

# Nitin Vericherla

I've worked over the last 15+ years as a developer, team lead, and solutions architect. I spent this time delivering robust solutions to clients, and mentoring large development teams. In most of my roles, I am actively involved in formulating best practices, hiring the right people, and leveling up developers.

## My experience

- |                  |   |                               |
|------------------|---|-------------------------------|
| Mar 21 - Present | <b>Staff Software Developer</b>   | <a href="#">Intuit Canada</a> |
|                  | I worked on the data-in and ui-platform teams, supporting TurboTax. The teams were responsible for the import and upload of tax documents, as well as the main UI of TurboTax online.   |                               |
| Nov 19 - Dec 20  | <b>Angular Architect</b>  | <a href="#">Contract</a>      |
|                  | <ul style="list-style-type: none"><li>• I helped large organizations leverage Nx by moving them towards monorepo and trunk-based development; and improving the developer experience and CI processes.</li><li>• I worked with developers 1:1 in order to level up their skills in state management, testing, and architectural design.</li></ul>   |                               |
| Mar 18 - Nov 19  | <b>Angular Architect</b>  | <a href="#">Nrwl.io</a>       |
|                  | <p>I worked as an Angular Architect at the consultancy behind Nx, which is a very popular framework for managing a monorepo within organizations. I worked with large enterprise clients to level up their development teams and to provide guidance on monorepos, state management, and complex Angular applications.</p> <ul style="list-style-type: none"><li>• I used the lessons learned from these clients to publish a book, "Enterprise Angular Monorepo Patterns," available on Gumroad or on Nrwl Connect.</li><li>• I provided guidance to several large clients on moving to trunk-based development in a monorepo: working out ownership structure as well as processes; moving code personally and training their developers.</li><li>• I managed 4 Nrwl developers to help them with their professional growth.</li><li>• I was responsible for driving the entire hiring process for the organization. It was an interesting challenge due to the high bar for candidates.</li><li>• I implemented a version of the 360-feedback system: communicating feedback about the company back to the founders, providing concrete feedback to developers from their peers and from the manager, and it helped to surface things that developers were going out of their way to do that would otherwise be invisible to the organization.</li></ul> |                               |

Sep 17 - Mar 18	<b>Solutions Architect</b>	<a href="#">Rangle.io</a>
	My second time around at Rangle as a Solutions Architect found me working with a client and Rangle team on two large initiatives: incorporating a headless CMS for creating fully dynamic pages, and exploring the possibility of transitioning to a mono-repo.	
Jan 17 - Sep 17	<b>Solutions Designer</b>	<a href="#">TD Wealth</a>
	I was the overall Solutions Designer on a large project to allow the bank to offer <b>targeted mortgage rates for customers</b> .	
	<ul style="list-style-type: none"> <li>• I led a team of 10 across the stack (front-end through to mainframes).</li> <li>• I was split 30-70 hands-on in delivering the front-end and nodejs API; and architecture.</li> <li>• The project represented a significant stream for the bank.</li> <li>• This project dealt with very complex integrations between large existing systems.</li> </ul>	
Nov 15 - Oct 16	<b>Application Development Lead</b>	<a href="#">Rogers Communications Inc.</a>
	I was the team lead on two large projects: the Fido dashboard for price plans and the Rogers.com Search and Support pages. I also regularly met with other team leads and architects to go over best practices and guidelines. Additionally, I interviewed candidates. This was a hands-on role, with a 50-50 split between hands-on delivery and architecture planning.	
Mar 14 - Nov 15	<b>Sr. javascript developer</b>	<a href="#">Rangle.io</a>
	As an early senior developer, I led several large projects from inception to delivery. I worked with node.js, Angular, Ionic and nwjs. The projects went through rigorous external security audits and code reviews and have always been at par or above with generally accepted best practices.	
	I also mentored 4 colleagues in their professional growth, and conducted their 360 reviews every quarter.	
Mar 13 - Mar 14	<b>Web/Mobile developer</b>	<a href="#">Koda Media Group</a>
	I led the development of a real-time platform for merchandise sales at concerts and events. The stack featured Angular, node.js, express and Neo4j.	
Sep 12 - Feb 13	<b>Web developer</b>	<a href="#">Brushfire North Inc.</a>
	I worked on full-stack development projects with .Net MVC and web forms architectures for ScotiaBank, kiosk applications and Audit Dashboard.	

May 07 - Sep 12	<b>Lead front-end developer</b>	<a href="#">Damashek Consulting Ltd.</a>
	I was the lead developer at a design agency focusing on sites for a sizeable SharePoint deployment. We were creating single-page applications with pure javascript before there were frameworks available for this.	
Dec 05 - Apr 07	<b>Web developer</b>	<a href="#">Glenbriar Technologies Ltd.</a>
	I was the lead developer working on B2B projects. The stack featured .Net and pure javascript on the front-end. I was heavily using AJAX to build single-page applications before it became widespread.	

## Projects of interest

Dec 18	<b>Published "Enterprise Angular Monorepo patterns"</b>	<a href="#">Nrwl.io</a>
	This book guides teams using monorepos and the Angular CLI. It contains detailed reasoning and patterns to use with code organization, team interactions (git strategies, versioning, making large-scale changes, and other challenges), and intelligent CI/CD techniques. It features case studies and lessons learned from consulting to large enterprises.	
	<i>angular, nx</i>	
Sep 17 - Dec 17	<b>Headless CMS-driven Angular application</b>	<a href="#">Rangle.io</a>
	This project was my second experience working with a headless CMS. Some tricky challenges are that we need to query the CMS on page load since it contains both the sitemap definition as well as the page layouts. Further, the CMS returns content as slug IDs and we need to make further calls to fetch the actual content. Using an intermediate server (for API calls as well as server-side rendering) eases some of these pain-points.	
	<i>angular, magnolia CMS</i>	
May 16 - Oct 16	<b>New architecture for Rogers.com pages</b>	<a href="#">Rogers Communications Inc.</a>
	I led a team between 4 and 10 people through the course of the headless-CMS project, which was to be the foundation for future pages on Rogers.com using a new back-end powered by Endeca. It introduced the following patterns: Redux for application state, node.js and koa for a dev server, HATEOAS-style interactions, Polymer elements, and using Nashorn to parse large JSON responses from Endeca. The architecture proved to be robust enough for future work by simple extension.	
	<i>Angular, node.js, koa, typescript</i>	

May 15 - Nov 15      **Mobile application for mechanics**

[Rangle.io](#)

Our client required a large mobile application for mechanics to manage their work-day. It featured integration with Google maps, multiple back-end systems, data synchronization, offline queueing and many complex business rules. This project featured the ability to automatically update the HTML, CSS and javascript of the cordova application on the fly without a need to re-deploy the application. I led a team of 6-8 developers on the project.

*ionic, angularjs, cordova, gulp*

Feb 15 - May 15      **Mobile communications tool for mechanics**

[Rangle.io](#)

Our client required a large mobile application which allowed mechanics to chat, receive alerts and track locations. I led a team of 3 developers on this project. (cordova/nwjs application with isomorphic javascript written using angularjs)

*angularjs, cordova, gulp*

Jul 14 - Apr 15      **Campaign BaaS**

[Rangle.io](#)

Our client required a large campaign management tool (BaaS) for an advertising agency that allowed them to create, maintain, and draw insights from their various ad campaigns. The back-end admin interface was very robust and allowed for storage of any type of data while also allowing for constraints on data types and validation rules, which could be specified at run-time. A cluster of node processes communicated to stay in sync. The back-end API was scalable and passed an external security audit on the first round. I led a team of 5 developers on this project. (The back-end was written in node.js with custom clustering, and used Redis and MongoDB. The front-end was written with Jade, Coffeescript and Backbone. The infrastructure used Grunt and browserify)

*Node.js, express, mongodb, coffeescript, nginx, vagrant, redis, JWT*

Mar 14 - Jul 14      **Public transit back-end**

[Rangle.io](#)

We built a large back-end to handle all transactions from public transit units, tracking passengers and fares. When buses and trains returned to the depot the custom-built hardware transmitted the trip details to the depot server (as binary), and we parsed this into documents to store in Mongo. The admin UI we built around this was deployed to three medium-sized city transportation systems and was fault-tolerant and highly secure. (This was written in node.js and the back-end admin UI was built with Angular.js. This project featured a very robust authentication and authorization mechanism which was isomorphic between the app and API)

*node.js, express, mongodb, angularjs*

I built a merchandise management PWA that allowed for some very in-depth querying capabilities, e.g. what songs on a tour led to the most sales on a particular type of product so that inventory could be managed accordingly. (This was built using angularjs and featured an API written in nodejs. The data was stored in Neo4j)

*angularjs, node.js, express, Neo4j*