a Docker image, allowing Red Lion to expand its support to more than RHEL and CentOS 6
- Enhanced and maintained GNU Make-based build system for assembling embedded Linux images
- Lead adoption of the CMake build system and Conan dependency manager across the team for simpler cross-compiling and faster feedback
- Introduced Java-inspired testing methodologies (GTest and GMock) and OOD principles to C++ projects
- Administered a handful of developer tools (CI, etc) in AWS (EC2 & S3) with Ansible and Docker

Sr. Application Developer: Union Pacific Railroad, Inc.**Jan 2014 – Sep 2017**

- *Sr. Application Developer • CADX Team • June 2015 – Sep 2017*
 - Designed and implemented change management application, meeting the requirements of developers, testers, and release managers (and supported both new and legacy modes of operation)
Tech stack: Python 3, Java 8, Maven, Spring 4 (MVC, JDBC), JUnit 4, Mockito, Pitest, NPM, ECMAScript 6, TypeScript 2, AngularJS 1.6, HTTP WebSocket
 - Migrated from Python to Java as application size increased and dynamic typing became unwieldy
 - Extracted core logic from CLI application into a library that could be shared between the CLI and newly developed RESTful service
 - Developed AngularJS application as a web-enabled alternative to CLI, complete with real-time stream of deployment logs
 - Utilized a common interface for both the file-based JSON storage on a local developer's computer and database storage of environment configurations in the RESTful service
 - Decoupled system architecture from Linux user IDs by redesigning pieces of the CADX core architecture to support system startup based on CLI arguments and environment variables
 - Added support for configurable installation directory
 - Created and maintained custom RPMs for runtime and compile-time dependencies
- *Volunteer Instructor • IT Training • Jan 2015 – Sep 2017*
 - Identified need for and spear-headed creation of Spring Framework course
 - Formed team of volunteers, designed, and taught the 32-hour Spring Framework curriculum
 - Invited to co-teach two-day Intro to Java course after 12 months with UP
- *Associate Application Developer • Finance Team • Jan 2014 – May 2015*
 - Designed reusable batch job framework, utilizing multi-module Maven
Tech stack: Maven, JUnit, Mockito, Spring 4 (Core, ORM), Hibernate, Oracle
 - Developed and deployed multiple batch jobs using the framework
 - Supported co-workers' development efforts as the framework was used by more team members

- Developed RESTful web app for administering user permissions
Tech stack: Java 7, Maven, Spring 4 (MVC, ORM), Hibernate, JUnit 4, Mockito, NPM, Grunt, JavaScript 5.1, ECMAScript 6, AngularJS 1.5
- Acted as primary mentor for intern projects

Open Source Experience

Lead Developer: Jumpstart - <https://github.com/DavidZemon/jumpstart> **Nov 2018 – Present**

- Leveraged Python's Django templating engine to generate a C/C++ project with modern project tooling
- Verified compatibility of disjunct tools such that an engineer is presented with a single, cohesive, and modern development experience
- Published instructions and easy-to-use Docker image for end-users to generate custom projects
- Tech stack: CMake 3, Conan 1, GTest, GMock, Valgrind, gcov, clang-tidy, cppcheck, CLion, gdb, Python 3, Django, Docker

Lead Developer: PropWare - <https://david.zemon.name/PropWare/> **Apr 2013 – Present**

- Designed hardware abstraction layer, including ground-up libraries for UART, SPI, SD, and FAT, for Parallax Propeller in C++
 - Weighed legibility, runtime performance, and code size during design phase
 - Designed with future expandability in mind
 - Debug serial routines with oscilloscope
- Implemented CMake build system, providing balance between flexibility and Arduino-like simplicity
- Integrated existing HALs from Parallax and other community members with source code compatibility
- Documented all steps, including installation, setup, examples, and APIs
- Prioritized features based on user requests
- Tech stack: CMake 3, GNU Make, PropGCC (GCC 4.6), C++11, C99, PASM (Assembly), Logic 8 scope

Community

Webmaster: St. Louis Woodworkers Guild, Inc. **Oct 2018 – Present**

- Rebuilt website; Integrated membership list into online database; Modernize IT infrastructure

Education

Missouri University of Science and Technology
B.S. in Computer Engineering – Rolla, Missouri

Dec 2013