# Bradley S. Isenbek

AI systems architect and machine learning engineer: agent platforms, data architecture, and the hardware they run on.

brad@bradley.io · bradley.io · github.com/tinymachines · Grand Rapids, MI

**Open to full-time roles in AI systems and data architecture.** Remote, or on site in the Grand Rapids area.

## SUMMARY

AI systems architect and machine learning engineer. More than fifteen years designing secure, large-scale data systems for government and enterprise clients, including a classified project for the United States Government and systems that process billions of records.

Since February 2026, building with Claude and other AI models every day:

- **Campaign Brain**: 84 self-hosted services, ready for production. They give AI agents a memory, plan and run their work, and gather public information.
- **MyFinalWishes (mywishes.com)**: digital estate planning, shipped on the SysForge platform.
- **A 6502 chip and an NES**: simulated transistor by transistor, at tinymachines.ai.
- **TerraPulse**: an open platform for measured climate and earth-science data.

Every working session, and what it cost, is on the public record.

## EXPERIENCE

### **Founder and AI Systems Architect**, SysForge.ai — 2024 to present

AI systems and development firm. Since 2026 the work has centred on Campaign Brain, a production-ready platform built on the SysForge fabric, on products shipped through it, and on the self-hosted infrastructure under both.

#### CAMPAIGN BRAIN, THE PLATFORM

- Built Campaign Brain, a production-ready platform on the SysForge fabric: 84 Python services and about 3,390 documented endpoints, each read endpoint also an MCP tool. Over 20,000 commits of his own since February 2026, AI pair-programmed, chores excluded.
- Designed a journaled state kernel for agents: five classical primitives behind one loop, with the search strategy as configuration. Every change goes to an append-only journal, so a crashed run replays to its exact next step.
- Merged three in-house workflow engines into one: YAML graphs that can wait days at an approval gate and resume exactly once. Its planner turns two plain sentences into a graph checked against the live service catalog, and nothing runs until a person clicks Run.
- Built the service mesh underneath: each service announces a card describing itself, peers find each other by broadcast, membership expires by TTL so a dead node ages out, and one gateway calls services by name.
- Built the platform's intelligence crawler: an async jobs API running YAML pipelines (crawl, archives, screenshots, transcripts, vector search) on a self-built cluster of workers. Its largest measured run processed about 950 GB in nine days.

#### DISASTER RECOVERY

- Rehearsed a from-nothing rebuild on a second host: fresh secrets, 63 empty databases, 47 storage buckets, every service in dependency order. 67 of 93 units started clean, the other 26 became tickets, and the run became the first fresh-host runbook.
- Designed a whole-server encrypted offsite backup that fits a nightly window, refuses to run if key material would leave the building, and cannot erase its own history. Restored a 235 MB database from backup in 33 seconds with no count mismatches.
- Designed ZFS replication for an embedded database and its files, with a failover guard that never promotes a lagging standby.

#### MYFINALWISHES, SHIPPED THROUGH SYSFORGE

- Built MyFinalWishes, a digital estate-planning platform, as its only engineer: a prototype rebuilt as one TypeScript application (TanStack Start, React 19, SQLite with Drizzle), with 459 commits and 68 migrations from July to September 2026.
- Designed per-user envelope encryption: AES-256-GCM under a per-user key wrapped by a master key held outside the database, each value bound to its user, table and column, and files sealed in 64 KiB chunks.
- Took it to production with a replicated standby, nightly verified off-site backups and a health check that proves sign-in, decryption and mail delivery. About 1,500 unit and end-to-end tests gate every change.

#### INFRASTRUCTURE

- Run more than 50 hostnames on owned hardware with nginx, systemd, certbot and self-hosted BIND DNS, version-controlled units and encrypted secrets. No cloud host.

MyFinalWishes: mywishes.com

Python · FastAPI · TypeScript · Claude API · MCP · SQLite · ZFS · WireGuard · OpenWrt · systemd · nginx

### **Research and open source**, tinymachines.ai, TerraPulse, tinychase — 2026 to present

Chips simulated from die photographs, an open climate and geophysical data platform, and a cyclone model that runs in the browser, all built in public.

#### TINYMACHINES.AI

- Designed and shipped tinymachines.ai in six weeks (763 commits, August to October 2026): four projects on Next.js 16, FastAPI and Rust/WebAssembly, an API spoken as REST and MCP, documentation in English and Japanese, and 42 Playwright specs on the live site.

- Built a switch-level MOS 6502 simulator in Rust and WebAssembly on halfphi, an MIT-licensed library: 1,725 nodes and 3,510 switches, matching visual6502 bit for bit across a 3,000 half-cycle trace. Faster engines derived from it are held to it at the pins, up to a GPU kernel stepping 128,000 machines at once.
- Extended the method to the NES: switch-level Ricoh 2A03 and 2C02 chips, an NTSC signal and CRT model, and seven cartridge board types, passing blargg's CPU, timing, sprite-hit and APU test ROMs. More than 830 commits in seven weeks.
- Built a bench that holds the model to an unmodified NES under identical inputs: an inline pad bridge counting the console's latch and clock pulses in hardware, and a Raspberry Pi driving reset, power and the oscilloscope.

**TERRAPULSE, ARCHITECT AND BUILDER**

- Architected and built TerraPulse, a measured-data climate and geophysical platform: 80 indexes holding 14.2M records from 183 upstream datasets (USGS, NOAA, NASA, EPA, USDA and others) on PostgreSQL/PostGIS, DuckDB and Parquet.
- Shipped its public Model Context Protocol server: 11 read-only tools over the indexes, research workspaces and a 1,353-node knowledge graph, connected to claude.ai as a custom connector.

**TINYCHASE, ENGINEERED**

- Engineered tinychase.com in nine days (128 commits): an installable PWA where DuckDB-WASM reads static Parquet and a tinygrad cyclone transformer runs in the browser on WebGPU, within 0.0000064 kt of the Python reference.

**BRADLEY.IO**

- Run bradley.io on owned hardware: live dashboards fed by real instruments, a public record of every AI working session and its cost, and a staged deploy with automatic rollback behind more than 400 releases since February 2026.

`tinymachines.ai` · `tinymachines.ai/6502` · `terrapulse.info` · `tinychase.com` · `github.com/tinymachines`

`Rust · WebAssembly · WebGL2 · WebGPU · tinygrad · Next.js · FastAPI · PostgreSQL/PostGIS · DuckDB · Parquet · MCP`

### **Architect and Developer**, VictoryText, LLC

`Grand Rapids, MI · 2022 to present`

Enterprise-grade, high-volume messaging platform with strict uptime and compliance requirements.

- Architected a high-availability messaging platform handling millions of messages at 99.9% uptime, using AI for anomaly detection and delivery optimization.
- Developed the RESTful API layer in FastAPI, integrating multiple carriers (Twilio, Bandwidth, Kaleyra, Telnyx) with automated failover.

`Python · FastAPI · DynamoDB · Lambda · CloudWatch · QuickSight`

### **Architect and Developer**, ConservativeConnector, LLC

`Grand Rapids, MI · 2018 to 2022`

Large-scale email data management systems processing billions of messages and more than 100 million contact records.

- Designed high-volume data infrastructure on AWS, with ML-driven data quality scoring and deduplication.
- Integrated and normalized 4.9 billion data points for ML model training, relationship discovery, and predictive analytics.

`Python · AWS · Snowflake · PostgreSQL · ML pipelines · ETL`

### **Senior Architect**, TransUnion, TruLookup Division

`Boca Raton, FL · 2014 to 2018`

Investigative intelligence platform providing insight on people, assets, and relationships from more than 10,000 data sources. Served law enforcement, government agencies, and legal clients.

- Lead developer on a classified project with the United States Government, covering secure system architecture and federal compliance.
- Architected distributed infrastructure processing billions of records and millions of queries a day at better than 99.9% uptime.

`C# · .NET · Apache Solr · Elasticsearch · SQL Server · Distributed caching`

## EARLIER CAREER, 1997 TO 2014

`2012 to 2014` **Senior Architect**, nextSource / PeopleTicker; `2011 to 2012` **Application Developer**, nextSource; `2009 to 2011` **Senior Developer**, Linx Communications, Inc.; `2008` **Software Development and System Design**, J. Lack Consulting; `2008` **Senior Developer**, EnablePay, Inc.; `2007 to 2008` **Senior Developer**, AmeriCorp, Inc.; `2005 to 2007` **Senior Developer**, Evo Merchant Services, Inc.; `2004 to 2005` **Senior Developer**, Direct Insite, Inc.; `2002 to 2004` **Senior Developer**, CCSI; `1997 to 2002` **Chief Technology Officer and co-founder**, PipeLive, LLC.

**Awards:** PC Magazine Winner's Circle (PipeLive, LLC, 1997 to 2002); TMC Labs Innovation Award (PipeLive, LLC, 1997 to 2002).

## EXPERTISE

| | |
|---|---|
| **AI and machine learning** | Claude and the Anthropic API · Model Context Protocol (MCP) · Agent state and orchestration · OpenAI API · Agentic frameworks · Prompt engineering · RAG pipelines · Vector databases · Hugging Face · Entity resolution · Relationship mapping · Recommendation engines · Statistical modeling · scikit-learn · Weka |
| **Languages** | Python · Bash · SQL · C# and .NET · Rust · C and C++ · Java · JavaScript and TypeScript |
| **Cloud and infrastructure** | AWS · Linux administration · Docker · Distributed systems · High-availability design · Load balancing · Disaster recovery · WireGuard and OpenWrt · ZFS · nginx, systemd and BIND |
| **Data and search** | Snowflake · PostgreSQL · MySQL · SQL Server · DynamoDB · Redis · Elasticsearch · Apache Solr · ETL pipelines · Hadoop and Hive · Vector embeddings · SQLite · DuckDB and Parquet · PostGIS |
| **Development** | FastAPI · REST APIs · Microservices · Git · CI/CD · Message queues · Real-time processing |

## EDUCATION

**Advanced Object-Oriented Programming**, Columbia University · **Computer Science coursework**, Plymouth State College · **Physics Engineering**, University of Buffalo