SYSTEMS ARCHITECTURE AND INTEGRATION

Dominic Hanzely

Minnesota — remote or Twin Cities
email@dominichanzely.com
dominichanzely.com · uncompose.org

Systems architect and backend engineer focused on integration at scale: platforms that put many third-party systems behind one coherent model, and migrations carried all the way through deletion. A decade on a 35-service content-distribution platform as its second-largest backend contributor, with a record of building durable tooling and testing practices that other engineers adopt and maintain. Now building Uncompose, an open-source music source-separation toolkit in Rust.

EXPERIENCE

BuzzFeed

Minneapolis, MN

Staff Software Engineer, 2022–2026 · Senior Software Engineer, 2017–2022 · Software Engineer, 2015–2017

- **Architected and operated a social publishing platform from its inception through a decade of production use**: 35 services putting six third-party APIs (Facebook, Instagram, YouTube, X/Twitter, TikTok, Threads) behind one draft, job, and credential model, serving an editorial UI and an automated rules-driven pipeline.
- **Split the original auto-publishing monolith into five event-driven stages**: ingest, safety screening, relevance scoring, rule matching, and post creation. The pipeline stayed in place for eight years, and the architecture diagram I created remained a reference through four successive team incarnations.
- **Led the collaborative redesign and implementation of the distribution rules engine**, replacing a ~22,000-line configuration file (507 rules, 119 destination channels) with a versioned relational model and tiered UI access for product and editorial users. Landed the v1-to-v2 cutover through a 21-step migration with zero duplicate social posts or interruption to automated publishing, then deleted the legacy system.
- **Created 14 production services, 12 from scratch**, each with tests, CI, and documentation in its first commit. Extracted repeated queue-reader behavior into a shared package so parsing, retries, error handling, and instrumentation could evolve centrally; 20 deployed services depended on it, with eight engineers contributing over 6.5 years.
- **Built the request-path selection engine for an editor-configured on-site campaign-targeting service**, using simple primitive comparisons, data structures, and caching to reduce p95 from 267 ms to 3.7 ms at 200 requests/second on one 512 MB container. Built an AI skill around the monorepo load-testing suite that generated test artifacts and was reused by other teams with little or no modification.
- **Selected for BuzzFeed's AI tooling/workflow accelerator**; conceived and shipped a VS Code chat extension doing client-side retrieval over internal documentation, then implemented an MCP tool surface giving agents access to the automated-publishing system and its operational state. Engineers could use it to investigate questions like why a particular piece of content failed to auto-publish to a destination.
- **Mentored a cloud security engineer through a three-month formal program that continued for nearly three years**, supporting his growth in Python and software development as he earned a promotion. Taught a product owner with no engineering background to make production rule changes directly; she shipped 168 reviewed commits in a year. Managed a 2018 intern who was hired full-time in 2019 and remained at BuzzFeed through my departure.

University of Minnesota, Office of Information Technology

Minneapolis, MN

Senior Web Developer, Jul 2013 – Feb 2015

- Built and maintained the academic technology behind the university's online courses and video services (PHP, JavaScript, MySQL, AngularJS, Symfony/Silex, Gearman).
- Integrations, interface work, and core customizations to the university's fork of Moodle, including contributions published in the Moodle plugins directory.

Media Logic Group

Excelsior, MN

Director of Development, Apr 2010 – Jul 2013 · Formerly WeatherNation; renamed mid-tenure

- Architected a wind-energy forecasting application combining near-real-time turbine telemetry with NOAA data and custom-tuned WRF model output to predict farm power generation; wrote custom Modbus and OPC-XML implementations to pull SCADA data off wind-farm turbines in near real time.

EARLIER EXPERIENCE

Star Tribune, Web Applications Developer, 2007–2010 · Minneapolis, MN
Full-stack internal and public applications spanning subscriptions, data interactives, and advertising.

Cipher Multimedia, Head of Technology, 2002–2008 · **Uniloc**, Software Engineer, 1999–2002
Digital rights management, web-service integration, binary repackaging, and reverse engineering in C/C++ and related stacks.

PROJECTS

Uncompose Ecosystem

uncompose.org

Sole Architect and Software Engineer, Jul 2026 – Present

- **Built an open, local-first toolkit for separating recorded music into stems, preserving provenance, and evaluating separation results**: three Rust command-line tools sharing one command namespace, all published on PyPI under MIT with a live family site.
- **Designed a language-agnostic contract between the Rust core and Python separation engine**,keeping them isolated behind a process boundary. Connected the tools through versioned JSON schemas rather than shared code, with integrity states, atomic writes, and advisory locking.
- **Built entirely in public with a human-gated AI agent workflow**: 30+ architecture decision records, ~135 issues and ~80 pull requests, with agent-authored branches merged through human review. Went from first commit to three published packages and a live site in 13 days while learning a new domain.

TECHNICAL SKILLS

LANGUAGES	Python, Rust, TypeScript, JavaScript, SQL, PHP, Perl, C/C++
BACKEND	FastAPI, Django, REST and OpenAPI design, JSON Schema, event-driven and message-queue architecture (NSQ, RabbitMQ, SQS), MCP tool surfaces
DATA	PostgreSQL, MySQL, MongoDB, Redis, Memcached, schema migration and versioning, embedding retrieval
FRONTEND	React, Redux, AngularJS, Backbone, Web Audio
INFRASTRUCTURE	AWS (EC2, ECS, S3, SQS), Docker, GNU/Linux, Cloudflare, CI/CD, load testing, metrics and structured logging
PRACTICE	Spec-driven and AI-assisted development with coding agents, test-driven development, pytest, mypy, ruff, code review, mentorship