EPIC CASH

Quiet Private Money

The case for Epic Cash, and why the world has not noticed... yet.

Todd Pitner



Quiet Private Money

The case for Epic Cash, and why
the world has not noticed... yet.

Todd Pitner

BEFORE THE BEGINNING

Why I wrote this

My name is Todd Pitner, founder of my575e.com, a site dedicated to helping people keep more of what they earn, and cohost of Decentralize.TV with Mike Adams — a show about decentralized living and escaping centralized control.

Before cohosting Decentralize.TV, I spent several years learning everything I could about privacy coins and sharing my discoveries on my YouTube channel.

After a year of due diligence, I was drawn to Epic Cash, which became and remains my favorite privacy coin. For three years I published an Epic Cash video every Friday to an audience that mostly already agreed with me. Many of those episodes I have since taken down, which is why the archive is thinner than the effort was.

That is the problem with video. The people who find it are looking for it. Someone who has never heard of Epic Cash will not type its name into YouTube, which means the argument never reaches the only person who needs it: someone who suspects their money is not entirely theirs, and has not yet been given a way to check.

A book can be handed to that person. A video cannot. So this is not the videos turned into prose. It is the argument I could never make ten minutes or an hour at a time — the whole case, with the working shown, in an order that lets a skeptic verify each step before the next one is asked of them.

What I have tried to do is not persuade you that I am right. It is to give you a test sharp enough to settle the question without me. If

you finish this book and conclude that I am wrong, you will have concluded it from evidence you can check, which is more than most writing in this category offers.

Two disclosures, because they matter. I hold EPIC, and I have for years. Assume that colors everything in this book, then check the work anyway.

Every number here is either arithmetic, visible on a public chain, or attributed to a named source with a date. If the argument only works because you trust me, it does not work.

I have also been wrong about the timing. Repeatedly. I thought the world would notice sooner than it did, and I said so in public more than once. The design has held up better than my forecasts, which is the most honest way I can describe the last seven years.

That is the whole of my case for writing this: not that I was right about when, but that the thing I was pointing at is still standing, still small, and still checkable.

Which is why the book exists now rather than five years ago: the question stopped being theoretical. Bank failures that look routine, stablecoin balances frozen by an administrative decision, and a surveillance apparatus that has grown faster than any coin's privacy.

The rest of the book is the checking.

Cheers,

Todd Pitner

P.S. People learn differently: some visually, some by listening, some by doing. This book serves the visual learner. If you would rather listen, the whole book is recorded, chapter by chapter, at quietprivatemoney.com.

CONTENTS

Contents

PART ONE · THE PROBLEM WITH THE MONEY YOU HAVE

1. Two things money must do 7
2. What your bank account actually is 12
3. What Bitcoin became 21
4. Privacy is a property, not a feature 28

PART TWO · THE TEST

5. Thirty-two attributes 36
6. Scoring the field 42

PART THREE · HOW EPIC CASH IS BUILT

7. Mimblewimble, without the math 50
8. Fungibility by construction 57
9. Sound money on a schedule 64
10. Mining that no one controls 70
11. Small enough to run yourself 76
12. Sending it 82

PART FOUR · THE HONEST PART

13. What Epic Cash does not have 91
14. Why the gap is the opportunity 99

PART FIVE · LIVING WITH IT

15. Getting it, holding it, using it 107

APPENDICES

A. The thirty-two attributes, in full 116
B. Buying Epic Cash 120
C. Wallets and self-custody 124
D. Glossary 128
E. Sources and method 132

PART ONE

The problem with the money you have

CHAPTER ONE

Two things money must do

On the second of November 2026, a bank in Paraguay stopped working.

Ueno Bank had reported a capital ratio of fifteen percent in March, against a required minimum of twelve. On the second of November the central bank's board voted unanimously to begin resolution, because after adjustments to its books its solvency had fallen below half of what regulators require. The announcement came overnight. Cards stopped. Transfers stopped. The app displayed a message that customers photographed and posted: *we cannot display the balance.*

Three million people woke up owning something they could no longer see, spend or move. The guarantee that covers them is capped at roughly thirty-eight thousand dollars per depositor, and the first payout was announced at about sixteen hundred and seventy-five dollars, "as soon as possible." Eight executives were brought before prosecutors. The president of the bank resigned on the fifth.

Nothing about this was illegal, and nothing about it was unusual. It is what happens when the thing you call your money is not money but a promise from an institution, and the institution stops being able to keep it.

Hold that story for a moment. We will come back to it, because it is the clearest recent demonstration of the first of the two jobs

money has to do — and of what happens when a system that is supposed to do that job fails in a single night.

Money has two jobs, and they pull against each other.

The first is to **store value**: to hold what you earned today so you can spend it in ten years without discovering that a third of it quietly evaporated. The second is to **exchange value**: to move from you to someone else, quickly, cheaply, and without asking permission.

You have used money that was good at one and bad at the other your whole life, and you have probably never separated them in your head, because the money you use is presented as a single thing with a single set of properties.

Consider the dollar. As a medium of exchange it is close to perfect: accepted everywhere, settles instantly, costs nothing to use at the point of sale. As a store of value it has lost about eighty-eight percent of its purchasing power since 1971, which is not a conspiracy theory but an arithmetic consequence of a supply that grows and a target inflation rate of two percent that compounds. The Federal Reserve Bank of Chicago explained the mechanism in its own pamphlet in 1961, in language plain enough for a schoolchild: banks do not lend out the money you deposited, they create new money when they make a loan, and the deposit you still see in your account is a claim on a pool that has been lent several times over. That pamphlet's own example turns a thousand dollars of deposits into ten thousand dollars of money. In March 2020 the reserve requirement that limited this process was set to zero.

Now consider gold. As a store of value it has done its job for five thousand years. As a medium of exchange it is miserable: heavy, slow, verifiable only by destroying a piece of it, and impossible to send to another continent without an intermediary who will take a cut and ask questions.

Now consider Bitcoin, which was designed to fix both at once, and did fix one of them completely. Twenty-one million, no issuer, no permission, seventeen years of uptime against every prediction of its death. But the second job — exchange — has been steadily abandoned. Capacity is limited, fees are volatile, and the asset has been absorbed into the same financial system it was built to escape, with ETFs, custody arrangements and a price that now moves with the Nasdaq. As a store of value it has been extraordinary. As a way to pay a person, almost nobody you know has ever used it, and that is not because they are lazy.

Two jobs. One coin rarely does both well. That is the first thing this book asks you to hold in your head, and it is the reason the rest of the book is not a comparison of prices.

There is a third property, and it is the one that gets left out of these discussions, usually because the people having them are not the people it protects.

Money has to be **yours**. Not in the sense that a court would agree, but in the sense that no one can stop you using it.

The Paraguayan depositors owned their balances in every legal sense. It did not matter, because ownership of a bank balance is ownership of a claim, and a claim is only as good as the

institution's ability to honor it. When the institution stops, the claim stops, and the owner watches from outside.

This is not a South American problem, or a 2026 problem. It is the structure of deposit banking everywhere. In the United States, forty-three percent of bank deposits sit above the insured limit, which means that for nearly half of the money in the system the guarantee is partial and the rest depends on the bank staying solvent. The FDIC's fund holds about a dollar forty-eight for every hundred dollars it insures. Those are not opinions; they are the published figures, and they are worth knowing the next time someone tells you that your money is safe because it is insured.

So the standard this book will use has three parts, not two:

Store of value. Can more of it be created, and by whom?

Medium of exchange. Can you actually use it, at a cost that does not punish ordinary payments?

Sovereignty. Can anyone stop you holding it, moving it, or spending it — and if they can, what would it take?

Most money fails at least one of these. Most cryptocurrencies fail at least one of these. The interesting question — the question the rest of this book is about — is whether a design exists that satisfies all three at once, and what it costs to get there.

I should say what this book is not.

It is not an argument that you should buy anything. It contains no price target, no forecast, and no claim about what will happen to

any of these assets in any timeframe. Where it quotes a price, it is to show a relationship between sizes, not to suggest a direction.

It is not neutral either. I think the design described in Part III is the most complete answer anyone has built to the three questions above, and I will say so. But the book's method is to give you the questions first, show you the working, and let the conclusion arrive on its own. If you disagree with me at the end, you will disagree with an argument you can trace, not with a mood you caught.

And it is not finished. There is a chapter in the middle of this book, the thirteenth, that lists what Epic Cash does not have. It is the hardest chapter to write and the most important one to read, because any book that omits it is asking for trust rather than earning it.

The next chapter is about your bank. Not banks in the abstract — the specific mechanics of what happens to a deposit when the institution behind it stops being able to keep its promises, and why the system is built that way on purpose.

Then we will do the same thing to Bitcoin, which is less comfortable, because Bitcoin got more right than anything before it.

Only after that will we take out the scorecard.

CHAPTER TWO

What your bank account actually is

The message Ueno's customers saw on the third of November was not a glitch.

We cannot display the balance.

That sentence is the most honest thing a bank has ever said to its customers, because it describes the relationship accurately. What you call your balance is a number the bank keeps for you. When the bank stops keeping it, the balance does not become zero. It becomes unverifiable, and for every practical purpose those are the same thing.

The rest of this chapter is about how that number came to exist, what stands behind it, and why the answer is stranger than most people assume.

A deposit is a loan you made and forgot about

When you put money in a bank, you are not storing it. You are lending it.

The legal description is unromantic: the bank becomes your debtor, and you become its creditor. You do not own any particular banknote sitting in any particular drawer. You own a claim on the bank, payable on demand, subject to the bank's ability to pay.

This is not a technicality invented by critics. It is the structure, and it explains a great deal that otherwise looks inexplicable — why deposits are insured at all, why a bank failure is a legal process rather than a technical fault, and why the word "withdrawal" exists. You are withdrawing a claim, not retrieving a thing.

Compare it with the arrangement it replaced. If you put a gold coin in a vault, the coin remains yours. The vault company is providing a service — a building, a door, a guard — and if the company fails, the coin is still in the building. The failure is an inconvenience. With a deposit, there is no coin. There is a promise, and a promise is only as good as the party making it.

Every bank balance in the world sits on that foundation, including yours.

How the promise multiplies

The mechanism was described plainly by the Federal Reserve Bank of Chicago in a booklet called *Modern Money Mechanics*, published in 1961 and used for decades to train bank examiners.

> "The actual process of money creation takes place primarily in banks."

And, on the part that surprises people:

> "Of course, they do not really pay out loans from the money they receive as deposits. If they did this, no additional money would be created. What they do when they make loans is to accept

> promissory notes in exchange for credits to the borrowers' transaction accounts."

Read that second sentence again. The bank does not lend out your deposit. It creates a new deposit when it makes a loan, and the loan is the asset that balances it on the other side of the ledger. The booklet walks through the arithmetic with an example: deposit a thousand dollars, the bank keeps a hundred and lends nine hundred; that nine hundred is spent and deposited at the next bank, which keeps ninety and lends eight hundred and ten; and so on until the sequence runs out. A thousand dollars of original deposit becomes ten thousand dollars of deposits across the system.

Your thousand is still on the statement. Nine thousand more now exists beside it.

The booklet assumed a ten percent reserve requirement, which is where that example comes from. That assumption is history. On 26 March 2020 the Federal Reserve set the reserve requirement ratio to zero, and it has stayed there. The constraint that limited the sequence in 1961 no longer exists in the form the booklet described. Lending is still limited by capital rules, by the demand for loans, and by prudence — but not by a requirement to hold a fraction of deposits in reserve.

You can see the result in the aggregates. In 1971, the year the dollar's link to gold was severed, the United States money supply was about 686 billion dollars. It is now above 23 trillion. That is roughly thirty-four times as many dollars, against an economy that did not grow thirty-four times.

The consequence is not a secret and it is not a conspiracy. The Federal Reserve targets two percent inflation per year, publicly. At two percent, prices double in about thirty-five years, which means a dollar held without earning interest loses half its purchasing power within the working life of the person holding it. Measured from 1971, the dollar has lost about eighty-eight percent of what it could buy.

So: the supply of the thing in your account is a policy variable. It is set by people, on a schedule, with a stated target. That is the first half of the problem, and it is the half everyone argues about.

The second half is quieter, and it is the subject of the rest of this chapter.

What stands behind the claim

When a bank cannot pay, the question becomes what happens next, and the answer is a process rather than an event. Deposits are insured up to a limit, the institution is resolved, and customers are paid out according to a schedule they do not control.

In the United States the insured limit is 250,000 dollars per depositor, per bank, per ownership category. That number is reassuring until you look at what sits beside it. Forty-three percent of United States bank deposits are above the insured limit, which means that for nearly half of the money in the banking system, the guarantee is partial and the remainder depends on the bank remaining solvent. The fund that pays the insurance held about 1 dollar 48 cents for every 100 dollars it insured as of June 2026.

Those are the published figures, not an interpretation.

Now the timeline. In Korea, the statutory period to resolve a failed bank is 150 days. Regulators there have proposed shortening it to three or four days, which tells you two things at once: that the current period is long enough to hurt, and that the people who run the system know it.

Ueno is the case in front of us, and the numbers are worth reading slowly.

Three million customers. About 8.9 percent of all deposits in the Paraguayan banking system. A guarantee capped at 228.3 million guaraníes per depositor, per bank — roughly thirty-eight thousand dollars. A first payout announced at about 10 million guaraníes, roughly sixteen hundred and seventy-five dollars, "as soon as possible." A resolution that began with a unanimous board vote on the second of November, was announced overnight, and took the app, the cards and the transfers with it. Eight executives brought before prosecutors. The president of the bank resigned on the fifth.

In March, that institution had reported a total capital ratio of fifteen percent, against a required minimum of twelve. By the time the central bank adjusted the books, solvency was below half of the required minimum. Whatever the ratio measured in March, it was not a forecast of November.

And here is the detail that matters more than any of the others: the money inside those accounts included pension funds. Public money, held by people who had done nothing wrong, in an institution that had reported healthy capital eight months earlier, in a country whose central bank acted exactly as designed.

The system worked. That is the point. It worked, and three million people still could not see their balance.

Access is a permission, not a property

There is a second layer, and it operates every day rather than in a crisis.

Your ability to move money is granted by an institution and conditioned on its rules. Accounts are frozen, restricted and closed routinely — for sanctions matches, for anti-money-laundering flags, for patterns a model did not like, for documents that did not arrive. Most of the time the customer is guilty of nothing except being inconvenient to a compliance system optimized for its own risk rather than theirs.

This is not a scandal. It is the normal operation of a system that must, by law, know who you are and what your money is doing. The complaint is not that banks break their own rules. The complaint is that the rules make your access conditional, and the condition is not yours to satisfy once a decision has been made somewhere you cannot see.

Under stress, the permissions tighten. That is when people need their money most, and it is precisely when the system is least able to provide it — as Ueno's customers discovered at the moment the app stopped showing them a number.

One clear illustration from the same three weeks, outside banking entirely: on the 29th of October, two stablecoin issuers froze more than eight billion dollars of tokens, about 2.4 billion of it within hours, because addresses had been linked to an incident. The tokens were on a public ledger. The freeze did not require a

court, a hearing, or the holder's consent. It required an administrative decision by the issuer, and it took an afternoon.

A public address is a fixed target. So is a bank account. So, in the end, is anything that requires someone else's cooperation to move.

What "yours" would actually require

Strip away the institutions and ask what property in money would have to look like for the word *yours* to be true.

It would have to be **bearer**: held by whoever holds it, with no registry naming an owner.

It would have to be **final**: settled without a counterparty who can reverse it, freeze it, or fail.

It would have to be **permissionless**: movable without an institution deciding whether the movement is allowed.

And it would have to be **private**: not merely unrecorded in a database you cannot see, but unreadable by anyone who is not party to the transaction.

Physical cash satisfies the first three and fails the fourth in an age of serial numbers and cameras. It also fails at distance: you cannot send a banknote across an ocean in a minute, and carrying one across a border is a decision someone else gets to review.

Digital bearer money has been the design problem of the last forty years: how to make something that behaves like cash while existing as information. Bitcoin was the first serious answer, and it

answered the issuance question completely — a fixed supply, no issuer, no permission to participate — while leaving two parts of the problem open, which is what the next chapter is about.

What this chapter is not claiming

It is not claiming that banks are useless, or that the system is about to collapse, or that you should empty your account and bury the proceeds.

Banks do real work: they aggregate capital, they underwrite risk, they move money between people who never meet, and they do it at a scale and speed that has no historical equivalent. Most days the system works, and on the days it does not, most people are made whole.

The claim is narrower and harder to argue with. The money in your account is a claim on an institution, denominated in a supply that someone else controls, reachable only through permissions that someone else grants. That is true on the good days too. It is a description of the instrument, not a prediction about the bank.

Which brings us to the three questions this book keeps asking.

Store of value. Can more of it be created, and by whom?

Medium of exchange. Can you use it, at a cost that does not punish ordinary payments?

Sovereignty. Can anyone stop you holding it, moving it, or spending it?

The first question is a monetary policy question, and the answer for the dollar is: yes, by a committee, on a target, indefinitely.

The second is a settlement question, and the answer for the dollar is: yes, cheaply and universally, within the perimeter.

The third is a property question, and the answer for the dollar is: yes, at several layers, starting with the institution holding the number and ending with the state that charters it.

Bitcoin was built to answer the first question differently, and it did. It was not built to answer the third, and the last seventeen years have shown what happens when a transparent ledger meets institutions that would rather not see certain transactions.

That is where we go next.

CHAPTER THREE

What Bitcoin became

Start with what is true, because the rest of this chapter only earns the right to be read if this part is said properly.

Bitcoin is the most important monetary invention of the last hundred years. In January 2009 it demonstrated, for the first time, that a currency could exist with a fixed supply, no issuer, no company, no board and no permission — and that it could keep running while every institution that predicted its death went on to have a worse decade than it did. Seventeen years of uninterrupted operation is not luck. It is engineering.

It also answered the first of this book's three questions completely. **Can more of it be created, and by whom?** No, and by no one. Twenty-one million, enforced by consensus rather than policy, with a schedule that nobody has been able to renegotiate. Whatever else is said in this chapter, that achievement stands, and everything that follows is built on it — including the coin this book is about.

But Bitcoin answered one question, and the other two were left where they were.

Transparency was the point

The ledger is public. Every transaction is visible, every address is visible, every amount is visible, and the whole history is retained forever.

This was not an oversight. It is how Bitcoin verifies without trusting anyone: if everyone can check the books, no one has to be trusted to keep them. The design is elegant, and it worked.

The cost was built in at the same time. Transparency means the ledger is a permanent, public, queryable record of who paid whom. Addresses are pseudonymous, not anonymous — a distinction that has been commercially exploited for a decade by an industry that exists specifically to attach identities to them. Chain-analysis firms sell clustering, heuristics and attribution to exchanges, governments and anyone else with a budget. Once an address is linked to a person, everything that address ever did is linked to that person, permanently, including the transactions that happened before the link was known.

There is no mechanism to opt out. There is no version of Bitcoin where the history is not there. The record is the security model, and the record is the exposure.

A coin is not always a coin

Once history is public, history can be graded, and graded history becomes a price.

Coins that passed through a hack, a sanctioned address, or a market the authorities dislike are marked. Exchanges run screening on deposits and sometimes freeze or refuse them. In 2018, the United States sanctions list named Bitcoin addresses for the first time, which turned an analytical category into a legal one: an address you transacted with could place you, unknowingly, on the wrong side of a list.

The word for the property being lost is **fungibility** — the condition in which any unit is interchangeable with any other, so that a holder never has to ask where their money has been. Gold has it. Cash mostly has it. Bitcoin increasingly does not.

Andreas Antonopoulos, who spent his career explaining Bitcoin to the people who would use it, put the stake plainly: tainted coins are destructive, and if you break fungibility and privacy, you break the currency. That is not a critic of Bitcoin saying it. It is one of its most influential teachers.

The part that is harder to argue with

Everything above concerns the ledger. There is a second layer, and it concerns who produces the blocks.

Mining is the process that orders transactions, and a miner chooses which transactions to include. That is not a flaw in Bitcoin; it is what mining is. The question is whether anyone has ever used that discretion to exclude a payment, and whether the structure makes that easy or hard.

The record says: yes, and easier than the popular account suggests.

In May 2021, Marathon mined block 682,170 and left out transactions associated with sanctioned addresses, on purpose. After a backlash it reversed the policy four weeks later. In October 2023, the pool behind roughly fourteen percent of blocks, F2Pool, skipped sanctioned payments; its co-founder confirmed the filter publicly. In December 2024, nine F2Pool blocks left out sanctioned payments again, and other pools mined them later.

Nothing was stolen. The payments eventually settled. But a filter was applied at the point of production, by a private company, using information that every miner already has — and it was applied because the company decided to apply it.

What a filter needs is visibility, which Bitcoin provides by design, and a small number of willing pools. The second condition is closer than most people think. Three pools have mined a majority of all blocks in recent measurements — Foundry, AntPool and F2Pool. Around forty percent of hashrate has run through AntPool and its proxy pools. More than eighty percent of Bitcoin mining chips have come from a single manufacturer, Bitmain.

And the off switches have been pulled before, from outside the protocol entirely. In 2017, Bitmain's firmware contained a function that could remotely stop Antminers; it was removed after disclosure. In 2021, a mining ban in China cut Bitcoin's hashrate by about half in five weeks. In January 2022, an internet shutdown in Kazakhstan took out thirteen percent of hashrate. In 2024, the Texas grid asked miners to power down during a demand event and hashrate fell twenty-five percent in four days.

The network survived all four. It is remarkably resilient. But in each case, a government, a grid operator or a chip manufacturer decided how much of it ran, and for how long.

Why the base layer stayed small

The capacity question was argued in public for years, and the argument was settled on purpose.

Through 2015 and 2016 the network debated raising the block size limit — the cap on how much transaction data a block may

carry. Bitcoin kept the cap, activated SegWit in 2017, and pushed throughput to layers built above the chain. Part of the community disagreed hard enough to fork, and Bitcoin Cash exists because of it.

The decision was defensible on its own terms. Small blocks keep validation cheap, which keeps node operation within reach of ordinary people, which is what stops the network from becoming a service run by a handful of institutions. That is the same argument this book will make about chain size in a different context, and it deserves to be stated fairly here.

But the consequence is unavoidable. A base layer that settles a handful of transactions per second cannot be the layer where people pay each other. Payments move up to Lightning, Lightning requires channels, channels require liquidity and an online node, and for most users the practical path back to the base layer runs through a custodian. The design kept the node small and moved the payment experience into exactly the intermediaries the design was meant to remove.

That is not a contradiction unique to Bitcoin. It is the shape of the trade-off: you can have cheap validation or cheap payments on the base layer, and Bitcoin chose validation. The interesting question — the one this book keeps coming back to — is whether a different chain design can have both, which is what cut-through and a four-gigabyte ledger will be asked to demonstrate later.

What it became

None of this makes Bitcoin a failure. It makes it a success that drifted.

The drift is visible in what people do with it. Bitcoin was written as peer-to-peer electronic cash, and the phrase is in the title of the paper. What exists today is closer to a settlement asset and a speculative instrument: a store of value held through custodians, wrapped into funds, traded on venues, and increasingly correlated with the same equity markets it was designed to be independent of.

Consider the ordinary facts of ownership. Most holders do not self-custody. Most exposure runs through an exchange, a fund, or a brokerage account. The people who own Bitcoin in the way its author intended — keys on their own device, no intermediary — are a minority of a minority.

This is not hypocrisy. It is gravity. A scarce asset with a rising price attracts capital, and capital arrives through institutions, and institutions reintroduce exactly the intermediaries the design was built to remove. The asset did not fail. It was absorbed.

The price is real, and so is the achievement. Both things are true at once, which is why the argument has to be made carefully rather than loudly.

What this leaves open

Come back to the three questions.

Store of value. Answered. Twenty-one million, no issuer. Whatever happens next, this part is done, and everything after Bitcoin has to match it rather than improve on it.

Medium of exchange. Partly answered. It settles, but capacity is a handful of transactions per second, fees move with demand, and the base layer was never designed for a coffee. The answer

to this one has been pushed to layers above, which work, and which reintroduce custodians and channels.

Sovereignty. Answered at the issuance layer, less so at the production layer. No one can stop you holding a private key. Someone can decline to mine your transaction, and a handful of pools have. Someone can also read every payment you have ever made, which is its own kind of control — the kind that does not need to block anything to change your behavior.

So the design Bitcoin left unfinished is specific. It is not speed, and it is not supply. It is the combination of privacy and fungibility, at the base layer, without asking the user to opt in and without adding a second system that can fail on its own.

That last clause matters more than it sounds, and it is where the next chapter goes. Because the obvious way to add privacy to a transparent chain is to build a shielded pool beside it — a second mechanism, with its own cryptography, its own assumptions, and its own bugs. That approach has produced most of the privacy in this industry. It has also produced every major failure in this category since 2019.

There is another way to do it, and it starts by not having anything to hide in the first place.

CHAPTER FOUR

Privacy is a property, not a feature

Think about the last thing you bought with a card.

The receipt is gone, probably. The record is not. There is a row in a database with a timestamp, a merchant, an amount, a card number, and — through the card network, the acquirer, the issuer and the merchant's processor — a chain of custody that runs from your account to a till in a shop. Add the loyalty program, the delivery address, the phone that was in your pocket, and the record stops being a payment and becomes a profile.

None of that is a scandal. It is how the system pays for itself. But it is worth being precise about what has happened: the money moved, and a permanent, searchable, saleable description of the movement was created at the same time. You did not agree to that in any meaningful sense. You agreed to the terms of a service, which is a different thing.

This chapter is about why that matters, and about the two ways the crypto industry has tried to fix it — and what is wrong with both.

Three different privacies

The word "privacy" collapses several distinct things, and most arguments about it go wrong because the participants are talking about different ones.

Privacy of identity. Does anyone know that a given transaction belongs to you?

Privacy of transaction. Does anyone know the amount, the counterparty, or the timing?

Privacy of association. Can anyone link two of your transactions together and infer that they belong to the same person, even without knowing who that person is?

Bitcoin gives you a version of the first and almost none of the second or third. Your name is not on the ledger, but the ledger is a graph, and a graph with enough edges identifies its nodes. That is the entire business model of the chain-analysis industry: not breaking cryptography, but counting.

The two leading privacy coins take different approaches to this problem, and both are serious. It is worth describing each on its own terms before saying anything critical, because both have earned that.

Monero: privacy as the default

Monero made privacy mandatory and built it into every transaction from the start.

Ring signatures hide which of several possible inputs is the real one, so a spender cannot be identified from the inputs alone. Stealth addresses generate a one-time address for each payment, so a recipient's address never appears on the chain. RingCT hides the amounts, using commitments and range proofs so that the network can verify a transaction balances without

learning the values. Dandelion++ obscures where a transaction originated on the network.

The design is coherent, it has been running since 2014, and its defaults are the strongest in the industry: there is no transparent mode to fall back into, so there is no way for a user to accidentally leak. Monero's supply is also uncapped — 0.6 XMR per block, forever — which is a deliberate choice rather than an oversight, and a reader who values scarcity should know that this is one of the places where the two designs genuinely differ rather than merely rank.

What Monero costs is size and time. Every ring signature needs decoys, every output is retained, and the chain is roughly two hundred gigabytes for a full node, seventy for a pruned one. It is not a chain you run on a laptop you also use for work.

Zcash: privacy as an option

Zcash made a different bet, and a technically remarkable one.

Its shielded transactions use zero-knowledge proofs — the sender proves that the transaction is valid without revealing the sender, the recipient or the amount. The cryptography is among the most sophisticated ever deployed in a monetary system, and the team behind it has repeatedly advanced the state of the art, including moving the protocol away from the trusted setup that early versions required.

But Zcash is a dual system. There is a transparent chain, which behaves like Bitcoin's, and a shielded pool beside it. Privacy is something a user chooses, per transaction, and the choice is

visible: the network can see whether a transaction is shielded or transparent even when it cannot see what is inside.

This is a defensible design, and it has one large structural consequence: **optional privacy leaks.** A privacy feature that most people do not use does not protect the people who do, because the flows around them are visible, and the boundary between the two systems is where the analysis happens. The proportion of value inside the shielded pool has historically been a small fraction of the whole.

There is a second consequence, and it is the one this chapter is really about.

The problem with adding a second system

When you add privacy to a transparent chain, you are not changing the chain. You are building a second machine beside it — new cryptography, new assumptions, new code — and connecting the two.

That second machine has to be correct in a way that ordinary software does not, because its failures are silent. A bug in a payment app shows up as an error message. A soundness bug in a proof system shows up as nothing at all: valid-looking proofs for invalid statements, accepted by every node, potentially for years.

This is not a hypothetical objection. In 2026 the privacy-coin category had a bad year, and the failures were concentrated exactly where the added machinery lives — proof systems and shielded pools. Zcash disclosed a soundness vulnerability in its Orchard pool rated 9.3 on the CVSS scale, present since 2022

and patched within days of disclosure. Other privacy projects had comparable failures in the same period. The pattern is not that these teams were careless. The pattern is structural: every system that adds a privacy layer adds a surface where the layer itself can be wrong.

A privacy design that does not add a layer has a different risk profile. It is not that it is free of bugs — no software is — but the class of bug is different, because there is no separate proof system whose soundness is the only thing standing between a valid chain and an invalid one.

What fungibility actually requires

Step back from the cryptography and ask what the property is for.

Fungibility means that any unit is interchangeable with any other, so a holder never has to care where their money has been. That is not an abstract virtue. It is what makes money usable by ordinary people: you do not run a background check on a twenty-dollar bill, and if you had to, cash would stop working.

For a digital currency, fungibility requires that there be nothing to check. Not that checking is difficult — that there is no data to check with. If a coin has a history, that history can be graded. If it can be graded, it can be priced. If it can be priced, there is a market in clean coins and a discount on the rest, and the discount falls on people who did nothing wrong except receive money that had been somewhere.

So the requirement is stronger than "private if you choose." It is: private in a way that leaves no optionality to get wrong, and no

separate pool to be traced across.

The honest position on both

Neither Monero nor Zcash is a failure. Monero has held the strongest privacy defaults in the industry for over a decade and has been tested by real adversaries. Zcash pushed zero-knowledge proofs into production and funded research that the whole field uses. A book that dismissed either would be signalling its own unseriousness.

What can be said, accurately, is this: Monero's privacy is real but expensive to run and uncapped in supply; Zcash's privacy is optional and therefore partial, and its architecture carries the risk of the second system. Both are answers to the question of how to add privacy to a chain that was not designed for it.

There is a third answer, and it is the one this book spends its second half on: design the chain so that there is nothing to hide in the first place. No addresses, no amounts, no transparent mode, no shielded pool, no second machine.

That is not a small claim, and it is not enough to assert it. It has to be tested against everything else money has to do — supply, settlement, mining, governance, custody — and that is what the next part of the book does.

Before it can, though, the reader needs the test itself.

Where this leaves the three questions

Store of value. Bitcoin answered it. Supply is fixed, and the question is now about who else can match that and what else

they can offer alongside it.

Medium of exchange. Bitcoin partly answered it and pushed the rest upstairs. Monero answers it better at the base layer, at the cost of scale. Zcash answers it at roughly Bitcoin's level, with an optional privacy layer on top.

Sovereignty. Here the category splits. All three let you hold keys. Only one of them — Monero — makes every transaction private by default, and none of them removes spent data, which is why their chains grow forever and why a full node is a commitment rather than an afternoon.

Three questions, three partial answers, and no design that answers all of them at once. That is not a rhetorical setup. It is a description of the state of the art as of this writing, and it is the reason the next chapter does something unusual: instead of comparing coins, it defines what superior money would have to be, and only then looks at who meets the definition.

PART TWO

The test

CHAPTER FIVE

Thirty-two attributes

Every comparison of money you have ever read was decided before it started.

Not dishonestly, usually. Someone picks the five or six properties they think matter — scarcity, speed, privacy, adoption, security — weighs them by feel, and arrives at the conclusion their weighting implied. Then they write it up as though the weighting were obvious. It is not obvious. It is the argument, disguised as a preamble.

The alternative is to define the test first, in public, before anyone is scored — and then to apply it the same way to everything, including the thing you are hoping will win. That is harder than it sounds, because it means you can lose, and it means the reader can catch you.

This chapter is about a document that tried to do it that way.

Where the list came from

Todd Lewis, Epic Cash's lead engineer, and Todd Pitner, the author of this book, began with a thought experiment: what would superior money actually have to do, if you wrote the list down before looking at any coin? It became a series of videos comparing Epic Cash, Bitcoin, gold, Federal Reserve notes and central bank digital currencies against the same questions, and then a document, so that the questions could be argued with rather than just watched.

The list that came out of it is thirty-two attributes long. It is not a marketing document dressed as a framework; it is a framework that happened to produce a result its authors liked, which is a different and more dangerous thing. The distinction matters, and this chapter is largely about how to tell them apart.

Definitions before scores

The document does something that most comparisons skip. Before scoring anything, it defines each attribute — and the definitions are where the work actually happens.

Take three.

Private by default is defined as: all transactions are confidential, revealing neither addresses nor amounts nor IP. Notice what that excludes. A coin where privacy is a setting is not private by default, because the default is what most people use. A coin that hides amounts but shows addresses is not private by default either. The definition is doing the work that an adjective would otherwise be allowed to fudge.

Fungible is defined as trading two assets with no meaningful distinction between them. Then the document explains the mechanism it is describing: coins that melt down and reappear with no history. Again, the definition is precise enough to fail. A coin whose units can be graded fails it, no matter how the marketing reads.

Fair launch is defined as the team acting in good faith to assure fair access from the moment of mainnet — announced on ordinary channels, mineable by hardware people already own. That definition excludes a presale, a foundation allocation, a

venture round, and a quiet early unlock. It is a hard test, and it is meant to be.

This is the part of the document worth stealing even if you never buy anything. A defined attribute can be checked. An undefined one can only be asserted.

The thirty-two

They fall into five rough families — what money must do, who controls it, what it protects, how it is built, and how it reaches people — though the document presents them as a single flat list and scores them equally. Here they are in the document's own order.

#	ATTRIBUTE	#	ATTRIBUTE
1	Means of payment	17	Fair launch
2	Peer to peer	18	Publicly verifiable
3	Divisible	19	Trust minimized
4	Digital	20	Permissionless
5	Borderless	21	Immutable
6	Unit of account	22	Neutral
7	Private by default	23	Survivable
8	Decentralized	24	Eco-friendly
9	Fungible	25	Open governance
10	Surveillance resistant	26	Regulation friendly
11	Politically resistant	27	Durable
12	Censorship resistant	28	Scalable and fast

#	ATTRIBUTE	#	ATTRIBUTE
13	Confiscation resistant	29	Portable and mobile
14	Scarce	30	Available to the unbanked
15	Store of value	31	Widespread acceptance
16	Inflation resistant	32	Quantum future proofed

It is a list you are meant to fill in yourself, and that is the most important design decision in the document. A scorecard that cannot be used on a new entrant is a marketing table. A scorecard with a column left for you is a tool.

Why thirty-two, and not five

Because five is where the discretion hides.

Pick five attributes and any asset can be made to win: choose privacy, scalability, low fees, fair launch and community, and one answer emerges; choose liquidity, institutional adoption, regulatory clarity, brand recognition and settlement volume, and a completely different one does. Both lists would be defensible. Neither would be honest, because the choice of list was the argument.

Thirty-two is long enough that the list itself stops being a lever. It includes attributes that favor Bitcoin (brand, liquidity, settlement history), attributes that favor Monero (default privacy, ring signatures, a decade of adversarial testing), attributes that favor gold (durability, neutrality, five thousand years of monetary history), and attributes that favor nobody in particular but matter anyway (quantum agility, survivability, regulation friendliness).

A long list is not automatically a fair list. It is simply harder to rig without it showing.

The weakness of the method, stated plainly

There are two objections to scoring a coin this way, and both are fair.

The first is **equal weighting**. If all thirty-two count the same, then a coin that is excellent at trivia and mediocre at the things that decide whether money works can still score well. Anyone who has ever built a rubric knows this. The document does not weight, and it does not claim to; it counts. The reader is free to look at which attributes a coin missed and decide whether the misses matter.

The second is **self-scoring**. The authors defined the attributes, applied them, and published the result. That is exactly the structure of a rigged test, and the only defense is transparency: every definition is in the document, every tick is visible, and the reader can disagree with any of them in public. A rigged test hides its definitions. This one puts them on page two.

Which brings us to the result, and to the thing that makes the whole exercise credible.

The result the authors published

Epic Cash scores thirty-one out of thirty-two.

The attribute it misses is **widespread acceptance** — mass adoption, integrated anywhere people would use money. The framework's own authors, who built the list, who defined every

term, who are plainly sympathetic to the subject, scored it against themselves and published the miss.

That single empty box is worth more than the thirty-one ticks, and it is the reason this book can use the scorecard without embarrassment. A document that produced a perfect score would have told you what kind of document it was. This one told you what kind of question it was.

The next chapter runs the same test across the field — Bitcoin, gold, Monero, Zcash, a stablecoin and a fiat currency — and shows the working. Then Part III asks the only question the scorecard leaves open: what is actually inside the design that produces those ticks, and is any of it overstated?

Before we go on

If you take one thing from this chapter, take the method rather than the result.

Write down what money must do before you look at any coin. Define each property so precisely that an asset can fail it. Apply the list to everything, including the thing you already own. Then look at what is missing, and ask whether the missing thing is a flaw in the asset or a fact about the world that no design can fix yet.

That is the whole discipline. Everything else in this book is an application of it.

CHAPTER SIX

Scoring the field

Scoring is easy when the subject is someone else's coin.

It becomes uncomfortable the moment you apply the same list to the thing you already own, because the list does not care what you paid for it. That is the point of writing the definitions down first: they stop being negotiable when the result arrives.

What follows is the same thirty-two questions, applied to six assets. The judgments are mine and they are argued in the open. Where a cell is a matter of interpretation rather than fact, the text says so. Where you disagree, the definitions are on the previous pages and the arithmetic is on this one.

A note on scope. The original document scored Epic Cash, Bitcoin, gold, Federal Reserve notes and central bank digital currencies. I have added Monero, Zcash and a stablecoin, because those are the three things a reader of this book is most likely to be holding instead, and a comparison that omits them is not a comparison.

The count

ASSET	PASSES	WHAT IT MISSES THAT MATTERS MOST
Epic Cash	**31 / 32**	Widespread acceptance
Monero	**24 / 32**	Scarcity, scale, unit of account, adoption
Bitcoin	**20 / 32**	Privacy, fungibility, scale, unit of account
Zcash	**20 / 32**	Default privacy, fungibility, fair launch

ASSET	PASSES	WHAT IT MISSES THAT MATTERS MOST
Gold	15 / 32	Digital, divisible in practice, portable, scalable, verifiable
Stablecoin	12 / 32	Decentralized, scarce, private, permissionless, uncensorable
Fiat (USD)	11 / 32	Scarce, private, permissionless, censorship resistant, store of value

Counts are ours, applied to the definitions in the previous chapter; the full matrix, attribute by attribute, is in Appendix A. Disagreements belong in the margin.

Read the column, not the ranking. The number of ticks is a summary of the argument, not a substitute for it, and the assets at the top and bottom of this table are not comparable in the way a league table implies. What matters is which boxes each one misses, and whether those misses are the ones that would hurt you.

Bitcoin: the settlement layer that forgot to be cash

Bitcoin passes twenty. It wins the attributes that made it famous: scarce, decentralized, permissionless, immutable, trust minimized, publicly verifiable, fair launch, store of value, inflation resistant. Those are not small things, and no honest scoring can take them away.

It fails the privacy cluster almost completely. Not private by default, not fungible, not surveillance resistant, not confiscation resistant in any meaningful sense once an address is linked to a person. It fails the exchange cluster too: unit of account, scale, cost per payment.

The most interesting entry in its column is the one that reads as a compliment and functions as a criticism — **store of value, yes; medium of exchange, barely**. Bitcoin solved the problem of issuing money without an issuer and then became an asset people hold rather than spend. That is a legitimate outcome. It is not the outcome described in the title of the paper.

Gold: the honest answer that cannot move

Gold passes about fifteen, and it earns most of them the hard way. Scarce, durable, fungible in practice, neutral, no issuer, five thousand years of monetary history, and a record of holding value through every currency collapse of the modern era.

Then the digital column: not digital, not scalable, not portable, not verifiable without an assay, not divisible in any practical sense, and dependent on vaults and custodians the moment it leaves your hand. A gold coin in your pocket is money. A gold bar in a vault is a claim on an institution, which is the thing chapter two was about.

Gold is the strongest possible answer to "what is money that cannot be printed," and the weakest possible answer to "can I send it to my daughter in Osaka this afternoon."

Monero: the best defaults in the industry

Monero passes twenty-four and would pass more if the list weighted privacy, which it does not.

It wins every privacy attribute outright — private by default, fungible, surveillance resistant, censorship resistant, confiscation

resistant — and does so without asking the user to choose. Its ring signatures, stealth addresses and RingCT are the most battle-tested privacy construction in production. If the only question were "which of these can I spend without creating a record," Monero wins and the chapter ends here.

It misses scarcity, because its supply is uncapped by design: 0.6 XMR per block, forever. It misses scale: a full node is roughly two hundred gigabytes and grows with every ring signature. It misses unit of account and adoption, as all of these do. And it keeps spent data forever, which is the opposite of the property that makes a chain small enough to run on a laptop.

None of that makes it a bad coin. It makes it a coin with a different center of gravity: privacy maximalism at the cost of monetary hardness and light validation.

Zcash: the cryptography is not the problem

Zcash scores around twenty, and the arithmetic is unkind to it in a way that has nothing to do with the quality of its mathematics.

It fails **private by default**, because privacy is optional and the transparent chain is the default path for most transactions. It fails **fungible**, for the same reason — if some coins move through a shielded pool and some do not, the two are distinguishable. It fails **fair launch**, because twenty percent of the early issuance went to the founders over four years. It fails scale and eco-friendliness for the usual proof-of-work reasons, and adoption as everyone does.

What it passes, it passes impressively: scarce, decentralized enough, immutable, publicly verifiable, trust minimized at the

protocol level, and secure by construction in ways that took real cryptographic ambition to achieve. The Orchard pool is a genuine technical achievement.

The scoring simply records a structural fact: a privacy feature that most users do not use protects neither the users who do nor the ones who do not.

The stablecoin: convenience with an owner

The stablecoin scores about twelve, and every one of its passes is a usability attribute — means of payment, divisible, digital, borderless, portable, scalable, unit of account over short horizons, available to the unbanked in the sense that anyone with a phone can hold one.

It fails everything structural. Not decentralized: an issuer decides. Not scarce: supply expands when the issuer decides to expand it. Not private: transfers are visible and the issuer knows who you are. Not permissionless: addresses are frozen, routinely, in the billions of dollars, in an afternoon. Not censorship resistant, not confiscation resistant, not trust minimized, not immutable.

A stablecoin is a bank account with a public transaction history and an owner who can switch it off. That is genuinely useful for moving dollars across borders. It is not money in the sense this book has been using the word, and the eight billion dollars frozen in a single day in October 2026 is the cleanest demonstration available.

Fiat: the baseline

Eleven out of thirty-two, and most of those are usability. It works everywhere, divides, settles, prices, and is accepted by everyone. It is also not scarce, not private, not permissionless, not censorship resistant, not a store of value over a lifetime, and not something you hold — you hold a claim on an institution, denominated in a unit that institution's central bank can issue more of.

The score is not a criticism of the dollar's usefulness. It is a description of what the dollar is for: settlement inside a jurisdiction, at scale, with a central authority deciding the terms.

Epic Cash: thirty-one, and the box that is empty

Epic passes thirty-one of the thirty-two, and the one it misses is the one this book has been circling since the first page: **widespread acceptance**.

That is not a rounding error and it is not a technicality. An asset that nobody accepts is not yet money, whatever its properties. It is a design that has not been adopted, which is a different thing from a design that has failed — but the difference is only visible in hindsight, and no scorecard can settle it.

Every other row in the Epic column is a claim that chapter seven onwards has to justify. Private by default, with no addresses or amounts to hide. Fungible, because there is no history to grade. Scarce, at twenty-one million, on a schedule that converges with Bitcoin's in May 2028. Decentralized, mined across three algorithms on hardware people already own. Scalable, because

spent data is cut away and the chain is about four gigabytes after three million blocks.

Those are the claims. The next three chapters take them one at a time.

What the table cannot tell you

Two things, and both matter.

It cannot weight. Equal weighting is the fairest default and the least useful one, because it treats "durable" and "available to the unbanked" as equally decision-relevant. If you are a person in a country with capital controls, censorship resistance is worth more than divisibility. If you are a business settling invoices, speed and stability are worth more than neutrality. The table gives you the raw material for a decision; it does not make the decision.

It cannot measure the future. Widespread acceptance sits in the Epic column as a miss today. It sat in the Bitcoin column as a miss in 2011. A scorecard is a snapshot of a design and its adoption, not a forecast, and anyone who reads it as one is using it wrong.

With those caveats stated, the arithmetic is on the table. Epic Cash is the only asset here that satisfies the monetary, privacy and construction attributes simultaneously — and the only one whose missing attribute is a matter of time rather than architecture.

That claim now needs to be examined rather than asserted. Part III opens the design.

PART THREE

How Epic Cash is built

CHAPTER SEVEN

Mimblewimble, without the math

In 2016, an anonymous author published a short paper under a pseudonym lifted from a Harry Potter villain, describing a spell that ties an enemy's tongue so they cannot speak a secret.

The joke was apt. The proposal described a way to write a ledger in which there is almost nothing to say — no addresses, no amounts, no history of where a coin has been. The author never revealed themselves, the paper circulated as a curiosity, and then a small group of cryptographers turned it into software.

This chapter is about what that design actually does, in plain language, and what it costs. Both parts matter. Mimblewimble is the reason Epic Cash can claim privacy without a shielded pool and a small chain without pruning a database, and it is also the reason using it feels different from using Bitcoin in a way that took years to smooth out.

What a transaction has to prove

Strip a payment down and there are only two things the network must be convinced of.

That no money was created. The inputs are worth exactly what the outputs are worth, minus a fee.

That the owner authorized it. Whoever is moving the coins knew the private key that controls them.

Bitcoin proves both by showing everything: here are the inputs, here are the outputs, here are the signatures, check the arithmetic yourself. The proof is the disclosure. That is why the ledger is transparent, and it is why the ledger is a surveillance record.

Mimblewimble asks a different question: can you prove the arithmetic without showing the numbers?

Sealed envelopes that still add up

The tool is called a commitment, and the intuition is a sealed envelope with a number written inside it.

You write an amount on a slip of paper, seal it in an envelope, and put the envelope on a table. Anyone can pick it up and see that it is a valid envelope. Nobody can see the number inside without opening it. Now do that with every input and every output in a transaction, and add a second trick: the envelopes are constructed so that the arithmetic still works while sealed. The network can verify that the input envelopes and the output envelopes balance, without ever opening one.

Cryptographers call this a Pedersen commitment. The construction uses a blinding factor — a random number mixed in with the amount — so that two envelopes holding the same amount look different, and one envelope cannot be compared to another to guess what is inside.

The network ends up knowing exactly one thing: this transaction did not create money. It knows nothing about who sent it, who received it, or how much moved.

Why you cannot hide an amount and allow a negative

There is an obvious attack on the scheme as described. If the amounts are hidden, what stops someone from writing a transaction whose outputs are a large positive number and a large negative number, so that they balance but create money out of nothing?

The defense is a **range proof**: a cryptographic proof that each hidden amount falls within a valid range — no negatives, no absurdly large values — without revealing the value itself. This is where zero-knowledge proofs enter, and it is worth being precise about a distinction that confuses people: Mimblewimble is not free of zero-knowledge cryptography. It is free of a shielded pool. The proofs that keep the amounts honest are zero-knowledge proofs, and Epic uses a modern construction called Bulletproofs, which are compact and require no trusted setup — no ceremony, no toxic waste, no group of people who could collude to forge proofs if they kept a secret.

That last property matters more than it sounds. Several privacy systems have needed a multi-party ceremony to generate parameters, and a ceremony is a trust assumption, however well managed. Epic's range proofs do not require one.

No addresses, and what that changes

Bitcoin transactions reference addresses — destinations that persist, accumulate history, and can be clustered. Mimblewimble has no addresses in the ledger at all. Outputs are one-time objects that exist only as an entry, and the ledger keeps no link from an output to the person who can spend it.

This is the technical reason for the claim on the cover of this book. Amounts and identities are hidden, and there is nothing on the chain to look up or hand to an analysis firm. Linking some payments is possible only through costly, targeted surveillance of the network itself.

It also creates a problem that nobody expects until they try to use it: **if there are no addresses, how does a sender construct a payment to a receiver?**

Under the hood, both parties contribute to the transaction. The sender cannot build a valid transaction alone: each party proves its own output is valid, and the two wallets jointly sign the kernel. That means the wallets have to talk to each other, exchange a small amount of data, and sign.

This is the interactivity cost of Mimblewimble, and it is real. Early implementations handled it by passing a file back and forth. That works between two people who are both at a computer, and it is hopeless for buying coffee.

Which is why the practical question for any Mimblewimble chain is not "does it hide the amounts" — it does — but "how does the wallet handshake happen in the real world."

The relay, and why the Onchain Note exists

Epic's answer is a relay. Wallets can exchange the handshake data through an encrypted mailbox — the Epicbox relay — or over Tor, or as a file, or in the extreme as a message pasted into a chat. The sender addresses the payment to a mailbox identifier, the receiver's wallet collects it, and the transaction completes without either party being online at the same time.

That design decision has a visible consequence, and it is the single most practical thing to understand about using this coin.

Because there are no addresses on the chain, an exchange has nothing to credit when your deposit arrives. It cannot look at the ledger and see that a payment was sent to it, because the ledger does not say who received what. So it does something else: it gives you a number — an onchain note — and asks you to put that number in the payment. When the transaction lands, the exchange matches the note to your account and credits you.

Forget the note, and the exchange is holding funds it cannot attribute to anyone. This is not a quirk or a bug in the exchange. It is the direct, predictable consequence of a chain with no addresses, and it is why the guide in the appendix spends a page on a field your wallet labels "optional" and the exchange labels "required."

A privacy design that removes the address has to replace the address with something. The relay and the note are what replaced it.

Cut-through: why the chain is four gigabytes

The second half of Mimblewimble is about what the ledger does not keep.

In Bitcoin, every transaction ever made is stored forever. A coin that was spent in 2013 is still in the database, because the history is the proof.

Mimblewimble removes the history that no longer matters. When an output is spent, both the output and the input that consumed it

can be deleted — they cancel out — leaving only the proof that the transaction happened. This is called **cut-through**, and it applies within a block and, more importantly, across blocks.

What remains is a set of unspent outputs, a small header, and one **kernel** per transaction: roughly a hundred bytes that prove a valid transaction existed and paid a fee, without saying who, to whom, or how much.

The arithmetic is stark. Epic has produced more than three million blocks and its full chain is about four gigabytes — small enough to download on a home connection, validate on a laptop, and keep on a ordinary disk. A full Bitcoin node is measured in hundreds of gigabytes; a full Monero node is around two hundred; a Zcash node can pass three hundred. Epic is in a different category, and the difference is not engineering effort. It is a property of the ledger format.

This is the point where privacy and practicality stop trading off against each other. Cutting the spent data is what makes the chain small, and having no addresses and no amounts is what makes the remaining data useless to a surveillance analyst.

What this does not do

Mimblewimble hides what is inside a transaction. It does not hide that a transaction happened at all, and it does not, by itself, hide where it came from on the network.

If your wallet broadcasts a transaction from an IP address that is also associated with you, the cryptography has done its job and the network layer has given you away. This is why the design includes Dandelion++, which routes a transaction through a small

number of random relays before broadcasting it widely, making it difficult to connect the first sender to the eventual diffusion.

It is also why the wallet matters. Sending over Tor, or through the relay, changes what the network can observe. A private ledger with a careless network path is a locked door with the key under the mat.

And one more limit, honestly stated: the cryptography here is elliptic-curve based, like every other coin in this book. A sufficiently large quantum computer would break the commitments and the signatures alike. "Quantum future proofed" in the scorecard means the ability to change the scheme without rebuilding the monetary system, not immunity today.

What we now have

Three claims, and the machinery behind each one.

Private by default. No addresses, no amounts, no optionality — and the range proofs that keep the hidden amounts honest.

Fungible. Nothing on the ledger records where a coin has been, so there is nothing to grade, taint, or refuse.

Small. Cut-through removes the spent history, leaving unspent outputs and a hundred bytes per transaction.

The next chapter takes the second of those and asks the harder question: what does it actually buy you when a government, an exchange and a mining pool all decide they would rather not process your coins?

CHAPTER EIGHT

Fungibility by construction

In 2018, the United States Treasury named Bitcoin addresses on a sanctions list for the first time.

Nothing was seized. Nothing was banned. A list was published, and the list had addresses on it, and every institution that wanted to stay on the right side of American law now had a reason to look at those addresses and treat anything touching them as radioactive.

That is the whole mechanism. It needs a list, an address, and an intermediary with something to lose. No police, no court order against you personally, no proof that you did anything. Just a list, and a financial system that would rather not find out whether your coins are on it.

This chapter is about that machine, and about what happens to money that it cannot get hold of.

The machine has three steps

Refusing a coin requires three things, in order, and each one depends on data that the ledger provides.

It gets listed. An address associated with something the authorities dislike is published. The address is the handle. Without it, there is nothing to publish.

It gets frozen. Institutions that hold customer funds are obliged to check deposits against the list and freeze anything that matches. The freeze is not a punishment of the sender; it is the institution protecting itself.

It gets skipped. A miner choosing which transactions to include can decline the ones that reference listed addresses. This is the step most people do not know exists, and it is the one that turns a banking policy into a network-level event.

THE MACHINE OF REFUSAL, AND WHAT IT NEEDS TO RUN

	IT GETS LISTED	IT GETS FROZEN	IT GETS SKIPPED
Bitcoin	Addresses are public	Deposits match a list	Pools have filtered
Epic Cash	No addresses exist	Nothing to match	Nothing to select

Each step needs the one thing a private ledger withholds: the identity of the coins.

Remove any one of the three and the machine stops. And all three depend on the same precondition: **visibility**. To list an address, you must be able to see addresses. To freeze a deposit, you must be able to match it. To skip a payment, you must be able to tell one payment from another.

Bitcoin provides visibility by design, which is why all three steps have been exercised on it.

The record on Bitcoin

It is worth being precise, because "Bitcoin can be censored" is usually said as a slogan and the specifics are more interesting.

In May 2021, Marathon mined block 682,170 and deliberately left out transactions associated with sanctioned addresses. The policy was reversed four weeks later after public backlash.

In October 2023, F2Pool — the pool behind roughly fourteen percent of blocks at the time — skipped sanctioned payments. Its co-founder confirmed the filter publicly. In December 2024, nine F2Pool blocks left out sanctioned payments again; other pools mined them later.

None of those payments was destroyed. They were delayed, sometimes for hours, by a private company making a policy decision about which valid transactions it would include.

The structural facts make this easier than it should be. Three pools have mined a majority of all blocks in recent measurements. Around forty percent of hashrate has run through AntPool and its proxies. More than eighty percent of mining chips have come from a single manufacturer.

And a filter doesn't need the whole network — it needs enough pools to matter, which is a much smaller number.

What Epic changes is not the miners' behavior, it's their information

Here is the part that is easy to miss. Epic does not ask miners to be virtuous. Miners on any chain can decline to include a transaction; that is what mining is, and no design removes it.

What Epic removes is the ability to know which transaction to decline.

Nothing to list. There are no addresses in the ledger. When a payment is constructed, the output is a one-time object with no link to a person, an account, or a previous output. There is no handle to publish.

Nothing to flag. Amounts are hidden behind commitments, and spent outputs are cut away, so there is no provenance to grade, and nothing on the chain to look up or hand to an analysis firm. Linking some payments is possible only through costly, targeted surveillance of the network itself.

Nothing to skip. A miner can verify that a block balances and that the signatures are valid. It cannot see who sent what to whom, which means a miner asked to exclude certain coins would have to exclude everything or nothing. Selective censorship requires selection, and selection requires visibility.

This is what the one-page essay means by the line "unstoppable is a claim, fungible is a design." No network can promise that a hostile government will not try. What a design can do is make the attempt require the one thing a privacy chain withholds: the identity of the coins.

Better than cash, in one specific way

Physical cash is the closest thing most people have to fungible money, and it is not perfectly fungible either.

Banknotes carry serial numbers. A note used in a robbery can be recorded, and a note recorded can be watched for. Banks and retailers mark notes, check them, and occasionally refuse them. In countries with capital controls, the physical form is the first thing controlled, because it is the easiest to count at a border.

Epic's position is that a coin should carry no serial number at all — not a hidden one, not an encrypted one, but none, because the output that would carry it is deleted when it is spent.

That is a stronger claim than cash makes, and it is the precise reason the word "fungible" belongs in the design section of this book rather than the marketing section.

What fungibility does not protect you from

This is where honesty costs something, so it is worth paying in full.

Fungibility makes coins interchangeable. It does not make **you** invisible to the institutions you choose to deal with.

An exchange can still refuse your deposit. It can require identification, decline your jurisdiction, close your account, or decide your transaction pattern looks wrong. It simply cannot do those things on the basis of where your coins have been, because it cannot see where they have been. The refusal becomes about you rather than about the coin — which is a meaningful improvement, and not the same as immunity.

A government can still ban the exchange, block the website, or make it illegal to hold the asset. It cannot blacklist your coins, because there is no list that would work.

And a miner can still leave your transaction out of a block for any reason it likes, including no reason at all. It just cannot do so selectively across a category of coins, because it cannot tell which coins are which.

So the accurate claim is narrower and stronger than "uncensorable." Epic removes the *coin-level* handle that censorship needs. It does not remove the human one.

Why this cannot be added later

The tempting conclusion is that any chain could adopt this if it wanted to. It cannot, and the reason is structural rather than political.

Fungibility requires that the ledger contain no history. A chain that has been transparent for a decade has a decade of history that cannot be un-recorded — every address, every amount, every link, permanently public and permanently copyable. You can add a private mode on top, and you will still have the old data underneath, and a coin that moves from the private mode into the transparent one becomes traceable again at the boundary.

You cannot retrofit the absence of a record. You can only build a chain that never made one.

That is the argument for starting over rather than upgrading, and it is the argument this book has been building since chapter four. Epic launched in 2019 with no addresses and no amounts in the first block. There is no earlier version of it with a transparent history, because there is no earlier version.

What this is worth

Put the two halves together and the design claim becomes concrete.

Bitcoin can be filtered because every payment is visible, and it has been filtered, by named pools, in named blocks, on the public record. Epic cannot be filtered in the same way because the information the filter requires does not exist — not concealed, not encrypted, absent.

Whether that matters to you depends on what you think the next decade looks like. If you believe financial systems will remain open, permissionless and neutral toward holders of privacy coins, the property is a nicety. If you think sanctions lists will keep growing, that exchanges will keep tightening, and that "where did these coins come from" will become a standard question at every on-ramp, it is the whole game.

The book is not going to pretend the first view is unreasonable. It is going to point out that the second view requires no prediction at all — it is a description of the direction of travel since 2018.

Next: the money itself. Supply, emission, and the uncomfortable arithmetic of a coin that is already mostly mined.

CHAPTER NINE

Sound money on a schedule

Here is the number that makes people uncomfortable: about ninety-two percent of all the Epic Cash that will ever exist has already been mined.

A Bitcoin holder hearing that will usually say some version of *so there is nothing left to earn.* A reader who has been paying attention to the last eight chapters will ask a better question: why would a coin designed in 2019 compress the emission of a coin designed in 2008?

The answer is in the design brief, and it is unusually candid.

Turning the clock back

Bitcoin's emission schedule was a bet that scarcity plus time would produce money. It worked. It also meant that anyone born after 2010 arrived to a table where most of the food had already been eaten, and the remaining portions were being distributed over the next century at a rate that halves every four years.

Epic's first five eras emit roughly ninety-seven percent of the total supply in about nine years — compressing twenty years of Bitcoin's emission into a shorter window. The 2019 paper used a phrase for it that no marketing department would have written: *a chance to turn back the clock for those who missed out on the spectacular rise of Bitcoin.*

That is an honest statement of intent, and it carries an honest cost, which this chapter will get to.

The mechanism

Blocks arrive every minute. Each block pays a fixed reward, and the reward steps down across thirty-three eras, in a pattern that will look familiar to anyone who knows Bitcoin's halvings.

The difference is the pace. Bitcoin halves every 210,000 blocks, which at ten-minute blocks is roughly four years. Epic halves on a schedule that compresses the early, high-reward period, so the first five eras account for the overwhelming majority of supply and the later eras stretch out toward the cap.

The cap is twenty-one million, and the divisibility is eight decimals: one Epic divides into one hundred million units, each called a freeman in the protocol's own vocabulary — the same structure as a satoshi, deliberately.

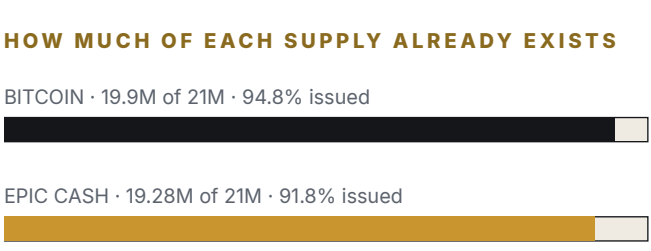
Then, in May 2028, something happens that no other coin does.

The Singularity

On or about the twenty-fourth of May 2028, Epic's cumulative supply draws level with Bitcoin's.

From that point, the schedule changes character. Epic adopts Bitcoin's halving cadence — reward halves every four years — while continuing to produce a block every minute rather than every ten. The two curves then track each other toward their respective caps, and the difference in circulating supply stays roughly constant rather than compounding.

The argument for doing this is not sentimental. It is that scarcity is easier to believe when it has a reference point people already trust. A holder does not have to model an unfamiliar emission curve to understand where Epic's supply will be in 2036; they can look at where Bitcoin's was in 2016 and add the gap.



Both chains are near the end of their issuance. Epic reaches Bitcoin's supply on 24 May 2028.

The argument against is that the schedule is the claim, and the claim only means something if the software enforces it. Which it does — but a reader should know that this is a property of the consensus code, not a promise from a company. There is no company.

The levy, disclosed in full

Every coin has to fund its own bootstrapping, and the industry's usual instruments are a premine and a venture allocation: supply handed to insiders before the public ever participates.

Epic used a different instrument, and it is worth describing precisely because it is the thing a skeptic will attack first.

A small, protocol-defined, time-limited share of block rewards — called the levy — was paid toward development and infrastructure during the bootstrap years. It began at 8.88 percent

of block rewards for the network's first 120 days, then declined on a fixed annual schedule: 7.77, 6.66, 5.55, 4.44, 3.33, 2.22, and 1.11 percent, where it stands today.

It ends at block height 4,377,600, in late December 2027 — months before the Singularity — by consensus rule. Cumulatively it totals 1,280,478 EPIC, approximately 6.10 percent of terminal issuance. Miners have received no less than 91 percent of emission at any point in the network's history, and currently receive about 98.9 percent of it.

Compare that to the canonical alternative: Zcash's founders' reward was twenty percent of issuance for the first four years, structured as a return for the investors who financed a company. Epic had no investors to repay. Its levy expires rather than compounds, and it was set before the first block was mined.

None of which makes it free. Six percent is six percent, and the honest way to state the difference is this: the levy is a funding tool that ends, not a profit center that renews — and the numbers are published so that a reader can decide whether that is acceptable rather than being told it is.

What front-loading costs

Two costs, and both are real.

The security budget fades early. A proof-of-work chain pays for its security through block rewards, and those rewards fall over time by design. Epic's rewards fall sooner than Bitcoin's, which means the transition to a fee-funded security model arrives earlier — and fee revenue depends on usage, which depends on adoption. A chain with low fees and low usage has a thin security

budget. That is the trade for a supply curve that reaches its cap sooner.

The counter-argument is the one the design makes: security also depends on how many people can mine and validate, and a chain that ordinary hardware can mine and a laptop can validate does not need the same budget as one secured by industrial facilities. Whether that holds at scale is an open question, and this book will not pretend otherwise.

The "early" argument is about price and adoption, not emission. When someone says you are early to Epic Cash, they are not saying you are early to the mining rewards — those are mostly distributed. They are saying the market has not yet priced the design. Those are different claims, and only the second one is available here.

What scarcity does and does not do

A fixed supply is necessary for sound money and nowhere near sufficient.

The dollar has no fixed supply and functions as money everywhere. Gold has a fixed-ish supply and functions as money in a shrinking number of places. Bitcoin has a fixed supply and functions as a store of value more than a currency. Supply is the property that makes saving rational; it is not the property that makes spending possible.

Epic's monetary policy answers the first question in the scorecard — *can more of it be created, and by whom* — with a cleaner answer than almost anything else in the field: twenty-one

million, no issuer, a schedule in the consensus code, and a levy that expires before the supply curve flattens.

The other two questions are answered by other parts of the design, and the next chapter takes the one that people underestimate: who actually produces the blocks, and what happens when the answer is "a lot of ordinary people with gaming PCs."

The number that is easy to get wrong

One more piece of arithmetic, because it appears in every argument about this coin and is usually confused.

Epic's market capitalization at the time of writing is roughly fifteen million dollars, against a circulating supply of about nineteen million coins. A price of roughly eighty cents reflects a market that has priced the design at a rounding error compared to the category leaders.

That is either a mistake or an opportunity, and no chapter can tell you which. What the chapter can tell you is what the supply does next: it does not inflate, it does not surprise, and in May 2028 it lines up with the most credible monetary schedule in the industry.

The rest is adoption, which is the one line on the scorecard that no design can fill in.

CHAPTER TEN

Mining that no one controls

Bitcoin's mining layer has three facts that are hard to hold together with its reputation.

Three pools have mined a majority of its blocks in recent measurements. Around forty percent of hashrate has run through one pool and its proxies. More than eighty percent of the chips doing the work have come from a single manufacturer.

None of that is a scandal. It is the natural end state of a single-algorithm proof-of-work network: the reward goes to whoever buys the most efficient hardware, the most efficient hardware comes from one supplier, and the pools that coordinate the largest hashrate win the most blocks. Bitcoin still works. But the claim that "no one controls it" has to be understood as *no one controls the protocol* rather than *no one controls the production*.

Epic's answer to this is the most unusual part of its design, and the least discussed.

Three algorithms, one hardware class each

Epic produces blocks with three different proof-of-work algorithms, each aimed at a different kind of machine:

RandomX, for general-purpose CPUs — the chips already in every laptop, desktop and workstation.

ProgPoW, for commodity graphics cards — the hardware in hundreds of millions of gaming PCs.

Cuckatoo31+, for specialized hardware — retained at a small share.

The block production is allocated 48 percent, 48 percent and 4 percent respectively.

The purpose of the split is not novelty. It is that a miner cannot dominate Epic by being excellent at one thing, because two-thirds of the blocks cannot be reached with the hardware that wins at the third. An industrial ASIC operation, however efficient, competes for four blocks in a hundred.

The allocation is enforced, not encouraged

Here is the detail that separates this from every other "multi-algorithm" coin that quietly became single-algorithm in practice.

At each block height, Epic's consensus policy engine — called Feijoada — fixes which algorithm the next block must use, drawn from the policy's proportions. Miners do not choose. The mix is a property of the chain, not of miner behavior, and it holds whether miners like it or not.

Without that enforcement, the split would collapse. Miners follow profitability, profitability follows hardware prices, and hardware prices follow whatever is cheapest to produce at scale. A voluntary three-algorithm network becomes a one-algorithm network as soon as one algorithm pays better, which is precisely what has happened to other projects that tried this.

Difficulty is tracked per algorithm rather than globally. Each new block retargets the difficulty of its own algorithm from the preceding sixty observations of that algorithm's blocks, so the three lanes do not contaminate each other's adjustment. At the target mix that is roughly a two-hour lookback for RandomX and ProgPoW, against Bitcoin's fixed two-week epoch.

The 48/48/4 split itself was set by a network upgrade in June 2023. The framework is open-ended: algorithms can be added and proportions rebalanced by consensus if the hardware landscape changes. The numbers are an instrument, not a doctrine.

Why 4 percent of ASIC is not a contradiction

The obvious objection: if ASIC resistance is the goal, why allow ASICs at all?

Because fighting them has a poor record. Every coin that promised ASIC resistance and depended on it for its security has either forked repeatedly, watched its hashrate collapse when a manufacturer shipped a machine anyway, or ended up trusting a small group of developers to keep changing the algorithm faster than silicon can be designed.

Epic takes the other route: give specialized hardware a small, fixed seat at the table and keep the other two seats for hardware that already exists in billions of homes. The ASIC operator is a participant rather than an existential threat, and the network does not depend on winning an arms race it would eventually lose.

That is a design judgment, not a law of nature. It is at least an answer to the question, rather than a slogan that dissolves the

first time someone tapes a hundred mining boards together.

The honest part: latent is not the same as realized

The tempting claim here is that Epic is secured by six and a half billion devices — every phone, laptop and gaming PC on earth.

The 2026 technical paper explicitly refuses that argument, and the refusal is worth quoting because it is the kind of honesty that makes the rest of a document credible:

The installed base of CPU and GPU devices worldwide is roughly 6.6 billion, but that base "is not committed defensive hashrate, and this paper does not treat it as such: realized security is only the hashpower that mining rewards actually attract."

That is the correct way to think about it. A billion laptops that could mine are not security. They are potential. What secures the chain today is the hashpower actually pointed at it, and that number is a function of the reward, the electricity price, and whether anyone finds it worth doing.

A reader should hold the design claim and the realized number separately: the design makes it *possible* for ordinary hardware to participate, which is a structural property. Whether ordinary people actually do is an empirical question, answered only by looking at the network.

The cost of a deterministic schedule

There is a vulnerability in this design, and the same technical paper names it.

Because Feijoada decides which algorithm must produce the next block, the chain's liveness depends on miners for that algorithm showing up. If the required algorithm's hashpower disappears — because it became unprofitable, because a jurisdiction banned it, or because someone deliberately withheld it — the chain can stall even if the other two algorithms have plenty of miners.

A probabilistic design would let the remaining miners carry on. A deterministic one cannot, by construction.

The mitigation is that the three algorithms run on different hardware in different places, so a single cause is unlikely to remove all three. But "unlikely" is not "impossible," and a book that pretended otherwise would be doing what this book has criticised everywhere else.

What this buys, stated without inflation

Not "unhackable." Not "unstoppable." Not "no one can attack it."

What it buys is narrower and more useful: **no single hardware vendor, and no small group of pools, can decide what gets mined on this network.** The production of blocks is distributed across hardware classes by consensus rule rather than by market share, and the mix survives miner preferences because miners do not get a vote on it.

For a chain whose entire argument rests on nobody being in charge, that property is not decoration. It is the difference between a claim about governance and a claim about reality.

The next chapter takes the other half of that reality: what it costs to verify the chain yourself, and why four gigabytes matters more

than any of this.

Three numbers worth watching

If you want to check these claims rather than take them, three figures do most of the work.

Hashrate per algorithm. The policy fixes 48/48/4. Reality is whether miners actually show up in all three lanes. A lane that is thin relative to its allocation is a lane where the chain depends on fewer participants than the design assumes — and it is visible in any block explorer that labels the algorithm.

Node count. Not miners: nodes. Miners produce blocks; nodes decide whether to accept them. A chain that is cheap to validate attracts validators who do not mine, and those are the people who enforce the rules when a miner tries something. Four gigabytes is what makes that number plausible.

Fee revenue. As the block reward declines, fees become the security budget. Watching fee revenue against hashrate tells you whether the transition to a fee-funded model is happening or stalling — the single most important open question about any proof-of-work chain's long-term security, including this one.

None of the three is a price. All three are checkable today, by anyone, without asking permission.

CHAPTER ELEVEN

Small enough to run yourself

"Decentralized" is the most abused word in this industry, and the abuse is almost always the same: the claim is about the protocol, and the reader assumes it is about the power.

A network can be architecturally decentralized and operationally concentrated at the same time. Bitcoin is the clearest example: no one controls the rules, and a few pools produce most of the blocks while a shrinking number of operators hold the archive. The protocol is decentralized. The practice is drifting.

The number that decides which of those is true is not hashrate. It is how many people can check the books themselves.

What a node actually does

A node is a computer that downloads the chain, validates every rule, and refuses anything that breaks them.

Miners produce blocks. Nodes decide whether to accept them. If a miner produces a block that pays itself too much, or double-spends, or violates the emission schedule, every honest node rejects it and the miner has wasted electricity. That is the entire enforcement mechanism of a proof-of-work currency, and it only works if enough independent people are running the software to make rejection meaningful.

This is why chain size matters more than almost any other technical metric. A chain that requires a data-center-grade machine to validate is a chain whose rules are enforced by the sort of operator who can afford one. A chain that fits on a laptop is a chain whose rules are enforced by anyone who cares.

The arithmetic

Epic's complete archival history — every block since September 2019, more than three million of them — occupies approximately four gigabytes.

Bitcoin's is roughly 750 gigabytes. A pruned Bitcoin node can run on seven to ten gigabytes of disk, but only after downloading and verifying all 750 first, and the archive it depends on stays roughly two hundred times larger than Epic's entire history.

Monero's full chain is around two hundred gigabytes. Zcash's node data is in the same range, having passed three hundred.

Four gigabytes is the size of a photo library. It downloads over a domestic connection in an afternoon, syncs on a laptop that was mid-range five years ago, and lives on a disk you already own.

This is not an engineering achievement so much as a property of the ledger format. Bitcoin stores every transaction forever because the history is the proof. Mimblewimble deletes what no longer matters: when an output is spent, the output and the input that consumed it cancel and are removed, leaving a small kernel that proves a valid transaction happened without saying anything about it.

Seven years and three million blocks later, the result is measured rather than projected. That is the difference between a scalability claim and a scalability fact.

What you can still check

Deleting history raises an obvious question: if the chain forgets, what stops someone from inventing money?

The answer is in the kernels and the commitments. Every node can verify that the sum of all outputs equals the sum of all inputs plus the scheduled emission — that the money supply is exactly what the protocol says it should be at this height. It can verify that no output is spent twice, because the set of unspent outputs is complete even though the spent ones are gone. It can verify every signature.

What it cannot do is reconstruct who paid whom in 2021, because that data no longer exists anywhere. That is not a side effect. It is the privacy property and the size property arriving as one thing.

There is an asymmetry here that the 2026 technical paper states rather than hides: Epic's default sync verifies the full state but skips historical header proof-of-work outside a recent window, where Bitcoin's default replays full validation from genesis. Epic offers a full header-validation mode by configuration. A reader comparing "validation" across chains should know that the defaults are not identical.

The exception the book has to mention

Epic's operational record has one interruption, and it belongs in this chapter rather than in a footnote.

In March 2021 an attacker exploited a value-overflow flaw — the same class of bug as Bitcoin's CVE-2010-5139, which in 2010 minted 184 billion counterfeit bitcoin before being caught and rolled back. Epic's response followed the same precedent: patch the flaw, invalidate the exploited history by rolling back to a pre-exploit block, and resume. The chain was paused for several weeks.

Two things should be said about that, and neither cancels the other.

The first is that a rollback is a real centralization event. Someone decided which history would stand, and the network followed — because at that point in its life there was no realistic alternative. A reader who values immutability should treat this as a scar, not a rounding error.

The second is that the alternative was worse: a chain carrying billions of counterfeit coins, which is the outcome Bitcoin itself refused in 2010 for the same reason. Every proof-of-work currency has this choice in its history or in its future. Epic's is public, dated, and documented.

The incentive nobody talks about

Running a node does not pay.

There is no reward for validating, no yield, no token. People run nodes because they want to verify their own transactions without

trusting someone else, because they run a business that accepts the coin, or because they believe the network should have more independent validators than it currently does.

That means node count is a volunteer number, and volunteer numbers are fragile everywhere. Every small chain has the same structural weakness: miners are paid, node operators are not, so the ratio drifts toward miners. Epic is not exempt from this, and no honest chapter can claim otherwise.

What Epic can claim is that the barrier is low enough that the decision is available to an ordinary person. That is not the same as saying many ordinary people make it. Node connections have been observed from more than 130 countries as of July 2026 — and the technical paper is careful to call that a proxy for geographic reach measured by geolocated IP addresses, not a count of users or a measure of decentralization.

What this is worth

The argument of this book is that money should be verifiable by the people who hold it. That argument is either supported by a chain you can download or it is a slogan.

Four gigabytes means a person with a laptop and an evening can verify, independently, that:

The supply is exactly what the protocol says it is. No coin has been spent twice. The block that paid a miner was valid. Nobody changed the rules while they were not looking.

That is what "governed by no one" means in practice. Not that no one is in charge — someone is always trying — but that no one

can change the books without every validating machine on the network noticing and refusing.

Running one

The practical requirements are modest, and worth stating plainly because most people assume they are not.

A machine with a few gigabytes of free disk, a domestic internet connection, and the patience to let it sync. The software is open source and free. There is no account, no registration, no permission, and nothing to configure beyond a data directory and, if you want it, a Tor proxy.

What you get for that is a node that answers to you. It validates your own transactions before your wallet broadcasts them, it tells you the real state of the chain rather than a block explorer's summary of it, and it participates in the only mechanism that makes "governed by no one" more than a phrase — the collective refusal of the machines that validate to accept a block that breaks the rules.

What you do not get is money. There is no reward, no yield, no incentive program. If that sounds like a weakness, it is: it is the weakness every proof-of-work network shares, and the reason node count is always lower than the design would like.

Everything else in this book rests on that. Which is why the next chapter is about the thing you do with a verified chain: send a payment to another person, and have it arrive in a minute for about a cent.

CHAPTER TWELVE

Sending it

Everything in this book so far has been about whether the money is sound. This chapter is about whether it works.

The test is not philosophical. It is this: can you pay another person, on the other side of the world, in under a minute, for about a cent, without asking anyone's permission and without creating a record that outlives the sale?

Bitcoin cannot do that. Monero can, at a higher cost in size, and its recipient need not be online. Zcash can, but only on its shielded side. A stablecoin can, as long as the issuer has not frozen the address.

Here is what it looks like on Epic.

A payment, step by step

You open the wallet. You paste or scan the recipient's identifier. You type an amount. You press send.

The wallet builds the transaction jointly with the recipient's wallet — that handshake from the last chapter — and broadcasts it. A block is produced within about sixty seconds. The payment is in it, and from the recipient's point of view it has arrived.

The fee is around a cent.

There is no address on the chain pointing at either of you. There is no amount visible. There is no record of where the money was before, because the outputs it came from have been cut away.

What remains is a kernel: a hundred bytes proving a valid transaction occurred, paid a fee, and balanced.

That is the whole experience. It is not more complicated than a bank transfer, and in most of the ways that matter it is less.

Why a minute matters

Bitcoin's ten-minute block time means a payment is confirmed in ten minutes at best, and exchanges typically wait for three or six confirmations before crediting — half an hour to an hour, in practice.

Epic produces a block every minute, so one confirmation is one minute. That is ten times Bitcoin's confirmation cadence, and it changes what the currency is good for. A ten-minute wait is acceptable for settling a large transfer between businesses. It is not acceptable for buying lunch, which is why Bitcoin's payment use migrated to intermediate layers that reintroduce custodians.

A minute is not instant. It is close enough to instant that the ordinary use of money — paying a person, in a shop, for a service — becomes practical on the base layer rather than in a second system.

For larger amounts, the recipient can wait for more confirmations. Two blocks is two minutes. Three is three. The trade-off between speed and certainty is a dial rather than a fixed cost.

The fee, and why it stays small

The fee exists to compensate the miner who includes the transaction and to make spam expensive. It is not a market

mechanism for allocating scarce block space, because block space is not scarce at current usage.

At about a cent, the fee is a rounding error on any purchase that matters. It is also, importantly, not set by bidding: the wallet computes it from the transaction's size and the network's current fee policy, so congestion does not push it up. Its dollar cost moves with the coin's price.

This is one of the quiet advantages of a chain that keeps its history small. Bitcoin's fees rise when demand for block space exceeds supply; Epic's blocks are not full, and if usage grew, the design's answer to capacity is a path to scaling rather than an immediate fee auction.

That path is a claim, not a fact. At current volumes, the fee is a cent and the blocks are mostly empty. What happens at a thousand times the volume is a question the next few years will answer.

The network layer, which is where privacy is won or lost

A private ledger with a careless network path is a locked door with the key under the mat.

When your wallet broadcasts a transaction, the first node that sees it can, in principle, connect it to your IP address. That is true of every cryptocurrency, and it is the layer where most privacy failures actually happen — not in the cryptography, in the metadata.

Epic's design addresses this in three ways.

Dandelion++ sends the transaction along a randomised "stem" path through a few nodes before broadcasting it widely in the "fluff" phase. An observer watching the network cannot easily tell which of the stem nodes originated the transaction, so the link between you and your payment is blurred at the moment it matters most.

Tor can be used for the wallet's connection, so the node does not see your IP at all. Node operators can also exchange onion addresses, routing the stem traffic end to end over onion connections.

The relay handles the handshake separately from the broadcast, so the two pieces of metadata — who is paying and where the transaction entered the network — are not tied together in the same message. The relay itself still sees which identifiers talk to each other, and from which addresses, unless the connection runs over Tor. It separates the two facts from each other; it does not make you invisible.

Each of these is a mitigation rather than a guarantee. Traffic analysis is a real discipline and a determined adversary with global visibility is a real threat. But the design assumes the network layer is hostile, which is more than most chains do.

What can go wrong

Four things, and they are worth knowing before you send anything.

The recipient's wallet has to be reachable. Because the payment requires a handshake, the recipient either needs their wallet online, or needs the relay to hold the half-built transaction

until they collect it. The relay solves this for ordinary use, but it is a dependency: if you are sending to someone who has never set up a wallet, the first payment is a conversation, not a click.

The deposit method. Because there are no addresses, an exchange has to be told which account a payment belongs to, and it does that one of two ways. The older one runs through a relay: the exchange gives you an identifier and a number — the onchain note — and you put the note in the wallet's note field. Leave it out and the coins arrive with nothing to attribute them to; the exchange holds money it cannot credit, and recovering it is a support ticket. The newer one puts your account in the delivery string itself: the exchange runs a wallet listening over HTTP, your account number is already inside the deposit URL, and there is no note to forget and no relay to be down. Where an exchange offers both, take the URL. Of the two, the note is the single most common way a beginner loses access to their coins on this chain, and that risk is a consequence of the privacy design rather than a defect in it.

The wrong identifier. A mistyped relay address is refused before anything is sent: the wallet checks the address's checksum, its version bytes and its key before it posts anything. If no one answers a valid address, the payment never completes and the wallet returns the coins. Check the address anyway, and send a small amount first. There is no chargeback, no support desk, and no reversal.

Confirmations still matter. A minute is one block. For a payment of any size, wait for more. A transaction in a single block can, in principle, be reorged out, and the recipient who hands over

goods after sixty seconds is taking a small risk for a small convenience.

The honest state of the software

Wallet software here moves faster than print can follow, so this page names the strongest option at the time of writing rather than for all time. The most capable wallet on this chain today is **BT Wallet**, open source, made by BT LABS, whose founder is a long-time Epic Cash developer, and published at epiccash.btlabs.uk/releases. It runs on Windows, Linux, macOS, Android and iPhone, from the Android and Apple app stores or as a signed download, and every file there has a SHA-256 checksum to compare against BT LABS' signed list before you run it. It does considerably more than the applications the project's own download page still points at: invoices you send to a paired contact, who approves and pays or declines while you can withdraw until it is paid; contacts and pairing, so a payment goes to a person rather than a pasted string; payment links; a build with Tor and a build without it; a Kill switch mode; and support for running your own node, which the wallet sets up and manages for you.

Whatever you install, the first payment should be small and the seed should be on paper. Those two rules survive every change of software, which is more than any version number can say.

What does not change is the shape of the problem. A wallet holds your keys. It needs a node to tell it the state of the chain, and either it runs one or it trusts somebody else's. Neither arrangement is free, and the choice is worth making deliberately rather than by default.

Whether that option is worth taking is contested, and the objection deserves to be quoted rather than paraphrased. A developer who reviewed this chapter, on the wallet-node arrangement:

> "My personal opinion of the wallet node thing has always been that it creates more trouble than it offers value. Users expect it to be instantly available, and many failed transactions have been due to the local node still syncing, which takes a while. The other issue is more technical: local nodes start and stop with the wallet being open. This adds more traffic between nodes, because it adds to the peer list, then the node becomes unavailable, and a public node must search for a new one, because that local node peer is gone. Nodes should run on a separate device from a wallet and run 24/7, or don't bother."

That is a fair summary of the case against, and I do not have a good answer to it. BT LABS' answer is that the wallet sets the node up and manages it, so the handling is theirs rather than yours — a reasonable answer, and not quite the same thing as infrastructure. A reader who wants the strongest guarantee should still run a node on a machine that stays on, and point the wallet at it.

None of this software is as polished as a bank app, and anyone who tells you otherwise has not used both. The interactivity requirement means the onboarding conversation is longer than "here is my address" — it is "here is my identifier, and here is how the wallet will complete the payment." That is a real friction cost, and it is the price of a design with no addresses.

Whether that price is worth paying is the question this book has been building toward. The answer depends on what you think the

alternative costs — and the alternative, for most people, is a bank account that can be frozen and a payment record that can be sold.

The point of all of it

Money that cannot be sent cheaply is not money. It is an asset you hold and hope.

The design work in the previous five chapters exists to make this chapter possible: a payment that is private by default, settled in a minute, priced at a cent, verified by a chain you can run yourself, and impossible to filter because there is nothing in it to filter.

That is a complete payment system, and it is running today. What it does not have is enough people using it — which is the subject of the next chapter, and the only line on the scorecard that is still empty.

PART FOUR

The honest part

CHAPTER THIRTEEN

What Epic Cash does not have

A book that spends twelve chapters building a case and then stops is asking you to trust it.

This chapter exists so that it is not doing that. Everything the previous chapters argued is true as far as I can verify it, and none of it changes the fact that Epic Cash has serious problems. They are listed here in order of how much they should worry you, with the numbers attached, and with what would have to happen for each one to stop being a problem.

One: almost nobody uses it

The scorecard's empty box, in numbers.

Epic's market capitalization was about fifteen million dollars at the time of writing, against roughly nineteen million coins in circulation. Zcash is around twenty-one billion. Monero is around eleven billion. The entire category is small; Epic is a rounding error inside it.

Market cap is not usage, so here is usage. Merchant acceptance is thin and concentrated — a handful of businesses, some community projects, a payment processor integration or two. Exchange listings are few: NonKYC is the deepest market, with a small presence elsewhere. There is no published count of nodes, only observations from more than 130 countries, which measures reach rather than adoption.

This is the one attribute on the list that no design can fix, and it is the one that matters most. Money that nobody accepts is not yet money. It is a design that has not been adopted, which is a different thing from a design that has failed — but the difference is only visible in hindsight, and no chapter can settle it for you.

What would change it: one integration that matters. A payment processor, a marketplace, a merchant with real volume. Adoption is not a smooth curve; it is a series of doors, and Epic has not walked through the first one.

Two: the liquidity is shallow enough to hurt

This one is not theoretical, and I have the order books.

On the EPIC/USDT market, the best ask has held around four coins — about three dollars — with a spread near half a percent. On EPIC/XMR, the best ask has been around six coins, roughly five dollars, at a similar spread. A buyer wanting a thousand dollars of Epic is not buying a thousand dollars of Epic at the quoted price; they are walking up the book and paying progressively more.

That cuts both ways. A seller of size gets less than the screen says, and anyone who needs to exit quickly pays for the privilege. The thinness also makes the price easy to move, which means the quoted market cap is a softer number than it looks.

What would change it: market makers. Depth is a business decision by someone with capital and an incentive, and until there is one, every buyer and seller is trading against a shallow book.

Three: there is no consolidated audit

This is the claim I would not let into the book without qualification, and I want to be precise about what is and is not true.

Epic's cryptography does not come from nowhere. The Mimblewimble design, the Bulletproof range proofs and the secp256k1-zkp libraries have been examined publicly, repeatedly, across the projects that share them. That inheritance is real and checkable.

What does not exist, as far as I have been able to establish, is a single consolidated third-party audit of the Epic protocol — a named firm, a date, a scope, and a public report that a skeptic can read. A marketing sheet in circulation refers to eight independent audits of the building blocks. I have not been able to locate the list, and a book that printed the number without one would be doing the thing this book has criticised in every other project.

What would change it: the list. Firm, date, scope, link. If it exists, it belongs in an appendix with citations, and the sentence becomes the strongest in the book. If it does not, the honest formulation is the one above: reviewed lineage, no consolidated audit.

Four: the tools are behind the design

The design is elegant. The software is improving and is not yet easy.

The interactivity requirement — the handshake between two wallets — is the structural cost of having no addresses, and it

shows up as friction everywhere: onboarding, receiving a first payment, running a shop, integrating with an exchange. The relay solves the worst of it, at the cost of a dependency that has to be running.

The mobile wallet is a light wallet, which means it trusts a node. The desktop wallet is not a full node either, though it can run one — optional, a setting you turn on, never the default. Turn it on and the wallet verifies rather than trusts, at the price described in chapter twelve: a slow first run, and a peer that comes and goes with the lid. Neither is a bank app, and comparing them to one is the fastest way to lose a non-technical user.

What would change it: time and users. Tooling improves when there are enough people hitting the rough edges to justify the work. This is the same chicken-and-egg that adoption has, wearing a different hat.

Five: there is no company, which cuts both ways

There is no Epic company and no marketing budget, and no one to call. A declining levy does fund development — the foundation output the consensus code pays out — and it ends on schedule.

That is the design, and chapter nine argued it is a feature. It is also the reason development moves at the speed of volunteer effort, the reason exchange integrations happen slowly, and the reason a journalist looking for a spokesperson finds a Telegram group instead of a press office.

The bootstrap levy that funded early development ends in December 2027. After that, the protocol has no funding mechanism at all. Whether the ecosystem can sustain itself on

donations, grants and commercial self-interest is an open question, and nobody should pretend otherwise.

What would change it: sustained independent funding of the kind other projects get from a foundation or a company — without reintroducing the capture that a foundation brings. Nobody has solved this.

Six: access can disappear, and has for others

Privacy coins have been delisted by exchanges in several jurisdictions, and the regulatory direction of travel is not friendly. Epic is a small asset with thin listings; if two of its venues decide the compliance cost is not worth it, the asset becomes much harder to buy and sell.

This is not a flaw in Epic's design. It is a fact about the environment every privacy coin operates in, and it belongs in the honest column because a reader planning to hold this asset should know that the on-ramp is the fragile part.

In November 2026 the Treasury's FinCEN did exactly that to MEXC Global — not by banning a coin, but by ordering every U.S. financial institution to refuse funds moving to or from that exchange, under section 9714. The stated grounds were sanctions evasion and money laundering, and no privacy coin was needed to justify the order. That is the lever worth understanding: the asset is rarely the target, the on-ramp is. A venue that cannot touch a dollar cannot easily be used to buy anything with one, and if your coins are sitting on that venue when the order lands, someone else's conduct has reached your

balance. The defense is not a better coin. It is holding your own keys, so that the only person who can lose your money is you.

What would change it: nothing Epic controls. This one is a function of politics, and the design's answer to it — that you can hold your own keys and transact peer to peer without an exchange — is real but partial.

Seven: the record, including the part that is uncomfortable

In March 2021 the chain was rolled back to remove a value-overflow exploit, and it was paused for several weeks while that happened. Chapter eleven covers the mechanics; here is the cost.

A rollback is a centralization event. Someone decided which history would stand. The precedent is Bitcoin's own in 2010, and the alternative was a chain carrying counterfeit coins — but a reader who values immutability is entitled to treat this as a scar rather than a footnote, and I do.

There is also the reputational residue of the years since: a price that fell a long way from its earlier highs, a community that has argued with itself in public, and a persistent assumption among people who have not looked closely that the project is dormant. It is not dormant. It is small, which looks similar from a distance.

What would prove this book wrong

A book that cannot be falsified is propaganda. Here is what would falsify this one.

If the privacy were broken in practice. Not a theoretical weakness — a demonstrated deanonymization of ordinary transactions at scale. If that happened, the central claim collapses and most of the book with it.

If the supply were not what the code says. If a value-overflow class bug returned, or if the commitments failed to enforce the cap, the monetary argument dies.

If adoption stayed flat for another decade while a rival solved privacy with better tools. The thesis is not "Epic eventually wins." It is "the design is sound and the adoption is a matter of time." If the second half is false for ten more years, the first half is an interesting museum piece.

If a soundness bug hit Epic's own range proofs. The 2026 failures in this category were proof-system failures. Epic uses Bulletproofs, which have a long record, and "long record" is not "immune."

None of these has happened. All of them could.

Why this chapter is not the end of the argument

Because "not yet" and "cannot" are different words, and the distinction is the whole book.

Epic Cash does not have adoption. That is a fact about the present. It does not follow that it cannot have adoption, and the argument for why it might — the design, the verification, the moment — is what the previous twelve chapters were for.

What the book will not do is pretend the gap is small, or that the missing pieces are somebody else's fault, or that a good design

is owed a market. It is not.

The next chapter is the one that has to earn the optimism back: why a gap this large, in a category this early, is the thing that makes the asset interesting rather than the thing that makes it doomed.

CHAPTER FOURTEEN

Why the gap is the opportunity

Here is the field, drawn to scale.

All the money in the world — cash, checking, savings, money market funds, across every country — is roughly one hundred and fifty trillion dollars. All the gold ever mined is about thirty trillion. Bitcoin is one and a half trillion. Zcash is twenty-one billion. Monero is eleven billion. Epic Cash is fifteen million.

Drawn as squares, Epic is invisible at the first zoom. It takes two enlargements to find it.

THE FIELD, DRAWN TO SCALE (LOGARITHMIC)



Market value, November 2026. Equal distances are multiples of ten, not equal dollars.

The temptation at this point is to do the arithmetic: if Epic were merely the size of Monero, it would be about seven hundred and

thirty times larger; if it were the size of Bitcoin, a hundred thousand times. Every one of those multiplications is arithmetically correct and intellectually worthless, because none of them is an argument. A number that large is not a forecast. It is a fantasy wearing a spreadsheet.

So this chapter is not going to do that. It is going to do something harder and more useful: explain why the peer group has been wrong the whole time.

The wrong comparison

For four chapters the scorecard has compared Epic to Bitcoin, gold, Monero, Zcash, a stablecoin and the dollar. That was necessary — those are the alternatives a reader might hold — but it has left an impression that needs correcting.

It made Epic look like the smallest competitor in a small category, which is true and misleading. The category is not the competition.

Monero and Zcash are not the field. They are two answers to a question that most of the world's money has not even been asked. The field is the one hundred and fifty trillion dollars sitting in accounts that are not private, not scarce, and not yours in any sense that survives a bank resolution.

If private, fixed-supply, self-custodied money is going to matter, it will not matter because it took market share from Monero. It will matter because some fraction of that hundred and fifty trillion decides it wants a property that bank deposits do not have.

The same three questions, applied to the field

Chapter two asked three questions of money, and the whole book has been scored against them.

Can more of it be created, and by whom? For the dollar: yes, by a committee, on a target, indefinitely. The supply of the world's reserve currency is a policy variable, and the policy is two percent inflation per year, which halves purchasing power in about thirty-five years.

Can you use it, cheaply? Yes — better than any alternative in history, inside the perimeter. That is not a small achievement, and it is why the dollar wins.

Can anyone stop you holding it, moving it, or spending it? Every day, at several layers, and the freeze in Paraguay and the eight billion dollars of stablecoins frozen in an afternoon are what the answer looks like when it is applied.

The point is not that the dollar is bad. The point is that it fails two of the three questions structurally, for reasons that no reform can fix, because reform would mean giving up the properties that make it useful to the people who issue it.

A hundred and fifty trillion dollars of money failing two of three tests is the largest addressable question in this book. Epic's market cap is not a measure of that question. It is a measure of how few people have noticed it is being asked.

Why the question is live now

Four pressures, none of them speculative, all of them dated.

Bank failures that look ordinary. Ueno was not a scandal in the sense of fraud; it was a solvent- looking institution that stopped being solvent and locked three million people out in three days. The system worked as designed, and the design still meant people could not see their own balances.

Stablecoin freezes at scale. More than eight billion dollars of tokens frozen by two issuers, much of it within hours, because an administrative decision was made about addresses. That is a demonstration, in public, that a public ledger is not the same thing as permissionless money.

The financialization of Bitcoin. ETFs, custody arrangements and a price that now moves with equity markets have made the largest cryptocurrency a very good store of value and a very poor private payment rail. That is not a criticism of Bitcoin; it is a description of what happens when capital arrives through institutions.

Machines that transact. This one is early and worth watching. If the next decade of commerce includes software agents buying things on behalf of people, those agents will need to move value without publishing a balance sheet to every counterparty. A transparent chain gives a competitor a free look at your strategy. A private one does not.

None of these pressures guarantees anything. Together they explain why the question — can private money work at scale — is being asked now rather than in 2015.

The honest case against

The gap is only an opportunity if the small thing can become large, and most small things do not.

The base rate for crypto assets is failure or irrelevance. Thousands of coins with good designs and no adoption have stayed that way, and the ones that survived usually did so on distribution rather than design. A design advantage with no distribution advantage is a beautiful thing in a museum.

There is also a specific version of this risk for Epic. Its adoption problem is not just "nobody knows"; it is that the easiest places to use it — exchanges — are exactly the places most exposed to the regulatory pressure that privacy coins attract. The design's answer, peer-to-peer settlement, requires both parties to have wallets and to complete a handshake, which is a high bar for ordinary commerce.

So the honest statement of the opportunity is not "this will certainly grow." It is narrower: **the question is large, the answer is well built, and the price of being wrong is small relative to the price of the question being answered.**

What asymmetry actually means

The word is overused, so here is the precise version.

If you buy a small amount of a small asset and it stays small, you lose what you put in, or most of it. That is a bounded outcome. If the question it answers becomes urgent — if private settlement becomes something a meaningful number of people and businesses need — the asset does not need to capture the field

to be worth many times what it is now. It needs to be the thing that answers the question.

That is an asymmetric bet in the technical sense: bounded downside, uncertain but large upside. What it is not is a prediction, and the fact that a bet is asymmetric does not make it likely. Most asymmetric bets lose. The reason to take one is not that it will probably work; it is that the loss is survivable and the question is worth answering.

Which is why this chapter will not give a number. Any number would be a forecast dressed as arithmetic, and the book has spent thirteen chapters insisting that claims be checkable.

What you are actually buying

Strip away the price and the market cap and the multiples, and here is the position.

You are buying a claim on a specific answer to a specific question: can money be private by default, fixed in supply, settled in a minute for a cent, and verifiable by anyone with a laptop — without a company, a foundation, or a permission.

Everything in Part III is an argument that the answer is yes, and Part IV is the honest accounting of what it cost: adoption, liquidity, tooling, audit, access.

The gap between the design and the adoption is the entire investment case, and it is also the entire risk. There is no way to have one without the other. A coin that had already been adopted would not be available at fifteen million dollars, and a coin

available at fifteen million dollars is available at that price because the market does not yet believe it will be adopted.

The reader's decision is not "will this go up." It is "do I want exposure to the question, and is this the best-built answer to it that I can verify myself."

That is a question a book can help with. It is not a question a book can answer.

What comes next

The last chapter is practical: how to get it, hold it, and use it without losing it — the wallet, the words, the exchange, the note that is labelled optional and is not.

Everything before this was the argument. The last chapter is the part you can act on.

PART FIVE

Living with it

CHAPTER FIFTEEN

Getting it, holding it, using it

Everything before this was the argument. This chapter is the part you can act on.

It is deliberately procedural, because the failures in this space are almost never philosophical. People lose coins to a missing memo field, a mistyped address, a seed phrase photographed into a phone that was later replaced, and an exchange that closed while the balance was still on it. None of those require an adversary. They require an ordinary Tuesday.

Four things, in order: buying it, getting it off the exchange, keeping it, and spending it.

Buying it

Epic is listed on a small number of exchanges. NonKYC is the deepest market and the one this book uses for its worked examples, because it lists Epic against both a stablecoin (USDT) and Monero (XMR).

The two pairs are not the same trade.

EPIC/USDT prices the coin in dollars, which makes the arithmetic simple and the mental accounting easy. It puts a stablecoin in the path, which means an issuer sits between you and the coin for the duration of the trade.

EPIC/XMR keeps the stablecoin out entirely, which is the coherent choice if the reason you want Epic is that no one can freeze it. It costs you a conversion in your head, because every price is quoted in Monero, and it typically carries a wider spread.

Neither is wrong. Pick the one that matches why you are buying.

Three practical rules apply to both.

Use limit orders. The order books are thin — the best ask has held single-digit coins on both pairs — and a market order walks up through several levels paying progressively worse prices. A limit order at or near the touch fills quickly and never surprises you.

Price from the book, not from the chart. Click the row you want and let the exchange fill in the number. It removes a whole category of rejected orders and mispriced fills.

Keep fee headroom. The trading fee is charged on top of the order value. If you spend your entire balance, the ticket can reject for insufficient funds. Keep a few percent back.

The deposit method, and the field that is not optional

This is the single most common way a beginner loses access to their coins, and it deserves its own section rather than a bullet.

Because Epic has no addresses, an exchange cannot look at the chain and see that a payment was sent to it. It settles the question one of two ways. The first runs through a relay: the exchange gives you a number — an onchain note — and asks you to include that number in the payment, then matches the note to your account and credits you. The second puts your account

number inside the deposit URL itself, so the exchange knows whose payment it is from the string you paste.

Where both are offered, use the URL. There is no note to leave out and no relay to be down, which removes both failure modes at once.

If you are on a deposit page that only does the relay method, the note is required, whatever your wallet says. The wallet labels that field **optional**; the exchange's page labels it **required**. The wallet is describing the protocol field; the exchange is describing its own need. Both are telling the truth, and only one of them will get your money credited.

If you leave it blank, the funds arrive and cannot be attributed to anybody. The exchange holds money it cannot give to you, and recovering it depends on a support process you do not want to rely on.

Write the note down when you first generate a deposit address. It stays the same for that address. Keep it with the address, not in a screenshot folder.

Getting it off the exchange

The only coins that are yours are the ones you control the keys to. Everything on an exchange is a promise, and chapter two is about what promises are worth.

First, set up the wallet. BT Wallet, from BT LABS at epiccash.btlabs.uk/releases or from the app store on your phone. Check a downloaded file against the SHA-256 on that page before you run it. Write the twenty-four words on paper — not in

a note, not in a password manager you have not tested, not as a photograph. Verify the backup by restoring it before you put anything in it. If this wallet received coins before 2025, choose "Scan from the start" when restoring, or the wallet will restore an empty balance.

Second, send a test amount. Ten dollars, or the smallest amount that clears the withdrawal minimum. Confirm it arrives, confirm you can see it, confirm the note was credited if it went to an exchange.

Third, send the rest. Withdraw to the wallet's identifier. Wait for confirmations. Then, if the amount matters, point your wallet at your own node, so the balance it shows comes from a chain you checked yourself.

Withdrawal fees are small — around a dollar on most venues — but not zero. Consolidating several small buys into one transfer is worth doing.

Keeping it

Six rules. They are not complicated and almost every loss in this space is a violation of one of them.

The words are the money. Anyone who has the twenty-four words has the coins, no matter what the wallet says. Anyone who does not, does not — no support desk, no developer, no court can restore them.

Never type them into anything except your wallet during a restore. Not a website, not a chat, not a support form, not a helpful person on Telegram who offers to check them for you.

Keep two copies, in two places, offline. Paper or stamped metal. A house fire should not be a total loss, and neither should a burglary.

Test the restore. A backup you have never restored is a hope. Restore it once, on a spare device, before the balance is large enough to hurt.

Do not keep the whole balance in one place. Two wallets with separate seeds is a meaningful improvement over one perfect wallet.

Remember that the wallet application is not the money. Wallets get abandoned and replaced; the words are the asset, and any compatible wallet restores them.

Using it

Sending is covered in chapter twelve at the level of principle. At the level of practice, three things.

The handshake. Your wallet and the recipient's have to exchange data for the payment to be built. The wallet handles this through the relay: you paste the recipient's identifier, the wallet does the rest, and the recipient collects it. If the recipient is online and using the same wallet, this is nearly as fast as any payment app.

The confirmations. One block is about a minute. For a coffee, that is enough. For a car, wait for several, and for anything between strangers, treat it like any settlement: time is the collateral.

The receipt problem. Every payment leaves a kernel in a public block, and explorers exist — but they show that a payment's

kernel was confirmed, not who paid or how much. To show a merchant the amount, share the transaction from your wallet. Your wallet's history is the record, and the merchant's node is theirs. For a business, this means the point-of-sale process has to record the transaction on both sides rather than pointing at a URL. It is a real integration cost and the ecosystem is still building for it.

When something goes wrong

A deposit never arrives. Open a support ticket with the transaction ID, the amount, the time and the onchain note if there was one. On a privacy chain, the exchange genuinely cannot find your money without your help.

You forgot the onchain note. Contact the exchange immediately with the transaction ID. Recovery is possible — the funds are in their custody and identifiable from their side — but it depends on their process and goodwill.

You sent to the wrong identifier. There is no reversal and no recovery. This error has no remedy, and it is the reason the test transaction rule exists.

You lost the seed. Nothing can be done. This is not a support failure; it is the design working as intended.

The one page version

If you remember nothing else from this book, remember these five sentences.

Buy with a limit order, priced from the book, and keep a few percent back for the fee.

Write the onchain note down with the deposit address, because the wallet says optional and the exchange says required.

Send a small test before you send anything that matters.

The twenty-four words are the money; two copies, offline, in two places, tested once.

And the only coins that are yours are the ones you can move without asking anyone.

Everything else — the scorecard, the mining, the Mimblewimble, the entire argument of this book — is worth exactly as much as your ability to do those five things correctly.

The quiet conclusion

On the second of November, three million people opened an app and were told that their balance could not be displayed.

Nothing in this book can change what happened to them. But everything in this book exists because of the question they must have asked themselves that morning: what did I actually own?

The answer, for almost everyone reading this, is a claim — on an institution, denominated in a supply that someone else controls, reachable only through permissions someone else grants. That is true on the good days too. It is a description of the instrument, not a prediction about the bank.

Epic Cash is an attempt to answer that question differently, and this book has tried to show its work rather than assert its

conclusion: thirty-two attributes defined before anything was scored, six assets judged against them, a design opened up and examined, a list of failures stated in full, and a gap named plainly rather than explained away.

What you do with that is not something a book can decide. It can only try to leave you with something more useful than an opinion — a test you can run yourself, on any asset, without asking anyone's permission.

That is the whole of it. The rest is patience.

APPENDICES

Reference, and how this was made

APPENDIX A

The thirty-two attributes, in full

The scorecard in chapter five compresses thirty-two attributes into a grid. This is the same test, expanded, so a reader can argue with a specific line rather than the whole table.

Each row is a question. If you disagree with the verdict, the disagreement is now narrow enough to be useful.

A note on the word *yes*. It means the property holds in the base protocol, not that it holds for every user in every circumstance. Someone who buys EPIC on an exchange and leaves it there has a custodial claim on a private ledger, which is a different object from coins in a wallet they control. Several of these attributes are lost at that boundary, and the loss is the exchange's business model, not Epic's design.

#	ATTRIBUTE	WHAT IT ASKS	EPIC
1	Means of payment	Can you pay someone with it directly?	Yes
2	Peer to peer	Can you pay without an intermediary?	Yes
3	Divisible	Can it be split into ordinary amounts?	Yes
4	Digital	Does it exist natively online?	Yes
5	Borderless	Does it cross borders without permission?	Yes
6	Unit of account	Can prices be set in it?	Yes
7	Private by default	Is privacy the default?	Yes
8	Decentralized	Is control spread beyond any one party?	Yes
9	Fungible	Is one unit interchangeable with another?	Yes

#	ATTRIBUTE	WHAT IT ASKS	EPIC
10	Surveillance resistant	Are who-paid-whom links hidden from observers?	Yes
11	Politically resistant	Are its rules out of a government's reach?	Yes
12	Censorship resistant	Is it safe from selective blocking?	Yes
13	Confiscation resistant	Are holdings safe from seizure without the keys?	Yes
14	Scarce	Is the supply capped?	Yes
15	Store of value	Does it hold value across time?	Yes
16	Inflation resistant	Is expanding supply at will hard?	Yes
17	Fair launch	Did everyone start on equal terms?	Yes
18	Publicly verifiable	Can anyone check the ledger?	Yes
19	Trust minimized	Can you verify without trusting a third party?	Yes
20	Permissionless	Can you use it without approval?	Yes
21	Immutable	Is rewriting history hard?	Yes
22	Neutral	Is it indifferent to who uses it?	Yes
23	Survivable	Can it outlast its founders?	Yes
24	Eco-friendly	Is securing it cheap in energy?	Yes
25	Open governance	Can anyone see and influence decisions?	Yes
26	Regulation friendly	Can it comply without changing its nature?	Yes
27	Durable	Does it resist decay?	Yes
28	Scalable and fast	Does it settle quickly under load?	Yes
29	Portable and mobile	Can you carry it anywhere?	Yes
30	Available to the unbanked	Does it work without a bank?	Yes
31	Widespread acceptance	Will a stranger take it?	**Not yet**

#	ATTRIBUTE	WHAT IT ASKS	EPIC
32	Quantum future proofed	Does it survive quantum computing?	Not today

Thirty of those answers are a plain yes, and they read the way they should: nobody can map your payments, change the rules underneath you, pick your coins out of the set, or seize them without your keys. Two are qualified — the design is not quantum-proof *today*, and it is not widely accepted *yet* — and only the second of those is a failure no design can fix.

Five rows people argue about

12, censorship resistant. Safe from *selective* blocking, which is the row's meaning. A miner can still leave a transaction out of a block, and a state can cut your connection — but no one can pick your coins out of the set and refuse those in particular, because the chain gives them nothing to pick by.

24, eco-friendly. The claim is comparative, not absolute. Most of Epic's blocks are mined on commodity CPUs and GPUs rather than purpose-built ASICs — though a minority, around four percent, use Cuckatoo, which ASICs can mine. The comparison that matters is against chains secured entirely by industrial hardware, not against a hypothetical zero. A reader who thinks any proof-of-work is disqualifying will mark this row no, and that is a coherent position.

26, regulation friendly. This does not mean Epic is compliant with any particular regime, and it does not mean regulators approve of it. It means the protocol does not require an identity field to function, so an exchange or custodian can meet its own obligations at the edges without the base layer being rebuilt.

31, widespread acceptance. The only unambiguous failure, and the one this book is about. Everything else on the list is a property the design supplies. This one is supplied by other people deciding to use it, which is why no amount of engineering settles it.

32, quantum future proofed. Not today. A large enough quantum computer would break the commitments and the signatures alike, as chapter seven says. What makes the row survivable is that the scheme can be replaced without changing the money — same chain, same supply, new cryptography.

APPENDIX B

Buying Epic Cash

This is the practical appendix: how to actually acquire EPIC without a bank, a broker, or a form that asks for your employer. It condenses two longer guides — one for the EPIC/XMR pair, one for EPIC/USDT — which are available separately.

Nothing here is advice. It is a description of what the screens ask for, and where people lose money using them.

The venue

NonKYC is a small exchange that runs on the premise that it does not want to know who you are. That premise is worth exactly as much as the operator's honesty, and no more: an exchange that does not collect identity documents still holds your coins while they are on the exchange. **Custodial until you withdraw** is the rule; "no KYC" is not the same as "no risk.

The two pairs that matter:

EPIC/XMR — Epic priced in Monero. For someone arriving from Monero, this is the cleanest path: no stablecoin, no bank rail, no identity.

EPIC/USDT — Epic priced in Tether. Easier if you already hold dollars on-chain, and it introduces the issuer risk that chapter two describes: a balance that can be frozen by an administrative decision.

Both books are thin. Sizes that would be a rounding error on a major exchange move the price here, which is a fact about the

asset and worth knowing before you place an order.

The account

An email address, a password, and two-factor authentication. Turn on the authenticator before you deposit anything; an account without it is an account someone else can empty. Withdrawals additionally require an eight-digit PIN sent to your email, which is a second factor and also a second thing to lose access to.

The deposit, and the field that is not optional

NonKYC offers two deposit methods for EPIC, and the choice matters.

Epicbox gives you a deposit address **and an onchain note number**. The note is how the exchange attributes an incoming payment to your account, because on a ledger with no addresses it has no other way to know who sent what. The payment travels through the exchange's relay, and if the relay is unreachable or your wallet cannot reach a node, the send fails.

HTTP gives you a custom URL with your account number already inside it. Paste that as the destination and there is no note to forget and no relay in the path. **Use this one.** It removes the failure mode the next paragraph is about.

If you use epicbox anyway, the note is not optional. Your wallet may label that field *(optional)* — that is the protocol's description of it, not the exchange's. Leave it out and the coins arrive, cannot be attributed, and become a support ticket — the privacy design working exactly as designed.

Write the note down when you create the account. If you lose it, you can retrieve it from the deposit page, but not from the blockchain.

The trading screen

The market is an order book, not a swap box. Two columns of resting orders — bids on one side, asks on the other — and a form that either takes the best available price (market) or waits at a price you name (limit).

Three things trip people up:

Decimals. Prices are quoted to eight places, quantities to three. `0.00128652` is a price, not a typo, and a limit order at the wrong decimal is a gift to whoever takes it.

Market orders on a thin book. A market buy for a size the book cannot absorb walks up through every ask, and you pay the worst price on the way through. On EPIC/XMR, check the depth before you send anything larger than a few hundred dollars.

The spread. The gap between the best bid and the best ask is your cost of entering and leaving. On a quiet day it is around half a percent, which is cheap for a privacy coin and expensive compared with a major pair.

The fees

Trading is 0.2 percent at the base tier, falling toward 0.06 percent at the top, with a further 25 percent discount for holding the exchange's own token. A maker order — one that rests in the book — is cheaper than a taker order that removes liquidity.

Withdrawal of EPIC costs 1.4 EPIC, which is a flat fee and therefore proportionally cruel on small amounts.

The exit

Withdraw to a wallet you control. Paste the address from **your** wallet, not from a chat message, and confirm the first characters and the last on both screens. There is no chargeback and no support line that can reverse a payment on a chain designed to make reversal impossible.

Four ways beginners lose money

The onchain note, omitted. Described above. It is the most common, and it is silent — the transaction succeeds, the balance does not appear.

The wrong network. USDT exists on a dozen chains. A deposit sent on the wrong one is usually gone.

The seized exchange. A custodial balance is a promise. Promises are not coins.

The panic sale. Thin books fall faster than thick ones. If you cannot name the price at which you would be wrong, you will find it during a bad hour.

A worked example

Buying 500 EPIC at 0.76551 USDT costs 382.76, plus 0.77 in fees: 383.53. Selling the same 500 straight back at the bid, 0.76170, returns 380.85, less 0.76 in fees: 380.09. The round trip loses 3.44, about 0.9 percent — roughly what the spread and two fees cost, and the reason a thin market punishes people who trade rather than hold.

APPENDIX C

Wallets and self-custody

Two shapes of software, doing two different jobs.

BT Wallet is the one to install. It is self-custodial, open source, and made by BT LABS, whose founder is a long-time Epic Cash developer, and it ships for Windows, Linux, macOS (Apple silicon), Android and iPhone, updating itself from inside the app (copies from Google Play or the App Store update through the store). Every file on its release page at epiccash.btlabs.uk/releases has a SHA-256 checksum you can compare with BT LABS' signed list before you run it.

What it does that the older wallets did not: invoices — you ask a paired contact to pay an amount, and they approve and pay it or decline it, while you can withdraw it until it is paid; contacts and pairing, so you pay a person you know rather than a pasted string; payment links, which fill in the payment for you; a version with Tor* and a version without it, because the law on Tor differs from place to place, with a Tor mode that sends every connection through Tor; a Kill switch mode that stops the wallet connecting at all; and support for running your own node, which the wallet sets up and manages for you. BT Wallet is in active development, and the work aims at making payments between wallets more private and more reliable, using new cryptographic tools.

* No Tor on iPhone for now.

A wallet without a node of its own keeps your keys on the device and connects to nodes spread across the network to fetch chain updates and check balances. The trade-off is the same as

every light wallet: you are trusting a node to tell you the truth about balances. A node never holds your keys, but a node can be hostile, so use nodes the community trusts; status.btlabs.uk lists public nodes and shows whether they are online and in step with the chain. Run your own node and the wallet checks the chain itself, at the price of a first sync and some disk space.

New wallet software deserves a small first payment, whichever wallet it is, and the recovery phrase written down either way — a wallet is replaceable, the words are not.

No wallet here is a bank app. There is no password reset, no support line that can move your coins or see your history, and no way to reverse a transaction. That is the point, and it is also the cost.

Ten rules, in the order they matter

01. The recovery phrase is the money. Twenty-four words in BT Wallet (some other wallets use twelve), written on paper, stored somewhere that does not burn and is not photographed. Anyone who has them has the coins. Anyone who does not, does not — including you, if you lose them.

02. Never type the phrase into anything but the wallet. No support agent needs it. No website needs it. No helpful person in a chat needs it. A recovery phrase requested is a theft in progress.

03. Test the restore before you trust the backup. On a second device, or a fresh install of the wallet, choose Restore wallet, enter the phrase, and confirm the balance appears. If the wallet received coins before 2025, tap Start from a date or height, then

Scan from the start. Keep the original wallet as it is until it does, and close the test copy afterwards. A backup you have never tested is a hope.

04. Two copies, two places. One at home, one somewhere else that is physically separate. Fire and flood are more likely than cryptography failing.

05. Do not store the phrase digitally. No screenshots, no cloud notes, no password manager entry that syncs, no email to yourself. A copy of the phrase on a connected device, outside the wallet, is a copy someone else may find.

06. Verify addresses character by character. Epic has no addresses on its chain, but to get paid a wallet still hands out a long string that tells the sender's wallet where to deliver the payment. Anywhere such a string is pasted, check the first and last characters on both screens. Malware can swap a copied string for another, and a swapped string pays someone else.

07. Send a small amount first. Any new destination, any new wallet, any new exchange: send the minimum, confirm it arrives, then send the rest.

08. Keep the deposit note. Where the exchange uses the relay method, the note is part of the address. Store it with the deposit address, not separately, and in BT Wallet put it in the Message field when you send; a deposit link or QR from the exchange fills it in for you. Without it the exchange may not know the coins are yours. Where the exchange offers a URL deposit, use the URL and leave the message empty.

09. Mind what happens around the chain. Epic's chain shows that payments happened, not who paid whom or how much: there are no addresses on it, amounts are hidden, and nodes drop spent outputs as they go, so there is no public list of your payments for anyone to search. What can leak is around the chain: when you connect, from which IP address, and through which node and server. A well-resourced observer watching one particular wallet over time could learn something from that. Tor and your own node make it harder.

10. Write down what you did. A short note, offline: which wallet, which version, where the backup is, which exchange account exists. The person recovering your coins may be you, years later, with no memory of any of it.

If something goes wrong

A wallet that will not sync is usually a network problem, not a loss: check the internet connection, check the network mode in Settings › Network & privacy (Kill switch and Node only hold some or all connections), and check whether the node you point at is answering. A wallet that will not open is not a loss while the recovery phrase exists. A phrase that is lost is a loss, and it is final.

The recovery path is always the same: install the wallet, choose Restore wallet, enter the phrase, then Start from a date or height and Scan from the start if the coins arrived before 2025, and wait for the chain to catch up. The coins come back from the phrase; payment history, notes and contacts come back only from an exported wallet data file. It works or it does not, and testing it in advance is the only way to know which.

APPENDIX D

Glossary

Aggregated signature. A single small proof that stands in for thousands of signatures, which is how Mimblewimble keeps a chain small.

Block reward. New coins paid to whoever finds a block. Epic halves this on a schedule; Bitcoin halves roughly every four years.

CoinJoin. A technique that combines several payments into one transaction so an observer cannot tell which input funded which output.

Confiscation resistance. The property that holdings cannot be taken without the keys, regardless of who asks.

Consensus. The rules every node enforces, and the reason a ledger with no administrator still has one agreed history.

Custodial. Held by someone else — an exchange, a broker, a bank. You have a claim, not the coins.

Cut-through. Removing intermediate outputs — within a block, and across blocks as the chain is compacted — keeping only what is still unspent.

Dandelion++. A relay protocol that sends a transaction through a random "stem" before broadcasting it, so the first node to see it is not obviously its author.

Emission curve. The schedule by which new supply is created over time.

Era. Epic's term for a period between halvings of the block reward.

Fungibility. The property that one unit is indistinguishable from another. A coin with a history is worth less than a clean one; a coin that *cannot have* a history is worth the same as every other.

Genesis block. The first block in a chain.

Halving. The scheduled cut in block reward. Epic's are annual in its early eras, then stretch out.

Kernel. In Mimblewimble, the part of a transaction that proves it balances and commits to the fee, without revealing amounts.

Levy. Epic's declining issuance allocation, which funds development and expires before the supply curve flattens.

Light wallet. A wallet that does not store the chain and asks a node about it instead, trading verification for convenience.

Mimblewimble. The protocol family Epic uses: no addresses, no amounts on-chain, and a compact ledger built from cut-through and aggregated signatures.

Mempool. The queue of transactions waiting to be included in a block.

Mixnet. A network that hides who is talking to whom by routing traffic through several relays.

Node. A computer that validates the chain and relays transactions.

Onchain note. NonKYC's term for the number that attributes a deposit to your account on a ledger with no addresses.

Output. A discrete amount of coin, spent as a whole and re-created as change.

Pedersen commitment. A cryptographic commitment that hides an amount while still allowing the network to verify the books balance.

Permissionless. Usable without anyone's approval.

Range proof. A zero-knowledge proof that a hidden amount is a valid, non-negative number. Epic uses these; they are a genuine zero-knowledge construction, and the protocol has no general-purpose zero-knowledge circuit beyond them.

Reorg. A reorganization of recent blocks, where the chain replaces its tip. Normally shallow; consequential when deep.

Seed phrase. The words that regenerate a wallet. Whoever holds them holds the coins.

Slate. In Mimblewimble, the file exchanged between two parties to build a transaction together — the closest thing the protocol has to an invoice.

Solo mining. Mining without a pool, keeping the full reward and bearing the full variance.

Stem phase. The first, private hop of a Dandelion++ relay.

Taint. The informal marking of coins by their history, which exchanges and analytics firms use to refuse or flag deposits.

Tail emission. A small permanent block reward, used by some chains to pay for security indefinitely. Epic has none; its supply is capped.

UTXO. An unspent transaction output — the individual coin being spent.

Zero-knowledge proof. A proof that a statement is true without revealing why it is true.

APPENDIX E

Sources and method

A book that asks you to check its work should say what the work was.

What the argument rests on

Primary documents. The Epic Cash white paper, first published in 2019 and revised since; the 2026 technical edition; and the protocol's own source code and consensus rules, which are the only authority on what the chain actually does. Where the documents and the code disagree, the code wins, and the book says so where it matters.

The chain itself. Supply figures, emission schedule, block times and the date of the singularity are properties of the ledger, not claims about it. A reader with a node can reproduce every supply number in this book.

The public record. Exchange listings and fee schedules; the Zcash Orchard advisory; the Zano incident report; disclosures from Firo and Beam; the Bogatyy paper on Grin transaction tracing; and the market data used in chapter fourteen, captured on **5 November 2026**. Prices move; the capture date is stated so a reader can tell the difference between an error and a stale figure.

The Quest for Superior Money. The thirty-two attribute framework, published in 2026, which supplies the test in Part Two. The book applies it; it did not invent it.

What could not be verified, and is marked as such

No consolidated third-party audit of the full protocol has been published. Epic's cryptography inherits from the Grin and secp256k1-zkp lineage, and its range proofs are a genuine zero-knowledge construction — but a lineage is not an audit of this codebase. Chapter thirteen says this plainly rather than letting the reader assume otherwise.

There is no published post-mortem of the 2021 episode. The book describes what is publicly known and declines to narrate what is not.

The genesis date is inconsistent across sources. The 2019 white paper gives 1 August 2019; later material gives 2 September 2019. The chain settles this question and the sources do not; the discrepancy is noted rather than resolved by preference.

Privacy has not been measured. No study has attempted to trace Epic transactions at the scale Bogatyy attempted on Grin. The claim that Epic resists the tracing he claimed on Grin is an argument from its construction, not an experimental result, and is presented as such.

How the book was made

The research, drafting, typesetting and figure work were done with substantial AI assistance, directed and edited by the author. This is disclosed because a reader is entitled to know how a text was produced, and because the alternative — leaving it out — would be the kind of omission this book criticizes elsewhere.

Every number in the text was checked against a source at the time of writing. Where a figure is an estimate, the text says so.

Where a claim is contested, the text gives the contest. Where the author does not know, the text says he does not know.

If you find an error, the useful response is not to trust the book less. It is to check the source, and then to say which line is wrong — that is the whole method, and it only works if readers use it.

Every source named here is listed, with links, at quietprivatemoney.com — along with the corrections readers have sent, so that the next printing is better than this one.



QUIET PRIVATE MONEY

The book, the audio, and every source cited in it:

quietprivatemoney.com

Todd Pitner · 2026