Greg Little

Software engineer, generalist · Ph.D. Computer Science, MIT
Berkeley, CA (open to relocation) · glittle@gmail.com · (617) 308-4673 · glittle.org · github.com/dglittle

In my 2011 MIT thesis I wrote: “I see programs that are able to repair themselves … hiring other workers to verify the solution before integrating the changes.” Then the workers were people on Mechanical Turk. Now they are AI agents, and lately I have worked on running them in production, and on the checks that say when to trust them. I also design real-time sync algorithms and protocols (CRDTs, Braid).

EXPERIENCE

Staff Engineer, AI Agents — Ergeon Jun – Oct 2026

Home-improvement company whose sales, customer-service and delivery desks are run by Claude agents.

- Built a replay harness with blind LLM judges to test agent changes on real production runs before shipping. It showed a fresh context per run cost the sales desk about the same as a long-lived session, with fewer flagged errors, and it caught two bugs in agent skills.
- Cut the sales desk’s cost per agent run 16% by matching prompt-cache lifetime to its traffic, and fixed the cost meter, which would have reported the saving as a 42% increase.

Independent Engineer & Consultant 2018–2026

- **Braid / Invisible College** (braid.org): co-designed **Antimatter**, the first peer-to-peer sync algorithm that prunes CRDT/OT history under arbitrary partitions, down to zero overhead in steady state. Designed the **Sync9** text CRDT. Co-authored the IETF Internet-Draft **Braid-HTTP**.
- Sole engineer on a stealth Figma-style collaborative design tool (2019–2024): full stack, with AI integration.
- **Enso** (2022–2024; CTO from 2023): end-to-end-encrypted real-time collaboration for text and files.
- **SIV** (2020): an implementation of Neff’s verifiable secret shuffle for e-voting.
- Other contracts: a traffic simulation for a personal-rapid-transit client; a GPU Equihash miner.
- Co-instructor, Bradfield School of Computer Science, San Francisco (2018): Languages, Compilers and Interpreters.

Labor Scientist — oDesk (now Upwork) 2011–2013

- Researched distributed hiring and on-the-fly teams; ran public experiments, e.g., hiring experts in 20-minute chunks.

Game Developer — ScreamingGames 2000–2002

- Shipped *Operation: Blockade* and *Operation: Steel Tide* (Infogrames, Windows, 2002). One of two programmers: built the 3D graphics, the game AI (aircraft movement, strafing-path planning) and the game editor.

EDUCATION

MIT: Ph.D. Computer Science 2011 (*Programming with Human Computation*, advisor Rob Miller); S.M. 2007; three TA appointments. **Arizona State**: B.S. Computer Science.

SELECTED RESEARCH (first author; ~5,400 citations overall)

- **TurKit** (UIST 2010): crash-and-rerun programming, which reruns a script without paying again for human steps already done. An early form of today’s durable agent workflows. Covered by *MIT Technology Review*.
- **Keyword Programming in Java** (ASE 2007; *Automated Software Engineering* 2009): loose keywords → valid code.
- **Koala** (CHI 2007, IBM Almaden): web automation that became IBM CoScripter.

SKILLS

Python/Django · TypeScript/Node · LLM agents (Claude), evals, prompt caching · CRDTs and sync protocols · WebGL