JIM MYHRBERG

SOFTWARE ENGINEERING MERCENARY

I'm a senior software engineer with over 12 years of professional experience, with a focus on using Ruby to build back-end systems for distributed high-performance queue processing, REST APIs, microservices, and fully featured Ruby on Rails websites.

I am a quick learner, making me excellent at researching and implementing new technologies, protocols and standards.

I always:

- Create a full and comprehensive test suite for code I write.
- Follow best practices relevant to the work at hand.
- Leave existing code I touch in a better state than I found it.
- Deal with tech debt as early as possible based on project constraints.
- Do things properly, as much as project constraints allow.

OPEN-SOURCE

- rbheap (2018)
  A tool to help with tracking down memory leaks in Ruby.
- Git Common-Flow (2017)
  A specification born after one too many GitFlow vs GitHub Flow arguments.
- stub.sh (2014)
  Stubbing/faking library for Bash, making it easier to write tests for Bash scripts.
- tmux-themepack (2013)
  A popular set of Tmux themes.
- Tmuxifier (2012)
  A powerful Tmux window and session management tool.
- Git Aware Prompt (2009)
  Have bash display your current Git branch and dirty state.
- Redistat (2009)
  A easy to use, Redis-backed statistics collection and querying library written in Ruby.
- parseCSV (2006)
  Was the first fully featured CSV library available for PHP. Even going so far as to auto-discover delimiter characters and more.

jimeh.me jimeh jimeh

contact@jimeh.me jim@heartb.it

SKILLS

- Ruby
- Ruby on Rails
- Go
- AMQP
- RabbitMQ
- MySQL
- PostgreSQL
- Redis
- REST API
- Microservices
- TDD
- BDD
- RSpec
- Cucumber
- Linux
- Docker
- Kubernetes
- OpenShift
- Ansible
- ...

EXPERIENCE / PROJECTS

SSO Integration (Jan 2019 - Mar 2020)

ruby rails rest amqp openshift prometheus

The client (UKCloud) is rolling out Keycloak as a Single Sign-On solution across all its systems and vendor products, so customers no longer need to sign in to each system and vendor product separately.

Main Achievements:

- Built AMQP workers to process and sync data between SSO (Keycloak) and various vendor products.
- Created a formal set of message schema specifications for all communication between internal systems and vendor products.
- Established best practices and standardized metrics that all things should report to Prometheus.

Protective Monitoring (May 2017 - Sep 2018)

ruby rails rest amqp microservices oauth 2.0

As a cloud provider the client (UKCloud) protects against and scans for all manner of security related issues and incidents. They wanted to automatically expose relevant data directly to customers through a dashboard and API.

Main Achievements:

- Built a customer facing dashboard to expose potential security threats.
- Designed and built an internal microservice REST API to power the dashboard.
- Built a AMQP worker which populated the internal API with data.

Service-Oriented CMS (Jan 2016 - Nov 2016)

ruby rails rest redis microservices

To improve performance and scalability of their platform, the client (Which?) wanted to move away from it's Rails-based monolith to a suite of microservice REST APIs as the source for all it's content.

Main Achievements:

- Designed and built a number of microservice REST APIs that made up the client's new content management system.
- Worked on various data migration components that pulled data out of the existing monolith.

Spread Betting Platform (May 2012 - Jul 2013)

ruby rails rest mysql backbone.js

The client (Fitzdares) needed a in-house web-based spread betting platform built to their specific needs, as they had been unable to find a commercial product that fit their needs.

Main Achievements:

- Built a number of server and client side components for a manual bet settling and payout system.
- Built various interactive front-end user interfaces using Backbone.js.

Patient Dashboard (Sep 2018 - Jan 2019)

ruby rails postgresql

Managing the stages of treatment for each patient is a very complex process, and the client (Healios) needed a dashboard that was integrated into their online portal to replace the shared Excel spreadsheet they were using previously.

Main Achievements:

- Added comprehensive tests, improved performance, and performed a major refactor of the existing unfinished and untested code-base for the internal patient overview dashboard.
- Successfully launched said dashboard after implementing all missing features.

Queuing System Improvements (Nov 2016 - May 2017)

ruby amqp rabbitmq

The client (UKCloud) brought me in to improve the reliability, testability, and maintainability of a number of legacy AMQP workers written in Ruby.

Main Achievements:

- Improved overall code quality of workers by adding comprehensive test suites, and performing major refactors of the code bases.
- Improved observability of workers by adding extended logging and error reporting to the code bases.

Fixed Odds Betting Platform (Jul 2013 - Jan 2016)

ruby rails rest mysql pdf

The client (Fitzdares) needed to replace the DOS-based Fixed Odds bet management platform they had, with a modern web-based solution built on top of their existing in-house Spread Betting platform.

Main Achievements:

- Designed and built real-time and near real-time data import systems to handle large amounts of sport and betting data from BetRadar and other sources.
- Implemented critical business logic and rules for automatic bet settling and payout to customers.
- Used prawn to generate elegant financial PDF statements for customers.

Microservice REST APIs (Aug 2011 - Apr 2012)

ruby rest microservices

My employer (Global Personals) needed to expand into the mobile space, and they decided to build a new suite of high-performance microservice REST APIs to power it, as their legacy ColdFusion-based monolith was not suitable for the task.

Main Achievements:

- Designed and built various internal microservice REST APIs on top of the company's legacy platform.
- Designed and built a aggregator API which sat on top of all microservice APIs, exposing the platform as a single API.