

Charles Lowell

Student of Software: structured programming, functional programming, developer experience that feels like flying.

cowboyd@cogentdude.com · github.com/cowboyd · cogentdude.com

Stance on AI

AI has not changed the fundamental physics of software quality; only the masses involved. Instead of launching rocks, we're now hurling planets. As such, precision in software structure and comprehension of the mechanics involved are more important than ever.

Selected Work

Original projects I created and led.



Effection

Structured concurrency and effects

TypeScript

Delimited Continuations

Algebraic Effects

The lifetime of every async operation is bounded by a scope so that cancellation, cleanup, and error propagation happen automatically. Based on delimited continuations, originally derived by [translation of OCaml shift and reset primitives](#).



Bombshell Terminal

Layout engine for terminal emulators

C

WASM

Terminal I/O

TTY Protocols

Brings the box model, flex-like flow, z-index compositing, and animations to text-mode UIs.

The Ruby Racer

Embedded interpreter

[Ruby](#) [C++](#) [V8](#) [Native Extensions](#) [FFI](#)

Embeds the V8 JavaScript engine inside Ruby. For the entire Rails-asset-pipeline era this was how Ruby apps ran JS at build and request time; it was a foundational gem that required an intimate knowledge of both interpreters and how to share memory between them.

Simulacrum

Service simulation

[Simulation](#) [HTTP](#) [TypeScript](#)

Run an application against simulated HTTP services that speak the same protocol as the vendor you'd otherwise stub out, so it exercises the same code paths as production does. Ships simulators for GitHub, LDAP, and advanced OAuth flows out of the box.

Interactors

Semantic user actions, not selectors.

[Acceptance Testing](#) [Page Objects](#) [Browser](#) [DOM](#)

Composable page objects for actuating components via semantic action. This lets agents control a web interface with a semantic control language rather than a mish-mash of DOM selectors and timing hacks.

Speaking & Writing

Selected essays

[The await event horizon in JavaScript](#) (2023)

[What is Strict Structured Concurrency?](#) (2026)

[RYE: Repeat Yourself Enough](#) (2009)

[Math is Just Another Framework](#) (2018)

[See all](#) →

Talks & interviews

[devtools.fm ep. 134](#)— Structured concurrency & Effection (2025)

[Learn With Jason](#)— Fixing async / await (2024)

[JavaScript Jabber #337](#)— Microstates (2019)

[EmberConf](#)— Immutability & composable components (2016)

Podcasts

[The Frontside Podcast](#) — host / co-host, 100+ episodes

[Drunk and Retired](#) — co-host with Michael Coté, mid-to-late 2000s

Employment

Frontside Software

Founder, Head of R&D • 2005–present

Founded Frontside on the premise that the patterns of software quality and architecture are universal and apply just as much to the frontend as to the backend. Over two decades we have provided custom software solutions to Fortune 500 companies for the most challenging problems faced by their development teams.

ThoughtWorks, LLC

Software Consultant • 2000–2003

Consulted on Java and Ruby engagements during the years when Agile and XP were still an argument, not a certification.

FundsXpress, LLC

Founding Programmer • 1996–1999

I was the second programmer hired at FundsXpress, an Austin startup founded in 1996 to build web-based retail and business banking software for community banks (in an era when the phrase "internet banking" was itself still novel). I worked to implement their backend, billpay system, and frontend.

Education

University of Michigan

BS, Computer Science & Linguistics • 2006

4.0 GPA in major

LBJ Science Academy, Austin, TX

High School Diploma • 1994