

RYAN J. PETERS

8108 Equestrian Drive, Severn, MD 21144 | (410) 218-3673 | ryan@binarydad.com

Professional Summary

- 17 years of professional .NET web engineering experience
- Self-driven to improve application performance and maintainability using tried-and-true best patterns and practices
- Works well with a team to create a suitable and pragmatic solution in both new and existing code bases
- Able to articulate architectural and implementation details clearly and concisely to customers

Work Experience

ClearOne Advantage – June 2017 – Current, Baltimore, MD
Technical Lead, Staff Application Engineer

- Worked closely with our head of DevOps to break ground on moving our critical web services tier from on-premise IIS to **AWS EC2/ECS** via use of **Docker** containers, thus allowing for ease of scalability, and ensuring consistent and stable releases.
- Co-hosted round-table discussions with our DevOps team regarding which services to consider for **AWS serverless implementation (Lambda) or Docker (ECS)**, weighing pros and cons of each in regard to deployment, scalability, cold-start delays, and ease of integration of our code base.
- Taught members of our development team how to migrate applications from .NET Framework to .NET Core 2/3 or .NET 6, enabling support for **Docker**, in preparation of **AWS ECS hosting**. This effort laid the groundwork for future migration efforts by members of the team.
- Consistently pushed for efforts to improve efficiency, performance, and maintainability of primary applications:
 - Worked with and lead our development and DevOps team on how to migrate critical web functionality from single monolith processes to a distributed microservice architecture, thus allowing for quick, efficient deployments with less errors due to loose coupling of functionality. This resulted in a vast reduction in deployment errors due to less changes to the production environment, as well as less risk due to the isolated nature of the microservice architecture.
 - Architected a "service wrapper" application layer to facilitate and abstract-away protocol-specific web service implementations, which resulted in less time spent by developers writing code and less fragmentation in code consuming these services. Group discussions were held outlining their benefits and explaining proper integration into the code.
 - Implemented and taught caching best practices as well as application setting and connection string management via a robust set of utility libraries, streamlining repetitive common tasks performed by developers.
 - Instructed other developers on how to follow DI patterns through refactoring, allowing for clear separation-of-concerns and facilitated use of mock dependencies for creation of isolated unit tests, as well as IoC frameworks such as StructureMap to handle dependency registration and injection. This enabled developers to make changes to their code without risk of unintended side effects often associated with tight code coupling.
 - Streamlined external dependency upgrades by moving shared project libraries to versioned NuGet packages for faster implementation and enabling direct insight into feature sets provided by package version.
 - Architected a strong foundation and conversion path for migrating legacy WCF web services to Web API, incorporating core API features that enabled for rapid development and fostered improved testing practices on the new platform.
 - Improved Google PageRank and GTmetrix scores on landing pages by working to reduce overall load and first-paint time by incorporating asset minification, dynamic loading of scripts, and use of critical CSS and JavaScript, thus improving page rank.
- During our Salesforce CRM migration, worked closely with their development team:
 - Designed and created APIs to handle incoming lead traffic from multiple campaigns and enabled bulk migrations into our legacy CRM, eliminating previous manual processes.
 - Engineered a comprehensive Salesforce API .NET wrapper to handle CRUD operations against Salesforce, including the ability to convert .NET expression trees used for querying into a dynamic SOQL command. This feature enabled the developer to query data without having to know Salesforce's SOQL syntax.
 - Participated in training sessions for Salesforce Apex and Lightning Component development, using this knowledge to mentor other developers on how to create basic interface components and debug applications in Visual Studio Code.
- Designed, created, and maintained our credit report services to handle failover between multiple credit bureaus, allowing for 100% uptime during our lead intake process on both landing pages and over the phone.
- Wrote and refactored code as asynchronous where possible, allowing long I/O operations to natively release threads and thus not starve the application.
- Created our Creditor Portal as a distributed system to manage bulk settlement operations, including OCR of bulk offer letters in varying image and document formats, allowing for quick web-based settlement updates from creditors. This much-needed application empowered creditors to bulk-disposition settlements and reduced time spent by our negotiators creating offers on our clients' behalf.
- Lead "TED Talk" style forums discussing best practices and patterns to apply throughout our code base. Topics ranged from new application

functionality I've developed to discussion of better ways to improve efficiency in both performance and development. I frequently held 1-on-1 help and knowledge sessions with other developers do go over ways to improve their writing abilities, as well as discuss new concepts and when to consider applying them. Feedback was extremely positive and resulted in more consistent practices and coding styles from other members of our development team.

AscendOne Corporation - February 2010 – December 2016, Baltimore, Maryland
Senior Web Application Developer

Nava Health and Vitality

- Designed and created a feature-rich CRM for Nava, which was responsible for client, membership, and lab result management, as well as the processing of medical protocols relating to hormone dosing and evaluating a client's Custom Vitality Plan (CVP). The result was a recommendation roadmap for a client based on medical history, lifestyle, and latest lab results. This web application was created using ASP.NET/C# and the MVC 5 framework, and ran on both desktops and iPads used by practitioners in the centers working real-time with clients.
- Developed Nava's client-facing website, navacenter.com using Sitefinity CMS. The website was created using responsive design for the purpose of maintaining a more consistent UI across desktop, tablet, and mobile devices.
- Worked with another developer to create a completely custom marketing infrastructure, integrating with ExactTarget (via SOAP API), to allow the business to send automated client emails, such as welcome messages, appointment confirmations, reminders, payment received notifications, and issues with payments.
- Ensured optimal performance by analyzing the amount of page asset downloads, use of caching, and file size. Image sprites were created using Photoshop.

CareOne Debt Management

- Maintained a fleet of client-facing and internal web applications and services to support our customer portal and agent CRM.
- Worked with developers to refactor and optimize the DAL, used by the CareOne websites, by creating a customized runtime ORM. This shrunk the DAL codebase by roughly 70% by eliminating redundancies and promoting code reuse.
- Developed CareOne's Android and iPhone mobile application, created using PhoneGap, which utilized pure HTML/CSS/JavaScript and incorporated the Knockout.js framework for view model binding. The application allowed clients of CareOne to retrieve their balances, manage settlement offers, creditor proposals, modify payments, and view a graph of their debt pay-off.
- Ensured that client-facing web applications were optimized for performance by use of image/CSS/JavaScript compression and caching, as well facilitating the use of session- and application-level caching by using extension methods when retrieving data from remote storage.

Personal Development

For over a decade, I've enjoyed creating and hosting my own services and .NET projects, as well as hosting other services as email, remote document access, blogs, wikis, chat servers, databases, GIT repositories, and container platforms on custom-built web servers typically running Hyper-V. For the last few years, I've leveraged **Docker** more often due to the ease of deployment and upgradability. After taking a deep-dive into learning Linux, it resulted in allowing me to migrate several personal applications from Microsoft SQL Server and .NET Framework to MariaDB/MySQL and .NET 6 running on Ubuntu.

The ecosystem is an evolving challenge, but an enjoyable one, as it allows me to try-out and learn DevOps and networking operations such as port forwarding, reverse proxy, SSL/TLS certificate management, containerization, shared password management, DNS, DHCP, VPN, file backup, and other networking and DevOps concepts.

While most applications are hosted locally, at home, **some are hosted on Azure as a mix of Ubuntu virtual machines and Docker App Services.**

I'm part of several local .NET Meetup groups and regularly take part in online developer sessions and courses.

Hobbies and Interests

When I'm away from the keyboard, I enjoy cooking, mountain biking, drumming, recording, and camping. I enjoy watching application development, DevOps, and networking channels on YouTube and applying new concepts in my home virtual environment.

Education

University of Maryland, Baltimore County – Fall 2001 – June 2005

Major: Computer Science

Degree: Bachelors of Science