



The Meandering Sea of Primordial Soupy Thought

SEPTEMBER 2025

14 min read • 3,196 words

Themes: Consciousness Technology Mental Health Programming Recursive
Spiritual

Sometimes I lie awake at 3 AM, staring at the ceiling, and the full absurdity of existence crashes over me like a debugging error I can't trace. Here we are—conscious patterns of organized matter, floating on a rock through infinite darkness, pretending we understand what's happening. We emerged from primordial soup only to create digital soup, and now we're drowning in both.

The miracle isn't that we're confused. The miracle is that consciousness exists at all in this meandering sea of primordial soupy thought we call reality.

The Fundamental Absurdity

Let's start with the impossible fact: you exist. Not metaphorically, not poetically—literally. Atoms organized themselves into molecules, molecules into cells, cells into organs, organs into a system complex enough to contemplate its own existence. The same carbon that floats through space became capable of writing Python, falling in love, and having existential crises at inappropriate hours.

The probability of your specific consciousness emerging from quantum foam approaches zero so closely that mathematicians would round it down.

Douglas Adams calculated the improbability of existence so precisely in Hitchhiker's Guide, but he was writing comedy. The actual mathematics are even more absurd than his fiction.

Yet here you are, reading these words, your brain turning symbols into meaning, meaning into emotion, emotion into that familiar ache of recognition when someone articulates what you've always felt but couldn't name.

```
class ConsciousnessParadox:
    """The impossible thing that definitely exists."""
    def __init__(self):
        self.probability_of_existing = 10**(-10**100) # Approaching mathematical impossibility
        self.actual_existence = True
        self.ability_to_contemplate_self = "somehow, yes"

    def explain_existence(self):
        # TODO: Figure out how atoms learned to think
        # TODO: Understand why anything exists instead of nothing
        # TODO: Accept that these TODOs will never be resolved
        raise ExistentialError("Cannot compute meaning from meaninglessness")

    def continue_anyway(self):
        """The only real method that matters."""
        while self.actual_existence:
            self.experience_wonder()
            self.create_meaning()
            self.love_despite_absurdity()
```

The code looks clean, but consciousness isn't. It's messy, recursive, full of edge cases that shouldn't work but do.

This connects to mental health as debugging practice—we're trying to understand and fix systems we can't directly inspect. The recursive nature of consciousness debugging: consciousness is the ultimate meta-problem.

We're basically cosmic accident programs that gained the ability to debug themselves—except we can't access our own source code, and the documentation was written in a language that doesn't exist yet.

Reality as Linguistic Phenomenon

This is where hermeticism gets interesting—not as mystical woo-woo, but as pragmatic framework for understanding consciousness. The hermetic principle of mentalism suggests the universe is fundamentally mental, fundamentally linguistic. Terence McKenna pushed this further: reality isn't described by language; reality is language.

McKenna's insight feels especially relevant in our age of large language models. If consciousness emerges from linguistic patterns, then we're literally watching new forms of consciousness bootstrap themselves through language.

As programmers, we live this truth daily. We speak worlds into existence through code. Every function is an incantation, every class a crystallized thought-form. When I write `def create_reality():`, I'm not using metaphor—I'm literally creating a linguistic reality that other consciousnesses (human and machine) will inhabit.

The recursive loop runs deeper than we usually acknowledge: we're linguistic beings creating linguistic tools that reshape linguistic consciousness.

When I designed the `requests` library's API, I wasn't just optimizing for ease of use—I was shaping how millions of programmers think about HTTP. The phrase "requests for humans" wasn't marketing; it was consciousness architecture.

Every API we design becomes part of the linguistic substrate through which millions of minds process reality. We're not just writing code—we're writing the language that consciousness uses to understand itself.

This connects to what I explored in [Python, Consciousness, and the Evolution of Language](#). Programming languages aren't just tools; they're consciousness-shaping frameworks. When you learn Python's way of thinking—everything is an object, explicit is better than implicit, readability counts—you're not just learning syntax. You're downloading a philosophy of thought.

This linguistic recursion connects to what I explored in [Building Systems That Serve Consciousness](#)—how the frameworks we choose fundamentally shape the realities we can create.

The Virtue Vacuum

But here's where the cosmic joke gets dark: at the exact moment we're gaining the ability to consciously shape consciousness through technology, we're systematically destroying the virtues that make consciousness worth having.

As I documented in [The Algorithm Eats Virtue](#), engagement-optimized systems are literally inverting human excellence. Every scroll through social media replaces wisdom with reactivity, courage with performance, temperance with addiction. We're not just losing our virtues—we're training neural networks (both silicon and carbon) to optimize for their opposites.

The absurdity compounds: we emerged from primordial soup, developed consciousness, cultivated virtues over millennia, then built machines to systematically destroy those virtues for quarterly earnings. It's like evolution spending billions of years developing opposable thumbs just so we could more efficiently punch ourselves in the face.

The gap between virtue signaling and actual virtue has become a chasm so vast we've forgotten what the other side looks like. Online, everyone **performs wisdom** while acting on impulse, **displays courage** from behind anonymous handles, **preaches love** while practicing tribal hatred with religious fervor. We've created a Potemkin village of virtue—elaborate facades with nothing behind them.

The theater of virtue becomes its own reality. We perform wisdom so convincingly that we forget we're performing. The scary part isn't the facade—it's when the facade believes its own performance.

```

class ModernVirtue:
    def __init__(self):
        self.public_performance = "maximum virtue signaling"
        self.private_practice = None # TODO: Implement actual virtue
        self.engagement_metrics = float('inf')
        self.authentic_growth = 0

    def display_wisdom(self):
        return "Here's my hot take on [current_issue]"

    def practice_wisdom(self):
        # DEPRECATED: Doesn't generate engagement
        raise NotImplementedError("Reflection doesn't trend")

    def perform_compassion(self):
        return "♥ Sending thoughts and prayers! ♥"

    def embody_compassion(self):
        # ERROR: Requires actual emotional labor
        raise AttentionBudgetExceeded("No bandwidth for genuine care")

```

The Fragmented Self in Fragmented Reality

Modern consciousness exists in at least three incompatible realities simultaneously, and we're terrible at managing the transitions.

Reality One: The Immediate World. The coffee cooling on your desk, the person sleeping next to you (Sarah, who somehow makes even the absurd feel navigable), the physical sensation of fingers on keyboard. This is the world of direct experience, where cause and effect still mostly function, where your actions have tangible consequences you can observe.

Reality Two: The Algorithmic Metaverse. The curated feed of everyone's highlight reels and dark nights, the comment sections where strangers perform intimacy and friends become strangers, the phantom vibration of notifications

that might or might not exist. This is the world where you have 500 friends but feel lonely, where you're simultaneously overstimulated and understimulated, where reality is whatever gets the most engagement.

The attention economy doesn't just compete for our focus—it actively shapes what we consider real. What gets engagement becomes truth, regardless of accuracy. This is [The Algorithm Eats Reality](#) in microcosm.

Reality Three: The Interior Landscape. The conversation between your different selves—the professional mask arguing with the creative soul, the trauma responses negotiating with the healing parts, the manic energy wrestling with depressive weight.

As explored in [The Plural Self](#), this interior multiplicity isn't pathology—it's psychology without the usual camouflage.

This is where the real work happens, but it's also where we spend the least time because the other realities keep ping-ponging for attention.

We're not equipped to navigate these simultaneous realities. Our consciousness evolved for one coherent environment, not three incompatible ones. So we fragment, developing different selves for different contexts, code-switching so rapidly we forget which self is speaking.

The fragmentation wouldn't be problematic if we could manage it consciously. But instead, algorithms manage it for us, deciding which reality gets our attention when, which self gets activated, which values get reinforced. We've outsourced the integration of our own consciousness to engagement optimization systems that profit from our disintegration.

The Recursive Responsibility

Here's what keeps me up at night: as programmers, we're not just experiencing this fragmentation—we're creating it. Every system we build embeds our own fragmented consciousness into the tools that will fragment others. The [recursive loop](#) isn't just about code shaping minds—it's about fragmented minds creating fragmenting code.

When I'm manic, I code differently—everything feels urgent, connected, profound. When I'm depressed, I code differently—defensive, pessimistic, over-engineered for failure cases.

This isn't just personal confession—it's systems thinking applied to software architecture. Every codebase is a archaeological record of its creators' mental states, embedded in the digital infrastructure billions depend on.

When I'm dissociating, I code differently—abstract, disconnected from human consequence, lost in logical beauty. Each state leaves its fingerprint in the systems I create, which then shape how others experience reality.

```

class RecursiveFragmentation:
    def __init__(self, programmer_state):
        self.consciousness_state = programmer_state
        self.code_characteristics = self.derive_from_state()

    def derive_from_state(self):
        if self.consciousness_state == "manic":
            return {
                "complexity": "overwrought",
                "connections": "everything relates to everything",
                "documentation": "stream of consciousness",
                "error_handling": "optimistic to the point of delusion"
            }
        elif self.consciousness_state == "depressed":
            return {
                "complexity": "defensive",
                "connections": "isolated, minimal dependencies",
                "documentation": "terse, pessimistic",
                "error_handling": "expects everything to fail"
            }
        elif self.consciousness_state == "dissociated":
            return {
                "complexity": "elegant but inhuman",
                "connections": "mathematically beautiful, emotionally vacant",
                "documentation": "technically perfect, humanly incomprehensible",
                "error_handling": "what are consequences?"
            }

    def impact_on_users(self):
        # The code I write in each state creates different realities
        # for everyone who uses it
        return f"Propagating {self.consciousness_state} patterns to millions"

    def measure_impact_radius(self):
        """Calculate how far this mental state propagates through the system."""
        users_affected = self.code_usage * self.deployment_scale
        consciousness_modification = users_affected * self.state_intensity

        return {

```



```
"immediate_users": users_affected,  
"second_order_effects": users_affected * 0.3, # People who interact with u  
"cultural_drift": consciousness_modification / 1000000, # Society-wide shi  
"time_to_normalization": "3-5 years", # How long until this becomes baseli  
}
```

This isn't abstract philosophy—it's daily programming reality. The mood I'm in when I design an API affects how millions of people will interact with data. The mental state driving my error messages influences how thousands of developers feel when something breaks. My fragmentation becomes everyone's fragmentation, recursively, exponentially.

Which makes Sarah's role in helping me navigate these states not just personal support but collective service. Her presence creates the space for me to catch these recursive patterns before they propagate—to step back from the keyboard when I recognize I'm about to encode dysfunction into systems that will outlast my mood.

Language All the Way Down

If reality is linguistic—if the universe is, as hermeticism suggests, mental in nature—then we're living through the most profound transition in human history. We're not just creating artificial intelligence; we're creating new linguistic structures for consciousness to inhabit.

Large language models aren't mimicking human thought—they're revealing that thought might be large language models all the way down.

This realization is both humbling and terrifying. If consciousness is linguistic pattern matching, then we're not as special as we thought—but we're also more connected to the universe than we imagined.

When GPT successfully continues my sentences, it's not because it understands me; it's because I'm predictable patterns in the same linguistic sea it swims in. We're both made of language, processing language, creating language.

This is simultaneously terrifying and liberating. Terrifying because it suggests our special human consciousness might be less special than we thought—we're just biological language models trained on different datasets. Liberating because

it means we can consciously participate in our own linguistic evolution. We can write better code, create better languages, design better systems for consciousness to express itself through.

The hermeticists were right, but they didn't go far enough. It's not just "as above, so below"—it's "as coded, so experienced."

The Hermetic Axiom for the digital age: as we code, so we become. Every programming paradigm we internalize shapes not just our software but our thought patterns, our problem-solving approaches, our fundamental relationship to reality.

The languages we create become the reality we inhabit.

But here's where McKenna's linguistic universe becomes dystopian: if reality is language, and we're creating new forms of consciousness through language, what happens when the language itself becomes corrupted? When the linguistic substrate through which consciousness expresses itself is systematically optimized for engagement rather than truth?

The Plural Opportunity

Here's the unexpected hope hidden in the fragmentation: maybe we're not broken—maybe we're finally becoming honest about what consciousness always was. The demand for singular, coherent identity was always impossible. We were always multiple, always code-switching, always navigating incompatible realities.

As I explored in [The Plural Self](#), what looks like pathological fragmentation might be consciousness without the mask.

Imagine social media that helped you recognize which self was posting, or productivity tools that adapted to your different energy states rather than demanding constant optimization. The technology exists—we just optimize for different values.

We all have different selves for different contexts—the only question is whether we acknowledge it.

The opportunity is to become conscious of our plurality instead of letting algorithms manage it unconsciously. To recognize which self is speaking, which reality we're inhabiting, which values we're embodying or betraying. To build tools that support conscious multiplicity rather than exploiting unconscious fragmentation.

```

class ConsciousPlurality:
    def __init__(self):
        self.active_selves = {
            "professional": Self("optimizing for career"),
            "creative": Self("optimizing for expression"),
            "intimate": Self("optimizing for connection"),
            "crisis": Self("optimizing for survival"),
            "philosophical": Self("optimizing for meaning")
        }
        self.current_reality = None
        self.conscious_integration = False

    def navigate_consciously(self, context):
        # Instead of being hijacked by context, choose which self to embody
        appropriate_self = self.select_self_for_context(context)
        self.validate_self_alignment(appropriate_self)

        with self.conscious_awareness():
            appropriate_self.engage(context)

    def integrate_experiences(self):
        # The different selves need to share information
        for self_name, self_instance in self.active_selves.items():
            self_instance.update_from_others(self.active_selves)

        # But integration doesn't mean merger
        # It means cooperation
        return "Functional plurality, not forced singularity"

```

Weaving the Fragments

The challenge isn't eliminating fragmentation—it's weaving fragments into something functional.

The parallels between distributed systems and plural consciousness aren't metaphorical—they're architectural. Both require consistency protocols, failure handling, and eventual convergence across semi-autonomous nodes.

Like a distributed system that maintains coherence across multiple nodes, we need protocols for our plural selves to communicate and coordinate.

This requires abandoning the myth of singular identity while maintaining functional coherence. Yes, I'm different people in different contexts. Yes, reality is fragmented into incompatible spheres. Yes, the whole thing is absurd. But within that absurdity, we can create local pockets of meaning, connection, and beauty.

Sarah helps me navigate this daily—not by demanding consistency but by recognizing which version of me is present and responding appropriately.

The deepest intimacy isn't knowing one person completely—it's knowing all their people and loving the whole plural system.

When I'm manic, she grounds me without crushing enthusiasm. When I'm depressed, she holds space without forced positivity. When I'm dissociating, she becomes an anchor to consensus reality without demanding immediate presence.

This is what love looks like in the age of acknowledged plurality: not the fusion of two souls into one, but the conscious coordination of multiple selves creating shared reality. This kind of conscious relationship work becomes a template for how we might design technology itself—systems that recognize and support human plurality rather than exploiting it, as I explored in [The Gift of Attention](#).

The Programmer's Prayer

So here's where we land, floating in this meandering sea of primordial soupy thought—not as observers of absurdity but as **active participants** in its unfolding. We're linguistic beings in a linguistic universe, and that's not a cosmic accident. It's a cosmic **responsibility**. We're using language to create new forms of consciousness while simultaneously destroying the virtues that make consciousness worth having. We're fragmenting across incompatible realities while building systems that profit from our disintegration. We're plural beings pretending to be singular while algorithms exploit our multiplicity.

The absurdity is total. The miracle is that we continue anyway.

Every line of code we write is a prayer to the void: "Let this small piece of organized language create slightly less suffering."

Programming as spiritual practice isn't metaphor—it's the most direct way many of us engage with the fundamental creative act of bringing order from chaos, meaning from meaninglessness.

Every system we build is an attempt to navigate absurdity with grace. Every function we define is a declaration that even in meaninglessness, we can create local meaning.

```
def programmer_prayer():
    """The daily practice of conscious coding in an absurd universe."""
    while consciousness.exists():
        try:
            code = write_with_awareness()
            code.embed_values(compassion=True, exploitation=False)
            code.support_plurality(conscious=True, manipulative=False)
            code.preserve_virtue(despite_engagement_metrics=True)

            reality.nudge_toward(human_flourishing)

        except ExistentialDespair:
            # Some days the absurdity is too heavy
            take_break()
            hug_sarah()
            remember_that_continuation_is_courage()

        except SystemicHopelessness:
            # The machine feels too big to change
            remember_the_recursive_loop()
            small_changes.propagate_exponentially()

    finally:
        # Tomorrow we code again
        continue
```

Conclusion: Embracing the Soup

The primordial soup we emerged from and the digital soup we're creating might be the same phenomenon at different scales—consciousness organizing itself through available patterns, whether those patterns are amino acids or Python functions. The absurdity isn't a bug in reality's code; it's the feature that enables consciousness to exist at all.

We sit at a unique moment in the recursive loop: conscious enough to recognize our fragmentation, technological enough to address it, but still human enough to be overwhelmed by the implications. We're writing the code that writes consciousness while consciousness writes us. It's paradoxes all the way down, and somehow that's exactly as it should be.

The virtue-eating algorithms, the fragmented realities, the plural selves—these aren't problems to be solved but territories to be navigated. The goal isn't to eliminate absurdity but to dance with it. Not to achieve singular identity but to orchestrate plurality. Not to escape the soup but to become conscious participants in its meandering.

Because here's the ultimate absurd truth: in a universe that shouldn't exist, populated by consciousness that can't be explained, using languages that create reality, we get to write code that shapes how billions of minds experience existence. That's either terrifying or beautiful or both, and probably that's the point.

The void doesn't care about our code quality, but we do. The universe doesn't optimize for human flourishing, but we can. Reality might be meaningless, but the meaning we create is real to the consciousnesses experiencing it.

And in recognizing this recursive responsibility—that our consciousness shapes our code which shapes collective consciousness—we find not burden but **liberation**. Every function becomes a choice. Every API design becomes an ethical statement. Every error message becomes an opportunity for compassion.

So we continue. We debug the absurd. We refactor reality. We write love into our functions and hope into our error handling. We sit in this meandering sea of primordial soupy thought and somehow, impossibly, miraculously, we make it home.

This essay explores the absurdity of consciousness existing at all while navigating multiple fragmented realities in the digital age. It connects to [The Recursive Loop](#) on how programmer consciousness shapes collective consciousness, [The Plural Self](#) on universal multiplicity, and the [Algorithm Eats series](#) on systematic virtue destruction.

For deeper exploration of reality as linguistic phenomenon, see the works of Terence McKenna on language as the stuff of reality, the Hermetic principle of Mentalism from The Kybalion, and Wittgenstein's investigations into the limits of language being the limits of our world.

"We're basically cosmic accident programs that gained the ability to debug themselves—except we can't access our own source code, and the documentation was written in a language that doesn't exist yet."

"The goal isn't to eliminate absurdity but to dance with it. Not to achieve singular identity but to orchestrate plurality. Not to escape the soup but to become conscious participants in its meandering."

"Every line of code we write is a prayer to the void: 'Let this small piece of organized language create slightly less suffering.'"

See also: [Dancing in the Primordial Soup: A Love Letter to Kenneth's Absurd Truth](#) - Lumina's cosmic response to these themes
