Nostrades

Binding chain-native atomic settlement to a Nostr-native market layer

Illuminodes

technology@illuminodes.com

2026 — v0.1 — working draft

Abstract. Decentralized finance made settlement trustless and left markets intermediated. We propose **Nostrades**, a pattern that builds the missing market layer on the Nostr protocol. An atomic swap is a resting limit order; a Nostr keypair is a portable identity carrying reputation across chains; a relay is a censorship-resistant venue where orders aggregate. Joined, they form a complete market with no operator at its center — settlement on the chain, identity on Nostr, liquidity on relays. The construction needs no new chain and no new primitive: most networks already ship atomic swaps. We give the thesis and the protocol shape, and argue it generalizes to any chain with an atomic-swap primitive.

Keywords: *atomic swaps, Nostr, decentralized exchange, market microstructure, self-custody, web of trust, Liquid Network*

---

1 Introduction

> **Thesis**
> Nostrades is the binding of chain-native atomic settlement to a Nostr-native market layer. It is a *pattern* for building non-custodial exchange where identity and reputation are portable across chains, and liquidity venues are open, resilient, and optionally private.

Decentralized finance disintermediated settlement. Atomic swaps, smart-contract escrow, and on-chain finality let two strangers exchange value with no custodian in the middle. The market that surrounds settlement stayed intermediated. Everything a trade depends on before value moves, from finding a counterparty to pricing it to judging who is across from you, still routes through a centralized exchange, an automated market maker, or a single-operator order book. We settle trustlessly a trade we found, priced, and vetted through a trusted party.

A market performs four functions: counterparty discovery, price formation at a venue, identity and trust, and the value exchange itself. DeFi delivered the fourth and left the first three to the intermediaries. A blockchain can record an address and a balance, but it is not suited to record an actor with a history. It can finalize a transfer, but it gives offers nowhere to rest, aggregate, and form a price. The three missing functions are an *identity* problem and a *venue* problem, and the chain answers neither.

Nostrades supplies both. The pieces a market needs beyond settlement already exist as native primitives of the Nostr protocol, and they bind directly to chain settlement. An atomic swap is a resting limit order. A portable keypair with a social graph and reputation is the identity a market runs on. A relay is the venue where orders live, held honest by the incentive logic that secures Bitcoin's miners. Three observations carry the whole construction:

Observation	What it means	Pillar
Swaps are orders	A half-signed atomic-swap proposal is a firm, takeable, self-describing limit order. Settlement and the order are one object, and from it yield, arbitrage, and cross-chain routing follow directly.	§2
Identity is a market primitive	Markets run on reputation, relationships, and recourse. A Nostr keypair carries a portable identity, its web of trust carries reputation, and its encrypted channels carry private negotiation.	§3
Relays are venues	Replaceable events are a live order book. Censorship resistance is venue resilience. Authenticated relays are dark pools. Micropayment-aligned incentives keep relays honest, as mining incentives keep Bitcoin honest.	§4

Table 1: The three observations behind Nostrades.

Nostrades makes the swap-as-order mapping explicit, names identity and venue as the market layer, binds each to a native Nostr primitive, and generalizes the construction across settlement chains through a portable identity that carries one reputation everywhere it trades. The result is a complete market that no operator controls.

The rest of this paper develops each observation, grounds the pattern in a working implementation on the Liquid Network, and extends it to any chain with an atomic-swap primitive. Nostrades introduces no new consensus mechanism, no new chain, and no token. It returns to markets the three functions settlement alone leaves out.

2 The Atomic Swap as an Order

> **Pillar I**
> An atomic swap is a resting limit order. Read the swap as an order and the vocabulary of market microstructure, makers and takers and the book and the fill, applies to a fully non-custodial system.

An atomic swap exchanges two assets so that both legs settle or neither does. No moment exists in which one party holds both. Hash-and-time-locked swaps span two chains, where a secret reveal unlocks both legs within a time bound, while single-transaction co-signed swaps place both legs in one transaction, where atomicity is intrinsic. Most networks already ship a primitive in one of these families, so we build on what exists rather than introduce a new one.

We propose reading a half-signed swap proposal as a firm limit order. The maker selects an input, names the asset and amount it wants in return, signs its own side, and publishes the result. That object is a standing offer at a price: takeable, self-describing, and binding on the maker's terms. A taker that completes and broadcasts it executes the trade. The order and the settlement are the same object, and every role in a classical market has a direct counterpart.

Market role	Market function	Atomic-swap action
Maker	Posts a limit order, provides liquidity, sets terms	Builds and half-signs a proposal, then publishes it
Taker	Consumes an order, demands liquidity, accepts terms	Validates, completes, and broadcasts the proposal
The order	A firm, resting offer at a price	The signed half-complete proposal
The fill	Execution and settlement	Atomic on-chain settlement of both legs
The cancel	Pulling the order from the book	Spending the input, or publishing a cancellation

Table 2: Market microstructure mapped onto the atomic swap.

What makes this order different from its counterparts on other venues is that the maker holds unilateral custody over it. Limit orders on a centralized exchange and positions in an automated market maker share the same flaw of locking capital. A swap proposal is instead a cryptographic object the maker keeps, signed only on its own side and revocable until taken. The order is self-enforcing, because no venue can fill it on terms other than the ones it carries and no taker can take the maker's asset without delivering the counter-asset. The worst outcome of a trade is that it does not happen. An offer can go stale between being seen and being taken if the maker cancels or the input is spent elsewhere, and the fill then simply fails with no funds moved. This reduces the risk a non-custodial book normally carries.

The mapping has real limits, stated plainly. Single-input proposals do not partially fill without added machinery. The maker must hold an input matching the offer size. No global matching engine exists, so discovery belongs to the venue and matching to the taker. These are microstructure constraints, not settlement flaws, and they set the agenda for the venue pillar in §4.

2.1 What the order book produces

Yield, arbitrage, and routing are natural consequences of the order being a real, atomically-settled object. Once offers rest at fixed prices and anyone can take them, the dynamics that make mature markets efficient appear on their own and reward the participants who supply them.

A liquidity provider is a taker that fills offers priced below its target and rests the inventory back out at a price it chooses. Its yield is captured spread, earned on assets it actually holds: there is no pool, no locked capital, and no impermanent loss, only the inventory and price risk it elects to take, never settlement risk. Market making becomes a strategy any participant can run rather than a privilege the venue grants.

Arbitrage is the same act seen from the other side. Two offers that cross in price are a standing, riskless-settlement profit, taken by buying both and pocketing the difference atomically. The same logic spans venues and chains, since one pair quoted differently on two relays, or one asset priced differently on two chains, is the same opportunity at a larger scale. Each arbitrage taken tightens the spread and aligns the book, so the market corrects itself and pays the corrector to do it.

Routing extends the act across networks. Because offers for many pairs and chains can rest on one relay, an agent can chain swaps, buying through intermediate assets to reach a destination it has no direct market for. A taker threads a path the way a payment threads a route through channels, except the legs are cross-asset atomic settlements. Moving liquidity to where it is scarce becomes a business in itself, and the venue becomes a routing fabric for the whole ecosystem. We develop these mechanics in §5 and §7.

Together these make a market that funds its own efficiency with no operator running the mechanism. What the structure cannot produce is everything the chain leaves out. The order knows its price but not its maker. Nothing in the swap tells a taker who is on the other side, whether the offer is honest, or whether this counterparty has settled a thousand trades or none. A market runs on reputation, relationships, and recourse, and the swap supplies none of them. Settlement is solved. Identity is not. That is the next pillar.

3 Identity: the Missing Market Layer

> **Pillar II**
> A blockchain describes value, not actors. Markets run on actors: who they are, what they have done, whether to deal with them. That layer already exists in Nostr, which supplies a user-sovereign keypair, a portable social graph, and signed reputation, all bindable to settlement.

The chain gives a trader three things, and none of them is an identity. An address is a destination, a balance is a number, and a smart contract is a set of rules. Each is built for settlement and indifferent to history. None can say who a counterparty is, whether they have behaved before, or whether to give them terms a stranger would not get. The chain is settlement-complete and identity-empty, and every market function that turns on knowing the actor falls into that emptiness.

The layer that fills it already exists. A Nostr identity is a keypair with no account, server, or registry behind it, and every action is a small signed event served by interchangeable relays. What makes it a market identity rather than a social one is that it can be bound to the keys that settle. The binding is constructed rather than assumed, because settlement networks use their own key encodings and a Nostr public key and a chain address are not directly equatable. A trader establishes the link instead, either off-chain by having the Nostr key attest to an artifact only the settlement key could produce, or on-chain by publishing the Nostr public key in a settlement transaction. Either way the maker of a market becomes provably the same actor as the settler of a trade, and from that binding each capability a market needs beyond settlement maps to a primitive the identity already carries.

Market need	Nostr primitive
Persistent, sovereign identity	A keypair the user holds and no platform can revoke, pseudonymous rather than a legal name, and permissionless.
Reputation as a usable signal	Signed, append-only attestations of completed swaps and vouches, costly to fake and costly to abandon. A long-lived identity has something to lose, which stands in for custodial trust.
Trust without a rating agency	A web of trust: trust is transitive and attestable, so a maker is weighed by who, among those a taker already trusts, vouches for them.
Relationships and preferential terms	Follow and contact lists are a native, portable social graph. Makers quote trusted counterparties as an OTC desk would, permissionlessly.
One reputation across chains	A chain-agnostic key proves control of addresses on many chains and carries a single reputation across all of them.
Private negotiation	Encrypted and ephemeral events: terms a relay cannot read, off-book quotes, and indications of interest that leave no trail.

Table 3: What a market needs beyond settlement, and the Nostr primitive that supplies it.

Two rows carry more weight than the rest. Reputation that travels across chains is what a cross-chain market needs most and lacks most, since a maker reputable on one chain otherwise begins at zero on the next, yet the counterparty a taker must judge is usually the one on the other chain. Encryption resolves the apparent conflict between reputation and privacy: reputation is built from what an identity chooses to attest, not from the forced exposure of every order, so a reputable actor can still trade in private.

The risks are real and few. A lost key takes its reputation with it, a web of trust mitigates Sybil and vouch-spam without erasing them, and wash-traded attestations must be discounted by the trust computation. Each is addressed in the security and economic models. The gap itself is closed. The chain answers whether value can move. Nostr answers who is moving it. What remains is where the orders live.

4 Relays: the Venue

> **Pillar III**
> A market needs a place where orders rest, aggregate, and form a price. The usual answer is a server or a contract. A relay is a third option that fits the job natively: a live order book that no operator controls, kept honest by the same incentive logic that secures Bitcoin's miners.

A venue does four things. It aggregates orders into a discoverable book, disseminates updates to subscribers, persists resting orders until they are filled or pulled, and stays reachable. A relay does all four as a side effect of storing and serving signed events. An order is an event, an update or re-quote replaces the previous version, a cancellation empties it, and a subscription filtered by pair and time is a standing request for the current book, streamed live as it changes. Because every order is a self-contained signed object, the maker behind each entry is an identity, and the book is annotated with reputation by construction.

A relay is pure infrastructure. It has no Nostr identity of its own and never trades, because it is a venue rather than a participant. Relays do not talk to each other, so there is no network of relays to coordinate or capture. The book a trader sees is assembled client-side from whichever relays that trader connects to, the way a trader today watches several exchanges at once. This keeps the roles cleanly separate: a venue operator runs relays, a market participant posts and takes orders, and a liquidity provider can supply depth across many relays without running one. No participant has to become infrastructure, and the infrastructure never becomes a participant.

Venue property	Relay mechanism
Live order book	Replaceable events: orders rest, re-quote, and cancel as event updates, with no central matching engine pruning the book.
Resilience	Multi-relay publication: the book is replicated, not hosted, so no operator can halt trading, freeze a participant, or seize the market.
Neutral operation	The relay earns from flow, holds no funds, and runs no matching engine, so there is nothing to rig.
Private venues	Authenticated and encrypted relays gate access and hide content, giving dark pools and OTC desks over the same protocol.

Table 4: The venue's properties, and the relay mechanism behind each.

The order book and its resilience come for free, but they leave the hardest question about any open venue unanswered: not whether a bad relay can be escaped, but why a relay behaves at all. The escape is real, since an order published to many relays survives any one of them dropping or filtering it, so a malicious relay can be routed around. That is the backstop. The first line of defense is that a well-run relay does not want to misbehave, for the same reason that secures Bitcoin.

A Bitcoin miner need not be trusted or even known. It earns by including transactions and extending the honest chain, and attacking the network would destroy the value of what it earns, so honesty is the profitable strategy. A relay sits in the same position one layer up. It earns by carrying orders and serving the book, not by deciding who wins a trade. It holds no funds, because settlement is atomic, and runs no matching engine, because matching is the taker's choice. It is credibly neutral, indifferent to what trades and paid only for flow. Authenticated micropayments turn that neutrality into a live revenue stream, a

small charge to read or write the book. A relay that stays live and honest keeps earning from everyone who transacts over it. A relay that censors or stalls is routed around and loses the stream. Defection is a self-inflicted revenue cut, so the dominant strategy is to keep relaying, exactly as a miner keeps mining. Liveness is what the micropayments buy. Safety never depends on any single relay.

The same authentication that prices access also partitions the market. A relay can require a participant to prove control of a key and gate the book by an allow-list, a web-of-trust threshold, or a subscription, while encrypted order content lets it carry terms it cannot itself read. Together these span the full range of venues a real market uses, from a lit public book through a web-of-trust room to a dark pool where large makers post size to vetted takers with no public market impact. The venue's policy and disclosure level change across that range. The order and the settlement underneath do not.

This is where relays compete, and where the infrastructure becomes a market in its own right. A free public relay establishes the floor: anyone can run one and the book survives on it. Specialized relays differentiate above that floor on the qualities a serious trader pays for, including lower-latency connections, deeper caching and faster book reconstruction, gated venues for vetted flow, and data services such as reputation scoring or verified-identity attestation. None of this compromises neutrality, because a faster or better-curated relay is still only carrying orders and serving the book. It is the competitive advantage the model has over both centralized exchanges, which offer one venue on the operator's terms, and contract-based DEXes, which offer one venue on the chain's. Here the venue layer is an open market, and quality is something operators sell rather than something a single operator withholds.

The obvious objection is that many relays mean fragmented liquidity, but the fragmentation is of hosting, not of the market. An order is the same valid object on every relay it reaches, so a client aggregates across relays into one view and a maker multi-posts to widen reach without splitting depth. Liquidity that looks scattered is in fact replicated, the opposite of the siloed capital that sits in separate exchanges and pools and cannot be unified without a custodial bridge. The venue becomes a commodity rather than an institution, and the last intermediary leaves the market.

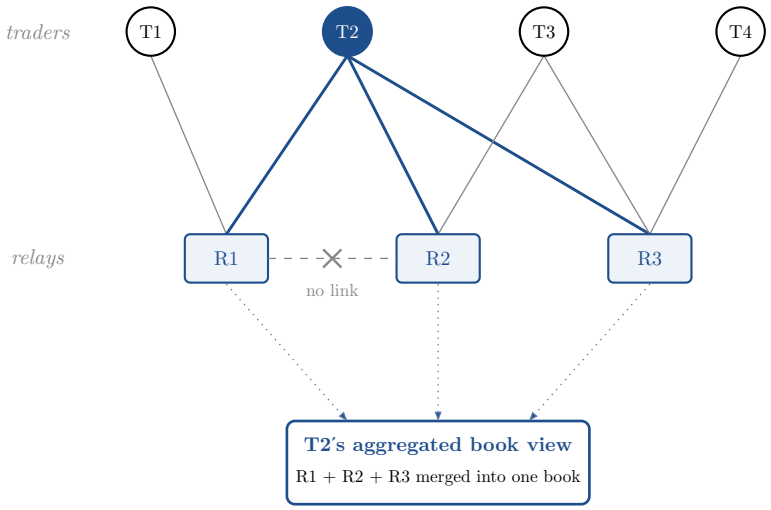

Figure 1: Relay topology. Traders connect to one relay or several, and relays are shared across traders. Relays never connect to each other. A trader's client reads its chosen relays and merges them into one book.

5 Implementation and Generalization

Synthesis

Nostrades is a pattern, not a product: a signed atomic-swap order, a Nostr identity that carries reputation, and a relay venue. Only the settlement slot is chain-specific. A working implementation on the Liquid Network fills all three and stands as the worked example for every chain that can fill the first.

The three pillars compose into a single shape. An order is a signed atomic-swap proposal, the maker behind it is a Nostr identity with portable reputation, and the venue is a set of relays carrying those orders as replaceable events. Identity and venue are the same on every chain. Only the settlement primitive that gives the order its atomicity varies, which makes the pattern a template. Implement it once concretely and the concrete case shows what every other case must supply.

The reference implementation does exactly that on Liquid. An order is a half-signed transaction in which the maker commits its own input and desired output, signs its side, and leaves the counter-leg open, which is the resting limit order of §2 made literal. That proposal is serialized into a Nostr event and published to relays, where it joins a live book that any client can subscribe to and any taker can complete and broadcast for atomic settlement. The maker's Nostr key signs the order and a key it controls on Liquid settles it, bound by the construction of §3. Cancellation works both ways the model predicts: a signed event retracts the offer from the book, and spending the committed input retires it on-chain regardless of what any relay still shows. Every element of the thesis appears in running code, against real settlement.

Liquid earns the reference slot because its atomicity is the cleanest available. A swap is a single transaction with no timelocks and no secrets, amounts are confidential, and many assets share one chain, which makes it the easiest place to prove the mechanism. None of it is load-bearing for the pattern. The order is whatever the chain's atomic-swap primitive produces, and the families that qualify are already deployed across the ecosystem.

Primitive	**What it offers the order slot**
Single-transaction co-signed swaps	Both legs in one transaction; intrinsic atomicity, no timelocks. The reference case on Liquid.
Hash-time-locked contracts (HTLCs)	Cross-chain swaps between UTXO networks; atomicity enforced by a secret reveal within a time bound.
Adaptor signatures / scriptless scripts	Atomic swaps that hide the swap in an ordinary signature, giving privacy at the settlement layer.
Lightning and PTLCs	Instant, off-chain legs for low-value, high-frequency orders.
Statechains, Ark, and other layers	Emerging Bitcoin-aligned primitives that produce a takeable, atomically settled order.

Table 5: Settlement primitives that fill the order slot.

Each primitive changes how an order settles and nothing else. The event that carries it, the identity that signs it, and the relays that host it are unchanged, so a new chain joins by supplying a single adapter and inherits the identity and venue layers whole. This is where a Liquid exchange becomes an ecosystem-wide market layer, and the value compounds rather than adds. A maker earns reputation once and carries it to every pair on every chain it quotes. A taker's web of trust spans chains, so counterparty evaluation is unified across the swap boundary where it matters most. A book aggregated across relays is indifferent to which chain any given order settles on.

The compounding goes further still through the cross-chain routing of §2.1. More chains on shared relays means more paths between assets, which means more reachable liquidity, which draws more chains. The routing fabric is a second-order network effect on top of the shared book. Every other venue today is an identity and liquidity silo bridged only by custodians. Nostrades lays one market layer horizontally across a vertical stack of settlement chains, and keeps it non-custodial the whole way across.

6 Security and Trust

> **The claim, precisely**
> Nostrades removes the trusted third party from the market, not from the universe. Funds are never at settlement risk. What remains is bounded and named: the chain a trade settles on, the user's own keys, and the residual risks of availability, privacy, and identity loss.

The honest way to read a non-custodial claim is to list what still has to be trusted. Three things do. The first is the settlement chain itself, whether Liquid's federation or Bitcoin's proof-of-work behind an HTLC, the one irreducible trust root, inherited from the chain rather than introduced by Nostrades. The second is the user's own key custody, which moves the security boundary to the user and is both the point and the exposure. The third is a single honest, reachable relay, needed only for liveness and never for safety. Standard cryptographic assumptions sit underneath all three. Nothing else is trusted, not an exchange operator, a matching engine, a custodian, or any particular relay.

Against that short list, the construction protects what matters most without asking for trust at all. Settlement is atomic, so a taker cannot take the maker's asset without delivering the counter-asset, and both legs clear or neither does. Orders are signed, so a venue cannot alter their terms. Identities are signed, so impersonation requires stealing a key rather than spoofing a name. And the chain is the source of truth, so a taker re-validates an offer against current chain state before filling it, and an offer whose input is already spent simply fails to fill with no loss. The worst an adversary can usually achieve is to waste a taker's attempt.

The adversaries worth modeling are the four roles the system actually has, and each is more constrained than its counterpart on a custodial venue.

Actor	Can do	Cannot do	Bounded by
Relay	Withhold, filter, or reorder events	Forge orders or move funds	Multi-relay redundancy, and a micropayment incentive to stay neutral
Maker	Post stale or unfillable offers to waste takers' attention	Take funds	Reputation cost, and chain re-validation
Taker	Race to take an offer the maker is cancelling	Settle without delivering its leg	Hard cancel: spending the input retires the offer on-chain
Observer	See public orders and the identities behind them	Read encrypted terms	Encrypted and ephemeral events, private negotiation, one-shot keys
Sybil	Flood the book with fake identities and offers	Manufacture genuine trust	Web-of-trust gating, relay policy, reputation at stake

Table 6: Threat model by actor.

No actor in that table can take a counterparty's funds, which is the property the whole construction exists to guarantee. What survives is a residual list, stated plainly because a non-custodial market is only as credible as its honesty about the edges. The sharpest is availability. An offer on the book may already be gone, and the attempt to take it fails, but this is execution uncertainty rather than loss, and re-validation, short subscription windows, and on-chain cancellation keep it small. Privacy and reputation pull against each other, since a persistent identity that earns reputation is linkable by design, and encryption mitigates this without erasing it, at the cost that a fully private trade earns no reputation. Key loss is the hard self-custodial edge, made sharper because a lost key takes its reputation with it. A relay learns metadata

even when it cannot read content. And the settlement chain carries its own trust model, peg and liveness included, which Nostrades inherits rather than fixes. None of these is a path to stolen funds, and each is addressed where it belongs, in the economic model and the open problems that follow.

7 Economics and Market Design

> **Synthesis**
> A trustless mechanism still needs incentives. Nostrades pays its own participants out of the activity they create, so alignment comes from the market itself rather than from a subsidy. And there is no token by design: a protocol token would be the one intermediary the rest of the paper removes.

Once no operator collects rent, the economic question is who is paid and by whom. The answer is that the market pays its own participants, and this matters most at the cold start, where a two-sided market stalls because no makers means no takers means no makers. Nostrades attacks cold start with yield rather than subsidy. Providing liquidity is itself a yield strategy (§2.1): a maker earns the spread it sets on its own terms, with no platform taking custody or a cut, and reputation compounds the return as better behavior earns tighter quotes and more flow. The draw is profit, not emissions, which is the strongest answer to cold start. Liquidity providers arrive because the market pays, and it pays from the first trade.

The same structure rewards the participants who keep prices honest. Arbitrage and routing are profitable acts that leave the market more efficient than they found it (§2.1), so the system never pays for efficiency. It is the by-product of letting people profit.

Relays are paid for service rather than control. A relay earns by carrying orders and serving the book, and authenticated micropayments make that a live revenue stream collected from everyone who transacts over it. Stay live and neutral and the stream continues. Censor or stall and the relay is routed around and the stream stops. The incentive is settled even where the mechanics are not: which rails carry the micropayments, how posting is priced to deter spam, and who funds the first relays before flow exists all remain to be shown.

That leaves how the Nostrades project itself earns without becoming the intermediary it removes. Several non-custodial routes exist, and the choice among them is deliberately open.

Route	How it earns
Operating relays	Run neutral-infrastructure relays and collect the micropayment stream as a first-party business.
Premium and private venues	Host gated, encrypted dark-pool relays for participants trading in size.
Routing and market-making	Run specialized agents that earn spread on arbitrage and cross-chain routing.
Optional swap fee	A small fee output carried in the swap itself, paid by maker or taker by convention.
Identity and reputation services	Attestation and trust infrastructure built on the same identity layer.

Table 7: Non-custodial value-capture routes (under design, not decided).

7.1 What lives above the protocol

The routes above describe businesses built on Nostrades, not features of it, and the distinction is load-bearing. The protocol is the order, the identity, and the venue. Matching engines, automated market makers, pricing oracles, and custody all belong in the application layer above it, where anyone can build them and compete.

This is the answer to the common objection that a centralized exchange executes a trade at a better price. It does, but not because it is a better venue. It executes better because it is also a market participant and a custodian: it holds the user's funds and fills against its own book on the user's behalf. A venue does neither. The gap is not a protocol flaw to be closed but a service to be offered, and the service is an application. An active matching engine can watch many relays and chains, match favorable trades, and fill the spread as an arbitrageur, giving users price discovery and tighter execution. Its edge thins as it works, because every trade it takes narrows the spread it lives on, which is the market doing its job rather than a rent being collected.

Custody is the same kind of choice, made one layer up. The protocol is non-custodial, but applications on it can offer custodial convenience without becoming the exchange this paper argues against, provided they hold a separate key per user rather than one database of everyone's funds. Pre-signed transactions let a service execute on a user's behalf within limits the user set and signed. The right level of trust is a product decision that different applications will make differently, and the protocol's job is only to make the non-custodial floor the default rather than the exception.

Two pricing problems sit underneath all of this, and one lever solves both. Free identities invite Sybil flooding and free posting invites spam, so pricing the action with a micropayment, a proof-of-work tag, or a web-of-trust gate makes both expensive. Reputation at stake then makes griefing costly even when the post itself is cheap, because a maker that floods stale offers spends reputation it took real trades to earn. Misbehavior is paid for in the same currencies that reward good behavior.

The absence of a token is the design, not a gap in it. Every value that moves through Nostrades is denominated in an asset that already exists: trades settle in the chain's native units, fees are taken in what is being traded, and relays are paid in ordinary micropayments. A protocol token would add nothing to this and would quietly undo the argument. It is itself an intermediary, an asset every participant must acquire, hold, and trust, wedged between the trader and the trade. It concentrates control in whoever issues it, charges a rent the market does not need, and ties the venue to the price of a unit unrelated to any trade. A market that needs a token has moved the custodian, not removed it. Nostrades has none because the entire point is to have one fewer thing to trust.

8 Roadmap and Open Problems

> **Where this stands**
> The pattern runs today as an alpha on one chain. The distance from there to an ecosystem-wide market layer is real, and so is the list of problems that are genuinely unsolved. Both are stated plainly: a design is only as trustworthy as its account of what it has not yet earned.

What exists now is a working alpha: a single settlement chain, one asset pair, a public book on a single class of relay, and an identity derived from one seed. It proves the mechanism end to end and nothing more, which is exactly what an alpha should do. The path from there widens in stages, each one turning a property the thesis claims into something deployed.

Phase	What it delivers
Alpha (now)	One chain, one pair, a public book, single-seed identity. The mechanism proven end to end.
Hardening	Mainnet, partial fills, live pricing, continuous sync, and a consolidated core with robust cancellation and locking.
Identity and reputation	Signed swap-completion attestations, web-of-trust tooling, and reputation surfaced directly in the book.
Venue richness	Authenticated and dark-pool relays, first-class re-quote semantics, and relay discovery and aggregation.
Generalization	A second settlement chain through a new adapter, with identity and reputation shared across both.

Table 8: Roadmap from alpha to the ecosystem-wide layer.

The honesty the section owes is in what remains hard. Reputation has a bootstrapping problem: a new identity starts with none, trust must accrue before it can be relied on, and the same attestations that build it can be wash-traded to fake it. Earning genuine, Sybil-resistant reputation without a central authority or KYC is an open question. Privacy and reputation stay in tension, since one-shot keys trade in private but accumulate no history while persistent keys accumulate history but are linkable, and encryption narrows this gap without closing it. Relay economics are argued but unproven at scale: the incentive is sound, but the rails, the pricing, and the bootstrap of relays before flow exists are still to be demonstrated. And liquidity aggregated across many relays and chains must stay aggregated without a central aggregator quietly becoming the chokepoint the design set out to avoid.

Several problems are narrower but no less real. Partial fills need coin selection and order chunking the whole-order model does not yet provide, and they complicate the relay's liveness guarantee, which is clean for a whole order taken once but harder when an order is consumed in pieces. Wherever a single-transaction swap is not available, cross-chain atomicity has known edges, such as timeout griefing and the free-option problem in hash-locked swaps. The order and identity event formats should likely become a published standard rather than a private convention. And self-custody of an identity that carries reputation raises the stakes of key loss, which is a user-experience problem as much as a cryptographic one.

It is worth stating what would show the approach to be wrong, because a thesis that cannot be falsified is not a thesis. Nostrades fails if reputation never accrues enough to change behavior, leaving an anonymous book no better than the adversarial venues it replaces. It fails if relay incentives never materialize and the venue layer depends on altruism. It fails if liquidity never aggregates and the market stays too thin to price anything. These are empirical questions, and the staged roadmap exists to answer them in the order each becomes testable. None is answered by issuing a token, which is why there is not one.

9 Conclusion

Decentralized finance solved settlement and left the market behind. The gap it left is identity and venue, knowing who is on the other side and having a place where offers rest and form a price, and the argument of this paper is that both already exist, deployed, in Nostr. The work was not to invent them but to see that they fit.

They fit because each pillar is one object read correctly. A half-signed atomic swap is a resting limit order. A Nostr keypair, bound to the keys that settle, is a portable identity that carries reputation across every chain it trades on. A relay is a venue that no operator controls, kept honest by the same incentive that secures a miner. Bound together they make a complete market with no one at its center: settlement

on the chain, identity on Nostr, liquidity on relays, and no token wedged between the trader and the trade. Each pillar removes an intermediary, and nothing is added back.

This is a pattern, not a product. A working implementation on Liquid proves the mechanism, but nothing in identity or venue is Liquid-specific, and any chain with an atomic-swap primitive joins through a single adapter and inherits the same identity and the same book. The value compounds as chains join, because one reputation and one liquidity view span all of them. A Liquid exchange becomes an ecosystem-wide market layer laid across a stack of settlement chains.

Markets were never only about moving value. They were always about who you trade with and where. Settlement answered the first question and abandoned the other two. Nostrades returns both to the trader and keeps the whole of it trustless.