Ethereum: A Microcosm for Protocol Studies

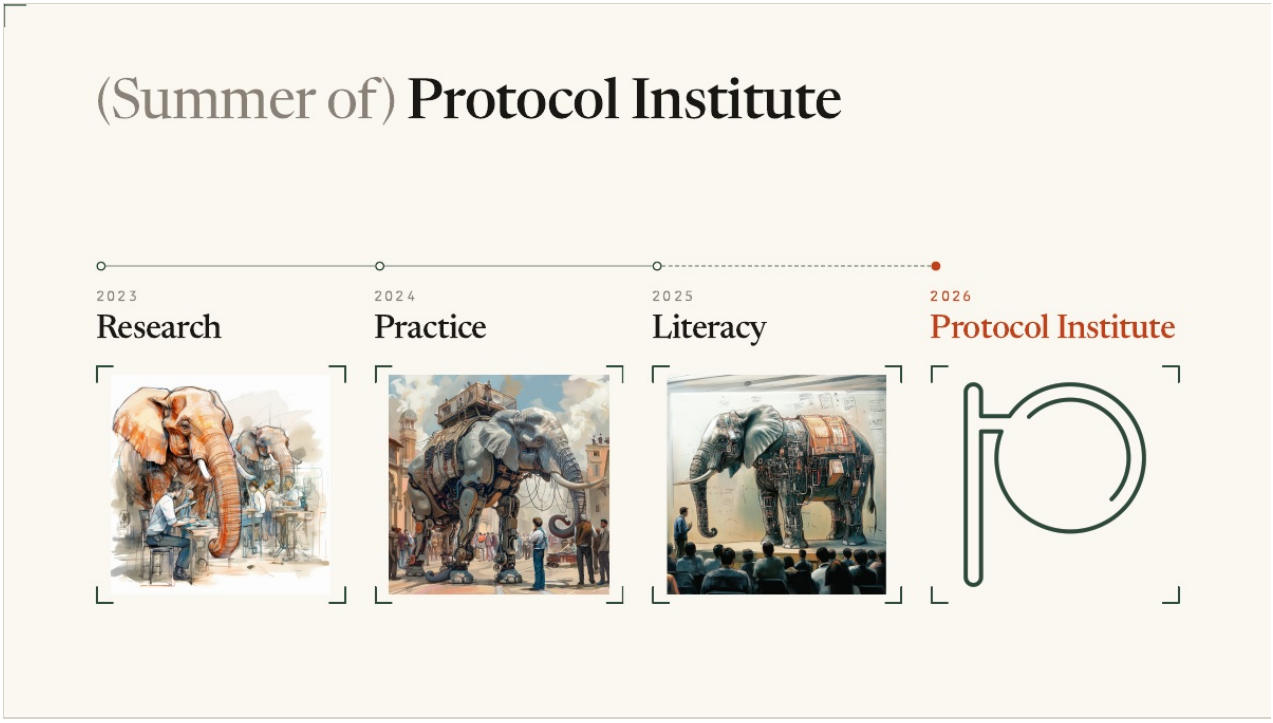
Adapted from my talk at the Protocol Symposium on September 23, 2026.



I spent most of the last decade coordinating Ethereum core protocol development. Around 2022, I realized I didn't have a good way to think about that work. There were frameworks for software engineering, product management, markets and financial models, but nothing for managing or stewarding protocols.

So I reached out to the one person I could think of who might help, and who knew enough about blockchains to wrap his head around the problem: Venkatesh Rao. After talking for a while, we realized the answer wasn't to study Ethereum more closely, but to zoom out and study the broader class Ethereum was an instance of: protocols.

From Summer of Protocols to the Protocol Institute



That was the start of Summer of Protocols, and now the Protocol Institute.

In the first year, we weren't sure this was a good idea. The image I kept coming back to was the parable of the blind men and the elephant: everyone touches a different part and comes away with a very different picture, but they're all touching the same animal. When we started, it wasn't clear there was a protocol elephant at all.

So in 2023, we funded researchers working on protocols across very different domains, to see whether studying their own work through the lens of protocols would produce insights, mental models or frameworks that carried over. We were quite happy that it did.

My own work, though, was always as a practitioner. I'm not an Ethereum researcher: my job was getting the protocol live in production. In 2024, we wanted to see whether the theory from the first year translated into practical techniques, so we funded a cohort of protocol practitioners working on everything from adding encryption to messaging protocols, to wildfire prevention in California, to new governance and voting protocols. Again, the goal was to find things that generalized, or at least transferred from one domain to the next, and again, we were happy with the results. It felt like protocols were a useful applied field, alongside the research and theory.

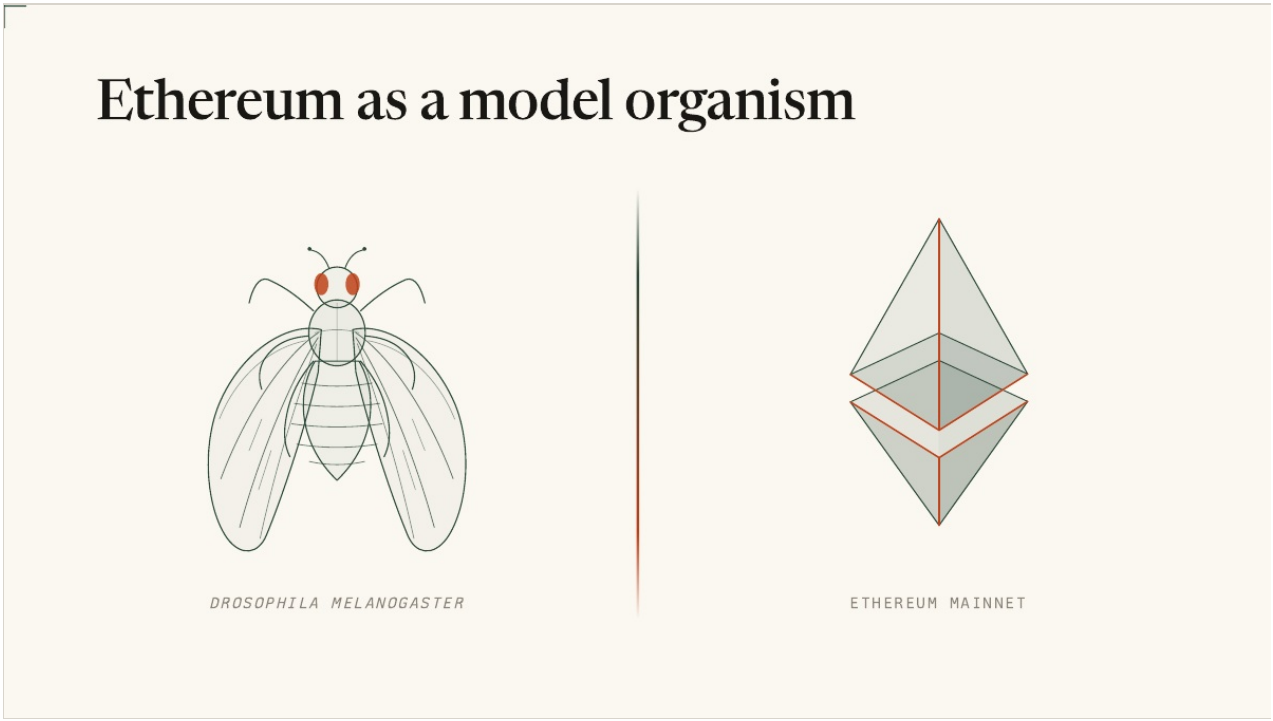
By 2025, we had built up enough knowledge that funding another huge research cohort felt less useful than bringing the work out into the world and making the world more protocol literate. We funded teachers across many fields, from technical ones to the social sciences and the arts, to add protocols as a lens on their subject matter. Once more (obviously, or the Protocol Symposium wouldn't exist), the protocol dimension added something useful.

After three years, it felt like the field needed more than a single summer program. Longer-lived efforts like the special interest groups (SIGs) had sprawled out around it, and it needed a durable home. I'm pretty stoked that it now has one in the Protocol Institute.

One fear we had at the start was that this would be seen as a narrow blockchain research thing. We explicitly deemphasized crypto and Ethereum early on: pretty much none of the projects were about crypto, even though the Ethereum Foundation funded the program. We wanted the field to grow on its own and be multidisciplinary, without blockchains sucking the air out of everything. With over 50 sessions at this year's Protocol Symposium and almost none directly about crypto, I think we succeeded on that front.

Which means I can come back to the thing I actually know. I'm not an expert in protocols, but I am, to some extent, an expert on Ethereum. And Ethereum is a really interesting protocol for protocol studies folks to understand, research and experiment with, even if they don't particularly care about crypto.

Ethereum as a model organism



Ethereum is a good model organism for protocols, a bit like fruit flies in biology.

Biologists have studied fruit flies for over a century for a few reasons. They have a relatively simple genome, they're cheap to breed and keep, and they reproduce quickly, so you can observe biological mechanisms across many generations in a short amount of time. Because of these neat properties, a huge body of tooling, best practices and knowledge has been built around them, and they've become the default for a large share of biological experiments.

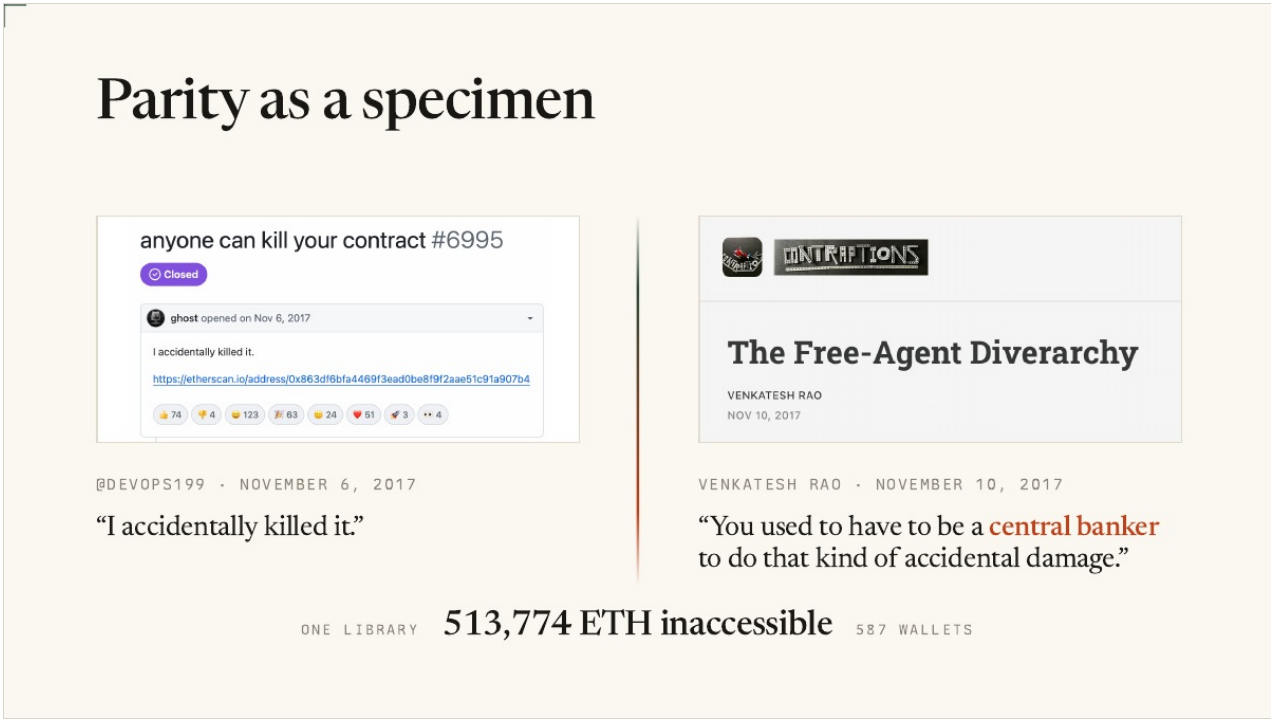
Ethereum offers something similar for protocol research.

First, it has a short but extremely rich history. It covers protocol innovation both at the core layer and in what's built on top of it, plenty of technical crises and the entire governance apparatus needed to bring all of this to reality.

Second, it's unusually inspectable. Not only are the transactions, code and state of the chain public, but so are the governance discussions, specifications and call transcripts. You can map the technical history alongside the social coordination around it, more or less in real time, and go down to any arbitrary point.

Third, if you want to experiment on it, the entire premise of the system is that it's open and permissionless. Anyone can come in and build anything, without a third party or gatekeeper telling them what they can or can't do. This has made it fertile ground for experimentation at a reasonable scale.

Parity as a specimen



One episode from Ethereum's early years shows what's possible, the stakes involved and a bit of the origin of all of this.

In 2017, a popular application on Ethereum was the multi-signature wallet. An organization controlling a wallet wants some restrictions on who can move funds, so instead of a single key, a multisig has several administrators and a threshold. The Protocol Institute, for example, could have a multisig where Timber, Venkat and I are administrators, and 2 of the 3 need to approve any transaction.

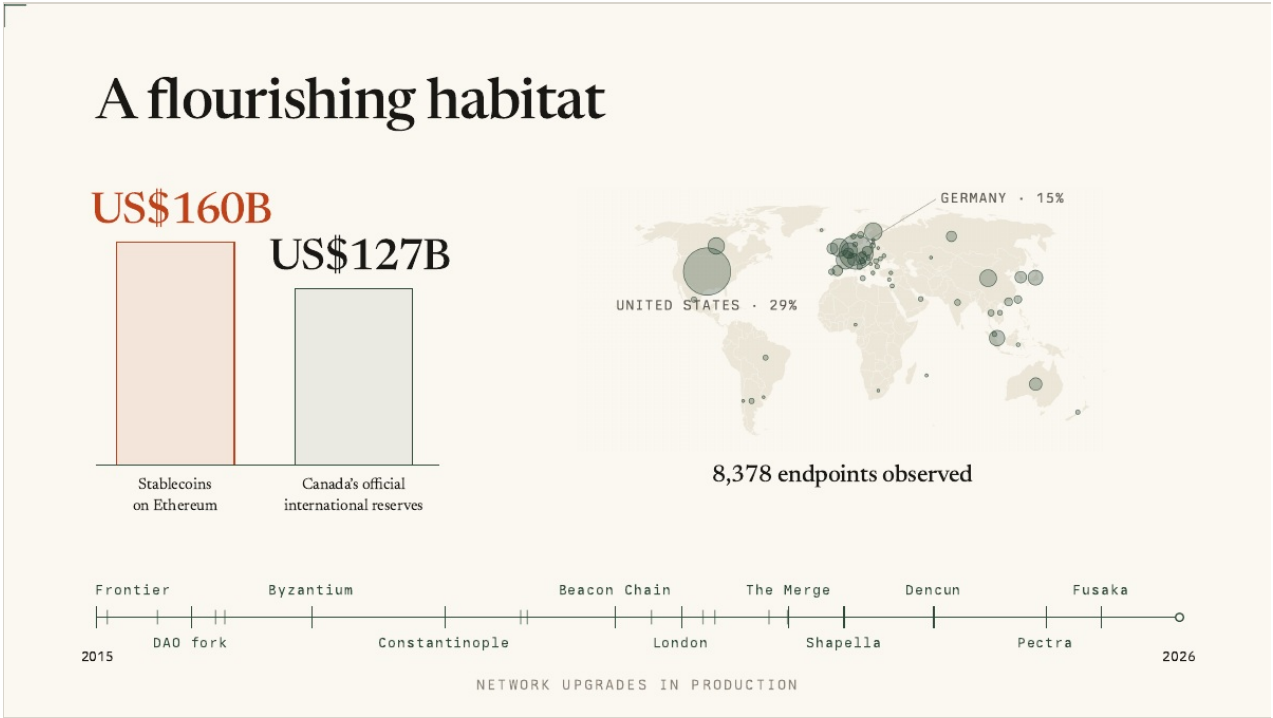
The most popular multisig at the time was built by Parity. Deploying smart contracts was quite expensive back then, so they had a clever idea: store all of the wallet's business logic in a single shared library contract, and have each individual wallet only store its configuration. A Protocol Institute wallet would record who the admins are, but the code that actually manages the funds would live somewhere else.

On November 6, 2017, a GitHub user going by devops199 noticed the shared library had never been properly initialized, made themselves its owner and then killed it. They opened an issue titled "anyone can kill your contract", which read "I accidentally killed it." 513,774 ETH (about $150M at the time) spread across 587 wallets became inaccessible. Everything was still onchain except the logic needed to use the funds.

One person who wasn't a crypto insider at the time, but noticed this and appreciated its scale, was Venkat. A line from his newsletter a few days later, in a post called "The Free-Agent Diverarchy", has stuck with me ever since: "You used to have to be a central banker to do that kind of accidental damage." Ethereum had expanded the frontier of how far a small action could go in causing institutional-scale consequences.

This obviously wasn't the most positive experiment, but it shows that even early on, Ethereum was a playground for things that weren't happening, and couldn't have played out the same way, anywhere else.

A flourishing habitat

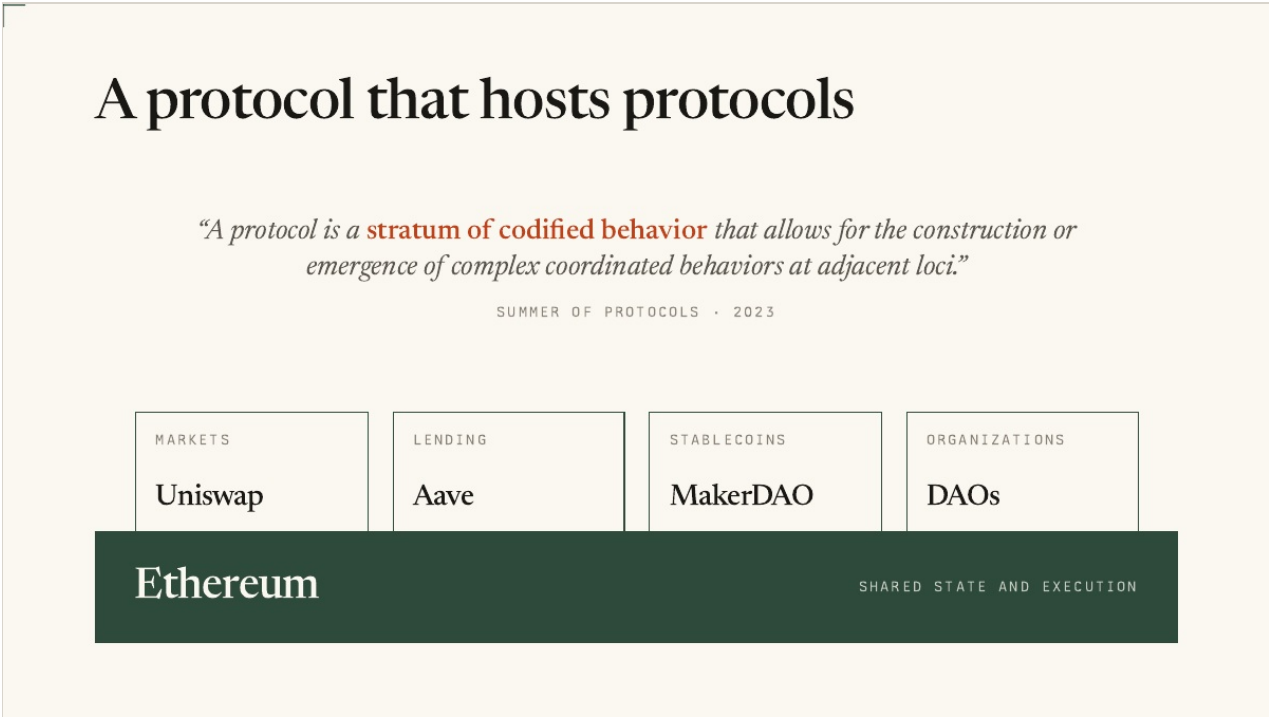


That was quite some time ago. Since then, Ethereum has matured. Multisigs are much safer, and things operate at a much bigger scale.

Beyond ETH itself, Ethereum hosts thousands of other assets, notably stablecoins. There's about US$160B of stablecoins on Ethereum. For perspective, Canada (where I'm from) holds about US$127B in official international reserves.

The network is also globally distributed, with over 8,000 nodes around the world, and the protocol itself has evolved dramatically. There have been over 20 upgrades to the core protocol since launch, all without the network ever going down. That makes for an uninterrupted slice of history, from launch to today, where you can observe both the growth in usage and scale and the social coordination around it.

A protocol that hosts protocols

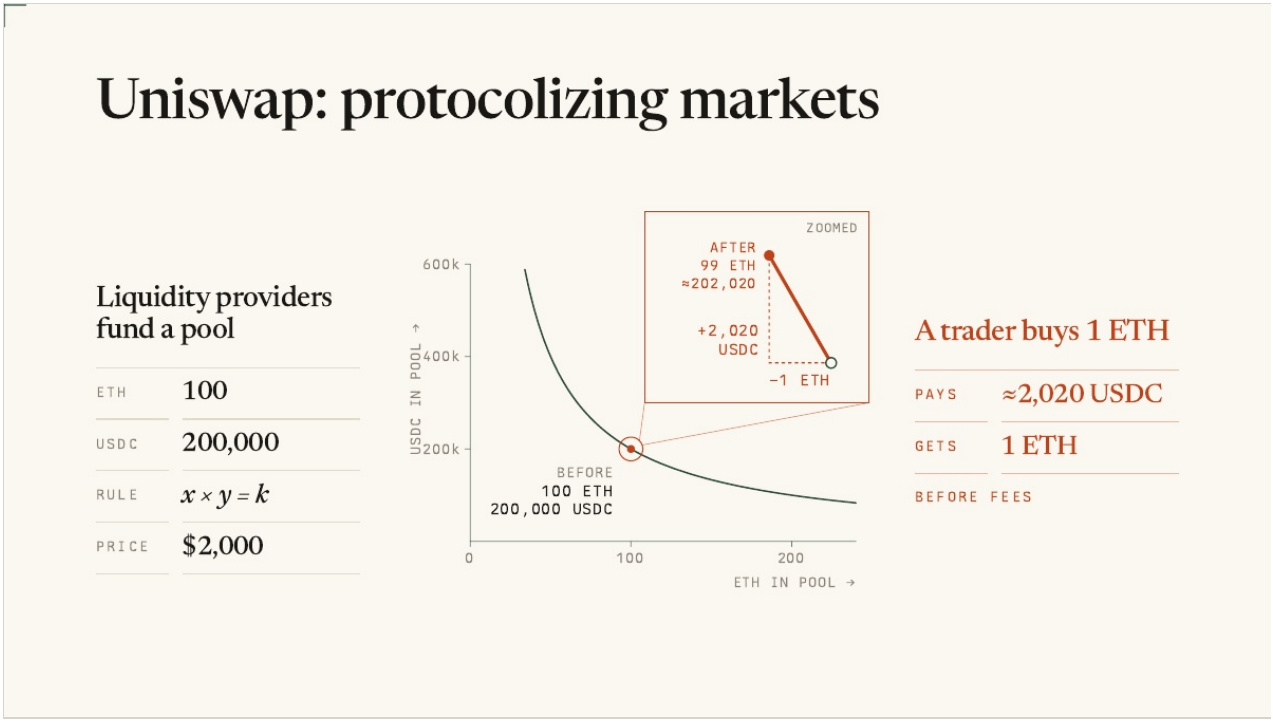


Ethereum is interesting to protocol researchers for more than the protocol itself, though. It's also a platform for other protocols.

This was part of the inspiration for the definition of protocols from the first Summer of Protocols: "a stratum of codified behavior that allows for the construction or emergence of complex coordinated behaviors at adjacent loci." In plainer English, a protocol is a set of rules that can serve as a foundation for more complex coordination on top of it.

That's what Ethereum is. It provides shared state and execution, and a bunch of protocols have sprung up on top of it over the years. Uniswap protocolized markets, Aave did the same for lending and borrowing, MakerDAO created stablecoins out of volatile assets and DAOs did it for organizations.

Uniswap: protocolizing markets



Uniswap is one of the cleanest examples of what protocolizing something looks like.

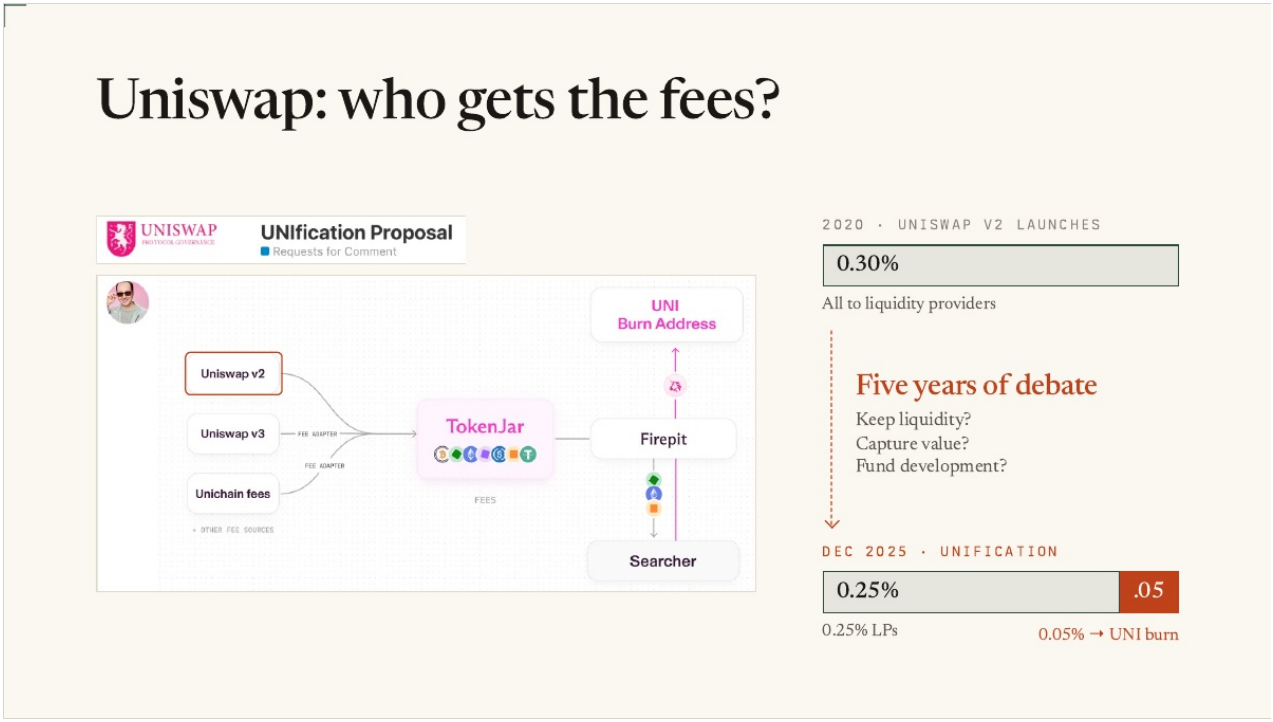
A traditional trading venue has buyers and sellers posting bids and asks, which get matched when prices cross. Uniswap instead lets anyone create a pool, a trading venue for two arbitrary assets, and fund it by depositing some of both. Trades are priced by a single rule: the product of the two balances in the pool stays constant ($x \times y = k$).

Say a pool has 100 ETH and 200,000 USDC, which implies a price of 2,000 USDC per ETH. Traders aren't trading against a counterparty, they're trading against the pool. Buying 1 ETH leaves the pool with 99 ETH, so to keep the product constant, the buyer needs to add about 2,020 USDC (before fees) rather than the 2,000 the starting price implied. Taking 1% of the pool's ETH costs roughly 1% over that price, and leaves the pool quoting a higher price for the next buyer. That's how the price equilibrium gets set.

This is kind of wild when you think about it. Markets, which outside of crypto are really complicated infrastructure, get reduced to a single protocolized rule. And it's enabled a ton. The most obvious is permissionless price discovery. Before Uniswap, getting an asset to trade generally meant finding a centralized exchange, in crypto or elsewhere, willing to list it. Now anyone can create and fund a pool for any pair of assets. And because pools are open and transparent, they're fully composable, so more complex financial products on Ethereum can be built directly on top of them.

This isn't some niche thing, either. Uniswap had done about $4 trillion of cumulative trading volume as of 2025, the last figure I could find.

Uniswap: who gets the fees?



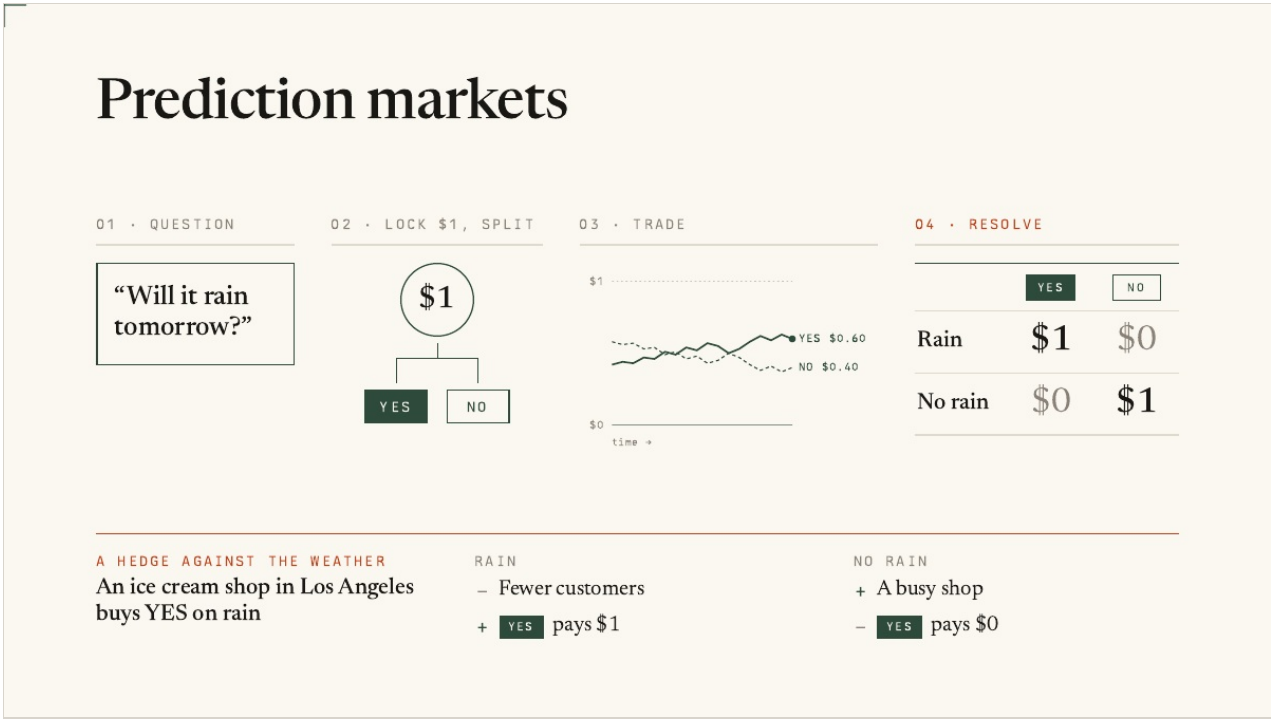
That example simplifies things a bit, and what it leaves out is where the human side of protocols comes in.

In practice, Uniswap charges a fee on every trade. When v2 launched in 2020, the full 0.30% went to liquidity providers (LPs), to attract liquidity to the pools. The team knew this probably wasn't the best long-term design, so the protocol shipped with a way to allocate the fee differently, initially set to send 100% to LPs, with the expectation of a debate as the protocol gained scale.

Five years of debate followed, all of it in public. Do you keep everything going to LPs to attract liquidity? Do you let the protocol's owners capture some of the value? Do you use it to fund future development? In December 2025, a proposal known as "UNIfication" was agreed to. It slightly reduced the LP share to 0.25%, and the remaining 0.05% of each trade now effectively goes to burning UNI, Uniswap's governance token. That captures some value and, since development is mostly funded through the token, helps sustain the protocol.

It's a really interesting case study: a successful protocol that's managed to capture part of finance, with the debate about how to steward it over the long term playing out in parallel, in public.

Prediction markets

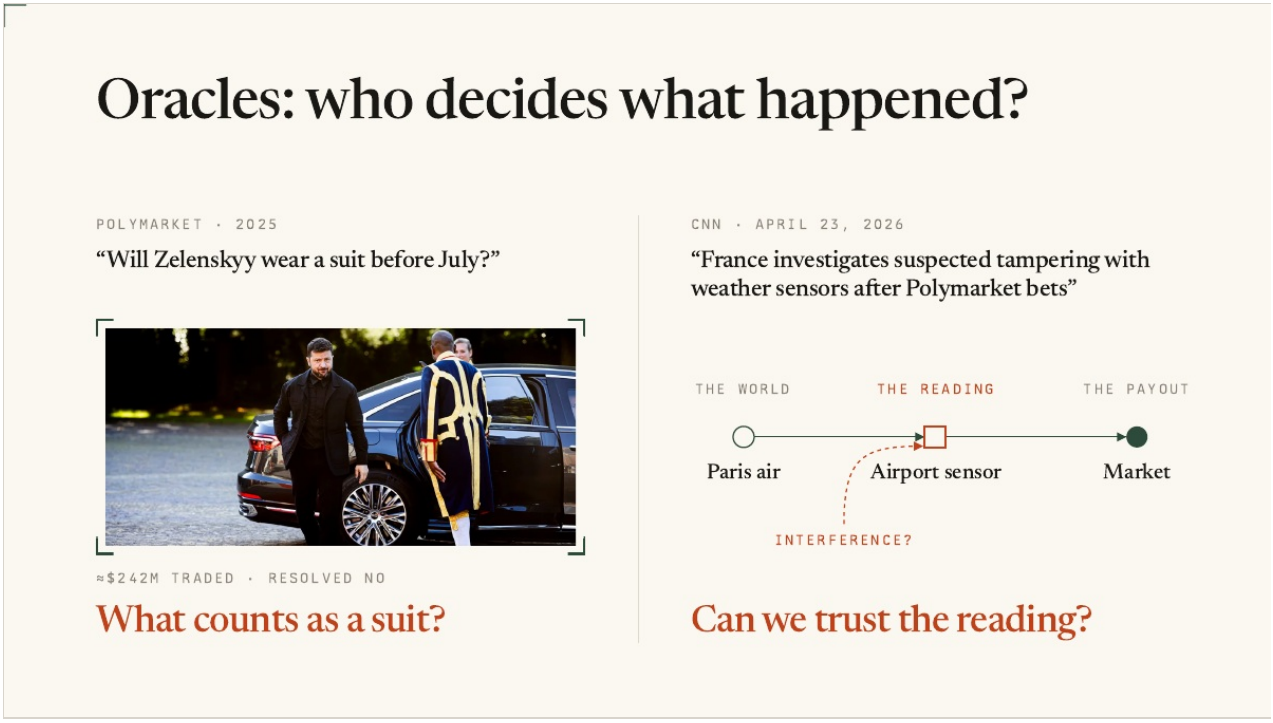


That said, Ethereum isn't only useful for making existing things more efficient. What it's really special for is being fertile ground for experiments that might not be possible at the same scale outside of blockchains.

Prediction markets are a good example, and they benefit from the same permissionless property that lets anyone create a Uniswap pool. The mechanism is quite straightforward. Start with a question you want a prediction for, say, "Will it rain tomorrow?" Every dollar in the market is split into a YES token and a NO token, whose prices always sum to $1. Markets usually start at 50/50 and move as people trade on the forecast. When the market resolves, if it rained, YES holders get $1 and NO holders get nothing, and vice versa.

Beyond pricing in new information about the world (people use them for elections and whatnot), prediction markets also enable real hedges outside of blockchains. One nice example is an ice cream shop in Los Angeles that buys YES on rain. If it rains, they get fewer customers, but their tokens pay out $1 each. If it doesn't, the tokens go to zero, but the shop is busy. It's a simple way to smooth out a business that's heavily weather-dependent.

Oracles: who decides what happened?



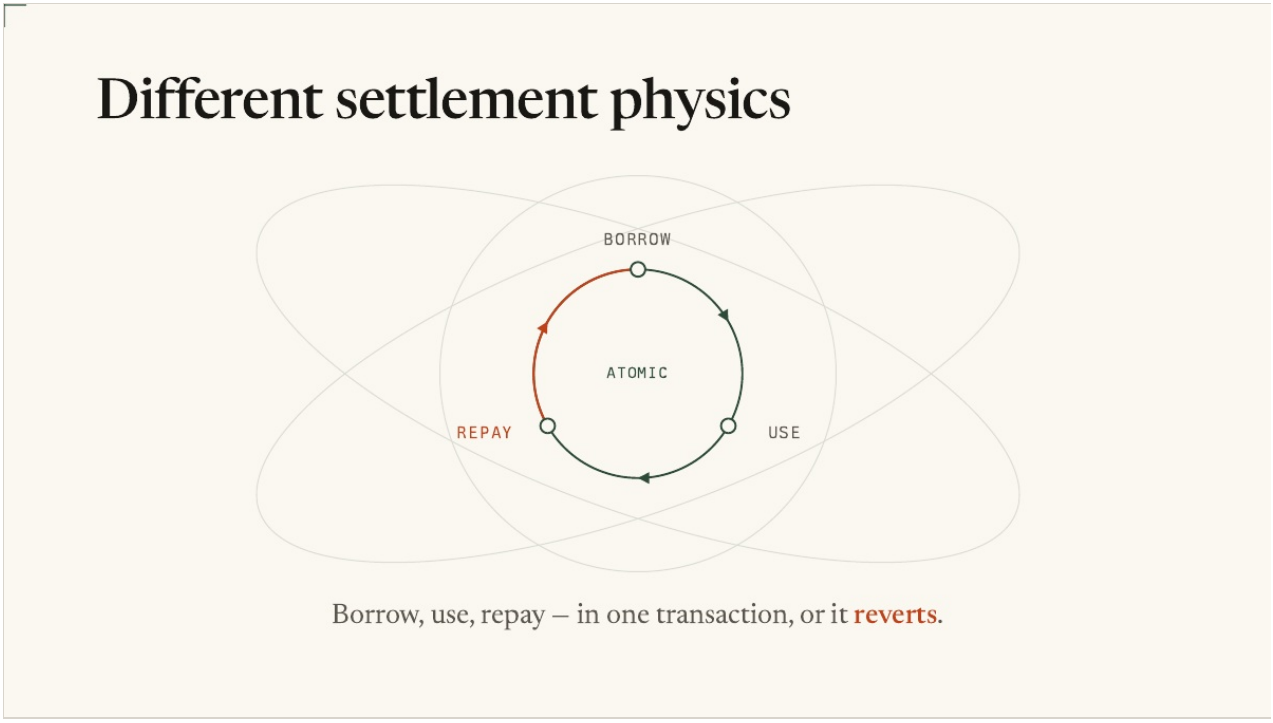
The interesting part, again, is the human coordination side of governing these things. Blockchains are good at executing arbitrary logic and verifying things that are provable, but the real world tends to be fuzzier. Figuring out a fact about the outside world that the chain then needs to adjudicate is known as the oracle problem, and prediction markets have produced plenty of examples of where it breaks down.

In 2025, Polymarket ran a market on whether President Zelenskyy would wear a suit before July. He'd historically avoided suits at government meetings during the war, then showed up to a NATO dinner in a sort of blazer/sports coat, and whether that counted as a suit became extremely contentious. About $242M traded on the market, which resolved NO based on the specific terms defining a suit.

In April 2026, France opened an investigation into suspected tampering with weather sensors after bets on Polymarket. The market was on the temperature in Paris, read by a sensor at the airport, and someone was suspected of using a hair dryer on it to push the reading up while betting on the outcome.

Even when the underlying fact is fairly objective, designing these protocols means thinking through the failure cases and how to resolve them in an elegant way. That gets complicated fast once you move from blockchains and mathematical proofs, which are much easier to settle, to the real world.

Different settlement physics

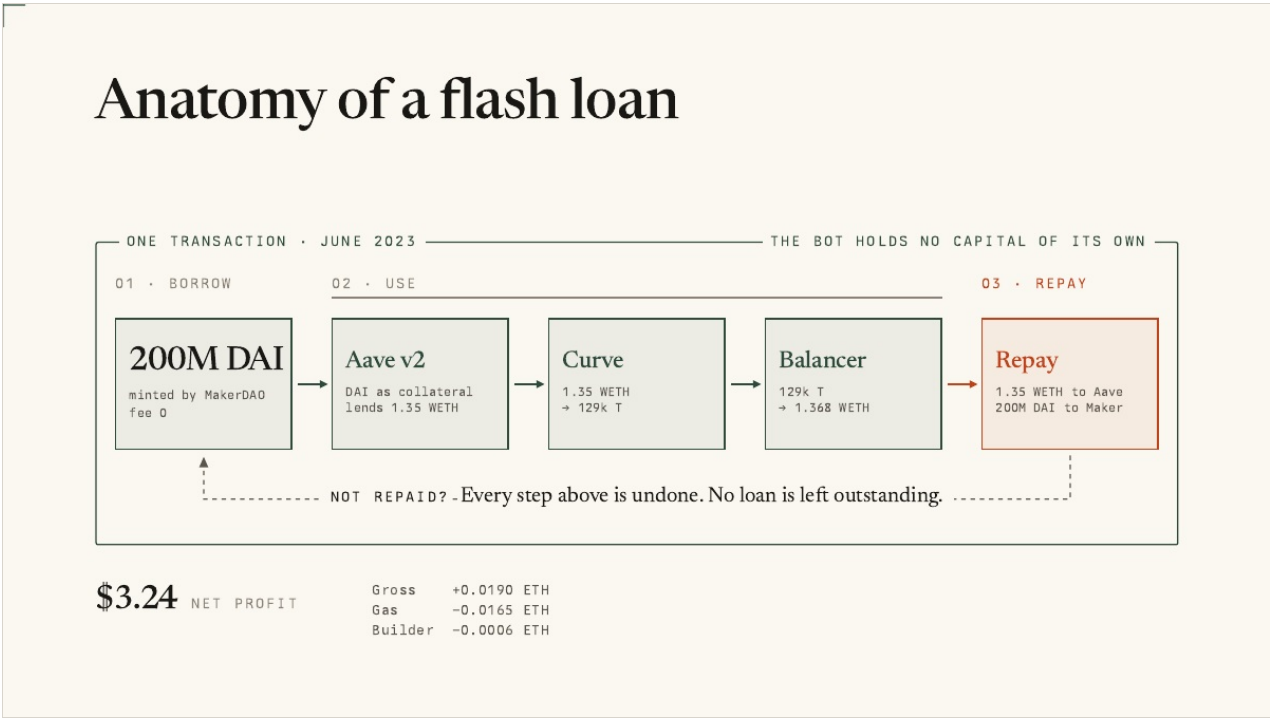


So what can you do if you restrict yourself to what's possible only onchain?

One property that's quite unique to Ethereum is that transactions are fully atomic, even when they touch many unrelated protocols. Either a transaction happens, exactly as specified and with everyone agreeing that it did, or it reverts and it's as if it never happened. That's a different type of settlement than you see in traditional finance.

It's a bit like space manufacturing. In zero gravity, you can manufacture things that are impossible to assemble on Earth. Ethereum's guarantee of atomicity is a similar invariant, and it lets us build things that aren't possible in traditional financial systems.

Anatomy of a flash loan



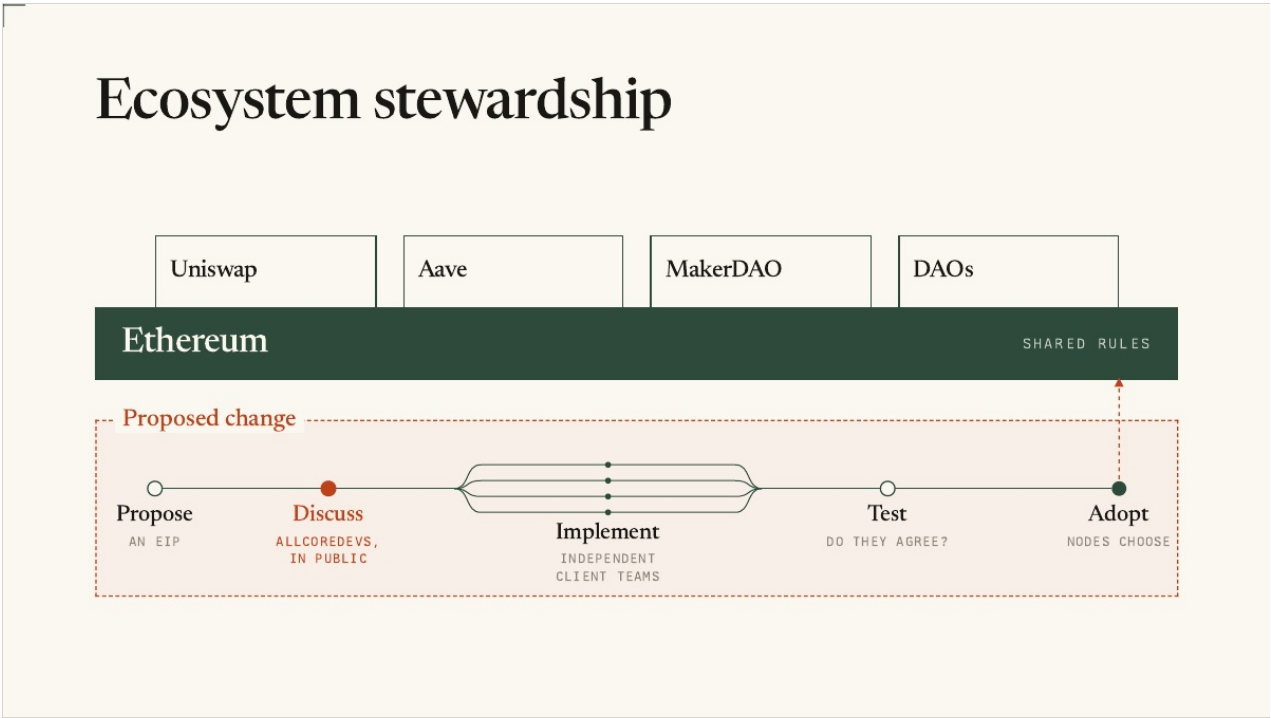
A really simple example is a large loan that would never get approved anywhere else. On Ethereum, these are called flash loans: borrow, use and repay in one transaction, or the whole thing reverts.

One of the more extreme examples happened in June 2023. An arbitrage bot holding zero capital of its own spotted a tiny arbitrage. In a single transaction, it flash-minted 200M DAI from MakerDAO, posted it as collateral on Aave to borrow 1.35 WETH, ran that through Curve and Balancer to close the arbitrage, then repaid everything, netting a $3.24 profit.

The only reason this is possible is that MakerDAO knows one of two things will happen. Either the transaction completes and the 200M DAI comes back, or it doesn't, the entire transaction is invalid and it's as if the loan was never originated in the first place.

The point isn't that Ethereum is the best place for arbitrage opportunities, but that you can build structures with completely different trust assumptions than in other domains. That works because Ethereum enforces extremely strong and rigid rules, and everybody knows exactly how the protocol will execute, so they can allow actions they'd never be comfortable allowing elsewhere.

Ecosystem stewardship



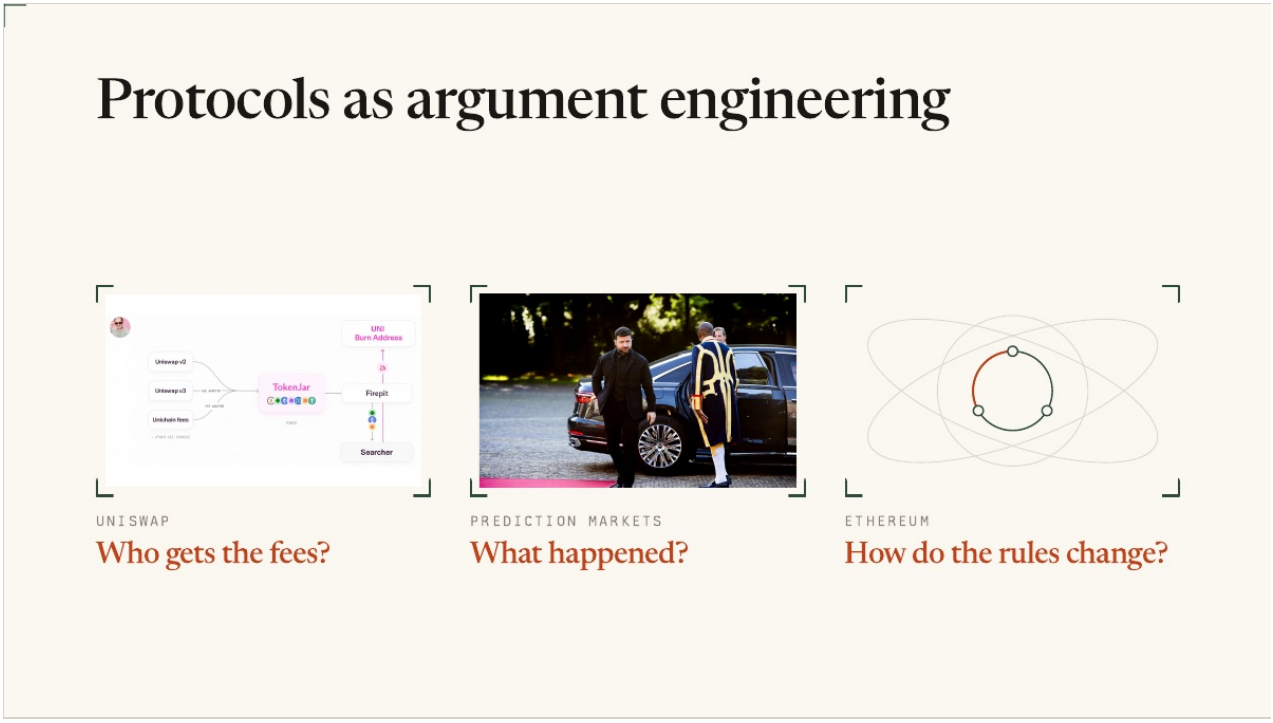
Which brings us to the last example: how do you change a system like this while preserving all the properties people rely on?

This was my day job when we started Summer of Protocols. Ethereum needs to keep improving, but people have built markets, lending protocols, prediction markets and more that make pretty significant assumptions about its behaviour. Every proposed change raises the question of whether it will break something people rely on, or introduce a new degree of freedom that affects the conditions things depend on. That's not something a single team or operator can answer.

So the process is, again, completely open and public. Anyone can propose a change to Ethereum through an Ethereum Improvement Proposal (EIP). Proposals are debated publicly on AllCoreDevs (the calls I used to run), then implemented and tested in parallel by multiple independent client teams, before being adopted into the shared rules, or not. Adoption itself comes down to node operators choosing to run the upgrade.

That's the crux of the challenge: keeping the system dependable while still improving and adapting it over time.

Protocols as argument engineering

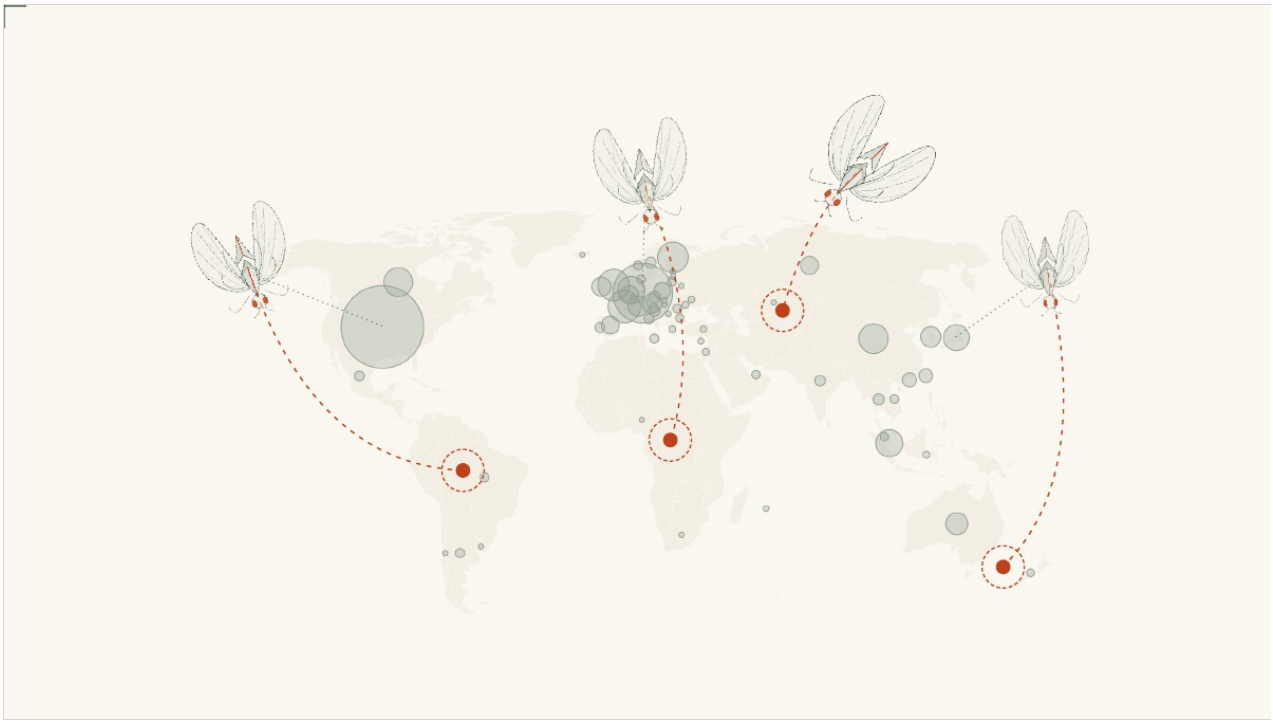


Zooming out, these examples bring us to a definition of protocols we arrived at later on: an engineered argument.

You can think of a protocol as something that draws a boundary between what formal rules settle and what people still have to decide, where the formal rules give you full predictability. The more room you leave for judgment and flexibility, the more room you leave for authority, and for mechanisms to evolve around resolving disagreements.

Uniswap's code executes every trade and manages each pool's liquidity, but you still need governance to decide what to do with the fees. A prediction market can automate the whole process of who gets paid once a question resolves, but the oracle still requires human input and judgment. And Ethereum is the most general version of this: a protocol that executes arbitrary software rules, wrapped in a huge human coordination process that decides how those rules change.

Good argument engineering means designing both sides of that boundary, and figuring out how they work together cleanly.



Ethereum is an unusually rich and open ecosystem to study this in. You can follow every public debate and all of the code as it evolved, along with its implications for the platform and its users. I helped start protocol studies because I wanted to better understand Ethereum, and now that the field has a life of its own, I hope Ethereum can offer something back as a case study.

And for anyone who wants to go beyond studying protocols and experiment with them, I think Ethereum is the perfect environment: tools, existing protocols to build on and the freedom to try something new and engineer your own arguments.