

# Daniel Micaletti

Staff / Founding AI Engineer · Context engineering · MCP servers, sub-agents, agent skills · Eval-driven development

Phoenix, AZ · danielmicaletti@gmail.com · 415-996-6494

Live agent → [danielmicaletti.dev](https://danielmicaletti.dev) · [github.com/dmic23](https://github.com/dmic23) · [linkedin.com/in/daniel-micaletti](https://linkedin.com/in/daniel-micaletti)

## SUMMARY

---

Senior Engineer at Republic Services (2019–present), tech-lead developer on RISE's **Haulgorithms team** (~17K collection vehicles, 5M+ daily pickups). In parallel, shipped **14 production AI/agent projects in the last 18 months** — production MCP servers (stdio + Streamable HTTP), an autonomous agent daemon, a personal-AI **context engine** over a temporal knowledge graph, and an agent-native portfolio whose **live agent at danielmicaletti.dev** runs as a Streamable HTTP MCP endpoint, stdio MCP via npm, CLI, and web UI from one handler. Builder *with* models, not *trainer of* them — context engineering, MCP servers, sub-agents, agent skills, eval-driven development. Targeting Staff/Founding scope at AI-native teams. Remote-first.

## SELECTED PROJECTS

---

### Agent Portfolio — [danielmicaletti.dev](https://danielmicaletti.dev) (live agent demo)

One handler package, four interfaces: **Streamable HTTP MCP at /mcp**, stdio MCP via `npx @danielmicaletti/portfolio`, city CLI, Next.js 16 web UI. The portfolio *is* a working agent harness — anyone can talk to the live agent or wire the MCP server directly into Claude Desktop / Cursor.

- **91 tests** across core, mcp, and cli packages; eval-fixtured via `KNOWLEDGE_USE_EXAMPLES` for stable assertions across forks.
- Turborepo + pnpm workspaces, TypeScript 5.6+ (Bundler resolution), `@modelcontextprotocol/sdk@1.29.0`, Zod 3.25, Vitest 2.
- Stateless /mcp route handler verified end-to-end in Claude Desktop (initialize → tools/list → invocation handshake).
- AI provider layer: Vercel AI SDK v6 (Agent abstraction, ToolLoopAgent, tool approval) + AI Gateway with Anthropic primary / Gemini fallback.

### The Francine — Multi-Workspace Claude Code Orchestration Platform

Tauri desktop shell + SvelteKit frontend + Cloudflare Worker API + mobile PWA that spawns, monitors, and controls multiple Claude Code sessions through a 4-tier agent hierarchy. Live: [francine-work-orders-app.pages.dev](https://francine-work-orders-app.pages.dev).

- **Agent hierarchy:** Meta-Orchestrator → Supervisor → Worker → Verifier — strategy selection + budget enforcement, atomic-task decomposition, quality assurance.
- **Hybrid security model:** local vault for secrets/PII → masked proxy (tokenize / mask / block) → cloud AI; 7 data categories (PII, financial, credential, health, contact, identifier, location), 4 sensitivity levels.
- **Intelligence layer:** knowledge graph + world model + episodic / procedural / semantic memory; event sourcing for all state changes.
- **Cross-platform:** Tauri 2.x + Rust on desktop; SvelteKit 2.49 + Svelte 5 (runes) + xterm.js on the front-end; Cloudflare Workers + D1 on the back-end; mobile PWA via Cloudflare Pages.
- **1,650 passing tests** across a 4-workspace pnpm monorepo (desktop 1,238 · worker 219 · shared 106 · mobile 87).

### Penny — Autonomous Agent Daemon (PRAO architecture)

Goal-driven daily cycle that discovers, prioritizes, acts, and observes across a 14-project portfolio. Subsumed an entire prior "Chief of Staff" daemon system.

- **Parallel execution by project (3x throughput)**, multi-turn retry with auto-rewind on MEDIUM+ risk steps, file-checkpointing for crash recovery.
- Thompson-Sampling prioritization + git-activity scoring for goal selection.
- **91% cycle success rate over 20+ runs**, 219 tests, ~6,500 LOC, multi-dimensional cost attribution per cycle.
- Python 3.11, Claude Agent SDK, SQLite (WAL), Slack, custom project-health MCP — observability via structured logs + Slack digests.
- Paired with **Agent Harness** — eight additional production scheduled agents on Anthropic Desktop scheduled tasks (daemon-health, weekly dependency / stale-branch / CLAUDE.md-staleness scans, daily test-suite runs, weekly git-activity reports) with severity-tiered Slack notifications.

### Nanda — Personal AI Context Engine

**Agentic retrieval** over a personal corpus (healthcare, projects, email, calendar) with a temporal knowledge graph as the memory layer. Slack-native, with a deployed Next.js chat companion.

- Ingestion pipeline → Zep temporal KG (619 facts) → query orchestrator with **agent-driven query planning** and source-conflict resolution.
- **Hybrid retrieval:** Voyage 4 embeddings + PostgreSQL + pgvector + Redis Streams event bus.
- slack-bolt daemon (Socket Mode, **63 tests**) + Next.js 16 chat UI deployed to Vercel.
- Python backend / TypeScript frontend; AI SDK v6 + `@ai-sdk/anthropic` streaming.

**Production MCP Servers (Gmail · Reddit · project-health · agent-portfolio)**

Four production MCP servers across **stdio and Streamable HTTP transports**, security-hardened, used across the personal portfolio.

- **Gmail** (stdio) — OAuth2 token files locked to `00600`, error sanitization, no path leakage in stack traces.
- **Reddit** (stdio) — input validation on all identifiers, SSRF prevention via redirect domain restriction, thread-safe rate limiter, public-JSON mode that auto-upgrades to PRAW when credentials arrive.
- **project-health** (stdio) — wired into Penny for live cross-project status; reusable via `.mcp.json`.
- **agent-portfolio** (stdio + Streamable HTTP) — same handler package serves both transports plus a CLI and a Next.js route handler; verified in Claude Desktop end-to-end.

### SWPPP Field Inspection App — Live on Vercel

Mobile-first PWA for construction-site stormwater (SWPPP) compliance inspections. Production end-to-end: 65+ files, 18 routes, full e2e verified. [swppp-app.vercel.app](https://swppp-app.vercel.app).

- **Offline-first** with Dexie.js + IndexedDB sync; 4-step inspection wizard with photo capture, GPS tagging, signature pad, weather auto-fetch (Open-Meteo); auto-corrective actions from failed BMPs; PDF report generation (jsPDF).
- **Multi-tenant Supabase** (12 tables, 8 enums, 3 storage buckets, full RLS) with admin / inspector / viewer roles.
- Stack: Next.js 16, Tailwind v4, shadcn/ui, Supabase, Vercel.

## EXPERIENCE

---

### Senior Software Engineer · Republic Services

Aug 2020 — Present · Scottsdale, AZ

- Tech-lead developer on the **Haulgorithms team** within **RISE**, Republic's digital dispatch + routing platform — primary internal operations system used to create, update, route, and close routes in real time across **~17,000 collection vehicles** and **5M+ daily pickups** at **367 operating locations** in 41 states + Puerto Rico.
- RISE is part of Republic's publicly disclosed **>\$1B multi-year digital-platform investment**, with leadership tying RISE routing to a "nine-figure" multi-year efficiency opportunity (CEO, multiple 2024–2025 earnings calls).
- Lead a **5-developer full-stack pod** inside a 12-person cross-functional team (4 QA Engineers, PO, BA, Scrum Master); drive architecture and code review across the pod; cross-team integration with Map, Tablet, and BA teams. Stack: **Angular + NgRx** front-end; **Node.js + AWS Lambda + API Gateway + DynamoDB + S3** back-end; **Terraform IaC**; **Splunk** observability.
- Designed and shipped **Hours of Service (HOS) notification system** — automated DOT-compliance alerts covering **18K+ drivers across 41 states**, triggering at 3 regulatory thresholds (60-hr weekly · 12 / 13 / 14-hr daily) to mitigate **\$16K+ per-violation FMCSA fines**; integrated **SendGrid** with timezone-aware scheduling across 4 U.S. time zones.
- Built **Route Goals** platform with 4 goal types (planned · dispatched · dynamic · actual) — real-time and historical operational targets per route driving efficiency metrics across **350+ divisions**; architected the **S3 data pipeline** for collection-time analysis supporting **5M daily pickups**. Delivered **Sequence Adherence** tracking — active + historical stop-sequence compliance across **18K+ routes daily**, improving route execution quality and on-time service for **14M+ customers**.
- Shipped **RISE Crew Out Route Close** — digital end-of-day route-completion capture with multi-format route counting, digitizing paper-based processes across **18K+ daily routes**; built **Small Container Standards** division-level rule engine for container standardization and disposal-trip calculations, eliminating manual per-division setup across **350+ divisions**.
- Senior engineer on **DRIVE**, Republic's enterprise route-optimization initiative integrated with a third-party Routing-as-a-Service vendor. As Haulgorithms tech lead, owned the constraint-input surface (time/access, driver requirements, equipment, disposal-site contracts, waste-stream compatibility) feeding DRIVE's optimization pipeline; built **ESRI-based real drive-time** features replacing estimates with geospatial actuals across the route network serving **5M daily pickups**, improving route accuracy and reducing windshield time; owned **disposal-facility data infrastructure** (schema, sync/write services, config-driven deployments) as single source of truth for downstream routing and reporting systems.
- Shipped DRIVE prototype → pilot within an enterprise architecture: **Mashery proxy + mTLS** for RaaS integration, async **Lambda + S3 pre-signed URLs + webhook callbacks** under a **5-minute SLA**, **Splunk** observability with cross-system trace IDs — supporting day-before optimized plans on driver tablets and improved route-margin "swing percentage" (continuous service flow, no backtracking).
- Pulled onto Republic's **AI/ML R&D team (2024)** to architect an end-to-end ML pipeline for **container fingerprinting** — computer vision identifying waste containers from photos and matching them to customer accounts across Republic's **13M+ customer base** via **AWS SageMaker** (training pipelines, inference endpoints, model versioning); built the **ESG image data pipeline** ingesting container photos and metadata into the AI platform; delivered a **Portal POC** for AI-powered disposal-ticket classification.
- Built and shipped **rsgSuperAgents** — a zero-dependency **21-agent VS Code Custom Agents framework** (~14,600 LOC) with **34 reusable skills**, **15 instruction modules**, and **8 security/audit hooks**, adopted internally for PRD-driven dev workflows: single **@Orchestrator** auto-delegates to specialist **sub-agents** with **3 automated quality gates**, backed by deep procedural **skills** (Rally integration, PRD generation, background orchestration); production safety via handoff-loop guard, conflict-aware **LockedPaths** concurrency, PII/secret context scrubbing pre-delegation, and session persistence.
- Early **evangelist for AI-assisted development workflows** — leading live demos for engineers and QAs to drive AI-tooling adoption across the engineering org; positioned as **AI implementation point of contact** for defining implementation strategies

(per 2025 annual review).

- Established **coding standards and PR review process** for both the RISE and overall Republic Services org teams; primary **RISE Portal** technical liaison with functional partners — translating business requirements into delivery strategy and architectural decisions across **197+ delivered stories over 5+ years**; mentor new engineers from onboarding to proficiency, **recognized internally for elevating team performance through knowledge sharing and code review**; participate in **SOX compliance** efforts safeguarding data integrity across platform features serving Republic's **\$16B+ annual revenue operations**; active contributor to the **RISE Architecture Review Board** (weekly) and **Developer Community of Practice** (weekly).

### **Full-Stack Software Developer (Contract) · Republic Services**

*Jan 2019 – Aug 2020 · Scottsdale, AZ*

- One of the early engineers on team for the internal data-aggregation portal that became RISE.
- Built early Angular + AWS Lambda components and the REST integration layer; pair-tested with QA, partnered with BAs for product alignment.

### **Frontend Software Developer (Contract) · Matson Navigation**

*Jul 2018 – Jan 2019 · Phoenix, AZ*

- React + Redux UI for an international freight & shipping platform; integrated against backend microservices.

### **Full-Stack Web Developer · Periwinkle Labs (Freelance)**

*Nov 2013 – Jul 2018 · San Francisco / Curitiba / Moscow*

- Production Python/Django + React applications for clients across the US, Brazil, and Russia. Owned full project lifecycle: scope, design, build, launch.

### **Sales Engineer / Regional Channel Manager · vWorkApp Inc.**

*Oct 2011 – May 2013 · San Francisco, CA*

- Direct sales + integration delivery for a SaaS field-service platform across North and South America. Channel partnerships with Salesforce, Google, TomTom, Zoho.

## **SKILLS**

---

**AI / Agents** — Context engineering · Claude Agent SDK · Vercel AI SDK v6 (Agent · ToolLoopAgent · tool approval · AI Gateway) · Model Context Protocol (servers · sub-agents · agent skills · stdio + Streamable HTTP)

**Retrieval** — Agentic retrieval · hybrid pgvector + temporal KG (Zep) · Voyage embeddings · agent-driven query planning · source-conflict resolution

**Evals & Observability** — Eval-driven development (golden datasets · LLM-as-judge · CI quality gates) · multi-dimensional cost attribution

**Languages** — TypeScript (5.6+) · Python (3.11+) · JavaScript

**Web** — Next.js 16 (App Router) · React · Tailwind v4 · shadcn/ui · Node.js

**Data** — PostgreSQL · Supabase · SQLite · pgvector · Redis

**Platform** — Vercel · AWS Lambda · Docker · Turborepo · pnpm workspaces

**Enterprise / Backend** — Angular · NgRx · AWS API Gateway · DynamoDB · S3 · Terraform · Splunk · ESRI · SendGrid

**Practices** — TDD (2,000+ tests across portfolio) · agent system design · security-hardened MCP · autonomous-pipeline architecture · multi-tenant RLS