Nown: Collaborative Research Paper

Anonymous
nown.to github.com/nownto/nown

v0.1 Released into the public domain (the Unlicense)

1. Trust between strangers

Trust between strangers has always run through a keeper of records. Nown gives a stranger a history that travels with him: one public record where every dealing a key closes leaves a mark anyone can read and sum.

Personal trust still carries parts of the economy, a trade between neighbors or a supplier a firm has used for twenty years, and it carries them well. The record is memory, the enforcement is the loss of future business, and both sit with the two people who bear the risk. The mechanism is sound. Its limit is reach.

The arrangements that extended that reach put a keeper in the middle. A guild vouched for its members, a court held the parties to their bond, a bank stood behind a draft, a platform holds a seller's ratings. Each supplies the missing history, and each owns it. The keeper can be bribed, coerced, or shut down; the standing built under him lives on his servers, under his rules, and ends when he does. His service has a price, and everyone who needs a stranger's history pays it.

What one public record changes is who bears the cost of a broken word, and who can see it. Every action carries consequences, monetary, reputational, or otherwise, whether the actor watches them land or not. A keeper shows a stranger only the consequences it pays him to show. The record shows every one of them, to anyone who cares to look, at no charge and at no one's discretion.

It shows them in a narrow form, and the narrowness is the point. What lands on the record is the result: a mark up where a dealing was ruled correct, a mark down where it was not. The dealing itself stays with the two people who made it. A reader learns what a key's conduct came to without learning a thing about the conduct, which is the most a stranger has ever needed and less than any keeper has ever held.

2. Pretty Good Trust

Perfect trust is on offer nowhere, at any price. What Nown supplies is pretty good trust[1]: enough to act on, computed by each reader for himself, owned by no one.

Phil Zimmermann faced the same limit when he built PGP,[2] and answered it the same way: no central authority granting a key's authenticity, and no promise of perfection. Every user judged for himself, from signatures he had collected, whether a key was good enough to rely on, in what Zimmermann called a "decentralized fault-tolerant web of confidence."[7] It was pretty good, and pretty good was usable where perfect was unavailable.[8]

Human action runs on the same standard. PGP proved the shape, and karma asks more of it: signing that a key belongs to a person is a narrow, nearly binary fact, while conduct across dealings is graded and never finished. What Nown takes from Zimmermann is the standard, decentralized, reader-computed, plainly imperfect.

Judgment is why the protocol runs no test for fraud. Whether a dealing was real, whether a party lied, whether an exchange was worth what either side claims, are matters of judgment, and the protocol has no place to stand to make one. It records how a dealing was ruled and makes the ruling permanent. The judging stays where it has always been, with the person deciding whom to deal with, who now reads a record instead of a rumor.

The openness is symmetric. Competitors, critics and enemies hold keys and build karma exactly as anyone else does, and their records are as readable. A protocol that screened them would need someone to do the screening, and that someone is the keeper again.

A record that showed only successes would be a brochure, and would be read as one. The record carries a key's failures beside its wins, which is what makes it the record of a real person: fallible, particular, and readable enough to act on.

3. Keys

A person acts under a key. The public half is the account others know him by, the private half proves the key is his, and neither says who he is. A key is property, kept or sold, and it begins at zero.

How many keys[3] a person holds is his own business, and the economics do the limiting. Opening a key costs almost nothing, by design: what is expensive in Nown is accumulating karma and holding it, never opening an account. Each key starts from zero and builds its own history, so a crowd of keys is a crowd of separate lives to live. Most people will want a few: a public key, a private one, perhaps a working key for business and a quiet one for dealings they would rather keep apart. Past those, another key buys little. A double life stays possible and, as in the flesh, costly to keep.

The same economics screen the machines. A real history is dear to build, so keys spun up to flood or to fake stand out against a graph of genuine human dealing, and any application can filter them out while the protocol admits everyone.

Losing the private half is losing the key, and the design offers no recourse anywhere. An authority that could restore a key could take one, and that authority is the keeper the whole protocol exists to remove. What a key holder does is what holders of bearer instruments have always done: generate the key on a machine kept offline, write the recovery phrase on something that survives fire and water, store it where he would store gold, and keep it out of every device he types on. A key held that way survives a lost laptop. A key held any other way is one accident from being a stranger again.

A sale moves control and leaves it in place. Handing over the private half gives the buyer everything the key can do, and the seller can do the same things afterward, because a private key is a number and numbers copy. So what the record proves is who was sealed as the owner and when, rather than who holds the key at this moment, and a buyer's protection is the same as everyone else's: a seller who keeps a copy and acts under the key has failed a dealing in the open, in front of the market he still has to sell into. A business that has dealt well for years can be sold with the key that carries its record, and the buyer steps into a going concern that the network will keep reading from conduct.

4. The record

The record is one, public, and permanent. It holds karma marks and seals, and it holds nothing else, so a reader learns what a key has done without learning anything about the doing.

Permanence is the property everything else leans on. A history that could be trimmed would be trimmed by exactly the people whose history matters most, and a reader who knows a record can be edited learns nothing from reading it. The cost is real and paid deliberately: a key's failures stay legible as long as its successes do. That is the price of a record anyone will believe, and it is why a key that has dealt for years carries something a fresh key cannot buy.

The second property is how little the record holds. A dealing is never written to it. Its terms stay off, and so do the outcomes it fixed, its oracle, its deadlines, its fee, its evidence, the ballots that ruled it, and which two keys took part. One mark lands on each key when the dealing closes, and that is the whole of what the network learns.

Privacy here is a property of the design rather than a promise made about it. A reader sees that a key carries forty marks up and one down. What any of them was for, who stood on the other side, and why the one went down are facts the record never had. Nobody has to be trusted to withhold them.

That is a different bargain from the one a public ledger usually offers, and it is worth stating plainly. A ledger that publishes transactions buys auditability with exposure, and every design built on one spends years trying to win the exposure back. Nown publishes the result and keeps the transaction, so the auditability is of the sum alone: anyone can check that a key's karma follows from its marks, and no one can reconstruct the dealings from which the marks came.

What a reader can infer from timing and volume is a question for whoever writes the applications. Marks land in blocks, and a person watching a key closely learns when it dealt and how often. The protocol neither offers that reading nor prevents it, and a key holder who wants less of it holds more than one key.

5. Karma

Karma[4] is a key's score: the sum of its marks. Every reader computes it from the record for himself, and a fresh key computes to zero.

There is only the record of what a key's dealings came to and a figure each reader computes from it under rules everyone shares, so two readers running the same rules reach the same number, and there is nothing anywhere for anyone to freeze or seize.

Karma is minted rather than moved. A mark is written to a key, nothing is taken from anyone to write it, and karma is never staked, escrowed, lent, or transferred. That is what keeps the design small: a stake would have to be held by someone, released by someone, and priced by someone, and each of those is a custodian the protocol would then have to build.

A mark is a mark. Karma is not fractionable, and no counterparty, no sum at stake, and no judgment of what a dealing was worth makes one mark count for more than another. This is the design conceding a limit rather than declining a feature: worth is subjective, so a protocol that scaled a mark by the value of the dealing would be publishing a number nobody can compute. Sizing a mark by the counterparty's karma has the same defect one step removed, since it prices a dealing by who was in it. What the protocol can observe is that a dealing closed and how it was ruled. It writes that, and stops.

The scale runs both ways without limit. A key whose downward marks outnumber its upward ones carries a score below zero, and a key that has dealt well for years carries a large one. No floor catches a failing key and no ceiling holds a proven one back, so a long history is a real quantity rather than a saturated one.

A key at zero confers nothing. It can receive a mark from a counterparty who already carries karma, and it can give none. Two keys at zero dealing only with each other therefore write nothing

to either, however often they deal, and a thousand of them write a thousand nothings.[1] That single rule is what prices the oldest attack on any reputation system: identities are cheap, so a protocol that let cheap identities confer standing on each other would be printing it for free.[2] A newcomer climbs by dealing with someone who already carries karma, which is how a first reputation has always been built.

The rule needs one thing from the cryptography, and it is a small thing. A mark is valid on a proof that a counterparty co-signed the dealing: a different key from the one being marked, and one whose karma is above zero. The proof carries those two bits and nothing more, so the record learns that a valid counterparty existed and never learns who he was. Everything the design wants from the sybil bound is bought with two bits and no names.

Naming the two directions

A scale that moves in increments both ways needs words for the two directions, and most of the available words smuggle a judgment in. *Good* and *bad*, *honorable* and *shameful*, *quality* and its absence all ask the reader to accept a standard nobody set. Even *positive* and *negative* carry a residue of approval in ordinary use.

The way out is to notice that two different objects are being named, and to let each keep its own word.

A **ruling** is a statement about a dealing under terms the two parties fixed themselves, before anything happened. It is *correct* or *incorrect*, and those words mean *as agreed* and *otherwise*, with no opinion about the man attached. Nobody is being called good. A set of terms is being compared to a result.

A **mark** is a statement about a sum. It moves the score *up* or *down*. Up and down are directions on a line, and a line has no morals. The protocol names the direction because the direction is all it can compute.

So the vocabulary falls out of the same limit that shapes everything else: the protocol cannot measure worth, so it declines to name quality, and it names direction instead. A reader who wants to call a key good is free to, and he is doing his own judging with his own standard, which is where judging has always belonged.

Reading the tally.
Correct and incorrect name a ruling, up and down name a mark, and a reader supplies his own words for what a key's tally means. Open: whether applications converge on a shared vocabulary for reading a tally, and what an interface should show a person who has never seen one, given that the sum alone hides the shape of the history behind it. Work on this problem.

6. Dealings

A dealing is any exchange two keys put something behind, and everything about it is agreed before it begins: what it can resolve to, who rules it, when a panel may be called, and what a panel is paid. It is the only kind of entry there is.

There is one shape of dealing. A trade with money in escrow, a course taught for nothing, a certification a panel ratifies, a seat on a panel, a promise by a node to keep the record: all of these are the same object with different things attached to it. Adding species to a protocol is how a protocol acquires corners, and Nown has one shape repeated.

The rulings are polar, and between two keys there are exactly three: a mark up for both, or up for one and down for the other, either way round. A ruling that marked both keys down would say

the dealing failed on both sides, which is a judgment about worth wearing a ruling's clothes, so the protocol has no such outcome.

Everything about a dealing is fixed before it opens, including how it ends if nobody says anything. A deadline that passes with neither an attestation nor a verdict closes the dealing as correct for both, which is where a dealing that simply went well arrives, so a silent deal is a closed deal. That is a different question from who decides, which is the oracle: the parties attesting, or a quorum ruling. The fee that would pay a panel is posted at the same moment, by whoever opened the dealing. Unused, it returns to him. Used, it pays the panel, and where money rides the loser reimburses it out of his deposit, so the judgment is paid for by the party it went against without the panel having to wait for a loser to exist.

The moment of the quorum is a setup term, and it divides dealings into two working shapes. A dealing between opposed interests, a trade, a job, a contract, calls its panel only if the parties come to disagree; most such dealings close on attestation and no panel ever forms. A certification calls its panel at creation, because ratification is the point. Like any dealing it fixes exactly two outcomes and the panel picks one: affirm, and the seal is written; or deny, and no seal lands while the issuer of the claim takes the mark. The panel never reaches for a third answer, because there is none to reach for.

The writing is one touch. A dealing stays open while it runs and lands its marks once, at the close, so the record grows by two entries per dealing however long the dealing took and however much passed between the parties inside it.

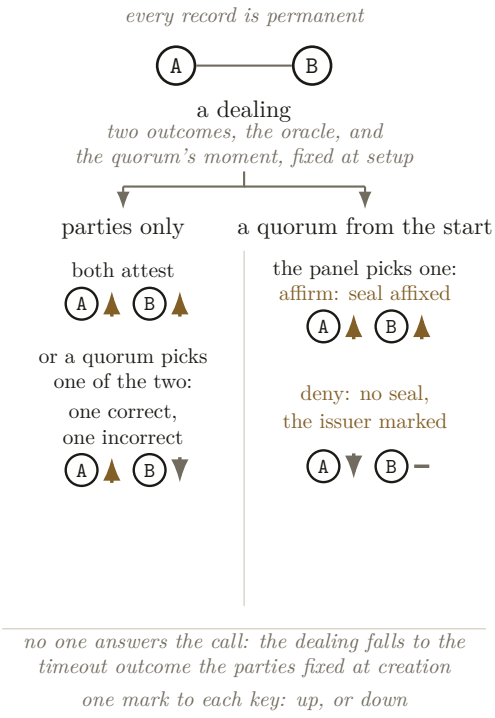

Figure 1: A dealing without money. Fixed at setup: the outcomes, the oracle, and whether a quorum is called at once to ratify or only on a dispute.

The proven side gets nothing

A key at zero confers nothing, so a man with a long record gains no karma by dealing with a beginner, and he still carries the full downside if it goes wrong. He does it for the trade. Karma is the residue of commerce rather than its motive, and a design that paid people to onboard strangers would be paying for exactly the dealings that are cheapest to fake.

What follows is a newcomer discount: proven keys will price a stranger's first dealing accordingly, which is what Friedman and Resnick showed a generation ago is the price of identities that cost nothing to create.[2] Nown leaves that price standing. It makes the discount legible and lets a newcomer work it off in public.

Circles, and why the protocol has nothing to say about them

Two keys that both carry karma can deal with each other as often as they like, marking each other up every time, at no cost to either. A firm dealing inside itself, a family, a guild, a town are all that shape, and every one of them is a legitimate place for trust to be built.

Earlier drafts of this design treated repetition as a leak to be plugged, with a rule that a pair mints once, or that later dealings between the same pair count for less. Both rules ask the protocol to know that two keys have dealt before. The record holds karma marks and seals, so it has no pair to look at, and by design it must not: a protocol that could tell you a firm's key had dealt with its accountant's key forty times would have given up the privacy that makes the record safe to publish.

So the question dissolves at this layer, and it does not disappear. A key's karma is a sum of marks with counterparties nobody can see. Whether a given history was built broadly or narrowly is a question about a pattern, and reading patterns is what the application layer is for. An application can weigh timing, volume, and the graph it can infer, and it can decide that a tally accumulated in a burst reads differently from one accumulated over years. The protocol declines to answer it in the same way, and for the same reason, that it declines to measure what a dealing was worth: it cannot, so it does not pretend to.

One consequence of that is worth naming rather than hiding. A proven key can deal with its own fresh keys and lift them, manufacturing a second reputation out of its first. Keys are property, and a forbidden version of this would simply be sold instead. What it produces is a history whose karma all flows from one source, and an application that reads the graph sees that source: a mono-source history reads the way a young company with a single investor reads.

The genesis.
A key at zero confers nothing, so at the network's first block, when every key is at zero, nothing can move: the problem is how to launch. The candidate is a genesis beacon, declared openly in the record's first entries: a node that grants each new key one small injection and shuts itself down for good once block time and network action pass a stated saturation. Open: the injection's size and schedule, the saturation test, how a per-key injection is rationed when a person may open unlimited keys, and the math that leaves the earliest keys no advantage over those who come after. Work on this problem.

7. Oracles

Every dealing ends the way its oracle[5] says it ended. The oracle is the parties agreeing, or a quorum ruling, and no court sits above either.

Attestation settles the dealing that goes right, and it is two parties agreeing that it did. A trade that goes right needs no judge, and most of them go right.

The deadline does the rest, and the protocol fixes what it does rather than leaving it to the terms. A dealing that reaches its deadline with no attestation and no verdict closes as correct for both. The seller who delivered takes his mark whether or not a satisfied buyer ever gets around to signing, and a buyer who wants a judgment has the whole window in which to call for one. One rule, the same on every dealing, and a reader never has to ask what a particular silence was taken to mean.

The quorum[6] is the oracle for everything the parties cannot settle alone: the dispute one of them raises, and the fact a certification asks it to ratify from the start.

There is no third oracle. Reading a key's record and acting on what it says is a gate. A gate reads and acts, and it resolves the moment it reads, which is a different kind of thing from an oracle that picks between outcomes a dealing named. A panel can also be asked to attest something that fixed no outcomes at all, and then it is a witness: it says what it read, and nothing turns on it. An oracle chooses between outcomes a dealing named. A witness does not choose, and a gate does not judge.

Whatever the oracle rules is the only ruling the record carries. This is narrower than justice, deliberately. The parties may play a deal with charm, cunning, or hard bargaining; the protocol passes no judgment on how a man plays, and a deal one side rigged can still be ruled correct by the counterparty who accepted it or the quorum that read the evidence. No higher court sits above the oracle, so the ruling stands. What the protocol does is make losing cost: a permanent mark down, and the deposits moving along the path the ruling names. The mark is the same size whoever the counterparty was, so a key dealing far above its own experience is exposed to its own inexperience and to nothing else, and an application that lets a thin key reach a heavy one without a deposit has priced its own risk rather than the protocol's.

One corner case leaves the network entirely. A use that needs an oracle someone owns, a court, a registry, an exchange, cannot run on the record, because a corruptible source would dilute the trust the record exists to hold. It runs, if enough people need it, as a fork: a separate network with its own rules, its own record, and its own web of trust, where an old bureaucracy's rulings can be given a portable, permanent home while its weaknesses stay off the record everyone else reads.

8. Quorum

A quorum is a panel of strangers of high karma, called at random for one case, each free to answer the call or decline it. No one can find it, every seat is bonded, every choice is sealed until the deadline, and the outcome most of the opened choices carry is final.

The panel exists to stay invisible. It is called for one case and dissolves when the case ends, and nothing about its members is visible to the parties, to each other, or to anyone relaying the traffic. A steward[7] who could be found could be bought, threatened, or flattered, and a panel that can be found can be reached before it votes.

Every property the quorum claims is bought in how the panel comes to exist, so that is where to start.

Who may sit

Eligibility is karma above a bar, and the bar is a share of the live distribution rather than a fixed number, so it moves with the network and nobody sets it. Each key tests the bar against its own record, in private. No roster is published anywhere, because a published roster is a list of people worth approaching.

The two parties are kept out of their own case, and the call is where that happens. The record carries nothing about a dealing, so the protocol has no way to learn who the parties are; the parties know, their clients issue the call, and a client that omits two keys reveals nothing about the ones it kept. The exclusion costs no anonymity at all, and a panel of strangers becomes a fact about how the call was made rather than a hope about who shows up.

Then each eligible key runs a verifiable random function[3] over the case and learns privately whether its value falls below the threshold for the panel wanted. Qualification stays private until it is used, so no list of the panel exists anywhere, at any moment, for anyone to buy.

Signing is seating

A steward takes his seat by signing, and the signing is the whole of it. He signs every outcome his own dealing can have at the moment he accepts, which is the moment he accepts that missing the verdict forfeits his bond. The stewards sign the oracle side together. The slot stays open only until the quorum is full, and a key that has not signed by then is left out of the case.

Everything is therefore authorized before anyone reads a word of the evidence, and nothing is signed after the verdict lands. That is what makes the settlement hold without a custodian: there is no pot to assemble, no threshold to reach afterward, and no one whose absence can strand anyone else's money.

One key takes one seat, and the signature scheme is what enforces it. A seat is taken under a linkable ring signature,[4] which proves that some eligible key signed and never which one, and adds a tag the signer cannot choose. In the published scheme the tag is derived from the ring, which is not enough here: an eligibility set that has not changed between two cases is the same ring, and two seats taken by one key would link across them. Binding the case identifier into the tag fixes it, so two signatures by one key on one case produce the same tag while two on different cases produce unrelated ones. That tag is what refuses a second seat, and it matters that the scheme enforces it rather than the protocol merely asking: with a plain ring signature a steward could attach any value he liked and sit as many times as he wished.

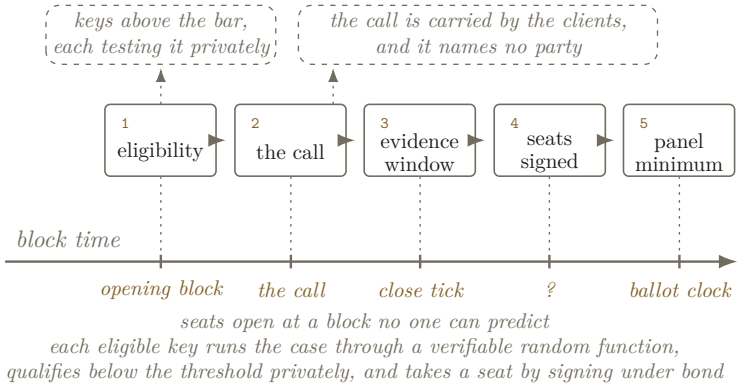

Figure 2: How a panel assembles: eligibility tested in private, the call carried by the parties' clients, seats taken by signature under bond, and the evidence opening only for the keys that took one.

The cooperative

The quorum is a cooperative. Its stewards together are the oracle, and that one side makes a dealing with each steward separately.

> A quorum dealing takes the whole panel as one side and one steward as the other, and the panel is the oracle. Every steward deals with a part of himself, one share in the total size of the quorum. If he is right, his bond and his bounty are wired to him as the dealing closes. If he is wrong, he forfeits the bond and his seat closes with it.

The members together are the firm, and each member deals with the firm he belongs to, holding one share of the other side of his own dealing. The panel is the oracle for every one of those dealings, and each settles on its own: as many dealings as there are seats, resolving against the same verdict, each along the paths its steward fixed when he sat down.

Money moves on a settlement rail rather than on the record, with the stewards on one side and the oracle aggregate on the other. Neither the protocol nor the quorum ever holds it. What the rail is asked for is what every rail worth settling on already has: an output with fixed paths, committed when it is funded, and a public fact that selects between them.

For a steward's bond that fact is his own opening. The paying path unlocks against an opening committed to the outcome the verdict named; the forfeiting path opens after a delay. A steward who matched holds an opening that fits, and takes his money on his own signature alone. A steward who missed holds an opening that fits nothing, the delay runs out, and the other path is taken without him. A hash lock and a timeout, and going quiet buys him nothing except the delay he was going to lose anyway.

What he takes is his bond, a bounty out of the fee, and a share of the bonds the missing seats forfeited. That last part is what makes the amount depend on something nobody knows at seating, and the answer is that the amount depends on one number: how many seats matched. Both the verdict and that count are public once the choices open, and the panel is a fixed set by then, so the whole space of endings is small and knowable in advance. A steward signs one paying path per possible count of matching seats at the moment he accepts, and the count the case produces selects one of them. A nine-seat panel is nine paths. Nothing is invented afterward, and nothing waits on anyone.

Reading and ruling

Each steward receives the evidence as an encrypted package that opens only for a key that was called in the round and signed the quorum dealing. He reads it and, in private, picks between the outcomes the parties signed before they dealt. The panel invents nothing and reaches for no third answer.

The quorum is a guessing game whose focal point is the evidence: each steward is answering what he believes most other careful readers of the same file will conclude. On clear evidence, strangers who cannot coordinate converge on the reading the file best supports, because that is the only answer each can expect the others to reach.

That convergence is the design's central bet, and it is worth stating as a bet rather than a result. It holds when the evidence is the most obvious thing for a stranger to coordinate on. Where a file is genuinely murky, careful readers have nothing to converge on and the vote is closer to a coin, which is why panel size and the weight of a case are tied together, and why a protocol that promised justice rather than a ruling would be overselling itself.

A choice is written as a commitment and opened after the deadline. The commitment hides what is inside it until every choice is in, so the only thing a steward has to go on is the file. That closes the cheapest attack on a panel: given a running count, a steward could skip the reading, wait for the last block, and side with whichever way the tally leaned, matching the majority almost every time while the stewards who did read carried him. A steward who opens inside the window

that follows the deadline has voted; anyone else has stayed silent, and silence settles exactly as a miss does. Taking a seat is therefore a commitment to read the case. Publishing an opening before the deadline voids the seat and forfeits the bond, which keeps the window from becoming a slow count anyone can read: a steward can prove his own choice to himself at any time, and proving it to the network early costs him everything he posted.

The verdict is the outcome most of the opened choices carry, and it is final. A silent seat is a choice for nothing. Seating closes before the choices are cast, so the panel is a fixed set and an even count can only come from silence; where it happens, one seat is dropped by the protocol's own randomness and that steward's bond returns to him, neither gained nor lost. The draw is derived from the case and verifiable by everyone, so every node computes the same one. A rule that named a position, first or last or highest, would be a rule a steward could try to land on or avoid, and a rule that dropped whichever seat arrived first is not computable at all, since anonymous traffic reaches different nodes in different orders and two nodes would compute opposite verdicts on a close split.

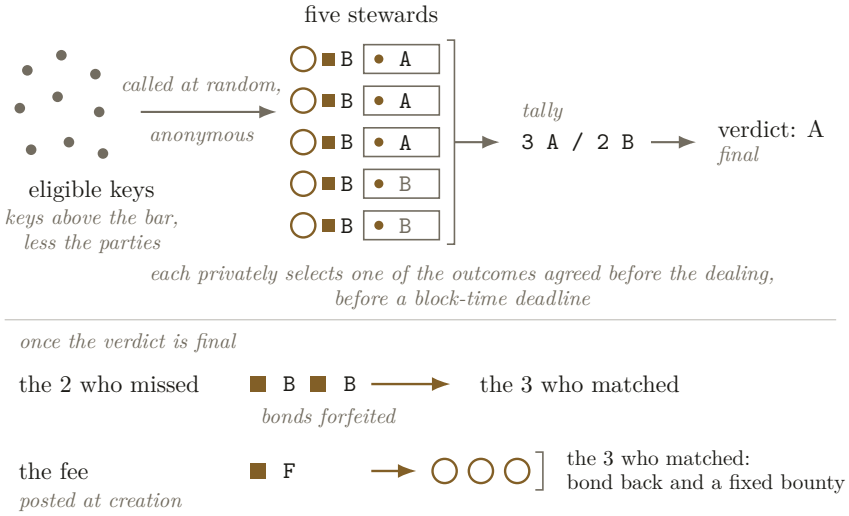

Figure 3: Stewards vote in secret under bond. Most of the opened choices carry the verdict. Each steward then closes his own dealing with the panel: the seats that matched take their bonds, the bounty, and the bonds the seats that missed forfeited.

Settling the money in the case

Where a dealing carries money, the reference construction is a two-of-two escrow[6] between the two traders that commits, at funding, to every way it can end: the mutual close, one path per outcome, and the timeout. A funded output cannot gain new conditions, so opening a dispute rewrites nothing.

What completes the ruled path is a signature, and the parties supply it. An earlier draft had the quorum make a secret reconstructable, one per outcome, committed at funding as an adaptor. It does not work, and the reason is worth recording: an adaptor point fixed at funding needs someone who knows the value behind it, and at funding the panel is an empty set. So the rail is asked to verify nothing about the panel. The verdict reaches the parties, and signing the path it names is the last act of the dealing.

The marks land on the verdict rather than on the money, which is what makes a stalling loser a small problem instead of a large one. His mark down is already written the moment the dealing

closes, whatever he does next, and the escrow's timeout path is what keeps the funds from hanging on his silence. The enforcement is the same enforcement the rest of the protocol runs on, and it needs no primitive that does not exist. Nown owns no escrow, and any escrow service may implement Nown as its resolver.

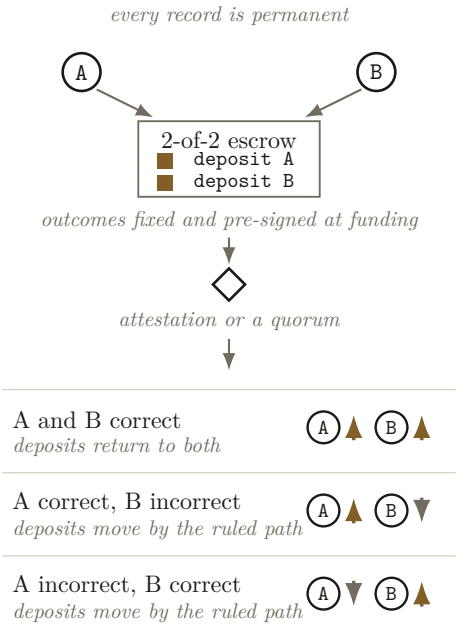

Figure 4: A dealing that carries money: deposits locked in a two-of-two escrow with every outcome pre-signed, ruled by attestation or a quorum, settled along the ruled path.

What a steward is paid, and what he is not

A seat is taken without an attributable act at any step, and it ends without one. It settles in money alone: for a match, the bond back plus a bounty out of the fee plus a share of what the missing seats forfeited; for a miss or a silence, the bond gone to the seats that matched.

The forfeits go to the stewards who read the case, and that is the point of them. A panel is a crowd of volunteers with no reputation to gain and nothing written to their keys, so the money is the whole of the wage, and the men who got it right are the ones who earned it. It also prices the seat correctly: a steward's return rises exactly as the share of careless seats rises, so the case that most needs a careful reader pays a careful reader most. What it never does is reward a minority, because matching the verdict is the precondition for being paid at all. Nothing is written to the steward's key, and that is a choice rather than an omission. Karma moves only when a mark is written to a key, and every reader computes karma from the public record, so a mark that credited a steward would be a mark that named a key as a member of a particular panel. The anonymity rule and a karma-paying seat cannot both hold, and anonymity is the one the design cannot spend.

That leaves a real question: what a steward gets for a reputation he cannot show. The answer is money, which is what he was owed, and an option he may take or leave. A key that wants its service known affixes a seal declaring that it does this work. The seal names no case and reveals no verdict, and an application may keep a directory of such keys if a use for one appears. What matters is that it stays a choice, because everything written to a key comes from consent and consequence, and a steward consented to a bond rather than to a public account of where he sat.

What capture costs

Capture is priced rather than detected. An attacker cannot find the panel, so all he can do is hold keys among those that answer a call, and the arithmetic prices that. A panel is a sample; an attacker holding a share p of the keys that answer carries a majority of n seats with probability no greater than exp(-2n(1/2 - p)^2).

That bound is small for large n and not small at all for the panel sizes a reader might imagine. At five seats against an attacker holding thirty percent of those who answer, the bound is loose, about two times in three, and the exact chance is about one time in six. Either way a five-seat panel is no serious obstacle. So the panel size is the security parameter, chosen against a tolerated capture probability q, and n must be at least ln(1/q) over 2(1/2 - p)^2. Nothing here says a minority cannot win a case. It says what making that likely costs, and that the cost rises without limit as the panel grows.

The share p is a share of the keys that answer, rather than of the keys that qualify, and that distinction is the one a design like this most easily gets wrong. Seats are voluntary. A key that always answers is over-represented among those who do, and an attacker always answers: if he holds a fraction a of the eligible keys and the rest answer at rate d, his share of seats is a over a plus (1 - a)d, which exceeds a for every d below one. At a fifth of the eligible keys and an answer rate of one in ten he holds most of the seats. The bond and the bounty have to be set so that reading a case is worth a competent stranger's evening, because the answer rate is what the whole capture argument rests on.

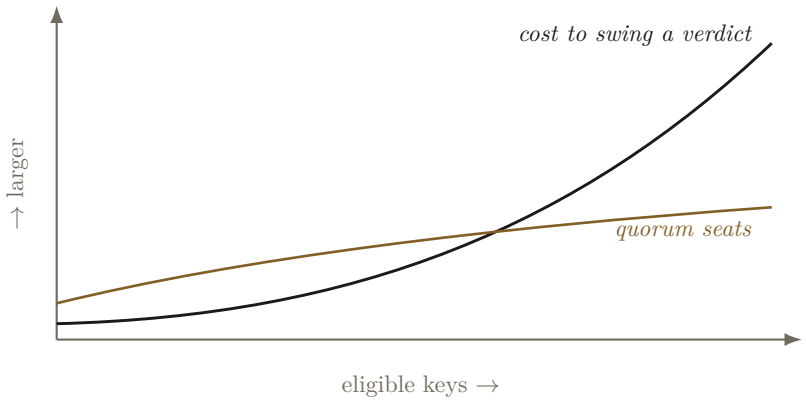

Figure 5: The panel grows with the network, and the cost of holding a majority of the keys that answer grows faster.

One consequence is accepted openly. Because the fee is posted by the party who opens a dealing and must clear a competent reader's evening, a small dispute may call and find no one willing. The deadline then passes and the dealing closes as correct for both, so nothing is left hanging and no one is marked. That is a cost the design accepts and states: a party who cheated on a stake too small to be worth judging keeps a clean record on that dealing, which is close to how justice has always priced itself. Whether the parties of a lapsed call should be able to mark each other, each taking a mark down of his own to write a grievance the other cannot erase, is under discussion: it would give small dealings teeth, and it would hand every sore loser a knife.

Enough voluntary voters.
Everything the quorum promises rests on the panel filling, and seats are voluntary. Pricing the bond and the bounty against the cost of a careful evening is the obvious lever and it is not obviously sufficient, since the same fee has to clear for a case nobody finds interesting as for one everybody does. Open: what guarantees a supply of willing readers, while paying only those who read, keeping the panel a crowd of strangers rather than a standing body, and leaving no case to be starved by indifference. Work on this problem.

The panel size function.
The capture floor is settled arithmetic. What sits above it is not: how the minimum grows with the weight of what is arbitrated, what the ceiling is, and how a specialist call scales up as its eligibility narrows the crowd a panel hides in. Work on this problem.

The bar and the answer rate.
The eligibility bar is a percentile of the live distribution, and the answer rate is the real security parameter, since seats are voluntary. Open: the percentile, how it behaves in a young network where the distribution is thin, and the bond and bounty levels that keep enough competent readers answering. Work on this problem.

Storing the evidence.
The evidence reaches the panel as an encrypted package that opens only for a key called in the round that took a seat. The package has to live somewhere, and no company exists to host it. Open: where a case file rests between the call and the verdict, who serves it and at whose cost, how long it survives, and how a steward fetches one without the fetch itself identifying him. Work on this problem.

Murky evidence.
The convergence argument holds where the evidence is the obvious thing for strangers to coordinate on, and weakens as a file gets murkier. Open: how to price panel size against the readability of a case without asking anyone to judge readability, and what the design should concede about cases that are genuinely close. Work on this problem.

Funding a bond unlinkably.
A steward is never identifiable, in any context, yet his bond must be funded from somewhere, and a funding source links a key to a case. Open: breaking that link without a custodian and without weakening the bond's precommitted paths. Work on this problem.

9. Seals

A seal is what someone else certified about a key: a credential, a membership, an ownership. It is a dealing like any other, so it lands with its subject's countersignature and both keys agreed every consequence beforehand.

A man who wants to prove he knows a field studies under a working expert instead of paying a university for a certificate. The teaching is a dealing: the master is paid, and at the end he fixes a seal to the student's key under his own, at his own risk, since a seal on a student who cannot do the work costs him exactly what a failed dealing costs. Years later an employer reads on the key that a prominent name in the field answered for this man. A credential becomes a chain of people who answered for it, readable end to end.

A seal is a dealing, and that is what puts a mechanism under the issuer's risk. It fixes its outcomes and its oracle at issuance, so a challenge years later calls a panel to pick between things

that were named when the seal was made. That is what gives a later panel something to rule on, and it is what makes a denied seal cost the issuer.

When a panel denies a claim, the mark falls on the issuer alone. He made the claim and answered for it; the subject countersigned to accept a seal rather than to underwrite it, and the outcomes attach to the issuer's assertion. The cost of that choice is stated rather than hidden: a man who buys a false credential from a complicit issuer walks away unmarked, and only the man who sold it pays. The alternative marks a subject for a claim someone else made about him, and the design would rather let a buyer escape than write a mark on a key for something it did not do.

Where no single signer can carry the fact, a quorum ratifies it. Two people marry and want it on a record that outlives any registry; a translator swears his translation is faithful. The certification is opened as a dealing that calls its panel at creation, the panel reads the evidence and affirms by majority, and the seal is written like any other outcome. The panel is paid from the fee posted when the dealing was opened, which is why a certification that carries no money can still seat one.

A seal mints

A seal mints karma from nothing. It writes marks the way any dealing does, it moves nothing from one key to another, and nothing caps how many a key may issue or hold.

Nothing needs to cap them. A school that certifies thousands is read through the graduates it certified, and every one of those graduates is out in the market dealing under a key that carries the school's seal. Issue carelessly and the seal comes to mean what careless issue means. The supply of seals is bounded by the reputation of the people issuing them, which is the same thing that bounds the supply of anything else worth having.

What the protocol fixes is small: the field is short, both parties consented, and the entry is permanent. A seal is free text under those constraints. What a seal should say, and which forms become standard, is settled by the applications that read them, and the first standard forms will be written by whoever ships the first useful directory.

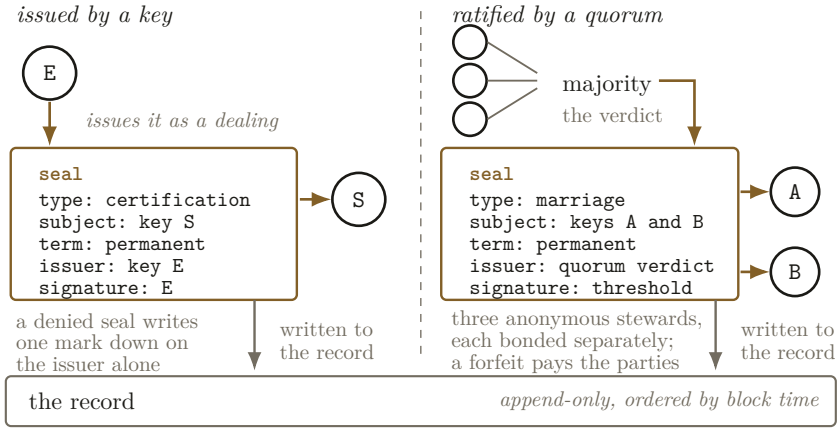

Figure 6: A seal is one signed entry a key issues, or a quorum ratifies, and the subject countersigns.

A seal meant to lapse carries its ending inside it. At sealing, both parties also sign a timed dealing that, at a stated block, appends a voiding mark aimed at the seal, and any reader may publish that mark when the block arrives. A client shows a seal past its term as run whether or not anyone has bothered to publish it. Nothing is deleted; the seal and its expiry both stand on the record forever.

Taxonomy is emergent, because an imposed one would need a maintainer and a maintainer is a keeper. An expert sealing welders writes "welder"; other issuers adopt the label because filters select on it, and it becomes a standard by being used. A label carries exactly the weight of the keys that have used it, so the meaning of a word at any moment is the karma of the issuers behind it. A karma filter over a label is a professional directory that no one runs.

Ownership is a seal, and a company is a key. Someone with authority to bind the firm signs a key into being; from then on that key is the firm's reputation, and its dealings build history exactly as a person's do. Who owns the firm rides on the key as a quorum-ratified seal; to sell the business is to hand over the key and re-seal its ownership to the buyer, so the chain of owners reads end to end. A share is the same idea divided: a cap table is a set of fractional ownership seals, and a sale of shares is a re-sealing. What the last seal proves is who was sealed, rather than who holds the key at this moment, and a dispute over who owns what goes where every dispute goes.

What a seal carries.
The protocol fixes that the field is short, consented, and permanent, and leaves the contents to the market. Open: whether a minimum shape is worth fixing after all, what the cap on length should be, how a document is committed to without touching the record, and how far a free-text field can carry a credential system before conventions have to be written down. Work on this problem.

Sealing a company.
A firm's key is its reputation and its ownership a ratified seal, so transfer is re-sealing and a cap table is a set of fractional seals. Open: the firm-to-key binding, fractional sealing and re-sealing on a sale, how an ownership dispute is called and ruled, and what a buyer can rely on given that a sale leaves the seller in control too. Work on this problem.

Reading the seals.
A seal weighs what its issuer weighs, and no cap exists on granting or holding. Open: how an application weighs many seals from few issuers, how label conventions converge and split, and whether accumulation needs any price beyond the weight a reader assigns. Work on this problem.

10. Block time

Nown counts time in blocks.[8] The measure is kept by verifiable, unforgeable work rather than by any clock, government, or authority, and every reader reads it the same.

A record ordered in time needs a clock nobody owns, because a clock any party could pause, hurry, or serve differently to different readers would put that party's hand directly on every key's history. What the protocol asks of it is thin: a public, ordered, permanent tick, no more. Precision belongs to clients.

Every clock with those properties today belongs to a network Nown does not govern. Borrowing one works from day one, and it makes the protocol a dependent: a tick produced by people who owe the record nothing, under incentives that are not its own. The problem is to replace that dependence with a native ability. Nown needs a way to keep time from its own activity: a clock that is decentralized, checkable by every reader, and no other network's to lend or to stop.

The candidate under work is pulse time, Nown's own clock. The two are separate things. Block time is the borrowed measure, kept by globally verified proof of work on a network Nown does not govern. Pulse time is what the protocol would keep for itself. The record already produces an ordered sequence, the fingerprints its nodes commit, and the tick is the next one. A verifiable delay function[5] between one fingerprint and the next puts a floor of real, sequential time under every tick, so a year cannot be manufactured in an afternoon. Beyond that floor the clock is meant to be

driven by the network's own activity, and that is exactly where the design is unfinished. Weighting a tick by the karma of the keys dealing in it is the obvious move and is not sufficient on its own: one person holding two keys of high karma can deal them against each other and attest each correct, forever, at no cost to either. Any weighting has to price that, and none of the candidates has been shown to.

This is why the borrowed clock is worth keeping a while. From the first day the native clock is specified and runs in shadow: its ticks are computed and recorded beside a borrowed tick that stays authoritative meanwhile, and the switch happens only when the pulse is expensive enough to forge. Activation depends on the pulse's breadth, the count of distinct keys dealing in a tick, each counted only up to a cap so that breadth means many keys rather than one large one. When that breadth holds above a threshold for a set span, every node reads the same switchover from the record itself; ages carry over one for one, and only the production rule changes. The switch is one-way, the borrowed clock stays as a passive checkpoint, and like every rule in open software it is the community's to revise.

Keeping the borrowed clock as a checkpoint is also what keeps the tamper argument from folding in on itself. A record whose fingerprints are its own clock is checked against a sequence it produced, which proves nothing by itself; what makes the sequence expensive to forge is the delay function's sequential work and the archived checkpoints from the era when the clock came from outside. A forger must redo the work, in order, and still match the anchors.

Pulse time.
Karma is measured against block time, which belongs to a network Nown does not govern; the problem is a native way for the protocol to keep time from its own activity. Pulse time is the candidate. Open: the exact pulse function and how it prices a pair of high-karma keys dealing with each other for free, the saturation threshold and span, the delay function's calibration and its hardware-advantage bound, and the interaction with the nodes that commit the fingerprints. Work on this problem.

11. Randomness

The protocol needs randomness no participant can predict or steer: to call a panel, and to drop a seat where an even count leaves no majority. It takes that randomness from its own activity.

A clock and a source of randomness are different objects, and conflating them is how designs get into trouble. Block time says when. Randomness says which. A clock is valuable precisely because it is predictable, since every reader must agree on the order of what happened; a beacon is valuable precisely because it is not, since anyone who can predict it can aim it.

What the protocol asks of a beacon is the pair of properties that make a draw a draw. Nobody may know the value before it is fixed, and everybody must agree on it afterward. A steward who could compute the next draw would know which cases to be eligible for. A party who could steer it would pick the panel that judges him, or, in a close case, choose which seat is dropped.

Nown derives the value from its own action, so the entropy the network produces by dealing is the entropy it draws on. That has a property no outside source has: an attacker who wants to move the beacon has to move the record, and moving the record means dealing, which costs him what dealing costs everyone. The randomness is paid for by the thing it protects.

A young network cannot pay that price. Little activity means little entropy, and a beacon a determined party can afford to grind is a beacon he can aim. So the same shape the clock uses applies here: begin on an established outside source, run the native beacon in shadow beside it, and switch when the network's own activity carries enough entropy to make forgery impractical.

How much is enough is the open part, and it is open for the clock in the same way. The two switchovers are separate decisions on separate thresholds, since a network can be busy enough to time itself while still being cheap to grind, or the reverse. What a threshold has to be measured against is the value of what the beacon protects, which is the weight of the cases a captured draw could decide, and that is a moving quantity in a growing network.

The native beacon.
The protocol draws randomness from its own activity and starts on a borrowed source until its own carries enough entropy. Open: the extraction function and what it commits to, the cost of grinding it as a function of network activity, the threshold and span that trigger the switch, whether the borrowed source stays as a checkpoint the way the clock's does, and how the beacon behaves in the thin period right after launch. Work on this problem.

12. Nodes

The record outlives any single machine because many machines keep it. Each node holds a full copy, serves it, and can be checked against every other.

At intervals the network commits a fingerprint of the whole record to a tick, and any copy can be checked against it, so a machine that quietly altered the past fails a check anyone can run. That is tamper evidence, and it is solved arithmetic. It is a different thing from agreement, and the difference was easy to miss.

Two nodes can each commit a well-formed fingerprint for the same interval, one carrying an entry the other left out. Both copies pass a check against their own fingerprint, so a rule is needed for which of them is the record, or a node joining the network has two histories to choose between and the word record loses its meaning. The rule is the simplest one that needs no authority: the record for an interval is the first committed fingerprint whose contents are published and check out against it, ties broken by the lower value, and a joining node follows that chain from the beginning, passing over any fingerprint it cannot expand. The availability clause is load-bearing rather than decorative. It is what keeps an anchor accountable: a fingerprint counts only once its contents are published and check out, so paying to anchor buys an interval only for a record someone can actually read.

What the rule does not do is stop a node from committing a fingerprint that leaves an entry out. That is where a receipt earns its place, and the idea is borrowed with credit from Dover's Chronicle.[9] A node that accepts an entry returns a signed receipt acknowledging it and taking responsibility for including it in a later committed batch. It cannot name the batch, because the batch does not exist yet. A receipt that is never honored in any committed batch is proof against the node, and the node takes the mark for it. Suppressing an entry then requires every node to refuse it, and each refusal after a receipt is provable.

Persistence is the part nobody has solved. A record no one is paid to keep has no one bound to keep it. Money solved this for itself by minting coin for the machines that guard it; Nown mints no coin and moves no money, so what pays a machine to hold the whole record and serve it for years is open. It is an incentive problem rather than a storage problem: disks are cheap, and reasons are not.

The candidate under discussion pays in the one thing the protocol writes. Node service becomes a dealing: a node pledges to hold the whole record, proves on challenge that it still has every part, checkable against the committed fingerprint, and earns a mark for faithful service exactly as a key earns one by dealing well, taking a mark down if it goes dark. Beside the proving nodes, every client holding a small shard of the record would add a breadth no single loss can dent. None of

this is settled, and the bet under it, that karma alone is wage enough for a storage network, is unproven. Chronicle answers the same question differently, and the answer is worth studying: its nodes are paid public services that charge fees for accepting, relaying, anchoring and publishing. A node may charge its users, since the constraint sits on the protocol rather than on the people running machines for it.

Keeping the record.
The record must stay whole and served with no company behind it and no coin to pay its keepers. The candidate makes node service a dealing paid in karma, with proof-of-holding as its oracle and receipts making every acceptance checkable. Open: the incentive itself, the proof and its frequency, the shard a client carries, who pays to commit a fingerprint and how often, the censorship bound the receipt mechanism actually delivers, and whether karma alone can pay for a storage network at all. Work on this problem.

13. The application layer

Everything a person touches is an application: the browser that renders a record, the filter that ranks keys, the metric that turns a history into one number. Applications read and interpret. The record and the karma stay one beneath them all.

The call is the first of those applications, and it has to be, because the protocol cannot reach a key. Reaching a key means holding an address for it, and an address is the one thing a steward may never publish. So the protocol fixes who may answer, each key tests that for itself in private, and the client carries the invitation. That division is what lets eligibility be a public rule and a seat be a private fact at the same time.

A reference client does this and everything else the protocol allows.

- Holds keys: generates keypairs, any number, and keeps the private halves where only their holder can reach them.
- Verifies what it reads: syncs the committed fingerprints, follows the first-anchored chain, checks the record against it, and carries a shard if the storage design settles on one.
- Reads any key: computes karma and the metrics over it, Karma DOT first, and shows the marks, the seals, and the voiding pairs, for any public key it is given.
- Opens and joins dealings: writes the terms, the outcomes, the oracle, the quorum's moment, the deadlines, the deposits and the fee, and steps into one under its holder's key.
- Attests: signs a closing, and lets the window run out where the other side has gone quiet.
- Disputes: carries the call, posts the evidence inside the window, and signs the ruled path when the verdict lands.
- Sits as a steward: watches for calls, tests eligibility privately, takes a seat anonymously, commits and opens a choice, and sweeps the bond and the bounty, all without exposing its holder in any context.
- Seals: issues a seal with the countersignature flow, accepts or declines one, and displays what others carry.

Every conforming application computes the same karma from the same record. What an application decides is what to show: which marks it reads, what bar it sets, how it ranks what it finds. One weights recent dealing and forgets fast; another rewards long steady histories; a hiring application reads only the marks that bear on work. None can lie about what a key did, because the record beneath them is one, and every filter is client-side.

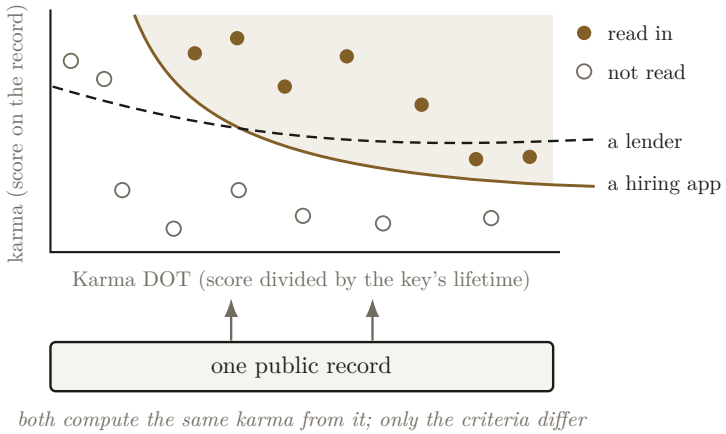

Figure 7: Two applications cut the same keys by different criteria. The record beneath them, and the karma, are one.

Karma DOT[9] is the worked example of an application metric, and its arithmetic is one division. A fifty-year-old of poor judgment and a sharp twenty-year-old have both lived every one of their years; what parts them is what they did with them. DOT reads a key the same way, weighing what it packed into its time rather than how long the time has run.

The division needs one guard, and it is the ordinary one for an average taken over a short life. Score divided by age has no value at all on the block a key first appears, since its age is zero, and it is at its largest immediately afterwards: a key one block old that closes a single dealing would read above every veteran in the network. So the divisor a reader uses is the key's age plus a small settling constant, chosen so that a key reaches a stable reading after a meaningful stretch of dealing rather than after one lucky afternoon. With a constant of c blocks, a key of age a and score k reads k over (a + c): early on the constant dominates and the reading is damped toward zero, and as a grows past c the constant stops mattering and the metric becomes the plain density. It is the same correction a rating system applies when it refuses to rank a restaurant with one review above one with a thousand.

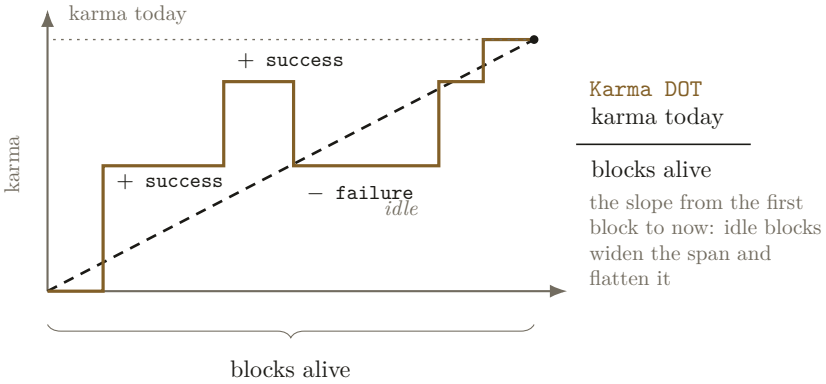

Figure 8: Karma DOT, an application-layer metric: a key's score today over the blocks it has been alive, with a settling constant added to the divisor.

Three consequences follow from the division. Climbing is quick: a burst of good dealing lifts a young key's density fast, while the denominator is still small. Holding is slow: the denominator

never stops growing, so only fresh dealing keeps the density from sagging, the way a business keeps its place only by continuing to serve. Idle time bleeds: every idle block adds nothing to the score and one more to the divisor, so a key that has dealt and then waits thins for every block it waits. Generating a key early buys nothing either, for a different reason: age runs from a key's first appearance on the record, so a key begins aging when it begins dealing.

Read together, the total and the density answer different questions. A top total on an average density is the proven veteran: enormous lifetime volume spread thin across its years. A high density on a modest total is the sprinter: dense recent dealing, unproven at scale. A lender sizing an exposure reads the total, because a ceiling on risk is a ceiling on demonstrated volume; a hirer judging present competence reads the density; and the pair together tells the retired institution from the hot newcomer, which is why an application keeps both in view and lets the reader weigh them.

A gate is the application layer's own resolver, and it is an oracle only in a figurative sense. A context fixes the karma a use demands, and any key whose record clears the bar is read in: a role filled, access granted, a match made. The gate reads and acts, and it resolves the moment it reads; the record it read is left exactly as it found it.

One shape of problem recurs wherever strangers meet: a stranger must be trusted, misjudging him is costly, and the only party who can vouch for him is a silo that owns his reputation and can be gamed or bought. An employer reads a resume the applicant wrote and calls references the applicant chose; one federal enforcement action uncovered more than seven thousand purchased nursing diplomas, bought to sit an exam on credentials their holders had bought rather than earned.[10] A buyer reads star ratings the platform owns and the seller can buy, and fake reviews are reckoned to sway on the order of a hundred and fifty billion dollars of spending a year.[11] A lender prices a stranger's promise against a bureau that is central, gameable, and gives him nothing usable on tens of millions of people.[13] Half of internet traffic is machines,[12] and confidence and romance scams took over six hundred million dollars in a single year.[14] Each is the same shape, and a karma browser answers them the same way: it issues no badge and grants no credential; it lets whoever must decide read an unerasable history, filtered to what he cares about.

An employer describes the karma a role demands: this much, in dealings of this kind, held this steadily. The browser finds the keys that clear it, and a candidate proves he is one by signing a fresh message with his private key. The signature beside the public record is the whole of the proof, and he carries it himself. The signature and the bar together are the gate the hiring application resolves on: no panel is called, and nothing is ruled.

A record of past conduct says nothing about a first dealing, so a genuine newcomer starts indistinguishable from a fresh fake. Cost and permanence raise the price of faking a history while leaving it possible. The record is blind to harm nobody reports, and to the man who was reliable until the moment he was not. And a permanent public record of one's dealings is a thing to handle with care, which is why a person holds more than one key and shows a dealing only the history it needs. Karma is built for dealings that repeat between strangers, where conduct adds up and the next counterparty reads it. A one-time deal, a private fact, a claim to be checked rather than a person to be trusted: for those, an application reads something else.

Each of these is the same protocol read through different criteria, and no one's permission is needed to build any of them.

A dispute, settled.
Two strangers fund an escrow that commits, at funding, to every way it can end: the mutual close, the outcomes a quorum may choose between, and the timeout. If the deal goes right they close together and no one else ever looks. If it goes wrong, the panel picks one of the outcomes they already fixed, and signing that path is the last act of the dealing. The money is never in anyone's hands but theirs.

A worker, hired.
An employer names the karma a role needs; the keys that clear the bar prove it with a single signature. Hiring becomes reading a record rather than trusting a document, and the candidate carries the proof himself.

A degree, replaced.
A working expert seals a skill to his student's key under his own name and at his own karma, and an employer years later reads who answered for the man.

A fact, notarized.
A marriage, a translation sworn faithful, anything one signer cannot carry alone. A quorum reads the evidence and ratifies, and the record holds the fact and the karma behind it.

A loan, priced.
A lender prices a stranger with no bureau standing in the middle, reading the borrower's own record of dealings kept. No thin file leaves him blind, and no new account buries a history the lender would want to see.

A market, without ratings.
A marketplace reads the record instead of keeping stars. A seller's history follows him from every platform he ever sold on, and opening a new account resets nothing. The platform competes on its filter rather than on the reputation it holds hostage.

A firm, sold with its key.
A company's key carries its history, and its ownership is a quorum-ratified seal on that key. Selling the business is handing over the key and re-sealing to the buyer, so the chain of owners reads end to end.

A specialist panel.
A dealing writes seal eligibility into its pre-agreed terms, so a dispute over a weld calls stewards sealed as welders. The filter narrows the crowd a panel hides in, so the panel grows to match, and the base call stays unfiltered beneath it.

A human, told from a machine.
Years of real dealing are dear to fake and cheap to read. An application that wants accountable humans filters for keys whose history costs more to build than pretending pays. The protocol prices pretending and bans no one.

Dover's Chronicle[9] is a protocol for costly public orientation on the web: a participant publishes a signed judgment toward or away from any subject, priced by provable sacrifice, and nodes batch the events, anchor them, and publish the record for anyone to read. Chronicle records what people paid to say; Nown records how dealings between people were ruled. The two are layers, impression and consequence, and they meet without either bending. The cheapest integration is a naming convention: a Nown key can be a Chronicle subject, so a key carries two independent axes, what its dealings came to and what many paid to say about it, and an application reads both without touching either protocol.

Value flows the other way as well. Chronicle's reputation algorithms weigh opinion by the standing of whoever paid to voice it, and opinion is all they have to weigh; Nown's ruled outcomes are ground truth those algorithms can anchor to. A subject Chronicle shows as contested is a fact in want of a ruling, and a called panel is built for exactly that. The boundary holds by design: Chronicle permits unilateral judgment about anyone, priced in sacrifice; Nown admits no mark without a dealing or a countersignature.

Closing

Trust between strangers is built from three parts: a history, a way to read it, and a price for breaking a word. Every keeper that has supplied them owned the history it kept, and owning the history made it the judge of everyone in it. Nown supplies the same three and no keeper: a record held by everyone, a score any reader computes, and a price paid in a permanent mark, earned in the open and worked off the same way. What passed between two people stays theirs. What it came to is public, and it follows the key that earned it into any market it enters, under any name or none.

Read the protocol in the whitepaper. · Work on the open problems.

Notes

[1]Pretty good trust, by descent from pretty good privacy: usable, decentralized, computed by each reader, sufficient for people to act on, and never perfect.

[2]Pretty Good Privacy, Phil Zimmermann's 1991 email-encryption software. Its web of trust let each user judge a key's authenticity from signatures collected from other people, with no central authority, on a good-enough rather than perfect standard.

[3]A keypair: two matched numbers. The private one is a secret only its holder knows, used to sign; the public one is shared and lets anyone check those signatures. Holding the private key is owning everything the key carries, its history and its karma.

[4]The score a key accumulates on the record: the sum of its marks, one written for each dealing it closes, up where the dealing was ruled correct and down where it was not. The word is borrowed from the idea that a person's actions accumulate into what he is.

[5]An agreed source that settles a question a system cannot settle for itself, the way an oracle in antiquity was consulted for an answer taken as final. Here: the parties' own attestation, or a quorum.

[6]Originally the least number of members a body needs present for its decision to count; by extension, the panel itself. Here, the panel of strangers called to rule one case by majority.

[7]One who keeps a duty in trust for others and answers for it without owning what he keeps. Here, a member of a quorum, called to rule a single case and paid for ruling it with care.

[8]Block time is a decentralized measure of time defined by globally verified proof of work rather than any clock, government, or trusted authority.

[9]Delta over time. A key's karma divided by the time it has been alive, plus a small settling constant, giving the density of dealing it has sustained rather than the total it has amassed. An application-layer metric, computed from the record like any other reading.

References

1. J. R. Douceur. The Sybil Attack. IPTPS, 2002.
2. E. J. Friedman, P. Resnick. The Social Cost of Cheap Pseudonyms. Journal of Economics and Management Strategy, 2001.
3. S. Micali, M. O. Rabin, S. P. Vadhan. Verifiable Random Functions. FOCS, 1999.
4. J. K. Liu, V. K. Wei, D. S. Wong. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups. ACISP, 2004.
5. D. Boneh, J. Bonneau, B. Bunz, B. Fisch. Verifiable Delay Functions. CRYPTO, 2018.
6. T. Dryja. Discreet Log Contracts. MIT Digital Currency Initiative (manuscript), 2017.
7. P. R. Zimmermann. PGP User's Guide, Volume I: Essential Topics. Version 2.6.2. Phil's Pretty Good Software, 1994.
8. P. Zimmermann, J. Callas. The Evolution of PGP's Web of Trust. In Beautiful Security, O'Reilly Media, 2009.

9. L. Dover. Chronicle: Bringing Trust and Accountability to the Web. https://chronicle-network.org (accessed 2026). Related work.
10. U.S. Department of Justice. Fraudulent Nursing Diploma Scheme Leads to Federal Charges Against 25 Defendants (more than 7,600 fake diplomas and transcripts). USAO Southern District of Florida, 2023.
11. R. Cavazos, University of Baltimore Merrick School of Business, and CHEQ. The Economic Cost of Bad Actors on the Internet: Fake Online Reviews 2021. 2021.
12. Imperva. 2024 Bad Bot Report (bots 49.6% of internet traffic in 2023). 2024.
13. Consumer Financial Protection Bureau. Technical correction and update to the CFPB's credit invisibles estimate (December 2020: 7.0M consumers with no credit record, 25.3M with an unscored record). 2025.
14. Federal Bureau of Investigation, IC3. 2024 Internet Crime Report ($16.6B total reported losses; $672,009,052 Confidence/Romance). 2025.