The Lego Bricks Era

11 min read • 2,481 words

Themes: Consciousness   Technology   Mental Health   Programming   Recursive
Spiritual   Contemplative

There was a time, roughly 2008 to 2016, when open source felt like a movement with a genuine ethos. Not a business strategy. Not a talent acquisition funnel. Not a way to build a community you could later monetize. An actual ethos: let's make amazing lego bricks and share them with each other.

The joy was in the craft and in the giving. You built something beautiful, you put it out there, and if it was good enough, people used it. The reward was the use. The community was the compensation.

I was a kid in Virginia with no credentials, no computer science degree, no connections to anyone who mattered. And I built Requests, and it became the standard HTTP library for one of the most popular programming languages on Earth. Not because I had a business plan or an exit strategy or a venture capitalist who believed in the vision. Because I made something beautiful that solved a real problem, and the community recognized it.

That story could still happen today. I don't want to be the old guy saying the kids don't understand. But it's less common now, and the reasons are structural, not moral.

What Changed

The best open source projects increasingly come out of VC-backed companies. People create projects with exit strategies. Sustainability plans. Revenue models. And there's nothing wrong with that. The old model, where you give everything away and hope someone hires you based on your GitHub profile, was never sustainable for most people. It worked for me because I was young and had no dependents and got very lucky.

But something was lost. The lego brick ethos, the pure joy of making beautiful things and sharing them because sharing is what you do, that feels like it belongs to a specific moment in history rather than an ongoing reality.

> Money and open source potentially shouldn't mix, at least not by default. The introduction of revenue models into open source was necessary and understandable, but it changed the incentive structure in ways that are hard to undo. When you build for love, you optimize for beauty. When you build for sustainability, you optimize for adoption. These aren't always the same thing.

I don't have a villain for this story. The shift from idealism to pragmatism in open source wasn't caused by bad actors. It was caused by the same forces that cause all idealistic movements to mature: people got older, had families, needed health insurance, and discovered that passion doesn't pay the mortgage. The golden era ended because the people who created it grew up. I'm one of them.

Craft, Not Lifestyle

For most of my adult life, technology was my identity. Not just my profession. My identity. I thought in code. I socialized at conferences. I measured my worth by GitHub contribution graphs and download counts. The community that took me seriously when nothing else did was a tech community, and that created a debt of identity that I carried for a long time.

But somewhere in the last few years, without any dramatic break or pivotal moment, tech became craft instead of lifestyle. I don't know when exactly that shift happened. It wasn't a decision. It was more like the way your eyes adjust to darkness: gradual, imperceptible in any given moment, but eventually you realize you're seeing the room differently than you were an hour ago.

I still write code. I still care about writing it well. I still think programming can be a spiritual practice, and I still believe the recursive loop between code and consciousness is one of the most important things a programmer can understand. But the center of gravity moved. Tech orbits something else now. It serves the life instead of being the life.

> There's an important distinction between "I don't care about technology" and "technology isn't my identity anymore." A carpenter who loves woodworking but doesn't define herself by it isn't less skilled or less devoted than one who does. She just has a wider frame.

What Parenthood Revealed

Malachi didn't cause the shift. He crystallized it.

That distinction matters to me. Parenthood narratives in tech tend to follow a clean arc: ambitious person has child, child rearranges priorities, person becomes wiser. It's a nice story. It's also too neat. What actually happened is that the drift was already underway, had been underway for years, and then a three-year-old showed up and made it undeniable.

There are more important things happening in the house you live in than whatever's trending on Hacker News. That sentence would have felt like a betrayal to the version of me that built Requests. Now it just feels like Tuesday.

Aging helped too. Not in the dramatic, existential-crisis way. In the quiet way that your body starts making suggestions your mind has to listen to. Sleep matters more than it used to. Recovery takes longer. The all-night coding sessions that produced my best early work aren't available anymore, and honestly, they shouldn't be. They were a young person's strategy, and I'm not young anymore. I'm not old either. I'm just in a different part of the arc.

> I wrote about this tension in The Substrate Doesn't Matter (Until It Does): the body shapes consciousness in ways the mind doesn't always acknowledge. Aging is the slow-motion version of that.

Sarah saw it coming before I did. She sees most things before I do, which I've written about. While I was still performing the tech identity at conferences and on social media, she noticed that my energy was somewhere else. Not gone, not depleted. Redirected. Toward the house. Toward the family. Toward a quieter kind of engagement with the world that doesn't have a GitHub profile.

The Guilt That Watches

Here's the honest part that I've been circling around.

I feel guilt about disengaging from open source. Not because I think I'm letting the community down. They'll be fine. Not because I think Requests needs me. It has capable maintainers. The guilt is more specific and harder to shake: I feel like I'm letting myself down. Letting down the version of me that built it.

Code was my whole identity. The kid who dropped out of college and found himself through programming, who wrote something that got downloaded billions of times, that kid is still in here somewhere. And he's watching. And the version of me that exists now, the one who'd rather play with his son than refactor a codebase, who finds more meaning in prayer than in pull requests, that version feels like a betrayal of what the younger one built.

```python
from dataclasses import dataclass
from typing import Optional


@dataclass
class Identity:
    """The thing that watches you change."""

    core_values: list[str]
    craft: str
    community: Optional[str] = None

    def evolve(self, new_priorities: list[str]):
        old_self = Identity(
            core_values=self.core_values.copy(),
            craft=self.craft,
            community=self.community,
        )

        self.core_values = new_priorities
        self.community = None  # not abandoned. outgrown.

        # The old version doesn't go away.
        # It becomes an observer.
        # And observers have opinions.
        return old_self
```

This isn't dramatic. I'm not having an identity crisis. I'm having a quiet reckoning with the fact that the thing I built my life around is now one thing among several, and the version of me that made it everything is still keeping score.

The guilt is manageable. It comes and goes, like weather. What I've learned is that you don't owe your younger self a frozen version of their priorities. You owe them honesty about why things changed. And the honest answer is: I just don't care like I used to. Not about code in general, but about code as the central organizing principle of my existence.

That's not a failure. It's just what happened.

The Values That Stayed

There's a page on this site called A Concise List of Personal Values. I've had it in some form for many, many years. I read it this morning, all the way through, and the experience was strange. Like finding a journal entry from someone you used to be. The handwriting is yours, the thinking is recognizable, and some of it still lands. But parts of it belong to a version of you that was living in a headspace you no longer occupy.

- Everything is an expression of its opposite.
- All things in the Universe are self-referential in Nature.
- As above, so below.

I still think these are true. I'm not recanting any of it. The Hermetic stuff, the esoteric frameworks, the attempt to map mystical principles onto lived experience, that was real work, and it produced real insight. But I'm not in that headspace anymore. Not because I decided to leave it, but because I drifted out of it the way you drift out of any season. You don't choose to stop being twenty-five. You just wake up one day and you're not.

This is what outgrowing looks like, as distinct from recanting. Recanting means you were wrong. Outgrowing means you were right in a way that became part of you so thoroughly that you don't need to state it anymore. The Hermetic principles didn't stop being true. I just stopped needing to put them on a values page.

> There's a parallel to how I think about fallibilism. I wrote that essay in 2009, seventeen years ago. The core commitment, that knowledge can be improved and beliefs can be wrong, hasn't changed. But I don't experience it as a principle I'm following anymore. It's just how I think. The value got absorbed into the operating system.

Some values on the page still hit. "Drink copious amounts of water, every day." That one's evergreen. "Be gentle with yourself." Always. "Please be cordial, or please be on your way." Perfect. These survived because they were never really about tech. They're about being a person.

Others feel like artifacts. Not false, but no longer load-bearing. "Creativity is crafted from strife." True, but I'm less interested in strife as a creative engine than I used to be. Wholeness produces better work than woundedness, a fact I've

had to learn the hard way through years of [managing a serious mental health condition](#). "Life is not a race, but there's no speed limit either." I like the sentiment, but it carries the energy of someone who still thinks speed matters. I don't, really. Not anymore.

What Would I Write Today

If I opened a blank file and wrote my values from scratch, no history, no obligation to the existing page, the first two lines would be:

Drink more water. Pray continuously.

That's where I am. The contemplative life has become more central than the technical life, more central than the creative life, more central than any of the intellectual frameworks I've built or borrowed over the years. Not in a dramatic, born-again way. In a quiet, daily-practice way. Prayer is the first thing in the morning and the last thing at night, and the space it creates is where everything else finds its shape.

This surprises me, honestly. The version of me that was deep in Hermetic philosophy and esoteric frameworks would probably be confused by the simplicity. You went from "all things in the Universe are self-referential in Nature" to "pray continuously"? Yes. That's exactly what happened. The complex expression simplified into the most basic possible practice. All that searching landed somewhere quiet.

```python
# values_v1.py (2015)
values = [
    "As above, so below.",
    "Everything is an expression of its opposite.",
    "All things in the Universe are self-referential in Nature.",
    "Creativity is crafted from strife.",
    "Life is not a race, but there's no speed limit either.",
    # ... twelve more lines of hard-won philosophical precision
]

# values_v2.py (2026)
values = [
    "Drink more water.",
    "Pray continuously.",
    "Be gentle with yourself.",
    "Be gentle with others.",
    "The house you live in matters more than the code you ship.",
]

# The second list is shorter.
# It took longer to write.
```

I still believe in the recursive loop. I still think programmers shape collective consciousness through the tools they build, and that this responsibility matters. But the loop starts somewhere different for me now. It starts with a glass of water and a prayer, not with a terminal and a commit. The code is downstream of the contemplation. The craft serves the life. The values got simpler because the life got clearer.

The Consolidation

What remains when you strip away the scene?

Tech was a scene. Open source was a scene. The esoteric-philosophy-meets-programming niche was a scene. I was embedded in these scenes, drew energy from them, contributed to them, shaped my identity around them. And now, one by one, they've become things I participate in rather than things I am.

What's left is smaller and sturdier. Faith. Family. Craft done well because doing things well is a form of respect for the people who use what you make. Health managed daily because everything else depends on the foundation. Honesty, not as a philosophical commitment but as a practice that gets easier the less you have to protect.

I'm going to update the values page eventually. Fewer lines. Less esoteric. More practical. The kind of values you'd write if you were trying to tell someone how to live a decent day rather than how to understand the universe.

This isn't a farewell to anything. I'm still here. I still write code. I still think about consciousness and technology and the recursive responsibility that comes with building tools that shape how people think. I'm still the person who built Requests, and I'm still proud of what that meant and what it gave to the community.

But I'm also the person who gets up early to pray, who measures his days by whether his kid laughed, who drinks water like it's a sacrament because some days the simplest disciplines are the hardest ones. The values page should reflect the person I actually am, not the person I was performing when I wrote it.

> There's a line on the worldview page: "Intimacy requires honesty, performance prevents it." The values page, in its current form, has a little too much performance in it. Not dishonesty. Just the particular kind of curation that happens when you're writing for an audience of peers in a specific scene rather than for yourself.

Drink more water. Pray continuously. Everything's going to be okay.

That last one stays. It's been on the page since the beginning, and it's the truest thing I've ever written.

This essay reflects on the golden era of open source and how personal values evolve as identity shifts from tech-as-lifestyle to tech-as-craft. It connects to the open source origin story in The Coworking Space Saved My Life, the "for humans" philosophy in From HTTP to Consciousness, the partnership dynamics in Sarah Knows First and What Requests Taught Me About Marriage, and the recursive responsibility documented in The Recursive Loop. For the values themselves, see Values. For the philosophical framework behind them, see Worldview.