

Nicholas Quandt

Milwaukee, WI nquandt.com linkedin.com/in/nquandt github.com/nquandt

Staff Software Engineer · distributed systems, integrations, and platform services

Five years at Milwaukee Tool, from the public site through platform and integrations. Led the re-platform of milwaukeeetool.com off Sitecore onto React and Next.js, and built search, authentication, and image delivery as their own services. Now tech lead for warehouse integrations, and maintainer of METI, the SDK and portal that 200+ apps use to move about a million messages a day.

Experience

Staff Software Engineer

Milwaukee Tool
MetcoIS · WMS Integrations

Nov 2025 – Present
Milwaukee, WI (Remote)

Tech lead for the warehouse integrations. Maintain METI, a C# and .NET 10 SDK and portal other teams build their apps on.

- **Tech lead for the WMS integrations team.** The services keep Oracle, the central ERP, in sync with the warehouse system at each distribution center. Weekly PagerDuty rotation. P1s and P2s stay open until they are fixed, including overnight. If I have not fixed one in 45 minutes, I bring in a second person.
- Added a **third warehouse integration**. The two already running are different systems, so this one needed its own design. An architect wrote that design, and I built it. It is in production, and the post-launch support period is over. On a warehouse that writes straight to SQL Server, I added a check for duplicate primary keys and the rules for what to do with the duplicate.
- **Inherited and maintain METI**, the SDK and web portal other teams build their apps on. It runs their async work on Azure Service Bus and records each step. About **200 apps across 20 to 30 teams** run on it, at about **a million messages a day**. I designed its status-publisher SDK so another system can look up a message's state over HTTP.
- Added **idempotency rules** and a Redis queue with one entry per message. The Redis entry tracks each message until the work finishes. Service Bus locks expire after five minutes, so longer work was being delivered again before it finished. Also added a pause that holds every queue while an Oracle ERP patch rolls out.
- Refactored METI **task cancellation** so a cancelled task still records its status and does not lose its message. The old code was dropping **a few hundred status updates a day**.
- Moved user identities into groups and assigned the roles to the groups. Assigning a role to every identity was using up the subscription's **Azure role-assignment** limit. Also standardized SQL provisioning with Terraform modules.
- Built **Met.Otel**, a .NET 10 OpenTelemetry package any service can take in a few lines (Entra-authenticated exporters, ASP.NET Core wiring, instrumentation defaults). Apps send to an internal collector, and the platform team chooses where the data goes (Datadog today). The warehouse services were the first to move off Application Insights and onto it.
- Added on-call alerts for **receipt, advance ship notice, and shipment** notifications. When one of those fails, someone has to fix it quickly. I learned that by working on the floor during distribution-center launches.

Staff Software Engineer

Milwaukee Tool
Dotcom · milwaukeeetool.com

Dec 2023 – Nov 2025
Milwaukee, WI (Remote)

Led the re-platform of milwaukeeetool.com from proof of concept to production, then the stabilization work after launch.

- Led the **re-platform of milwaukeeetool.com** from a Sitecore Razor monolith to a TypeScript frontend on React and Next.js **App Router**. Home, product, catalog, category, support, and navigation pages moved in phases across four locales, legacy and new served side by side until each cutover. **Lighthouse scores on the first pages improved about 10x**. About **85% of pages** moved; the business dropped the rest.
- **Primary owner of PIPELINE** for 2024, the first cycle delivered on the headless stack, from the backend services to the frontend components.
- Shipped the **product page redesign** on the new stack (PIM-sourced galleries, sticky CTA with regional pricing, spec comparison, FAQ modules) and built new category pages from Figma to production. Those became the templates for the pages that followed.
- Migrated MET.Search and the other extracted services from AKS to **Azure Container Apps** and built the surrounding infrastructure in Terraform: Front Door, Service Bus, private DNS, Key Vault, and container registry across environments, with WAF, TLS, and penetration-test fixes.
- Consolidated per-service host code into a single **MET.Kernel app-builder** so logging, endpoints, and startup ship once for every service. Set up A/B testing on Statsig and trunk-based release pipelines.

Senior Software Engineer

Milwaukee Tool
Dotcom · milwaukeeetool.com

Jul 2022 – Nov 2023
Milwaukee, WI (Remote)

Pulled search, authentication, and images out of the Sitecore monolith into their own C# services, introduced Terraform, and ran the proof of concept that chose React and Next.js for the re-platform.

- Designed and shipped **MET.Search** (FastEndpoints, Solr) to replace the Sitecore Coveo plugin: a product data API with a caching layer and Service Bus-driven index refresh. Category API responses dropped from **about 27 seconds to under 1 second**.
- Designed and built **MET.Authentication** end to end, a YARP reverse proxy that took session and cookie auth out of .NET Framework Sitecore: Auth0 and Azure AD as dual identity providers, Redis sessions, CSRF validation, and per-page routing to React or Razor. Session cookie went from **about 10 KB to 377 bytes**. The routing flag is what made a gradual headless migration possible.
- Built **METIMG** end to end, an on-demand image service. Cache is checked at the CDN, then Redis, then blob, with SQL as cold storage. Unseen sizes resize in the background (PNG to WebP where supported), and the nearest larger variant is served until then. **About 20x fewer image bytes** from 4000-pixel sources, typical load time from **about 300 ms to about 80 ms**, and the AKS cluster from **12 nodes to 4**.
- Built a Svelte/SvelteKit **proof of concept** that beat the legacy site on page metrics, then ported it to **React + Next.js** in a few days so the team could compare the two directly. React won on ecosystem and hiring pool.
- Owned the backend for the 2023 cycle of **PIPELINE**, Milwaukee Tool's annual new-product launch site: search proxies, product lookup, legacy API compatibility, and Service Bus refresh.
- Introduced **Terraform** after a Pulumi proof of concept and provisioned the first production services outside the monolith with CI/CD. Every later service used the same setup.
- Ran the 2023 **security hardening** work (HSTS, CSRF, Auth0 session controls, rate limiting, Snyk gates in the pipeline) to close that year's penetration-test findings, and built the **SEO infrastructure**: hreflang, canonical URLs, regional sitemaps including product pages.
- Ran every intern interview for three years.

Software Engineer II

Milwaukee Tool
Dotcom · milwaukeetool.com

Mar 2022 – Jul 2022
Milwaukee, WI (Remote)

Four months shipping customer-facing features, including the first two Svelte features on the site, while still running the deployment stack.

- Built the product **Compare Tool** end to end: Sitecore API endpoints, computed spec and SKU fields, and a Svelte UI with a bottom sheet and comparison cards. First interactive feature on the site outside Razor.
- Built a **Svelte merchandiser** (query-parameter filtering, carousel, chip navigation) with a custom Sitecore field type, so content authors set up each campaign's merchandising themselves.
- Kept ownership of the **AKS deployment patterns** and Helm charts, moved the build from Gulp 3 to Gulp 4 on Node 16, and led the Brightcove to Vimeo player migration across nineteen rendering surfaces before Brightcove's end-of-life date.

Software Engineer I

Milwaukee Tool
Dotcom · milwaukeetool.com

Aug 2021 – Mar 2022
Milwaukee, WI (Remote)

Joined the dotcom team on Sitecore 10, .NET Framework 4.8, and AKS. Spent the first six months learning the cluster and the deploy pipeline, then took them over.

- Rebuilt the Sitecore deployment pipelines and front-end build, cutting a full container deployment from **about 65 minutes to under 10**.
- Took over the team's **AKS cluster**, which a departed contract team had set up and nobody remaining had learned: control-plane and node upgrades, Helm charts, Ingress-NGINX configuration, and startup, readiness, and liveness endpoints.
- Rewrote the promotion winner-selection service around a purpose-built hash set and correct async access to Azure Blob Storage, removing repeated enumeration and stored-procedure calls. **About 15x more throughput** during peak promotion periods.
- Extended strongly typed Glass.Mapper models across the HR, Identity, Navigation, and Catalog layers, so a renamed or removed content field failed at compile time.
- Removed deprecated libraries from the account, navigation, and redemption modules and replaced static Sitecore context access with dependency injection.

Education

- **M.S. Computer Science**, Marquette University · 2019 – 2021
- **B.S. Mathematics**, Marquette University · 2017 – 2019