Michael J. Ryan

https://tracker1.dev/
resume@tracker1.dev
(480) 270-4509

Phoenix, AZ, USA
Remote

I have been working as a full-stack software developer, architect and in leadership roles specializing in web-facing applications in environments from dozens to millions of simultaneous users for nearly 3 decades.

I would enjoy a role with the flexibility to design and implement great solutions that solve real issues. The past few years has been more focused on solutions utilizing React, Redux and Material UI (mui) using TypeScript on the front-end. Back-end services have been a combination of C# (ASP.Net/MVC) as well as Node.js. Database development has been a combination of SQL, NoSQL and search engines.

WORK EXPERIENCE

Personal (Independent Contract Work) — *Lead Software Developer*

DECEMBER 2023 – CURRENT

Kiosk Platform – Working with a local startup on a hardware kiosk platform for used in retail environments. Front end work is written in TypeScript with a combination of React and Next.js with Next.js, Express and Apollo GraphQL backend development.

Chrome Extension – Developed a chrome extension utilizing Deno, TypeScript and built with ESBuild. This extension records the current tab for capture and upload stream to a backend application for purpose of reviewing content development.

Social Network Platform – Proof of concept on a stealth social media platform. Front-end using Cloudflare Pages with HTMX. using a Hono based application with a Turso backend.

Yoh (First American Title) — *Lead Software Developer*

MAY 2023 – AUGUST 2023

Fluent Title and Platform - Lead ongoing development, refactoring and enhancements to title and platform team for a multi-agency title process management and workflow application.

- AWS DynamoDB was the primary data source(s) used. Terraform was used for deployments to AWS and Github was used for source control.
- Node.js was used for back-end services that were deployed to AWS Lambda functions, and communication to the client was via Apollo GraphQL connecting to back-end services also written in Node.js and deployed to Lambda functions.
- React was used with TypeScript on the front-end with functional components and hooks for lifecycle events. For data interaction, commu-

PRIMARY SKILLS

- Architect product solutions using a range of best-fit technologies.
- Full-Stack developer with experience across a broad range of client, server and database systems.
- Extensive experience with web-focused application development and JavaScript.

LANGUAGES

- JavaScript (25 years)
- TypeScript (10 years)
- C# (20 years)
- Rust (Learning)
- SQL (25 years)

Not an exhaustive list.

SKILLS (word soup)

TypeScript, Node.js, Express, ASP.Net, Material Design, Material UI (mui), Microsoft SQL Server, PostgreSQL, Azure, Amazon Web Services (AWS), CI/CD, Github Workers, C#, Rust, Go

nications and caching the Apollo GraphQL client library was used. For components themselves, the MUI (Material Design) component library was used as a base layer against an internally built micro-ui platform library.

Vaco (HCA/HealthTrust) — *Senior Software Developer*

OCTOBER 2022 – APRIL 2023

Marketplace – Lead the UI refinement, refactor and established ongoing best practices for an application for multiple client locations to access office and medical supplies to simplify their procurement process. The role was a combination of hands-on, and mentoring other developers. *Do Not Contact.*

- Azure hosting was used for platform services with Azure's API Gateway, CDN in addition to hosted Microsoft SQL Server, Azure Cognitive Search, Azure Functions and Azure Storage. CI/CD via Azure DevOps and Azure infrastructure via Terraform. Github was used for source control.
- C# (7/Core) was used on the back-end with ASP.Net WebAPI/MVC communication with the Microsoft SQL database was done using the ADO.Net client library and the Dapper object mapper library. NSwag was used to generate API documentation, swagger-ui interfaces and client api tooling. Razor views were used for some internally facing tools.
- React was used with TypeScript on the front-end with functional components and Redux (hooks) for life-cycle events. For data interaction, communications a generated client library from the API and NSwag tooling was used.. For components themselves, the MUI (Material Design) component library was used directly, this was a migration away from components based on Bootstrap that were used before to better align with corporate direction.

ServiceTitan — *Staff Engineer II*

OCTOBER 2021 – SEPTEMBER 2022

Application Platform API - Leadership, principal development and design of systems and architecture related to the platforms used to expose and interact with customer driven API services.

- React was used with TypeScript on some front-end applications with functional components and hooks for life-cycle events MobX was used for state management. For data interaction, direct communication with WebAPI Fetch was used. For components themselves, the MUI (Material Design) component library was used.
- C# (6/Core) and ASP.Net MVC and Razor was used for other internal

tooling via partially generated templates.

- C# (5/Core) and ASP.Net MVC with WebAPI was used with Dapper for externally facing API used for customer tooling. Some services integrated to the Microsoft SQL Server database(s) others would use PostgreSQL for department tooling.
- Other services include Azure, Kubernetes, Helm, Kafka and Identity Server

Runbeck Election Services — *Senior Software Developer*

JUNE 2018 – OCTOBER 2021

Vocem – Architect, design and principal front-end development of this web application used for scanning and validation of petitions used by several states and local jurisdictions. This application's UI/UX was commented during a meeting with the Washington Secretary of State, "that looks sexy," which is high praise for an application in the government space and I took as a great compliment personally having worked long and hard to be able to present scanned documents as well as the markup overlays and interface features.

- Azure DevOps was used for source control, CI/CD work flows and managed deployments.
- Microsoft SQL Server was used for the main database and storage interfaces for the application. This involved heavy usage of Stored Procedures, User Defined Functions and the bulk of the application process logic on the backend was inside the database itself.
- C# (4.x) with ASP.Net MVC and WebAPI was used for the back-end API.
- React was used with JavaScript/Typescript and functional components using Redux with Fetch API to call to the back-end. MUI (Material UI, Material Design) was used for the bulk of web component base. This was started with class based components and higher order components(HOC) in 2018 but migrated to functional components and hooks as they became available through 2019.

Auth App – Architect, design and principal development of this authentication and role based access control application used by applications built at Runbeck for simpler authentication where a more advanced authentication platform (Azure AD, Auth0, etc) wasn't already in place. This was used for internal application deployments as well as a few states and local jurisdictions. It was built with defaults to meet or exceed NIST guidelines with options to match specific needs as overrides. This needed to be able to deploy to embedded applications on hardware as well as state/local deployments and cloud usage.

- SQLite was used as an embedded key/value store. Adapters were

also written to be able to (re)use Microsoft SQL Server or PostgreSQL server as alternatives.

- C# (.Net Core 3) was used during the development, including a difficult migration from Core 2 to Core 3 during development which was required for ARM based deployments. ASP.Net MVC with WebAPI was used here. The use of a KV interface allowed for layered security for an optional encryption layer used with SQLite at rest.
- React was used with functional components using Redux (hooks) with Fetch API to call to the back-end. MUI (previously Material UI, Material Design) was used for the bulk of web component base. This was started with class based components and higher order components(HOC) in 2018 but migrated to functional components and hooks as they became available through 2019.

  Authentication Adapters – Wrote authentication adapters for role based access controls (RBAC).

- C# (.Net Core 3) was used with ASP.Net MVC for various applications to integrate withAzure AD, Auth0, and Okta.

Vocem Online – Architect the first petition platform for online ballot endorsement used in the US for ballot endorsement. This was built as a pilot application and proof of concept for the City of Boulder in Colorado.

- The application and related services were deployed with Terraform on Azure Kubernetes Service managed with Rancher with source control and CI/CD workflows under Azure DevOps.
- PostgreSQL was used as the primary database store for this application. It was chosen as it is more portable, with relatively easy scaling options such as CockroachDB as an alternative deployment option while not being too alien to use vs MS-SQL which most employees and developers in the organization were already familiar with. It was deployed using Azure hosting.
- Node.js was used on the back-end with Koa as the applicaiton library in place.
- React was used with functional components using Redux (hooks) with Fetch API to call to the back-end. MUI (previously Material UI, Material Design) was used for the bulk of web component base.

Duo – worked on Auth App integration with this simple mail sorter with controller using an ARM platform.

- C# (.Net Core 3) for embedded Blazor based application interface

with the Auth App mentioned above.

Sentio – worked on designing an architecture for a new software solution to manage the ballot on demand printers used by several counties.

- The legacy solution was using C# with .Net Framework 4.x using a WinForms based UI and Microsoft SQL Express on Windows.
- The new solution would still be on Windows, but separate services from a user accessible embedded web based application.

Berkadia — *Senior Software Developer*

OCTOBER 2017 - MAY 2018

Worked as a Senior Software Developer. Was also a member of the Platform Architecture Team.

DesignView is an application to create InDesign based marketing materials leveraging design concepts in order to facilitate more rapid design creation.

- DesignView UI - Angular 4, Redux, Webpack 4, Babel 7
- DesignView API - Docker, Node.js, Koa, MySQL
- InDesign Server - Adobe ExtendScript ExtendScript (ES~3), Webpack, Babel, Developed composite build system for creating ExtendScript output while writing modern JS.
- CEP Extension for Indesign - React, Adobe ExtendScript, Webpack, Babel

NOTE: Phoenix Development Team Was Disbanded

EmailAge — *Senior Software Developer*

JANUARY 2017 - AUGUST 2017

UI/UX development and prototypes. .Net application support. ETL pipelines for AWS DynamoDB and Lambda. Other technology used includes React, Redux, Material UI, JavaScript, Node, Express, C#, SQL Server, Cassandra, Docker, Elastic Beanstalk (Docker), DynamoDB.

Brooksource — *Senior Architect*

NOVEMBER 2016 - DECEMBER 2016

Designed React UI projects (React, Material UI, Redux, JavaScript) with a Node/Express and Oracle back-end. *Project scope reduced and team dismissed at end of year.*

IntraEdge (American Express) — *Senior Developer*

SEPTEMBER 2015 - OCTOBER 2016

UI/UX Development for projects focusing on customer acquisition and email

marketing campaign management. Technologies: JavaScript (custom embedded, focused on light payload), Angular, React, Node, JavaScript, MongoDB.

Inxsol — *Senior Developer*

MAY 2015 – AUGUST 2015

Implementation refresh and polish for Command Plan, first responder training simulator. Technologies include Flash/Flex front-end with ActionScript, Administrative UI (JavaScript, jQuery, Bootstrap), C# (ASP.Net MVC) Back-end and some Node development, this including text-to-speech integration as well as a custom simulation server-side event loop.

Collector Car Network — *Software Architect*

JULY 2014 - MAY 2015

Design, collaborate and develop public-facing and internal applications for the ClassicCars.com website. Technologies include React, Material Design, Flux, SQL Server, RabbitMQ, Cassandra, Redux, ElasticSearch, Azure

GoDaddy — *Software Developer III*

AUGUST 2013 - JULY 2014

Worked with the Website Builder product team. Migration from ASP.Net MVC back-end with MS SQL Server to a Node backend with Redis and Cassandra. Front-end development using custom internal controls using jQuery.

EDUCATION

While I don't have a formal education past High School, I have spent my entire adult life studying, experimenting, researching and otherwise working on my knowledge and skills roughly 15 hours a week. I've spent nearly 3 decades developing software solutions. This includes actual course development for college use as well as aerospace engineering simulations, courses and tools.

SKILLS (more word soup)

TypeScript, JavaScript, ActionScript, ExtendScript, C#, .Net, .Net Core, Rust, Go, Golang, Adobe CEP, React, Redux, Material UI, ExtJS, AngularJS, Angular, jQuery, Bootstrap, HTML5, CSS3, JSON, Material Design, REST, RPC, Web Services, NginX, Node.js, Express, Koa, Visual Basic .Net, ASP.Net WebForms, ASP.Net MVC, Classic ASP, Windows Communication Foundation (WCF), Cloudflare Pages, Cloudflare Workers, RabbitMQ, Cassandra, PostgreSQL, ElasticSearch, Azure Storage Tables, Azure Blob Storage, Azure Cognitive Search, AWS DynamoDB, Redis, SQLite, MySQL, Microsoft SQL Server, Stored Procedures, Triggers, Docker, Containerization, Kubernetes, Helm, Amazon Web Services (AWS), AWS Lambda, AWS Elastic

Beanstalk, AWS Simple Storage Service (S3), Azure, Azure DevOps, Github, Github Workers, CI/CD, Windows, Linux, MacOS