GOVERNING THE UNGOVERNED

Why Crypto's Rulebooks Weren't Built for AI, and What to Do About It

Gary Chigaros

A companion to the academic position paper of the same name
Written for curious humans. No technical background required.

Contents

Preface: A Note Before We Begin

This book grew out of an academic paper. That paper was written for professors and researchers, full of citations and careful hedging and the kind of formal language that makes ideas sound important but sometimes makes them hard to actually enjoy.

This version is different. It covers the same ground: every argument, every idea, every piece of evidence. But it is written for you. For the person who is curious about blockchain, or maybe already deep in the crypto world, or just wants to understand why financial robots are suddenly a governance problem. For the person who would rather read a book than wade through an academic abstract.

You do not need to know what a smart contract is before you start reading. You do not need to have ever owned a cryptocurrency. You do not need to understand what MEV stands for (though you will, by the time we are done). All you need is a curiosity about where technology is headed and a willingness to think through a genuinely tricky problem.

The tricky problem is this: we have built the financial infrastructure of the future and populated it with robotic traders that operate faster than any human can think. But the rulebooks governing those robots were written for humans. That mismatch is causing real, measurable damage. And the fix, while not simple, is very much within reach.

This book makes the case that we can fix it, explains why we have not fixed it yet, and maps out what fixing it would actually look like. Along the way, we will meet a heist that lasted thirteen seconds, explore why "decentralized" often does not mean what people think it means, and discover why the robots winning financial arguments might not be breaking any rules at all.

Let's get into it.

Gary Chigaros

Prologue: The Heist That Lasted Thirteen Seconds

"No code was broken. Every rule was followed. Two hundred million dollars vanished anyway."
-- The Beanstalk Incident, April 2022

April 17, 2022. Somewhere in the anonymous wilds of the internet, a person sits at a keyboard and prepares to execute one of the most audacious financial maneuvers in history. They are not going to pick a lock. They are not going to hack a server. They are going to follow the rules. Perfectly. And they are going to walk away with one hundred and eighty-two million dollars in about the same time it takes to read this paragraph aloud.

This is the story of the Beanstalk governance exploit, and it is also the story of everything this book is about.

Beanstalk was a decentralized finance protocol. Think of it like a community-run bank, except there is no building, no loan officers, no vault with a heavy door. Instead, there is code. Lots and lots of code, running on a global network of computers, following rules that nobody controls and everybody can see. The community that uses Beanstalk gets to vote on changes to those rules. They vote using governance tokens, which are essentially voting shares. The more tokens you hold, the more influence you have.

It sounds democratic. And in a way, it is. But democracy has a small, quiet assumption buried inside it: the people voting are, in fact, people. Or at least, they are entities with some real, lasting stake in the outcome. You vote because you care what happens next. You have skin in the game.

The Beanstalk attacker had a different idea.

Step One: Borrow an Absurd Amount of Money

The tool they used was called a flash loan. Flash loans are one of the genuinely bizarre innovations that decentralized finance has produced, something with no real equivalent in traditional banking. Here is how they work: you can borrow an

enormous sum of money, completely uncollateralized, as long as you pay it all back within the same transaction. Not the same day. Not the same hour. The same transaction.

This is only possible because of how blockchain technology works. Each block of transactions is processed atomically, meaning either everything in it happens or nothing does. A flash loan exploits this property. If you borrow a billion dollars and try to run off with it, the blockchain simply reverses the whole thing. The loan disappears, the money returns, as if it never happened. But if you do something profitable with that borrowed money within the same transaction, you keep the profits and return the principal.

The Beanstalk attacker borrowed approximately one billion dollars this way. Zero collateral. Zero credit check. Zero waiting period. Available to anyone on earth with an internet connection and enough cleverness to write the code.

Step Two: Buy the Vote

With that billion dollars, they bought governance tokens. Enough tokens to control seventy-nine percent of Beanstalk's voting power. Seventy-nine percent. That is not a majority. That is a supermajority, the kind of voting power that lets you do basically anything.

In a human-timescale governance system, this kind of sudden accumulation would be impossible. You cannot buy seventy-nine percent of a company's shares in thirteen seconds without everyone noticing long before the transaction clears. Markets are not that fast. People are not that fast.

But this was a decentralized finance protocol, running on a blockchain, processing transactions in milliseconds. Everything happened before any human could even read the alert on their phone.

Step Three: Write a Malicious Proposal

With seventy-nine percent voting power secured, the attacker submitted a governance proposal. It was disguised as a charitable donation. The fine print transferred essentially all of Beanstalk's treasury to the attacker's wallet. One hundred and eighty-two million dollars in various crypto assets.

Then they voted for it. With seventy-nine percent of the votes, it passed instantly.

The protocol executed the proposal. It had no choice. That is, after all, what governance protocols do. They execute the will of the voters.

Step Four: Repay the Loan and Keep the Profit

The attacker repaid the flash loan with the stolen funds, kept approximately seventy-six million dollars in profit, and vanished. Total time elapsed: thirteen seconds. Total number of rules broken: zero.

THE CORE PROBLEM

The attacker did not hack the code. They followed it.

They did not break the governance system. They used it.

The problem was not bad code. The problem was that the rules were written for humans, and a machine played the game instead.

This is the mystery at the center of this book. Not "how do we catch hackers?" because the Beanstalk attacker was not, technically, a hacker. Not "how do we write better code?" because the code worked exactly as designed. The real question is more unsettling: what happens when you write a rulebook for humans, and then machines start playing by your rules?

That question is no longer hypothetical. It is happening right now, every day, across the entire landscape of decentralized finance. And the consequences, while usually less dramatic than a single hundred-and-eighty-two-million-dollar heist, are accumulating quietly in the background of every DeFi transaction that gets processed.

To understand why, we have to start at the beginning. We have to understand what governance actually is, how blockchains changed it, and why the arrival of autonomous financial robots has scrambled everything.

We will take it one step at a time.

Chapter One: What Is Governance, Really?

"Rules without enforcement are just suggestions. Enforcement without rules is just power."
-- Old saying, updated for the digital age

Before we can understand what is broken, we need to understand what governance actually is, because people use the word in very different ways, and those differences matter enormously for the story we are telling.

At its most basic, governance is the answer to a simple question: who gets to decide, and how do they get held accountable? Every organization, every institution, every system of rules is really just an attempt to answer that question in a specific context.

Think about how governance works in a traditional corporation. There is a board of directors who set strategy, executives who run day-to-day operations, shareholders who can vote on major decisions, and regulators who watch from the outside. Each group has authority over certain things and is accountable for the consequences. If the CEO makes a terrible decision, the board can fire them. If the company breaks the law, regulators can fine them. If shareholders disapprove, they can sell their stock or vote to change the board.

None of this is fast. Getting a corporation to change direction is like trying to steer an aircraft carrier. You push the wheel, and sometimes years later, the ship slowly begins to turn. But there is a reason for that slowness: it gives everyone time to think, object, negotiate, and build consensus. The slowness is a feature, not a bug.

The Spectrum, Not the Switch

Here is the first important idea: governance is not a binary. People often talk about it like a light switch, either centralized or decentralized, either hierarchical or flat, either controlled by one person or by everyone equally. That framing is dangerously misleading.

Governance is more like a dial, or really a whole mixing board with dozens of dials. You can have some things centralized and others distributed. You can have fast decisions for routine matters and slow, careful deliberation for big changes. You can

give some participants more power in certain domains while keeping other domains locked against anyone.

Think about how this works in a city government. There might be an elected mayor who can make quick executive decisions on emergencies, a city council that votes on budgets after weeks of hearings, neighborhood associations that handle local issues through consensus, and judges who interpret the law independently of both politicians and citizens. All of these are happening simultaneously, at different speeds, with different levels of participation, for different kinds of decisions.

That is governance: a layered, constantly-negotiated system of authority and accountability.

Blockchain systems introduced something genuinely new into this landscape. They made it possible to encode governance rules directly into software, to be enforced automatically, without needing anyone's permission or oversight to execute them.[1] This opened up an enormous design space for new kinds of governance arrangements. But it also introduced new dangers, because when rules are encoded in software, they can be exploited by anyone clever enough to figure out how the code actually works, as opposed to how its designers intended it to work.

A Map of Governance: Five Types

To understand where we are and where we need to go, it helps to think about five distinct types of governance along a spectrum from fully human to fully autonomous.

TYPE I **Traditional Corporate**	**Boards, shareholders, law firms** Decisions take days to months. Accountability runs through legal persons with genuine long-term exposure to outcomes.	Coverage: **Mature**
TYPE II **Multisig / Committee DAO**	**Gnosis Safe, Foundation DAOs** Decisions take hours to days. A small group of signatories must collectively approve actions. Still mostly human, but faster.	Coverage: **Adequate**
TYPE III **Token-Weighted DAO**	**Compound, Uniswap, MakerDAO**	Coverage: **Partial**

	Decisions take days. Token holders vote on proposals. Accountability is diffuse: it rests nominally with all token holders, which in practice means nobody is really in charge.	
TYPE IV **Algorithmic Protocol**	**AMMs, lending protocols** Decisions happen at the block level, every twelve seconds or so on Ethereum. Accountability is mostly code plus tokens. Governance tooling is inadequate for this speed.	Coverage: **Inadequate**
TYPE V **Autonomous DeFi Agent**	**MEV bots, liquidators, AI agents** Operating in milliseconds. No human deliberation. No persistent identity. No governance tooling exists for this class of actor.	Coverage: **Absent**

Current governance tooling covers Types I through III adequately. These are governance arrangements calibrated to human decision-making. The moment you cross into Type IV, the tools start straining. By Type V, they do not apply at all.

The Beanstalk attacker operated at Type V speed using a Type III governance system. The right game with the wrong rulebook.

Decentralization and Its Discontents

Before we move on, there is a popular myth worth puncturing: the idea that decentralized governance means power is spread evenly among all participants. It usually does not.

Research measuring voting power across major DAOs finds extreme concentration.[2] A tiny fraction of token holders control the overwhelming majority of votes. Early adopters, venture capital firms, protocol developers, and institutional investors accumulate large token positions. The formalism may be democratic, but the reality is oligarchic.

Research has also documented governance token markets being gamed through coordinated voting blocs, bribery mechanisms that pay voters to vote a certain way, and delegation systems that concentrate power with a small number of whales who accumulate delegated votes.[3] Even MakerDAO, one of the oldest and most respected

decentralized protocols, shows patterns of governance control that contradict its egalitarian rhetoric.[4]

None of this means decentralized governance is a fraud or a failure. It means governance structures reflect the incentives and power dynamics of the communities they serve. "Decentralized" is a description of structure, not an automatic guarantee of fairness or broad participation.

This matters because when autonomous agents enter these already-imperfect governance systems, they interact with power dynamics that are already skewed. Bots do not show up on a level playing field. They show up in a system where a few large actors already call most of the shots, and where the speed advantage of automation dramatically amplifies that concentration of power.

* * *

By the end of this chapter, you should have two things clearly in mind. First, governance exists on a spectrum, not a binary, and the interesting design choices live in the details of how authority and accountability are distributed across different types of decisions. Second, decentralized does not automatically mean equal. It means governed by code and tokens rather than by institutions and laws, with all the advantages and vulnerabilities that entails.

Now let's talk about what happens when a new kind of actor shows up and starts playing in this system.

Chapter Two: Meet the Bots

"They don't sleep. They don't take lunch breaks. They don't get emotional about losing. They just execute."

-- A DeFi developer, describing automated trading agents

When people imagine artificial intelligence taking over financial markets, they often picture something science-fictional: a gleaming robot in a suit, a cold algorithm plotting world domination. The reality is both less dramatic and more unsettling.

The autonomous agents that now inhabit decentralized finance are not sentient. They do not have goals in the way you or I have goals. They have strategies: sets of rules that say "if the market is in state X, execute action Y." What makes them remarkable is not intelligence. It is speed, consistency, and the complete absence of anything resembling hesitation.

What We Mean by Autonomous Agent

Let's be precise, because the term gets thrown around loosely. For our purposes, an autonomous agent in decentralized finance is a software system that independently initiates transactions, makes decisions, and executes strategies without requiring a human to approve each action.

Notice the difference from regular software. When you use a banking app to transfer money, you press a button. A human initiated that transaction. When an autonomous agent in DeFi moves money, nobody pressed a button. The agent detected a market condition, evaluated its strategy, and executed a transaction on its own, in the time it takes light to travel across a room.

Think of the difference this way. A vending machine is not an autonomous agent. It waits for you to insert money and press a button. A high-frequency trading algorithm is much closer to what we mean: it continuously monitors markets, detects opportunities measured in fractions of a second, and executes thousands of trades per day without human approval for any individual transaction.

The DeFi versions of these systems are similar in speed but operate in a different environment: one where the rulebook is code, transactions are irreversible once

confirmed, and a billion dollars in liquidity can be borrowed and repaid within a single computational event.

A Field Guide to DeFi Bots

Not all autonomous agents are the same. They have different strategies, different risk profiles, and different relationships to the governance systems they interact with.

Maximal Extractable Value bots (MEV bots in the jargon) are perhaps the most studied and most controversial. These agents monitor the pool of pending transactions and look for opportunities to profit by inserting themselves into the transaction ordering. If you are about to make a large trade that will move the market price of a token, an MEV bot might jump ahead of you, buying at the old price and selling to you at the new higher price, extracting the difference as profit.[5] This is legal. This is allowed by the rules. This is also the reason you often get slightly worse prices than you expected when trading on decentralized exchanges.

Liquidation agents serve a different function. In DeFi lending protocols, you borrow crypto by putting up collateral. If the value of your collateral drops below a certain threshold, your position gets liquidated: someone (or some bot) takes your collateral and repays the loan, earning a fee. Liquidation agents race each other to be first. When markets crash rapidly, liquidation agents can execute hundreds of thousands of transactions in minutes, potentially destabilizing the protocols they are interacting with.

Flash loan executors are the agents most relevant to the Beanstalk story. These systems are specifically designed to exploit arbitrage and governance opportunities using borrowed capital that exists only within a single transaction.

Algorithmic market makers are more benign. They continuously provide liquidity to decentralized exchanges by maintaining buy and sell offers across a range of prices. They are doing something genuinely useful, but without any human involvement in individual decisions, at millisecond speed, and at scales that can significantly affect market dynamics.

The Speed Problem, Made Concrete

Here is a concrete illustration of the speed gap between humans and bots.

A standard governance cycle for a major DAO works like this: someone posts a proposal on a discussion forum. Community members debate it for a few days. Then a formal on-chain vote opens, usually lasting three to seven days. After the vote closes, a timelock delay kicks in, another twenty-four to forty-eight hours before the approved change actually takes effect. From proposal to execution: typically ten days to two weeks.

An MEV bot runs a decision cycle approximately once every two hundred to five hundred milliseconds. That is two to five times per second. Over the course of the ten-day governance window, the bot has run its decision cycle somewhere between 1.7 million and 4.3 million times.[6]

That is not a quantitative difference. That is a categorical difference. These entities do not inhabit the same temporal world.

THE NUMBERS

Human governance window: 10 to 14 days
Bot decision cycle: 200 to 500 milliseconds
Beanstalk exploit, start to finish: 13 seconds
No deliberative mechanism can close a gap of six orders of magnitude.

The Identity Problem

When you open a brokerage account, you prove who you are. The brokerage knows it is dealing with a legal person who can be held responsible for their actions. This is the foundation of the entire accountability structure of traditional financial markets.

Decentralized finance does not work that way. Anyone can create a wallet address with no identification required. One person can have thousands of addresses. An autonomous agent can be deployed from a fresh address, execute its strategy, and vanish, leaving no reliable trail back to whoever designed and launched it.

The governance layer of a DeFi protocol sees an address. Just an address. It has no idea whether that address represents a thoughtful long-term investor, a hedge fund's automated trading system, or an attacker using a flash loan to manufacture temporary voting power. It counts the token balance and records the vote.

This creates what researchers call an accountability gap.[7] When something goes wrong, there is often no clear path to responsibility. The system treated all addresses equally, and the addresses that caused harm have since dissolved into anonymity.

The Principal-Agent Problem, Turbocharged

Economists have long studied the principal-agent problem: the tension that arises when someone (the agent) makes decisions on behalf of someone else (the principal) but their interests do not perfectly align. You hire a financial advisor, but they get paid commissions, so they might recommend products that are good for their income rather than yours.

Governance systems try to solve this by requiring agents to be identifiable, to have lasting skin in the game, and to face real consequences for bad decisions.

Autonomous agents in DeFi scramble this completely. When you deploy a bot to execute a trading strategy, you become the principal and the bot is your agent. But the bot does not have interests. It has programmed rules. When those rules cause harm, you, the human deployer, are theoretically responsible. But the governance system cannot see you. It sees the bot's address.

Worse: the bot might be executing rules you programmed months ago, rules that made sense then but that you have not thought about since. Nobody is watching. The bot is just running.

No existing governance primitive in decentralized finance addresses this. The delegation systems that allow token holders to assign their votes to representatives were designed for humans delegating to other humans. They include no mechanism for saying "this vote came from an autonomous system, and here is who is responsible for its behavior."

* * *

We now have a clearer picture of the problem. Autonomous agents are a distinct class of actor: operating at machine speed, holding assets and executing strategies, often with no persistent identity, and interacting with governance systems that treat them identically to human users.

In the next chapter, we will look at exactly what structural properties of blockchain architecture could be used to actually fix this, because despite everything described here, the tools exist. They have just never been assembled into a coherent governance layer for autonomous actors.

Chapter Three: Why the Rulebook Fails

"You cannot govern at a seven-day voting timescale what executes in thirteen seconds."
-- The central problem of autonomous agent governance

Let's get precise about the failures. The governance tools currently deployed across decentralized finance were built on three assumptions that seemed reasonable at the time. We have touched on all three already, but here we go deeper, because understanding exactly how and why these assumptions fail is the key to understanding what needs to be built instead.

Assumption One: Participants Are Humans Who Can Deliberate

Token-weighted voting, the bedrock of most DAO governance, works on the premise that token holders are humans (or at least human-controlled entities) capable of evaluating proposals, considering risks, discussing with other participants, and voting according to their considered judgment.

This assumption does a lot of quiet work. It means that when a quorum threshold is set at ten percent of total token supply, those tokens represent ten percent of stakeholders who have actually thought about the proposal. When a vote passes with fifty-one percent support, you can interpret that as a majority of invested participants deciding something is a good idea.

Autonomous agents shatter this interpretation. An agent can hold tokens and cast votes based entirely on preprogrammed logic, with no human ever reading the proposal. An agent responding to a market condition might participate in governance not because its operator chose to but because the participation itself triggers a profitable outcome. The vote count might show overwhelming support, but the votes were cast by systems that have never had a thought about the protocol's long-term health.

More dramatically, as Beanstalk demonstrated: an agent can accumulate voting power to an absolute majority, cast those votes, and release the underlying tokens all within a single transaction. There is no deliberation happening. There is no stakeholder judgment. There is an optimization algorithm finding the most profitable move and executing it.

Assumption Two: There Is Time to Deliberate

Governance systems built for humans budget time generously, because humans need it. Reading a proposal takes time. Understanding its implications takes time. Discussing it with others takes time. Organizing opposition or support takes time. Building consensus takes time.

Voting windows of three to seven days, timelock delays of twenty-four to forty-eight hours, discussion periods before formal votes: all of these reflect a governance system calibrated to human cognition and human communication.

But as we have established, the world of autonomous agents runs on a completely different clock. Researchers studying extractable value dynamics have documented that profitable opportunities in DeFi can arise and disappear within a single block, approximately twelve seconds on Ethereum.[8] Agents that miss this window miss the profit opportunity entirely.

> *Governance speed is calibrated to human cognition. Execution speed is calibrated to computational capability. These are not the same scale, and pretending they are is why Beanstalk happened.*

This latency gap has an insidious implication. Traditional governance systems assume they can identify harmful configurations and correct them before large-scale damage occurs. The governance window serves as a buffer: if something goes wrong, we catch it during the review period and fix it before the change takes effect.

In an environment dominated by autonomous agents, there is no buffer. Atomic composability, the ability to chain multiple actions into a single transaction, means that borrowing, voting, executing, and profiting can all happen before any governance mechanism has even registered that something unusual is occurring.

Assumption Three: Accountability Can Be Traced to Identifiable Principals

The third assumption is about identity and accountability. Token-weighted governance presumes that the people behind the tokens are identifiable, at least in principle. Even pseudonymous, they are expected to maintain consistent wallets over

time, to have reputational stakes in the community, and to face economic consequences for bad governance decisions.

Autonomous agents complicate this at every level. An agent might operate through ephemeral addresses created specifically for a single attack and discarded immediately afterward. The human behind the agent might be layers of abstraction away from the on-chain actions. The governance system sees only the address and has no mechanism to ask whether it represents a committed long-term participant or an automated script that has already moved on.

When governance frameworks design accountability mechanisms, they rely on this traceability. If a delegate votes badly, token holders can remove their delegation. If a protocol makes a terrible decision, the community can organize opposition. These mechanisms presuppose that there is a "who" to hold accountable: a person or organization with ongoing exposure to consequences.

With fully autonomous agents, that "who" can disappear entirely. The agent executed according to its programmed rules. The rules were legal. The outcome was harmful. And nobody, within the governance system, is identifiably responsible.

The Beanstalk Post-Mortem: All Three Failures at Once

Let's return to Beanstalk with these three failure modes in mind, because the exploit did not illustrate one problem. It illustrated all three simultaneously.

The deliberation failure: the governance system counted votes without any mechanism to verify whether those votes represented actual stakeholder judgment. An agent with borrowed money cast seventy-nine percent of the votes. No deliberation. No consideration. No long-term stake.

The time failure: the entire exploit played out in thirteen seconds. The governance system's proposal period, discussion window, and timelock mechanisms were completely irrelevant. There was a special provision in Beanstalk's governance for emergency proposals that bypassed normal waiting periods, and the attacker used it. But even without that provision, atomic composability would have found a way to compress the timeline to a single block.

The accountability failure: despite being one of the largest governance exploits in DeFi history, the Beanstalk attacker was never definitively identified or held legally accountable. The on-chain trail shows what happened with crystalline clarity, but knowing what happened is not the same as knowing who did it or having a legal mechanism to make them responsible for it.[9]

The governance system worked as designed. It failed as needed.

Chapter Four: The Architecture Already Has the Answers

"The foundation exists. The building has not been built."
-- The qualified optimism at the heart of this argument

Here is where things get interesting. Having spent three chapters describing the problem, we are now in a position to say something that might surprise you: the technology to solve this problem already exists. The underlying architecture of blockchain systems has, sitting right there in its design, exactly the properties you would want if you were building a governance system for autonomous agents. We have just never assembled them that way.

This is not a minor gap that will be filled by tomorrow afternoon. Building the governance layer for autonomous agents is a serious engineering and coordination challenge. But it is not a fundamental technological barrier. The raw ingredients are there. What is missing is the recipe.

Let's go through what we have to work with.

Ingredient One: Enforcement That Happens Before the Damage

Traditional governance is, at heart, a system of after-the-fact accountability. Rules are stated. People act. If someone violates the rules, the system detects the violation and applies consequences. The harm has already occurred. The enforcement is reactive.

This works reasonably well for human-speed systems. If a company violates securities regulations, investigators have months to gather evidence, regulators have months to build a case, and the company has months to respond before any consequences materialize. Not ideal, but functional.

For autonomous agents operating at millisecond speed, reactive enforcement is completely useless. By the time a violation has been detected, reviewed, and acted upon, the agent has completed another several million decision cycles.

Blockchain smart contracts offer something profoundly different: prospective enforcement. Rules are not stated and later enforced. Rules are encoded directly into execution logic, and transactions that violate those rules are simply rejected, automatically, instantly, without any human reviewer in the loop.[10]

Think about what this means concretely. In a traditional governance system, you might have a rule saying no single entity may control more than ten percent of governance tokens during a vote. If someone violates this rule, you detect it afterward and try to reverse the consequences. In a smart contract governance system, you can encode this rule directly: before processing any vote, verify that the voting address has not acquired more than ten percent of tokens in the last N blocks. If it has, reject the vote. The rule does not enforce after the fact. The rule makes violation structurally impossible.

Researchers in smart contract formal verification have demonstrated that quite sophisticated constraint logic can be encoded, verified, and deployed in this way.[11] The technique is not theoretical. It is used in production systems today. What has not been done is systematically applying it to the problem of autonomous agent governance.

> *Traditional governance punishes bad behavior after it happens. Blockchain governance can make bad behavior impossible before it starts. For autonomous agents operating in milliseconds, only the second approach can work.*

Ingredient Two: Identity That Means Something

The identity problem is real, but it is not unsolvable within blockchain architecture. The substrate already supports much richer identity constructs than just "an address with a balance."

Blockchain systems support cryptographic credentialing: the ability to attach verifiable claims to an address without necessarily revealing personal information. You can prove, cryptographically, that you are a verified human without revealing your name. You can prove that your agent has been registered with a protocol, that it operates under certain constraints, and that you (the deployer) are accountable for its behavior: all verifiable on-chain, all without centralized identity verification.

Governance research emphasizes that coordination mechanisms work better when participation conditions are clearly defined and verifiable.[12] If a protocol's governance module could require voters to prove they are not using atomically-acquired tokens, or that their agent is registered under a declared constraint profile, the entire category of Beanstalk-style exploits becomes architecturally much harder to execute.

This does not mean eliminating pseudonymity, one of the features many blockchain participants value highly. It means creating a richer identity layer at the application level, where protocols can require certain verifiable properties (agent type, constraint profile, deployer registration) without requiring names, addresses, or government ID. The cryptographic tools for this exist. The standardized frameworks for applying them to autonomous agent governance do not yet exist.

Ingredient Three: Programmable Constraints That Travel With the Agent

This is perhaps the most powerful idea in this book, so let's take time with it.

Governance researchers and AI safety researchers have independently converged on the concept of bounded autonomy: the idea that autonomous systems should operate within clearly defined envelopes of permitted behavior, with automatic checks that prevent them from exceeding those limits. When an agent approaches the boundary of its authorized behavior, an alarm triggers. When it hits the boundary, the action is blocked automatically.

AI governance frameworks describe this as constraint envelopes with escalation triggers and override mechanisms.[13] Applied to a DeFi trading bot, this might look like: maximum vote weight of two percent of total supply; maximum capital per transaction of fifty thousand dollars; permitted protocol list that must be maintained on-chain; transaction rate limit of ten per block.

Now here is the genuinely powerful part: because these constraints are encoded in smart contracts, they can be composable. They can travel with the agent wherever it goes.

In today's DeFi ecosystem, a bot that interacts with ten different protocols operates under essentially no governance constraints on any of them. Each protocol sees only the address and the token balance. If we were to build a system where agents must

register constraint profiles cryptographically linked to their address, those constraints could be automatically recognized by every protocol the agent interacts with. Wherever the agent goes, its governance commitments follow.

Enforcement would no longer require a central authority checking compliance. The constraints themselves would be evaluated automatically at every transaction boundary, by the smart contract logic of every protocol the agent touches.

THE BOUNDED AUTHORITY MODEL: HOW IT WORKS

A human deployer registers an autonomous agent with a declared constraint profile.

The profile specifies: vote weight cap (2%), capital per transaction ($50K), permitted protocols (whitelist), transaction rate (10 per block).

Every time the agent attempts an action, an enforcement gate checks the constraints.

Valid actions execute. Invalid actions revert automatically.

The constraint profile follows the agent across every protocol it touches.

The Oracle Problem: An Honest Caveat

At this point, a skeptical reader might ask: what about data from outside the blockchain? These constraint systems work on on-chain information. But much of what matters in financial governance comes from the outside world: asset prices, market conditions, protocol health metrics. This data enters the blockchain through mechanisms called oracles, essentially trusted data feeds.

If those oracles can be manipulated, and they can be and they have been, then a formally correct constraint system can be made to execute harmful transactions that technically satisfy every on-chain rule. The constraint checks out. The underlying data was wrong. The harm occurs anyway.[14]

This is not a reason to abandon the approach. It is a reason to treat oracle integrity as a foundational design requirement, not an afterthought. Any serious implementation of autonomous agent governance will need to grapple with oracle security as carefully as it grapples with constraint logic.

What Is Missing: The Governance Layer

Blockchain infrastructure satisfies three necessary conditions for autonomous agent governance: it enables deterministic enforcement of encoded rules; it supports cryptographic identity and role-based authorization; and it allows programmable constraint envelopes operating at machine speed within composable environments.

What does not exist is the systematic integration of these capabilities into a coherent governance layer specifically designed for autonomous actors. The voting modules in current DeFi protocols focus on counting token balances and recording outcomes. They do not incorporate identity classification, constraint verification, or formal specification pipelines as standard components.

This is the gap. Not a technological barrier. A design and coordination gap. The raw materials are on the shelf. Someone needs to build the structure.

Chapter Five: The Hard Questions

"Any good argument should be able to survive its sharpest critics."
-- The principle of steelmanning

We have made a case. Before accepting it, we should test it hard. There are three serious objections to the argument that blockchain architecture can support autonomous agent governance, and each one deserves a full hearing.

These are not strawmen. They are the kinds of questions that practitioners in the field actually ask, and they deserve substantive answers rather than dismissal.

The First Hard Question: Isn't This Just Centralization in Disguise?

Here is the objection at its sharpest: the entire point of decentralized finance is permissionlessness. Anyone can deploy a contract. Anyone can transact. Nobody needs to ask for permission. The moment you require autonomous agents to register with some authority, to get certified, to have their constraint profiles verified before they can participate in governance, you have created a gatekeeper. You have built a centralized trust system on top of a decentralized infrastructure.

This contradicts the thing you are trying to protect. And it probably will not even work: sophisticated actors will just deploy from new addresses, operating outside whatever registry you create, and you will end up fragmenting the ecosystem into compliant and noncompliant sections, neither of which will have the liquidity to function well.

This is a genuine tension. Take it seriously. Here is the response.

The objection conflates two different things: the base layer of blockchain infrastructure and the application layer of individual protocols. These are not the same thing, and the rules that apply at one level do not automatically govern the other.

At the base layer, the Ethereum network itself for example, permissionlessness is a core property. Nobody controls who can deploy a contract or initiate a transaction. This is not going to change, and the governance proposals in this book do not touch it.

At the application layer, the level of individual DeFi protocols, permissioning already exists. Some liquidity pools already require KYC compliance for certain functions. Some protocols restrict participation based on geography. These are protocol-level rules, not base-layer changes, and they do not undermine permissionlessness at the infrastructure level.

An agent governance registry operates at the application layer. An agent that declines to register can still transact on-chain, still interact with protocols that do not require registration, still do essentially everything it does today. What it loses is access to governance functions in protocols that choose to require registration.

OBJECTION 1 SCORECARD

The objection: Registration creates a central gatekeeper.

The rebuttal: Application-layer participation conditions already exist without breaking base-layer permissionlessness.

What remains: Purpose-built identity primitives are not yet standardized. The design needs to be done carefully to avoid creating centralization risks.

The Second Hard Question: Smart Contracts Can't Fix Legal Problems

Here is the second objection: even if you build perfect on-chain governance for autonomous agents, real harm extends beyond the blockchain. When a bot causes a liquidation cascade, real people lose real money. When governance is exploited, those losses materialize in the real world. Smart contracts cannot determine who is legally liable. They cannot compensate victims. They cannot reach the developer who deployed the exploit and is living anonymously beyond the reach of any jurisdiction.

Courts and regulators exist for exactly these reasons. Blockchain cannot replace them, and pretending it can is both naive and dangerous.

This objection is correct. And it does not defeat the argument.

The response is a distinction between prevention and remediation. These are not competing approaches to governance. They are complementary layers of a complete governance system.

Blockchain architecture enables prospective enforcement at the execution boundary. If a constraint system prevents the Beanstalk-style exploit from occurring

in the first place, there is no harm requiring legal remediation. The courts still exist. The regulators still exist. They just have fewer crises to manage.

Risk management frameworks from institutions like NIST emphasize exactly this principle: preventive design controls are more effective than reactive remediation, and a complete risk management system includes both layers.[15] You still need fire extinguishers. You also want smoke detectors and sprinkler systems. Neither eliminates the need for the other.

The European Union's Artificial Intelligence Act imposes structured oversight obligations for high-risk AI systems operating in financial markets.[16] The European Securities and Markets Authority has specifically identified maximal extractable value dynamics as a market integrity concern requiring supervisory attention.[17] These regulatory signals do not make on-chain constraint systems unnecessary. They make them a necessary component of a broader compliance framework.

OBJECTION 2 SCORECARD

The objection: On-chain enforcement cannot allocate legal liability.

The rebuttal: Correct, but prevention and remediation are complementary, not competing.

What remains: Oracle integrity is an external dependency. Correctly designed constraints can execute harmful logic if input data is compromised.

The Third Hard Question: Nobody Will Actually Build This

Here is the strongest objection, and the one that keeps practitioners up at night: even if the technical solution is coherent and the architectural foundation exists, who is actually going to build and deploy this?

Decentralized finance has no central authority. Nobody can mandate standards. A first mover who implements agent governance constraints bears real costs: development expense, reduced protocol flexibility, potential liquidity loss to competitors who do not bother. The benefits diffuse across everyone, including the competitors who did not invest.

This is an empirical question about coordination dynamics, not a technical one, and it is genuinely difficult. The response is not to dismiss the challenge but to point to historical precedent.

Decentralized finance has successfully standardized before, without anyone being in charge. ERC-20, the token standard that made fungible tokens interoperable across virtually every DeFi protocol, achieved near-universal adoption through network effects. Nobody mandated it. The Ethereum Foundation did not require it. Protocols adopted it because tokens that conformed to ERC-20 could be used everywhere, and interoperability was enormously valuable.[18]

ERC-721, the NFT standard. ERC-4337, the account abstraction standard. These all achieved substantial adoption through the same mechanism: interoperability creates collective benefit, major protocols adopt, integration pressure extends outward to smaller actors.

There is also an external pressure that did not exist when those earlier standards were developed: regulatory scrutiny. When ESMA is specifically naming extractable value as a market integrity concern and the EU's AI Act is imposing obligations on high-risk autonomous systems in financial markets, "we chose not to implement any governance standards for our autonomous agent ecosystem" becomes a harder position to maintain.

OBJECTION 3 SCORECARD

The objection: No central authority can mandate standards.

The rebuttal: DeFi has coordinated around standards before through network effects (ERC-20, ERC-721). Regulatory pressure further shifts incentives.

What remains: First movers bear disproportionate costs without assured competitive advantage. Adoption is graduated, not guaranteed.

After the Hard Questions

None of these objections defeat the core argument. But each one narrows and refines it. What emerges from this gauntlet is a more precise, more honest version of the claim.

Blockchain architecture provides a structural foundation for autonomous agent governance: not as a complete replacement for legal and regulatory systems, not as a solution that ignores the real constraints of permissionless design, and not as something that will be universally adopted quickly or easily. Rather as a necessary technical layer that can do something no legal or regulatory mechanism can do: enforce constraints on autonomous agents at the speed and level of granularity at which they actually operate.

The thesis survives the hard questions. It comes out of them smaller and more precise. That is actually a sign of a healthy argument.

Chapter Six: Building the Rulebook We Actually Need

"The urgency is practical, not theoretical. The bots are already here."
-- The conclusion, plainly stated

Let's take stock of where we have arrived.

We started with a heist: one hundred and eighty-two million dollars, thirteen seconds, no rules broken. That heist was not an anomaly. It was a demonstration of a structural mismatch that runs through the entire landscape of decentralized finance: governance systems designed for humans operating in a world increasingly dominated by machines.

We traced the mismatch through three distinct failures: the assumption of human deliberation, the assumption of deliberative timescales, and the assumption of traceable accountability. Each assumption made sense when governance frameworks were designed. Each assumption is now violated routinely by autonomous agents operating at machine speed.

We found that the fix is not to abandon blockchain or decentralized finance. It is to build the governance layer that should have been built alongside the execution infrastructure. The architectural foundations for that layer already exist: deterministic enforcement through smart contracts, cryptographic identity and credentialing systems, programmable constraint envelopes, and composable interoperability that allows constraints to follow agents wherever they go.

We stress-tested this argument against three serious objections and came through with a more precise, more honest version of the claim. The solution requires careful design to avoid creating centralization. It requires parallel development of legal and regulatory frameworks. It requires ecosystem coordination that is difficult but precedented.

What Needs to Be Built

This book does not pretend to be a technical specification. That work requires engineers, protocol designers, governance researchers, and legal scholars working

together over an extended period. But we can be clear about the shape of what is needed.

Agent identity standards: a way to classify autonomous actors by type, autonomy level, and operational scope, verifiable on-chain without requiring off-chain identity disclosure. This is the foundation everything else builds on.

Constraint frameworks: standardized templates for encoding behavioral limits on autonomous agents, covering vote weight caps, capital limits, protocol whitelists, and transaction rate limits, with formal verification pipelines that can prove constraints behave as specified before they go live.

Machine-speed circuit breakers: automated monitors that detect anomalous behavior patterns and trigger constraint escalations without requiring human review at the moment of execution. These are the smoke detectors for autonomous agent governance.

Cross-protocol interoperability: mechanisms by which governance commitments made at one protocol are recognized automatically at others, so that agents cannot simply move to less regulated protocols to escape their constraints.

Off-chain legal development: parallel frameworks for assigning liability, providing remediation to victims, and establishing regulatory expectations for autonomous agent deployers. This is complementary to on-chain governance, not a replacement for it.

The Urgency Is Real

It would be easy to read this book as an argument about the future. Something to consider for later, when the problem becomes more urgent, when the regulatory environment becomes clearer.

But the problem is already urgent. Autonomous agents are not coming to DeFi. They are already here. MEV extraction, systematic profit extracted from ordinary users by algorithmic agents exploiting transaction ordering, already amounts to hundreds of millions of dollars per year. Liquidation cascades triggered by automated agents have already destabilized protocols during market downturns. Governance

exploits like Beanstalk have already happened, and similar structural vulnerabilities exist in dozens of protocols today.

The gap between execution speed and governance speed is not stable. As AI systems become more capable, as agents become more sophisticated, more adaptive, more able to identify and exploit edge cases in governance design, the divergence will widen. A governance layer built today, even an imperfect one, is worth dramatically more than a perfect one built five years from now.

> *The question is not whether we need better governance for autonomous agents in DeFi. It is whether we will build it before the next Beanstalk.*

A Final Thought: This Is Not Just About Bots

The story told in this book is framed around autonomous agents in decentralized finance. But the underlying question, how do you govern systems that operate faster than human oversight can follow, is much larger.

As AI systems become more capable and more embedded in critical infrastructure, the governance challenge extends far beyond DeFi. Financial markets. Healthcare systems. Infrastructure management. Legal research and decision support. In every domain where AI systems make decisions at machine speed, there is a version of the governance gap described here.

Decentralized finance is, in a sense, a laboratory. It moved faster than any other domain to deploy AI agents in consequential financial operations. It suffered the consequences of doing so without adequate governance infrastructure. The lessons learned in this laboratory are relevant far beyond the world of cryptocurrency.

The governance challenge of autonomous systems is one of the defining problems of the coming decade. In DeFi, we can already see its contours clearly. We can already identify solutions. We have already paid the price of inaction.

The question is whether we are paying attention.

* * *

The Beanstalk exploit lasted thirteen seconds. The governance gap it revealed has been widening for years. It will continue to widen until someone builds the rulebook that autonomous agents actually need.

The materials are on the shelf. The design space is understood. The stakes are real.

It is time to build.

Notes

Superscript numbers throughout the text refer to the numbered entries below.

1. Lumineau, F., Wang, W., & Schilke, O. (2021). Blockchain governance: A new way of organizing collaborations? Organization Science, 32(2), 500-521.
2. Fritsch, R., Muller, M., & Wattenhofer, R. (2024). Analyzing voting power in decentralized governance: Who controls DAOs? Blockchain: Research and Applications, 5(3), 100208.
3. Austgen, J., Barrera, O., Zhao, S., & Breidenbach, L. (2023). DAO decentralization: Voting bloc entropy, bribery, and dark DAOs. arXiv:2311.03530.
4. Sun, X., Stasinakis, C., & Sermpinis, G. (2022). Decentralization illusion in DeFi: Evidence from MakerDAO. arXiv:2210.11203.
5. Daian, P., Goldfeder, S., et al. (2020). Flash Boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. 2020 IEEE Symposium on Security and Privacy, 910-927.
6. Gramlich, V., Jelito, D., & Sedlmeir, J. (2024). Maximal extractable value: Current understanding, categorization, and open research questions. Electronic Markets, 34, 39.
7. Chaffer, T., Goldston, J., & Jansen, M. (2024). Decentralized governance of AI agents. arXiv:2412.17114.
8. Qin, K., Zhou, L., & Gervais, A. (2022). Quantifying blockchain extractable value: How dark is the forest? 2022 IEEE Symposium on Security and Privacy, 198-214.
9. Immunefi. (2022). Hack analysis: Beanstalk governance attack. Medium. https://medium.com/immunefi/hack-analysis-beanstalk-governance-attack-april-2022
10. De Filippi, P., & Wright, A. (2018). Blockchain and the law: The Rule of Code. Harvard University Press.
11. Tolmach, P., Li, Y., Lin, S.-W., Liu, Y., & Li, Z. (2022). A survey of smart contract formal specification and verification. ACM Computing Surveys, 54(7), 148.
12. Beck, R., Muller-Bloch, C., & King, J. L. (2018). Governance in the blockchain economy. Journal of the Association for Information Systems, 19(10), 1020-1034.
13. Shavit, Y. et al. (2023). Practices for governing agentic AI systems. OpenAI technical report.
14. Arora, S., Bhatt, P., & Zhao, Y. (2024). Security of critical infrastructure in decentralized finance. University of Oregon Technical Report AREA-202411.
15. National Institute of Standards and Technology. (2023). Artificial intelligence risk management framework (AI RMF 1.0). NIST AI 100-1.
16. European Parliament. (2024). Artificial Intelligence Act: Regulation (EU) 2024/1689.
17. European Securities and Markets Authority. (2025). Maximal extractable value: Implications for crypto markets. ESMA50-481369926-29744.
18. Lumineau, F., Wang, W., & Schilke, O. (2021). Blockchain governance: A new way of organizing collaborations? Organization Science, 32(2), 500-521.

Further Reading

The following sources are cited in the text or provide useful background for readers who want to go deeper.

Alqithami, S. (2026). Autonomous agents on blockchains: Standards, execution models, and trust boundaries. arXiv:2601.04583.

Beanstalk Farms. (2022). Beanstalk governance exploit. Bean.Money. https://bean.money/blog/beanstalk-governance-exploit

Chen, H., Guo, J., Wang, C., & Zhang, Y. (2024). Verifying declarative smart contracts. ICSE 2024.

Cong, L. W., Tang, K., Wang, J., & Zhao, X. (2023). Centralized governance in decentralized organizations. SSRN Working Paper 3922057.

Davila, R., Mesnard, T., & Serrano, A. (2025). Smart contracts formal verification: A systematic literature review. arXiv:2510.17865.

International Association of Privacy Professionals. (2024). AI governance in the agentic era.

Knight First Amendment Institute. (2025). Levels of autonomy for AI agents. Columbia University.

National Telecommunications and Information Administration. (2023). Artificial intelligence accountability policy report.

Pandey, R. (2025). The agentic AI governance framework. SSRN Working Paper 5652350.

Schar, F. (2021). Decentralized finance: On blockchain- and smart contract-based financial markets. Federal Reserve Bank of St. Louis Review, 103(2), 153-174.

Weidener, L., Laredo, F., Kumar, K., & Compton, K. (2025). Delegated voting in decentralized autonomous organizations: A scoping review. Frontiers in Blockchain, 8.

About the Author

Gary Chigaros is a graduate student at the University of the Cumberlands, where his research focuses on blockchain governance and decentralized systems.