Privacy Guardians

Maximally Private Payments on Ethereum, within the Law.

by Leo Glisic

written in collaboration with Fable

July 2026

Grateful for feedback from

Puja Ohlhaver

David Casey

Graham Novak

David Sneider

Abstract

This paper presents Privacy Guardians, a private payment network built on Ethereum. Payment systems available today require their users to accept one of three trade-offs. Traditional payment rails expose each transaction to a chain of intermediaries (the merchant's processor, the acquiring bank, the card network, the issuing bank, and the data and fraud vendors behind them), and every account within them can be suspended by its operator. Public blockchains remove the operator's ability to suspend a self-custodied account, but they publish every balance and every transaction permanently. Anonymity-focused systems provide strong privacy but offer no lawful-access mechanism, and as a result they have been sanctioned or restricted in many jurisdictions, preventing strong network effects.

The design presented here is intended to provide privacy, censorship resistance, and regulatory compliance simultaneously. Users transact through Privacy Guardians of their choosing, which may be regulated institutions, merchant networks, or, where law permits, decentralized operators. A user's own Guardian has complete visibility into that user's account; no other party, including other Guardians, can observe balances, amounts, or counterparties. Guardians operate the network's private accounting but never take custody of funds, which are held in a neutral settlement contract on Ethereum. Each Guardian's books are proven consistent with the funds behind them, cryptographically, at every settlement, so misstated books cannot settle. Any user can withdraw to Ethereum, or move to a different Guardian, without their current Guardian's cooperation. Reserves are held in yield-bearing form, and the resulting yield funds the network: in the example figures used throughout, half compensates Guardians and half funds privacy insurance, including a standing bounty (the honeypot) that gives anyone who obtains a Guardian's dataset a standing incentive to reveal the theft; once the theft is proven, compensation is automatic.

Compliance is structured jurisdictionally. Each government determines which Guardians may serve its residents, lawful process is answered by a user's Guardian in the same manner as by a bank, and withdrawals settle to the public ledger, so funds can always leave the network and every withdrawal is publicly visible. The design is intended to make available, within each jurisdiction, the maximum financial privacy that its law permits, which in most jurisdictions is considerably more than existing payment rails provide. The protocol defines no investment or governance token (user balances constitute a fully reserve-backed private stablecoin), and the design is open-sourced for anyone to implement; the author is not building it himself. This document is a white paper: it specifies the architecture and the incentive structure, and defers engineering detail to a subsequent document. The body of the paper presents a single dollar asset; §9 extends the design to many currencies, and to several backings within one currency.

1. Introduction

1.1 The migration of payments on-chain

Stablecoin settlement volume has grown into the hundreds of billions of dollars, and the largest institutions in the payments industry are now building on-chain payment rails of their own. We take the direction as settled: a meaningful share of the world's everyday payments will move onto public blockchains. The remaining question is which properties the resulting networks will have. Payment systems concentrate around a small number of winners through network effects, and the properties of

those winners tend to persist; we would therefore expect the properties of on-chain payment networks to be determined early and to be difficult to change afterward.

This paper concerns two of those properties, privacy and censorship resistance. Both are largely absent from the on-chain payment systems being built today, and the design presented here is intended to show that both can be provided without any reduction in regulatory compliance.

1.2 Questions every payment system must answer

Two questions determine much of the character of any payment system: which parties can observe a payment, and which parties can suspend an account.

Traditional finance provides weak answers to both. An ordinary card payment is observed by the merchant's processor, the acquiring bank, the card network, the issuing bank, and the analytics and fraud vendors behind them. Each records the payer, the payee, the amount, and the time; each retains the record for years; and each represents a potential point of breach. Privacy in this system is a policy commitment made in a terms-of-service document rather than a property of the architecture. With respect to the second question, any account can be suspended by the institutions that operate it. In most cases that power is exercised lawfully; the relevant observation is that the power exists, is concentrated, and is not visible to the user until it is exercised.

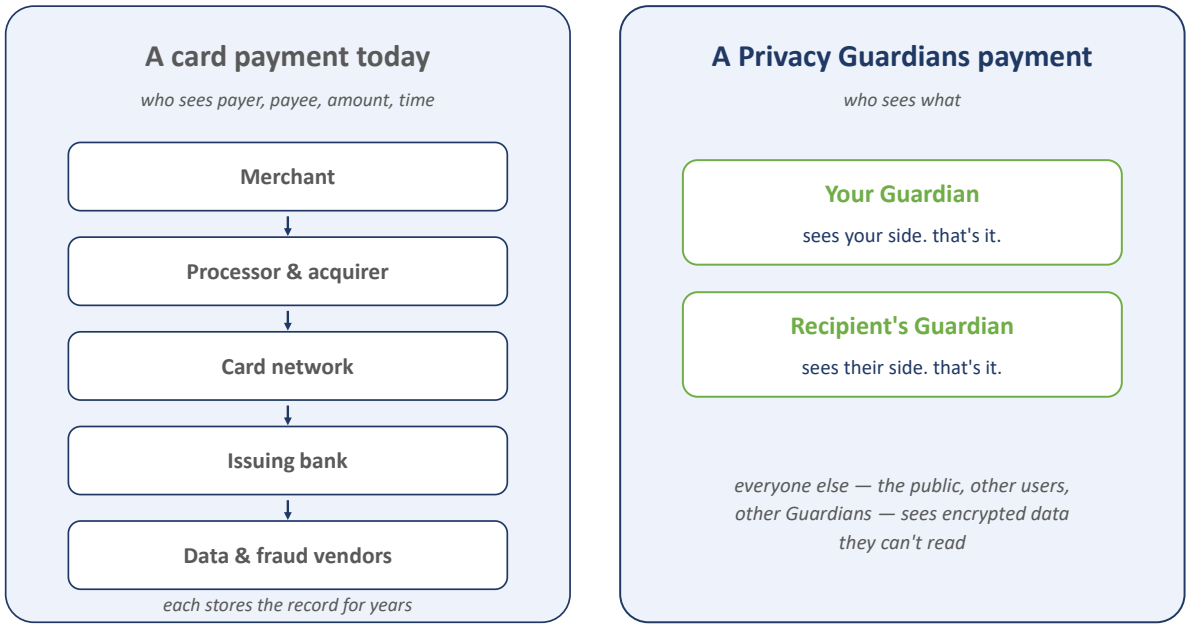

Figure 1: The parties able to observe an ordinary payment today, and the parties able to observe a payment on the Privacy Guardians network.

Public blockchains invert both answers. No party can freeze a self-custodied account, which resolves the second question. However, the first question receives a weaker answer than before: every balance and every transfer is published, permanently, to all observers. A salary paid on Ethereum today is public information, and a merchant accepting payment on a public chain reveals its complete revenue history, in real time, to its competitors.

A third family of systems, the anonymity technologies, conceals activity from all observers. These systems resolve both questions in the user's favor, but they provide no lawful-access mechanism, and governments have responded by sanctioning and restricting them. Whatever position one takes on those decisions, the practical consequence is that anonymity systems have not carried, and in our view cannot carry, the everyday payments of people subject to functioning legal systems.

In other words, the systems available today offer three combinations: broad institutional visibility with consumer protections, full public exposure with self-custody, or strong privacy without legal standing.

1.3 The design goal

The goal of the design presented here is to answer both questions well at the same time, within existing law. Concretely, the network must satisfy four properties at once: it must be decentralized; it must be censorship-resistant, a property which, on Ethereum, follows from being non-custodial; it must provide more privacy than traditional finance provides today; and it must be built for full compliance in every country where it operates. Sections 3 through 8 specify a design that we believe satisfies all four.

The motivation for this work is twofold. First, we would like to see Ethereum widely adopted, and we consider everyday payments the largest unmet use case standing between Ethereum and ordinary users. Second, we consider the erosion of financial privacy to be an architectural problem, and one that a payments layer designed for privacy can address directly. A payments network that users prefer on its own merits (immediate, inexpensive at the point of use, and private) would serve both purposes.

2. Privacy Within the Law

2.1 Privacy as an option

The design treats privacy as an option to be exercised rather than a requirement to be imposed. Some users will exchange privacy for lower fees, for stronger consumer protections, or for familiarity, and we would expect many to do so. The design accommodates that choice. A protocol cannot make users value privacy more than they do; it can only ensure that the private option exists, is competitive, and is fairly priced. In other words, the design standard is that each user receives the amount of privacy they choose, and that the most private option available is a competitive product.

2.2 The maximum varies by country

The amount of financial privacy available to a person is ultimately bounded by their country's law, which determines which records must exist, which parties may see them, and under what process. The network does not exist to change those laws, and it does not need to. Its purpose is to deliver the full amount of privacy that the citizens of each country have already secured under their own legal system.

It is worth noting how large that amount is. In most countries, the lawful maximum is considerably more private than what the incumbent rails deliver. The gap exists not because the law forbids private payments, but because payment data is commercially valuable to the intermediaries that operate the incumbent systems, and no alternative has been built. Governments retain every power they currently hold, including warrants, subpoenas, and lawful process generally (§7). What changes is the set of other parties able to observe a user's payments: under this design, that set is the smallest it has to be.

2.3 Privacy Guardians

The mechanism is a market. Every user transacts through a Privacy Guardian, the party that processes their payments, maintains their account, and is accountable for it. Any entity can operate a Guardian: a regulated bank, a licensed financial technology firm, a merchant network, a telecommunications company, or, where law permits, a fully decentralized protocol, in which case users need not trust any institution. There is deliberately no default; each user selects, from among the Guardians authorized in their jurisdiction, the one whose protections and pricing they prefer.

Each government, in turn, determines which Guardians may serve its residents, which is the same authority it exercises over any regulated business. A government that objects to a particular Guardian can prohibit that Guardian from operating. We would expect no government to need to prohibit the protocol itself, because the protocol takes no positions a government could object to: every policy that governments regulate (identity standards, records, disclosures) is implemented by Guardians, within national law (§7).

2.4 Alignment by incentives

The arrangements described above could be attempted with a terms-of-service agreement and a trusted operator, and they would then depend on that operator's continued good behavior. The premise of this design is that such promises are durable only when they are enforced by mechanisms stronger than reputation. There are three such mechanisms, and they organize the remainder of the paper.

- **Non-custody is enforced by contract.** Guardians never hold user funds and cannot freeze or trap them. Any user can exit, or switch Guardians, without any party's cooperation (§3).
- **Honesty is enforced by proof:** a Guardian's books must be shown consistent with the funds behind them, cryptographically, at every settlement. Books that do not match the funds cannot settle (§3).
- **Privacy is enforced by price:** a Guardian that leaks its users' data faces detection rewarded by a standing bounty, automatic compensation to affected users, and the loss of its own stake, with verification and payment executed by smart contract (§4, §5).

The remainder of the paper describes the machinery. Section 3 specifies the payment protocol: how funds enter without revealing a Guardian relationship, how payments remain private, how settlement is proven, and how users exit without permission. Section 4 describes the economics that fund the network and its insurance. Section 5 specifies the honeypot, the mechanism that creates a standing incentive for any data breach to be revealed, and compensates it automatically once proven. Section 6 describes the user experience, including consumer protections. Section 7 sets out the compliance model, and Section 8 the governance posture and directions for further work.

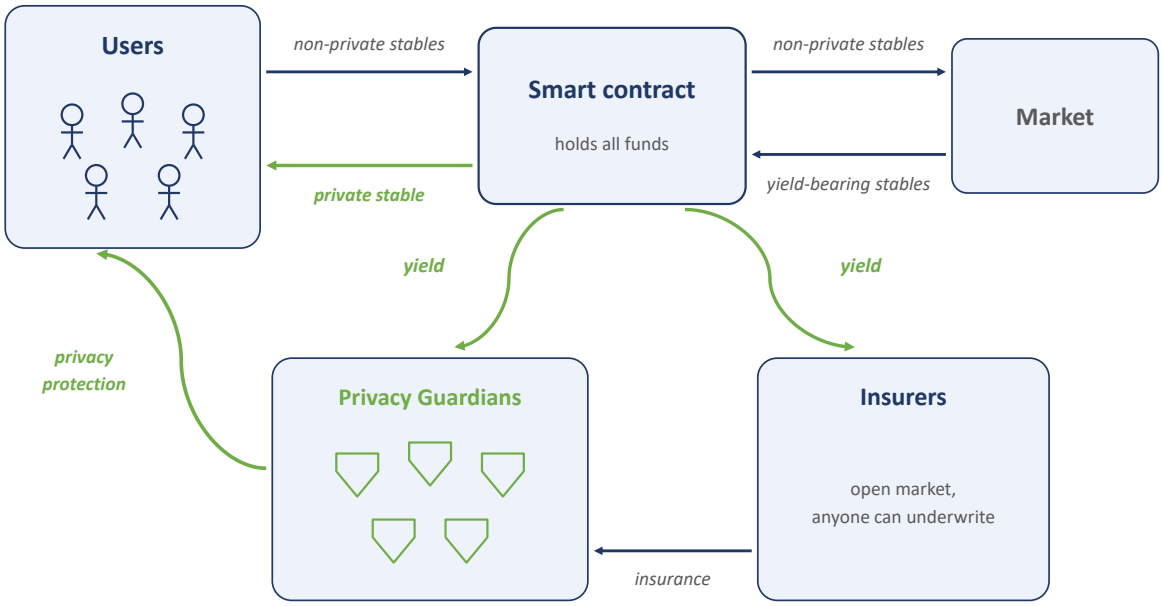

Figure 2: The flow of value. User funds are pooled in a single contract and held in yield-bearing instruments; the yield funds the Guardians and the privacy insurance (§4).

3. The Encrypted Protocol

This section specifies the network's core machinery: how a user joins and selects a Guardian without that relationship being publicly visible; how payments execute so that no observer of the chain can reconstruct who paid whom, or how much; how value settles between Guardians; and how any user can exit, without the permission of any party, at any time.

Two terms recur throughout. The Depository is the protocol's single settlement contract on Ethereum mainnet; it holds every deposited dollar. A Guardian is a service operator, in most jurisdictions a regulated one (§7), that a user chooses as their counterparty for privacy, service, and compliance. Guardians operate the network's private accounting, and they do not hold user funds at any point.

3.1 What the protocol must achieve

The design must satisfy five properties simultaneously. Each property is straightforward to state in isolation; the substance of this section lies in satisfying all five at once.

P1. Private association. When a user joins the network and selects a Guardian, only the user and that Guardian learn of the relationship. Nothing on the chain associates the two.

P2. Private activity. Balances, payment amounts, and counterparties are invisible to the public, to other users, and to every Guardian other than the user's own. The aggregate amounts that necessarily appear on the chain must not reveal individual payments.

P3. Non-custody. Guardians never hold or control user funds. All funds reside in the Depository, a neutral contract that no Guardian, and no consortium of Guardians, can use to trap value.

P4. Permissionless departure. A user can always withdraw to Ethereum, or move to a different Guardian, without their current Guardian's cooperation or consent.

P5. Provable totals. The total value under each Guardian's guardianship is publicly provable at all times, while revealing nothing about any individual account.

A sixth property is one of service rather than protocol: payments must appear immediate to the user, even though the canonical ledger settles periodically. The mechanism that reconciles an instant payment experience with periodic settlement recurs throughout this section.

One boundary of the privacy model should be noted at the outset. A user's own Guardian sees that user's account completely: identity (to whatever standard its jurisdiction requires), balance, and every payment it processes. This visibility is a deliberate design decision rather than an information leak. The network's privacy claim is privacy from the world, paired with accountability to a single party the user chose, and that party can answer lawful process the way a bank can, within its own jurisdiction's rules (§7). This accountability distinguishes the network from anonymity systems and allows for mass adoption.

3.2 Components

Four components make up the network.

- **The Depository.** A single settlement contract on Ethereum mainnet. It holds the network's reserves; maintains one public running total per Guardian (that Guardian's "bucket"); holds newly arrived deposits in a pending pool until they are claimed; stores one commitment (the state root) that summarizes every private account balance in the network; verifies the zero-knowledge proofs that advance that commitment; and processes exits. The Depository is designed to run without discretionary control; its few parameters and its upgrade posture are treated in §8.
- **The message board.** A high-throughput, low-cost data plane on which all protocol messages are posted: encrypted payment intents, cross-Guardian messages, receipts and attestations, and the per-epoch record of changed account commitments. The design requires four properties of this venue (canonical ordering, guaranteed data availability, forced inclusion, so that in the last resort anyone can place a message on the board directly, and low per-byte cost) and deliberately does not require a particular venue. A general-purpose rollup, an application-specific rollup, or Ethereum blob space could all satisfy these requirements.
- **Guardians.** Service operators, each running private books for its users, an encryption endpoint for messages addressed to it, and a prover that periodically demonstrates, in zero knowledge, that its books are exactly consistent with the canonical message record. Guardians hold working capital and carry insurance (§4); they may be banks, licensed fintechs, merchants, new business entities, or, where permitted, decentralized operators (§7).
- **Users.** Each user runs a client (typically a mobile payment app) that holds a single seed. All other information the client requires can be reconstructed from public data (§3.3).

The network operates on two layers. The experience layer is the Guardians' private books: when a user pays, their Guardian confirms in under a second. The canonical layer is the encrypted ledger committed

to by the state root, which advances once per epoch, an interval expected to range from minutes to about an hour (§3.7). The books serve as a cache, while the proven ledger is the canonical record. This division is the same one card networks have used for decades between instant authorization at the terminal and clearing and settlement completed later. The differences are that clearing here completes in minutes (card clearing takes days), and that correctness is checked by a mathematical proof rather than by an audit.

The Depository accepts deposits in the major dollar stablecoins and holds its reserves as a diversified basket of yield-bearing dollar instruments (§4); balances are denominated in dollars and redeemable one-for-one against the reserve (in effect a private, fully reserve-backed stablecoin, its reserve risks insured in the order §4.1 and Appendix F describe; the network issues no investment or governance token). Examples throughout use USDC as the deposit asset.

The dollar is the first currency, not the only one. §9 extends the design to several currencies, and every statement in the body of this paper carries over asset by asset.

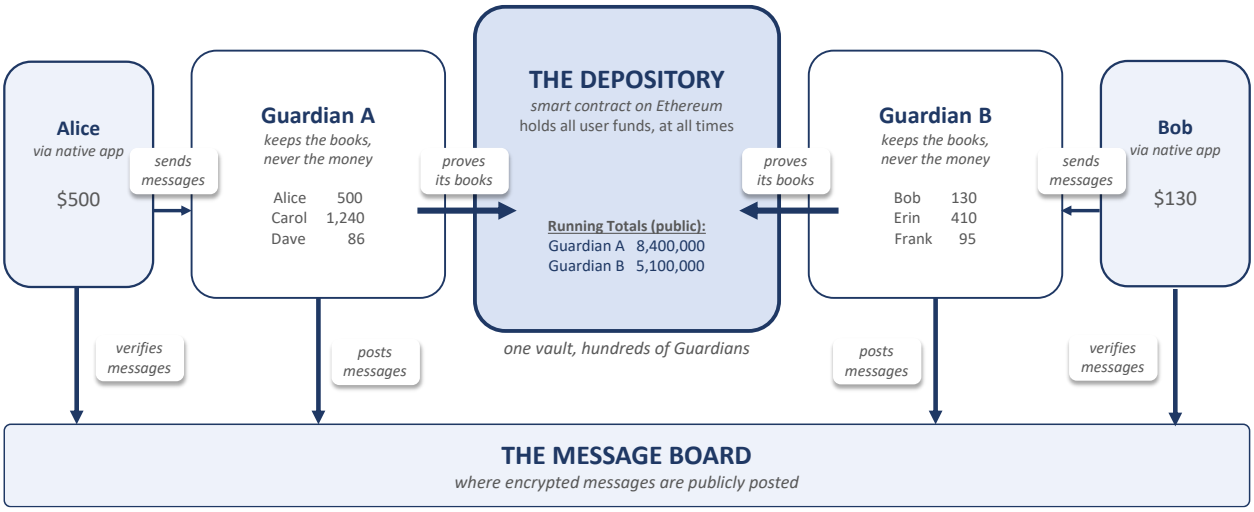

Figure 3: The four components, with a cross-Guardian payment traced through them (the payment and settlement paths are described in §3.6 and §3.7).

3.3 Accounts, keys, and statelessness

From one seed, a user's client derives a signing keypair, which authorizes payments and exits, and an encryption keypair, which receives private messages. The account identifier is a hash of the public keys: a pseudonym. The only on-chain link between a user's Ethereum address and their account is the deposit transaction itself, and the deposit, as §3.5 describes, names no Guardian and no account.

In the canonical ledger, an account is a single entry (a "leaf") containing hiding commitments to its contents (account key, balance, nonce, security-policy hash), one commitment per field, so that possession of any part of an account's data is separately provable (§5). A hiding commitment reveals nothing about its contents, which is why the protocol can publish these entries freely; the policy hash pins the user's chosen exit rules (§3.8) so that even the user's Guardian cannot weaken them unilaterally.

Every message that affects a user's account (intents, credit notices, receipts, balance attestations) is posted to the message board encrypted to the user, and all randomness the client uses is derived from the seed. A client holding only the seed can therefore reconstruct its keys, its full history, its receipts, and

its current ledger entry from public data. As a result, the loss of a device costs only the time required to restore, and there is no file whose loss forfeits funds. To keep repeated payments to the same person unlinkable, recipients share payment handles from which a sender derives a fresh one-time tag for each payment; routing information inside a handle is readable by Guardians, not by other users.

Nothing in the design ties a person to a single account. A user may maintain separate accounts, under separate seeds, with different Guardians. The protocol cannot recognize them as one person, because it has no notion of a person, only of accounts. Each Guardian still identifies the holder of every account it serves, to the standard its jurisdiction requires (§7), so separate identities provide privacy from correlation rather than exemption from identity checks. The arrangement is comparable to holding relationships with several banks: each institution knows its customer, and none sees the customer's activity at the others.

3.4 The partitioned private ledger

The canonical ledger is partitioned into one subtree per Guardian, each containing the ledger entries of that Guardian's users; it is neither a single shared tree nor any one Guardian's database. The state root stored in the Depository commits to all subtrees at once. Each Guardian knows the plaintext contents of its own subtree (its users' balances) and nothing about any other Guardian's.

Each epoch, the commitments that changed are published to the message board. Because they are hiding commitments, this publication reveals nothing beyond an approximate count of active accounts; what it provides is data availability, the raw material any user needs to prove their own balance against the state root without their Guardian's help. This capability is what allows the exits of §3.8 to proceed without any Guardian's cooperation.

The partition also turns the network's solvency condition into an arithmetic identity. Every epoch proof is required to demonstrate:

$$\textit{for every Guardian } G\text{: } \mathrm{bucket}(G) = \Sigma \textit{ balances in } G\textit{'s subtree}$$

The left side is public (the Depository's running total for that Guardian). The right side is private. The proof shows they are equal without revealing any individual balance. As a result, a Guardian cannot credit its users, in aggregate, with more than its bucket holds. In other words, an epoch containing fractional reserve is unprovable and cannot settle, so the condition is prevented at settlement rather than detected after the fact. The identity also closes globally: value enters a bucket only from the pending pool and leaves only through a paid exit, cross-Guardian payments cancel in aggregate, and an epoch that creates or destroys value anywhere admits no valid proof. This is property P5, delivered continuously and by construction: anyone can read every Guardian's total under guardianship off the Depository at any time, with a proof behind it, and no individual account revealed. The totals serve purposes well beyond transparency (insurers price against them; §4). Within the protocol itself, they serve the narrower but critical purpose of ensuring that dishonest books cannot settle.

3.5 Joining: deposits that name no Guardian

A deposit that named its Guardian would publish a relationship the network is built to keep private (P1). Joining therefore proceeds in five steps.

1. Handshake. The user contacts their chosen Guardian (through its own app or through any compatible payment app) and completes whatever onboarding that Guardian's jurisdiction requires (§7). This happens before any funds move, so a Guardian never receives a deposit from a user it would decline.
2. Deposit. The user sends funds, for example 500 USDC (illustrative), to the Depository, attaching a commitment:

$$C = H(\text{account id} \,\|\, \text{Guardian id} \,\|\, \text{salt})$$

(H is a hash function; ‖ denotes concatenation; the salt, derived from the user's seed, prevents guessing.) Publicly, an address deposited 500 USDC into the network. No Guardian is named, no account is visible, and the funds rest in the Depository's pending pool. The deposit is locked (not refundable by the depositor) for a lock window T_lock, a protocol parameter on the order of a day.

3. Service begins immediately. The user has already sent their Guardian the commitment's opening during the handshake. The Guardian verifies that C names it (which means no other Guardian can ever claim this deposit), and the lock means the depositor cannot withdraw it before T_lock expires. The Guardian's claim is therefore exclusive and guaranteed. On that guarantee it opens the account on its books at once, issues a signed receipt, and the user can begin paying within seconds of depositing.
4. Claim, performed later and alongside others. At a moment of its choosing within the lock window, the Guardian claims the deposit inside a routine epoch proof. The proof demonstrates that some set of pending deposits carries valid commitments naming this Guardian and that their sum is correct, without identifying which deposits. The sum folds into the Guardian's single net settlement figure for that epoch (§3.7), so not even the subtotal of claimed deposits appears separately.
5. Refund path. A deposit still unclaimed when the lock expires becomes refundable by the depositor, permissionlessly. As a result, neither entry into the network nor exit from it requires any party's permission.

If the user spends before their deposit has been claimed, the canonical debit cannot come from their ledger entry, because that entry does not exist yet. It comes instead from the Guardian's Advance Payment Pool (Appendix D), an ordinary entry in the Guardian's own subtree funded with its own money. When the Guardian later claims the deposit, the same epoch proof creates the user's entry and repays the Advance Payment Pool internally, a movement inside one subtree that is visible nowhere. A Guardian that claims in the very next epoch needs no advance at all; the advance exists so that claiming can wait for a batch of other deposits to form. The capital this requires scales with what new users spend in their first hours rather than with what they deposit; a Guardian onboarding millions of dollars a day would be expected to need working capital orders of magnitude smaller.

The privacy of the assignment should be characterized precisely. An observer sees deposits enter the pending pool and sees each Guardian's bucket grow at settlements; the anonymity of the deposit-to-Guardian assignment derives from the set of deposits claimed alongside it. The size of that set depends on how busy the network is, and a network with lower activity provides a smaller one. Guardians can strengthen the anonymity set where needed (minimum claim-batch sizes, randomized claim delays, patient claiming for users who prefer it), and the protocol leaves the exact policy open. §3.10 treats what the chain shows, and does not show, in full.

3.6 Paying: intents, receipts, and crossing Guardians

A payment begins as an intent: recipient tag, amount, asset, a per-account nonce, and an optional memo, signed with the payer's account key and encrypted to the payer's own Guardian. The client hands it to the payer's Guardian, which posts it to the message board along with binding commitments to its core fields, one per field; the client watches the board for its own intent, and an intent not posted within seconds expires harmlessly and is simply re-sent (Appendix A). The Guardian that decrypts an intent can neither alter it nor invent one: the settlement circuits verify the user's signature under the commitment, reject any repeated nonce, and reject any account whose balance would go negative.

When payer and payee share a Guardian, the payment does not move value between Guardian subtrees or alter any public Depository balance. The Guardian decrypts, updates both accounts on its books, and issues two signed receipts: the payer's shows the new balance and nonce; the payee's shows the credit. Receipts play a substantial role in the design: every confirmation a Guardian issues is a signed, dated instrument, and it is these instruments that make within-epoch balances enforceable claims and insurable exposures (§4). At the next epoch the Guardian's proof applies the intent to its subtree, and the canonical ledger is brought into agreement with the books.

When the payee is served by a different Guardian, the intent carries a nested encryption: inside the envelope addressed to the payer's Guardian A sits a payload only the payee's Guardian B can read, containing the credit's details. Guardian A debits the payer on its books and posts a cross-Guardian message to the board, addressed to B, bound by the same commitment, and signed by A. Guardian B decrypts, credits the payee on its books instantly, and posts a credit notice encrypted to the payee (a memo from the payer, encrypted end-to-end, can accompany the notice; Guardian B learns the amount and that the payment came through Guardian A, but by default not from whom. Jurisdictions that require originator information for large transfers can mandate an encrypted originator payload readable by the receiving Guardian; the protocol carries the slot and leaves the requirement to §7).

The two sides are tied by the binding commitment: the sending Guardian's epoch proof must consume it as a debit, and the receiving Guardian's as a credit, each exactly once. As a result, value cannot be created, destroyed, or double-counted between subtrees; a movement of value between subtrees requires the payer's signature at the root of the chain of evidence.

Payments to recipients outside the network operate as follows. The payer sends a payout instruction to their Guardian, which pays the recipient's Ethereum address immediately from its own Interop Pool (Appendix C) and posts the corresponding commitment to the board; at the next settlement the payer's account is debited and the Guardian reimbursed from its bucket. The recipient receives an ordinary transfer and need not know the network exists. The payer's account and balance remain private; the paying Guardian, deliberately, does not: value leaving the network is always sent by an identified Guardian, so that an outside payment has an accountable sender of record. Dealing with the outside world therefore means accepting that the counterparty learns one's Guardian, much as receiving a bank transfer reveals one's bank; activity that stays inside the network reveals not even that. Payments may also cross currencies, with both amounts fixed at the moment of payment; §9 describes the mechanism.

Payments flow inward the same way. An outside payer, on Ethereum or on a conventional rail the Guardian supports, pays the recipient's Guardian with a reference supplied by the recipient's app; the Guardian then makes an ordinary deposit on the recipient's behalf, entering the pending pool and

claimed inside a routine batch, so the chain shows a Guardian depositing and nothing about the account served. In both directions the Guardian is the network's adapter to other payment systems, and privacy degrades gracefully with the counterparty's venue: between two network accounts, full privacy; against the public chain, the outside party and the amount are visible while the network-side party appears only as a Guardian; across a conventional rail, whatever that rail discloses under its own policies, the network-side party still appears only as a Guardian.

3.7 Settling: epochs, netting, and what "final" means

An epoch is the interval between settlements. Its boundary is defined by the message board itself, as an unambiguous slice of the canonical message sequence: every message inside the slice is in the epoch, everything outside is not, and no Guardian chooses which messages count. The completeness rule is central to this construction: a Guardian's epoch proof must consume every message it signed within that epoch, and it cannot selectively omit any. As a result, once Guardian A signs a cross-Guardian message, A cannot settle around it: if A settles at all, the message settles with it. What the rule cannot do is force a settlement to occur, so every Guardian also posts Working Collateral in the Depository, and everything a failed Guardian signed but never settled is paid from it (§3.11, Appendix D).

Settlement is parallel by construction. Each Guardian proves its own subtree transition (deposits claimed, intents applied, cross-messages consumed, exits honored, the solvency identity of §3.4 intact). A permissionless aggregation step combines the subtree proofs, checks cross-Guardian consistency, and submits one proof to the Depository. The Depository verifies it, adopts the new state root, and applies a single net figure to each Guardian's bucket. Across all Guardians, cross-Guardian payments cancel in aggregate, so the figures sum to the epoch's claimed deposits less its paid exits. The public record of an epoch is that list of net figures; it does not include bilateral flows, payment counts, or individual transaction amounts. A Guardian that cannot produce a valid transition (an outage, a lost key, or books that exceed what its bucket can prove) fails to settle; after a bounded number of missed epochs it is delinquent and its subtree freezes at the last proven root (§3.8, §3.11). In other words, a failure halts and exposes the failing Guardian rather than resulting in the unobserved creation of funds.

It is worth being precise about what "final" means in this context. For the recipient, finality arrives the moment their Guardian's signed receipt does, less than one second after the intent was posted. The receipt is a protocol-enforceable claim, the sending Guardian's message cannot be settled around, and over-issuance is unprovable. Consumer-protection rules may later create an offsetting entry under §6.3, but they do not revoke or rewrite the confirmed entry. The residual risk (a Guardian failing mid-epoch with obligations outstanding) is borne by the failing Guardian itself: its Working Collateral is held in the Depository and is forfeitable against unsettled obligations (Appendix D), and the adaptive settlement cadence described below keeps each Guardian's unsettled position within that collateral. In other words, a Guardian that credits an incoming payment at confirmation is not extending uncollateralized credit to the sending Guardian, and the bound is enforced message by message: every message a Guardian signs carries an attestation that its unsettled obligations, that message included, still fit under its Working Collateral. As with card networks, users and merchants experience finality at confirmation; here the interval to settlement is minutes rather than days.

The cadence itself is adaptive. An epoch closes at the earlier of a clock and any Guardian's unsettled exposure crossing a threshold tied to its capital. Quiet periods settle at an unhurried pace, while a surge

of volume triggers settlement within minutes. We would expect operators to keep epochs short, as shorter epochs mean smaller advances and smaller exposures.

3.8 Leaving: exits and Guardian changes

Permissionless exit is the network's neutrality backstop. However, precisely because it moves funds out of the network's protections, it is also the natural target for theft and coercion. The protocol therefore allows each user to select an exit policy, pinned by the policy hash in their ledger entry: withdrawal delays tiered by amount; optional approver sets (family, a service, possibly the user's own Guardian) with thresholds by tier; and a hard-exit backstop, a long-delay path requiring no approvers at all, so that a user abandoned by every counterparty can still leave. Policy changes take effect slowly, so that an attacker who compromises a device cannot first remove its protections. The full policy space, and the reversibility window for in-network payments, are treated in §6.

Exits come in two forms. A cooperative exit is a ticket processed through the user's Guardian: policy checks, inclusion in the next epoch proof, payment from the Guardian's bucket after the delay. A unilateral exit requires nothing from the Guardian: using the published commitments of §3.4, the user proves in zero knowledge that a ledger entry under the current state root opens to their key and balance, signs with their account key, and is paid by the Depository after the same policy delay. If the Guardian is delinquent and its subtree frozen, exits proceed against the frozen root. In every case an exit finalizes against the newest proven state, after a delay no shorter than one epoch plus the dispute interval, so that every in-flight payment lands in a root before funds leave.

Changing Guardians uses the same mechanism. Cooperatively, the user's entry moves from one subtree to another inside a routine epoch proof, folded into the net settlement figures: an observer cannot determine that a switch occurred, nor which account was involved. Unilaterally (against a hostile or non-operational Guardian), the user exits without cooperation and redirects the proceeds into the pending pool under a fresh commitment naming the new Guardian, rejoining as §3.5 describes, under a new pseudonym if desired.

Two operations should be distinguished here. A withdrawal to Ethereum is necessarily public (an address and an amount), because it moves funds from the private ledger onto the transparent chain; the network therefore guarantees that a user can depart at any time, and every withdrawal is visible when it occurs. A Guardian change (described earlier in this section) moves an account between Guardians without any funds leaving the network; it is not publicly visible, and the account remains under a Guardian's compliance regime throughout. This distinction is relevant to the compliance analysis in §7.

Exit rights themselves come in three account classes, chosen at account creation and pinned by the policy hash. A self-custodial account, the default, is the account described throughout this paper: no party can hold its funds. Two supervised classes permit a lawful hold, posted as a public attestation referencing the legal process behind it; holds expire unless renewed, no account is ever reclassified without its holder's signed consent, and the on-chain footprint of every class is identical. Appendix G specifies the two supervised forms, the treatment of supervised accounts when their Guardian fails, and the manner in which authorities locate the Guardian that serves a person.

3.9 A payment, end to end

To illustrate the mechanics described above, this section traces a single payment from deposit to settlement (Appendix A walks through the same payment step by step). All figures are illustrative.

On Monday morning Alice installs a payment app, chooses Guardian A from those available in her jurisdiction, and completes its onboarding. At 9:00 she deposits 500 USDC; the chain shows an address sending 500 USDC to the Depository, and nothing else. By 9:00 and a few seconds, Guardian A has verified her commitment, opened her account, and issued her a receipt for 500.

At 9:12 she pays Bob 20 USDC for last week's dinner. Bob is served by Guardian B. Alice's app sends the encrypted intent to Guardian A, which posts it to the board, debits her on its books, and posts the cross-Guardian message; Guardian B credits Bob and sends him the notice with Alice's memo. Bob's phone shows the 20, which is final from his perspective, approximately one second after Alice initiated the payment. Since Alice's deposit is still unclaimed, Guardian A advanced the 20 from its Advance Payment Pool; the advance is invisible to Alice, and no aspect of her payment depends on her awareness of it.

At 10:00 an epoch closes. Guardian A's proof claims a batch of a few hundred pending deposits (Alice's among them, indistinguishable inside the batch), creates her ledger entry at 500, applies her payment, repays its own advance, and nets the 20 owed to Guardian B against everything else the two exchanged that epoch. The Depository verifies and adopts the new state root, and the public record shows one net figure per Guardian: Guardian A's bucket increased by an amount aggregating hundreds of unrelated claims, payments, and exits, and Guardian B's changed by another. Alice's 500, her 20, and her choice of Guardian A appear nowhere in it.

It is worth enumerating the information held by each party at the end of the day. Guardian A knows Alice's account completely; that is its role. Guardian B knows Bob received 20 through Guardian A, and does not know from whom. Bob knows the payment came from Alice, because she told him so in the memo. The chain shows that some address deposited 500 USDC that morning, and that at 10:00 the Guardians settled to new totals. Months later, Alice becomes dissatisfied with Guardian A's fees and moves her account to Guardian C. No observer, including Guardian A's competitors, can determine that the move occurred.

3.10 What the chain still shows

A precise statement of what remains public is a necessary part of any privacy analysis. The public record retains:

- **Deposits and refunds:** the depositing address, the amount, and the time. The anonymity of the subsequent Guardian assignment is provided by the cohort of deposits claimed alongside it (§3.5); refunds of unclaimed deposits are likewise public.
- **Exits:** the receiving address, the amount, and the time. A withdrawal can be initiated at any time, and it is publicly visible in every case (§3.8).
- **Bucket totals:** each Guardian's total under guardianship, continuously, by design (P5).
- **Settlement deltas:** one net figure per Guardian per epoch, aggregating deposits claimed, exits paid, and all cross-Guardian traffic. Individual payments are not recoverable from it, with the caveat that an epoch containing few other events provides a small anonymity set (the same caveat applies to deposits).

- **Message traffic:** counts and timing of encrypted messages per Guardian inbox. Guardians may pad their batches to fixed sizes to limit traffic analysis; the ciphertexts themselves reveal nothing.
- **Account counts:** approximate active-account counts per Guardian, from the published commitment updates (§3.4).
- **Settlement timing:** the adaptive cadence of §3.7 reveals when aggregate exposure crossed a threshold, and only in aggregate; a cadence floor limits even that disclosure.

Everything else (the identities of payers and payees, individual amounts, account balances, and user-to-Guardian assignments) is hidden from public observers. A user's own Guardian sees that user's account, and edge payments disclose the Guardian as described in §3.6; the protocol's privacy claim extends exactly that far.

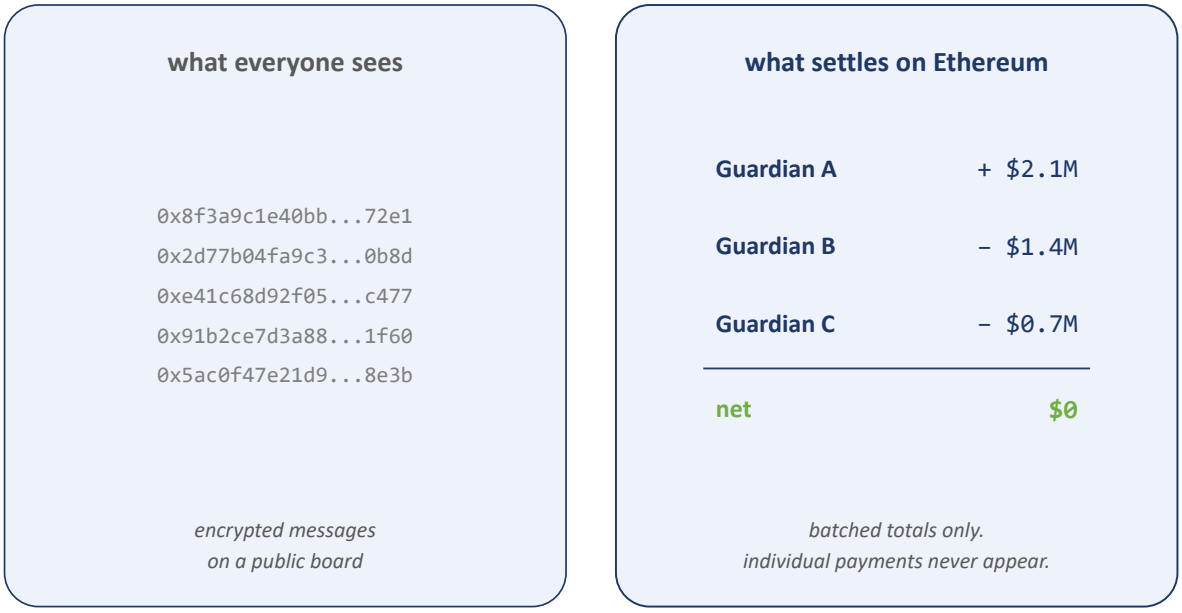

Figure 4: The observer's view. The message board carries only encrypted messages; settlement publishes one net figure per Guardian per epoch (§3.7), and individual payments never appear.

3.11 Failure modes

An assessment of the design should include its behavior under failure. The table below lists each failure mode, together with its consequence and its backstop.

Failure	Consequence	Backstop
Guardian outage or permanent failure	Confirmations pause; after a bounded number of missed epochs, the subtree freezes at the last proven root.	Unilateral exits against the frozen root; receipts and the failing Guardian's Working Collateral cover within-epoch credits; every other Guardian settles normally.
Guardian censors a user	Payments through that Guardian stop; nothing else does.	Permissionless switch or exit; forced inclusion on the message board.
Guardian over-issues (books exceed the bucket)	No valid epoch proof exists; the Guardian becomes delinquent by construction.	The solvency identity (§3.4) blocks settlement; Working Collateral covers valid unsettled credits (Appendix D).
Guardian data breach	Plaintext account data of that Guardian's users may be exposed; funds are unaffected.	Privacy insurance and the honeypot (§4, §5); threshold decryption is a future mitigation (§8).
User loses their device	Nothing is lost.	The seed restores keys; history, receipts, and the ledger entry re-derive from the message board (§3.3).
User is coerced to withdraw	The attacker is subject to the user's own exit policy.	Tiered delays, approver thresholds, and slow policy changes; the hard-exit path remains available (§3.8).
Prover unavailability	Settlement is delayed; books and receipts continue.	The Guardian may delegate proving to an operator with the witness data; aggregation and submission are permissionless.

Table 1: Failure modes and their backstops.

3.12 Alternatives considered

Four families of prior design shaped this one, and each was considered as a foundation.

On-chain encrypted balances (the Zether lineage) keep every balance on Ethereum as a ciphertext and prove each transfer correct individually. The primitives are clean and well studied. However, per-payment proofs on the base layer cannot carry a payments network's volume, and the model offers no natural locus for a compliance relationship. This design moves per-payment work onto Guardians' books and proves per epoch instead.

Shielded pools and private rollups (the Zcash lineage; Aztec as its most complete expression) offer the strongest privacy available. However, they address a different requirement: they conceal activity from everyone, operators included, while this network's premise is concealment from everyone except one accountable party the user chose. Rebuilding Guardian accountability atop a system engineered to prevent it would conflict with the design intent of both systems, and deploying on such a platform would add the operating assumptions of a comparatively new chain, and bridged assets, to the trust base. Their tooling, however, is directly reusable here (the circuits of this section could reasonably be written in the languages that ecosystem produced), and their note-encryption and viewing-key techniques set the standard from which this design borrows.

Encrypted execution (fully homomorphic encryption chains and trusted-hardware networks) would address these requirements most directly: balances that stay encrypted on chain and are mutated in place. At present the approach is bounded by performance and by trust in decryption committees or hardware vendors. It is the natural long-run direction, and the same trajectory (replacing each Guardian's single key with a threshold key held by several parties) appears in §8 as an upgrade path that changes no other part of the architecture.

Federated e-cash (the Chaumian lineage, in its modern federated forms) is closest in spirit: instant private payments through a service federation the user chooses. Its structural limit is that the federation custodies the funds. The Depository exists precisely to remove that custody (Guardians here operate accounting they can neither spend from nor freeze beyond a settlement cycle) while keeping the service relationship that makes the system usable and governable.

It is worth noting that the novelty of this design lies in its composition rather than in its components. Commitments, nullifier-style claiming, Merkle state, epoch proofs, and exit delays each have years of production history elsewhere, and the protocol requires no new cryptography. The one component that still involves genuine research questions, threshold decryption of Guardian traffic, is on the roadmap rather than the core design.

4. The Insurance Economy

4.1 The source of funds

A user's deposit does not remain idle. The Depository holds its reserves as a diversified basket of yield-bearing dollar instruments (tokenized treasury funds and similar assets); the eligible instruments are a governed list (§8), the composition within them is allocated by the capital that bears its first loss (Appendix F), and the reserve is public on-chain at every block. Balances remain denominated in dollars and redeemable one-for-one while the reserve is whole (an impairment is treated below, and in Appendix F), and the yield funds the operation of the network. In the example figures used throughout this paper, reserves earn a few percent annually, with half the yield paying Guardians and half paying privacy-insurance premiums. Premiums accrue in dollars and are paid to the underwriters, who may take them in dollars, convert them through the network itself, or compound them into their stake (Appendix F). There is no protocol token, and no fee is taken from payments; the economics of the network are funded entirely by reserve yield.

The reserve carries the risks of its instruments (de-pegs, duration, issuer failure), and those risks are borne first by the capital that chose the instruments. Diversification across multiple instruments bounds any single failure, and an impairment, should one occur, is absorbed by the Guardian's insurance pool before it reaches users, with any remainder shared pro-rata by that Guardian's balances rather than falling on the last users to redeem (Appendix F). This is the structure under which money-market funds have operated for decades, with two differences: the reserve here is visible on-chain at all times and auditable by anyone, block by block, and an insurance pool stands ahead of the balances. There is no per-user choice of basket. Each Guardian's book is backed by its own reserve sleeve, and a user choosing a Guardian reads

two public numbers: the composition of its sleeve and the coverage that stands in front of it (Appendix F).

4.2 Coverage pools and underwriters

Each Guardian stands behind an insurance coverage pool, funded from two sources: the Guardian's own co-fund, capital that it forfeits if it fails its users (15 percent of the pool, in the example figures), and underwriting capital drawn from an open market and paid from the premium stream. Any party may underwrite: funds, DAOs, individuals, and professional insurers. Underwriting is opt-in per Guardian, and premiums pay for diligence (§5.7): an underwriter backs the Guardians whose security it has examined, at prices that reflect its findings.

The natural underwriting capital is restaked ETH. Restaking lets holders of staked ETH extend the stake that already secures Ethereum to back additional commitments, and Guardian coverage is such a commitment: the stake stands behind a Guardian's coverage pool and is slashed to pay claims. Participation is open. Any holder of staked ETH may underwrite, alone or through a pool, and underwriting income adds to staking income, so the market described above remains open to every class of capital while restaked ETH is the class the design expects to dominate.

Coverage, the honeypot bounty, and the Guardian co-fund are denominated in ETH, and a payout is the promised amount of ETH at its price on the claim date. The reasons are structural. The underwriting capital is ETH, and denominating the liability in the unit of the capital removes the currency mismatch that would otherwise demand a hedge: an underwriter promises what it holds, keeps its full exposure to ETH, and needs neither derivatives nor additional collateral against exchange rates. The deterrence arithmetic of §5 consists of ratios between like-denominated quantities, and such ratios hold at any ETH price. The arrangement mirrors the base layer itself, where validators' ETH-denominated stakes secure activity denominated in every currency, and no one asks a validator to maintain a dollar peg. What users give up is a fixed fiat payout figure. What the mechanism requires in exchange is a sizing discipline rather than a hedge: the bounty is sized so that even after a severe, multi-year decline in the price of ETH it still exceeds what the underlying data would plausibly fetch in any illicit market, and that sizing is reviewed as market cycles turn. §9 adds a further advantage: in a network that carries many currencies, ETH is the single unit in which every Guardian's coverage remains comparable.

Compensation itself is credited privately inside the network, in ETH, and may be converted to the user's currency like any other balance (§9).

Diversification makes the coverage affordable. Hundreds of Guardians across dozens of jurisdictions fail, when they do fail, for mostly unrelated reasons; a breach in one country does not correlate with a breach in another. Capital can therefore back many pools at once, in the same way that every insurance market since Lloyd's has stretched capital across uncorrelated risks, and a unit of underwriting capital supports several units of user coverage. As a result, the premium stream purchases more protection than any single Guardian could purchase alone.

4.3 The visible quality signal

A user cannot inspect a Guardian's security practices, and the design does not require the user to do so. However, a user can read coverage: the size of the pool that stands behind its users, stated per user,

payable in the event of a leak of their data (§5; Appendix F describes how a payout is allocated), and standing first against losses of the reserve (Appendix F). That number is public, backed by posted capital, and directly comparable across Guardians, which makes it the market's quality signal. First, a Guardian that underinvests in security pays for that choice through higher premiums. Second, a Guardian that cannot attract underwriters cannot advertise coverage. Finally, a new entrant with excellent security and deep coverage can compete from its first day with a long-established incumbent. The capital that a Guardian posts behind its coverage terms therefore performs the role that the reputation of an established brand performs in conventional financial services. This also makes the market more competitive, as brand reputation is subject to economies of scale, whereas the insurance pool is linear with the number of users (as only the pool per user matters to a user).

4.4 Prospective Guardians

The economics admit a wide field of prospective Guardians. Banks and licensed fintechs already operate compliance functions, and guardianship is adjacent work that adds a revenue line. Merchant networks may be the most interesting entrants: a retail chain that becomes a Guardian earns the yield share on its customers' balances and is already present at the point of sale. It can fund sign-up discounts directly from that share, a distribution channel that every payments network in history has had to purchase at considerable expense. Telecoms bring existing billing relationships, and, where law permits, decentralized operators serve users who prefer not to trust any institution. None of this reach must be built in-house: a Guardian can connect to conventional rails through payment intermediaries, so outside connectivity confers no economies of scale, and the economics keep favoring a competitive market between Guardians rather than an oligopoly. The obligations are uniform for every entrant: posting the co-fund, publishing coverage terms, securing the data to the standard that §5 enforces, and answering lawful process (§7).

Each element described above (the co-fund, the premiums, the coverage numbers) rests on a single premise: that a breach, when it occurs, is detected and paid, and that no committee determines whether to acknowledge it. That trigger is the subject of the next section.

5. The Honeypot

The protocol's compliance thesis concentrates a significant risk in a single party. A user's own Guardian sees that user's account completely (§3.1): identity, balance, every payment. The architecture withholds that knowledge from every other party, but it cannot prevent the Guardian itself from leaking, selling, or losing it. However, the architecture can ensure that each of those outcomes is revealed under a standing incentive, compensated automatically once proven, and severely costly for the Guardian. That is the role of the honeypot.

Each Guardian posts a standing public bounty, the honeypot, alongside its insurance pool (§4), payable by smart contract, in proportion to what is proven, to anyone who proves possession of that Guardian's confidential dataset, in whole or in part. (The term is used in the bounty sense, a reward deliberately made public, rather than in the decoy sense of traditional security practice.) In the example figures used throughout this section, the pool is funded roughly 15 percent by the Guardian's own slashable co-fund, with the remainder underwritten by insurance providers in an open market, and the Guardian additionally posts the honeypot at 5 percent of the pool, so the bounty is the Guardian's own money and never diminishes the pool itself.

Data theft is ordinarily covert, deniable, and litigated years after the harm has occurred. Under this design, the same event proceeds differently: the theft becomes public, its occurrence can be proven, settlement occurs in code within a day or two, and the theft is reported by a party who holds the stolen data, since only such a party can make a valid claim.

5.1 What the mechanism must do

Four requirements shape the design.

H1. Detection without adjudication. A breach is proven to a contract and settled by it. Neither a committee, nor an insurer's consent, nor a court is required in the settlement path; those institutions may act afterward, but the payout does not wait for them.

H2. Verification without re-leaking. Proving that private data was stolen must not publish the private data. A naive mechanism (one that shows the contract what was taken) would make every claim a second breach.

H3. Compensation without a victim list. Paying the affected users must not identify them. A public roster of breach victims would itself constitute a disclosure.

H4. Claims without permission. Anyone can claim, pseudonymously, from a fresh address. The design assumes claimants may include criminals, disgruntled insiders, and the Guardian's own contractors, among other parties.

5.2 Ground truth: what "having the data" means

A bounty for possessing a dataset requires an incorruptible definition of the dataset. The protocol supplies one as a side effect of settlement. Every epoch, a Guardian publishes hiding commitments to each ledger entry it maintains, one per field (§3.3, §3.4), and every message it processes is bound by commitments to its core fields posted alongside the ciphertext (§3.6). The Guardian's operating database (account keys, balances, transaction histories, and the commitment randomness that accompanies them) is exactly the set of openings of those on-chain commitments, and the settlement proofs already anchor per-epoch roots for all of it. The phrase "Guardian G's confidential data from epoch X to epoch Y" therefore refers to a canonical, on-chain-committed object rather than to loose wording in an insurance contract, and knowledge of that object is mechanically testable. In other words, what a claimant proves they hold is precisely the database whose theft harms users, and because the commitments are per-field, holdings are testable field by field: the parties, the amounts, and the balances are separately provable.

The covered dataset is the protocol's own data. Records a Guardian keeps outside the protocol (identity documents gathered at onboarding, network logs) have no on-chain commitments to test against, and fall to the conventional layer of the insurance rather than to the honeypot (§5.7).

5.3 The claim game

A claim proceeds as a five-move game between a claimant and a contract, with no discretionary step at any point.

1. Claim. From any address (a fresh one, if the claimant prefers), a claim names a Guardian and a scope: a range of epochs, always covering all of that Guardian's accounts rather than a chosen subset. The

claim declares which fields the claimant holds (the parties, the amounts, the balances, or all of them) and what share of the scope, pays a fee, a tenth of what the claim would win if fully proven (as an example figure), retained by the pool whatever the outcome, and posts a binding commitment R to the claimant's copy of the claimed data, indexed in the same canonical order as the on-chain record.

2. Challenge. Only after R is fixed does the contract draw the challenge: indices sampled uniformly from the scope's ledger entries and messages, restricted to the declared fields (a few hundred samples, as an example figure), using a public randomness beacon.
3. Response. Within a short window (hours, as an example figure), the claimant submits one batched zero-knowledge proof asserting, for each sampled index, that the claimant knows the opening of the on-chain commitment and that it matches position i of the pre-committed R. The proof reveals nothing about the data itself (requirement H2).
4. Verdict. The contract verifies the proof against the per-epoch roots it already holds. The share of samples answered correctly measures the share of the claimed data the claimant actually holds, and the payout follows that measured share, up to the share declared. Understatement therefore caps a claim, and overstatement pays the fee for nothing; the honest declaration is the profitable one.
5. Consequences. Automatically, in the same transaction: the honeypot pays the claimant its proven share; the same share of the Guardian's co-fund is slashed into the pool; compensation begins for every account active in the scope, smallest accounts first, from the same share of the pool (§5.6); and the scope records the share proven. Later claims on the same scope pay only what they prove beyond the share already recorded, so a single breach cannot yield repeated payouts.

The pre-committed root R is the element that makes the sampling sound. Without it, a claimant holding nothing could wait for the challenge and then attempt to obtain just the sampled openings (by bribing a small number of users) inside the response window. Because R is fixed before the draw, answers acquired afterward are useless: they will not match a commitment made in ignorance of the sample. The measurement therefore reflects what the claimant held before the claim was initiated, not what could be gathered afterward.

Claims are public from the moment they are posted, and a pending claim against a Guardian is, by design, a source of uncertainty for observers. The game therefore resolves quickly (a day or two end to end, as an example figure), and an empty claim fails visibly and forfeits its fee. Observers are likely to learn rapidly that only proven claims are informative.

5.4 Why sampling is proof

The underlying statistics are those that secure decentralized storage networks, applied here to theft rather than to storage. A claimant holding a fraction of the claimed data answers each uniform sample with probability equal to that fraction, so the share of samples answered measures the share held, and enough samples make the measurement accurate to within a few percent. The measurement cannot be inflated: answers cannot be manufactured for openings never possessed, so claiming more than one holds only drags the measured share back toward the truth. Nor is there profit in grinding: every attempt pays the fee, the measurement is unbiased, and redrawing the samples buys noise, not data. In other words, the game pays what a claimant can prove they hold, no more, and the fee, charged on what a claim declares, makes overstatement a pure loss.

The rule that scopes span all accounts is what converts this arithmetic into a security property. Every user knows the openings of their own entries (statelessness guarantees this, §3.3), so fragments of the dataset are always legitimately in circulation. However, one user, or even a colluding hundred, holds only a sliver of the sampling frame, and a sliver claim wins only a sliver of the bounty, smaller than the trouble of assembling it. To profit meaningfully from a fabricated claim, a coalition would have to assemble a meaningful share of the entire customer base, and the situation becomes one of a customer base defrauding its own insurers, which is a scenario underwriters price rather than one the protocol can prevent (§5.7).

5.5 The economics of never wanting to win

Consider the example pool: the Guardian's slashable co-fund at 15 percent of it, and the honeypot at 5. All three quantities are denominated in ETH (§4.2), so the comparisons that follow are ratios of like units and hold at any ETH price.

The Guardian itself can always prove possession, at any share, since it holds the data. It never profits by doing so. A self-claim through a proxy collects a share of the honeypot, forfeits the same share of the larger co-fund, and endangers the fee stream and client base that made the guardianship business viable. Provided that the honeypot is smaller than what the Guardian stands to lose (the example figures leave a threefold margin at every share, before counting the business itself), self-breach is strictly value-destroying, including for a Guardian already winding down, since an orderly exit returns the co-fund intact.

Employees present the opposite case, and the one the mechanism is designed to address. An insider owns none of the co-fund; for an insider, the honeypot is a standing offer with no offsetting loss. A Guardian must therefore operate so that no employee (and no plausible coalition of employees) can assemble any share of the dataset worth claiming, which in practice means encrypting storage under keys no single person holds, sharding access, and placing alarms on data movement. As a result, the party with the strongest incentive to protect the users' data becomes the Guardian itself, and the honeypot funds that protection continuously, without the involvement of an auditor.

The subtlest effect concerns the covert market. A stolen copy escapes the honeypot only if every party who ever handles it declines the standing bounty (what escapes the game itself is a copy stripped of its commitment openings; §5.7). A buyer of stolen data can take delivery and then claim the bounty, anonymously and profitably, slashing the seller's co-fund in the process. Every covert sale must therefore survive a standing offer of payment for defection. In addition to detecting breaches, the honeypot undermines the mutual trust that transactions in stolen data require, and this trust problem is more difficult for the parties involved to overcome than encryption.

For the claimant, the bounty is designed to be more attractive than a covert sale: it pays without identity, without negotiation, and without the trust problem a buyer would otherwise present. The honeypot is therefore likely to act as a floor price on the dataset, in the sense that data worth less than the bounty is more profitably claimed against the pool than sold covertly. The exposure is also bounded by retention: a claimant can prove possession only of what the Guardian's database contained when it was taken, so a Guardian that retains less recorded history has less that can be stolen and less that can be claimed against it. The honeypot therefore prices data retention directly, and rewards data minimization within whatever retention the law requires. The effect is prospective only: deleting data cannot immunize a copy that has

already left the Guardian, because a claim is proven by whoever holds the copy, against the chain's own commitments, and the Guardian has no role in the verification.

It is worth noting that nothing in the mechanism imposes a deadline on claims. A Guardian's liability for a leaked scope therefore persists for as long as its pool stands, which is the permanent form of accountability the design intends. One consequence is that a lone holder of stolen data faces no time pressure to come forward, although the moment a second party holds the copy, each holder risks the other claiming first. If prompt disclosure were desired as a property in its own right, a bounty that decays with the age of the claimed scope would be a straightforward design option; this paper leaves the bounty flat.

5.6 After a proof

Every consequence of a proven claim executes in code, in a fixed order.

- **The bounty.** The honeypot pays its proven share to whatever address claimed. No identity is requested; requirement H4 is maintained through the final step.
- **The slash.** The same share of the Guardian's co-fund moves into the pool, joining the insurers' capital available for compensation.
- **Compensation.** Accounts active in the breached scope are compensated from the claim's share of the pool, smallest accounts first and as far as that share reaches, under the policy's terms, allocated as Appendix F describes, credited through the ledger itself. The pool's funds enter the Depository and reach affected accounts as ordinary private credits over the following epochs, through whichever Guardian serves each user by then. No roster of victims is ever published (requirement H3); the compensation is as private as the payments were.
- **The mark.** The scope records the share proven, and later proofs of overlapping scope pay only what they establish beyond it, so a single theft pays a single bounty.
- **Rotation.** The stolen dataset begins depreciating immediately. Affected users can move to a new Guardian under fresh account identifiers using the ordinary switching machinery (§3.8), cooperatively, or unilaterally if confidence has been lost. As a result, the stolen data describes accounts that, within days, no longer transact.
- **Repricing.** The effect of a proven breach on the Guardian's premiums, its co-fund requirements, and its viability is addressed in §§4.2–4.3.

5.7 What the honeypot cannot see

The mechanism has blind spots, and they are stated below.

- **Knowledge that is not the dataset.** An analyst's memory, a verbal disclosure, or a copy stripped of its commitment openings cannot be sampled; a commitment that plaintext alone could test would be one that anyone could brute-force, so this boundary is the price of the hiding itself. Such harms fall to the Guardian's contract terms, to privacy law, and to the conventional layer of the insurance.
- **Data without commitments.** Identity documents, network logs, and other records held outside the protocol have no commitments to test against. Extending commitments to such records is possible in principle and is deliberately excluded from the core design.

- **Small fractions.** A claim wins in proportion to what it proves, so a small enough share is worth less than the trouble of claiming it. The automatic mechanism therefore reaches as far down as a claim's economics reach, and no further; exposure below that line falls to the pool's ordinary policy terms (§4).
- **Manufactured Guardians.** A fabricated Guardian with fabricated users could pass its own sampling and drain a pool. The protocol gates the honeypot on a pool's age and deposit history; the decisive defense is that underwriting is opt-in. Insurers choose which Guardians to back, and diligence is what premiums pay for (§4).
- **Provenance.** A copy that left the Guardian through lawful process, and leaked from its recipient, is possession all the same. How fault and slashing should apportion in such cases is a genuine open question, and this paper leaves it open (§8).

Within those limits, the guarantee is exact. Any complete copy of a Guardian's dataset, wherever it resides, is a liability to whoever holds it, and anyone to whom the holder shows the copy gains an incentive to claim the bounty against them.

6. Using the Network

6.1 Everyday use

From the user's perspective, the mechanisms described in §3 reduce to a short sequence of steps. A user installs a payment app, selects among the Guardians available in their jurisdiction (with coverage, fees, and protections displayed side by side), completes that Guardian's onboarding process, and deposits funds from any wallet or exchange; a mature network adds deposits made directly from bank rails. From that point, payments to any other participant on the network confirm in about a second and carry no cost at the point of use (§6.4). Payments to anyone outside the network, on Ethereum or on conventional rails, go through the user's Guardian and reveal it to the counterparty (§3.6). The entire account is controlled by a single seed phrase, so the loss of a phone costs only the time required to restore access (§3.3). In addition, the Guardian can be changed in much the same way a phone number is moved between carriers, including over the objection of the previous Guardian (§3.8). Users do not handle gas, do not sign raw transaction data, and do not encounter an address unless they leave the network; Guardians perform the chain interaction on their behalf, which is one of the services the yield share funds.

6.2 User security policies

Permissionless exit is the mechanism that guarantees the network's neutrality, and it is also the path a thief would use, so each user sets their own exit policy, which is enforced by the protocol itself and cannot be removed by the user's Guardian (§3.8). In example configurations, the policy space is structured as follows: small withdrawals complete quickly; larger withdrawals are subject to delays scaled to the amount; and the largest withdrawals require approvals from a set of parties the user has chosen (family members, a lawyer, a service, and optionally the Guardian), at thresholds such as one-of-three for medium sums and two-of-three for large sums. Policy changes take effect slowly, so that compromising a phone does not compromise the policy. Beneath these layers is the hard exit, a long-delay path that requires no approvals at all, which ensures that a user abandoned by every counterparty can still withdraw their funds.

The same approver machinery also serves as a recovery mechanism: keys can be restored socially, without any Guardian's permission.

6.3 Bounded reversibility

Between Guardians, payments are final at confirmation (§3.7). For consumers, finality is adjustable within published limits: within a stated dispute window, a payment can be unwound (as a new, co-signed offsetting entry, evidence-based and jurisdiction-specific) under the protection policies of the Guardians involved. Different Guardians will offer different windows and standards, merchants advertise theirs, and users choose accordingly. Two constraints preserve the integrity of this reversibility. First, reversals are policy applied above the ledger; the canonical record itself is never mutated and only moves forward. Second, exit delays exceed dispute windows (§3.8), so value under dispute cannot leave the network before the dispute resolves. The purpose of bounded reversibility is deterrence, and any reassurance it offers users is secondary. A fraudulent scheme, a robbery, or a coerced transfer inside the network can be undone within the window, and as a result the network is an unattractive venue for theft.

6.4 Fees

We would expect ordinary payments to be free at the point of use, funded by the reserve yield in much the same way card rewards are funded by interchange, except that no interchange is charged. Guardians compete above the free tier, offering services such as instant external payouts, merchant tooling, invoicing, credit, and enhanced protection services.

7. Compliance

7.1 One protocol, many regimes

Compliance in this design belongs to Guardians. The protocol itself has no compliance policy of its own. A helpful analogy is the internet's transport layer, which has no content policy, while the services running on it answer to their regulators. Concretely, this works as follows. First, jurisdictions publish, or recognize registries of, the Guardians permitted to serve their residents. Alongside that list, a jurisdiction selects which account classes (§3.8) its approved Guardians may offer or must require for new accounts, and sets the waiting period that governs class degradation when a Guardian fails. Second, payment applications (native mobile apps, of which there may be several compatible implementations) enforce the applicable list at the user-experience layer. Finally, Guardians implement their jurisdiction's identity, recordkeeping, and reporting standards as their own policy. The protocol carries whatever those policies require (including the encrypted originator-information slot of §3.6 for travel-rule regimes) and mandates none of it globally. In other words, one protocol serves many regimes without imposing a one-size-fits-all standard, whether the weakest or the strictest, on any jurisdiction.

The same structure reaches payments that never touch the public chain. A cross-Guardian payment identifies the counterparty Guardian to each side, never the underlying account (§3.6), so a Guardian can screen at the clearing layer in the manner of a correspondent bank, declining to clear with counterparties that do not appear on the lists its jurisdiction recognizes. Value that circulates entirely inside the network therefore moves only between identified operators, each answerable to its own authority for the clearing relationships it maintains.

7.2 The audit model

A user's Guardian can respond to lawful process concerning that user in the same manner as a bank, providing identity information to its onboarding standard, transaction records, and supporting evidence. Two properties distinguish these records from a bank's. First, they are tamper-evident: a Guardian's history must open commitments that the chain fixed long ago (§3), so records cannot be retroactively altered by any party, including the Guardian. Second, they are precisely scoped: a Guardian can answer about its own users and is unable, as a matter of architecture rather than policy, to answer about anyone else's, because it never held the data (§3.4). As a result, lawful access flows through the party accountable for it, under the law that binds it. It is worth noting that because no data exists at the protocol level, there is no protocol-level mechanism through which such access could be granted.

The manner in which an authority locates the Guardian that serves a given person is addressed in Appendix G.

7.3 Nothing to ban

Two properties establish that no government needs to ban this network. First, entry is screened where jurisdictions want screening: Guardians decide whom to onboard, under their own law (§3.5). Second, withdrawals are public: funds leave the network only onto Ethereum's transparent ledger, at a visible address and amount, while a change of Guardian keeps the account inside the network, under another Guardian's compliance regime (§3.8). Value still inside the network moves only between Guardians that identify and may screen one another as clearing counterparties (§7.1). The protocol guarantees that a user can always leave, and it makes every withdrawal visible. In other words, anyone who exits the private layer surfaces on a transparent one, so the network cannot function as a repository in which value disappears permanently from public view.

Banning the protocol would therefore gain a government nothing that regulating Guardians does not already provide with more precision.

8. Governance and Roadmap

8.1 Governance

The settlement layer is designed to be boring. The properties that make the network trustless are immutable: the solvency identity, the proof requirements, and every exit right. No authority, however constituted, can move funds, alter proven state, or block an exit. What remains governable is a short list of parameters: the deposit lock window, epoch cadence bounds, sampling sizes, the honeypot and co-fund fractions, the list of eligible reserve instruments, and the operating parameters of the liquidity pools and Backup Pools (Appendix E). Parameter changes execute behind long timelocks with public notice, always longer than the hard-exit path, so that any user who objects to a pending change can leave, with everything they own, before the change takes effect. In other words, the exit right functions as the network's constitution. The body that proposes parameter changes is deliberately minimal and will be specified with the implementation.

8.2 Roadmap

These are upgrade paths for whoever implements the network, not commitments by the author.

- **Threshold Guardians.** The largest residual risk in the design is the compromise of a single Guardian's key. The upgrade path replaces each Guardian's key with a threshold key held across several independent parties, so that no single compromise (a hack, an insider, or a seizure) decrypts anything. Jurisdictions could even mandate committee structures for lawful access (for example, requiring two licensed parties to cooperate). Nothing else in the architecture changes.
- **Multi-Guardian routing.** Routing payments across multiple Guardians, so that even a user's own Guardian observes less than the whole of each transaction. This is the shared-key counterpart to threshold decryption.
- **Bank-rail onboarding.** Deposits and withdrawals directly from traditional bank accounts and payment applications, so that onboarding does not require an account at a cryptocurrency exchange.
- **Provenance adjudication.** The open question of §5.7, namely how fault should be apportioned when lawfully disclosed data leaks from its recipient. Approaches under consideration include decentralized adjudication.
- **Identity commitments.** The optional extension bringing off-protocol records, such as onboarding documents, under the honeypot's coverage (§5.7).

Payments are only the base layer of financial life. A network that holds payment history under user-controlled disclosure can support reputation; reputation, in turn, can support credit; and Guardians with merchant roots have extended credit since department stores invented the charge card. Private, uncollateralized lending against portable credit reputation is the natural next step, and it is the subject of a separate paper.

9. Many Currencies

The body of this paper described one asset: a private dollar backed by a diversified reserve. Nothing in the architecture is specific to that choice. The Depository can maintain a registry of assets, each defined by an underlying token and a reserve policy, and each backed one-for-one by its own asset pool; additional currencies, and several versions of the same currency with different backings, are separate entries in one registry. The non-custody guarantee is stated per asset, so the consequences of any issuer's action are confined to the asset whose holders chose that issuer; a user who prefers not to depend on any single issuer can simply hold more than one asset. Balances are kept per asset throughout. The Depository maintains one public running total per Guardian per asset, the solvency identity of §3.4 is proven for every asset separately in its own units, and no party ever supplies an exchange rate of its own: every rate a payment uses is one the Depository itself published (Appendix E).

Payments cross currencies at the same speed as any payment, with both amounts fixed at the moment of payment: the payer's instruction carries the pair of figures, each Guardian's leg proceeds in its own asset, and the conversion settles against the Depository at the rate the payment used. The Depository publishes the rates, carries the movement of rates between payment and settlement, and funds that role from reserve yield, a small conversion spread, and the fees of liquidity pools that it operates itself; no Guardian quotes a rate, holds currency inventory, or carries exchange-rate risk. Appendix E specifies the mechanics.

10. Conclusion

If we want mass adoption, expecting every government to change their laws for Ethereum is a non-starter. Mass adoption of payments requires native mobile apps. And governments can easily demand that Apple and Google ban a non-compliant app in their country.

The vast majority of people are not going to run Tor with a VPN, while storing their shielded-wallet keys on an encrypted USB just to make private payments.

Instead, they'll look to PayPal, or Visa, or Stripe, and say 'good enough for me, whatever.'

And we could have offered them so much better.

We can offer them so much better. But it has to check all the boxes.

That's the intent of this design. Do maximum good. Offer the most privacy to the most people we can. All the while driving Ethereum mass adoption.

And then we live to fight another day, so that we can offer even more privacy in the future.

But let's keep that part between us. Privacy being important and all.

Appendices

Appendix A: How a Payment Works

Appendix B: How a New User Joins

Appendix C: Interop Payments

Appendix D: Guardian Working Capital

Appendix E: Multiple Asset Pools

Appendix F: Reserve Risk and the Insurance Pool

Appendix G: Supervised Accounts and Lawful Process

Appendix A: How a Payment Works

This appendix follows one payment, Alice paying Bob twenty dollars, through the three components involved. All figures are illustrative.

The Depository. One smart contract on Ethereum that holds all deposited funds at all times; the vault.

Guardians. Regulated service businesses. Each keeps the books for its own customers and never holds their money. Alice uses Guardian A; Bob uses Guardian B.

The message board. A public venue for encrypted messages. Observers see sealed mail moving; none of it can be read.

The vault holds the money, Guardians keep the books, and the board carries the mail.

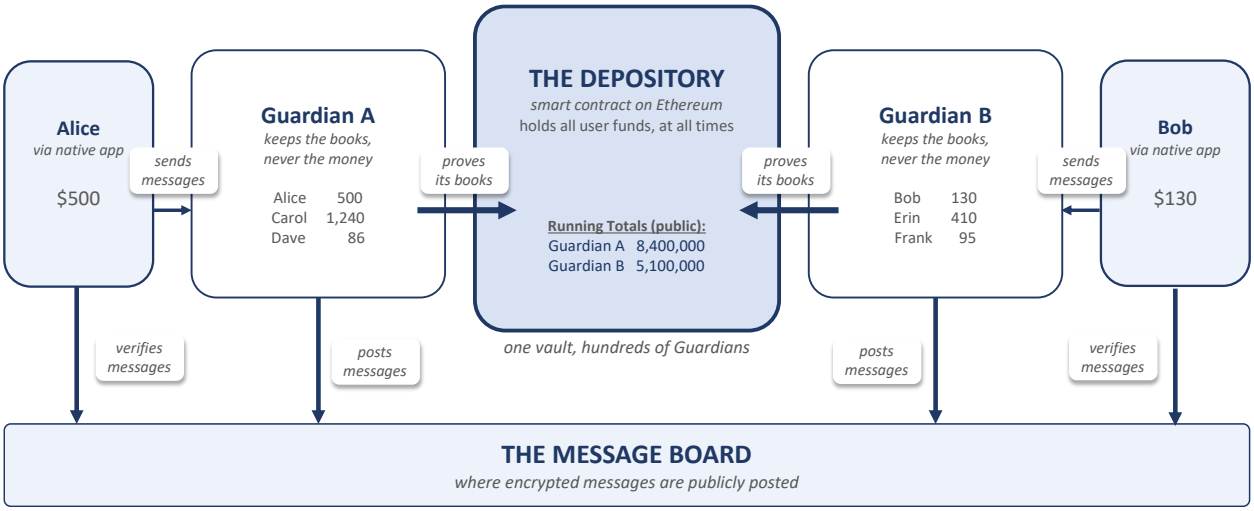

Figure A1: The components. One vault holds the money; Guardians keep the books; the board carries sealed mail.

A.1 The instruction

Alice's app builds an instruction: pay 20 to Bob, addressed by a one-time tag derived from a handle he shared, with an optional memo. The app signs the instruction with her key, so no other party, her Guardian included, can create, alter, or replay a payment from her account. It seals the result in layers, an outer envelope only Guardian A can open, an inner envelope only Guardian B can open, a memo only Bob can read, and sends the sealed envelope to Guardian A, which posts it to the message board beside a fingerprint of the contents: a short public code that fixes the payment's terms.

The app then watches the board for its own instruction. The instruction carries a signed expiry: if it has not appeared within a few seconds, it lapses and a late posting cannot execute it, so the app can simply send it again. An occasional lapse is an ordinary failure, cured by the retry. A Guardian that repeatedly drops one user's instructions while visibly posting everyone else's is censoring, and does so publicly; the remedy is to switch Guardians.

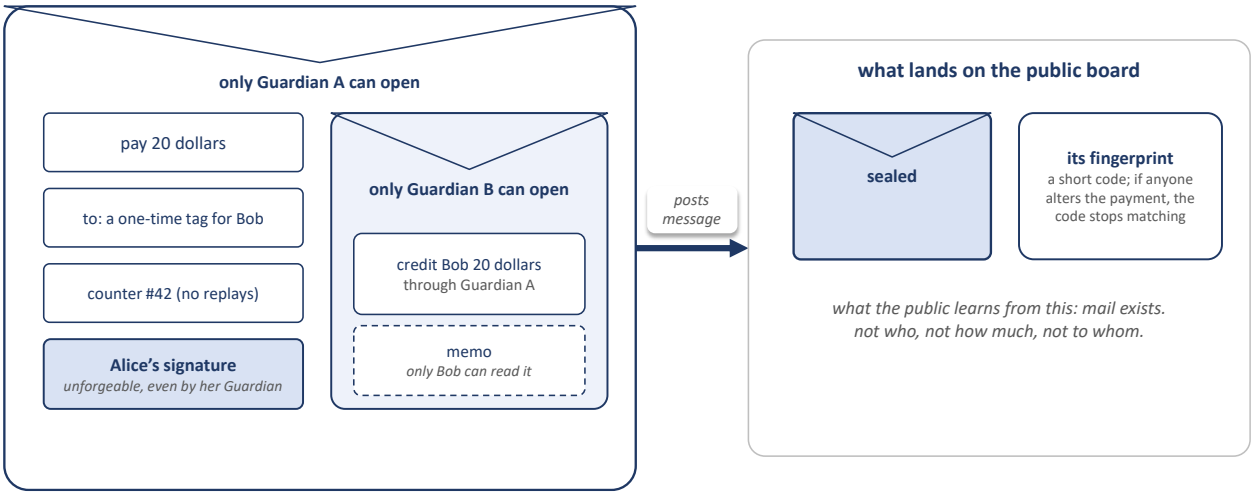

Figure A2: The layered instruction. Each party can open only its own layer.

The 'one-time tag for Bob' explained:

Bob's handle, the QR code he shared, is not his account address; it is a small kit for building addresses. It holds a key of Bob's for deriving tags, a key for sealing the inner envelope, and routing information that tells a Guardian where to forward that envelope, sealed so only Guardians can read it. For each payment, Alice's app combines the derivation key with fresh randomness to produce the tag: a value that looks random to everyone, that Guardian B can recognize as Bob's with a matching key it holds, and that shares nothing computable with any other payment's tag. Alice can pay Bob every week from the same handle, deriving a different tag each time, and Guardian A, which reads her instructions, can never tell those payments go to the same person.

The handle also keeps Bob's choice of Guardian to itself. Its sealing key is a blinded copy of Guardian B's key, freshly blinded for every handle Bob mints, so it matches nothing in the public directory of Guardians, and no two of Bob's handles can even be linked to each other. Alice's app can use what the handle carries without being able to identify any of it: it seals an envelope for Bob's Guardian, and attaches routing that reaches Bob's Guardian, without ever learning which Guardian that is.

Handles are therefore cheap to be generous with. Bob can print one on every invoice for years, or hand a distinct one to each counterparty, and no collection of his handles, however public, maps him to a Guardian or clusters his payers.

A.2 The one-second path

Guardian A opens the outer envelope, verifies the signature, checks the payment counter against replays, and confirms the balance covers the amount. It writes Alice's 500 down to 480 on its own books, returns a signed receipt, and forwards the inner envelope to Guardian B on the board, now carrying Guardian A's own signature.

Guardian B opens its envelope, writes Bob's 130 up to 150 on its own books, and sends him a notice enclosing the memo. Bob's balance is final from his perspective approximately one second after Alice initiated. The vault has not yet been involved.

A.3 Why instant credit is safe

Guardian B credited Bob on nothing more than a signed message from Guardian A, and two mechanisms close the two ways A could fail it. First, a completeness rule: a settlement proof is valid only if it accounts for every message its Guardian signed. If Guardian A settles at all, the payment settles with it; A cannot settle around it. What the rule cannot do is force a settlement to occur, so, second, every Guardian posts Working Collateral (part of its Working Capital, which is covered in Appendix D) in the Depository, and every message it signs carries a small proof that its total unsettled obligations, this payment included, still fit under that collateral; Guardian B checks the proof before crediting Bob. A Guardian that fails to settle is frozen, and every obligation it signed is paid from its Working Collateral, which those proofs kept sufficient. Selective omission is impossible; outright failure is collateralized.

Bob's own claim is the receipt, a signed instrument from his Guardian, backed by the audit that follows.

A.4 Settlement

Settlement runs on a cycle of minutes to an hour. At each cycle every Guardian publishes a zero-knowledge proof: a short object anyone can verify, establishing that its customers' balances sum exactly to its running total at the vault, that every message it signed was applied, and that no balance went negative, without revealing any balance. Books that violate these conditions admit no valid proof, so a dishonest cycle cannot settle; fractional reserve is prevented rather than detected.

The Depository verifies the proofs and applies one net figure per Guardian: A's total falls, B's rises, each figure folding that Guardian's entire cycle of deposits, payments, and withdrawals into a single number. The individual payment is not recoverable from the public record.

The money itself never moved. The vault holds the same twenty dollars; what changed is who it owes, Alice 20 less and Bob 20 more. A payment here is a reassignment of the vault's obligations, private and provable.

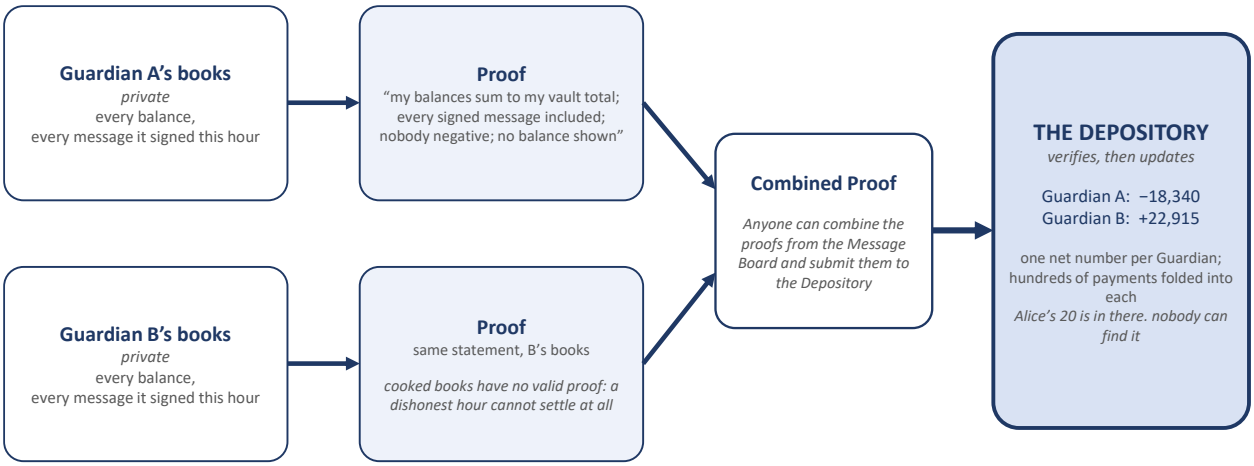

Figure A3: Settlement. Each Guardian proves its books; the vault applies one net figure per Guardian.

A.5 What each party knows

Alice knows her own payment. Guardian A knows its customer sent 20 to an account at Guardian B, under a one-time tag that links to none of her other payments. Guardian B knows its customer received 20 through Guardian A, sender unnamed. Bob knows the sender only because the memo named her. Public observers see encrypted traffic and, at settlement, netted totals: no names, no amounts, no payments.

Alice	Guardian A	Guardian B	Bob	everyone else
-20, to Bob *does not know Bob's Guardian*	Alice paid 20 to someone at Guardian B *the tag is one-time: it can't tell her next payment goes to the same person*	Bob received 20 through Guardian A *from whom? it has no idea*	+20, from Alice *he knows the sender only because her memo said so*	sealed mail moving; hourly totals shifting *no names, no amounts, no payments*

one payment, five views, each exactly as wide as it needs to be

Figure A4: What each party ends the day knowing.

A.6 The same-Guardian case

When payer and payee share a Guardian, the payment is two entries in one book and two receipts, confirmed in the same second. The receipts count against the Guardian's Working Collateral until the next proof lands, so within-cycle credits are safe here too. Only encrypted messages appear on the public board; the payment itself remains invisible inside the Guardian's next net settlement figure. The two-Guardian path above is the more complex case.

A.7 Summary

The vault holds the money, so no Guardian can take or freeze it. Guardians keep the books, so payments stay private, with a regulated party accountable for each account. Signatures make payments unforgeable, proofs make books honest, and Working Collateral makes instant credit safe.

Appendix B: How a New User Joins

Appendix A followed twenty dollars between two established users. This appendix follows the harder case: Carol installs a payment app, deposits 500 dollars, and pays David, served by Guardian D, within a few minutes, before the canonical ledger contains an account for her. One component appears that Appendix A did not need: the pending pool, where deposits rest until their Guardian claims them. All figures are illustrative.

B.1 A deposit that names no Guardian

Carol chooses Guardian C from those available in her jurisdiction and completes its onboarding first, to whatever standard C's regulator requires. Nothing is at stake during this handshake: no funds have moved, and a Guardian never receives a deposit from a user it would have declined.

Her app then sends 500 dollars to the Depository. Attached to the deposit is a sealed code binding it to Guardian C and to her account, readable by no one; she handed Guardian C the code's opening during onboarding, so C alone can ever claim the money. The funds rest in the pending pool, locked for about a day. Publicly, the entire event is that an address deposited 500 into the network: no Guardian named, no account, and no link to anything after.

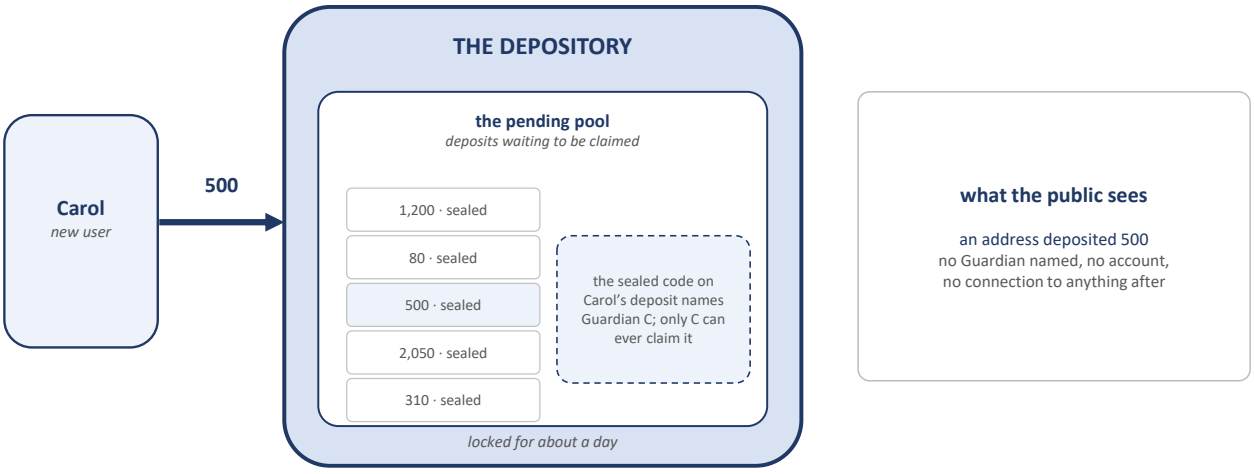

Figure B1: The deposit. Sealed in the pending pool; the public sees an amount, once.

B.2 Service in seconds

Guardian C verifies that the sealed code names it and opens Carol's account on its books at 500, with a signed receipt. It can do this safely before claiming anything: the lock means the deposit cannot leave, and the code means no rival can take it. The claim is guaranteed, so the service need not wait for it. Carol has a working account seconds after depositing.

B.3 Paying before the network knows her

Two minutes after joining, Carol pays David twenty dollars. Everything in Appendix A happens unchanged: the layered instruction, Guardian C's checks and posting, David's credit at Guardian D within the second, the receipts, and the same small proof that everything Guardian C has signed fits under its Working Collateral. Neither David nor Guardian D can tell that a brand-new user paid them.

One thing differs inside Guardian C. When the payment settles, the debit cannot come from Carol's balance, because she has none in the ledger yet; her 500 is still sitting in the pending pool. It comes instead from Guardian C's Advance Payment Pool (part of its Working Capital), a balance of Guardian C's own money kept for exactly this: fronting the payments of users whose deposits are not yet claimed, and recovering the money when they are. The advance is not publicly visible, costs Carol nothing, and no aspect of her payment depends on her knowing it happened.

B.4 The claim

At a moment of its choosing within the lock window, Guardian C claims Carol's deposit inside a routine settlement proof, batched with hundreds of other pending deposits. The proof establishes that the batch is valid and its sum correct, never which deposits it contains, and the sum folds into C's single net figure for the cycle. The same proof creates Carol's entry in the ledger at 480, her payment already applied, and repays the Advance Payment Pool its twenty, a movement inside Guardian C's own books that is visible nowhere.

Batching provides for Carol's anonymity. An observer sees deposits enter the pending pool and sees Guardians' totals rise at settlements; connecting the two means picking her deposit out of everything claimed alongside it. Waiting lets the anonymity set grow, which is why the claim is deferred and why the Advance Payment Pool exists.

B.5 Leaving before fully joining

The lock has a second job. A deposit still unclaimed when its lock expires becomes refundable by the depositor, permissionlessly. If Guardian C never claims, from failure or refusal, Carol recovers her full 500 by herself, from the same public contract that holds it. Entering the network requires a Guardian's consent; recovering your money never requires anyone's.

The same path protects her if Guardian C fails mid-flight. Carol recovers her deposit; David keeps his twenty; and any valid obligation left unsettled is paid from Guardian C's Working Collateral (Appendix D).

B.6 Who learned what

Carol has a working account in seconds and holds receipts for all of it. Guardian C knows its new customer in full, which is its role and its accountability. David and Guardian D receive an ordinary payment; nothing marks the payer as new, and nothing names her. Everyone else sees an address deposit 500, once. The protocol does not publicly link Carol's later in-network activity to that deposit or to her account: joining is the loudest thing she does here, and it discloses only the amount.

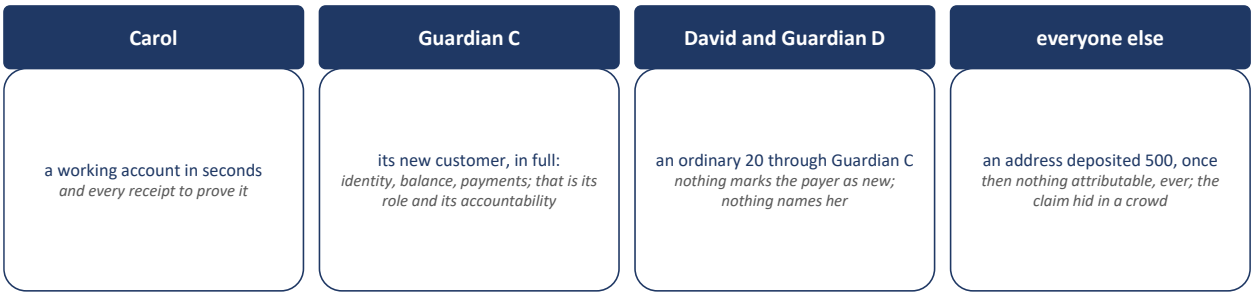

Figure B2: What each party ends the day knowing.

B.7 Summary

A new user is served in seconds because her Guardian's claim is guaranteed before it is exercised. The pending pool holds the money, the lock guarantees the claim, the Advance Payment Pool bridges the wait, the batch hides the Guardian assignment, and the refund path unwinds even a half-finished joining. The only unavoidable public disclosure is the originating address sending 500 into the protocol, because Carol begins with a balance on a public blockchain.

Appendix C: Interop Payments

The network has an inside and an edge. Inside, payments run between accounts with the full privacy of Appendix A. At the edge, money crosses into systems with different rules: Ethereum addresses, bank accounts, and cards. This appendix follows both directions of that crossing, which we call interop payments. One rule governs them: crossing the edge means using one's Guardian as the gateway and exposing that Guardian to the counterparty, much as a bank transfer reveals the sending or receiving bank. The protocol does not expose the user's network account. All figures are illustrative.

C.1 Paying out

Alice owes Ellen 20 dollars, and Ellen has never heard of the network; she has an Ethereum address, or a bank account. Alice's app sends Guardian A an instruction, as in Appendix A. Guardian A pays Ellen directly: an ordinary transfer to her address, or a payment across a conventional rail, its own or an intermediary's. The money comes from Guardian A's Interop Pool, a float of the Guardian's own funds held outside the protocol, part of its Working Capital (Appendix D). At the next settlement Alice's balance is debited and Guardian A is repaid inside the ledger, so the advance is outstanding for minutes. Ellen receives an ordinary payment from Guardian A and never needs to know the network exists; Alice's network account and balance appear nowhere on the settlement rail, and the transfer need not name her.

C.2 Being paid

Frank owes Alice 20, and this time the outsider is the payer. Alice's app shows him a code; the code tells him to pay Guardian A, on whatever rail he already uses, with a reference only Guardian A can resolve. Frank pays, and his part ends there: he installs nothing, joins nothing, and learns only that Alice is served by Guardian A.

Guardian A resolves the reference and, from Alice's side, does exactly what Guardian B did for Bob in Appendix A: it credits her on its books at once and posts her a signed notice on the message board, a receipt attested and backed the same way. One thing is missing compared to Appendix A: there is no payer inside the network, so no debit arrives from any other Guardian. Guardian A supplies that side itself, with an ordinary deposit on Alice's behalf into the pending pool, claimed inside a routine batch; the claim is what carries her credit into the settled ledger. The chain records that Guardian A deposited; which account it served is hidden inside the claim batch, as in Appendix B. If Frank's rail settles slowly, as cards do, the gap is carried by Guardian A, from its Interop Pool, never by Alice.

C.3 Three levels of privacy

The table below states what each scenario reveals to outside observers. Inside the network, no payment details are visible beyond the participants and their own Guardians. At the edge, the outside leg is as visible as its rail makes it, in both directions, while the network-side party appears only through a Guardian. In every case, the network itself adds no disclosure; what is revealed is revealed by the rail.

	the payer	the recipient	the amount
within the network	hidden *only the Guardian knows*	hidden *only the Guardian knows*	hidden *only the two Guardians know*
interop, paying out	appears only as the payer's Guardian	visible to the receiving rail	visible to the receiving rail
interop, being paid	visible to the paying rail	appears only as the recipient's Guardian	visible to the paying rail

Figure C1: What each scenario reveals.

C.4 Summary

A user can pay anyone, and be paid by anyone, however the other side transacts: in the network, on Ethereum, or on conventional rails, through their Guardian and its Interop Pool. The counterparty learns the Guardian; the user behind it appears on the Guardian's books and nowhere else. Full privacy remains wherever both sides are inside the network; crossing the edge discloses one fact, the institution the user chose, and nothing more.

Appendix D: Guardian Working Capital

The one-second experience runs on promises made ahead of settlement, and promises need money behind them. Each Guardian maintains that money in three pools with different jobs, known together as its Working Capital:

The Working Collateral lives in the Depository, in public: it backs everything the Guardian has signed but not yet settled, for everyone who credited on the Guardian's word. This balance is publicly visible because

its level tracks no user activity, and everyone needs to see it because it is collateral for the case where a Guardian stops operating (which we'll call a 'Guardian death').

The Advance Payment Pool is for fronting the payments of new users whose deposits are still unclaimed (Appendix B). This pool balance is private, so that a new user's first payments cannot be correlated with movements out of the pool.

The Interop Pool lives outside the protocol altogether: the Guardian's own wallets and accounts on other rails, for bridging payments that cross the network's edge (Appendix C). This balance needs no privacy and no proofs; the vault never sees it.

D.1 Working Collateral

Consider the payment of Appendix A at the moment its Guardian dies: Guardian A signed the message, Guardian B credited Bob, and A never settles. The same 20 dollars now stands promised to two people. It is promised to Alice, through her right to withdraw against the last proven ledger, which the design will never break; and it is promised to Bob, through B's credit backed by A's signed message. Two live promises, one 20 in the vault (the last proven ledger still shows Alice at 500; the debit died unproven), and honoring both takes 40. The second 20 has to have been put somewhere seizable in advance, by the party whose death created the double promise. That pre-posted second 20 is the Working Collateral.

What keeps the promises inside the pool is the small proof Appendix A mentioned: every message a Guardian signs carries an attestation that its total unsettled obligations, the new one included, still fit under its Working Collateral. The running total this proves is called the Guardian's proven outstanding liabilities. An obligation without a valid attestation earns no instant credit and no claim, so at Guardian death the claims can never exceed the pool, and every valid claim pays in full. A Guardian near its cap does not become dangerous; it becomes slow, until the next settlement clears its outstanding obligations and frees the pool again.

D.2 Advance Payment Pool

The Advance Payment Pool is, in every mechanical respect, a user balance. The Guardian funds it the way any user joins: a deposit into the Depository carrying a sealed code, claimed into an account the Guardian keeps on its own books, the Guardian serving as its own customer. The dollars sit in the vault alongside everyone else's; the balance stays private inside the Guardian's books; the public sees a deposit happen once and the level never again. What makes the account special is only its job: it never originates

payments. It silently funds the settlement legs of other users' payments: those of users whose deposits are not yet claimed (Appendix B), and either side of a payment that crosses assets (Appendix E).

The pool needs no oversight, because settlement supplies it: a proof in which any balance goes negative does not exist, so the pool cannot be overspent, and once an advance settles it is already paid, in the proven record, with the Guardian's own money. At a Guardian death there is accordingly nothing to seize and no claim to file: whatever the pool advanced and settled was paid when it settled, and anything that never settled is a claim on the Working Collateral instead. Nothing falls between the pools.

D.3 Interop Pool

The Interop Pool is the third pool, and the first that lives entirely outside the protocol: the Guardian's own wallets and bank accounts on other rails. No protocol proofs govern it, no protocol rules constrain its balance, and the vault never sees it.

Its job is bridging rail latency, in both directions. Paying out, the pool pays Ellen now, against a reimbursement that arrives minutes later at settlement (Appendix C). Being paid, the bridge is longer and points the other way: Frank's card payment may take days to arrive, while Alice's claim is increased right away, and the pool carries that gap as well.

The pool never carries user risk. An instant credit for an inbound interop payment is a receipt like any other: attested, counted against the Guardian's proven outstanding liabilities, and therefore a valid claim on the Working Collateral at a Guardian death, even if the deposit behind it was never made. Users never carry rail risk. The Guardian alone carries chargebacks, reversals, and its counterparty rails, inside a pool that is entirely its own money.

D.4 Capital efficiency

The two in-protocol pools are separate homes, but the split doesn't increase capital requirements. The portion of the Working Collateral above the proven outstanding liabilities, the Uncommitted Collateral, can be released toward the Advance Payment Pool at any moment, with a proof that what leaves was never spoken for. The reverse trip rides an ordinary settlement proof.

If there is a spike in new users, the Guardian releases a lump of Uncommitted Collateral toward the Advance Payment Pool, instantly and safely: no obligation loses its backing, and even a Guardian death the next moment strands nothing. The lump release is public while keeping individual advances private.

This allows Guardians to size their Working Capital to their transaction volume as a whole, keeping the vast majority in the Working Collateral, and instantly shifting funds to the Advance Payment Pool when the need arises.

D.5 Summary

Working Capital is the sum of the Working Collateral, the Advance Payment Pool, and the Interop Pool. No pool holds user funds; all three are the Guardian's.

Working Collateral, held in the Depository, covers the 'Guardian death' scenario. The Advance Payment Pool enables users to make private payments immediately after joining the protocol, without revealing their choice of Guardian. This balance is an account of the Guardian's own, because the risk it carries is the Guardian's alone: if an advance is never recovered (a deposit refunded before it was claimed), the loss falls on the Guardian and no one else. The Interop Pool, held on outside rails, bridges payments across the network's edge; its risks are the rails' risks, and they stop at the Guardian. The private pool advances new users' payments, the public pool backs the Guardian's signature, and the outside pool bridges the rails.

Appendix E: Multiple Asset Pools

The network is not limited to a single asset. The Depository can hold many, other currencies among them, and more than one version of the same currency, each backed one-for-one by its own asset pool. This appendix follows a payment that crosses from one asset to another, and the machinery that stands behind the crossing. All figures are illustrative.

E.1 A payment across assets

Alice holds dollars and owes George twenty of them. George lives in Europe, is served by Guardian G, and holds euros. At the current exchange rate, 20 dollars converts to 18 euros, so Alice's instruction fixes both figures: 20 dollars from her, 18 euros to George. Everything that follows executes at exactly those figures; neither side can be surprised by a rate.

Guardian A reads the instruction and sees that it crosses assets. It writes Alice down 20 on its own books and posts a message committing 20 dollars from its Advance Payment Pool to the Depository, to be settled to Guardian G. Guardian G reads the message and writes George up 18 on its own books, fronting the euros from its own Advance Payment Pool, the same fronting it performs for a new user in Appendix B. George's app confirms the 18 it was expecting. If the steps do not all confirm within a few seconds, the payment nullifies and costs nothing; otherwise it is complete, about a second after Alice sent it.

When the payment settles, two things happen: the 20 dollars are taken from Guardian A's Advance Payment Pool, with Alice's debit repaying the pool, and 18 euros are added to Guardian G's, repaying the credit it fronted to George. One job remains, and it is new. Dollar claims fell by 20 while euro claims rose by 18, so the Depository moves 20 dollars out of the dollar asset pool and 18 euros into the euro one, and every claim in every asset is again matched by reserves in that asset.

Everything is accounted for, except one thing. The exchange rate may have moved in the minutes between the payment and its settlement.

E.2 Two needs

The example leaves two needs behind it. The first is a single source of truth for exchange rates: Alice, George, and both Guardians all used the same 20-for-18, and every payment in the network must draw its rate from the same place. The second is a party to carry the movement of rates between payment and settlement: the payment locked 20 for 18, and if euros grew dearer in the interval, the 20 dollars collected no longer buy the 18 euros delivered.

The Depository serves both needs. It publishes the rates that payments bind to, and it takes the over or under when settlement arrives at different ones. The exposure is modest by construction: settlements run every few minutes, payments flow in both directions, and the differences largely cancel. What remains is small, and the Depository is funded to carry it.

E.3 The funding

Three streams fund the role.

The first is yield. Asset pools are held in yield-bearing form (§4.1), and a share of each asset's yield is directed to this machinery until it is funded.

The second is the conversion spread. A cross-asset payment converts at the published rate adjusted by a small spread, and the spread comes in two forms, chosen per payment: an immediate conversion at a wider spread, or a conversion delayed by about fifteen seconds at a considerably narrower one. The reason for offering two is treated in a subsequent document.

The third is trading fees, earned by liquidity pools that the Depository itself operates. The pools are the subject of the next section.

E.4 The liquidity pools

The published rates are made by trade. The Depository operates one liquidity pool per stable, and one common pool of ETH, each filled by the yield stream of E.3 until it holds enough to serve; an asset joins the cross-asset system only once its pools have filled. Anyone may trade with the pools, and every trade exchanges a stable for ETH, so ETH is the medium of exchange. The unit of account is not ETH but an index price: a weighted average of all the stables' prices, normalized, against which every price in the system, each stable's and ETH's, is measured. If USDC stands at an index price of 1.0 and ETH at an index price of 2,000, then 10,000 USDC costs 5 ETH. The settlement adjustment of E.1 moves through the same pools: the 20 dollars the dollar asset pool sheds enter the dollar liquidity pool, and the 18 euros come out of the euro one.

The two sides of a trade carry different fees, because they do different jobs. The stable side always charges 0.1 percent, paid in the stable itself. A stable's price moves with its trade volume, so whenever a stable drifts more than 0.1 percent from the wider market, arbitrage profits by trading it back; each stable therefore tracks the market to within 0.1 percent, and the fees collect in the very asset the pool trades. The ETH side charges either nothing or 0.5 percent, in both directions, according to which price the trading is adjusting. When one stable is out of line with the index, trade in its own pool corrects it, and ETH's price has no reason to move: the ETH side charges nothing. When the stables all sit at their index prices and the

flow persists anyway, it is ETH's own price that is out of line: the ETH price adjusts, and the ETH side charges 0.5 percent, paid in ETH.

E.5 The Backup Pools

The income does not sit in the liquidity pools. For every asset, and for ETH, the fees and the conversion spread accumulate to the side, as that asset's Backup Pool. The Backup Pools do three jobs. They absorb the rate movements the Depository carries between payment and settlement (E.2). They absorb impermanent loss: a pool that sells an asset through a sustained rise in its price is left holding less of it, and the Backup Pool makes up the difference. And they stand behind the liquidity pools themselves: a pool that runs dry can be refilled from its Backup Pool, under published controls and the oversight of an operational council.

The stables' Backup Pools stay small, because stables move little against the index. The ETH Backup Pool is deliberately large: ETH's price can multiply, the ETH pool sells ETH all the way up, and the Backup Pool must hold enough ETH, collected from the 0.5 percent fees, to restore what was sold.

If an asset's pools fall below their floor, the network stops accepting new cross-asset payments in that asset until they refill. Payments within the asset are untouched. The pause is not a new state; it is the state every asset starts in, before its pools first fill.

E.6 When the pools are full

A Backup Pool has a threshold, past which it holds more than its jobs require. Income arriving above the threshold flows through to the Guardians and the insurance providers, on top of the yield share of §4.1, and this is the expected steady state: the spread and the fees are sized to exceed the variances they absorb, so that over time the cross-asset machinery adds to the network's income rather than drawing on it. The buffers fill first; the surplus belongs to the parties who run and secure the network.

E.7 Summary

Each asset is fully reserved in its own asset pool. A cross-asset payment fixes its figures at the start, completes in about a second, and settles against the Depository, which converts between the asset pools at the rates the payments used. The liquidity pools make the rates, the spread and the fees fund the risk, and the Backup Pools absorb what remains. When the buffers are full, the income belongs to the Guardians and the insurers.

Appendix F: Reserve Risk and the Insurance Pool

The body of this paper states that the reserve carries the risks of its instruments (§4.1). This appendix specifies who chooses the composition of the reserve, who bears its losses, and in what order, together with the dynamics of the insurance economy in ETH: the conversion of premiums, the allocation of a payout, and the sizing of each pool. It is written at the level of §3: precise enough to constrain an implementation, without fixing engineering choices. All figures are example figures.

F.1 Allocation follows the first loss

The registry fixes, for each asset, the list of instruments its reserve may hold, and that list is a governed parameter behind the standard timelock (§8). Composition within the list is neither governed nor chosen by users; it is allocated by the capital that bears the first loss. Each Guardian's book is backed by its own reserve sleeve, and the sleeve's composition is allocated pro-rata by the contributors to that Guardian's insurance pool: the Guardian through its co-fund, and the underwriters through their stakes. In the example figures, a pool funded 15 percent by the co-fund gives the Guardian discretion over 15 percent of the sleeve, and the underwriters discretion over the remainder.

The premium half of the reserve yield (§4.1) accrues pro-rata to the same pool capital, so the parties that allocate the sleeve are paid from what it earns and stand in front of what it loses. An underwriter earns roughly half the yield on its share while bearing the first loss of that share, and therefore leans toward instruments that survive scrutiny. A Guardian that prefers a more ambitious sleeve must fund more of the pool itself, taking a larger share of the premium stream and of the first loss with it. The co-fund fraction above the floor of F.5 is therefore a business choice rather than a protocol parameter: at one end, a capital-light Guardian rents most of its balance sheet and accepts the conservatism of the capital it rents; at the other, a fully self-insured Guardian keeps the entire yield and the entire first loss. The protocol requires no equilibrium between them. What every allocator shares, at every fraction, is the preference for a diversified sleeve, since diversification bounds the loss that any single instrument can force on capital that cannot escape bearing it.

F.2 The loss waterfall

When an impairment of a sleeve's instrument is recognized, losses are met in a fixed order: first the Guardian's insurance pool, co-fund and underwriting stakes alike, and only past its exhaustion, the balances of that Guardian's users. Compensation from the pool is allocated by the rule of F.4, smallest balances first, and under the same discipline: the pool pays what it holds, and no account carries a fixed promise. The losses of one Guardian's sleeve never reach another Guardian's users, and a Guardian's Working Capital (Appendix D) stands outside this waterfall entirely.

The honeypot is never consumed by reserve losses. The standing bounty of §5 survives any depeg, so the dataset remains guarded precisely when a Guardian's attention is most likely to be elsewhere. A depeg that exhausts a pool therefore leaves a Guardian uninsured but not undefended. Such a Guardian may continue to operate: the protocol imposes no minimum pool, and the coverage figure of §4.3 reports its condition to every user who cares to read it. Coverage is the market's quality signal rather than a license condition, and a pool emptied by a depeg is repriced by the same market that filled it.

Two limits are stated for honesty. The pool is denominated in ETH (§4.2) while an impairment is denominated in the sleeve's currency, so the protection a pool provides against reserve losses floats with the price of ETH, exactly as its protection against breaches does. And underwriting capital stretched across many pools (§4.2) can be called by many pools at once when an instrument fails everywhere it appears, so coverage is strongest against the failure of one issuer among many and thinner against the failure of an instrument the whole market shares. That is the ordinary limit of insurance built from finite capital, and one more reason every allocator prefers the diversified sleeve.

F.3 The insurance economy in ETH

The premium stream and the underwriting capital live in different units by construction: premiums are a share of reserve yield and accrue in dollars, while stakes and coverage are denominated in ETH (§4.2). Income requires no matching, only a choice of where to convert. An underwriter elects, per position, to receive premiums in dollars, to convert them at the published rates (Appendix E), or to convert and compound them into its stake, increasing its coverage capacity. Compounding has a counter-cyclical character: a dollar stream purchases more ETH when ETH is inexpensive, so underwriters who compound accumulate units fastest in the markets in which the sizing discipline of §4.2 is under the most pressure.

The same two units make the pool self-sizing. The premium budget is fixed in dollars by the reserve, while the capital it compensates is priced in ETH, so the yield on stake rises when ETH falls and falls when ETH rises. Capital therefore enters in declines, when coverage is cheapest to add, and exits in advances, when less of it is needed against dollar-referenced requirements. No governance action sizes the pool; entry and exit against a fixed budget do.

F.4 Allocating a payout

When a proof establishes a breach (§5.6), the pool is allocated progressively. Each affected account is measured by its average balance over the period covered by the stolen data, an average provable from the per-epoch commitments the chain already holds, so the measurement cannot be reshaped after the fact. Compensation pays one half of that average balance, to the smallest accounts first and in ascending order, until the claim's proven share of the pool is exhausted. No party need hold the full list to run the ordering: each account proves its own average balance against those commitments and enters a claim, and after a claim window the pool pays the proven claims, smallest first.

The rule makes no fixed promise to any individual account, which is what lets the pool float in ETH without a broken promise anywhere in it: the size of the pool determines how far up the distribution compensation reaches. The figure a user compares across Guardians is the pool per user (§4.3). Small holders, who are least able to protect themselves against the consequences of exposure, are paid first and most fully. The ordering also removes the payout stream as an attack motive: capturing a meaningful share of it would require many small identified accounts, at a cost out of proportion to the shallow amounts collected, while large balances sit past the point of exhaustion.

F.5 Sizing the pool

Guardians size their own pools. A Guardian posts its co-fund and the honeypot, and underwriting capital fills the remainder of the pool at market prices, subject to the co-fund remaining at least 15 percent of the whole, a floor rather than a target: a co-fund share above 15 percent only strengthens deterrence. The honeypot is stated as a fraction of the pool, 5 percent in the example figures, and keeps that fraction as the pool is resized; a Guardian that funds more of its own pool owes no larger honeypot for it.

Withdrawals are governed by a ratchet stated in dollars. Every addition of capital raises a baseline, recorded as the pool's dollar value immediately after the addition, and the baseline never decreases. Capital may be withdrawn only while the pool's dollar value remains above both that baseline and an adequacy floor tied to the value of the data the pool insures, measured after an assumed severe decline in the price of ETH (§4.2). Declines in the pool's dollar value caused by the price of ETH alone are permitted

and never force a contribution; competition supplies the pressure to add coverage when it is cheap, since pool sizes are public. Every withdrawal, by a Guardian or an underwriter, completes only after a notice period longer than the claim game of §5.3 requires, and no withdrawal completes while a claim is open, so capital cannot leave between a breach and its proof.

Taken together, the flows of this appendix share one direction: premium compounding, underwriter entry, and co-fund additions all purchase ETH when it is inexpensive, and capital releases when it is dear. The security layer accumulates units in declines and releases them in advances.

F.6 Recognition and attribution

Two mechanics are specified with the implementation rather than here. Recognition is the rule by which an instrument's market loss becomes the ledger's loss; it must be objective, prompt, and anchored to references outside the protocol's own venues. Attribution is the treatment of accounts that change Guardians in the interval between an impairment and its recognition; the constraint it must satisfy is that a switch settles at recognized value, so that leaving early confers no advantage over staying, and the first-mover dynamic that afflicts conventional funds finds no purchase here. Both are implementation surfaces of the principle this appendix enforces: losses land on the capital that chose them, in public, and never by surprise.

Appendix G: Supervised Accounts and Lawful Process

This appendix specifies the account classes summarized in §3.8: the two supervised forms, the treatment of supervised accounts when their Guardian fails, and the manner in which authorities locate the Guardian that serves a person.

G.1 The two supervised forms

A supervised account permits a lawful hold, posted as an on-chain attestation that references the legal process behind it, and comes in two forms. Under portable supervision, an active hold blocks withdrawal to Ethereum but leaves the user free to move to another Guardian on the same jurisdiction's approved list, with the hold traveling to the successor, which inherits the obligation or declines the transfer. Under resident supervision, an active hold also fixes the account at the Guardian that received the legal process, which is the behavior some jurisdictions require of their institutions. Absent an active hold, a supervised account transacts, switches, and withdraws normally.

Three rules keep supervision honest. First, holds are attestations, public and auditable, so a lawful hold is distinguishable from an operator that has merely stopped responding, and holds expire unless renewed by fresh process. Second, no account is ever reclassified without its holder's signed consent: a jurisdiction that newly requires supervision obtains it for new accounts, and existing accounts retain the terms under which they were opened. Finally, the on-chain footprint of every class is identical, so an observer cannot distinguish account classes from public data.

G.2 When a supervising Guardian fails

Holds are enforced by the protocol's verifier rather than by the Guardian's continued existence, so a failure never erases a hold; the account instead degrades by one class, to the strongest promise that still

has a living enforcer. A resident-supervised account whose Guardian is confirmed delinquent becomes portable, and its holder may move to another approved Guardian, the hold traveling as usual. A portable-supervised account that no approved Guardian will serve regains the self-custody backstop after a waiting period that each jurisdiction sets for itself and publishes alongside its approved list; the holder sees that figure when the account is opened. The waiting period exists so that a hold cannot be defeated by briefly disabling a jurisdiction's Guardians: release requires the entire approved list to remain unavailable for the full period, in public view, while the authority may license a replacement at any time. Accounts without active holds are unaffected throughout; they exit or switch through the ordinary paths.

G.3 Locating the right Guardian

Locating the right Guardian is the same problem as locating the right bank, and it has the same two solutions. The default is the broadcast inquiry used in existing finance: the authority sends an identity to the Guardians approved in its jurisdiction, each checks its own records, and any Guardian with a matching relationship responds and takes the process from there. Nothing on-chain is consulted, because identity exists only in Guardians' records. A jurisdiction that prefers faster lookup may require its Guardians to file customer relationships in a registry it operates, as several banking systems already do. It is worth noting that holding identities at several Guardians (§3.3) evades neither mechanism, since every Guardian identified the person; separate identities limit what any single operator sees, not what lawful process can reach.